Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation Konnektor

Version:	5.14.0
Revision:	401206
Stand:	02.09.2021
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	gemSpec_Kon

Dokumentinformationen

Änderungen zur Vorversion

Anpassungen des vorliegenden Dokumentes im Vergleich zur Vorversion können Sie der nachfolgenden Tabelle entnehmen.

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
5.1.0	05.10.17		Initialversion Online-Produktivbetrieb (Stufe 2.1)	gematik
5.2.0	18.12.17		Einarbeitung Erratas 1.6.4-1 bis 1.6.4-3, P15.1	gematik
5.3.0	14.05.18		Einarbeitung P15.2, P15.4 und P15.5	gematik
5.4.0	26.10.18		Einarbeitung P15.8 und P15.9	gematik
5.5.0	18.12.		Einarbeitung P17.1	gematik
5.6.0	15.05.19		Einarbeitung P18.1	gematik
5.7.0	28.06.19		Einarbeitung P19.1	gematik
5.8.0	02.10.19		Einarbeitung P20.1/2	gematik
5.9.0	02.03.20		Einarbeitung P21.1	gematik
5.9.1	26.06.20		Einarbeitung P21.3	gematik
5.9.2	27.08.20		Einarbeitung P21.4	gematik
5.9.3	21.09.20		Einarbeitung P21.5	gematik
5.9.4	05.11.20		Einarbeitung P21.6	gematik
5.10.0	30.06.20		Einarbeitung P22.1	gematik
5.11.0	12.11.20		Einarbeitung Scope-Themen zu R4.0.1	gematik
5.12.0	09.12.20		Einarbeitung P22.5	gematik

5.13.0	30.06.21		Einarbeitung Konn_Maintenance_21.1, _21.2, _21.3 und gemF_gSMC-K_Laufzeitverlängerung	gematik
5.14.0	02.09.21		Einarbeitung Konn_Maintenance_21.5, Einarbeitung CI_Maintenance_21.2: redaktionelle Umbenennung von aAdG NetG in WANDA Basic, aAdG und NetG-TI in WANDA Smart	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die Anforderungen zu Herstellung, Test und Betrieb des Produkttyps Konnektor.

Dieses Dokument beschreibt die dezentrale Komponente zur sicheren Anbindung von Clientsystemen der Institutionen und Organisationen des Gesundheitswesens an die Telematikinfrastruktur – den Konnektor. Der Konnektor ist einerseits verantwortlich für den Zugriff auf die in der Einsatzumgebung befindlichen Kartenterminals sowie Karten und andererseits für die Kommunikation mit den zentralen Diensten der TI-Plattform und fachanwendungsspezifischen Diensten. Aus den Kommunikationsbeziehungen mit Clientsystem, Kartenterminals, Karten und zentralen Diensten der TI-Plattform und fachanwendungsspezifischen Diensten resultieren vom Konnektor anzubietende Schnittstellen, die gemeinsam in diesem Dokument sowie den fachanwendungsspezifischen Fachmodulspezifikationen normativ geregelt werden. Vom Konnektor genutzte Schnittstellen liegen zumeist in anderen Verantwortungsbereichen (zentrale TI-Plattform aber auch Schnittstellen der Kartenterminals und Karten). Diese werden in den übergreifenden Spezifikationen der TI sowie den Produkttypspezifikationen definiert.

Dieses Dokument regelt somit nur einen Teil des Konnektors (wenngleich auch den Wesentlichen). Für die Implementierung eines Konnektors ist entsprechend die Kenntnis aller weiteren Spezifikationen erforderlich. Die Gesamtheit aller für den Konnektor relevanten Dokumente wird im Produkttypsteckbrief des Konnektors erhoben.

1.2 Zielgruppe

Das Dokument richtet sich an Konnektorhersteller sowie Hersteller und Anbieter von Produkttypen (dies beinhaltet auch die Anbieter zur G2-Ausschreibung), die hierzu eine Schnittstelle besitzen.

1.3 Geltungsbereich

Dieses Dokument enthält normative Festlegungen zur Telematikinfrastruktur des Deutschen Gesundheitswesens. Der Gültigkeitszeitraum der vorliegenden Version und deren Anwendung in Zulassungsverfahren werden durch die gematik GmbH in gesonderten Dokumenten (z. B. Dokumentenlandkarte, Produkttypsteckbrief, Leistungsbeschreibung) festgelegt und bekannt gegeben.

Wichtiger Schutzrechts-/Patentrechtshinweis

Die nachfolgende Spezifikation ist von der gematik allein unter technischen Gesichtspunkten erstellt worden. Im Einzelfall kann nicht ausgeschlossen werden, dass die Implementierung der Spezifikation in technische Schutzrechte Dritter eingreift. Es ist allein Sache des Anbieters oder Herstellers, durch geeignete Maßnahmen dafür Sorge zu tragen, dass von ihm aufgrund der Spezifikation angebotene Produkte und/oder Leistungen nicht gegen Schutzrechte Dritter verstoßen und sich ggf. die erforderlichen

Erlaubnisse/Lizenzen von den betroffenen Schutzrechtsinhabern einzuholen. Die gematik GmbH übernimmt insofern keinerlei Gewährleistungen.

1.4 Abgrenzung des Dokuments

Spezifiziert werden in dem Dokument die von dem Produkttyp bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert.

Die vollständige Anforderungslage für den Produkttyp ergibt sich aus weiteren Konzept- und Spezifikationsdokumenten, diese sind in dem Produkttypsteckbrief des Produkttyps Konnektor verzeichnet.

1.5 Methodik

1.5.1 Anforderungen

Anforderungen als Ausdruck normativer Festlegungen werden durch eine eindeutige ID sowie die dem RFC 2119 [RFC2119] entsprechenden, in Großbuchstaben geschriebenen deutschen Schlüsselworte MUSS, DARF NICHT, SOLL, SOLL NICHT, KANN gekennzeichnet.

Sie werden im Dokument wie folgt dargestellt:

<AFO-ID> - <Titel der Afo>
Text / Beschreibung
[<=]

Dabei umfasst die Anforderung sämtliche innerhalb der Afo-ID und der Textmarke angeführten Inhalte.

1.5.2 Offene Punkte

Zum Zeitpunkt der Spezifikationserstellung konnten nicht alle Details abschließend geklärt werden, insbesondere, da Abstimmungsbedarf mit der umsetzenden Industrie besteht. Details, die keine produkttypübergreifenden Auswirkungen haben und die im Rahmen des Verhandlungsverfahrens mit der Industrie besprochen werden müssen, werden als „Offene Punkte" ausgewiesen und wie folgt im Dokument kenntlich gemacht:

Die XYZ müssen noch definiert werden.

1.5.3 Erläuterungen zur Spezifikation des Außenverhaltens

Der Konnektor stellt einen vergleichsweise komplexen Produkttyp dar, dessen Beschreibung eine Herausforderung darstellt und somit in vielen verschiedenen Varianten

möglich wäre. An dieser Stelle folgen daher wesentliche Informationen, die das korrekte Verstehen der Spezifikation fördern:

Die Spezifikation des Konnektors ist eine Black-Box-Spezifikation, das heißt alle Festlegungen dienen ausschließlich der Beschreibung des von der Komponente verlangten Verhaltens an der Außenschnittstelle.

Normative Festlegungen, die eine Festlegung des inneren Verhalten vermuten lassen (beispielsweise die Definitionen der Technischen Use Cases - TUCs) sind nur in so weit normativ, wie ihre Festlegungen auf die Außenschnittstelle wirken. Sie legen explizit nicht die intern zu verwendende Implementierung fest. Die Notwendigkeit für diese Art der „scheinbaren internen Beschreibung" ergibt sich aus der Komplexität der Gesamtkomponente, sowie dem Bedarf, wiederholt ähnlich Verhaltensweisen in Außenschnittstellen darstellen zu müssen. In diesem Fall werden die sich wiederholenden Verhaltensanteile in internen TUCs zur editoriellen Wiederverwendung gekapselt. Die konkrete konnektorinterne Modularisierung bleibt dem Hersteller freigestellt. Insbesondere bleibt es dem Hersteller freigestellt, intern bereits Mechanismen für kommende Releases zu realisieren, sofern diese an der Außenschnittstelle keine Auswirkung zeigen.

Die einzige Abweichung von dieser Vorgehensweise ergibt sich für Sicherheitsaspekte. Hier können interne Vorgänge normativ gefordert sein, die sich an der Außenschnittstelle nicht manifestieren (Beispiel „Verpflichtung auf sicheres Löschen eines temporären Schlüssels nach Gebrauch"). In diesem Fall erfolgt die Überprüfung der Einhaltung dieser Anforderungen im Rahmen der CC-Evaluierung.

1.5.4 Erläuterungen zur Dokumentenstruktur und „Dokumentenmechanismen"

1.5.4.1 Modulare Spezifikation über Funktionsmerkmale

Die Beschreibung des Konnektors erfolgt soweit wie möglich modular, d. h. alle Aspekte, die für einen logischen Bereich relevant sind, werden in einem Kapitel beschrieben. Diese logischen Bereiche werden als Funktionsmerkmal bezeichnet.

Funktionsmerkmale kennzeichnet ein eigener Verantwortungsbereich. In diesen Verantwortungsbereich greifen keine anderen Funktionsmerkmale ein. So kann ein logischer Bereich vollständig durchdrungen werden, ohne dass in anderen Kapiteln Anforderungen zu erwarten wären, die das Verhalten des Funktionsmerkmals beeinflussen. Da zwischen Funktionsmerkmalen Wechselwirkungen bestehen (Die Erkennung einer gesteckten Karte im Kartenterminaldienst löst eine Reaktion im Kartendienst aus), wurden zur „dokumententechnischen Interaktion" zwischen Funktionsmerkmalen ein interner Event-Mechanismus sowie Konfigurationsparameter und Zustandswerte eingeführt (siehe Folgekapitel).

Funktionsmerkmale bestehen (bis auf wenige Ausnahmen) immer aus folgenden Unterkapiteln:

1. Funktionsmerkmalweite Aspekte
2. Durch Ereignisse ausgelöste Reaktionen
3. Interne TUCs, **nicht** durch Fachmodule nutzbar
4. Interne TUCs, **auch** durch Fachmodule nutzbar
5. Operationen an der Außenschnittstelle
6. Betriebsaspekte

Die Unterkapitel 1-5 dienen der funktionalen Beschreibung des Funktionsmerkmals.

Punkte, die zum Funktionieren des Funktionsmerkmals relevant sind: Initialisierungsaspekte, durch den Administrator festzulegenden Konfigurationsparameter etc., werden im Unterkapitel Betriebsaspekte erfasst.

In jedem Funktionsmerkmal sind immer alle Unterkapitel enthalten, auch wenn es im konkreten Einzelfall dort keine Inhalte gibt. Diese feste Struktur innerhalb der Funktionsmerkmale erleichtert die Orientierung und erhöht somit die Lesbarkeit.

1.5.4.2 Technische Use Cases - TUCs

Innerhalb der Funktionsmerkmale in Kapitel 4 erfolgt eine Unterscheidung der TUCs in solche, die nur durch die Basisdienste des Konnektors aufgerufen werden dürfen (rein interne TUCs) und solche die neben den Basisdiensten auch durch Fachmodule genutzt werden dürfen. Diese Unterteilung ergibt sich ausschließlich aus dem Bedarf der editoriellen Steuerung der verschiedenen Spezifikationen (Konnektor- und Fachmodulspezifikationen). Es besteht im Rahmen der Implementierung des Konnektors keine Anforderung diese Trennung intern durchzusetzen.

Die Beschreibung der TUCs erfolgt nach folgendem Muster:

- TUC-Tabelle
- Aktivitäts- oder Sequenzdiagramm (optional)
- Fehlercodetabelle

Dabei wird innerhalb der TUC-Tabelle in der Zeile „Standardablauf" ausschließlich der Gut-Durchlauf beschrieben. Fehler, die innerhalb dieses Ablaufs auftreten können, werden in der Zeile „Fehlerfälle" erhoben. Dabei wird auf die jeweilige Schrittnummer innerhalb des Ablaufs referenziert. In dieser Tabellenzeile werden nur Fehlercodes erhoben, die im jeweiligen Fehlerfall geworfen werden müssen. Die genauen Festlegungen zu den Fehlern, neben Fehlercode auch: ErrorType, Severity und Fehlertext, werden in der Fehlercodetabelle festgelegt.

Die Spezifikation, in der ein TUC definiert wird, ist an den mittleren drei Buchstaben der TUC-Referenz zu erkennen:

- TUC_KON_xxx entsprechend in dieser Konnektorspezifikation
- TUC_PKI_xxx in der PKI-Spezifikation [gemSpec_PKI]
- TUC_VPN_ZD-xxxx in der Spezifikation des VPN-Zugangsdienstes [gemSpec_VPN_ZugD]
- TUC_VZD_xxx in der Spezifikation des Verzeichnisdienstes [gemSpec_VZD]

Festlegungen zur Schreibweise von Eingangs- und Ausgangsdaten von TUCs

a) Eingangs- und Ausgangsparameter werden in TUC-Tabellen wie folgt beschrieben:

Name des Eingangs- bzw. Ausgangsparameters

gefolgt von (falls definiert): `[Datentyp]`

gefolgt von (falls zutreffend):

- *optional; default: <Defaultwert>* bzw.
- *optional;*/<erklärender Text>

Hierbei bedeuten:

- optional; kennzeichnet optionale Ein- und Ausgangsparameter

default: <Defaultwert> definiert den Defaultwert für den Fall, dass der Eingangsparameter leer ist bzw. nicht übergeben wurde

/ <erklärender Text> beschreibt Bedingungen, unter denen der Eingangsparameter optional ist

gefolgt von (falls vorhanden): (<erklärender Text>)

b) Namen mit kleinem Anfangsbuchstaben bezeichnen Ein- und Ausgangsparameter; Namen mit großem Anfangsbuchstaben bezeichnen Datentypen.

Beispiel:

Eingangsdaten	• mandantId • allWorkplaces [Boolean] – *optional; default: false* (Dieser Schalter gibt an, ob eine mandantenweite Zugriffsberechtigung zum Tragen kommt…) • userId - *optional/verpflichtend, wenn cardType = HBAx*
Ausgangsdaten	• pinStatus [PinStatus] • leftTries – optional/verpflichtend, wenn pinStatus = VERIFYABLE (Anzahl der verbleibenden Versuche für die Verifikation der PIN)

Die im Dokument verwendeten Datentypen sind definiert in [Anhang L – Datentypen von Eingangs- und Ausgangsdaten].

Festlegungen zur Schreibweise des Aufrufs von TUCs

Ein TUC-Aufruf erfolgt nach folgendem Muster:

<TUC-Bezeichner> {

<TUC-Eingangsparameter Name> = <TUC Eingangsparameter Wert>;

… }

Beispiel:

```
TUC_KON_256 {
   topic = „CT/DISCONNECTED“;
   eventType = Op;
   severity = Info;
   parameters = („CtID=$CT.CTID, Hostname=$CT.HOSTNAME“) }
```

Vereinfachung:

Ist <TUC-Eingangsparameter Name> des aufzurufenden TUCs gleich der Variablen, die als < TUC Eingangsparameter Wert> gesetzt wird, so kann dieser Bezeichner ohne Zuweisung geschrieben werden.

Beispiel: (cardSession und pinRef sind Eingangsdaten des aufrufenden TUCs):

TUC_KON_022 „Liefere PIN-Status" {cardSession=cardSession; pinRef=pinRef}

vereinfachte Schreibweise:

TUC_KON_022 „Liefere PIN-Status" {cardSession; pinRef}

1.5.4.3 Event-Mechanismus

Der in Kapitel 4.1.6 spezifizierte Event-Mechanismus zur Unterrichtung von Clientsystemen wird innerhalb dieser Spezifikation auch zur internen Verzahnung der einzelnen Funktionsmerkmale eingesetzt. So wird ein Ereignis, das in der Managementschnittstelle durch Änderung eines Konfigurationsparameters ausgelöst wird, innerhalb des DHCP-Kapitels als Trigger für eine Lease-Erneuerung verwendet. Dies bedeutet nicht, dass im Rahmen der Implementierung intern ein Event-Mechanismus zwischen den Modulen verwendet werden muss. Auch hier dient die Form der Darstellung (Events) lediglich der editoriellen Kopplung verschiedener Verhaltensbeschreibungen.

Um den Ursprung eines Events erkennen zu können, verwenden alle Events ein Haupt-Topic passend zum Funktionsmerkmal: „DHCP/LAN_CLIENT/RENEW" wird im Funktionsmerkmal DHCP ausgelöst, „CARD/INSERTED" wird im Funktionsmerkmal Kartendienst ausgelöst usw.

1.5.4.4 Konfigurationsparameter und Zustandswerte

Werte die der Administrator des Konnektors einsehen oder verändern können muss, werden zusätzlich zu den Festlegungen in Kapitel 4.3 Konnektormanagement auch pro Funktionsmerkmal in den jeweiligen Unterkapiteln „Betriebsaspekte" erhoben. Diese **Konfigurationsparameter** werden über eine ReferenzID definiert. Definierte Konfigurationsparameter können in allen Kapiteln der Spezifikation referenziert werden. Den Ort, an welchem ein solcher Konfigurationsparameter definiert/erhoben und somit dessen Bedeutung beschrieben wird, lässt sich über den Präfix der ReferenzID erkennen: `CERT_CRL_DOWNLOAD_ADDRESS` (also „Cert") wird im Zertifikatsdienst definiert, `MGM_LU_ONLINE` (also „MGM") wird im Konnektormanagement definiert usw.

Die ReferenzIDs der Konfigurationsparameter besitzen in ihrer Schreibweise nur innerhalb des Dokuments Gültigkeit. In der Umsetzung können für die Konfigurationswerte herstellerspezifische Beschreibungen und Labels verwendet werden.

Vergleichbar zu diesen Konfigurationsparametern, sind **Zustandswerte.** Auch diese werden über ReferenzIDs definiert, nur können sie nicht durch den Administrator verändert oder eingesehen werden. Sie finden nur konnektorintern Verwendung und sind für die Beschreibung der Verhaltensweise notwendig, Beispiele sind `CTM_CT_LIST` für die Liste der durch den Konnektor verwalteten Kartenterminals oder `CM_CARD_LIST` für die Liste der aktuell erreichbaren Karten. Zustandswerte verwenden die gleichen Präfixe wie Konfigurationsparameter.

2 Systemüberblick

Der Konnektor ist ein Produkttyp der TI gemäß [gemKPT_Arch_TIP#5.3.9].

Er bietet seine Basisdienste sowohl intern den in ihm laufenden Fachmodulen an, als auch externen Clientsystemen über die Konnektoraußenschnittstellen.

Im lokalen Netz der Einsatzumgebung kommuniziert das Clientsystem mit dem Konnektor über dessen LAN-seitiges Ethernet-Interface. Alleinig der Konnektor kommuniziert mit den in lokalen Netzen angeschlossenen Kartenterminals und Karten. Auch die Kommunikation mit den zentralen Diensten der TI-Plattform und fachanwendungsspezifischen Diensten erfolgt ausschließlich über den Konnektor über dessen WAN-seitiges Ethernet-Interface.

Um die lokale Anzeige für die Signaturerstellung und Signaturprüfung zu realisieren, wird ein Signaturproxy verwendet, der die Schnittstellen I_Sign_Operations und I_SAK_Operations sowie ServiceDirectory kapselt. Der Signaturproxy ist aus Gründen der Übersichtlichkeit nicht in der Abbildung PIC_KON_116 dargestellt, seine Spezifikation findet sich in [gemSpec_Kon_SigProxy].

Abbildung PIC_KON_116 stellt die Schnittstellen im Umfeld des Konnektors dar.

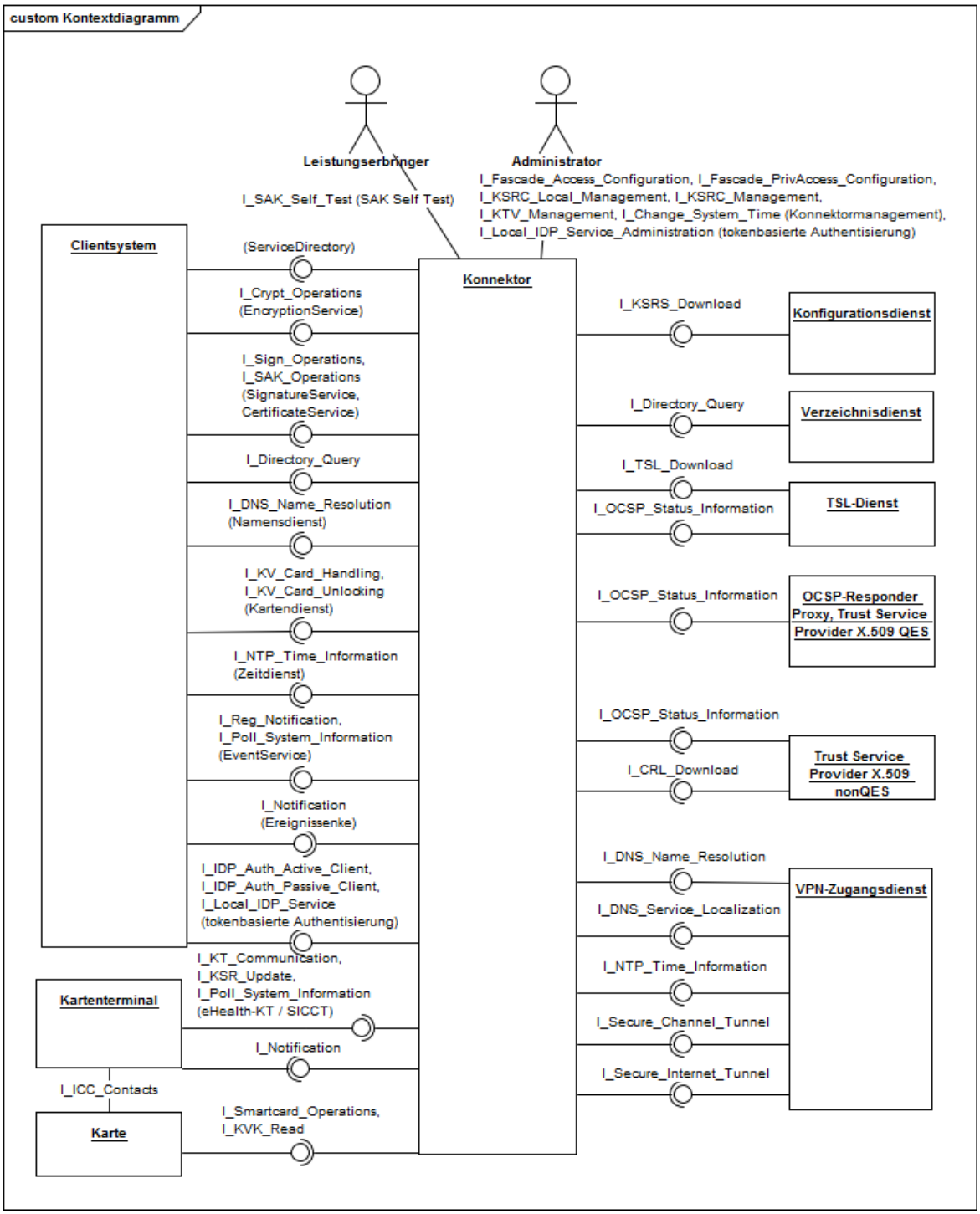


Abbildung 1: PIC_KON_116 Schnittstellen des Konnektors von und zu anderen Produkttypen

Die logischen Außenschnittstellen aus [gemKPT_Arch_TIP] werden im Konnektor technisch vorrangig als SOAP-Schnittstellen ausgeprägt. Von dieser Regel wird insbesondere bei Netzwerkschnittstellen abgewichen, wenn bereits etablierte Schnittstellenstandards für Basisdienste existieren (IPsec, TLS, NTP, DNS etc.). Eine

Übersicht der Zuordnung „logische Schnittstellen → technische Schnittstellen" findet sich in Anhang H.

Zum Nachweis der Sicherheit müssen Konnektoren im Rahmen der Zulassung nach Common Criteria gegen die Schutzprofile [PP_NK] und [PP_KON] evaluiert und zertifiziert werden.

Die zu verwendenden kryptographischen Verfahren und zugehörige Parameter (z. B. Schlüssellängen) für alle kryptographischen Operationen innerhalb der Telematikinfrastruktur, werden durch das Dokument „Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur" [gemSpec_Krypt] normativ geregelt.

2.1 Logische Struktur

Der Produkttyp Konnektor besitzt eine Vielzahl verschiedenster Operationen und Verhaltensweisen an seiner Außenschnittstelle. Um sein komplexes Gesamtverhalten sinnvoll beschreiben zu können, wird der Konnektor innerhalb dieser Spezifikation logisch unterteilt und strukturiert. Es wird primär zwischen Anwendungs- und Netzkonnektor unterschieden, begleitet von Mechanismen, die blockübergreifend beschrieben werden.

Der logische Aufbau des Konnektors ist in Abbildung PIC_KON_117 dargestellt.

- Der Anwendungskonnektor bietet anwendungsnahe Basisdienste (inklusive Signaturdienst) und Fachmodule zur Nutzung durch ein Clientsystem an.
- Der Netzkonnektor bietet transportnahe Basisdienste und verbindet das lokale Netz der Nutzer mit der zentralen TI-Plattform.
- Die gSMC-K ist zwar ein eigenständiger Produkttyp innerhalb der TI, wird im Konnektor jedoch als Verbaukomponente betrachtet. Sie enthält die kryptographischen Identitäten des Konnektors, sowie Steuerdaten (Umgebungsinformationen TU/RU/PU, zugehörige Adressbereiche, herstellerspezifische Konfigurationsdaten), die aus Sicherheitsgründen unveränderlich in den Konnektor eingebracht werden müssen.
- Das Konnektormanagement dient der administrativen Verwaltung und Steuerung des gesamten Konnektors.
- Der Sichere Datenspeicher dient der integeren, vertraulichen und authentischen Persistierung von veränderlichen Daten (siehe auch Kapitel 2.2).
- Der Signaturproxy ist eine Komponente, die zwischengeschaltet auf der Kommunikationsstrecke zwischen Client-System und Konnektor dafür sorgt, dass die zu signierenden oder zu prüfenden Dokumente dem Nutzer angezeigt werden. Die Beschreibung des Signaturproxy befindet sich in [gemSpec_Kon_SigProxy]

Abbildung 2: PIC_KON_117 Logische Zerlegung des Konnektors in Anwendungs- und Netzkonnektor

Diese logische Unterteilung schreibt in keiner Art und Weise die spätere Implementierung durch den Hersteller vor. Der Hersteller kann seine interne Modularisierung des Konnektors frei wählen. Normativ wirksam ist ausschließlich das durch die Detailfestlegungen in Summe beschriebene Verhalten an den Außenschnittstellen des Konnektors als Ganzes.

2.2 Sicherer Datenspeicher

Wie im vorherigen Kapitel dargestellt, wird für den Konnektor ein Datenspeicher angenommen, in welchem der Konnektor alle sicherheitskritischen, veränderlichen Daten dauerhaft speichert, die für seinen Betrieb relevant sind. Dieser Datenspeicher sichert die Integrität, Authentizität und Vertraulichkeit der in ihm hinterlegten Daten bzw. der aus ihm entnommenen Daten. Alleinig der Konnektor hat auf diesen Datenspeicher Zugriff. Für folgende, im weiteren Verlauf der Spezifikation anfallende Daten wird angenommen, dass diese im Sicheren Datenspeicher persistiert werden:

- Der Trust Store des Zertifikatsdienstes
- Die Konfigurationsdaten des Konnektormanagements
- Die Konfigurationsdaten aller Funktionsmerkmale

Ferner stellt der Konnektor den in ihm laufenden Fachmodulen ebenfalls eine Nutzung dieses Datenspeichers für ihre sensiblen Daten zur Verfügung.

Da es sich bei dem Sicheren Datenspeicher um ein internes Modul handelt, welches an der Außenschnittstelle nicht testbar ist, werden an dieses Modul im Rahmen dieser Spezifikation keine Anforderungen erhoben. Da dieses logische Modul aber essenzielle Sicherheitsfunktionen bietet, ohne die ein Konnektor nicht sicher betrieben werden kann, werden die Funktionen, die ein Hersteller für sein Konnektormodell real umsetzt, um die notwendigen sicheren Speicherfunktionen zu realisieren, im Rahmen der CC-Evaluierung geprüft werden. Näheres hierzu regeln die Schutzprofile des Konnektors.

2.3 Überblick Konnektoridentität

Die Geräteidentität des Konnektors (Konnektoridentität) teilt sich in drei Identitäten auf:

- ID.NK.VPN für den Netzkonnektor
  Die Identität des Netzkonnektors dient der Authentisierung gegenüber den zentralen Netzwerkdiensten und wird für die Anmeldung an den VPN-Konzentrator genutzt.
- ID.AK.AUT für den Anwendungskonnektor
  Die Identität des Anwendungskonnektors dient der Authentisierung gegenüber den Clientsystemen im Rahmen von TLS-Verbindungen.
- ID.SAK.AUT für die im Anwendungskonnektor enthaltene Signaturanwendungskomponente
  Die Identität des Signaturdienstes dient zur Authentisierung gegenüber den Kartenterminals. Darüber hinaus muss sich der Signaturdienst des Konnektors gegenüber dem Heilberufsausweis mittels eines kartenverifizierbaren Zertifikats (C.SAK.AUTD_CVC) mit entsprechendem Profil ausweisen, um Stapelsignaturen durchführen zu können.

In der Regel ergibt sich aus dem Kontext, welche Identität gemeint ist, sodass in diesen Fällen nur kurz von der Konnektoridentität geschrieben wird.

Die Geräteidentitäten werden durch asymmetrische Schlüssel und X.509-Zertifikate umgesetzt. In Abhängigkeit vom gewählten kryptographischen Verfahren werden RSA-Schlüssel bzw. ECC-Schlüssel verwendet.

2.4 Mandantenfähigkeit

Den Anforderungen aus [gemKPT_Arch_TIP#TIP1-A_2200] folgend, wird die Mandantenfähigkeit innerhalb des Konnektors nicht durch eine einzelne Funktion, sondern durch Berücksichtigung in einer Reihe von Funktionsmerkmalen umgesetzt.

Die Mandantenfähigkeit wirkt dabei auf:

- Zugriffsberechtigungsdienst: Kapitel 4.1.1
  (und über diesen auf alle Karten- und Kartenterminaloperationen)
- Systeminformationsdienst: Kapitel 4.1.6

2.5 Versionierung

Gemäß [gemSpec_OM] müssen Konnektor und Kartenterminals über eine Versionierung verfügen. Die relevanten Versionsinformationen sind durch das O&M-Schema ProductInformation.xsd definiert. Ferner definiert [gemSpec_OM], dass Konnektor und Kartenterminal das Konzept der Firmware-Gruppe verwenden müssen. Daher verfügen die beiden Produkttypen auch über eine aktuelle Firmware-Gruppenversion.

Versionsinformationen werden innerhalb des Konnektor an folgenden Stellen ver- und bearbeitet:

- Dienstverzeichnisdienst (Kapitel 4.1.3): Ausgabe der Konnektorversion über SOAP
- Kartenterminaldienst (Kapitel 4.1.4): Anzeige der Versionsinformationen der verwalteten Kartenterminals
- Konnektormanagement (Kapitel 4.3):
  - Anzeige der Versionsinformationen des Konnektors (Kapitel 4.3.2)
  - Software-Aktualisierung (KSR-Client) für Konnektor und Kartenterminals (Kapitel 4.3.9)

2.6 Fachanwendungen

Der Konnektor ist als Plattformkomponente der TI für die Erbringung von Basisdiensten verantwortlich. Fachliche Funktionalitäten werden über die Fachmodule bereitgestellt.

Das Fachmodul wird dabei als integraler Bestandteil des Konnektors verstanden (Konnektor als Monolith), d. h., die Spezifikationen zu Konnektor (als Plattformkomponente) und dem Fachmodul sind zwar getrennt, werden aber von einem Hersteller in einer Gesamtkomponente umgesetzt. Die inneren Schnittstellen zwischen Fachmodul und Konnektor sind von außen nicht erkennbar.

In dieser Ausbaustufe unterstützt der Konnektor die Fachanwendungen VSDM, AMTS, NFDM und ePA über jeweils ein Fachmodul.

Neben Fachanwendungen, die über ihr Fachmodul mit einem gesicherten Fachdienst kommunizieren, unterstützt der Konnektor einen Zugriff von Clientsystemen auf offene Fachdienste.

2.7 Netzseitige Einsatzszenarien

Der Konnektor unterstützt unterschiedliche netzseitige Einsatzszenarien, die in Anhang K beispielhaft dargestellt sind.

Der Konnektor bietet hierzu Konfigurations-Parameter, die je nach netzseitigem Einsatzszenario konfiguriert werden müssen.

2.7.1 Parameter ANLW_ANBINDUNGS_MODUS

Konfiguration 1: Konnektor als Gateway (ANLW_ANBINDUNGS_MODUS = InReihe):
Diese Konfiguration ist geeignet für Szenarien, in denen der Konnektor zwischen das lokale Netz und das Internet Access Gateway (IAG) (z. B. Router mit DSL-/Kabelmodem) geschaltet wird. (vgl. Anhang K, Szenario 1)

Konfiguration 2: Konnektor eingebettet in existierende Infrastruktur (ANLW_ANBINDUNGS_MODUS = Parallel): Diese Konfiguration ist geeignet für Szenarien, in denen der Konnektor als weiteres Gerät in die bestehende Netzwerkinfrastruktur integriert wird. (vgl. Anhang K, Szenario 3)

Aus Sicherheitsgründen soll die Kommunikation der Clientsysteme mit dem Konnektor hierbei verschlüsselt erfolgen (ANCL_TLS_MANDATORY=Enabled). Falls diese Kommunikation unverschlüsselt erfolgt (ANCL_TLS_MANDATORY=Disabled), übernimmt der Nutzer die Verantwortung für die Sicherstellung der vertraulichen Übertragung.

Für den Einsatz und die Nutzung von DHCP gibt es im Zusammenhang mit diesem Konfigurationsparameter folgende Möglichkeiten:

- Die Netzwerkinfrastruktur der Einsatzumgebung verwendet den DHCP-Server des Konnektors (siehe Kap. 4.2.2).
- Ein bestehender DHCP-Server im Netz der Einsatzumgebung wird weiter verwendet und derart konfiguriert, dass als Default Gateway und DNS-Server entweder bestehende Infrastruktur oder der Konnektor verwendet wird.
- Es kommt kein DHCP-Server zum Einsatz. Bei allen Clients im Netz der Einsatzumgebung werden das Default Gateway und der DNS-Server statisch auf den Konnektor gesetzt.

Die DHCP-Konfiguration ist in Konfiguration 1 in aller Regel die folgende: Die WAN-Seite des Konnektors verwendet den DHCP-Server des bestehenden IAG. An der LAN-Seite stellt der Konnektor einen DHCP-Server für alle Clients zur Verfügung.

2.7.2 Parameter ANLW_INTERNET_MODUS

Grundsätzlich routet der Konnektor im Modus ANLW_INTERNET_MODUS=SIS alle für das Internet bestimmten Pakete von Clients, die ihn als Default Gateway verwenden, in den VPN-Tunnel zum SIS, während er im Modus ANLW_INTERNET_MODUS=Keiner diese Pakete verwirft.

Im Unterschied zu (ANLW_ANBINDUNGS_MODUS = InReihe) ist die Nutzung des SIS bei (ANLW_ANBINDUNGS_MODUS = Parallel) optional. Alternativ können auch die Clients, die den Konnektor als Default Gateway verwenden, per Redirect direkt ins Internet verwiesen werden (ANLW_INTERNET_MODUS=IAG).

2.8 Lokale und entfernte Kartenterminals

Gemäß [gemKPT_Arch_TIP] ermöglicht die Telematikinfrastruktur dem Anwender die PIN-Eingabe zur Freischaltung eines HBAs oder einer SMC-B wahlweise lokal oder über das Remote-PIN-Eingabeverfahren durchzuführen. Deshalb unterscheidet auch der Konnektor zwischen einem lokalen Kartenterminal – räumlich („in Sichtweite") dem Arbeitsplatz zugeordnet – und einem entfernten Kartenterminal.

Ein lokales Kartenterminal befindet sich lokal an einem Arbeitsplatz und kann von diesem aus genutzt werden. Hingegen ist das entfernte Kartenterminal einem entfernten oder auch – für zentral steckende Karten – keinem Arbeitsplatz fest zugewiesen. Ein lokales Kartenterminal kann als sogenanntes Remote-PIN-KT verwendet werden, um die PIN für eine in einem entfernten Kartenterminal steckende Karte einzugeben.

2.9 Standalone-Szenario

Gemäß § 291 SGB V Absatz 2b müssen „Diese Dienste [zur Online-Aktualisierung der Versichertendaten auf der eGK] […] auch ohne Netzanbindung an die Praxisverwaltungssysteme der Leistungserbringer online genutzt werden können."

Dies bedeutet, dass der Konnektor ohne ein steuerndes Clientsystem ereignisgetrieben Fachanwendungen ausführen können muss. Aus Fachsicht „steht der Konnektor alleine", ohne Clientsysteme. Die konkreten Aktionen, die Fachanwendungen in diesen Fällen ausführen, sowie deren Auslöser werden in den jeweiligen Fachmodulspezifikationen beschrieben.

Ein solcher alleinstehender Konnektor mit Zugang zur TI muss zur Durchführung der Fachanwendungen durch einen weiteren Konnektor unterstützt werden, der in direkter Verbindung zum Clientsystem steht, selbst aber keine Online-Anbindung besitzt.

3 Übergreifende Festlegungen

Für die folgenden Inhalte bitte die Hinweise in Kapitel 1.5.3 „Erläuterungen zur Spezifikation des Außenverhaltens„ sowie Kapitel 1.5.4 Erläuterungen zur Dokumentenstruktur und „Dokumentenmechanismen" beachten.

In diesem Kapitel werden die Aspekte des Konnektors behandelt, die Funktionsmerkmalübergreifend geregelt werden müssen.

Die Managementschnittelle/Administrationsoberfläche des Konnektors wird dabei nicht als übergreifender Aspekt, sondern als eigenes Funktionsmerkmal gewertet. Die Festlegungen hierzu finden sich entsprechend in Kapitel 4.3.

Dokumentformate

Mit dem Aufruf einer Operation, die Dokumente verarbeitet, muss durch den Aufrufer festgelegt werden können, um welches Dokumentenformat es sich handelt, damit die unterschiedlichen Formate zur Verarbeitung und etwaigen Anzeige unterschieden werden können. Die nicht-XML-Formate werden dabei nach MIME-Typ-Klassen unterschieden:

- „PDF/A" für MIME-Typ „application/pdf-a" gemäß [ISO 19005],
- „Text" für MIME-Typ „text/plain",
- „TIFF" für MIME-Typ „image/tiff" gemäß [TIFF6]
- „Binär" für alle übrigen MIME-Typen.

Folgende Bezeichner werden verwendet:

`Alle_DocFormate`: XML, PDF/A, Text, TIFF, Binär

`nonQES_DocFormate`: XML, PDF/A, Text, TIFF, Binär

`QES_DocFormate`: XML, PDF/A, Text, TIFF

Für nonQES_DocFormate wird, trotz Gleichheit zu Alle_DocFormate, ein eigener Referenzbezeichner verwendet, da sich diese Liste noch ändern könnte. TIFF wird durch [gemKPT_Arch_TIP] nicht für die nonQES verlangt. Die Unterstützung dieses Formats für nonQES bedeutet jedoch keinen Mehraufwand, da die Routinen durch QES bereits implementiert sind und nachgenutzt werden können.

TIP1-A_4500 - Dokumentgrößen von 25 MB
Der Konnektor MUSS für alle Außenschnittstellen, in denen ein Dokument verarbeitet wird, Dokumente mit einer Größe <= 25 MB unterstützen. Der Konnektor KANN Dokumente mit einer Größe > 25 MB unterstützen.**[<=]**

A_19052 - Vorgaben für Dokumentformate und Nachrichten
Der Konnektor MUSS für die Verarbeitung von Dokumenten und Nachrichten die Vorgaben aus TAB_KON_775 erfüllen.**[<=]**

TIP1-A_4502 - Zeichensatzcodierungen UTF-8 und ISO-8859-15
Der Konnektor MUSS bei der Verarbeitung von Dokumenten der Formate XML und Text die Zeichensatzkodierungen UTF-8 und ISO-8859-15 unterstützen. Das verarbeitete Dokument MUSS der Konnektor mit demselben Zeichensatz kodieren, in dem das Eingangsdokument kodiert war.**[<=]**

TIP1-A_5541-01 - Referenzen in Dokumenten nicht dynamisch auflösen
Der Konnektor DARF in Dokumenten eventuell vorhandene Referenzen auf externe Ressourcen NICHT auflösen, es sei denn es sind Verweise auf im Konnektor sicher eingebrachte vorliegende Schemata oder dies wird im Einzelfall normativ gefordert.**[**<=**]**

Kartentypen

Der Konnektor unterstützt eine Reihe von Kartentypen. Die folgende Tabelle enthält die Liste der Referenzbezeichner für die verschiedenen Kartentypen, wie sie im weiteren Verlauf verwendet werden. Die Unterstützung von Karten der Generation 2 (G2.x: G2.0, G2.1 und höher) beschränkt sich bei diesen auf die Datenstrukturen und Schlüssel, die aus Gründen der Abwärtskompatibilität zu den Karten der Generation 1+ vorhanden sind. Eine Ausnahme hiervon bilden die Geräte-CVCs, die bereits für dieses Release basierend auf ECC verwendet werden.

Tabelle 1: TAB_KON_500 Wertetabelle Kartentypen

ReferenzID Kartentyp	Karten-generation	Beschreibung
EGK	G1+	Die elektronische Gesundheitskarte gemäß [gemSpec_eGK_P1] und [gemSpec_eGK_P2]
EGK	G2	Die elektronische Gesundheitskarte gemäß [gemSpec_COS] und [gemSpec_eGK_ObjSys] bzw. [gemSpec_eGK_ObjSys_G2.1]
HBA-qSig	-	HBA-Vorläuferkarte gemäß [HPC-P1] und [HPC-P2]
HBA	G2	Der elektronische Heilberufsausweis (HBA) gemäß [gemSpec_COS] und [gemSpec_HBA_ObjSys]
SMC-B	G2	Die Institutionskarte Typ B (Secure Module Card) gemäß [gemSpec_COS] und [gemSpec_SMC-B_ObjSys]
HSM-B		HSM-Variante einer SM-B. Das HSM-B wird in dieser Fassung als ein oder mehrere virtuelle Kartenterminals verstanden, in denen virtuelle Karten stecken.
SMC-KT	G2	Die Karte Typ KT (Secure Module Card) gemäß [gemSpec_COS] und [gemSpec_gSMC-KT_ObjSys]
KVK	-	Die Krankenversichertenkarte gemäß der Spezifikation [KVK]
ZOD_2.0	-	HBA-Vorläuferkarte gemäß [HPC-P1] und [HPC-P2]
UNKNOWN		Eine nicht erkannte Karte oder nicht lesbare Karte
		Zusammenfassende ReferenzIDs
HBA-VK		Adressiert die HBA-Vorläuferkarten HBA-qSig und ZOD_2.0. Wird dieser Referenzbezeichner verwendet, gelten die

		zugehörigen Aussagen und Festlegungen für beide Kartentypen.
HBAx		Adressiert sowohl den HBA, als auch die HBA-Vorläuferkarten (HBA-VK) Wird dieser Referenzbezeichner verwendet, gelten die zugehörigen Aussagen und Festlegungen für alle drei Kartentypen.
SM-B		Adressiert sowohl eine echte SMC-B als auch eine in einem HSM-B enthaltene virtuelle SMC-B. Wird dieser Referenzbezeichner verwendet, gelten die zugehörigen Aussagen und Festlegungen für beide Typen.

Übergreifende Festlegungen zum Aufbau von sicheren Verbindungen

TIP1-A_7254 - Reaktion auf OCSP-Abfrage beim TLS-Verbindungsaufbau
Der Konnektor MUSS beim Aufbau von TLS-gesicherten Verbindungen zu einem zentralen Dienst der TI-Plattform oder zu einem fachanwendungsspezifischen Dienst, bei denen eine OCSP-Abfrage des Serverzertifikats nach TUC_PKI_006 erfolgt, neben Fehlerfällen bei folgenden Warnungen gemäß [gemSpec_PKI#Tab_PKI_274]

- CERT_REVOKED
- CERT_UNKNOWN
- OCSP_CHECK_REVOCATION_FAILED

mit Abbruch des Verbindungsaufbaus reagieren.**[<=]**

In [gemSpec_Krypt#6] wird das Kommunikationsprotokoll zwischen einem Client und einer Vertrauenswürdigen Ausführungsumgebung (VAU) spezifiziert. Dabei wird ein sicherer Kanal auf HTTP-Anwendungsschicht zwischen dem Client und der VAU (Server) aufgebaut. Der Client ist hier ein Fachmodul des Konnektors; der Server ist ein Fachdienst.

A_17225-01 - Aufbau einer sicheren Verbindung zur Vertrauenswürdige Ausführungsumgebung (VAU)
Der Konnektor MUSS für Fachmodule den Aufbau einer sicheren Verbindung zur Vertrauenswürdigen Ausführungsumgebung (VAU) gemäß Kommunikationsprotokoll [gemSpec_Krypt#6] unterstützen und das vom Server übergebene Zertifikat wie folgt prüfen:
TUC_KON_037 „Zertifikat prüfen“ {
certificate = C.FD.AUT;
qualifiedCheck = not_required;
offlineAllowNoCheck = false;
policyList = oid_fd_aut;
intendedKeyUsage= intendedKeyUsage(C.FD.AUT);
validationMode = OCSP}
Der Konnektor MUSS die vom Fachmodul übergebene Rolle gegen die aus dem Zertifikat ermittelte Rolle prüfen. **[<=]**

A_17777 - sicherheitstechnische Festlegungen zum Abruf von kryptographischen Schlüsseln von einem Schlüsselgenerierungsdienst
Der Konnektor MUSS für Fachmodule für die Nutzung der Schlüsselableitungsfunktionalität die sicherheitstechnischen Festlegungen gemäß [gemSpec_Krypt#3.15.5 Schlüsselableitungsfunktionalität ePA] und [gemSpec_SGD] bereitstellen.**[<=]**

Der Gesamtablauf der Schlüsselableitungsfunktionalität gemäß [gemSpec_SGD#2.3] für den Konnektor als Client ist aufgeteilt zwischen Basiskonnektor und Fachmodul. Die kryptographischen Vorgaben (u.a. Durchführung des ECDH, Schlüsselerzeugung, Ver- und Entschlüsselung, Signaturerzeugung und -prüfung) werden dabei durch den Basiskonnektor realisiert.

3.1 Konnektoridentität und gSMC-K

TIP1-A_4503 - Verpflichtung zur Nutzung von gSMC-K
Der Konnektor MUSS das geheime Schlüsselmaterial zur Geräteidentität (ID.NK.VPN, ID.AK.AUT, ID.SAK.AUT) und der Rolle SAK (C.SAK.AUTD_CVC) über Smartcards des Typs gSMC-K gemäß [gemSpec_gSMC-K_ObjSys] nutzen. Der Konnektor MUSS mit einer gSMC-K bestückt sein. Er KANN mit mehr als einer gSMC-K bestückt sein.
[<=]

Die Notwendigkeit, den Konnektor mit mehr als einer gSMC-K zu bestücken, kann sich aus den Lastanforderungen aus [gemSpec_Perf#4.1.2] ergeben.

TIP1-A_4504 - Keine Administratorinteraktion bei Einsatz mehrerer gSMC-Ks
Verwendet der Konnektor mehrere gSMC-Ks, DARF eine Administratorinteraktion für diese Belange NICHT erforderlich sein.
[<=]

TIP1-A_5543 - Keine manuelle PIN-Eingabe für gSMC-K
Der Konnektor DARF Anwender und Administratoren außer bei der Inbetriebnahme (erstmalig oder nach Werksreset) NICHT auffordern, eine PIN für eine gSMC-K einzugeben.
[<=]

TIP1-A_4505 - Schutz vor physischer Manipulation gSMC-K (Sichere Verbundenheit der gSMC-K)
Die gSMC-K des Konnektors MÜSSEN durch den Einsatz physikalischer Sperren oder manipulationssicherer Siegel so mit dem Konnektor verbunden sein, dass physischer Missbrauch oder physische Manipulation erkennbar ist.
[<=]

gSMC-Ks gemäß [gemSpec_gSMC-K_ObjSys] verfügen über die Möglichkeit zur nachträglichen Generierung von Schlüsselpaaren und dem Nachladen der zugehörigen Zertifikate. Dieser Mechanismus wird erst in kommenden Releases durch den Konnektor unterstützt. Initial sind alle Identitäten bereits einmal auf der gSMC-K vorhanden.

TIP1-A_4506 - Initiale Identitäten der gSMC-K
In Abhängigkeit vom kryptographischen Verfahren MUSS der Konnektor folgende Objekte der gSMC-K als Quelle seiner Identitäten verwenden:

Tabelle 2: TAB_KON_856: Identitäten des Konnektors auf der gSMC-K

Identifier	Verzeichnis	Objekt der gSMC-K in Abhängigkeit vom kryptographischen Verfahren	
		RSA	ECC
ID.NK.VPN	MF/DF.NK	EF.C.NK.VPN.R2048	EF.C.NK.VPN2.XXXX
ID.AK.AUT	MF/DF.AK	EF.C.AK.AUT.R2048	EF.C.AK.AUT2.XXXX

ID.SAK.AUT	MF/DF.SAK	EF.C.SAK.AUT.R2048	EF.C.SAK.AUT2.XXXX
C.SAK.AUTD_CVC	MF/DF.SAK	-	EF.C.SAK.AUTD_CVC.E256

[<=]

3.1.1 Erneuerung der Zertifikate der gSMC-K

A_21736 - Verwendung erneuerter Zertifikate
Nach erfolgreicher Zertifikatserneuerung MUSS der Konnektor vor Ablauf der alten Zertifikate an allen Stellen, wo er die Zertifikate C.NK.VPN, C.AK.AUT, C.SAK.AUT, C.SAK.AUTD_CVC und C.CA_SAK.CS verarbeitet, die neuen Zertifikate verwenden, es sei denn die Spezifikation trifft andere Festlegungen.**[<=]**

A_21744 - Zertifikate regelmäßig erneuern
Der Konnektor MUSS die Zertifikate C.NK.VPN, C.AK.AUT, C.SAK.AUT, C.SAK.AUTD_CVC und C.CA_SAK.CS regelmäßig erneuern. Der Konnektor MUSS 180 Tage vor Ablauf des aktuell verwendeten C.NK.VPN-Zertifikats den Zertifikatserneuerungsprozess anstoßen. Solange die Zertifikate noch nicht vollständig erfolgreich erneuert wurden, MUSS der Konnektor genau einmal täglich durch Aufruf von TUC_KON_410 neue Zertifikate beziehen.**[<=]**

A_21849 - Anzeige verwendeter Zertifikate
Der Konnektor MUSS dem Administrator anzeigen, welche Zertifikate aktuell verwendet werden. Dies gilt für diese Strecken bzw. Anwendungsfälle: VPN-Tunnel (C.NK.VPN), TLS zum PS (C.AK.AUT oder lokales Zertifikat), TLS zum KT (C.SAK.AUT) und C2C für SUK (C.SAK.AUTD_CVC und C.CA_SAK.CS).**[<=]**

Es werden keine expliziten Festlegungen zu herstellerspezifisch verwendetem Material auf der gSMC-K getroffen. Es liegt in der Verantwortung des Konnektor Herstellers, dafür zu sorgen, dass der Konnektor nach Erneuerung und Aktivierung der spezifizierten Zertifikate insgesamt fehlerfrei lauffähig ist.

A_21879 - Erneuerte Zertifikate der gSMC-K manuell importieren
Der Konnektor MUSS es dem Administrator ermöglichen, erneuerte Zertifikate C.NK.VPN, C.AK.AUT, C.SAK.AUT, C.SAK.AUTD_CVC und C.CA_SAK.CS manuell von lokaler Datenquelle einzuspielen.
Der Konnektor MUSS dies auch im kritischen Betriebszustand `EC_NK_Certificate_Expired` ermöglichen.**[<=]**

A_21749-01 - TUC_KON_410 „gSMC-K-Zertifikate aktualisieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_410 „gSMC-K-Zertifikate aktualisieren" umsetzen.

Tabelle 3: TAB_KON_930 – TUC_KON_410 „Zertifikate aktualisieren"

Element	Beschreibung
Name	TUC_KON_410 "gSMC-K-Zertifikate aktualisieren"
Beschreibung	Dieser TUC bezieht neue gSMC-K-Zertifikate vom Downloadpunkt des TSP X.509 nonQES für

	Komponenten, oder diese werden vom Administrator übergeben.
Auslöser	A_21744, Administrator
Vorbedingungen	Automatische Aktualisierung: • MGM_LU_ONLINE=Enabled • Verbindung zum VPN-Konzentrator TI ist aufgebaut
Eingangsdaten	Manuelle Aktualisierung: • Zertifikate
Komponenten	Konnektor, TSP Komponenten
Ausgangsdaten	Keine

Standardablauf	Automatische Aktualisierung:
	1. Für jede verbaute gSMC-K wird die zip-Datei mit neuen Zertifikaten per HTTP vom Downloadpunkt TSP Komponenten bezogen ([gemSpec_X.509_TSP#A_21770]).
	2. Die zip-Dateien werden entpackt.
	3. Für jedes bezogene Zertifikat führt der Konnektor folgende Prüfungen durch:
	a. ICCSN des neuen und alten Zertifikats sind gleich
	b. Ablaufdatum des neuen Zertifikats liegt nach Ablaufdatum des alten Zertifikats
	c. Kryptografische Prüfung, dass öffentlicher Schlüssel zum privaten Schlüssel passt
	d. Neue Zertifikatsseriennummer ungleich alter Zertifikatsseriennummer
	e. Für C.NK.VPN-Zertifikat: OCSP-Abfrage gemäß GS-A_4657-03
	4. Erfolgreich geprüfte Zertifikate werden im sicheren Speicher abgelegt und zur Verwendung vorgemerkt.
	5. TUC_KON_256 {
	topic = „SMC_K/UPDATE/SUCCESS“;
	eventType = Op;
	severity = Info;
	parameters = „$Parameters“;
	doLog = true;
	doDisp = true }

Varianten/Alternativen	Manuelle Aktualisierung: 1. Die Files mit den neuen Zertifikaten werden vom Administrator in den Konnektor importiert. 2. Herstellerspezifisch, je nach Dateiformat

Fehlerfälle	(->1) Fehler beim Download:
	TUC_KON_256 {
	topic = „SMC_K/DOWNLOAD/ERROR“;
	eventType = Op;
	severity = Error;
	parameters = „$Parameters“;
	doLog = true;
	doDisp = true }
	(->3) Wenn eine der folgenden Prüfungen fehlschlägt, wird das bezogene Zertifikat verworfen und mit dem nächsten fortgesetzt:
	(-> 3a) ICCSN nicht gleich: Fail=Iccsn
	(-> 3b) Neues Ablaufdatum nicht später als altes Ablaufdatum: Fail=Date
	(-> 3c) Öffentlicher Schlüssel passt nicht zum privaten Schlüssel: Fail=Crypt
	(-> 3e) Zertifikat gesperrt oder unknown: Fail=Ocsp
	Automatische Aktualisierung: TUC_KON_256 {
	topic = „SMC_K/UPDATE/ERROR“;
	eventType = Op;
	severity = Error;
	parameters = „$Parameters“;
	doLog = true;
	doDisp = true }
	(->3) Wenn eine der folgenden Prüfungen fehlschlägt, wird das bezogene Zertifikat trotzdem zur Verwendung vorgemerkt:
	(-> 3d) Zertifikatsseriennummer identisch: Fail=Serial
	Warnung wird protokolliert

Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 4: Tab_Kon_931 Fehlercodes TUC_KON_410 „gSMC-K-Zertifikate aktualisieren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
herstellerspezifisch			

[<=]

A_21780 - Nutzerhinweis bezüglich Fehler bei Zertifikatserneuerung
Der Hersteller des Konnektors MUSS in seinem Handbuch den Nutzer/Administrator darauf hinweisen, dass die Ereignisse mit dem Topic=SMC_K/UPDATE/ERROR und dem Topic=SMC_K/DOWNLOAD/ERROR dringend durch das Primärsystem abonniert werden sollten und dass der Nutzer/Administrator sich bei Auftreten des Fehlers unverzüglich mit dem Hersteller in Verbindung setzen muss.
[<=]

3.1.2 Organisatorische Anforderungen und Sperrprozesse

TIP1-A_5392 - gSMC-K-Verantwortung durch den Hersteller des Konnektors
Der Hersteller des Konnektors MUSS die Rolle des Kartenherausgebers für in seinen Konnektoren verbauten gSMC-Ks einnehmen.
Der Hersteller des Konnektors KANN die von ihm verantwortete Personalisierung der gSMC-K durch einen von ihm zu beauftragenden Dienstleister in seinem Namen vornehmen lassen.
[<=]

TIP1-A_5696 - Prüfung der personalisierten gSMC-K
Der Hersteller des Konnektors MUSS sich von der korrekten Personalisierung der herausgegebenen gSMC-K überzeugen.
[<=]

A_18928 - Ausstattung mit dual-personalisierten gSMC-K-X.509-Zertifikaten
Der Hersteller des Konnektors MUSS die Konnektoren mit einer gSMC-K mit personalisierten RSA- und ECC-Zertifikaten gemäß TAB_KON_856 ausstatten.**[<=]**

A_18930 - Unterstützung von gSMC-K Personalisierungsvarianten
Der Konnektor MUSS unterschiedliche gSMC-K-Personalisierungsvarianten sowohl mit als auch ohne ECC-Zertifikate für ID.NK.VPN, ID.AK.AUT und ID.SAK.AUT unterstützen.**[<=]**

Die Anforderung ist für die Anwendungsfälle Registrierung, IPsec-Authentisierung und Autorisierung beim VPN-Zugangsdienst, TLS-Authentisierung zum eHealth-Kartenterminal, TLS-Authentisierung zum Primärsystem nachzuweisen. Wenn RSA-2048 in der TI abgekündigt wird, entfällt dadurch die Anforderung.

TIP1-A_5393 - Dokumentation der Konnektorzertifikatszuordnungen
Der Hersteller des Konnektors MUSS die Zuordnung von Konnektor und jeweils eingebrachtem C.NK.VPN-Zertifikat mit dem Ziel dokumentieren, anhand eines Sperrauftrages für einen Konnektor, das zu sperrende C.NK.VPN-Zertifikat identifizieren zu können.
[<=]

Das bedeutet, dass der Konnektorhersteller je Konnektor die für die Identifikation des C.NK.VPN-Zertifikates relevanten Daten wie z. B. Seriennummer des Konnektors und Art der verbauten Komponenten, Seriennummer der gSMC-K, etc. für seinen Sperrprozesse dokumentieren muss.

TIP1-A_5394 - Bereitstellen eines Konnektorsperrprozesses
Der Hersteller des Konnektors MUSS für die von ihm verantworteten Konnektoren einen Sperrprozess etablieren, unterhalten und der gematik zugänglich machen.
Der Hersteller des Konnektors KANN die operative Durchführung des Sperrprozesses an Dritte delegieren.
[<=]

Sperrberechtigt ist die gematik im Rahmen des Change-Verfahrens (siehe [gemRL_Betr_TI#5.4).

TIP1-A_5395 - Sperrberechtigung der gematik gegenüber Konnektorhersteller
Der Hersteller des Konnektors MUSS im Rahmen der Change-Durchführung erteilte Sperraufträge der gematik fristgemäß (gemäß Change-Auftrag) bei dem TSP X.509 nonQES (Zertifikatsaussteller) umsetzen.
[<=]

Dazu bedient er die standardmäßige Schnittstelle zum TSP (siehe [gemSpec_X.509_TSP#TIP1-A_3643]).

TIP1-A_5396 - Prüfung des Sperrauftrages für Konnektoren
Der Hersteller des Konnektors MUSS vor der Umsetzung des Sperrauftrages für einen Konnektor die Sperrberechtigung des Beauftragenden prüfen und verhindern, dass Konnektoren missbräuchlich gesperrt werden.
[<=]

TIP1-A_5397 - Umsetzung von Sperraufträgen für Konnektoren
Der Hersteller des Konnektors MUSS nach erfolgreicher Prüfung der Sperrberechtigung des Beauftragenden die Sperrung der entsprechenden C.NK.VPN-Zertifikate unverzüglich bei dem TSP X.509 nonQES (Zertifikatsaussteller) beauftragen.
[<=]

TIP1-A_5398 - Beschränkung der Sperrberechtigung des Konnektorherstellers
Der Hersteller des Konnektors DARF NICHT die Sperrung von C.NK.VPN-Zertifikaten bei dem TSP X.509 nonQES (Zertifikatsaussteller) beauftragen, wenn er nicht durch einen für den Konnektor Sperrberechtigten dazu beauftragt wurde.
[<=]

TIP1-A_5399 - Protokollierung der Sperrung von Konnektoren
Der Hersteller des Konnektors MUSS die Durchführung der Sperrung eines Konnektors protokollieren und der gematik auf Anfrage übermitteln.
Dabei MÜSSEN folgende Informationen protokolliert werden:

- Zeitpunkt der Beantragung und Umsetzung der Sperrung
- Grund der Sperrung
- Konnektoridentifikation

[<=]

Der Hersteller des Konnektors übernimmt im Rahmen der organisatorischen Sperrung die Aufgabe der Anwenderkommunikation gegenüber den betroffenen Anwendern. Die Eckpunkte zur Kommunikation sind Bestandteil des Beschlusses zur Außerbetriebnahme einer Konnektor-Baureihe und im Rahmen des Change-Verfahrens zwischen den Beteiligten abgestimmt.

TIP1-A_5400 - Fortführen des Konnektor-Sperrprozesses
Der Hersteller des Konnektors MUSS die Fortführung des Sperrprozesses über die Einstellung seiner Geschäftstätigkeit hinaus gewährleisten.
[<=]

Dies kann bspw. durch Übertragung der Aufgabe an einen Dritten realisiert werden. Dabei sind die Zuordnungen Konnektor zu Zertifikat gemäß Anforderung „Dokumentation der Konnektorzertifikatszuordnungen" zur Verfügung zu stellen.

Bei der Schlüsselerzeugung für die gSMC-K muss insbesondere auch mit technischen Maßnahmen die Vertraulichkeit der relevanten Schlüssel sichergestellt werden:

TIP1-A_7225 - Schlüsselerzeugung bei einer Schlüsselspeicherpersonalisierung
Der Hersteller des Konnektors, der Schlüssel für die gSMC-K erzeugt, MUSS diese Schlüssel mittels eines technischen Sicherheitsmoduls (HSM, Chipkarte, TPM etc.) erzeugen, welches

1. über einen Zugriffsschutz verfügt, sodass nur Berechtigte Schlüssel darauf nutzen können,
2. in einem zutrittsgeschützten Bereich aufbewahrt wird und
3. mindestens nach FIPS 140-2 Level 3 oder [COS-G2] (CC-zertifizierte Chipkarte der TI) zertifiziert ist.

Wird für die Schlüsselerzeugung eine Schlüsselableitung verwendet, so MUSS die Schlüsselableitung die fachlichen Anforderungen aus GS-A_5386 erfüllen.
Es ist zulässig, dass asymmetrische Schlüssel bei der Personalisierung auf der gSMC-K selbst erzeugt werden und symmetrische Schlüssel mittels einer Schlüsselableitung erzeugt werden, bei dem sich der Ableitungsschlüssel (Masterkey) innerhalb eines nach 3. zulässigen Hardwaresicherheitsmoduls befindet.
Es ist zulässig, sicherheitstechnisch geeignete Maßnahmen zur Sicherstellung der Verfügbarkeit der Ableitungsschlüssel (Masterkey) umzusetzen (bspw. Shamir Secret-Sharing-Verfahren).
Der Hersteller des Konnektors MUSS die Schlüsselerzeugung und die Schlüsselverwaltung in einem Konzept darstellen, das die technischen und organisatorischen Maßnahmen beschreibt, die den Schutzbedarf der verarbeiteten Informationsobjekte befriedigen. Der Hersteller des Konnektors MUSS dieses Konzept der gematik zur Verfügung stellen.**[<=]**

TIP1-A_5703 - Geschützte Übertragung von Daten zum Kartenpersonalisierer
Der Hersteller des Konnektors, der Daten für die gSMC-K erzeugt (bspw. Schlüssel), MUSS diese Daten bei der Übertragung zum Kartenpersonalisierer hinsichtlich Vertraulichkeit, Authentizität und Integrität mit einem Verfahren nach [gemSpec_Krypt] schützen.
[<=]

3.2 Bootup-Phase

TIP1-A_4507 - Isolation während der Bootup-Phase
Da während der Bootup-Phase des Konnektors noch nicht alle Sicherheitsmechanismen ihre Leistung erbringen können, DÜRFEN die Dienste des Konnektors während dem Bootup über physikalische Schnittstellen von außen NICHT erreichbar sein.
[<=]

TIP1-A_4508 - Konnektorzustand nach Bootup
Der Konnektor MUSS nach Beendigung der Bootup-Phase die Initialisierung der Funktionsmerkmale durchlaufen haben. Die Startreihenfolge der Funktionsmerkmale kann unter Berücksichtigung von TIP1-A_4507 herstellerspezifisch gestaltet werden.
Im Rahmen der Bootup-Phase MÜSSEN folgende TUCs ausgeführt werden: TUC_KON_025, TUC_KON_035, TUC_KON_272, TUC_KON_341, TUC_KON_343, TUC_KON_352 (die Reihenfolge der TUC-Ausführung ist herstellerspezifisch).
Treten während der Bootup-Phase Fehler auf, so MUSS die Bootup-Phase, sofern möglich, abgeschlossen werden.
Sobald die Bootup-Phase abgeschlossen ist, MUSS TUC_KON_256 „Systemereignis absetzen“ mit folgenden Parameter aufgerufen werden:

```
TUC_KON_256 {
   topic = "BOOTUP/BOOTUP_COMPLETE";
   eventType = Op;
```

```
severity = Info;
}
```
[<=]

Die hier gelisteten TUCs bilden nicht die abschließende Menge der während der Bootup-Phase zu erfüllenden Anforderungen. In den einzelnen Funktionsmerkmalen werden weitere Einzelanforderungen erhoben, die als Ausführungszeitpunkt die Bootup-Phase benennen (siehe Unterkapitel „Betriebsaspekte" der einzelnen Funktionsmerkmal-Kapiteln, sowie Kapitel 4.3 Konnektormanagement).

3.3 Betriebszustand

TIP1-A_4509 - Betriebszustand erfassen
Der Konnektor MUSS seinen Betriebszustand gemäß Tabelle TAB_KON_503 Betriebszustand_Fehlerzustandsliste über Fehlerzustände $EC erfassen.
Tritt die in Spalte „Beschreibung" charakterisierte Fehlersituation eines Fehlerzustandes $EC ein, wird sein Wert $EC.value = true. Sobald die Fehlersituation beendet ist, springt der Wert auf $EC.value = false. Die Fehlerzustände müssen dabei innerhalb der „max. Feststellungszeit" (Tabellenspalte) erfasst werden. Eine maximale Feststellungszeit von einen Tag (1 day) verlangt, dass einmal am Tag der Zustand geprüft werden muss, unabhängig davon, welche TUCs aufgerufen werden. Eine maximale Feststellungszeit von 1 sec, 10 sec, 1 min und 300 sec verlangt, dass nach der Feststellung einer Fehlfunktion innerhalb eines TUCs die Zustandsänderung innerhalb der angegebenen Zeit stattfinden muss.
Nach Abschluss des Boot-Vorgangs müssen sämtliche Fehlerzustände mit einer „max. Feststellungszeit" von „1 day" erfasst worden sein.
[<=]

TIP1-A_4597 - Unterstützung von Missbrauchserkennungen
Der Konnektor MUSS zur Unterstützung von Missbrauchserkennungen für alle Operationen, die in EVT_MONITOR_OPERATIONS gelistet sind und deren Alarmwert > 0 ist, kontinuierlich folgende Aktivitäten durchlaufen:

1. Minütlich gleitende 10-Minuten-Summe je in EVT_MONITOR_OPERATIONS gelistete Operation berechnen. Dazu gehen
   - erfolgreiche Abschlüsse der Operation mit dem OK_Val der Operation ein
   - eine fehlerhaft beendete Operation mit dem NOK_Val der Operation ein
2. Überschreitet der gleitende 10-Minuten-Summenwert einer in EVT_MONITOR_OPERATIONS gelisteten Operation den zugehörigen Alarmwert, so setze EC_CRYPTOPERATION_ALARM auf True.

[<=]

Erklärung „Minütlich gleitende 10-Minuten-Summe": Für die jeweilige Operation wird die Summe aller OK_Val und NOK_Val der letzten 10 Minuten gebildet. Diese Summe wird jede Minute neu berechnet.

TIP1-A_4512-03 - Ereignis bei Änderung des Betriebszustandes
Der Konnektor MUSS per Ereignisdienst TUC_KON_256 über Änderungen des Betriebszustandes (Tabelle TAB_KON_503 Betriebszustand_Fehlerzustandsliste) informieren.
Der Konnektor muss dazu für jeden Fehlerzustand $EC mit Error Condition $EC.errorcondition mit verändertem Wert $EC.value den technischen Anwendungsfall TUC_KON_256 „Systemereignis absetzen" mit folgenden Parametern aufrufen:

```
TUC_KON_256 {
   topic = "OPERATIONAL_STATE/$EC.errorcondition";
   eventType = $EC.type;
   severity = $EC.severity;
   parameters = („Value=$EC.value, $EC.parameterlist”)
}
```

Tabelle 5: TAB_KON_503 Betriebszustand_Fehlerzustandsliste

ErrorCondition (siehe Hinweis 1)	Beschreibung	Type	Severity	max. Feststellungs-zeit	Parameterlist (siehe Hinweis 2)
EC_CardTerminal_ Software_Out_Of_ Date ($ctId)	Software auf Kartenterminal($ctId) ist nicht aktuell	Op	Info	1 day	CtID=$ctId; Bedeutung= $EC.description
EC_CardTerminal_ gSMC-KT_Certificate_ Expires_Soon ($ctId)	Das Zertifikat der gSMC-KT im Kartenterminal($ctId) läuft in weniger als 5 Wochen ab	Op	Info	7 days	CtID=$ctId; Bedeutung= $EC.description
EC_Connector_ Software_Out_ Of_Date	I_KSRS_Download::list_ Updates liefert mindestens eine UpdateInformation mit einer UpdateInformation/Firmware/ FWVersion > aktuelle Version der Konnektorsoftware, deren UpdateInformation/Firmware/ FWPriority = „Kritisch“	Op	Info	1 day	Bedeutung= $EC.description
EC_FW_Update_Availabl e	I_KSRS_Download::list_ Updates liefert mindestens eine UpdateInformation mit einer UpdateInformation/Firmware/ FWVersion > aktuelle Version der Konnektor- oder Kartenterminalsoftware	Op	Info	1 day	Bedeutung= $EC.description

EC_FW_Not_ Valid_Status_ Blocked	Konnektor Firmware muss aktualisiert werden. Zugang zur TI momentan nicht erlaubt.	Sec	Fatal	1 day	Bedeutung= $EC.description
EC_Time_Sync_ Not_Successful	der letzte Synchronisationsversuch der Systemzeit war nicht erfolgreich.	Op	Info	1 sec	LastSyncAttempt= $lastSyncAttempt Timestamp; LastSyncSuccess =$lastSyncSuccess Timestamp; Bedeutung= $EC.description
EC_TSL_Update_ Not_Successful	das letzte Update der TSL war nicht erfolgreich.	Op	Info	1 sec	Bedeutung= $EC.description; LastUpdateTSL= $lastUpdateTSL Timestamp
EC_TSL_Expiring	Systemzeit t mit t > NextUpdate-Element der TSL – 7 Tage und t <= NextUpdate-Element der TSL	Sec	Info	1 day	NextUpdateTSL =$NextUpdate-Element der TSL; Bedeutung= $EC.description
EC_BNetzA_VL_ Update_ Not_Successful	Das letzte Update der BNetzA-VL war nicht erfolgreich	Op	Info	1 sec	LastUpdateBNetz AVL= $lastUpdateBNetz AVLTimestamp; Bedeutung= $EC.description
EC_ BNetzA_VL_ not_valid	Systemzeit t mit t > NextUpdate-Element der BNetzA-VL	Sec	War ning	1 day	NextUpdateBNetz AVL =$NextUpdate-Element der BNetzA-VL; Bedeutung= $EC.description
EC_TSL_Trust_ Anchor_Expiring	Gültigkeit des Vertrauensankers ist noch nicht abgelaufen, läuft aber innerhalb von 30 Tagen ab.	Sec	Info	1 day	ExpiringDateTrust Anchor= Ablaufdatum der Vertrauensanker gültigkeit; Bedeutung= $EC.description

EC_LOG_ OVERFLOW	Wenn im Rahmen der Regeln für die rollierende Speicherung von Logging-Einträgen Einträge gelöscht werden, die nicht älter als SECURITY_LOG_DAYS, LOG_DAYS bzw. FM_<fmName>_LOG_DAYS sind, tritt der Fehlerzustand ein. Der Fehlerzustand kann nur durch einen Administrator wieder zurückgesetzt werden. Unter Protokoll wird die Liste der auslösenden Protokolle angegeben.	Op	War ning	1 sec	Protokoll=$Protokoll; Bedeutung= $EC.description
EC_CRL_Expiring	Systemzeit t > NextUpdate der CRL – 3 Tage	Sec	War ning	1 day	ExpiringDateCRL= NextUpdate der CRL; Bedeutung= $EC.description
EC_Time_Sync_ Pending_Warning	MGM_LU_ONLINE=Enabled und keine erfolgreiche Synchronisation der Systemzeit seit d Tagen und d > NTP_WARN_PERIOD und d <= NTP_GRACE_PERIOD. Nach einer Korrektur oder Bestätigung der Systemzeit durch einen Administrator muss der Konnektor wie nach einer erfolgreichen Zeitsynchronisation verfahren, d.h., der Tagezähler wird auf 0 zurückgesetzt.	Sec	War ning	1 day	LastSyncSuccess= $lastSyncSuccess Timestamp; Bedeutung= $EC.description

EC_TSL_Out_ Of_Date_Within_ Grace_Period	Systemzeit t mit t > NextUpdate-Element der TSL und t <= NextUpdate-Element der TSL + CERT_TSL_ DEFAULT_GRACE_ PERIOD_DAYS und eine neue TSL ist nicht verfügbar	Sec	War ning	1 day	NextUpdateTSL =$NextUpdate-Element der TSL; GracePeriodTSL =CERT_TSL_ DEFAULT_ GRACE_PERIOD_ DAYS; Bedeutung= $EC.description
EC_CardTerminal_ Not_Available ($ctId)	Kartenterminal($ctId) ist nicht verfügbar. Dieser Betriebszustand bezieht sich auf die als „aktiv" gekennzeichneten KTs.	Op	Error	1 sec	CtID=$ctId; Bedeutung= $EC.description
EC_No_VPN_TI_ Connection	Kein sicherer Kanal (VPN) in die Telematikinfrastruktur aufgebaut. Der Wert 300 sec ist abgeleitet aus der maximalen Verbindungsaufbauzeit bei einem Standortausfall des VPN-Zugangsdienstes.	Op	Error	300 sec	Bedeutung= $EC.description
EC_No_VPN_ SIS_Connection	Kein sicherer Kanal (VPN) zu den Sicheren Internet Services aufgebaut. Der Wert 300 sec ist abgeleitet aus der maximalen Verbindungsaufbauzeit bei einem Standortausfall des VPN-Zugangsdienstes.	Op	Error	300 sec	Bedeutung= $EC.description
EC_No_Online_ Connection	Konnektor kann Dienste im Transportnetz nicht erreichen.	Op	Error	10 sec	Bedeutung= $EC.description
EC_IP_Adresses_ Not_Available	Die IP-Adressen des Netzkonnektors sind nicht oder falsch gesetzt.	Sec	Error	1 sec	Bedeutung= $EC.description

EC_CRL_Out_Of_ Date	Systemzeit t > Next Update der CRL	Sec	Fatal	1 day	NextUpdateCRL= $NextUpdate der CRL; Bedeutung= $EC.description
EC_Firewall_Not_ Reliable	Firewall-Regeln konnten nicht fehlerfrei generiert werden oder beim Laden der Firewall-Regeln ist ein Fehler aufgetreten.	Sec	Fatal	1 sec	Bedeutung= $EC.description
EC_Random_ Generator_ Not_Reliable	Der Zufallszahlengenerator kann die Anforderungen an die zu erzeugende Entropie nicht erfüllen.	Sec	Fatal	1 sec	Bedeutung= $EC.description
EC_Secure_ KeyStore_ Not_Available	Sicherer Zertifikats- und Schlüsselspeicher des Konnektors (gSMC-K oder Truststore) nicht verfügbar	Sec	Fatal	1 sec	Bedeutung= $EC.description
EC_Security_ Log_ Not_Writable	Das Sicherheitslog kann nicht geschrieben werden.	Op	Fatal	1 sec	Bedeutung= $EC.description
EC_Software_ Integrity_ Check_Failed	Eine oder mehrere konnektorinterne Integritätsprüfungen der aktiven Konnektorbestandteile sind fehlgeschlagen.	Sec	Fatal	1 day	Bedeutung= $EC.description
EC_Time_ Difference_ Intolerable	Abweichung zwischen der lokalen Zeit und der per NTP empfangenen Zeit bei der Zeitsynchronisation größer als NTP_MAX_ TIMEDIFFERENCE. Nach einer Korrektur oder Bestätigung der Systemzeit durch einen Administrator muss der Konnektor den Fehlerzustand zurücksetzen.	Sec	Fatal	1 sec	NtpTimedifference= Zeitabweichung; NtpMaxAllowed Timedifference =NTP_MAX_ TIMEDIFFERENCE; Bedeutung= $EC.description

EC_Time_Sync_ Pending_Critical	MGM_LU_ONLINE= Enabled und keine erfolgreiche Synchronisation der Systemzeit seit d Tagen und d > NTP_GRACE_PERIOD Nach einer Korrektur oder Bestätigung der Systemzeit durch einen Administrator muss der Konnektor wie nach einer erfolgreichen Zeitsynchronisation verfahren, d.h., der Tagezähler wird auf 0 zurückgesetzt.	Sec	Fatal	1 day	LastSyncSuccess =$lastSync SuccessTimestamp; NtpGracePeriod= NTP_GRACE_ PERIOD; Bedeutung= $EC.description
EC_TSL_Trust_ Anchor_ Out_Of_Date	Gültigkeit des Vertrauensankers ist abgelaufen	Sec	Fatal	1 day	ExpiringDateTrust Anchor= Ablaufdatum der Vertrauensanker gültigkeit; Bedeutung= $EC.description
EC_TSL_Out_ Of_Date_ Beyond_Grace_ Period	Systemzeit t mit t > NextUpdate-Element der TSL + CERT_TSL_DEFAULT_ GRACE_ PERIOD_DAYS und eine neue TSL ist nicht verfügbar	Sec	Fatal	1 day	NextUpdateTSL =$NextUpdate-Element der TSL; GracePeriodTSL =CERT_TSL_ DEFAULT_ GRACE_PERIOD_ DAYS; Bedeutung= $EC.description

EC_ CRYPTOP ERATION_ ALARM	Gemäß TIP1-A_4597 wurde ein potentieller Missbrauch einer Kryptooperation erkannt. Nur der Administrator kann die Alarmmeldung zurücksetzen.	Sec	War ning	1 min	Operation= $Operationsname; Count=$Summenwe rt; Arbeitsplatz= $<Liste operationsaufrufend en workplaceIDs>; Meldung= 'Auffällige Häufung von Operationsaufrufen in den letzten 10 Minuten'
EC_OTHER_ ERROR_ STATE($no)	Herstellerspezifische Fehlerzustände, die per $no (von 1 aufsteigend nummeriert) identifiziert werden. $Type, $Severity und $ParameterList legt der Hersteller nach Bedarf fest.	$Typ e	$Sev erity	<= 1 day	Bedeutung= $EC.description
EC_NK_Certificate_Ex piring	Das C.NK.VPN-Zertifikat läuft bald ab. Systemzeit t > (Ablaufdatum von C.NK.VPN – 180 Tage)	Sec	Warni ng	1 day	Iccsn=$Iccsn; Serial=$Serialnu mber; Bedeutung= $EC.description
EC_NK_Certificate_Ex pired	Das C.NK.VPN-Zertifikat ist abgelaufen. Systemzeit t > Ablaufdatum von C.NK.VPN	Sec	Fatal	1 day	Iccsn=$Iccsn; Serial=$Serialnu mber; Bedeutung= $EC.description

Erläuterungen zu TAB_KON_503:

Hinweis 1:
Jeder Fehlerzustand wird durch einen eindeutigen ErrorCondition identifiziert. Dieser kann einen Parameter enthalten. Sind etwa die Kartenterminals mit ctId=47 und das mit ctId=93 nicht erreichbar, so lauten die ErrorCondition „EC_CardTerminal_Not_Available(47)" und „EC_CardTerminal_Not_Available(93)".

Hinweis 2:
EC.description referenziert den Text, der in der Spalte „Beschreibung" des Zustandes spezifiziert wurde.

[<=]

Unter „kartenbasiert" sind nicht nur Lösungen mit Smartcards sondern auch solche mit HSMs (Hardware Security Modules) zu verstehen.

A_17085 - Bedingung für den Fehlerzustand EC_No_VPN_TI_Connection
Wenn MGM_LU_ONLINE=Enabled nicht erfüllt ist, DARF der Konnektor den Zustand EC_No_VPN_TI_Connection NICHT annehmen.**[<=]**

A_17086 - Bedingung für den Fehlerzustand EC_No_VPN_SIS_Connection
Wenn MGM_LU_ONLINE=Enabled oder ANLW_INTERNET_MODUS=SIS nicht erfüllt ist, DARF der Konnektor den Zustand EC_No_VPN_SIS_Connection NICHT annehmen.**[<=]**

A_17087 - Bedingung für den Fehlerzustand EC_No_Online_Connection
Wenn MGM_LU_ONLINE=Enabled nicht erfüllt ist, DARF der Konnektor den Zustand EC_No_Online_Connection NICHT annehmen.**[<=]**

Tabelle 6: TAB_KON_504 Ausführungserlaubnis für Dienste in kritischen Fehlerzuständen

	EC_Software_Integrity_Check_Failed	EC_Random_Generator_Not_Reliable	EC_Security_Log_Not_Writable	EC_Time_Sync_Pending_Critical	EC_Time_Difference_Intolerable	EC_CRL_Out_Of_Date	EC_TSL_Out_Of_Date_Beyond_Grace_Period	EC_TSL_Trust_Anchor_Out_Of_Date	EC_Secure_Key_Store_Not_Available	EC_FW_Not_Valid_Status_Blocked	EC_NK_Certificate_Expired
Technische Use Cases (TUCs) der Basisdienste relevant für Fachanwendung und die Kommunikation mit Weiteren Anwendungen und SIS											
Zugriffsberechtigungsdienst											
TUC_KON_000 Prüfe Zugriffsberechtigung	-	x	x	x	x	x	x	x	x	x	x
Dienstverzeichnisdienst											
TUC_KON_041 Einbringen der Endpunktinformationen während der Bootup-Phase	-	-	-	x	x	x	x	x	x	x	x

Kartenterminaldienst											
TUC_KON_051 Mit Anwender über Kartenterminal interagieren	-	-	-	-	-	x	x	x	-	x	-
Kartendienst											
TUC_KON_005 Card-to-Card authentisieren	-	-	-	-	-	x	x	x	-	x	-
TUC_KON_006 Datenzugriffsaudit eGK schreiben	-	-	-	-	-	x	x	x	-	x	-
TUC_KON_018 eGK-Sperrung prüfen	-	-	-	-	-	x	x	x	-	x	-
TUC_KON_024 Karte zurücksetzen	-	-	-	-	-	x	x	x	-	x	-
TUC_kON_026 Liefere CardSession	-	-	-	-	-	x	-	x	-	-	-
TUC_KON_200 SendeAPDU	-	-	-	-	-	x	x	x	-	x	-
TUC_KON_202 LeseDatei	-	-	-	-	-	x	x	x	-	x	-
TUC_KON_203 SchreibeDatei	-	-	-	-	-	x	x	x	-	x	-
TUC_KON_209 LeseRecord	-	-	-	-	-	x	x	x	-	x	-
Systeminformationsdienst											
TUC_KON_256 Systemereignis absetzen	-	x	x	x	x	x	x	x	x	x	x
Verschlüsselungsdienst											
TUC_KON_072 Daten symmetrisch verschlüsseln	-	-	-	x	x	x	x	x	-	x	-

TUC_KON_073 Daten symmetrisch entschlüsseln	-	-	-	x	x	x	x	x	-	x	-
Zertifikatsdienst											
TUC_KON_034 Zertifikatsinformationen extrahieren	-	-	-	x	x	x	x	x	-	x	x
Protokollierungsdienst											
TUC_KON_271 Schreibe Protokolleintrag	-	x	x	x	x	x	x	x	x	x	x
TLS-Dienst											
TUC_KON_110 Kartenbasierte TLS-Verbindung aufbauen	-	-	-	-	-	-	-	-	-	-	-
Verbindung zum VPN-Konzentrator											
TUC_VPN-ZD_0001 „IPsec Tunnel TI aufbauen”	-	-	-	-	-	-	-	-	-	-	-
TUC_VPN-ZD_0002 „IPsec Tunnel SIS aufbauen”	-	-	-	-	-	-	-	-	-	-	-
Feature Laufzeitverlängerung gSMC-K)											
TUC_KON_410 „gSMC-K-Zertifikate aktualisieren (automatisch)	-	-	-	-	-	-	-	-	-	-	-
TUC_KON_410 „gSMC-K-Zertifikate aktualisieren (manuell)	-	-	-	-	-	-	-	-	-	-	x
TUC_KON_411 „Konnektor mit neuem NK-Zertifikat registrieren (automatisch)	-	-	-	-	-	-	-	-	-	-	-

TUC_KON_411 „Konnektor mit neuem NK-Zertifikat registrieren (manuell)	-	-	-	-	-	-	-	-	-	-	x
Operationen der Basisdienste											
Kartendienst											
VerifyPin	-	-	-	-	-	x	x	x	-	x	-
UnblockPin	-	-	-	-	-	x	x	x	-	x	-
ChangePin	-	-	-	-	-	x	x	x	-	x	-
GetPinStatus	-	-	-	-	-	x	x	x	-	x	-
Systeminformationsdienst											
Schnittstelle der Ereignissenke	-	x	x	x	x	x	x	x	x	x	x
GetCardTerminals	-	x	x	x	x	x	x	x	x	x	-
GetCards	-	x	x	x	x	x	x	x	x	x	-
GetResourceInformation	-	x	x	x	x	x	x	x	x	x	-
Subscribe	-	x	x	x	x	x	x	x	x	x	-
RenewSubscription	-	x	x	x	x	x	x	x	x	x	-
Unsubscribe	-	x	x	x	x	x	x	x	x	x	-
GetSubscription	-	x	x	x	x	x	x	x	x	x	-
Verschlüsselungsdienst											
EncryptDocument	-	-	-	-	-	x	x	x	-	x	-
DecryptDocument	-	-	-	-	-	x	x	x	-	x	-
Signaturdienst											
SignDocument	-	-	-	-	-	x	x	x	-	x	-

VerifyDocument	-	-	-	-	-	x	x	x	-	x	-
GetJobNumber	-	-	-	-	-	x	x	x	-	x	-
StopSignature	-	-	-	-	-	x	x	x	-	x	-
ActivateComfortSignature	-	-	-	-	-	x	x	x	-	x	-
DeactivateComfortSignature	-	-	-	-	-	x	x	x	-	x	-
GetSignatureMode	-	-	-	-	-	x	x	x	-	x	-
Authentifizierungsdienst											
ExternalAuthenticate	-	-	-	-	-	x	x	x	-	x	-
Zertifikatsdienst											
ReadCardCertificate	-	-	-	-	-	x	x	x	x	x	-
CheckCertificateExpiration	-	-	-	-	-	x	x	x	x	x	-
VerifyCertificate	-	-	-	-	-	x	-	x	x	x	-
Zeitdienst											
I_NTP_Time_Information	-	-	-	-	-	x	x	x	x	-	-
Konnektormanagement											
Softwareaktualisierung	x	x	x	x	x	x	x	x	x	x	x
Protokolleinsicht	x	x	x	x	x	x	x	x	x	x	x
Werksreset	x	x	x	x	x	x	x	x	x	x	x
Sonstiges	-	x	x	x	x	x	x	x	x	x	x

In den kritischen Fehlerzuständen, in denen keine TLS-Verbindung ins LAN aufgebaut werden (EC_Random_Generator_Not_Reliable, EC_Software_Integrity_Check_Failed, EC_Security_Log_Not_Writable, EC_Time_Sync_Pending_Critical, EC_Time_Difference_Intolerable), kann keine Verbindung zu den Kartenterminals aufgebaut werden. Infolge sind hier keine Kartenoperationen zugelassen.

Wenn keine Verbindung zum VPN-Konzentrator des SIS aufgebaut werden kann, ist dadurch das Internet nicht über den Konnektor erreichbar. Wenn keine Verbindung zum VPN-Konzentrator der TI aufgebaut werden kann, sind Bestandsnetze nicht erreichbar.

Bezüglich der Administration des Konnektors im Zustand EC_FIREWALL_NOT_RELIABLE ist eine Abstimmung mit der Prüfstelle und der Zertifizierungsstelle notwendig.

A_16203 - Nutzbarkeit im Zustand EC_FIREWALL_NOT_RELIABLE
Im Zustand EC_Firewall_Not_Reliable DARF der Konnektor NICHT nutzbar sein. Möglichkeiten zur Behebung des Zustandes EC_Firewall_Not_Reliable sind mit dem CC - Evaluierer und Zertifizierer abzustimmen.**[<=]**

Die Architektur der TI ist so angelegt, dass die Fehlerzustände mit Severity=Fatal in den Tabellen TAB_KON_504 und TAB_KON_503 mit vernachlässigbarer Wahrscheinlichkeit von externen Einflüssen abhängen. Die SLAs für Dienste der zentralen TI-Plattform sind so gefasst, dass diese schwerwiegend verletzt werden müssten, um dadurch einen Konnektor in einen solchen kritischen Zustand zu bringen (externer Fehler aus Sicht des Konnektors). Dass beispielsweise der TSL-Dienst über den Zeitraum der Grace-Period-TSL (typisch: 7 Tage) nicht erreichbar ist (ErrorCondition EC_TSL_Out_Of_Date _Beyond_Grace_Period), kann nur bei massiver Verletzung der für zentrale Dienste festgelegten SLAs eintreten.

Um die konnektorinternen Fehlerquellen zu erfassen, die dazu führen, dass ein Fehlerzustand mit Severity=Fatal eintritt oder ein anderer Zustand, in dem der Konnektor nicht verwendbar ist, wird Folgendes gefordert:

TIP1-A_5148 - Performance - Konnektor - Mittlerer Abstand zwischen Ausfällen
Der Konnektorhersteller MUSS den mittleren Zeitabstand zwischen Ausfällen (MTBF) als Produkteigenschaft ausweisen. Der Konnektor soll einen mittleren Zeitabstand zwischen Ausfällen (MTBF) von mindestens 50 Jahren haben.
Ein „Ausfall" gilt dann als eingetreten, wenn

- der Konnektor nicht mehr gebootet werden kann, d. h. kein „BOOTUP/BOOTUP_COMPLETE" Event ausgelöst wird, und dies nicht auf einen externen Fehler zurückzuführen ist,
- oder sich der Konnektor in einem Fehlerzustand mit Severity=Fatal befindet, der nicht auf einen externen Fehler zurückzuführen ist,
- oder Funktionen des Konnektors ausgefallen sind, ohne dass dies auf externe Fehler zurückzuführen ist.

[<=]

Bei einem mittleren Zeitabstand zwischen Ausfällen (MTBF) von 50 Jahren ist die Wahrscheinlichkeit, dass ein Fehlerzustand mit Severity=Fatal auftritt, kleiner 2 % pro Jahr.

TIP1-A_4510-04 - Sicherheitskritische Fehlerzustände
Der Konnektor MUSS bei eingetretenem Fehlerzustand aus Tabelle Tab_ Kon_503 Betriebszustand_Fehlerzustandsliste mit Severity=Fatal dafür sorgen, dass von den Operationen der Basisdienste und Technische Use Cases (TUCs) der Basisdienste, die relevant für Fachanwendungen sind, nur erlaubte Operationen und TUCs gestartet und ausgeführt werden.
Welche Operationen und TUCs je eingetretenem Fehlerzustand ausgeführt werden dürfen, legt Tabelle „TAB_KON_504 Ausführungserlaubnis für Dienste in kritischen Fehlerzuständen" fest: Jede Erlaubnis ist dort durch ein „x" definiert.
Abweichend zu Angaben in der Tabelle TAB_KON_504 DÜRFEN folgende Operationen und TUCs NICHT im Zustand EC_Firewall_Not_Reliable ausgeführt werden:

- TUC_KON_000 PrüfeAufrufkontext
- TUC_KON_041 Einbringen der Endpunktinformationen während der Bootup-Phase
- GetCardTerminals
- GetCards
- GetResourceInformation
- Subscribe
- RenewSubscription
- Unsubscribe
- GetSubscription
- ReadCardCertificate
- CheckCertificateExpiration
- VerifyCertificate

Sind mehrere Fehlerzustände gleichzeitig eingetreten, dürfen nur die Operationen und TUCs ausgeführt werden, die für alle eingetretenen Fehlerzustände erlaubt sind. Der Konnektor MUSS Anfragen, die auf Grund eines kritischen Fehlerzustandes nicht ausgeführt oder abgebrochen werden, mit einem Fehler (Fehlercode 4002) beantworten.

Tabelle 7: TAB_KON_502 Fehlercodes „Betriebszustand"

Fehlercode	ErrorType	Severity	Fehlertext
4002	Security	Fatal	Der Konnektor befindet sich in einem kritischen Betriebszustand

[<=]

3.3.1 Betriebsaspekte

Der Konnektor soll per Signaleinrichtung am Konnektor die Fehlerzustände mit Severity „Error" und „Fatal" anzeigen (siehe [TIP1-A_4843]).

TIP1-A_4513 - Betriebszustände anzeigen und Fehlerzustände zurücksetzen
Der Konnektor MUSS es dem Administrator ermöglichen, den aktuellen Betriebszustand einzusehen und Fehlerzustände zurückzusetzen, soweit diese Möglichkeit in Tabelle „TAB_KON_503 Betriebszustand_Fehlerzustandsliste" für den jeweiligen Fehlerzustand festgelegt ist.
Ferner MUSS es die Managementschnittstelle dem Administrator ermöglichen, Konfigurationsänderungen gemäß Tabelle TAB_KON_505 vorzunehmen:

Tabelle 8: TAB_KON_505 Konfigurationswerte Missbrauchserkennung

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
EVT_MONITOR_OPERATIONS	Liste von: - Operationsname - OK_Val	Der Administrator MUSS in der Liste der zur Missbrauchserkennung überwachbaren Operationen alle Listeneinträge einsehen können. Er

	(Nummer) - NOK_Val (Nummer) - Alarmwert (Nummer)	MUSS den jeweiligen Alarmwert editieren können (0-9999, 0=deaktiviert). OK_VAL und NOK_VAL DÜRFEN durch den Administrator NICHT veränderbar sein.

[<=]

3.4 Fachliche Anbindung der Clientsysteme

Für die Schnittstellen des Konnektors zu den Clientsystemen kann gesteuert werden:

- ob die Kommunikation zwischen Konnektor und Clientsystemen hinsichtlich Vertraulichkeit, Integrität und Authentizität zwingend durch TLS gesichert werden muss
- ob sich Clientsysteme zwingend authentisieren müssen
- welche Clientsysteme auf den Konnektor zugreifen dürfen (Whitelisting)

Dabei werden die folgenden zwei Nutzungsszenarien nicht unterschieden:

- Nutzung von Fachanwendungen (in Form von Fachmodulen)
- Nutzung von Basisdiensten des Konnektors

Sowohl die Anbindung zur Administration des Konnektors, als auch die Anbindung zur Nutzung von Bestandsnetzen oder dem gesicherten Internetzugang sind nicht Gegenstand dieser Schnittstellenfestlegungen. Für die Anbindung zu Administration wird diese im Kapitel Konnektormanagement beschrieben, für die Anbindung von Bestandsnetzen bzw. dem gesicherten Internetzugang ist diese Art der Regelung nicht erforderlich, da es sich dort um Routing-Funktionen handelt.

Die seitens des Administrators einstellbaren Werte und Listen sind, der allgemeinen Struktur dieses Dokuments folgend, im Unterkapitel 3.4.1 Betriebsaspekte beschrieben.

TIP1-A_4514 - Verfügbarkeit einer TLS-Schnittstelle
Der Konnektor MUSS TLS in Richtung der Clientsysteme für alle Außenschnittstellen der Basisdienste:

- Dienstverzeichnisdienst
- Kartenterminaldienst
- Systeminformationsdienst
- Verschlüsselungsdienst
- Signaturdienst
- Zertifikatsdienst
- Kartendienst
- LDAP-Proxy

unterstützen.
Ferner MUSS der Konnektor für die SOAP-Endpunkte der Fachmodule TLS unterstützen.

Der Konnektor MUSS sich mittels ID.AK.AUT gegenüber dem Client authentisieren.
[<=]

TIP1-A_4515 - Verpflichtung zur Nutzung der TLS-Verbindung
Der Konnektor MUSS immer TLS-Verbindungsanfragen von Clientsystemen annehmen. Der Konnektor MUSS bei gesetzter Variable ANCL_TLS_MANDATORY = Enabled den Verbindungsversuch von Clientsystemen ohne TLS ablehnen. Ausgenommen hiervon sind Anfragen an den Dienstverzeichnisdienst bei gesetzter Variable ANCL_DVD_OPEN = Enabled.
[<=]

TIP1-A_4516 - Authentifizierung der Clients über Basic-Auth und X.509-Zertifikate
Der Konnektor MUSS zur Client-Authentifizierung die Verfahren Basic Authentication (Username/Password) [RFC2617] über HTTP/TLS [RFC2818] und zertifikatsbasierte Client-Authentifizierung (X.509) [gemSpec_PKI#8.3.1.4] über TLS anbieten.
Dabei MUSS für eine erfolgreiche Prüfung bei Basic Authentication:

- das seitens des Clientsystems präsentierte Credential in ANCL_CUP_LIST enthalten sein

Für eine erfolgreiche Prüfung mit zertifikatsbasierter Client-Authentifizierung MUSS:

- das seitens des Clientsystems präsentierte Zertifikat in ANCL_CCERT_LIST enthalten sein
- die Zertifikatsprüfung (nur Prüfung auf „mathematische Korrektheit“ und „Gültigkeit nicht abgelaufen“) erfolgreich durchlaufen werden

Schlägt die Prüfung fehl, MUSS der Verbindungsversuch des Clientsystem abgelehnt werden.**[<=]**

Bei der Authentisierung des Clientsystems geht es um eine Authentisierung in zwei Richtungen:

1. Authentisierung des Clientsystems in der Rolle eines Clients gegenüber dem Konnektor für die Übertragung von SOAP-Requests.
2. Authentisierung des Clientsystems in der Rolle eines Servers gegenüber dem Konnektor zum Empfang von CETP-Ereignismitteillungen des Systeminformationsdienstes.

Für beide Richtungen kann das Clientsystem dasselbe Zertifikat verwenden.

TIP1-A_5009 - Authentifizierungsvarianten für Verbindungen zwischen Konnektor und Clientsystemen
Der Konnektor MUSS für Verbindungen zu Clientsystemen als Authentifizierungsmethode ausschließlich folgende Varianten erlauben:

1. Für Verbindungen mit dem Konnektor in der Rolle des Servers (SOAP-Requests):
   - TLS-Server-Authentifizierung des Konnektors und TLS-Client-Authentifizierung des Clientsystems
   - TLS-Server-Authentifizierung des Konnektors und BasicAuthentifizierung des Clientsystems
   - TLS-Server-Authentifizierung des Konnektors ohne TLS-Client-Authentifizierung des Clientsystems
   - Keine Authentifizierung des Konnektors und des Clientsystems
2. Für Verbindungen mit dem Konnektor in der Rolle des Clients (CETP-Protokoll):

- TLS-Server-Authentifizierung des Clientsystems und TLS-Client-Authentifizierung des Konnektors
- TLS-Server-Authentifizierung des Clientsystems ohne TLS-Client-Authentifizierung des Konnektors
- Keine Authentifizierung des Konnektors und des Clientsystems

Alle anderen Verbindungsversuche von Clientsystemen MÜSSEN vom Konnektor abgelehnt werden.
[<=]

Für die Anbindung der Clientsysteme ergeben sich verschiedene Konfigurationsvarianten bezüglich der Absicherung der Verbindungen zwischen Konnektor und Clientsystemen. Tabelle TAB_KON_852 listet die Varianten für die Verbindungen zum Aufruf der WebService-Schnittstellen (Varianten SOAP1 bis SOAP4), für die Verbindungen zum Senden von Events (Varianten CETP1 und CETP2) und für Verbindungen zum Abruf des Dienstverzeichnisses (Varianten DVD1, DVD2 und DVD3).

Tabelle 9: TAB_KON_852 Konfigurationsvarianten der Verbindungen zwischen Konnektor und Clientsystemen

Konfigu-rations-variante	ANCL_ TLS_ MAN-DATORY	ANCL_ CAUT_ MAN-DATORY	ANCL_ CAUT_ MODE	ANCL_ DVD_ OPEN	Bedeutung
CETP1	Enabled	Irrelevant	Irrelevant	Irrelevant	Der Konnektor sendet Events ausschließlich über TLS. Er authentisiert sich, wenn ihn das Clientsystem im Rahmen des TLS-Handshakes dazu auffordert.
CETP2	Disabled	Irrelevant	Irrelevant	Irrelevant	Der Konnektor sendet Events immer über eine TCP-Verbindung ohne TLS.
SOAP1	Enabled	Enabled	CERTIFICATE	Irrelevant	Der Konnektor akzeptiert vom Clientsystem nur Aufrufe über TLS. Der Konnektor verlangt beim TLS-Handshake die Authentisierung des Clientsystems per Zertifikat.
SOAP2	Enabled	Enabled	PASSWORD	Irrelevant	Der Konnektor akzeptiert vom Clientsystem nur Aufrufe über TLS. Der Konnektor prüft auf Anwendungsebene, dass Aufrufer sich per Username/Password [RFC2617] authentisieren.
SOAP3	Enabled	Disabled	Irrelevant	Irrelevant	Der Konnektor akzeptiert vom Clientsystem nur Aufrufe über TLS. Der Konnektor nimmt

					keine Clientauthentifizierung vor.
SOAP4	Disabled	Irrelevant	Irrelevant	Irrelevant	Der Konnektor akzeptiert vom Clientsystem sowohl Aufrufe ohne TLS als auch über TLS. Im zweiten Fall sollte der Konnektor das Clientsystem nicht authentifizieren, wenn er es aber für den Sonderfall ANCL_CAUT_MANDATORY=Enabled aktuell tut, sehen wir das nicht als Fehler.
DVD1	Irrelevant	Irrelevant	Irrelevant	Enabled	Zugriff auf Dienstverzeichnisdienst kann über HTTP und HTTPS erfolgen.
DVD2	Enabled	*	*	Disabled	Zugriff auf Dienstverzeichnisdienst kann nur über HTTPS erfolgen. *) Bzgl. Clientauthentisierung wirken die Schalter wie in SOAP 1, SOAP 2, SOAP 3
DVD3	Disabled	Irrelevant	Irrelevant	Disabled	Zugriff auf Dienstverzeichnisdienst kann über HTTP und HTTPS erfolgen.

A_21224 - Authentifizierung für Verbindungen zwischen Konnektor und Clientsystemen bei LDAP

Bei der Verwendung des LDAP-Proxies im Konnektor, MUSS sich der Konnektor abhängig von der Stellung der Schalter ANCL_TLS_MANDATORY, ANCL_CAUT_MANDATORY und ANCL_CAUT_MODE gemäß der Tabelle TAB_KON_860 verhalten.

Tabelle 10: TAB_KON_860 Konfigurationsvarianten der Verbindungen zwischen Konnektor und Clientsystemen bei LDAP

Konfigu-rations-variante	ANCL_ TLS_ MAN-DATORY	ANCL_ CAUT_ MAN-DATORY	ANCL_ CAUT_ MODE	Bedeutung
LDAP1	Enabled	Enabled	CERTIFICATE	Der Konnektor akzeptiert vom Clientsystem nur Aufrufe über TLS. Der Konnektor verlangt beim TLS-Handshake die Authentisierung des Clientsystems per Zertifikat.

LDAP2	Enabled	Enabled	PASSWORD	Der Konnektor akzeptiert vom Clientsystem nur Aufrufe über TLS. Der Konnektor nimmt keine Clientauthentifizierung vor.
LDAP3	Enabled	Disabled	Irrelevant	Der Konnektor akzeptiert vom Clientsystem nur Aufrufe über TLS. Der Konnektor nimmt keine Clientauthentifizierung vor.
LDAP4	Disabled	Irrelevant	Irrelevant	Der Konnektor akzeptiert vom Clientsystem sowohl Aufrufe ohne TLS als auch über TLS. Im zweiten Fall sollte der Konnektor das Clientsystem nicht authentifizieren, wenn er es aber für den Sonderfall ANCL_CAUT_MANDATORY=Enabled, ANCL_CAUT_MODE=CERTIFICATE aktuell tut, sehen wir das nicht als Fehler.

[<=]

Aus A_24 resultiert direkt, dass als Client-Authentisierung für LDAPS nur Client-Zertifikate unterstützt werden müssen. Die Authentisierung mit Username/Passwort wird bei LDAPS nicht unterstützt.

Es sei noch einmal betont, dass TIP1-A_5009 sich nur auf SOAP und CETP bezieht und TIP1-A_4516 das Basic-Authentication Verfahren nur für HTTP fordert.

3.4.1 Betriebsaspekte

Damit sich ein Clientsystem mittels X.509 authentisieren kann, muss es über ein entsprechendes Zertifikat verfügen. Diese Zertifikate kann der Administrator entweder mit seinen lokalen Mitteln selbst oder mittels des Konnektors erzeugen. In beiden Fällen müssen diese Zertifikate sowohl im Clientsystemen (hier zusammen mit ihren privaten Schlüsseln), als auch im Konnektor vorhanden sein.

Da es sich um eine lokal begrenzte Authentisierung im Verantwortungsbereich des Betreibers des lokalen Netzes handelt, werden keine weiteren Vorgaben zu den Schlüsselspeichern auf Clientsystemseite erhoben. Auch hinsichtlich der außerhalb des Konnektors erzeugten Zertifikate gelten keine weiteren Vorgaben. Ferner ist eine Online-Prüfung dieser Zertifikate nicht erforderlich.

TIP1-A_4517 - Schlüssel und X.509-Zertifikate für die Authentisierung des Clientsystems erzeugen und exportieren sowie X.509-Zertifikate importieren
Der Konnektor MUSS die Erstellung und den Export von X.509-Zertifikaten für Clientsysteme und der zugehörigen privaten Schlüssel durch den Administrator über das Managementinterface ermöglichen. Hierbei MUSS der Konnektor dem Administrator die Möglichkeit geben, das kryptographische Verfahren RSA-2048 oder ECC-256 auszuwählen. Als Exportformat MUSS PKCS#12 verwendet werden. Die so erstellten Zertifikate werden zu ANCL_CCERT_LIST angefügt.
Der Konnektor MUSS dem Administrator ferner den Import von konnektorfremden X.509-Zertifikaten für Clientsysteme über das Managementinterface ermöglichen. Die so

importierten Zertifikate werden zu ANCL_CCERT_LIST angefügt.
[<=]

TIP1-A_4518 - Konfiguration der Anbindung Clientsysteme
Der Administrator MUSS in der Managementoberfläche die in TAB_KON_506 genannten Parameter im Managementinterface konfigurieren können.
Wird ANCL_TLS_MANDATORY auf ENABLED gewechselt, MÜSSEN alle nicht per TLS gesicherten http-Verbindungen geschlossen werden, sobald die in den Verbindungen jeweils aktuell laufenden Außenschnittstelle-Operationen abgeschlossen wurden, mit Ausnahme von http-Verbindungen zum Dienstverzeichnisdienst.
Der Konnektor MUSS den Administrator geeignet und verständlich auf seine Verantwortung für die Sicherung der Kommunikation hinweisen.

Tabelle 11: TAB_KON_506 Konfigurationsparameter der Clientsystem-Authentisierung

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
ANCL_TLS_MANDATORY	Enabled/Disabled	Der Administrator MUSS die verpflichtende Verwendung eines TLS gesicherten Kanals an- oder abschalten können. Wenn ANLW_ANBINDUNGS_MODUS = Parallel MUSS der Administrator vor dem Disablen von ANCL_TLS_MANDATORY einen Warnhinweis bestätigen, der ihn über die mit der Abschaltung verbundenen Risiken informiert und darlegt, dass in diesem Fall der Nutzer die Verantwortung für die Sicherstellung der vertraulichen Übertragung übernimmt. Default-Wert: Enabled
ANCL_CAUT_MANDATORY	Enabled/Disabled	Der Administrator MUSS die verpflichtende Authentifizierung der Clientsysteme an- oder abschalten können. Default-Wert: Enabled
ANCL_CAUT_MODE	CERTIFICATE / PASSWORD	Der Administrator MUSS konfigurieren können, welcher Client Authentifizierungsmodus genutzt werden kann bzw. genutzt werden muss. Default-Wert: CERTIFICATE
ANCL_CCERT_LIST	Liste von X.509-Zertifikaten zugeordnet zu ClientID	Whitelist an importierten oder vom Konnektor erzeugten X.509-Zertifikaten und dazugehörenden Clientsystem IDs. Der Administrator MUSS die Liste der Zertifikate und den zugehörenden Clientsystemen verwalten können, der Inhalt der Zertifikate MUSS menschlich lesbar dargestellt werden.

		Es muss für den Administrator erkennbar sein, welches kryptographische Verfahren (RSA-2048 oder ECC -256) dem jeweiligen Zertifikat zugrunde liegt.
ANCL_CUP_LIST	Liste von Benutzer/Passwort Kombinationen, zugeordnet zu ClientID	Whitelist an UserCredentials und dazugehörenden Clientsystem IDs. Der Administrator MUSS eine Liste von Credentials und zugehörendem Clientsystem verwalten können. Bei diesen Benutzer-/Passwortkombinationen handelt es sich nicht um personenbezogene Credentials, sondern um clientbezogene.
ANCL_DVD_OPEN	Enabled/Disabled	Der Administrator MUSS konfigurieren können, ob der Zugriff auf den Dienstverzeichnisdienst auch dann über einen ungesicherten http-Kanal erfolgen kann (ENABLED), wenn ANCL_TLS_MANDATORY = ENABLED ist. Default-Wert: Enabled

[<=]

Damit sich der Konnektor mittels X.509 gegenüber Clientsystemen authentisieren kann, muss er über ein entsprechendes Zertifikat und dazu passendes Schlüsselmaterial verfügen. Dieses Zertifikat und Schlüsselmaterial befinden sich auf der gSMC-K (ID.AK.AUT). Der Administrator hat neben der Konfiguration zur Nutzung des erneuerten C.AK.AUT , welches vom Konnektor heruntergeladen wurde, auch mehrere Möglichkeiten, die ID.AK.AUT von der gSMC-K für die Authentisierung gegenüber den Clientsystemen zu ersetzen:

- er kann ein Zertifikat und das dazugehörige Schlüsselmaterial Konnektor-extern mit seinen lokalen Mitteln erzeugen und in den Konnektor importieren oder
- er kann ein Zertifikat und das dazugehörige Schlüsselmaterial im Konnektor erzeugen und das Zertifikat ggf. aus dem Konnektor exportieren.

Da es sich um eine lokal begrenzte Authentisierung im Verantwortungsbereich des Betreibers des lokalen Netzes handelt, werden keine weiteren Vorgaben zur Erstellung und Handhabung in der LE-Umgebung der außerhalb des Konnektors erzeugten Zertifikate und Schlüssel erhoben. Eine Online-Status-Prüfung dieser Zertifikate ist nicht erforderlich und nicht vorgesehen.

A_21811 - Vorgaben für generierte und importierte Schlüssel und Zertifikate entsprechend gemSpec_Krypt
Der Konnektor SOLL bezüglich selbst generierter und importierter Schlüssel und Zertifikate für die TLS-Authentisierung gegenüber Primärsystemen die kryptographischen Vorgaben aus gemSpec_Krypt durchsetzen.**[<=]**

Bezüglich der Nutzung von Schlüsseln und Zertifikaten basierend auf elliptischer Kurven Kryptographie, sind für die generierten und importierten Daten für die TLS-

Authentisierung gegenüber Primärsystemen neben den in gemSpec_Krypt genannten Brainpoolkurven auch die NIST-Kurven gleicher Stärke gestattet.

A_21697 - Schlüsselpaar und dazugehöriges X.509-Zertifikat für Authentisierung des Konnektors gegenüber Clientsystemen importieren
Der Konnektor MUSS dem Administrator den Import eines extern generierten Schlüsselpaars und des dazugehörigen X.509-Zertifikats für die Authentisierung des Konnektors im Rahmen von TLS-Verbindungen gegenüber Clientsystemen über das Managementinterface ermöglichen.**[**<=**]**

A_21698 - Importiertes Schlüsselpaar und dazugehöriges X.509-Zertifikat für Authentisierung des Konnektors gegenüber Clientsystemen verwenden
Der Konnektor MUSS dem Administrator das Einschalten der Verwendung von importiertem Schlüsselpaar und dazugehörigem X.509-Zertifikat für die Authentisierung des Konnektors gegenüber Clientsystemen über das Managementinterface ermöglichen.**[**<=**]**

A_21699 - Schlüssel und X.509-Zertifikate für Authentisierung des Konnektors gegenüber Clientsystemen erzeugen
Der Konnektor MUSS die Erstellung eines Schlüsselpaars und eines dazugehörigen X.509-Zertifikats für die Authentisierung des Konnektors im Rahmen von TLS-Verbindungen gegenüber den Clientsystemen durch den Administrator über das Managementinterface ermöglichen. Hierbei MUSS der Konnektor dem Administrator die Möglichkeit geben, das kryptographische Verfahren RSA-2048 oder ECC-256 auszuwählen. Ferner MUSS der Konnektor dem Administrator die Möglichkeit geben, den Hostnamen des Konnektors im Zertifikat zu vergeben.**[**<=**]**

A_21701 - X.509-Zertifikate für Authentisierung des Konnektors gegenüber Clientsystemen exportieren
Der Konnektor MUSS dem Administrator den Export von intern generierten X.509-Zertifikaten, die für die Authentisierung des Konnektors im Rahmen von TLS-Verbindungen gegenüber den Clientsystemen verwendet werden, über das Managementinterface ermöglichen. Der private Schlüssel verbleibt im Konnektor.**[**<=**]**

A_21702 - Intern generierte Schlüssel und X.509-Zertifikate für Authentisierung des Konnektors gegenüber Clientsystemen verwenden
Der Konnektor MUSS dem Administrator das Einschalten der Verwendung von intern generierten Schlüsseln und X.509-Zertifikaten für die Authentisierung des Konnektors gegenüber den Clientsystemen über das Managementinterface ermöglichen.**[**<=**]**

A_21759 - Erneuerte ID.AK.AUT für Authentisierung des Konnektors gegenüber Clientsystemen verwenden
Der Konnektor MUSS dem Administrator das Einschalten der Verwendung von erneuerten C.AK.AUT für die Authentisierung des Konnektors gegenüber den Clientsystemen über das Managementinterface ermöglichen.
Der Konnektor DARF ein erneuertes C.AK.AUT NICHT automatisch verwenden.**[**<=**]**

A_21760 - ID.AK.AUT auf gSMC-K für Authentisierung des Konnektors gegenüber Clientsystemen verwenden
Der Konnektor MUSS dem Administrator das Einschalten der Verwendung von ID.AK.AUT auf der gSMC-K für die Authentisierung des Konnektors gegenüber den Clientsystemen über das Managementinterface ermöglichen.**[**<=**]**

3.5 Clientsystemschnittstelle

TIP1-A_5401 - Parallele Nutzbarkeit Clientsystemschnittstelle
Alle Schnittstellen, die der Konnektor den Clientsystemen zur Verfügung stellt, MÜSSEN parallel durch mehrere Aufrufer nutzbar sein.
[<=]

3.5.1 SOAP-Schnittstelle

Für die Beschreibung der SOAP-Schnittstelle zum Clientsystem wird in dieser Spezifikation WSDL Version 1.1 [WSDL1.1] eingesetzt. Die Interoperabilität zwischen verschiedenen SOAP-Implementierungen wird durch die Vorgaben des WS-I Basic Profile erreicht.

A_15601 - SOAP für Web-Services der Basisdienste
Der Konnektor MUSS für die an der Clientsystemschnittstelle definierten Web-Services der Basisdienste [SOAP1.1] verwenden.**[<=]**

TIP1-A_4519 - Web-Services konform zu [BasicProfile1.2]
Der Konnektor MUSS die für die Clientsystemschnittstelle definierten Web-Services konform zu [BasicProfile1.2] anbieten.
Abweichend von R1012 in [BasicProfile1.2] MUSS der Konnektor nur das Character Encoding UTF-8 unterstützen. Andere Kodierungen MUSS der Konnektor mit einem Fehler beantworten.**[<=]**

TIP1-A_4519-01 - ab PTV4: Web-Services konform zu [BasicProfile1.2]
Der Konnektor MUSS die für die Clientsystemschnittstelle definierten Web-Services der Basisdienste konform zu [BasicProfile1.2] anbieten.
[<=]

A_15606 - Character Encoding für Web-Services
Abweichend von R1012 in [BasicProfile1.2] und [BasicProfile2.0] MUSS der Konnektor nur das Character Encoding UTF-8 unterstützen. Andere Kodierungen MUSS der Konnektor mit einem Fehler beantworten.**[<=]**

Da der Konnektor UTF-16 nicht unterstützt, muss das Clientsystem den Request in UTF-8 kodieren. Diese Festlegungen gelten nur für die eigentliche SOAP-Nachricht. Sind in der SOAP-Nachricht base64-encodierte XML-Elemente vorhanden, so können diese XML-Elemente andere Zeichencodierungen aufweisen.

Fachmodule

Fachmodule können für Web-Services, die Clientsystemen bereitgestellt werden, entweder [SOAP1.1] mit [BasicProfile1.2] oder [SOAP1.2] mit [BasicProfile2.0] verwenden. Die genaue Ausprägung erfolgt in der jeweiligen Interfacebeschreibung des Web-Services für das Fachmodul.

A_15607 - SOAP für Web-Services der Fachmodule
Der Konnektor MUSS für die an der Clientsystemschnittstelle definierten Web-Services der Fachmodule [SOAP1.1] und [SOAP1.2] unterstützen. Die SOAP-Version pro Web-Service Endpunkt wird durch die WSDL des jeweiligen Web-Service definiert.**[<=]**

A_15608 - Web-Services der Fachmodule konform zu [BasicProfile1.2]
Der Konnektor MUSS für die an der Clientsystemschnittstelle definierten Web-Services der Fachmodule bei [SOAP1.1] die Profilierung konform zu [BasicProfile1.2] anbieten.**[<=]**

A_15609 - Web-Services der Fachmodule konform zu [BasicProfile2.0]
Der Konnektor MUSS für die an der Clientsystemschnittstelle definierten Web-Services der Fachmodule bei [SOAP1.2] die Profilierung konform zu [BasicProfile2.0] anbieten.**[**<=**]**

3.5.2 Statusrückmeldung und Fehlerbehandlung

Der Konnektor bietet Operationen an der Außenschnittstelle über SOAP-Webservices an. Treten bei der Ausführung einer Operation Fehler auf, so werden diese an das aufrufende System gemeldet. Die von den Basisdiensten des Konnektors angebotenen SOAP-Webservices melden Fehler, die bei der Ausführung einer Operation auftreten, über eine SOAP-Fault-Nachricht (siehe auch [gemSpec_OM#3.2.3]).

TIP1-A_5058 - Fehlerübermittlung durch gematik-SOAP-Fault
Der Konnektor MUSS Fehlermeldungen, die im Rahmen einer über die Außenschnittstelle aufgerufenen Operation auftreten, an das Clientsystem mittels gematik-SOAP-Faults melden.
[<=**]**

TIP1-A_5058-01 - ab PTV4: Fehlerübermittlung durch gematik-SOAP-Fault
Der Konnektor MUSS Fehlermeldungen, die im Rahmen einer über die Außenschnittstelle aufgerufenen Operation eines Basisdienst-SOAP-Webservices auftreten, an das Clientsystem mittels gematik-SOAP-Faults melden.
[<=**]**

Treten bei konnektorinternen Operationen (TUCs) Fehler auf, so werden diese an den Aufrufer (aufrufender TUC oder aufrufende Operation) zurückgegeben. Der Aufrufer kann den aufgetretenen Fehler in seinem Kontext neu interpretieren. Das bedeutet insbesondere, dass ein Error eines aufgerufenen TUCs nicht zwingend zum Abbruch des aufrufenden TUCs bzw. der aufrufenden Operation führen muss. So ist es dem Aufrufer möglich, einen Error als Warnung zu interpretieren und an den eigenen internen oder externen Aufrufer zurückzumelden. Diese dabei erzeugte Fehlerkette wird in Form einer Fehler-Trace-Struktur abgebildet, um eine Nachverfolgung von Fehlern zu ermöglichen.

Operationen an der Außenschnittstelle können die Fehlerkette zu Informationszwecken in der SOAP-Antwort an das Clientsystem senden. Dazu enthält jede SOAP-Antwort das Element Status, dass gemäß dem XML-Schema [ConnectorCommon.xsd] aufgebaut ist (siehe auch Abbildung PIC_KON_107 XML-Struktur des Status-Elements einer SOAP-Antwort).

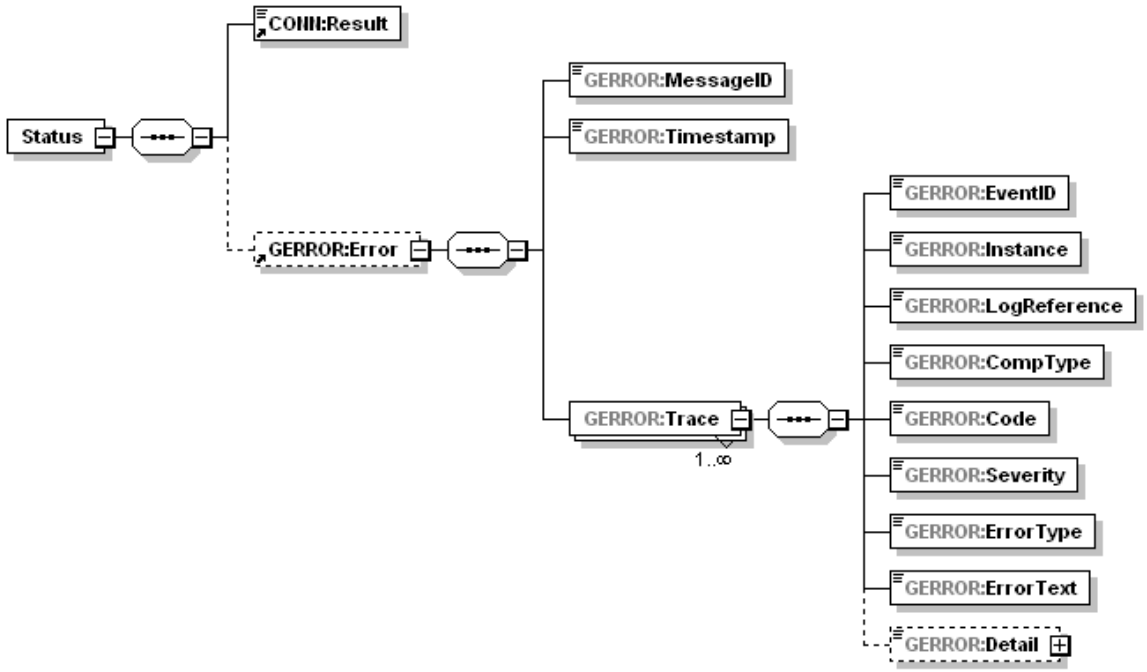


Abbildung 3: PIC_KON_107 XML-Struktur des Status-Elements einer SOAP-Antwort

Schlägt eine Operation fehl, so wird eine SOAP-Fault-Meldung an das Clientsystem versendet. Im Erfolgsfall wird das Status-Element in die Antwortnachricht an das Clientsystem aufgenommen. Ist der Fehler-Trace leer (Element GERROR:Error ist nicht vorhanden), so wird CONN:Result auf OK gesetzt. Andernfalls, d. h. wenn in GERROR:Trace Fehler der Schwere Info oder Warning (zu Informationszwecken) enthalten sind, wird CONN:Result auf Warning gesetzt.

TIP1-A_4521 - Protokollierung von Fehlern inkl. Trace-Struktur
Der Konnektor MUSS Fehler protokollieren, die in fachlichen und technischen Abläufen von der gematik spezifiziert oder herstellerspezifisch definiert sind und den Schweregrad (Severity) Warning, Error oder Fatal haben. Zur Nachvollziehbarkeit des Fehlers MÜSSEN Fehlerursache, fachliche und technische Auslöser des Fehlverhaltens aus den Protokolleinträgen erkennbar sein.
[<=]

A_14159 - Rückgabe von Fehlermeldungen an der Außenschnittstelle
Der Konnektor MUSS bei der Rückgabe von Fehlermeldungen an der Außenschnittstelle sicherstellen, dass im letzten" GERROR:Trace"-Element der GERROR-Struktur ein von der gematik spezifizierter Fehler steht. Die GERROR-Struktur kann weitere gematik- und herstellerspezifische Fehler enthalten.
[<=]

In der Regel ist es ausreichend, wenn die GERROR-Struktur an der Außenschnittstelle nur ein Element „GERROR:Trace" mit einem gematik-Fehler enthält.

Wenn für eine Fehlersituation kein Fehlercode spezifiziert ist, kann ein herstellerspezifischer Fehler zur Detaillierung verwendet werden. In diesem Fall muss ein passender gematik-Fehler als letztes GERROR:Trace-Element gewählt werden. Bei Fehlern in technischen Abläufen kann Fehlercode 4001 als letztes GERROR:Trace-Element verwendet werden. Die Wahl des letzten GERROR:Trace-Elements ist mit der gematik abzustimmen.

Zur Struktur von Fehlermeldungen siehe auch [gemSpec_OM#GS-A_3856].

3.5.3 Transport großer Dokumente

SOAP Message Transmission Optimization Mechanism (MTOM) ermöglicht den direkten Transport von binären Daten in Webservices, d.h. ohne dass eine zur Laufzeit aufwändige Verpackung der binären Daten in ein Base64-XML-Element notwendig wird. Auf die Definition der Webservices und ihre Funktionalität hat dieser Optimierungsmechanismus keinen Einfluss. Der Einsatz von MTOM dient der Verbesserung des Performance-Verhaltens für große Dokumente.

Das Clientsystem kann die Optimierung des Transports großer Dokumente per SOAP Message Transmission Optimization Mechanism (MTOM) anstoßen. In den WSDL-Dateien werden keine MTOM Serialization Policy Assertion [WS-MTOMPolicy] eingebettet.

TIP1-A_5694 - SOAP Message Transmission Optimization Mechanism für Basisdienste
Der Konnektor KANN SOAP Message Transmission Optimization Mechanism (MTOM) gemäß [MTOM] unterstützen.
Wenn der Konnektor MTOM unterstützt, MUSS er MTOM für Signatur- und Verschlüsselungsdienst unterstützen, DARF aber NICHT MTOM für andere Dienste unterstützen.
Wenn der Konnektor MTOM unterstützt, MUSS er, vergleichbar dem Einsatz des Attributs wsp:Optional="true" einer MTOM Serialization Policy Assertion [WS-MTOMPolicy], genau dann MTOM für die Antwortnachricht verwenden, wenn entweder

- die Aufrufnachricht eine `application/xop+xml` Nachricht ist
- oder der `Accept` HTTP header der Aufrufnachricht folgenden Wert hat: `multipart/related; type=application/xop+xml`

[<=]

TIP1-A_5694-02 - ab PTV4: SOAP Message Transmission Optimization Mechanism für Basisdienste
Der Konnektor KANN SOAP Message Transmission Optimization Mechanism (MTOM) gemäß [MTOM-SOAP1.1] für Basisdienste unterstützen.**[<=]**

TIP1-A_5694-03 - ab PTV5: SOAP Message Transmission Optimization Mechanism für Basisdienste
Der Konnektor MUSS SOAP Message Transmission Optimization Mechanism (MTOM) gemäß [MTOM-SOAP1.1] für Basisdienste unterstützen.
[<=]

A_15786 - SOAP Message Transmission Optimization Mechanism für Basisdienste - Einschränkung
Wenn der Konnektor MTOM für Basisdienste unterstützt, MUSS er MTOM für Signatur- und Verschlüsselungsdienst unterstützen, DARF aber NICHT MTOM für andere Dienste unterstützen.**[<=]**

A_15610 - Verwendung von MTOM für Antwortnachricht
Wenn der Konnektor MTOM unterstützt, MUSS er, vergleichbar dem Einsatz des Attributs wsp:Optional="true" einer MTOM Serialization Policy Assertion [WS-MTOMPolicy], genau dann MTOM für die Antwortnachricht verwenden, wenn entweder

- die Aufrufnachricht eine application/xop+xml Nachricht ist
- oder der Accept HTTP header der Aufrufnachricht folgenden Wert hat: `multipart/related; type=application/xop+xml`.

[<=]

A_15611 - SOAP Message Transmission Optimization Mechanism für Fachmodule
Der Konnektor MUSS SOAP Message Transmission Optimization Mechanism (MTOM) gemäß [MTOM] für Fachmodule unterstützen, wenn es in der Schnittstellenbeschreibung des Fachmodules explizit gefordert wird.**[<=]**

3.6 Verwendung manuell importierter CA-Zertifikate

TI-fremde X.509-Zertifikate werden im Rahmen des Verschlüsselungsdienstes verwendet. Um den Vertrauensraum für diese Zertifikate abzubilden, erlaubt der Konnektor, X.509-CA-Zertifikate zu diesen TI-fremden X.509-Zertifikaten in eine interne Liste (CERT_IMPORTED_CA_LIST) zu importieren.

Der Konnektor kann dann im Rahmen der Hybridverschlüsselung den symmetrischen Schlüssel empfängerspezifisch mit dessem TI-fremden X.509-Zertifikat verschlüsseln. Die TI-fremden Zertifikate dürfen nicht zu einem anderen Zweck als diesem eingesetzt werden.

TIP1-A_5433 - Manuell importierte X.509-CA-Zertifikate nur für hybride Verschlüsselung
Der Konnektor DARF End-Entity-Zertifikate, die lediglich gegen manuell importierte X.509-CA-Zertifikate geprüft werden, die von CAs außerhalb der TI stammen (CERT_IMPORTED_CA_LIST), NICHT für andere Zwecke als zur hybriden Verschlüsselung von Dokumenten verwenden.
[<=]

Die Berücksichtigung der CA-Zertifikate aus CERT_IMPORTED_CA_LIST muss auf folgende Anwendungsfälle beschränkt werden:

1. Prüfung eines Zertifikates im Rahmen der hybriden Verschlüsselung
2. Prüfung eines Zertifikates im Rahmen eines Aufrufes der Operation "VerifyCertificate"

TIP1-A_5660 - Hinweise im Handbuch für manuell importierte X.509-CA-Zertifikate
Das Handbuch des Konnektors MUSS sinngemäß folgende Hinweise enthalten:

- Der Administrator übernimmt die Verantwortung für die Verlässlichkeit der importierten CA-Zertifikate.
- Der Administrator kann sich bei seiner Entscheidung für einen Import von CA-Zertifikaten auf die Informationen der gematik stützen.
- Die gematik veröffentlicht dazu Informationen über CA-Betreiber, welche die Erfüllung der Sicherheitsanforderungen der gematik nachgewiesen haben.

[<=]

3.7 Testunterstützung

Gemäß Testkonzept Online-Rollout (Stufe 1) [gemKPT_Test_ORS1#TIP1-A_2839] muss ein Hersteller eines Konnektors seine Modelle in drei Ausführungen vorsehen: Eine für die Testumgebung, eine für die Referenzumgebung und eine für die Produktivumgebung.

Damit trotz dieser Forderung die Firmware je Konnektorversion für alle Umgebungen identisch ist, wird die Erkennung der Umgebung an die gSMC-K gebunden. Die Konnektor-Firmware muss zwischen den Umgebungen PU und RU/TU unterscheiden. Die gSMC-K besitzt hierzu den Datencontainer MF/EF.EnvironmentSettings, der die jeweilige Umgebungskennung enthält (PU bzw. TU/RU). Die Umgebungskennung wird read-only auf der gSMC-K gespeichert.

TIP1-A_4981 - Steuerung der Betriebsumgebung via gSMC-K
Der Produkttyp Konnektor MUSS sowohl in der Testumgebung (TU), der Referenzumgebung (RU) wie auch der Produktivumgebung (PU) betreibbar sein.
Die Information, ob eine Konnektorinstanz in der TU/RU oder PU betrieben wird, MUSS der Konnektor dem File MF/EF.EnvironmentSettings der gSMC-K entnehmen.
Abhängig von der ermittelten Betriebsumgebung MÜSSEN die Konfigurationswerte gemäß Tabelle TAB_KON_812 verwendet werden.

Tabelle 12: TAB_KON_812 Umgebungsabhängige Konfigurationsparameter

Betriebs umgebung	Konfigurations parameter	Konfigurations wert	Beschreibung
PU	NET_TI_ ZENTRAL	siehe [gemSpec_Net#Tab_ Adrkonzept_Produktiv]	Siehe TAB_KON_680. Dieser Wert MUSS für den Administrator über die Managementschnittstelle einsehbar, DARF aber NICHT änderbar sein.
	NET_TI_ GESICHERTE_FD	siehe [gemSpec_Net#Tab_ Adrkonzept_Produktiv]	Siehe TAB_KON_680. Dieser Wert MUSS für den Administrator über die Managementschnittstelle einsehbar, DARF aber NICHT änderbar sein.
	NET_TI_ OFFENE_FD	siehe [gemSpec_Net#Tab_ Adrkonzept_Produktiv]	Siehe TAB_KON_680. Dieser Wert MUSS für den Administrator über die Managementschnittstelle einsehbar, DARF aber NICHT änderbar sein.
	DNS_TOP_ LEVEL_DOMAIN_TI	telematik.	Siehe TAB_KON_731. Dieser Wert MUSS für den Administrator über die Managementschnittstelle einsehbar, DARF aber NICHT änderbar sein.
RU/TU	NET_TI_ ZENTRAL	siehe [gemSpec_Net# Tab_Adrkonzept_Test]	Siehe TAB_KON_680. Dieser Wert MUSS für den Administrator über die Managementschnittstelle mit dem Konfigurationswert voreingestellt und änderbar sein.

	NET_TI_ GESICHERTE_FD	siehe [gemSpec_Net# Tab_Adrkonzept_Test]	Siehe TAB_KON_680. Dieser Wert MUSS für den Administrator über die Managementschnittstelle mit dem Konfigurationswert voreingestellt und änderbar sein.
	NET_TI_ OFFENE_FD	siehe [gemSpec_Net# Tab_Adrkonzept_Test]	Siehe TAB_KON_680. Dieser Wert MUSS für den Administrator über die Managementschnittstelle mit dem Konfigurationswert voreingestellt und änderbar sein.
	DNS_TOP_ LEVEL_ DOMAIN_TI	telematik-test.	Siehe TAB_KON_731. Dieser Wert MUSS für den Administrator über die Managementschnittstelle einsehbar, aber nicht änderbar sein.

[<=]

TIP1-A_4707 - Betrieb in Test- und Referenzumgebung
Der Produkttyp Konnektor MUSS auch in der Test- und Referenzumgebung betrieben werden können. Dafür MUSS der Vertrauensanker des Konnektors für diese Umgebung ausgetauscht werden können. Dies KANN durch Lieferung eines neuen Konnektors oder durch Austausch der gSMC-K durch den Hersteller ermöglicht werden. Der Hersteller MUSS sicherstellen, dass Konnektoren ausschließlich mit den zu ihrer Einsatzumgebung gehörenden Vertrauensankern ausgestattet werden.
[<=]

TIP1-A_4982 - Anzeige von TU/RU in der Managementschnittstelle
Die Administrationsoberfläche MUSS, wenn der Konnektor in der Testumgebung (TU) oder Referenzumgebung (RU) betrieben wird, die Umgebungsbezeichnung zu jeder Zeit erkennbar in der Managementschnittstelle anzeigen.
Die Anzeige eines Betriebs in der Produktivumgebung DARF NICHT explizit angezeigt werden.
[<=]

4 Funktionsmerkmale

Für die folgenden Inhalte bitte die Hinweise in Kapitel 1.5.3 „Erläuterungen zur Spezifikation des Außenverhaltens„ sowie Kapitel 1.5.4 Erläuterungen zur Dokumentenstruktur und „Dokumentenmechanismen“ beachten.

4.1 Anwendungskonnektor

4.1.1 Zugriffsberechtigungsdienst

Der Zugriffsberechtigungsdienst ist ein interner Dienst. Er ermöglicht es Operationen eine Prüfung auf Zugriffsberechtigung für die von ihnen benötigten Ressourcen durchzuführen. Die Prüfung erfolgt direkt nach Aufruf einer Operation des Konnektors durch das Clientsystem und basiert auf den im Clientaufruf enthaltenen Parametern.

Der Zugriffsberechtigungsdienst definiert über ein Informationsmodell die erlaubten Zugriffsmöglichkeiten. Um dies zu erreichen, modelliert es Mandanten und ordnet ihnen Clientsysteme sowie die vom Konnektor verwalteten externen Ressourcen (Kartenterminal mit Slots, Arbeitsplatz mit Signaturproxy und SMC-Bs) zu. Diese durch einen Administrator persistent zu modellierenden Entitäten und Beziehungen beinhalten die erlaubten Zugriffswege vom Clientsystem über Arbeitsplatz zum Kartenterminal und dessen Slots. Sie werden im Konnektor administrativ konfiguriert. Der Signaturproxy hat keine eigene Identität im Informationsmodell, da er den Kontext des aufrufenden Clientsystems verwendet.

Neben diesen persistenten Entitäten und Beziehungen bildet das Modell auch die in den Slots temporär gesteckten Karten und die zugehörigen Kartensitzungen als transiente Entitäten und Beziehungen ab.

Abbildung PIC_Kon_100 stellt das Informationsmodell dar. Die persistenten Entitäten haben einen grünen Hintergrund, die transienten einen weißen.

Tabelle TAB_KON_507 beschreibt die Entitäten und legt ihren Identitätsschlüssel fest. Tabelle TAB_KON_508 beschreibt die Attribute. Tabelle TAB_KON_509 beschreibt die Entitätsbeziehungen und referenziert dabei die in Abbildung PIC_Kon_100 durch Zahlen in eckigen Klammern markierten Beziehungen. Tabelle TAB_KON_510 definiert Constraints, die zusätzlich zu den in Abbildung PIC_Kon_100 definierten Kardinalitäten gelten. Die Constraints werden mittels Object Constraint Language (OCL) definiert.

4.1.1.1 Funktionsmerkmalweite Aspekte

TIP1-A_4522 - Zugriffsberechtigungs-Informationsmodell des Konnektors
Der Konnektor MUSS die Entitäten, Attribute und Beziehungen des Informationsmodells intern vorhalten, dabei für die Einhaltung der definierten Constraints sorgen und die persistenten Entitäten und Beziehungen dauerhaft speichern. Der Konnektor MUSS dabei eine Mindestanzahl von 999 Mandanten unterstützen.
Das Informationsmodell ist definiert durch das UML-Diagramm „PIC_Kon_100 Informationsmodell des Konnektors„ und die Tabelle „TAB_KON_510 Informationsmodell Constraints“. Der Konnektor darf nur Daten in sein Informationsmodell übernehmen, die alle Eigenschaften des Informationsmodells, insbesondere die Constraints, erfüllen.
Die Entitäten werden in Tabelle „TAB_KON_507 Informationsmodell Entitäten“ beschrieben, die Attribute in Tabelle „TAB_KON_508 Informationsmodell Attribute“ und

die Beziehungen in Tabelle „TAB_KON_509 Informationsmodell Entitätenbeziehungen“.
[<=]

Hinweis zu den Bezeichnern der Entitäten und ihrer Attribute: Im Folgenden beginnen Entitäten mit einem Großbuchstaben, Attribute mit einem Kleinbuchstaben. Werden die Entitäten und Attribute in XML-Dokumenten verwendet, so beginnen die zugeordneten XML-Elementbezeichner grundsätzlich mit einem Großbuchstaben und verwenden den englischen Begriff, der im Folgenden in Klammern angegeben ist, wenn zur besseren Lesbarkeit im Modell ein deutscher Begriff verwendet wird.

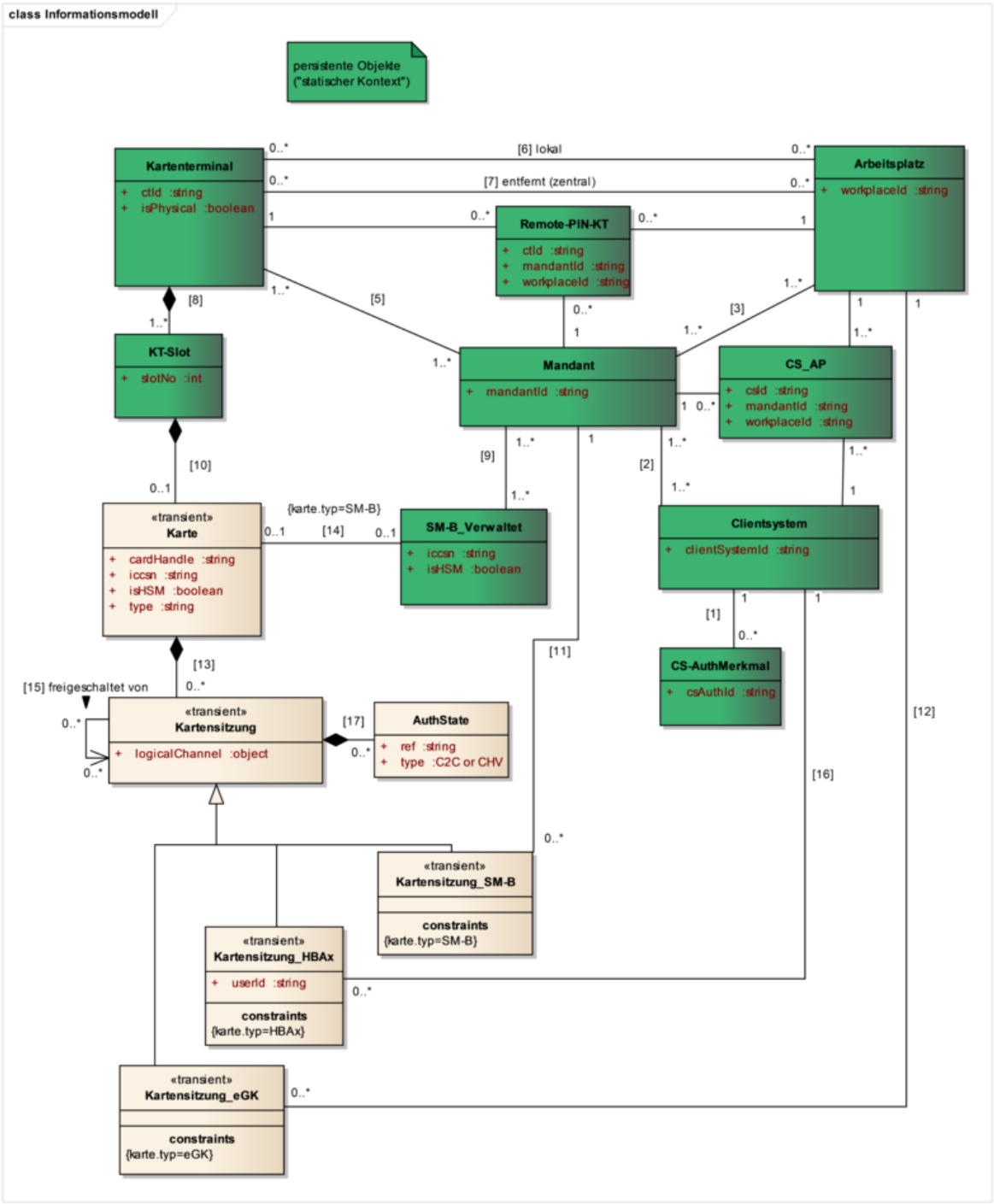


Abbildung 4: PIC_Kon_100 Informationsmodell des Konnektors

Tabelle 13: TAB_KON_507 Informationsmodell Entitäten

Entität	persistent/ transient	Identitätsschlüssel	Beschreibung

Mandant	persistent	mandantId	Zu Mandanten und Mandantenfähigkeit siehe Kapitel Mandantenfähigkeit.
Clientsystem	persistent	clientSystemId	Unter einem Clientsystem wird hier ein einzelnes oder eine Gruppe von Systemen verstanden, welche im LAN der Einsatzumgebung auf die Clientsystem-Schnittstelle des Konnektors zugreifen.
CS-AuthMerkmal (CS-AuthProperty)	persistent	csAuthId	Das Authentifizierungsmerkmal dient der Authentifizierung, wenn sich das Clientsystem gegenüber dem Konnektor authentisiert. Der Identitätsschlüssel csAuthId wird bei der Administration vergeben
Arbeitsplatz (Workplace)	persistent	workplaceId	alle dem Konnektor bekannten Arbeitsplätze
Kartenterminal (CardTerminal)	persistent	ctId	alle dem Konnektor bekannten Kartenterminals.
KT-Slot (CT-Slot)	persistent	ctId, slotNo	Die sich in den Kartenterminals befindenden Chipkartenslots (Functional Unit Type 00)
Karte (Card)	transient	cardHandle oder iccsn	Die in den Kartenterminals steckenden Smartcards des Gesundheitswesens, die persönliche Identitäten oder Rollen repräsentieren (eGK, HBA, SMC-B). Karten, die nur Geräteidentitäten tragen (gSMC-K, gSMC-KT) werden in diesem Modell nicht betrachtet. Karten im Sinne dieses Informationsmodells existieren maximal so lange, wie sie im Kartenterminal stecken. Die aktuell im System steckenden Karten werden

			vom Clientsystem über das cardHandle adressiert. Die iccsn erlaubt eine dauerhafte Adressierung einer Karte. Für den Kartentyp „SM-B" kann hier auch eine in einem HSM-B enthaltene virtuelle SMC-B abgebildet werden.
Kartensitzung (CardSession)	transient	siehe konkrete Kartensitzungen	Kartensitzungen stellen ein wesentliches Konzept im Sicherheitsmodell des Konnektors dar. Eine Kartensitzung verwaltet einen aktuellen logischen Sicherheitsstatus einer Karte. Die Kartensitzungen sind einer Karte fest zugewiesen. Zu einer Karte kann es mehrere Kartensitzungen geben, die voneinander logisch unabhängige Sicherheitsstatus einer Karte verwalten. Der Konnektor führt alle Zugriffe auf eine Karte im Kontext einer Kartensitzung zu dieser Karte aus. Das Attribut logischerKanal bezeichnet den logischen Kanal zur Karte, der im Rahmen der Kartensitzung verwendet wird (gemäß Standard [7816–4]).
Kartensitzung_eGK (CardSession_eGK)	transient	cardHandle	Kartensitzung für eine eGK. Die KVK ist im Modell nicht explizit dargestellt. Soweit anwendbar, gelten für die KVK die gleichen Aussagen wie für die eGK.
Kartensitzung_SM-B (CardSession_SM-B)	transient	cardHandle, mandantId	Kartensitzung für eine SM-B

Kartensitzung_HBAx (CardSession_HBAx)	transient	cardHandle, clientSystemId, userId	Kartensitzung für einen HBAx. Unter dem Typ „HBAx" sind auch die Vorläuferkarten wie „HBA-qSig" und „ZOD_2.0" inkludiert.
SM-B_Verwaltet (SM-B_managed)	persistent	iccsn	SM-Bs müssen im Gegensatz zu den übrigen Karten im Konnektor vor ihrer Verwendung persistent im Informationsmodell als „SM-B_Verwaltet" per Administration aufgenommen werden. Dies gilt auch für die in einem HSM-B enthaltenen virtuellen SMC-Bs.
CS_AP	persistent	mandantId, clientSystemId, workplaceId	CS_AP legt die von einem Clientsystem pro Mandanten nutzbaren Arbeitsplätze fest. Ein Clientsystem kann dabei mehrere Arbeitsplätze bedienen. Ebenso können Arbeitsplätze von mehreren Clientsystemen, auch gleichzeitig, genutzt werden, z. B. bei zwei unterschiedlichen, voneinander unabhängigen Praxisprogrammen.
Remote-PIN-KT	persistent	mandantId, workplaceId, ctId	Remote-PIN-KT legt pro Mandant und Arbeitsplatz fest, über welches Kartenterminal eine Remote PIN-Eingabe erfolgen soll, wenn an diesem Arbeitsplatz die PIN-Eingabe für eine Karte erforderlich ist, die nicht in einem dem Arbeitsplatz lokal zugeordneten Kartenterminal steckt.
AuthState	transient	cardHandle, (clientSystemId), (userId), ref	Zu einer Kartensitzung gibt es höhere AuthorizationStates, die durch (type =C2C) Freischaltung oder durch PIN-Eingabe (type=CHV) erreicht werden können.

			Für eGK G2.0 wird der Zustand der MRPINs nicht in AuthState gespeichert.

Tabelle 14: TAB_KON_508 Informationsmodell Attribute

Attribut	Beschreibung
cardHandle	Das Identifikationsmerkmal einer Karte für die Dauer eines Steckzyklusses. Es wird mit dem Entfernen der Karte aus dem Kartenterminal ungültig. Es wird automatisch vom Konnektor vergeben.
clientSystemId	Das Identifikationsmerkmal eines Clientsystems. Es wird per Administration dem Mandanten im Clientsystem und im Konnektor zugeordnet.
csAuthId	Das Identifikationsmerkmal eines Authentifizierungsmerkmals.
ctId	Das Identifikationsmerkmal eines Terminals. Es ist eine fixe Eigenschaft des Kartenterminals.
iccsn	Die Seriennummer einer Karte. Sie identifiziert eine Karte dauerhaft.
isHSM	Attribut der Entitäten Karte und SM-B_Verwaltet. Es ist false, wenn eine echte Smardcard abgebildet wird und true, wenn es sich um eine virtuelle SMC-B handelt, die in einem HSM-B enthalten ist.
isPhysical	Attribut des Kartenterminals das den Wert „Ja" hat, wenn es sich um ein tatsächlich existierendes Kartenterminal handelt. Ist der Wert „Nein", dann handelt es sich um ein logisches Kartenterminal im Zusammenhang mit einem HSM-B.
logicalChannel	Referenz auf ein Objekt, das einen logischen Kanal repräsentiert.
mandantId	Das Identifikationsmerkmal eines Mandanten. Es wird per Administration dem Mandanten im Clientsystem und im Konnektor zugeordnet.
ref	Das Identifikationsmerkmal eines AuthState zu einer gegebenen Kartensitzung. Im Falle C2C handelt es sich um die

	KeyRef (mit einer bestimmten Rolle) und in Falle CHV um eine referenzierte PIN.
slotNo	Das Identifikationsmerkmal eines Slot für ein bestimmtes Kartenterminal. Diese fortlaufende Nummer ist eine fixe Eigenschaft des Kartenterminals. Sie beginnt bei 1.
type	Als Kartenattribut: Typ einer Karte. Im Folgenden berücksichtigte Werte: „HBAx", „SM-B", „EGK". Als Attribute eines AuthState: Typ des AuthState. „C2C" steht für gegenseitige Kartenauthentisierung. „CHV" steht für Card Holder Verification per PIN-Eingabe.
userId	Das Identifikationsmerkmal des Nutzers im Clientsystem (Die userId wird durch das Clientsystem vergeben und verwaltet). Die userId wird im Kontext eine Kartensitzung_HBAx vom Konnektor verwendet, um als Bestandteil des Identitätsschlüssels die Kartensitzung_HBAx zu identifizieren.
workplaceId	Das Identifikationsmerkmal eines Arbeitsplatzes. Es wird per Administration dem Mandanten im Clientsystem und im Konnektor zugeordnet.

Tabelle 15: TAB_KON_509 Informationsmodell Entitätenbeziehungen

Entitätenbeziehung	persistent/ transient	Beschreibung
Authentifikationsmerkmale des Clientsystems [1]	persistent	Diese Relation legt für jedes Clientsystem eine Menge von Authentisierungsmerkmalen fest. Mit einem dieser Authentisierungsmerkmale muss sich ein Client gegenüber dem Konnektor authentisiert haben, um als das entsprechende Clientsystem vom Konnektor akzeptiert zu werden.
Clientsysteme des Mandanten [2]	persistent	Diese Relation weist Clientsystemen Mandanten zu.
Arbeitsplätze des Mandanten [3]	persistent	Diese Relation weist Arbeitsplätze Mandanten zu. Arbeitsplätze können von mehreren Mandanten genutzt werden. Z. B. kann ein von mehreren Mandanten genutzter gemeinsamer Empfang als ein Arbeitsplatz modelliert werden.
Kartenterminals des Mandanten [5]	persistent	Diese Relation weist Kartenterminals Mandanten zu.

Lokale Kartenterminals [6]	persistent	Diese Relation erfasst die Kartenterminals, die sich lokal an einem Arbeitsplatz befinden und von diesem genutzt werden können. Die Modellierung lässt es zu, dass Kartenterminals mehreren Arbeitsplätzen lokal zugewiesen werden. Jeder an der TI teilnehmende Arbeitsplatz wird in der Regel mindestens ein lokales Kartenterminal benötigen.
Entfernte Kartenterminals [7]	persistent	Diese Relation beschreibt, auf welche Kartenterminals Arbeitsplätze (remote) zugreifen dürfen. Dies ist für zentral steckende Karten vorgesehen.
Slot eines Kartenterminals [8]	persistent	Die Zuordnung von Slots zu einem Kartenterminal ergibt sich automatisch aus den Eigenschaften des Kartenterminals.
SM-B_Verwaltet eines Mandanten [9]	persistent	Diese Relation legt fest, welche verwalteten SM-Bs einem Mandanten zugeordnet sind.
Kartenterminal-Slot, in dem eine Karte steckt [10]	transient	Sobald eine Karte in ein Kartenterminal gesteckt wird, ergibt sich implizit eine Relation der Karte zu dem Slot, in dem sie steckt, [6] und indirekt über [4] zum Kartenterminal.
Mandant der Kartensitzung SM-B [11]	transient	Beim Anlegen einer Kartensitzung SM-B wird diese immer dem zugreifenden Mandanten zugeordnet.
Arbeitsplatz der Kartensitzung eGK [12]	transient	Eine Kartensitzung eGK ist immer einem Arbeitsplatz zugeordnet.
Karte einer Kartensitzung [13]	transient	Jeder Kartensitzung ist genau einer Karte zugeordnet.
Gesteckte SM-B [14]	transient	Wird eine SM-B gesteckt und handelt es sich um eine verwaltete SM-B, ergibt sich über die iccsn die Zuordnung.
Freischaltung einer Karte [15]	transient	Diese Relation erfasst die Freischaltung einer Karte durch eine andere Karte.
Bindung der Kartensitzung_HBAx an Clientsystem [16]	transient	Kartensitzungen HBAx sind einem Clientsystem zugeordnet.
AuthState pro Kartensitzung [17]	transient	Eine Kartensitzung kann erhöhte Sicherheitszustände (Authorization State) haben.

Tabelle 16: TAB_KON_510 Informationsmodell Constraints

#	Beschreibung	Definition mittels OCL (Die Constraints werden im UML ergänzenden Standard OCL definiert.)

C1	Eine eGK muss eine oder keine Kartensitzung haben.	`context Karte`
		`inv: self.type = "eGK" implies`
		`self.kartensitzung.size() <= 1`

C2	Wenn zwei Kartensitzungen einer HBAx dem gleichen Clientsystem zugeordnet sind und ihre userIds gleich sind, dann müssen die beiden Kartensitzungen identisch sein.	`context Kartensitzung-HBAx`
		`inv: forAll(k1, k2 : Kartensitzung-HBAx |`
		`k1.karte = k2.karte`
		`and k1.clientsystem = k2.clientsystem`
		`and k1.userId = k2.userId`
		`implies`
		`k1 = k2)`

C3	Wenn zwei SM-B-Kartensitzungen einer Karte dem gleichen Mandanten zugeordnet sind, dann müssen die beiden Kartensitzungen identisch sein.	`context Kartensitzung-SM-B`
		`inv: forAll(k1, k2 : Kartensitzung-SM-B |`
		`k1.karte = k2.karte`
		`and k1.mandant = k2.mandant implies`
		`k1 = k2)`

C4	Die Seriennummer iccsn einer Karte muss eindeutig sein.	`context Karte`
		`inv: Karte.allInstances ->`
		`isUnique(iccsn)`

C5	Die Seriennummer iccsn einer Karte muss für die vom Konnektor verwalteten SM-Bs eindeutig sein.	`context SM-B_Verwaltet`
		`inv: SM-B_Verwaltet.allInstances ->`
		`isUnique(iccsn)`

C6	Das CardHandle einer Karte muss eindeutig sein.	`context Karte`
		`inv: Karte.allInstances ->`
		`isUnique(cardHandle)`

C7	Die Identifikationsnummer des Clientsystems muss eindeutig sein.	`context Clientsystem`
		`inv: Clientsystem.allInstances ->`
		`isUnique(clientSystemId)`

C8	Die Identifikationsnummer des Mandanten muss eindeutig sein.	**context** Mandant **inv:** Mandant.allInstances -> isUnique(mandantId)
C9	Die Identifikationsnummer des Arbeitsplatzes muss eindeutig sein.	**context** Arbeitsplatz **inv:** Arbeitsplatz.allInstances -> isUnique(workplaceId)
C10	Die Identifikationsnummer des Kartenterminals muss eindeutig sein.	**context** Kartenterminal **inv:** Kartenterminal.allInstances -> isUnique(ctId)
C11	Die Identifikationsnummer (slotNo) des Kartenterminal-Slots für ein gegebenes Kartenterminal muss eindeutig sein.	**context** Kartenterminal **inv:** self.kT-Slot -> isUnique(slotNo)
C12	Es muss gewährleistet sein, dass nur Arbeitsplätze und Clientsysteme einander im Rahmen eines Mandanten zugeordnet werden, die diesem Mandanten selbst zugeordnet sind.	**context** CS-AP **inv:** self.arbeitsplatz.mandant.includes(self.mandant) **inv:** self.clientsystem.mandant.includes(self.mandant)
C13	Es muss gewährleistet sein, dass nur Kartenterminals und Arbeitsplätze einander im Rahmen eines Mandanten zur Remote-PIN-Eingabe zugeordnet werden, die diesem Mandanten selbst zugeordnet sind.	**context** Remote-PIN-KT **inv:** self.arbeitsplatz.mandant.includes(self.mandant) **inv:** self.kartenterminal.mandant.includes(self.mandant)

C14	Zur Remote-PIN-Eingabe muss ein lokales Kartenterminal ausgewählt sein.	`context Remote-PIN-KT`
		`inv: self.arbeitsplatz`
		`.localKartenterminal`
		`.includes(self.kartenterminal)`
		`inv: not self.arbeitsplatz`
		`.entferntKartenterminal`
		`.includes(self.kartenterminal)`

C15	Zur Remote-PIN-Eingabe darf pro Mandanten und Arbeitsplatz nicht mehr als ein Kartenterminal ausgewählt werden.	`context Remote-PIN-KT`
		`inv: forAll(r1, r2 : Remote-PIN-KT |`
		`r1.arbeitsplatz = r2.arbeitsplatz`
		`and r1.mandant = r2.mandant implies`
		`r1 = r2)`

C16	Eine Kartensitzung-HBAx muss immer eine zugehörige userId haben.	`context Kartensitzung-HBAx` `inv: self.userId <> null`

Hinweis zur Remote-PIN-Eingabe: Constraints C14 und C15 legen fest, dass auch im Fall mehrerer lokaler Kartenterminals an einem Arbeitsplatz nur eines (oder keines) dieser Kartenterminals pro Mandant für die Remote-PIN-Eingabe im Informationsmodell konfiguriert wird.

TIP1-A_4523 - Sicherung der Aktualität des Informationsmodells Zugriffsberechtigungsdienst
Der Konnektor MUSS seine Entscheidungen zur Zugriffsberechtigung basierend auf den aktuellen, realen statischen wie transienten Entitäten und Beziehungen des Informationsmodells treffen. Veränderungen an der statischen Definition (durch den Administrator), sowie Veränderungen an den Entitäten (Änderung der Verfügbarkeit und Zustandsänderung von Karten, Kartenterminals und Clientsystemen) MÜSSEN bei Zugriffsanfragen unmittelbare Auswirkung auf die Entscheidung des Zugriffsberechtigungsdienstes zur Folge haben.
[<=]

4.1.1.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.1.1.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.1.1.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.1.4.1 TUC_KON_000 „Prüfe Zugriffsberechtigung"

Vor Ausführung jeder Operation an der Außenschnittstelle muss der Konnektor prüfen, ob die Operation ausgeführt werden darf (Autorisierung). Diese Prüfung auf Zugriffsberechtigung wird in TUC_KON_000 „Prüfe Zugriffsberechtigung" gekapselt.

TUC_KON_000 „Prüfe Zugriffsberechtigung“ hat als Aufrufparameter den Aufrufkontext der Operation (siehe Abbildung PIC_KON_101), optional das cardHandle einer Karte, optional eine Kartenterminal-ID ctId und optional die Steuerungsparameter „needCardSession“ sowie „allWorkplaces“. Über den Steuerungsparameter „needCardSession“ wird festgelegt, ob zu den CardHandles im Rahmen der Operationsausführung eine Kartensitzung benötigt wird. Über den Steuerungsparameter „allWorkplaces“. wird festgelegt, ob die Auswertung im Rahmen der Operation arbeitsplatzübergreifend für alle vom Mandanten für das angegebene Clientsystem erreichbaren Kartenterminals erfolgen soll.

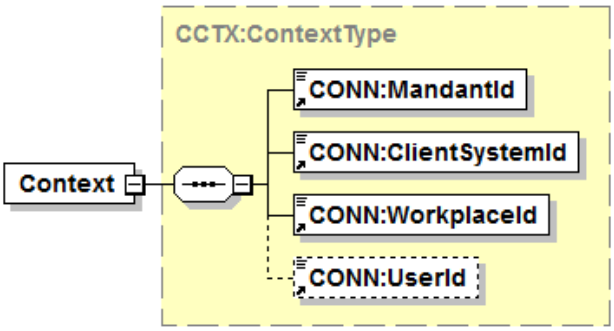


Abbildung 5: PIC_KON_101 Aufrufkontext der Operation

TIP1-A_4524-02 - TUC_KON_000 „Prüfe Zugriffsberechtigung”
Der Konnektor MUSS den technischen Use Case TUC_KON_000 „Prüfe Zugriffsberechtigung“ umsetzen.

Tabelle 17: TAB_KON_511 – TUC_KON_000 „Prüfe Zugriffsberechtigung“

Element	Beschreibung
Name	TUC_KON_000 ”Prüfe Zugriffsberechtigung”
Beschreibung	Es wird geprüft, ob eine Autorisierung im Rahmen der angegebenen Eingangsdaten erteilt wird. Die Autorisierung wurde erteilt, wenn der TUC erfolgreich durchlaufen wurde (kein Abbruch durch Fehlermeldung).“
Eingangs-anforderungen	keine
Auslöser und Vorbedingungen	Aufruf einer Operation des Konnektors durch das Clientsystem.

Eingangsdaten	• mandantId • clientSystemId • workplaceId • userId - *optional* • ctId - *optional* (Kartenterminalidentifikator) • cardHandle - *optional* • needCardSession [Boolean] – *optional; default: true* *(„*needCardSession"=true; „doNotNeedCardSession"=false*)* Dieser Schalter gibt an, ob eine Kartensitzung benötigt wird - true, der aufrufende TUC verwendet eine Kartensitzung - false, der aufrufende TUC verwendet keine Kartensitzung Die Berechtigungsprüfung geht im Default-Fall, davon aus, dass eine Kartensitzung benötigt wird, und prüft für diesen Fall die Berechtigung mit. • allWorkplaces [Boolean] – *optional; default: false* Dieser Schalter gibt an, ob eine mandantenweite Zugriffsberechtigung gemeint ist. Dieser Parameter muss dann (true) gesetzt werden, wenn die Berechtigungsprüfung nicht auf die vom angegebenen Arbeitsplatz erreichbaren Kartenterminals beschränkt ist, sondern sich auf alle vom Clientsystem (clientSystemId) und dem Mandant (mandantId) insgesamt erreichbaren Kartenterminals beziehen soll. Ist dieser Schalter gleich true, wird die Berechtigung unabhängig vom Eingangsparameter workplaceId geprüft.
Komponenten	Konnektor
Ausgangsdaten	• keine
Nachbedingungen	• Autorisierung erteilt

Standardablauf	1. Prüfe, ob die Pflichtparameter (mandantId, clientSystemId, workplaceId) vollständig gesetzt sind. 2. Falls ANCL_CAUT_MANDATORY = Enabled, dann prüfe, ob die gemäß [TIP1-A_4516] durchgeführte Authentifizierung über ein dem Clientsystem zugeordnetes CS-AuthMerkmal erfolgte. 3. Ermittle Zugriffsregel R zu den Aufrufparametern: 3.1. Falls der Parameter cardHandle nicht null ist, muss das Kartenobjekt des Informationsmodells Karte(cardHandle) ermittelt werden. 3.2. Zu den Parametern (ctId, cardHandle, needCardSession, allWorkplaces) muss mittels Tabelle „TAB_KON_513 Zugriffsregeln Regelzuordnung“ die Zugriffsregel R ermittelt werden. 4. Prüfe die Bedingungen der in Schritt 3 ermittelten Regel R: 4.1. Zur Regel R muss die relevante Spalte in Tabelle „TAB_KON_514 Zugriffsregeln Definition“ ermittelt werden. 4.2. Jede Zeile, die in der Spalte R ein „x“ hat, muss geprüft werden: 4.2.1 Prüfe, ob die in Spalte „Bedingung“ mittels OCL formulierte Bedingung für die Eingangsdaten erfüllt ist.
Varianten/ Alternativen	2. Bei einem Aufruf mit einem cardHandle zu den Kartentypen SMC-KT und UNKNOWN wird Schritt 3 in folgender Variante durchlaufen: Ermittle Zugriffsregel R zu den Aufrufparametern: 3.1. ctId wird zum cardHandle bestimmt Zu den Parametern (ctId, cardHandle = null, needCardSession = false, allWorkplaces = false) muss mittels Tabelle „TAB_KON_513 Zugriffsregeln Regelzuordnung“ die Zugriffsregel R ermittelt werden.
Fehlerfälle	Fehler im Ablauf (Standardablauf oder Varianten) führen in den folgend ausgewiesenen Schritten zu einem Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes: (→1) Es sind nicht alle Pflichtparameter gesetzt, Fehlercode: 4021 (→2) Clientsystem aus dem Aufrufkontext nicht authentifiziert, Fehlercode: 4204 (→3.1) Karte nicht als gesteckt identifiziert,

	Fehlercode: 4008 (→3.2) Zu den Parametern konnte keine Regel ermittelt werden, Fehlercode: 4019 (→4.2.1) Bedingung nicht erfüllt Fehlercode: wie in Spalte „ErrorCode" der geprüften Zeile aus Tabelle „TAB_KON_514-01 Zugriffsregeln Definition"
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	PIC_KON_118 Aktivitätsdiagramm zu „TUC_KON_000 Prüfe Zugriffsberechtigung"

[<=]

Eine Beschreibung aller Zugriffsregeln gibt Tabelle TAB_KON_512.

Tabelle 18: TAB_KON_512 Zugriffsregeln Beschreibung

Regel	Beschreibung
R1	Innerhalb des Mandanten m darf das Clientsystem cs verwendet werden.
R2	Innerhalb des Mandanten m darf das Clientsystem cs auf das Kartenterminal kt zugreifen.
R3	Innerhalb des Mandanten m darf das Clientsystem cs den Arbeitsplatz ap nutzen.
R4	Innerhalb des Mandanten m darf das Clientsystem cs über den Arbeitsplatz ap auf das Kartenterminal kt zugreifen.
R5	Innerhalb des Mandanten m darf das Clientsystem cs über den Arbeitsplatz ap auf die lokal gesteckte eGK zugreifen. Eine Kartensitzung wird nicht benötigt.
R6	Innerhalb des Mandanten m darf das Clientsystem cs über den Arbeitsplatz ap auf die lokal gesteckte eGK zugreifen. Eine Kartensitzung wird benötigt. Wenn bereits eine Kartensitzung besteht, ist sichergestellt, dass sie vom Arbeitsplatz ap gestartet wurde.
R7	Innerhalb des Mandanten m darf das Clientsystem cs über den Arbeitsplatz ap auf die SM-B zugreifen. Es wird dabei sichergestellt, dass es sich um eine im Mandanten verwaltete SM-B handelt.
R8	Innerhalb des Mandanten m darf das Clientsystem cs auf den HBAx zugreifen. Eine Kartensitzung wird nicht benötigt.
R9	Innerhalb des Mandanten m darf das Clientsystem cs auf den HBAx zugreifen. Eine Kartensitzung wird benötigt. Wenn bereits Kartensitzungen zum HBAx bestehen, wird der Zugriff auf den HBAx verhindert, wenn es eine

	Kartensitzung zum selben Clientsystem, aber einer anderen UserId gibt, deren Sicherheitszustand erhöht ist.

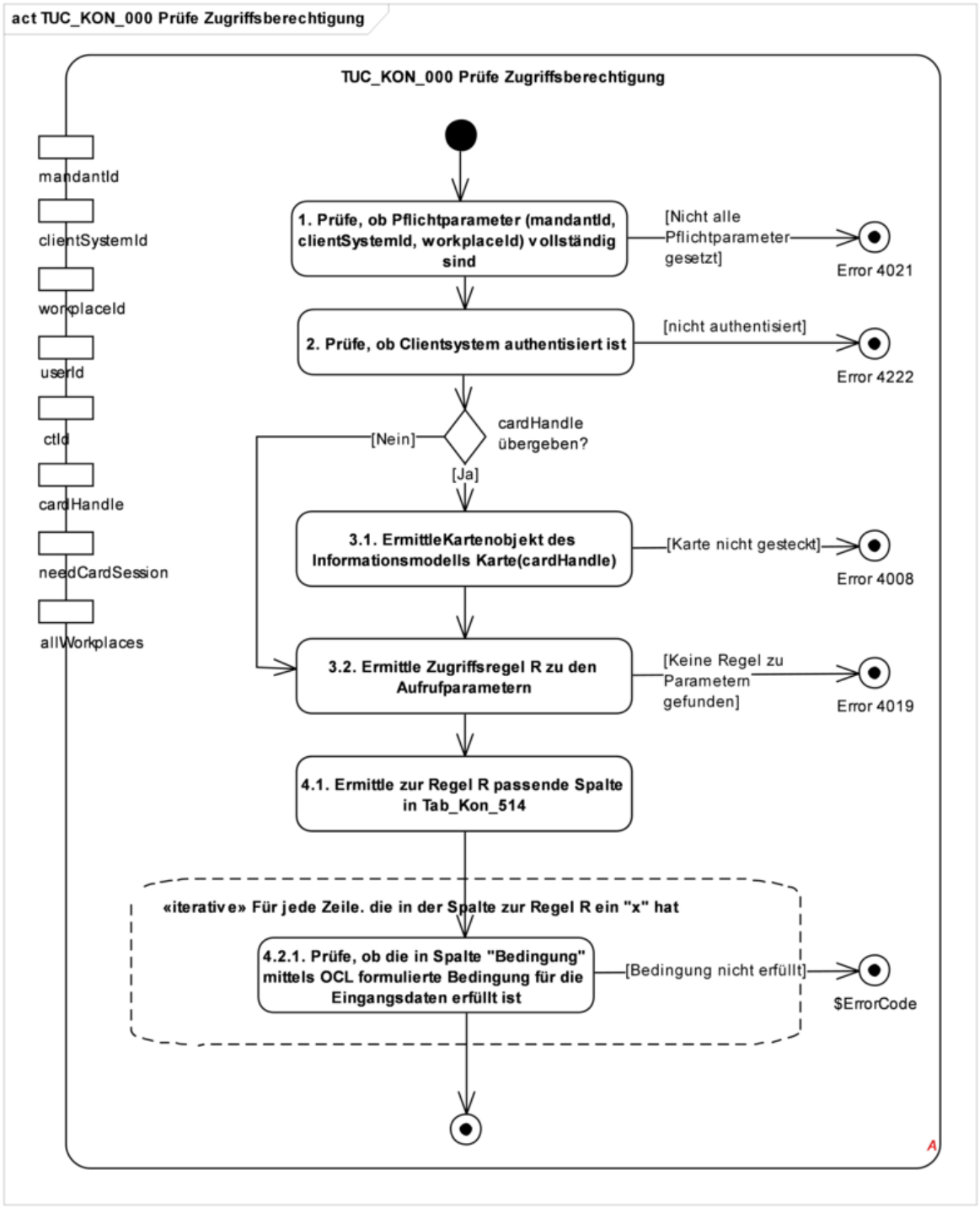


Abbildung 6: PIC_KON_118 Aktivitätsdiagramm zu „TUC_KON_000 Prüfe Zugriffsberechtigung“

Welche Zugriffsregel für einen gegebenen Satz an Aufrufparametern anzuwenden ist, wird in Tabelle TAB_KON_513 ermittelt. Die Pflichtfelder mandantId, clientSystemId und

workplaceId und das optionale Feld userId sind zwar für die Auswertung der Regeln wichtig, tragen aber nicht zur Auswahl der Regel bei und sind daher in der Tabelle nicht vorhanden. Zur Auswahl einer Regel ist relevant,

- ob ctId bzw. cardHandle als Aufrufparameter gesetzt sind (not null) oder leer sind (null),
- von welchem Typ eine Karte ist, falls der Aufrufparameter cardHandle gesetzt ist,
- und welchen Wert die Aufrufparameter „needCardSession" und „allWorkplaces" annehmen.

Tabelle 19: TAB_KON_513 Zugriffsregeln Regelzuordnung

Parameter	R1	R2	R3	R4	R5	R6	R7	R8	R9
ctId	null	not null	null	not null					
cardHandle	null	null	null	null	not null	not null	not null	not null	not null
Karte(cardHandle).type					eGK oder KVK	eGK oder KVK			
Karte(cardHandle).type							SM-B		
Karte(cardHandle).type								HBAx	HBAx
needCardSession	false	false	false	false	false	true	true oder false	false	true
allWorkplaces	true	true	false	false	false	false	false	false	false

Tabelle TAB_KON_514 definiert einzelne Bedingungen, ordnet sie den Regeln zu und definiert ErrorCodes für den Fall, dass eine Bedingung nicht erfüllt ist.

Die Bedingungen in Tabelle TAB_KON_514 sind wie folgt gruppiert:

- Entitäten: Hier wird geprüft, ob die Entitäten, die mit den Aufrufparametern adressiert werden, im Informationsmodell existieren.
- Mandantenbezug: Hier wird geprüft, ob die adressierten Entitäten im Informationsmodell dem adressierten Mandanten zugeordnet sind.
- Relationen: Hier wird geprüft, ob die benötigen Zugriffbeziehungen zum Zugriff auf die adressierten Entitäten im Informationsmodell existieren.
- Kartensitzungen: Hier wird geprüft, ob die benötigte Kartensitzung im Rahmen der bereits existierenden Kartenbeziehungen existieren darf.

Die Fehlercodes mit Beschreibung, ErrorType und Severity Tabelle TAB_KON_515.

Tabelle 20: TAB_KON_514-01 Zugriffsregeln Definition

	Bedingung (siehe Hinweis 1)	R1	R2	R3	R4	R5	R6	R7	R8	R9	Error Code
Entität (siehe Hinweis 2)	`inv : userId <> null`									x	4003
	`let m : Mandant = Mandant(mandantId)` `inv : m <> null`	x	x	x	x	x	x	x	x	x	4021 an der Außenschnittstelle 4004 im Protokoll (siehe Hinweis 3)
	`let cs : Clientsystem = Clientsystem (clientSystemId)` `inv : cs <> null`	x	x	x	x	x	x	x	x	x	4021 an der Außenschnittstelle 4005 im Protokoll (siehe Hinweis 3)
	`let ap : Arbeitsplatz = Arbeitsplatz (workplaceId)` `inv : ap <> null`			x	x	x	x	x	x	x	4021 an der Außenschnittstelle 4006 im Protokoll (siehe Hinweis 3)
	`let kt : Kartenterminal = Kartenterminal (ctId)` `inv : kt <> null`		x		x						4007
	`let k : Karte = Karte (cardHandle)` `inv : k <> null`					x	x	x	x	x	4008

Mandant bezug	`let m : Mandant = Mandant(mandantId)`
	`let cs : Clientsystem = Clientsystem (clientSystemId)`
	`inv : cs.mandant. includes(m)`
		x	x	x	x	x	x	x	x	x	4010

	`let m : Mandant = Mandant(mandantId)`
	`let ap : Arbeitsplatz = Arbeitsplatz (workplaceId)`
	`inv : ap.mandant. includes(m)`
				x	x	x	x	x	x	x	4011

	let m : Mandant = Mandant(mandantId) let kt : Kartenterminal = Kartenterminal(ctId) inv : kt.mandant.includes(m)		x		x						4012
	let m : Mandant = Mandant(mandantId) let k : Karte = Karte(cardHandle) inv : k.kT-Slot.kartenterminal.mandant.includes(m)					x	x	x	x	x	4012
Relation	let k : Karte = Karte(cardHandle) inv : k.SM-B_Verwaltet <> null							x			4009
	let m : Mandant = Mandant(mandantId) let k : Karte = Karte(cardHandle) inv : k.SM-B_Verwaltet.mandant -> includes(m)							x			4013
	let m : Mandant = Mandant(mandantId) let cs : Clientsystem = Clientsystem(clientSystemId) let ap : Arbeitsplatz = Arbeitsplatz(workplaceId) inv : CS_AP.allInstances -> exists(c : CS_AP | c.mandant = m and c.arbeitsplatz = ap and c.clientsystem = cs)			x	x	x	x	x	x	x	4014

	let ap : Arbeitsplatz = Arbeitsplatz (workplaceId) let kt : Kartenterminal = Kartenterminal (ctId) inv : ap.lokalKartenterminal .includes(kt) or ap.entferntKarten terminal .includes(kt)				x						4015
	let ap : Arbeitsplatz = Arbeitsplatz (workplaceId) let kt : Kartenterminal = Karte(cardHandle).kT- Slot.kartenterminal inv : ap.lokalKartenterminal .includes(kt) or ap.entferntKarten terminal .includes(kt)							x	x	x	4015
	let m : Mandant = Mandant (mandantId) let kt : Kartenterminal = Kartenterminal(ctId) let cs : Clientsystem = Clientsystem (clientSystemId) inv : CS_AP.allInstances -> exists(c : CS_AP | c.arbeitsplatz .lokalKartenterminal .includes(kt) or c.arbeitsplatz .entferntKartenterminal .includes(kt) and c.mandant = m and c.arbeitsplatz.mandant .includes(m) and c.clientsystem = cs)		x								4020

	`let ap : Arbeitsplatz = Arbeitsplatz (workplaceId) let kt : Kartenterminal = Karte(cardHandle).kT-Slot.kartenterminal inv : ap.lokalKartenterminal .includes(kt)`					x	x				4016
Karten sitzungen	`let ap : Arbeitsplatz = Arbeitsplatz (workplaceId) let k : Karte = Karte(cardHandle) inv : k.kartensitzung -> not exists(ks : Kartensitzung | ks.arbeitsplatz <> ap)`						x				4017
	`let k : Karte = Karte (cardHandle) let cs : Clientsystem = Clientsystem (clientSystemId) inv : k.kartensitzung -> not exists (ks : Kartensitzung | ks .clientsystem = cs and ks .userId <> userId and ks .authState.size() > 0 )`									x	4018

Erläuterungen zu TAB_KON_514-01:

Hinweis 1:
Jede Bedingung ist als Constraint mittels OCL definiert, ist einzeln prüfbar und hat als Eingangsparameter mandantId, clientSystemId, workplaceId, ctId, cardHandle und userId.

Hinweis 2:
Zur Bezeichnung einer Objektinstanz, die im Informationsmodell vorhanden ist, wird die Notation <<Entitätsbezeichner>>(<<Komma separierte Liste der Identitätsschlüssel>> verwendet.

Hinweis 3:
Bei manchen Bedingungen gibt es unterschiedliche Fehlermeldungen für die Außenschnittstelle und für die interne Protokollierung. Dann wird folgende Notation in Spalte "Error Code" verwendet:

"<<Fehlercode>> an der Außenschnittstelle" für den Fehlercode, der über die Außenschnittstelle zurückgegeben werden muss

"<<Fehlercode>> im Protokoll" für den Fehlercode, der für die interne Protokollierung verwendet werden muss.

Tabelle 21: TAB_KON_515 Fehlercodes TUC_KON_000 „Prüfe Zugriffsberechtigung"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4003	Technical	Error	Keine User-ID angegeben, die zur Identifikation der Kartensitzung_HBAx benötigt wird.
4004	Technical	Error	Ungültige Mandanten-ID
4005	Technical	Error	Ungültige Clientsystem-ID
4006	Technical	Error	Ungültige Arbeitsplatz-ID
4007	Technical	Error	Ungültige Kartenterminal-ID
4008	Technical	Error	Karte nicht als gesteckt identifiziert
4009	Security	Error	SM-B ist dem Konnektor nicht als SM-B_Verwaltet bekannt
4010	Security	Error	Clientsystem ist dem Mandanten nicht zugeordnet
4011	Security	Error	Arbeitsplatz ist dem Mandanten nicht zugeordnet
4012	Security	Error	Kartenterminal ist dem Mandanten nicht zugeordnet
4013	Security	Error	SM-B_Verwaltet ist dem Mandanten nicht zugeordnet
4014	Security	Error	Für den Mandanten ist der Arbeitsplatz nicht dem Clientsystem zugeordnet
4015	Security	Error	Kartenterminal ist weder lokal noch entfernt vom Arbeitsplatz aus zugreifbar

4016	Security	Error	Kartenterminal ist nicht lokal vom Arbeitsplatz aus zugreifbar
4017	Security	Error	Die eGK hat bereits eine Kartensitzung, die einem anderen Arbeitsplatz zugeordnet ist.
4018	Security	Error	Der HBAx hat mindestens eine Kartensitzung zu einer anderen UserId, deren Sicherheitszustand erhöht ist.(Sicherheitszustand wird bei PIN-Eingabe erhöht.)
4019	Technical	Error	Zu den Parametern konnte keine Regel ermittelt werden.
4020	Security	Error	Kartenterminal ist weder lokal noch entfernt über irgendeinen dem Clientsystem zugeordneten Arbeitsplatz aus zugreifbar
4021	Technical	Error	Es sind nicht alle Pflichtparameter mandantId, clientSystemId, workplaceId gefüllt.
4204	Security	Error	Clientsystem aus dem Aufrufkontext konnte nicht authentifiziert werden.

Hinweis zu Fehler 4018: Sicherheitszustand wird bei PIN-Eingabe erhöht.

4.1.1.5 Operationen an der Außenschnittstelle

Keine

4.1.1.6 Betriebsaspekte

TIP1-A_4525 - Initialisierung Zugriffsberechtigungsdienst
Der Konnektor MUSS mit Abschluss der Bootup-Phase den Ist-Zustand transienter Entitäten und Beziehungen des Informationsmodells erfasst haben.
[<=]

TIP1-A_4526 - Bearbeitung Informationsmodell Zugriffsberechtigungsdienst
Für die Administration MUSS der Konnektor eine Administrationsoberfläche zur Pflege des Informationsmodells zur Verfügung stellen. Die Oberfläche muss es ermöglichen, sämtliche persistente Entitäten und Beziehungen des durch Abbildung „PIC_Kon_100 Informationsmodell des Konnektors" und Tabelle „TAB_KON_510 Informationsmodell Constraints" definierten Informationsmodells initial anzulegen, zu ändern und zu löschen.
[<=]

Im Anhang I „Umsetzungshinweise" werden Empfehlungen zur Umsetzung der Administration des Informationsmodells gegeben.

4.1.2 Dokumentvalidierungsdienst

Der Dokumentvalidierungsdienst ist ein Dienst, der nur intern genutzt wird, d. h., dass dessen definierte Verhaltensweisen nur in anderen TUCs des Konnektors nachgenutzt

werden. Er bietet Schnittstellen zum Validieren von Dokumenten an. Dabei werden diejenigen spezifischen Dokumentformate unterstützt, die an den Außenschnittstellen anderer Dienste wie Signatur- und Verschlüsselungsdienst auftreten können (`Alle_DocFormate` gemäß Kapitel 3).

Die jeweils gültigen XML-Schemas der Fachmodule werden den Herstellern von der gematik bereitgestellt.

4.1.2.1 Funktionsmerkmalweite Aspekte

A_18780 - PDF/A-3 DARF NICHT unterstützt werden
Der Konnektor DARF Dokumente im PDF/A-3 Format NICHT unterstützen.
[<=]

4.1.2.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.1.2.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine

4.1.2.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.2.4.1 TUC_KON_080 „Dokument validieren"

TIP1-A_4527-02 - TUC_KON_080 „Dokument validieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_080 „Dokument validieren" umsetzen.

Tabelle 22: TAB_KON_143 – TUC_KON_080 „Dokument validieren"

Element	Beschreibung
Name	TUC_KON_080 „Dokument validieren"
Beschreibung	Dieser TUC prüft das Format eines Dokuments und führt dokumententyp-spezifische Validierungen durch. Unterstützt werden `Alle_DocFormate`(außer „Binär").
Auslöser	• Aufruf durch Fachmodul • Aufruf durch Basisdienst
Vorbedingungen	Keine
Eingangsdaten	• documentToBeValidated (Zu validierendes Dokument.) • documentFormat (mögliche Werte siehe Definition `Alle_DocFormate`; Formatangabe für das Dokument) Optional für XML-Dokumente:

	• xmlSchemas – optional/nur für XML-Dokumente (XML-Schema und ggf. weitere vom Hauptschema benutzte Schemata) • signaturePolicyIdentifier – optional/nur für XML-Formate gemäß einer referenzierten Signaturrichtlinie (URI identifiziert die Signaturrichtlinie)
Komponenten	Konnektor
Ausgangsdaten	• documentValidationProtocol (Prüfprotokoll) Die Ausprägung dieses Konnektor-internen Parameters erfolgt herstellerspezifisch.
Nachbedingungen	Keine

Standardablauf	**Validierung der Dokumente auf Typkonformität**
	Der Konnektor führt je nach Format des Dokuments (documentFormat) eine der folgenden Prüfungen durch:
	
	A) XML-Dokumentvalidierung
	Im Fall eines XML-Dokuments prüft der Konnektor:
	• Prüfe die XML-Wohlgeformtheit des Dokumentes (documentToBeValidated)
	• Wenn signaturePolicyIdentifier vorhanden ist, dann ermittle das xmlSchema aus der referenzierten Signaturrichtlinie und prüfe die Validität von documentToBeValidated in Bezug auf das hinterlegte XML-Schema. Der Eingangsparameter xmlSchemas wird ignoriert.
	• Wenn signaturePolicyIdentifier nicht vorhanden ist und xmlSchemas übergeben wurden, dann prüfe die Wohlgeformtheit von xmlSchemas und die Validität von documentToBeValidated in Bezug auf xmlSchemas.
	• Wenn nicht durch Prüfung gegen XML-Schema bereits erfolgt, dann prüfe die Eindeutigkeit der ID-Attributwerte im XML-Dokument.
	
	B) PDF/A-Dokumentvalidierung
	PDF/A-Dokumente werden geprüft, ob sie sich als PDF/A Dokumente in ihren PDF/A-Metadaten ausweisen: Es wird geprüft, ob diese eines der folgenden Elemente enthalten
	
	`<pdfaid:part xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/">1</pdfaid:part>`
	
	`<pdfaid:part xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/">2</pdfaid:part>`
	
	`<pdfaid:part xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/">3</pdfaid:part>`

	<rdf:Description rdf:about="" xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/" pdfaid:part="1" pdfaid:conformance="B"/> <rdf:Description rdf:about="" xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/" pdfaid:part="2" pdfaid:conformance="B"/> <rdf:Description rdf:about="" xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/" pdfaid:part="3" pdfaid:conformance="B"/> C) TIFF-Dokumentvalidierung Der Konnektor prüft, ob das Dokument an Hand seiner ersten 8 Byte als TIFF-Dokument [TIFF6] zu identifizieren ist. D) MIME-Dokumentvalidierung Die Struktur von MIME-Dokumenten wird entsprechend [MIME] validiert. E) Text-Dokumentvalidierung Der Konnektor prüft die Konformität zum im Dokumentenformat vorgegebenen Character-Encoding. Für Binärdokumente findet keine Validierung statt. Hinweis: Byte-order-marks (BOM) sind im Rahmen von UTF-8 kodierten Dokumenten gemäß UTF8 Standard ([RFC3629], Kapitel 6) erlaubt, aber nicht notwendigerweise im Dokument vorhanden.
Varianten/ Alternativen	
Fehlerfälle	**Standardablauf:** Bei der Dokumentenvalidierung protokolliert der TUC alle aufgetretenen Fehler im Rückgabewert documentValidationProtocol. (→A) Fehlerfälle bei XML-Dokumentvalidierung Wenn keine Schemata übergeben wurden (xmlSchemas oder signaturePolicyIdentifier nicht vorhanden): Fehlercode 4193 Wenn eines der übergebenen Schemata selbst nicht wohlgeformt oder invalide ist, wird Fehlercode 4026 gemeldet. Wenn das XML-Dokument nicht wohlgeformt ist, wird Fehlercode 4022 gemeldet. Das XML-Dokument ist nicht valide in Bezug auf das zur Validierung benutzte Schema (xmlSchemas bzw. signaturePolicyIdentifier): Fehlercode 4023. (→B) Fehlerfälle bei PDF/A-Dokumentvalidierung Bei fehlgeschlagener Validierung: Fehlercode 4024, Dokumentformat = PDF/A (→C) Fehlerfälle bei TIFF-Dokumentvalidierung Bei fehlgeschlagener Validierung: Fehlercode 4024, Dokumentformat = TIFF (→D) Fehlerfälle bei MIME-Dokumentvalidierung Bei fehlgeschlagener Validierung: Fehlercode 4024, Dokumentformat = MIME (→E) Fehlerfälle bei Text-Dokumentvalidierung

	Bei fehlgeschlagener Validierung: Fehlercode 4024, Dokumentformat = Text
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 23: TAB_KON_144 Fehlercodes TUC_KON_080 „Dokument validieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4022	Security	Error	XML-Dokument nicht wohlgeformt
4023	Security	Error	XML-Dokument nicht valide in Bezug auf XML-Schema
4024	Security	Error	Formatvalidierung fehlgeschlagen (<Dokumentformat>) Der Parameter Dokumentformat kann die Werte XML, PDF/A, TIFF, MIME und Text annehmen.
4026	Security	Error	XML-Schema nicht valide
4193	Security	Warning	kein XML-Schema für XML-Dokument vorhanden

[<=]

4.1.2.5 Operationen an der Außenschnittstelle

Keine

4.1.2.6 Betriebsaspekte

Keine

4.1.3 Dienstverzeichnisdienst

Der Dienstverzeichnisdienst liefert dem aufrufenden Clientsystem sowohl Informationen über die Version und Produktkenndaten des Konnektors, als auch die SOAP-Endpunkte, über die das Clientsystem die einzelnen Dienstoperationen erreichen kann.

4.1.3.1 Funktionsmerkmalweite Aspekte

Die Endpunkte der Basisdienste werden in WSDL spezifiziert. Diese Endpunkte und weitere konnektormodellspezifische Informationen werden dem Clientsystem in Form eines Dienstverzeichnisdienstes gesammelt angeboten.

Der prinzipielle Ablauf sieht dabei folgendermaßen aus:

Das Clientsystem ruft beim Initialisieren des Systems mit HTTP-GET die vordefinierte URL: https://<ANLW_LAN_IP_ADDRESS

oder MGM_KONN_HOSTNAME>/connector.sds oder*http://<ANLW_LAN_IP_ADDRESS oder MGM_KONN_HOSTNAME>/connector.sds* des Konnektors auf.

Der Konnektor stellt die Liste der Dienste, der Versionen und die Endpunkte der Dienste in einem XML-Dokument zusammen. Jeder über SOAP erreichbare Basisdienst des Konnektors wird in dieser Liste geführt. Ferner können Fachmodule ihre eigenen Endpunkte über TUC_KON_041 „Einbringen der Endpunktinformationen während der Bootup-Phase" einbringen. Die so erstellte Liste der Dienste wird als Antwort an das Clientsystem übergeben.

Das Clientsystem prüft, ob die gewünschten Dienste und Versionen unterstützt werden und merkt sich die Endpunkte der Dienste für die späteren Aufrufe. Danach kann das Clientsystem diese Dienstendpunkte nach Bedarf aufrufen.

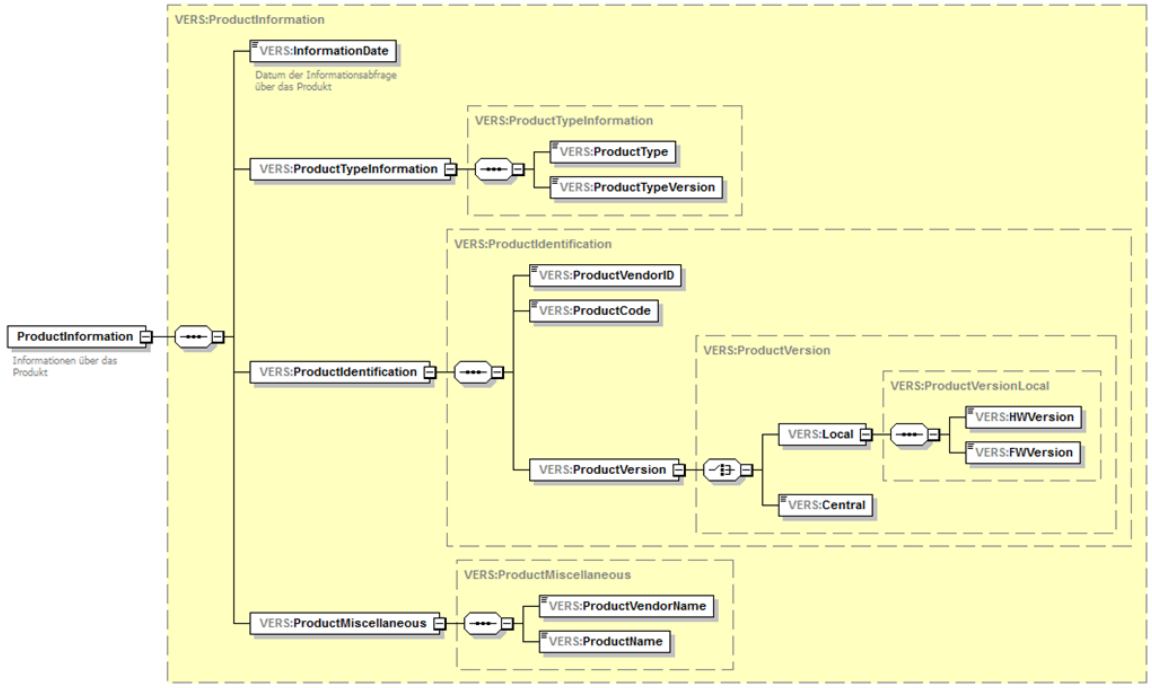
TIP1-A_4528 - Bereitstellen des Dienstverzeichnisdienst
Der Konnektor MUSS den Dienstverzeichnisdienst anbieten. Dieser Dienst veröffentlicht auf: https://$ANLW_LAN_IP_ADDRESS oder $MGM_KONN_HOSTNAME>/connector.sds
oder http://$ANLW_LAN_IP_ADDRESS oder $MGM_KONN_HOSTNAME>/connector.sds.
Die Datei MUSS über https erreichbar sein.
Wenn (ANCL_DVD_OPEN = Enabled) oder (ANCL_TLS_MANDATORY = Disabled) MUSS die Datei auch über http erreichbar sein.
[<=]

TIP1-A_4529-02 - Formatierung der Ausgabedatei
Das XML-Dokument, welches als „connector.sds" dem Aufrufer zurückgeliefert wird, MUSS gemäß dem Schema „`conn/ServiceDirectory.xsd`" formatiert sein.
`conn/ServiceDirectory.xsd` referenziert die Schemata „`tel/version/ProductInformation.xsd`" (siehe [gemSpec_OM]) und „`conn/ServiceInformation.xsd`".
TAB_KON_516, TAB_KON_517 und TAB_KON_518 beschreiben die Elemente der zu verwendenden Schemastruktur.

Tabelle 24: TAB_KON_516 Basisanwendung Dienstverzeichnisdienst

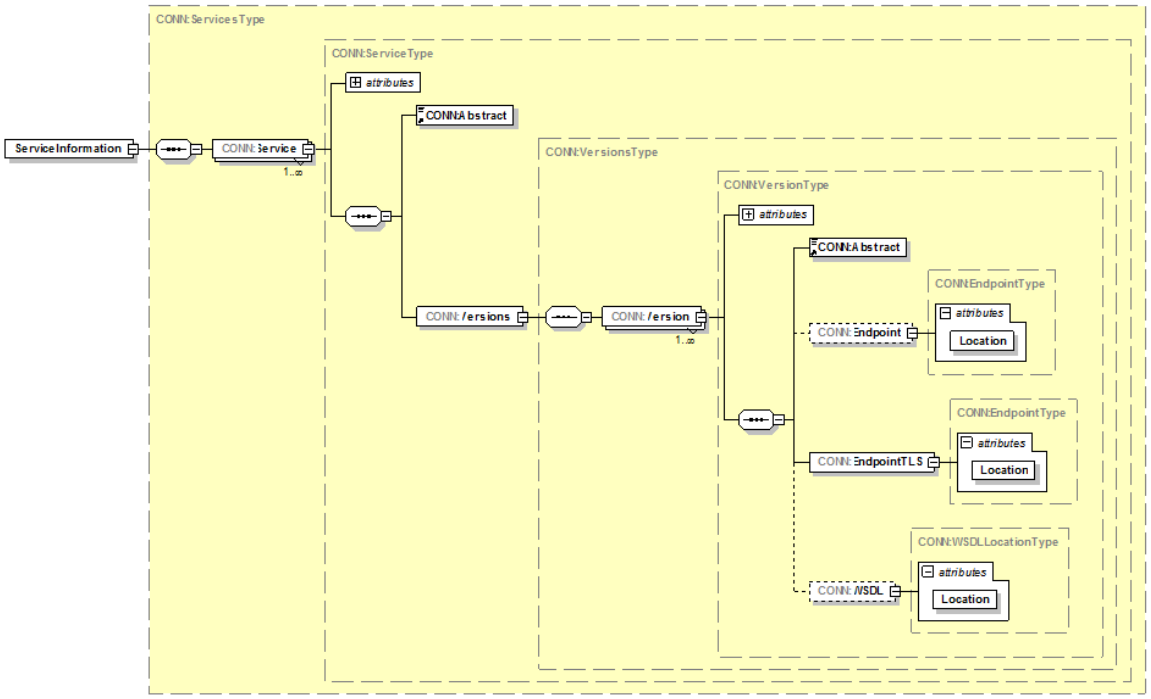
Name	ConnectorServiceDirectory
Version	3.1.0 (XSD-Version)
Namensraum	Siehe GitHub
Namensraum-Kürzel	CONN
Operationen	Lesen der vom Konnektor unterstützten Dienste
WSDL	Keine
Schema	ServiceDirectory.xsd

Tabelle 25: TAB_KON_517 Schemabeschreibung Produktinformation (ProductInformation.xsd)



Element	Bedeutung
ProductInformation/InformationDate	Datum der Informationsabfrage über das Produkt
ProductInformation/ProductTypeInformation/ ProductType	Produkttyp (Konnektor)
ProductInformation/ProductTypeInformation/ ProductTypeVersion	Produkttypversion des Konnektormodells
ProductInformation/ProductIdentification/ ProductVendorID	ID des Konnektorherstellers
ProductInformation/ProductIdentification/ ProductCode	Produktkürzel
ProductInformation/ProductIdentification/ ProductVersion/Local/HWVersion	Hardwareversion des Konnektormodells
ProductInformation/ProductIdentification/ ProductVersion/Local/FWVersion	Firmwareversion des Konnektormodells
ProductInformation/ProductMiscellaneous/ ProductVendorName	Name des Konnektorherstellers
ProductInformation/ProductMiscellaneous/ ProductName	Produktname

Tabelle 26: TAB_KON_518 Schemabeschreibung Serviceinformation (Serviceinformation.xsd)



Element	Bedeutung
ServiceInformation/Service	Element beschreibt einen Dienst oder ein Fachmodul
ServiceInformation/Service/@Name	Name des Dienstes. Dieser Wert korrespondiert mit dem Feld „Name" aus der jeweiligen Basisanwendung/Dienstbeschreibung (englischer Dienstname in Tabelle TAB_KON_798).
ServiceInformation/Service/Abstract	eine kurze textuelle Beschreibung des Dienstes/Fachmoduls
ServiceInformation/Service/Versions	die Liste der unterstützten Versionen
ServiceInformation/Service/Versions/Version	Beschreibung der Dienstversion/Fachmodulversion
ServiceInformation/ Service/Versions/Version/@TargetNamespace	der Namensraum der Dienst-/Fachmodulversion
ServiceInformation/Service/Versions/Version/@Version	Vollständige Versionsnummer (Konnektordienstversion) des Dienstes/Fachmoduls. Dieser Wert entspricht dem Wert „WSDL-Version" des jeweiligen Dienstes in Tabelle TAB_KON_798.

ServiceInformation/Service/Versions/Version/Abstract	Eine kurze textuelle Beschreibung dieser Version des Dienstes/Fachmoduls
ServiceInformation/ Service/Versions/Version/EndpointTLS/@Location	Absoluter URL des über TLS erreichbaren Dienstes.
ServiceInformation/ Service/Versions/Version/Endpoint/@Location	Absoluter URL des erreichbaren Dienstes (ohne TLS).
ServiceInformation/ Service/Versions/Version/WSDL/@Location	(optional) Absoluter URL der WSDL-Beschreibung

[<=]

TIP1-A_4530 - Aufbau Dienst URLs
Die URLs der Dienste KÖNNEN herstellerspezifisch aufgebaut werden.
[<=]

4.1.3.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.1.3.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine

4.1.3.4 Interne TUCs, auch durch Fachmodule nutzbar

Da der Konnektor als Black-Box mit inkludierten Fachmodulen ohne erkennbare Innenschnittstellen spezifiziert wird, stellt der folgende TUC lediglich einen Mechanismus zur editoriellen Kopplung der Fachmodulspezifikationen mit der Konnektorspezifikation dar:

4.1.3.4.1 TUC_KON_041 „Einbringen der Endpunktinformationen während der Bootup-Phase"

TIP1-A_4531 - TUC_KON_041 „Einbringen der Endpunktinformationen während der Bootup-Phase"
Der Dienstverzeichnisdienst des Konnektors MUSS es den Fachmodulen ermöglichen, die zum jeweiligen Fachmodul gehörenden Endpunkte während der Bootup-Phase des Konnektors in den Dienstverzeichnisdienst einzubringen.

Tabelle 27: TAB_KON_519 - TUC_KON_041 „Einbringen der Endpunktinformationen während der Bootup-Phase"

Element	Beschreibung
Name	TUC_KON_041 „Einbringen der Endpunktinformationen während der Bootup-Phase"
Beschreibung	Fachmodule MÜSSEN ihre Endpunktinformationen während der Bootup-Phase in den Dienstverzeichnisdienst einbringen können.

Auslöser und Vorbedingungen	Keine
Eingangsdaten	• serviceInformation (Ein XML-Dokument mit dem Wurzelelement „ServiceInformation" gemäß dem Schema „Serviceinformation.xsd". Eine Beschreibung des Schemas befindet sich in TAB_KON_518.)
Komponenten	Konnektor, Fachmodule
Ausgangsdaten	• Keine
Standardablauf	Die übergebenen Serviceinformationen des Fachmoduls werden in die Gesamtstruktur „connector.sds" aufgenommen. Falls beim Speichern der eingebrachten Endpunktinformationen ein Fehler auftritt, wird Fehler 4027 ausgelöst.
Varianten/Alternativen	Keine
Fehlerfälle	4027: Die Endpunktinformationen konnten nicht übernommen werden.
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 28: TAB_KON_520 Fehlercodes TUC_KON_041 „Einbringen der Endpunktinformationen während der Bootup-Phase"

Fehlercode	ErrorType	Severity	Fehlertext
4027	Technical	Error	Die Endpunktinformationen konnten nicht übernommen werden.

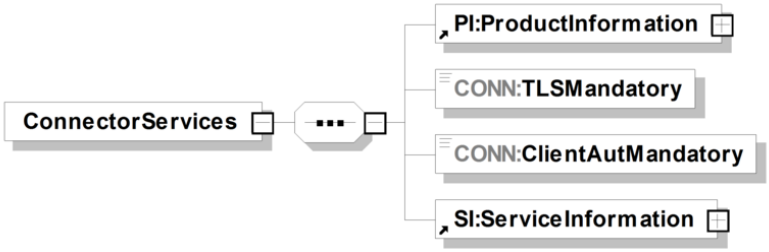
[<=]

4.1.3.5 Operationen an der Außenschnittstelle

TIP1-A_4532 - Schnittstelle der Basisanwendung Dienstverzeichnisdienst
Der Dienstverzeichnisdienst des Konnektors MUSS die in Tabelle TAB_KON_521 Schnittstelle der Basisanwendung Dienstverzeichnisdienst beschriebene Schnittstelle anbieten.

Tabelle 29: TAB_KON_521 Schnittstelle der Basisanwendung Dienstverzeichnisdienst

Dienstname	ConnectorServiceDirectory
Beschreibung	Der Aufruf liefert Angaben über den Hersteller, über das Konnektormodell und die Liste der Dienste, Konnektordienstversionen (KDV) und Endpunkte der Dienste.
Aufruf	GET /connector.sds HTTP/1.1 Host: ANLW_LAN_IP_ADDRESS oder MGM_KONN_HOSTNAME

<table>
<tr><td rowspan="8">Rückgabe</td><td colspan="2">Das Antwortdokument ist in der Schemadatei <code>ServiceDirectory.xsd</code> beschrieben.</td></tr>
<tr><td colspan="2">ConnectorServices</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>ProductInformation</td><td>Kurzbeschreibung des Konnektormodells</td></tr>
<tr><td>ServiceInformation</td><td>Beschreibung der Dienste</td></tr>
<tr><td colspan="2">ProductInformation:
Das Schema wird in TAB_KON_517 beschrieben. Die Felder sind gemäß [gemSpec_OM] zu befüllen und gemäß dem Schema „<code>ProductInformation.xsd</code>“ zu formatieren.</td></tr>
<tr><td colspan="2">TLS-Mandatory: Boolean Wert der festlegt, ob die Verwendung eines TLS-Kanals für Dienstaufrufe verpflichtend ist.
Definierende Variable ist: ANCL_TLS_MANDATORY
ClientAutMandatory: Boolean Wert der festlegt, ob Client Authentifizierung verpflichtend ist.
Definierende Variable ist: ANCL_CAUT_MANDATORY.</td></tr>
<tr><td colspan="2">ServiceInformation:
Das Schema wird in TAB_KON_518 beschrieben. Die Felder sind gemäß dem Schema <code>ServiceInformation.xsd</code> zu formatieren.
Falls (ANCL_CAUT_MANDATORY = Enabled) oder (ANCL_TLS_MANDATORY = Enabled), MUSS die Rückgabedatei ausschließlich https-Endpunkte enthalten.</td></tr>
<tr><td>Fehlercodes</td><td colspan="2">Die Standard HTTP1.1 Fehlercodes [RFC2616]</td></tr>
<tr><td>Vorbedingungen</td><td colspan="2">Keine</td></tr>
<tr><td>Nachbedingungen</td><td colspan="2">Keine</td></tr>
<tr><td>Hinweise</td><td colspan="2">Keine</td></tr>
</table>

[<=]

4.1.3.6 Betriebsaspekte

TIP1-A_4533 - Dienstverzeichnisdienst initialisieren.
Mit Abschluss der Bootup-Phase MUSS der Dienstverzeichnisdienst an der Außenschnittstelle die vollständige Liste aller Services bereitstellen, die der

Anwendungskonnektor den Clientsystemen anbietet, inklusive der Services der Fachmodule.
[<=]

4.1.4 Kartenterminaldienst

Die Aufgabe des Kartenterminaldienstes ist das Management aller vom Konnektor adressierbaren Kartenterminals. Dies umfasst alle administrativen Prozesse (insbesondere das Pairing, vgl. [gemSpec_KT#2.5.2]). Ferner kapselt der Kartenterminaldienst die Zugriffe auf Kartenterminals durch Basisdienste und Fachmodule.

Für die TLS-Verbindungen zu den Kartenterminals muss der Konnektor die Vorgaben aus [gemSpec_Krypt#3.3.2] und hinsichtlich ECC-Migration die Vorgaben aus [gemSpec_Krypt#5] befolgen.

Innerhalb des Kartenterminaldienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „CT"
- Konfigurationsparameter: „CTM_"

Der Kartenterminaldienst verwaltet hinsichtlich der Kartenterminals mindestens die in der informativen Tabelle TAB_KON_522 Parameterübersicht des Kartenterminaldienstes ausgewiesenen Parameter, weitere herstellerspezifische Parameter sind möglich. Die normative Festlegung wann welche Parameter mit welchen Werten belegt werden, erfolgt in den folgenden Abschnitten und Unterkapiteln.

Dabei beschrieben CTM_xyz-Bezeichner Parameter, die den Dienst als Ganzes betreffen. Zu jedem Kartenterminal selbst werden dessen Parameter in einem CT-Object gekapselt. Die folgende Tabelle zeigt die Attribute der jeweiligen CT-Objekte über Punktschreibweise.

Tabelle 30: TAB_KON_522 Parameterübersicht des Kartenterminaldienstes

ReferenzID	Belegung	Zustandswerte
CTM_CT_LIST	Liste von CT-Objekten	Eine Liste von Repräsentanzen (CT-Objects) der dem Konnektor bekannten Kartenterminals.
Pro CTM_CT_LIST Eintrag:		
Gerätekenndaten		
CT.CTID	Identifier	Eindeutige, statische Identifikation des Kartenterminals
CT.IS_PHYSICAL	Ja/Nein	Kennzeichnung, ob es sich um ein physisches oder logisches Kartenterminal handelt, zur Unterscheidung von eHealth-Kartenterminals und HSM-Bs. Da dieser Unterschied gemäß der aktuellen HSM-B-Lösung für den Konnektor transparent ist, wird der Parameter in dieser Spezifikation immer auf „Ja" gesetzt.

		Der Parameterwert „Nein" ist für zukünftige Nutzung vorgesehen.
CT.MAC_ADRESS	MAC-Adresse	Die MAC-Adresse des Kartenterminals
CT.HOSTNAME	String	SICCT-Terminalname des Kartenterminals, auch als FriendlyName bezeichnet
CT.IP_ADRESS	IP-Adresse	Die IP-Adresse des Kartenterminals
CT.TCP_PORT	Portnummer	Der TCP-Port des SICCT-Kommandointerpreters des Kartenterminals
CT.SLOTCOUNT	Nummer	Anzahl der Slots des Kartenterminals
CT.SLOTS_USED	Liste	Liste der aktuell mit Karten belegten Slots
CT.PRODUCT INFORMATION	Inhalt ProductInformation.xsd	Die Herstellerinformationen zum Kartenterminal gemäß [gemSpec_OM]
CT.EHEALTH_ INTERFACE_VERSION	Version	Die EHEALTH-Interface-Version des Kartenterminals, die mittels GET STATUS aus dem Element VER des Discretionary Data Objects ermittelt wurde.
CT.VALID_VERSION	Boolean	True, wenn die Version des Kartenterminals (CT.EHEALTH_INTERFACE_VERSION) durch den Konnektor unterstützt wird, d.h. zu den in CTM_SUPPORTED_KT_VERSIONS passt Default-Wert = false
CT.DISPLAY_CAPABILITIES	Datenstruktur	Displayeigenschaften wie in der Struktur Display Capabilities Data Object in [SICCT#5.5.10.17] beschrieben
Pairingdaten		
CT.SMKT_AUT	X.509-Cert	C.SMKT.AUT-Zertifikat des Kartenterminals, gespeichert im Rahmen des Pairings
CT.SHARED_SECRET		ShS.KT.AUT, gespeichert im Rahmen des Pairings
Verbindungsdaten		
CT.CORRELATION	bekannt zugewiesen gepairt	Der Korrelationsstatus zum Konnektor:

	aktiv aktualisierend	• bekannt (über Service Announcement/Service Discovery gelernte Kartenterminals), • zugewiesen (durch den Administrator aus dem Bereich der bekannten Kartenterminals oder manuell konfigurierte Kartenterminals), • gepairt (Pairing erfolgreich aber noch nicht zum Verbindungsaufbau freigegeben) • aktiv (durch Administrator zum Verbindungsaufbau freigegeben), • aktualisierend (ein laufender Updatevorgang, ausgelöst durch den Konnektor; Der Zustand tritt ein, wenn der Kartenterminaldienst das Event „KSR/UPDATE/START" fängt und endet mit dem Event „KSR/UPDATE/END")
CT.CONNECTED	Ja/Nein	Der Verfügbarkeitsstatus des Kartenterminals (Ja = nach Aufbau der TLS-Verbindung und erfolgter zweiter Authentifizierung)
CT.ACTIVEROLE	User/Admin	Benutzerrolle, die für die aktuelle Session verwendet wird
KT-Admin-Credentials		
CT.ADMIN_USERNAME	String	Username des Administrators am KT
CT.ADMIN_PASSWORD	String	Password des Administrators am KT

Zum besseren Verständnis sind die Zustände, die ein Kartenterminal einnehmen kann, im nachfolgenden Zustandsdiagramm PIC_KON_071 dargestellt.

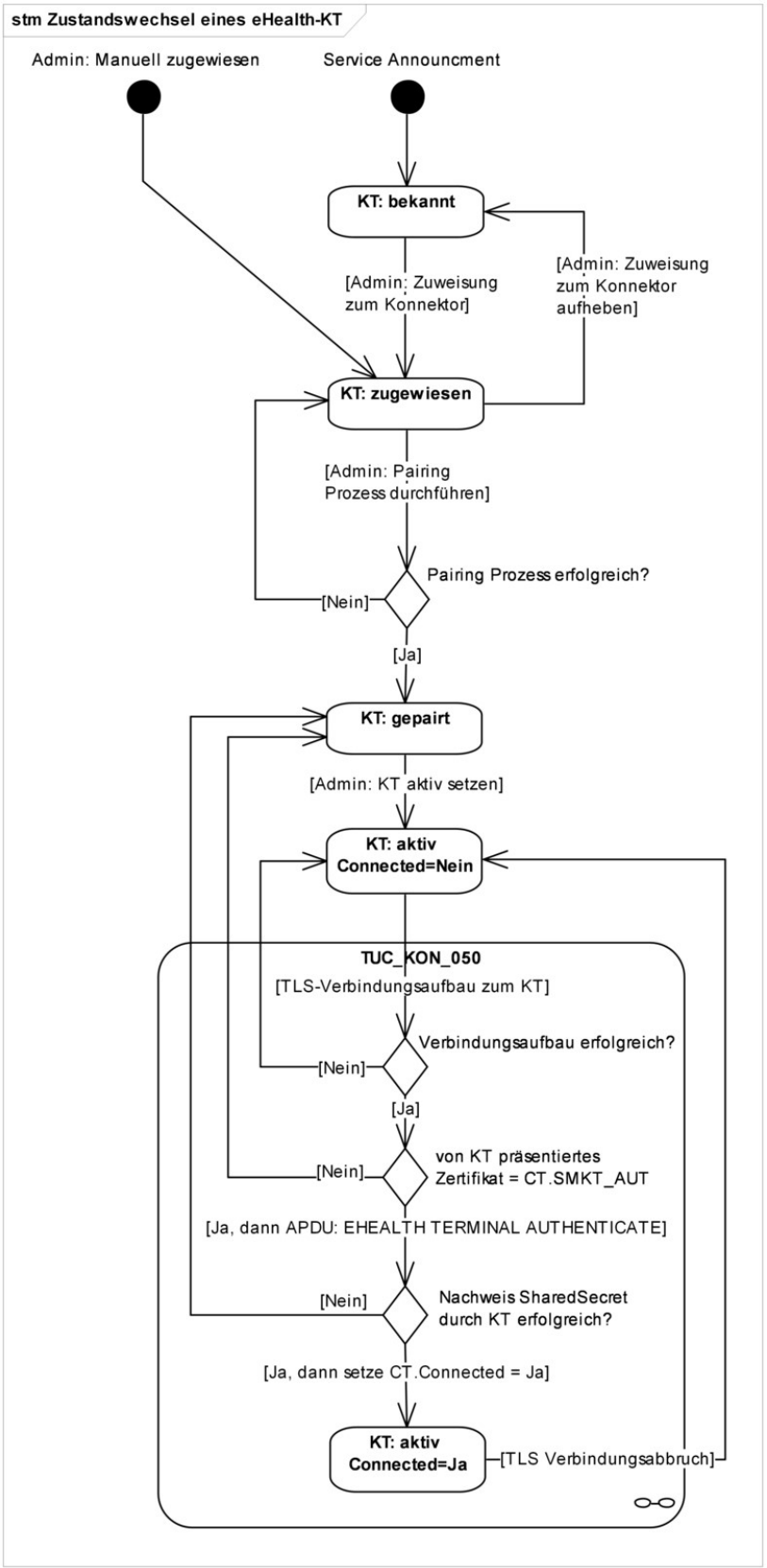


Abbildung 7: PIC_KON_071 Korrelationszustände eines eHealth-KT

4.1.4.1 Funktionsmerkmalweite Aspekte

TIP1-A_4534 - Kartenterminals nach eHealth-KT-Spezifikation
Der Kartenterminaldienst MUSS Kartenterminals nach der eHealth-Kartenterminal Spezifikation [gemSpec_KT] unterstützen.
[<=]

Zur Unterstützung von HSM-Bs benötigt der Konnektor virtuelle Kartenterminals (`CT.IS_PHYSICAL=Nein`), in denen virtuelle SMC-Bs „stecken" können (siehe Kapitel 4.1.4). Diese Kartenterminals werden innerhalb des Zugriffsberechtigungsdienstes sowie des Systeminformationsdienstes wie normale Kartenterminals berücksichtigt. Weitere Details zu den logischen Kartenterminals finden sich im Kapitel Betriebsaspekte.

TIP1-A_4535 - Unterstützung logischer Kartenterminals für HSMs
Der Kartenterminaldienst MUSS logische Kartenterminals mit logischen Slots unterstützen. Zu jedem verwalteten HSM (siehe Kartendienst) MUSS der Konnektor ein oder mehrere logische Kartenterminal mit folgenden Bedingungen vorhalten:

- Jedes logische KT MUSS als CT-Object mit eindeutiger CTID in CTM_CT_LIST enthalten sein
- Die CT-Attribute MÜSSEN gemäß TAB_KON_522 Parameterübersicht des Kartenterminaldienstes gesetzt werden.

[<=]

TIP1-A_4536 - TLS-Verbindung zu Kartenterminals halten
Der Kartenterminaldienst MUSS jede mit einem Kartenterminal etablierte Verbindung durch Nutzung des in [SICCT#6.1.4.5] definierten Keep-Alive Mechanismus halten. Der Konnektor MUSS für das Heartbeat-Interval gemäß [SICCT#6.1.4.5] den Wert CTM_KEEP_ALIVE_INTERVAL verwenden und beim Ausbleiben von Terminal-Antworten eines Kartenterminals nach der Anzahl von CTM_KEEP_ALIVE_TRY_COUNT Versuchen die Netzwerkverbindung zu diesem Kartenterminal beenden.
[<=]

TIP1-A_6725 - Lebensdauer von Textanzeigen am Kartenterminal
Der Konnektor MUSS steuern, dass Textanzeigen am Kartenterminal nur so lange angezeigt werden, wie sie im jeweiligen Anwendungskontext benötigt werden.
[<=]

Ziel der Textanzeigen am Kartenterminal ist die Kommunikation mit dem Benutzer zur Unterstützung der Anwendungsfälle. Die Anzeige am Kartenterminal muss daher einen engen zeitlichen Bezug zum jeweiligen Anwendungskontext haben.

Nachrichten, deren Anwendungskontext mit einem Event beendet werden, wie etwa die Aufforderung zum Stecken der Karte im Kontext von TUC_KON_056, deren inhaltliche Berechtigung mit dem Stecken der Karte erlischt, (ebenso zum Ziehen der Karte im Rahmen von TUC_KON_057) müssen sofort gelöscht werden, wenn das Event eintritt.

Nachrichten, deren Lebensdauer nicht durch ein natürliches Event beendet wird, müssen eine vordefinierte Lebensdauer erhalten, die per Konfiguration an die Bedürfnisse der Leistungserbringer anpassbar sein sollte. Das gilt für Ergebnisanzeigen oder Anzeigen von Fehlern.

TIP1-A_4537 - KT-Statusanpassung bei Beendigung oder Timeout einer Netzwerkverbindung
Tritt ein Timeout ein oder wird eine Netzwerkverbindung zu einem Kartenterminal (oder zu einem HSM, welches einem logischen Kartenterminal zugeordnet ist) beendet oder zurückgesetzt und ist CT.CONNECTED = Ja, so MUSS der Konnektor:

- CT.CONNECTED für das Kartenterminal auf „Nein" setzen
- Für jeden in CT.SLOTS_USED gelisteten Slot X zur weiteren internen Bearbeitung TUC_KON_256 {
  topic = „CT/SLOT_FREE";
  eventType = Op;
  severity = Info;
  parameters = („CtID=$CT.CTID, SlotNo=$X");
  doLog = false;
  doDisp = false }
  rufen
- TUC_KON_256 {
  topic = „CT/DISCONNECTED";
  eventType = Op;
  severity = Info;
  parameters = („CtID=$CT.CTID, Hostname=$CT.HOSTNAME") }
  auslösen
- CT.SLOTS_USED leeren

[<=]

TIP1-A_4538 - Wiederholter Verbindungsversuch zu den KTs
Sind in CTM_CT_LIST Kartenterminals mit CT.CONNECTED=Nein und CT.VALID_VERSION = True und CT.CORRELATION = „aktiv" und ist `CTM_SERVICE_DISCOVERY_CYCLE`>0, MUSS der Konnektor im Zeitabstand`CTM_SERVICE_DISCOVERY_CYCLE`-Minuten entweder eine Service Discovery-Nachricht an alle KTs als Broadcast oder an jedes einzelne dieser unverbundenen KTs als Unicast senden.
[<=]

TIP1-A_4538-02 - ab PTV4: Wiederholter Verbindungsversuch zu den KTs
Sind in CTM_CT_LIST Kartenterminals mit CT.CONNECTED=Nein und CT.VALID_VERSION = True und CT.CORRELATION = „aktiv" und ist `CTM_SERVICE_DISCOVERY_CYCLE`>0, MUSS der Konnektor im Zeitabstand`CTM_SERVICE_DISCOVERY_CYCLE`-Minuten an jedes einzelne dieser unverbundenen KTs eine Service-Discovery-Nachricht als Unicast senden.
[<=]

TIP1-A_6478 - Erlaubte SICCT-Kommandos bei CT.CONNECTED=Nein
Der Kartenterminaldienst DARF SICCT-bzw. EHEALTH-Kommandos NICHT an ein Kartenterminal senden, wenn für dieses Kartenterminal CT.CONNECTED=Nein gesetzt ist. Ausgenommen hiervon sind die in TAB_KON_785 gelisteten EHEALTH- bzw. SICCT-Kommandos.
[<=]

Tabelle 31: TAB_KON_785 Erlaubte SICCT-Kommandos bei CT.CONNECTED=Nein

SICCT-Kommando
SICCT CT INIT CT SESSION
SICCT CT CLOSE SESSION
SICCT GET STATUS

SICCT SET STATUS
SICCT CT DOWNLOAD INIT
SICCT CT DOWNLOAD DATA
SICCT CT DOWNLOAD FINISH
EHEALTH TERMINAL AUTHENTICATE

TIP1-A_4539 - Robuster Kartenterminaldienst
Das Ziehen einer Karte während einer Kartenaktion DARF NICHT dazu führen, dass das verwaltete Kartenterminal im Anschluss durch den Konnektor nicht weiter genutzt werden kann. Die entsprechende Ressource MUSS nach Erkennung der Fehlersituation freigegeben werden. Ein manuelles Eingreifen DARF NICHT erforderlich sein.
[<=]

TIP1-A_5408 - Terminal-Anzeigen beim Anfordern und Auswerfen von Karten
Der Konnektor MUSS beim Anfordern und Auswerfen von Karten die folgenden Display-Nachrichten für die Anzeige im Kartenterminal verwenden, wenn der Aufrufer keine konkrete Display-Nachricht übergeben hat. Der Verweis auf den Kartenterminal-Slot SOLL in der Display-Nachricht entfallen, wenn es keine Slot-Auswahl am Kartenterminal gibt.

Tabelle 32: TAB_KON_727 Terminalanzeigen beim Anfordern und Auswerfen von Karten

Kontext	Kartentyp	Terminal-Anzeige
Karte anfordern	EGK	Bitte•***0x0B***eGK•***0x0B***in •***0x0B***SLOT *X*•***0x0B***stecken
	HBA, HBAx, HBA-qSig	Bitte•***0x0B***HBA•***0x0B***in •***0x0B***SLOT *X*•***0x0B***stecken
	SMC-B	Bitte•***0x0B***SMC-B•***0x0B***in •***0x0B***SLOT *X*•***0x0B***stecken
	sonstiger Kartentyp oder kein explizit angegebener Kartentyp	Bitte•***0x0B***Karte•***0x0B***in •***0x0B***SLOT *X*•***0x0B***stecken
Karte auswerfen	EGK	Bitte•***0x0B***eGK•***0x0B***aus •***0x0B***SLOT *X*•***0x0B***entnehmen
	HBA, HBAx, HBA-qSig	Bitte•***0x0B***HBA•***0x0B***aus •***0x0B***SLOT *X*•***0x0B***entnehmen
	SMC-B	Bitte•***0x0B***SMC-B•***0x0B***aus •***0x0B***SLOT *X*•***0x0B***entnehmen
	sonstiger Kartentyp oder kein explizit	Bitte•***0x0B***Karte•***0x0B***entnehmen

	angegebener Kartentyp	

[<=]

4.1.4.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_4540 - Reaktion auf Dienstbeschreibungspakete
Der Konnektor MUSS Service Announcement für das Auffinden von Kartenterminals entsprechend [SICCT] und [gemSpec_KT] unterstützen. Der Konnektor MUSS Dienstbeschreibungspakete auf UDP Port `CTM_SERVICE_ANNOUNCEMENT_PORT` entgegennehmen.
Erhält er ein solches Dienstbeschreibungspaket, MUSS er

- TUC_KON_054 mit Mode=AutoAdded und IP-Adresse; TCP-Port; MAC-Adresse; Hostname aus dem Dienstbeschreibungspaket, aufrufen
- für das mit der MAC-Adressen in CTM_CT_LIST korrelierende CT-Object, wenn CT.CORRELATION > "bekannt" und CT.VALID_VERSION = true ist, TUC_KON_050 { ctId = CT.CtID; role = „User"} aufrufen.

[<=]

TIP1-A_4541 - Reaktion auf KT-Slot-Ereignis – Karte eingesteckt
Der Kartenterminaldienst MUSS auf SICCT-Ereignisnachrichten „Slot-Ereignis – Karte eingesteckt" ([SICCT#6.1.4.4], TAG ‚84') wie folgt reagieren:

- das meldende Kartenterminal CT in CTM_CT_LIST ermitteln,
- den in der Ereignisnachricht benannten Slot (FU-Nummer) in CT.SLOTS_USED aufnehmen,
- zur weiteren internen Bearbeitung rufe TUC_KON_256 {
  topic = „CT/SLOT_IN_USE";
  eventType = Op;
  severity = Info;
  parameters = („CtID=$CT.CTID,
  SlotNo=<FU-Nummer aus Ereignisnachricht>„);
  doLog = false;
  doDisp = false } auf.

[<=]

TIP1-A_4542 - Reaktion auf KT-Slot-Ereignis – Karte entfernt
Der Kartenterminaldienst MUSS auf SICCT-Ereignisnachrichten „Slot-Ereignis – Karte entfernt" ([SICCT#6.1.4.4], TAG ‚85') wie folgt reagieren:

- das meldende Kartenterminal CT in CTM_CT_LIST ermitteln,
- den in der Ereignisnachricht benannten Slot (FU-Nummer) aus CT.SLOTS_USED entfernen,
- zur weiteren internen Bearbeitung rufe TUC_KON_256 {
  topic = „CT/SLOT_FREE";
  eventType = Op;
  severity = Info;
  parameters = („CtID=$CT.CTID,
  SlotNo==<FU-Nummer aus Ereignisnachricht>„„);
  doLog = false;
  doDisp = false }
  auf.

[<=]

TIP1-A_4543 - KT-Statusanpassung bei Beginn eines Updatevorgangs
Tritt der Event "KSR/UPDATE/START" mit „Target=KT“ ein, MUSS der Konnektor:

- Setze CT = CTM_CT_LIST(CTID-Parameter des Ereignisses)
- CT.CORRELATION für das Kartenterminal merken und auf „aktualisierend“ setzen
- Für jeden in CT.SLOTS_USED gelisteten Slot X zur weiteren internen Bearbeitung TUC_KON_256 {
  topic = „CT/SLOT_FREE“;
  eventType = Op;
  severity = Info;
  parameters = („CtID=$CT.CTID, SlotNo=$CT.SLOTS_USED[X]“);
  doLog = false;
  doDisp = false
  } aufrufen

[<=]

TIP1-A_4544 - KT-Statusanpassung bei Ende eines Updatevorgangs
Tritt der Event "KSR/UPDATE/END" mit „Target=KT“ ein, MUSS der Konnektor:

- Setze CT = CTM_CT_LIST(CTID-Parameter des Ereignisses)
- CT.CORRELATION auf den beim „KSR/UPDATE/START“ gemerkten Wert setzen
- Aktualisiere Gerätedaten durch Aufruf TUC_KON_055 „Befülle CT-Object“ {ctId = CTID}
- Wenn CT.VALID_VERSION = true, Rufe TUC_KON_050 „Beginne Kartenterminalsitzung“ {ctId = CTID; role = „User“}
- Wenn CT.VALID_VERSION = false und CT.CORRELATION = „aktiv“, setze CT.CORRELATION=„gepairt“

[<=]

4.1.4.3 Interne TUCs, nicht durch Fachmodule nutzbar

4.1.4.3.1 TUC_KON_050 „Beginne Kartenterminalsitzung“

TIP1-A_4545-03 - TUC_KON_050 „Beginne Kartenterminalsitzung“
Der Konnektor MUSS den technischen Use Case „Beginne Kartenterminalsitzung” gemäß TUC_KON_050 umsetzen.

Tabelle 33: TAB_KON_039 – TUC_KON_050 „Beginne Kartenterminalsitzung“

Element	Beschreibung
Name	TUC_KON_050 „Beginne Kartenterminalsitzung“
Beschreibung	TUC_KON_050 baut eine TLS-Verbindung vom Konnektor zum Kartenterminal auf und beginnt eine SICCT-Sitzung. Anschließend erfolgt die 2. Authentifizierung des Kartenterminals (Prüfung SharedSecret).

Auslöser	• Neustart des Konnektors • nach dem Setzen eines Kartenterminals auf „aktiv“ • im Rahmen eines erneuten Verbindungsversuchs
Vorbedingungen	ctId ist in CTM_CT_LIST vorhanden
Eingangsdaten	• ctId (zu verbindendes Kartenterminal) • role (Benutzerrolle; gültig sind: „User“ und „Admin“)
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	keine
Nachbedingungen	• TLS-Kanal und SICCT-Session mit gewünschter Benutzerrolle aufgebaut, wenn CT.CORRELATION >= "gepairt" • TLS-Kanal und SICCT-Session mit leerem Username, Password und Session ID aufgebaut, wenn CT.CORRELATION <= „zugewiesen“ • Steck-Ereignisse für alle im KT befindlichen Karten ausgelöst, wenn CT.CORRELATION>=„gepairt“

Standardablauf	Setze CT = CTM_CT_LIST(ctId)
	1. Wenn CT.IS_PHYSICAL = Nein:
	prüfen, ob role = „User“
	Wenn CT.CONNECTED =
	Ja: TUC endet erfolgreich
	Nein:
	- Verbindung zu HSM in Slot 1 aufbauen
	- weiter mit Schritt 9
	2. Wenn CT.CONNECTED = Ja
	prüfen, ob CT.ACTIVEROLE = role
	Ja: TUC endet erfolgreich
	Nein:
	- Schließen der Cardterminal Session mit dem Kartenterminalkommando SICCT CLOSE CT SESSION,
	- weiter ab Schritt 6 (halten der TLS-Verbindung und
	nur Wechsel der Session)
	3. Aufbau einer TLS-Verbindung mit dem Kartenterminal unter Verwendung von ID.SAK.AUT. Dabei Prüfung des KT-Zertifikats mittels
	TUC_KON_037 {
	certificate= C.SMKT.AUT;
	qualifiedCheck=not_required;
	offlineAllowNoCheck=true;
	policyList= oid_smkt_aut;
	intendedKeyUsage= intendedKeyUsage(C.SMKT.AUT);
	intendedExtendedKeyUsage=id-kp-serverAuth;
	validationMode=NONE }
	4. Wenn CT.CORRELATION <=„zugewiesen“:
	a. Öffne eine Cardterminal Session mit dem Kartenterminalkommando SICCT INIT CT SESSION (siehe [SICCT#5.10]) mit leerem Username, Password und Session
	ID
	b. Nur Verbindung in niedriger Korrelation, daher setze CT.CONNECTED = Nein, um fachliche Nutzung des KT zu verhindert
	c. beende TUC erfolgreich
	5. Prüfe, ob das Zertifikat aus der TLS-Verbindung mit den zum
	Kartenterminal gespeicherten Referenzdaten (CT.SMKT_AUT) übereinstimmt.
	a. Läuft das Zertifikat CT.SMKT_AUT (oder C.SMKT.AUT, sie müssen hier identisch sein), dann geht der Konnektor über in den Betriebszustand EC_CardTerminal_SMC-KT_Certificate_Expires_Soon (ctId).
	6. Parallelisierung
	a. Generierung eines zufälligen Werts (Challenge) mit mindestens 16 Byte Länge gemäß [gemSpec_Krypt#2.2] (siehe [gemSpec_KT#DO_KT_0004]),
	b. Öffnen einer Cardterminal Session mit dem Kartenterminalkommando SICCT INIT CT SESSION (siehe

	[SICCT#5.10]) mit
	- ctId als Adressat
	- Wenn role = User
	dann mit leerem Username, Password und Session ID
	Wenn role = „Admin
	dann mit leerer Session ID aber mit CT.ADMIN_USERNAME und CT.ADMIN_PASSWORD
	7. Senden der Challenge mittels Kartenterminalkommando EHEALTH TERMINAL AUTHENTICATE (siehe [gemSpec_KT#3.7.2]) in der Ausprägung VALIDATE mit:
	- Kartenterminal als Empfänger
	- und mit der in Schritt 6a generierten Challenge im Shared Secret Challenge DO
	8. Prüfe Antwort des Kartenterminals, ob sie einen korrekten Hashwert über Challenge und angehängtes CT.SHARED_SECRET gemäß [gemSpec_KT#SEQ_KT_0002] Schritt 4-5 enthält
	9. Setze:
	a. CT.ACTIVEROLE = $role
	b. CT.CONNECTED = Ja
	10 . Wenn TLS-Verbindung neu aufgebaut werden musste, rufe TUC_KON_256 {
	topic = „CT/CONNECTED“;
	eventType = „Op“;
	severity = Info;
	parameters = („CtID=$CT.CTID, Hostname=$CT.HOSTNAME“) }
	11. Ermittle alle im KT gesteckten Karten und befülle entsprechend
	CT.SLOTS_USED
	Für jeden in CT.SLOTS_USED gelisteten Slot X zur weiteren internen
	Bearbeitung TUC_KON_256{
	topic = „CT/SLOT_IN_USE“;
	eventType = Op;
	severity = Info;
	parameters = („CtID=$CT.CTID, SlotNo=$CT.SLOTS_USED[X]“);
	doLog = false;
	doDisp = false }
	rufen.

Varianten/ Alternativen	Keine.

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu:
	Aufruf von TUC_KON_256 {
	topic = "CT/TLS_ESTABLISHMENT_FAILURE";
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („CtID=$CT.ID, Name=$CT.HOSTNAME, Error=$Fehlercode, Bedeutung=$Fehlertext“) }
	Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes
	
	(→1): Admin-Rolle für logische KTs nicht möglich (hätte bei korrekter
	Implementierung nicht stattfinden dürfen), Fehlercode: 4032
	(→1): Verbindungsaufbau zu HSM fehlgeschlagen, Fehlercode: 4032
	(→3): Fehler im TLS-Verbindungsaufbau bzw. Zertifikatsprüfung, Fehlercode: 4028
	und setze CT.CONNECTED auf „Nein“
	(→3): TLS-Verbindung konnte nicht innerhalb von CTM_TLS_HS_TIMEOUT Sekunden aufgebaut werden , Fehlercode: 4028 und setze CT.CONNECTED auf „Nein“
	(→5): Präsentiertes Zertifikat nicht das aus dem Pairing, Fehlercode:
	4029
	und setze CT.CORRELATION auf „gepairt“
	und setze CT.CONNECTED auf „Nein“
	und terminiere TLS-Verbindung
	(→6b): Hinterlegte KT-Admin-Credentials fehlerhaft, Fehlercode: 4030
	und in die User-Session zurückzuwechseln (damit das KT für den normalen Fachbetrieb weiterhin zur Verfügung steht)
	(→8): Prüfung auf Nachweis SharedSecret fehlgeschlagen, Fehlercode
	4029
	und setze CT.CORRELATION auf „gepairt“
	und setze CT.CONNECTED auf „Nein“
	und terminiere TLS-Verbindung

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	Abbildung PIC_KON_110 Aktivitätsdiagramm zu „Beginne Kartenterminalsitzung

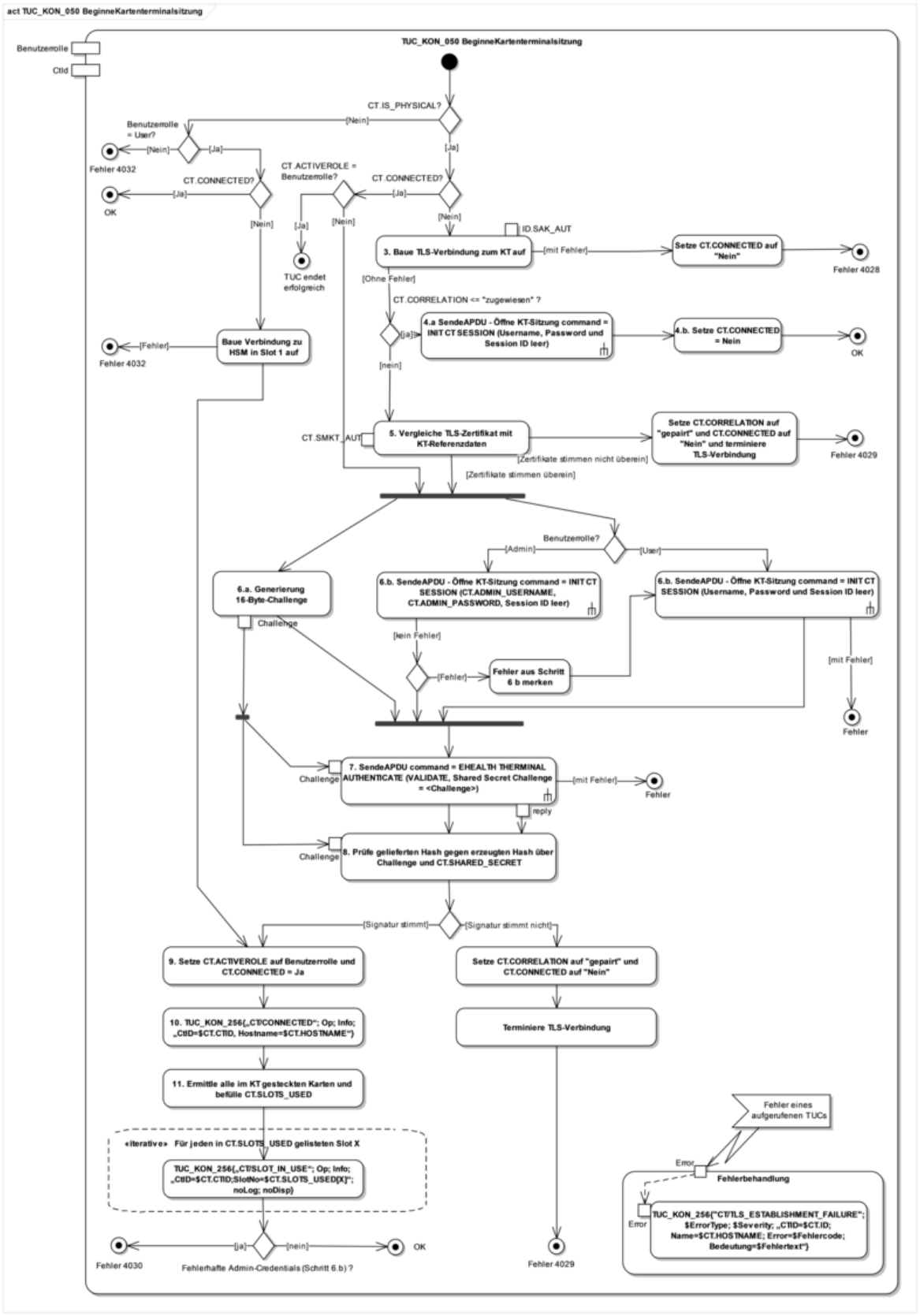


Abbildung 8: PIC_KON_110 Aktivitätsdiagramm zu „Beginne Kartenterminalsitzung"

Tabelle 34: TAB_KON_523 Fehlercodes TUC_KON_050 „Beginne Kartenterminalsitzung“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4028	Technical	Error	Fehler beim Versuch eines Verbindungsaufbaus zu KT
4029	Security	Error	Fehler bei der KT-Authentisierung. KT möglicherweise manipuliert
4030	Security	Error	Admin-Werte für KT fehlerhaft
4032	Technical	Error	Verbindung zu HSM konnte nicht aufgebaut werden

[<=]

4.1.4.3.2 TUC_KON_054 „Kartenterminal hinzufügen“

TIP1-A_4546 - TUC_KON_054 „Kartenterminal hinzufügen“
Der Konnektor MUSS den technischen Use Case TUC_KON_054 „Kartenterminal hinzufügen“ umsetzen.

Tabelle 35: TAB_KON_524 – TUC_KON_054 „Kartenterminal hinzufügen“

Element	Beschreibung
Name	TUC_KON_054 „Kartenterminal hinzufügen“
Beschreibung	Dieser TUC nimmt ein neues Kartenterminal in die zentrale Verwaltung auf (CTM_CT_LIST) oder aktualisiert die Einträge zu einem bereits erfassten Kartenterminal.
Auslöser	• ein empfangenes Dienstbeschreibungspaket ohne vorheriges Service Discovery • manuelles Hinzufügen eines KT-Eintrags • ein empfangenes Dienstbeschreibungspaket nach vorherigem Auslösen eines manuellen Service Discovery
Vorbedingungen	• entweder ist das KT noch nicht in CTM_CT_LIST enthalten • oder das KT ist unter anderer IP/anderem Hostnamen schon gelistet
Eingangsdaten	• Mode (AutoAdded, ManuallyAdded, ManuallyModified) • IP-Adresse (IPv4) • TCP-Port (optional) • MAC-Adresse (optional) • Hostname (optional)

Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	• keine
Nachbedingungen	• Das Kartenterminal ist mit allen Gerätekenndaten in CTM_CT_LIST vorhanden
Standardablauf	1. Sofern optionale Parameter nicht übergeben wurden oder Mode<>AutoAdded, ermittle Port, MAC und Hostname via Service Discovery als UDP/IP-Unicast an IP-Adresse und Port `CTM_SERVICE_DISCOVERY_PORT` 2. Finde CT in CTM_CT_LIST über MAC-Adresse 3. Wenn MAC-Adresse nicht in CTM_CT_LIST: a) Erzeuge neuen CT-Object-Eintrag in CTM_CT_LIST und - Generiere eindeutige CT.CTID - setzte CT.MAC_ADRESS auf MAC-Adresse - Setze CT.CORRELATION =„bekannt“ - Setze CT.CONNECTED = „Nein“ b) Wenn Mode= ManuallyAdded setze CT.CORRELATION =„zugewiesen“ 4. Wenn CT.CONNECTED = Ja und (IP-Adresse <> CT.IP_ADRESS oder TCP-Port <> CT.TCP_PORT), beende TLS-Verbindung und setze CT.CONNECTED = „Nein“ 5. Befülle: CT.IP_ADRESS, CT.Hostname, CT.TCP_PORT 6. Wenn CT.CORRELATION>=„zugewiesen“ rufe TUC_KON_055 „Befülle CT-Object“
Varianten/ Alternativen	Keine

Fehlerfälle	Fehler im Ablauf (Standardablauf oder Varianten) führen in den folgend ausgewiesenen Schritten zu einem Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes:
	
	(→1) Keine Antwort innerhalb `CTM_SERVICE_DISCOVERY_TIMEOUT`, Fehlercode: 4033
	(→1) Ermittelte MAC-Adresse weicht von übergebener MAC-Adresse ab, Fehlercode: 4035
	(→1) Ermittelter Hostname-Adresse weicht von übergebenem Hostname ab, Fehlercode: 4036
	(→2) Wenn Mode=ManuallyModified und nicht gefunden, Fehlercode: 4037
	Zusätzlich im Abbruchfall:
	- Aufruf von TUC_KON_256 {
	topic = "CT/CT_ADDING_ERROR";
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („IP=$IP-Adresse, Name=$HOSTNAME, Error=$Fehlercode, Bedeutung=$Fehlertext“) }

	- Keine Veränderung an CTM_CT_LIST
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	keine

Tabelle 36: TAB_KON_525 Fehlercodes TUC_KON_054 „Kartenterminal hinzufügen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4033	Technical	Error	Kartenterminal antwortet nicht, Zufügen fehlgeschlagen
4035	Technical	Error	Angegebener IP-Adresse gehört zu einer anderen MAC-Adresse als die, die übergeben wurde. Angaben zur MAC prüfen
4036	Technical	Error	Angegebener IP-Adresse gehört zu einem anderen Hostname als der, der übergeben wurde. Angaben zum Hostname prüfen
4037	Technical	Error	Verwaltung der Kartenterminals inkonsistent

[<=]

4.1.4.3.3 TUC_KON_053 „Paire Kartenterminal"

TIP1-A_4548-02 - TUC_KON_053 „Paire Kartenterminal"
Der Konnektor MUSS den technischen Use Case „Paire Kartenterminal" gemäß TUC_KON_053 umsetzen.

Tabelle 37: TAB_KON_041 – TUC_KON_053 „Paire Kartenterminal"

Element	Beschreibung
Name	TUC_KON_053 „Paire Kartenterminal"
Beschreibung	TUC_KON_053 führt das Pairing zwischen dem Konnektor und einem eHealth-Kartenterminal durch.
Auslöser	Dialoge zur Administration des Konnektors. Der Administrator hat ein Kartenterminal im Dialog der Managementschnittstelle ausgewählt und das Pairing aufgerufen.
Vorbedingungen	• KT ist in CTM_CT_LIST vorhanden • CT.CORRELATION = „zugewiesen" • CT.IS_PHYSICAL = Ja

Eingangsdaten	• ctId
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	• Keine
Nachbedingungen	- CT.CORRELATION = „aktiv“, wenn Pairing erfolgreich - CT.CORRELATION = „zugewiesen“, wenn Pairing nicht erfolgreich - CT.CONNECTED = „Ja“, wenn Pairing erfolgreich
Standardablauf	Setze CT = CTM_CT_LIST(ctId) 1. Prüfe `CT.VALID_VERSION = true` 2. Aufbau einer TLS-Verbindung mit dem Kartenterminal unter Verwendung von ID.SAK.AUT. Dabei: a. Speichern des präsentierten KT-Zertifikats in CT.SMKT_AUT b. Prüfung des KT-Zertifikats mittels TUC_KON_037{ certificate = C.SMKT.AUT; qualifiedCheck = not_required; offlineAllowNoCheck = true; policyList = oid _smkt_aut; intendedKeyUsage= intendedKeyUsage(C.SMKT.AUT) ; intendedExtendedKeyUsage = id-kp-serverAuth; validationMode = NONE } 3. Der Konnektor entnimmt den Fingerprint dem KT-Zertifikat und stellt dies dem Administrator im Dialog der Managementschnittstelle dar. Der Konnektor fordert den Administrator auf, den Fingerprint zu akzeptieren oder zurückzuweisen. 4. Wenn der Administrator den Fingerprint bestätigt, a. generiert der Konnektor einen neuen Schlüssel, das Shared Secret ShS.KT.AUT gemäß [gemSpec_Krypt#2.2] (siehe [gemSpec_KT#3.7]) und speichert es in `CT.SHARED_SECRET` b. und eröffnet der Konnektor mit dem Kartenterminalkommando SICCT INIT CT SESSION (siehe [SICCT#5.10]) mit - ctId als Adressat - und mit leerem Username, Password und Session ID eine Cardterminal Session. 5. Der Konnektor sendet mittels Kartenterminalkommando EHEALTH TERMINAL AUTHENTICATE (siehe [gemSpec_KT#3.7.2]) in der Ausprägung CREATE mit - ctId als Empfänger - und mit dem in Schritt 4.a generierten Schlüssel im Shared Secret DO und der Display Message „`KT:$CT.MAC_ADRESS MIT`

	KON:$MGM_KONN_HOSTNAME PAIREN OK?“, wobei die MAC-Adresse mit Trenner im folgenden Format dargestellt werden MUSS: „AABBCC:DDEEFF“ das Shared Secret an das Kartenterminal. 6. Der Konnektor prüft ob in der Antwort des Kartenterminals eine korrekte Signatur des Shared Secrets gemäß [gemSpec_KT#SEQ_KT_0001] Schritt 7, ausgeführt mit dem Schlüssel zum Zertifikat CT.SMKT_AUT vorliegt. 7. CT.CORRELATION wird auf „gepairt“ gesetzt 8. TLS-Verbindung, die zum Pairen diente, beenden und zuvor das Kartenterminalkommando SICCT CLOSE CT SESSION mit ctId als Adressat senden 9. Automatischer Zustandsübergang CT.CORRELATION = „gepairt“ nach „aktiv“ (implizite Aktion des Administrators durch Aufruf von TUC_KON_053). 10. „Arbeits“-TLS-Verbindung neu aufbauen durch Aufruf TUC_KON_050 { ctId; role = „User“}
Varianten/ Alternativen	(→4): weist der Administrator den Fingerprint in Schritt 3 ab, wird TUC_KON_053 nach Ausführung folgender Aktivitäten beendet: 4.1.a) Löschen von CT.SMKT_AUT 4.1.b) Abbau der TLS-Verbindung, Setzen von CT.CONNECTED auf „Nein“
Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu: a) Aufruf von TUC_KON_256 { topic = "CT/ERROR"; eventType = $ErrorType; severity = $Severity; parameters = („CtID=$ctId, Name=$CT.HOSTNAME, Error=$Fehlercode, Bedeutung=$Fehlertext“); doDisp = false } b) Löschen von CT.SMKT_AUT, CT.SHARED_SECRET c) Direkte Anzeige der Fehlermeldung für den Administrator d) Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes (→1) Version des KT wird nicht unterstützt, Fehlercode: 4042 (→2b) Zertifikat ist zeitlich nicht gültig, Fehlercode: 1021 (CERTIFICATE_NOT_VALID_TIME) (→2) Fehler im TLS Verbindungsaufbau bzw. Zertifikatsprüfung, Fehlercode: 4040 (→4b) Fehler in SICCT INIT CT SESSION, Fehlercode: 4041 mit Angabe des SICCT-Fehlers (→5) Fehler in EHEALTH TERMINAL AUTHENTICATE, Fehlercode: 4041 mit Angabe des SICCT-Fehlers (→6) Signaturprüfung fehlgeschlagen, Fehlercode: 4041

Zugehörige Diagramme	Siehe PIC_KON_057

Tabelle 38: TAB_KON_113 Fehlercodes TUC_KON_053 „Paire Kartenterminal"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4040	Security	Error	Fehler beim Versuch eines Verbindungsaufbaus zum KT
4041	Technical	Error	Fehler im Pairing, SICCT-Fehler(Nur wenn dieser Fehler wegen eines Fehlers auf der SICCT-Schnittstelle auftritt, ist der SICCT-Fehlercode mit anzugeben.): <SICCT-Fehler>
4042	Technical	Error	Die Version des Kartenterminals wird nicht unterstützt

Hinweis zu Fehler 4041:
Nur wenn dieser Fehler wegen eines Fehlers auf der SICCT-Schnittstelle auftritt, ist der SICCT-Fehlercode mit anzugeben.

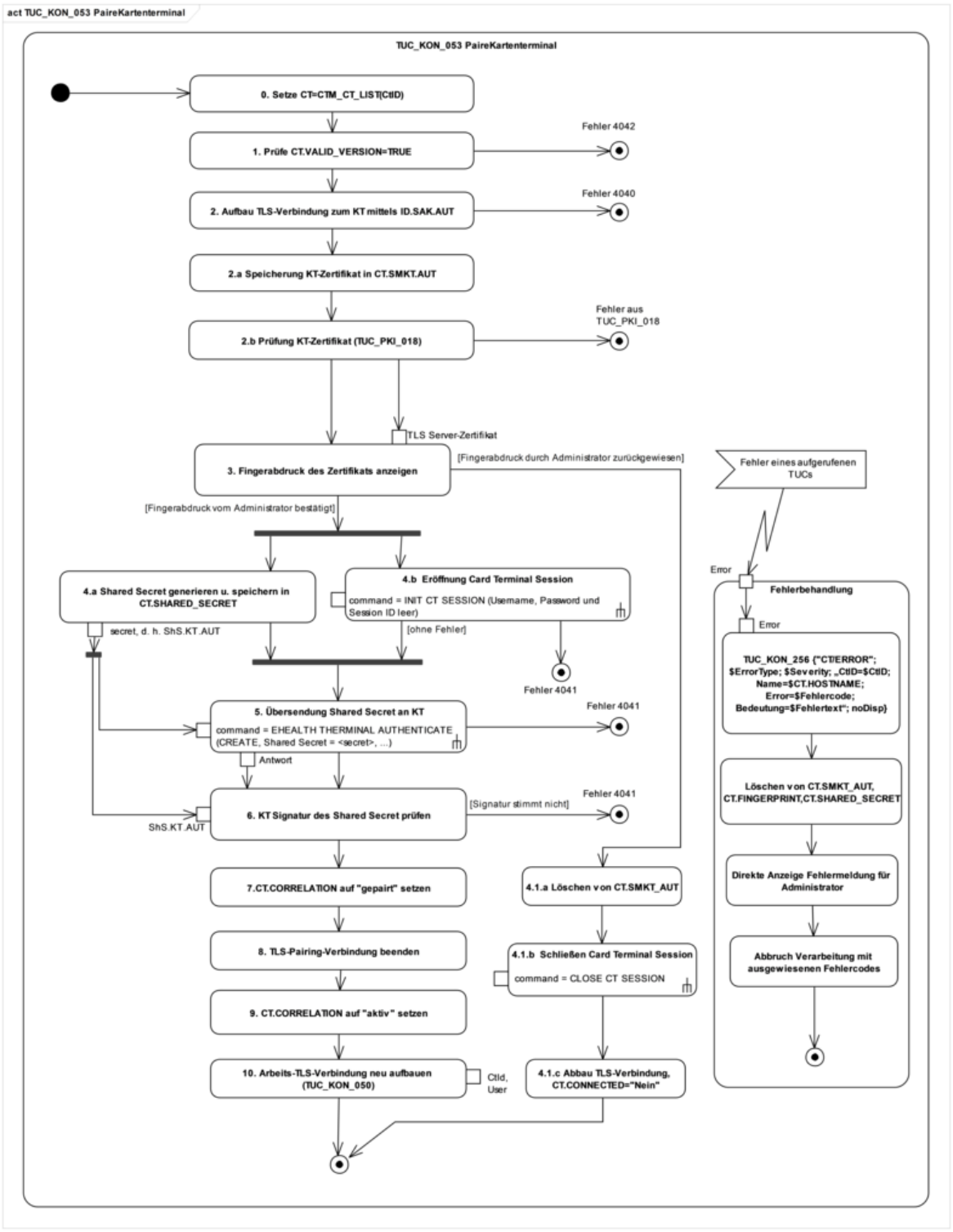


Abbildung 9: PIC_KON_057 Aktivitätsdiagramm zu „PaireKartenterminal"

[<=]

4.1.4.3.4 TUC_KON_055 „Befülle CT-Object"

TIP1-A_4985 - TUC_KON_055 „Befülle CT-Object"
Der Konnektor MUSS den technischen Use Case TUC_KON_055 „Befülle CT-Object" umsetzen.

Tabelle 39: TAB_KON_526 – TUC_KON_055 „Befülle CT-Object"

Element	Beschreibung
Name	TUC_KON_055 „Befülle CT-Object"
Beschreibung	Dieser TUC befüllt ein vorhandenes CT-Object aus CTM_CT_LIST mit den aktuellen Produktinformationen, die vom Kartenterminal bezogen werden und prüft, ob die Version des Kartenterminals unterstützt wird.
Auslöser	• TUC_KON_054 • Ereignis „KSR/UPDATE/END" mit „Target=KT" • Verändern von CT.CORRELATION auf „zugewiesen" • Administratoraktion
Vorbedingungen	• ctId ist in CTM_CT_LIST vorhanden
Eingangsdaten	• ctId
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	• Supported (Boolean, True, wenn die Version des KT unterstützt wird)
Nachbedingungen	• Die Gerätekenndaten des Kartenterminals in CTM_CT_LIST sind aktualisiert
Standardablauf	Setze CT = CTM_CT_LIST(ctId) 1. Wenn CT.CONNECTED=Nein: Rufe TUC_KON_050 { ctId, role = „User" } 2. Folgende CT.Werte via SICCT GET STATUS ermitteln und befüllen: • CT.SLOTCOUNT • CT.PRODUCTINFORMATION • CT.EHEALTH_INTERFACE_VERSION (Element VER aus Discretionary Data Data Object (DD DO)) • CT.DISPLAY_CAPABILITIES (aus Display Capabilities Data Object in [SICCT#5.5.10.17]) 3. Finde Eintrag in `CTM_SUPPORTED_KT_VERSIONS` anhand der ersten beiden Stellen (Haupt- und Nebenversionsnummer) aus CT.EHEALTH_INTERFACE_VERSION Eintrag gefunden: Die dritte Stelle der KT-Version ist im Vergleich zur dritten Stelle im gefundenen

	Eintrag: >=: Setze Result = True <: Setze Result = False Kein Eintrag gefunden: Setze Result = False 4. Setze CT.VALID_VERSION auf Result 5. Wenn Verbindung in (1) aufgebaut wurde, trenne Verbindung 6. Liefere Result zurück
Varianten/ Alternativen	(->5) Wenn CT.CORRELATION="aktiv", kann die in (1) aufgebaute Verbindung bestehen bleiben.
Fehlerfälle	-> 2) Kartenterminal antwortet mit einer spezifischen Fehlermeldung, Fehlercode <gemäß [SICCT]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	keine

[<=]

4.1.4.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.4.4.1 TUC_KON_051 „Mit Anwender über Kartenterminal interagieren"

TIP1-A_4547 - TUC_KON_051 „Mit Anwender über Kartenterminal interagieren"
Der Konnektor MUSS den technischen Use Case „Mit Anwender über Kartenterminal interagieren" gemäß TUC_KON_051 umsetzen.

Tabelle 40: TAB_KON_112 – TUC_KON_051 „Mit Anwender über Kartenterminal interagieren"

Element	Beschreibung
Name	TUC_KON_051 „Mit Anwender über Kartenterminal interagieren"
Beschreibung	Der TUC ermöglicht es, Meldungen an das Display eines Kartenterminals zu senden oder Eingaben vom Benutzer über das PIN-Pad eines Kartenterminals abzufragen (unter Anzeige einer Meldung).
Auslöser	Fachmodul im Konnektor oder anderer technischer Use Case ruft diesen Use Case auf.
Vorbedingungen	• KT ist in CTM_CT_LIST vorhanden • CT.CORRELATION = „aktiv" • CT.CONNECTED = Ja • CT.IS_PHYSICAL = Ja
Eingangsdaten	• ctId (Kartenterminalidentifikator)

	• displayMessage – *optional/nicht erforderlich bei opmode= OutputErase, sonst mandatory* (Text zur Darstellung am KT, Länge durch KT begrenzt); • opMode [KtOutputMode] (Kommando-Modus) • inputLength – *optional/nur bei opMode=Input* (erwartete Eingabelänge, 0 für „beliebig" lang) • waitTimer – *optional/nur bei opMode=OutputWait* (Wartezeit bis zur ersten Eingabe in Sekunden)
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	• opResult [OK | ABBRUCH] – *optional/verpflichtend, wenn opMode=Input oder opMode=OutputConfirm* (Nutzertastendruck) • inputData – *optional/nur bei opMode = Input* (Zifferneingabe des Benutzers)
Nachbedingungen	Wenn Mode=OutputKeep bleibt Data auf dem Display des KT stehen
Standardablauf	Setze CT = CTM_CT_LIST(ctId) 1. opMode= a. <u>Input:</u> Der Konnektor MUSS via SICCT INPUT am CT zur Eingabe auffordern. Dabei MUSS die KT-Ansteuerung so erfolgen, dass: • displayMessage zur Anzeige gebracht wird • maximal inputLength Ziffern akzeptiert werden. Bei inputLength=0 werden 1-n Zeichen akzeptiert (n=Maximalwert, definiert durch KT) • die eingegebenen Zeichen am Display angezeigt werden • die Eingabe explizit mit OK oder ABBRUCH beendet werden muss b. <u>OutputWait:</u> Der Konnektor MUSS via SICCT OUTPUT am CT displayMessage zur Anzeige bringen. Nach einer Wartezeit von waitTimer Sekunden MUSS der Konnektor die Anzeige des KT leeren. c. <u>OutputConfirm:</u> Der Konnektor MUSS via SICCT INPUT am CT displayMessage zur Anzeige bringen und auf eine Bestätigung durch den Nutzer warten (zulässig: OK, ABBRUCH) d. <u>OutputKeep:</u> Der Konnektor MUSS via SICCT OUTPUT am CT displayMessage zur Anzeige bringen. Die Anzeige

	bleibt erhalten, bis das KT neue Informationen am Display darstellen muss/soll. e. OutputErase: Der Konnektor MUSS via SICCT OUTPUT am CT die Anzeige leeren.
Varianten/ Alternativen	• Ist das Kartenterminal-Display durch einen anderen, zeitgleich im Konnektor ablaufenden Vorgang reserviert, so muss der Konnektor zunächst maximal 10 Sekunden lang versuchen, Zugriff auf das Display zu erhalten (und somit parallele Zugriffe auf das Display zu serialisieren). Erst nach Ablauf der Wartezeit wird Fehler 4039 geworfen.

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zum Aufruf von TUC_KON_256 {
	topic = "CT/ERROR";
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („CtID=$ctId, Name=$CT.HOSTNAME, Error=$Fehlercode, Bedeutung=$Fehlertext“)
	}
	
	(→1) Display und PinPad des Kartenterminals sind aktuell belegt (PIN, Eingabe, andere Ausgabe etc.), Fehlercode: 4039
	(→1) Kartenterminal antwortet mit einer spezifischen Fehlermeldung, Fehlercode <gemäß [SICCT]>

Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 41: TAB_KON_114 Fehlercodes TUC_KON_051 „Mit Anwender über Kartenterminal interagieren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4039	Technical	Error	Kartenterminal durch andere Nutzung aktuell belegt

[<=]

4.1.4.4.2 TUC_KON_056 „Karte anfordern“

TIP1-A_5409 - TUC_KON_056 „Karte anfordern“
Der Konnektor MUSS den technischen Use Case „Karte anfordern” gemäß TUC_KON_056 umsetzen.

Tabelle 42: TAB_KON_723 - TUC_KON_056 „Karte anfordern"

Element	Beschreibung
Name	TUC_KON_056 „Karte anfordern"
Beschreibung	Der TUC ermöglicht es, die Aufforderung zum Karte-Stecken an das Kartenterminal zu senden und dabei eine Meldung zum Anzeigen im Display des Kartenterminals mitzugeben.
Auslöser	Fachmodul im Konnektor oder Operation RequestCard ruft diesen Use Case auf.
Vorbedingungen	
Eingangsdaten	- ctId (Kartenterminalidentifikator) - slotId (Nummer des Kartenslots) - cardType - *optional* - displayMessage – *optional* (Text zur Darstellung am Kartenterminal, Länge durch KT begrenzt) - timeOut (Wartezeit in Sekunden)
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	• cardObject (Informationsobjekt der Karte)
Nachbedingungen	Im Erfolgsfall enthält die CM_CARD_LIST ein neues CARD-Objekt des geforderten Typs.
Standardablauf	Setze CT = CTM_CT_LIST(ctId) 1. Falls displayMessage nicht explizit angegeben ist, MUSS sie gemäß Anforderung [TIP1-A_5408] erstellt werden. 2. Der Konnektor MUSS das Kommando `SICCT REQUEST ICC` an das Kartenterminal CT senden. Die verfügbaren Optionen (`Optical Signal, Acoustic Signal`) können herstellerspezifisch sein bzw. über die Konfigurationsschnittstelle des Konnektors eingestellt werden. displayMessage wird als Eingabeaufforderung mitgegeben. Der übergebene timeOut-Wert wird dem SICCT-Kommando als Parameter übergeben.

	3. Falls die Karte noch nicht gesteckt war, wird durch das Stecken der Karte das Ereignis „Karte gesteckt“ ausgelöst, worauf der Konnektor reagiert [TIP1-A_4563]. 4. Die Verarbeitung wird fortgesetzt, wenn eines der Ereignisse eingetreten ist: a. SICCT REQUEST ICC kehrt mit '6201' zurück (eine aktivierte Karte steckte bereits) b. SICCT REQUEST ICC kehrt mit '9000' oder '9001' zurück und das Ereignis "Karte gesteckt" wurde gemäß [TIP1-A_4563] verarbeitet c. SICCT REQUEST ICC kehrt mit '9000' oder '9001' zurück und das Ereignis "Karte gesteckt" wurde nicht empfangen (eine deaktivierte Karte steckte bereits), die Karte wurde durch Rufe TUC_KON_001 { ctId; slotId } geöffnet. In allen Fällen liegt in CM_CARD_LIST ein neues CARD-Objekt vor. 5. Falls cardType angegeben ist, wird nach erfolgreicher Ausführung von SICCT REQUEST ICC der AID des MF der gesteckten Karte gelesen und mit dem gewünschten Kartentyp in cardType verglichen. Bei fehlender Übereinstimmung wird der Ablauf mit dem Fehler 4051 abgebrochen. 6. Es wird cardObject (ein Informationsobjekt der Karte, die sich in dem Slot mit der Nummer slotId befindet) zurückgegeben.
Varianten/ Alternativen	Die Ausgabe einer Display-Nachricht entfällt, wenn der adressierte Slot bereits eine gesteckte Karte enthält.

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu Aufruf von TUC_KON_256 {
	topic = "CT/ERROR";
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („CtID=$ctId, Name=$CT.HOSTNAME, Error=$Fehlercode, Bedeutung=$Fehlertext“) }
	
	(→2) Display des Kartenterminals ist aktuell belegt, Fehlercode: 4039
	(→2) Fehler beim Zugriff auf das Kartenterminal, Fehlercode: 4044
	(→2) Ungültige Kartenterminal-ID: Fehlercode: 4007
	(→2) Ungültige Kartenslot-ID: Fehlercode: 4097
	(→2) Kartenterminal nicht aktiv, Fehlercode: 4221
	(→2) Kartenterminal ist nicht verbunden, Fehlercode: 4222
	(→2) Kartenterminal antwortet mit einer spezifischen

	Fehlermeldung, Fehlercode <gemäß [SICCT]> (→4) Timeout. Es wurde keine Karte innerhalb der angegebenen Zeitspanne gesteckt, Fehlercode: 4202 (→5) Falscher Kartentyp, Fehlercode: 4051
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 43: TAB_KON_724 Fehlercodes TUC_KON_056 „Karte anfordern"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4039	Technical	Error	Kartenterminal durch andere Nutzung aktuell belegt
4044	Technical	Error	Fehler beim Zugriff auf das Kartenterminal
4051	Technical	Error	Falscher Kartentyp
4007	Technical	Error	Ungültige Kartenterminal-ID
4097	Technical	Error	Ungültige Kartenslot-ID
4202	Technical	Error	Timeout. Es wurde keine Karte innerhalb der angegebenen Zeitspanne gesteckt.
4221	Technical	Error	Kartenterminal nicht aktiv
4222	Technical	Error	Kartenterminal ist nicht verbunden

[<=]

4.1.4.4.3 TUC_KON_057 „Karte auswerfen"

TIP1-A_5410 - TUC_KON_057 „Karte auswerfen"
Der Konnektor MUSS den technischen Use Case „Karte auswerfen" gemäß TUC_KON_057 umsetzen.

Tabelle 44: TAB_KON_725 – TUC_KON_057 „Karte auswerfen"

Element	Beschreibung
Name	TUC_KON_057 „Karte auswerfen"
Beschreibung	Der TUC ermöglicht es, das SICCT-Kommando zum Auswerfen der Karte an das Kartenterminal zu senden und dabei eine Meldung zum Anzeigen im Display des Kartenterminals

	mitzugeben, die den Benutzer zum Entnehmen der Karte auffordert.
Auslöser	Fachmodul im Konnektor oder Operation EjectCard ruft diesen Use Case auf.
Vorbedingungen	
Eingangsdaten	• ctId (Kartenterminalidentifikator) • slotId (Nummer des Kartenslots) • displayMessage – *optional* (Text zur Darstellung am Kartenterminal, Länge durch KT begrenzt) • timeOut (Wartezeit in Sekunden)
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	keine
Nachbedingungen	Durch das Entfernen der Karte wird das Ereignis „Karte entfernt" ausgelöst, worauf der Konnektor reagiert [TIP1-A_4562].
Standardablauf	Setze CT = CTM_CT_LIST(ctId) 1. Der Konnektor prüft, dass entweder die Karte nicht reserviert ist oder der Aufrufer im Besitz des Karten-Locks ist. 2. Falls displayMessage nicht explizit angegeben ist, MUSS sie gemäß Anforderung [TIP1-A_5408] erstellt werden. 3. Der Konnektor MUSS das Kommando `SICCT EJECT ICC` an das Kartenterminal CT senden. Der Aufruf MUSS mit der Option „Delivery: Mechanical Throwout" erfolgen. Die anderen Optionen (Optical Signal, Acoustic Signal) können herstellerspezifisch eingestellt werden bzw. können über die Konfigurationsschnittstelle des Konnektors eingestellt werden. Der übergebene Wert timeOut wird dem SICCT-Kommando als Parameter übergeben.
Varianten/ Alternativen	Auch im Falle, dass nach der internen Buchführung des Konnektors in dem angegebenen Slot des Kartenterminals keine Karte steckt, MUSS der Konnektor das SICCT-Kommando SICCT EJECT ICC an das Kartenterminal senden. Meldet das Kartenterminal keinen Fehler, so meldet auch der Konnektor keinen Fehler und es kann davon ausgegangen werden, dass sich keine Karte mehr in dem Slot befindet.

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu Aufruf von TUC_KON_256 {
	topic = "CT/ERROR";
	eventType = $ErrorType;

	severity = $Severity; parameters = („CtID=$CtID, Name=$CT.HOSTNAME, Error=$Fehlercode, Bedeutung=$Fehlertext“) } (→1) Die Karte ist fremdreserviert, Fehlercode 4093 (→3) Display des Kartenterminals ist aktuell belegt, Fehlercode: 4039 (→3) Fehler beim Zugriff auf das Kartenterminal, Fehlercode: 4044 (→3) Karte deaktiviert, aber nicht entnommen, Fehlercode: 4203 (→3) Ungültige Kartenterminal-ID: Fehlercode: 4007 (→3) Ungültige Kartenslot-ID: Fehlercode: 4097 (→3) Kartenterminal nicht aktiv, Fehlercode: 4221 (→3) Kartenterminal ist nicht verbunden, Fehlercode: 4222 (→3) Kartenterminal antwortet mit einer spezifischen Fehlermeldung, Fehlercode <gemäß [SICCT]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 45: TAB_KON_796 Fehlercodes TUC_KON_057 „Karte auswerfen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4039	Technical	Error	Kartenterminal durch andere Nutzung aktuell belegt
4044	Technical	Error	Fehler beim Zugriff auf das Kartenterminal
4203	Technical	Error	Karte deaktiviert, aber nicht entnommen
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4007	Technical	Error	Ungültige Kartenterminal-ID
4097	Technical	Error	Ungültige Kartenslot-ID
4221	Technical	Error	Kartenterminal nicht aktiv
4222	Technical	Error	Kartenterminal ist nicht verbunden

[<=]

4.1.4.4.4 TUC_KON_058 „Displaygröße ermitteln“

A_17473 - TUC_KON_058 „Displaygröße ermitteln“
Der Konnektor MUSS den technischen Use Case „Displaygröße ermitteln“ gemäß TUC_KON_058 umsetzen.

Tabelle 46: TAB_KON_854 – TUC_KON_058 „Displaygröße ermitteln“

Element	Beschreibung
Name	TUC_KON_058 „Displaygröße ermitteln“
Beschreibung	Der TUC liefert den Inhalt der Variable CT.DISPLAY_CAPABILITIES zurück.
Auslöser	Fachmodul im Konnektor ruft diesen Use Case auf.
Vorbedingungen	
Eingangsdaten	• ctId (Kartenterminalidentifikator)
Komponenten	Konnektor
Ausgangsdaten	CT.DISPLAY_CAPABILITIES
Nachbedingungen	Keine
Standardablauf	Setze CT = CTM_CT_LIST(ctId) 1. Der Konnektor prüft, ob CT.DISPLAY_CAPABILITIES belegt ist. 2. Falls CT.DISPLAY_CAPABILITIES belegt ist, wird der Inhalt der Variable zurückgegeben.
Varianten/ Alternativen	Keine

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu Aufruf von TUC_KON_256 {
	topic = "CT/ERROR";
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („CtID=$CtID, Name=$CT.HOSTNAME, Error=$Fehlercode, Bedeutung=$Fehlertext“) }
	
	(→2) CT.DISPLAY_CAPABILITIES ist nicht belegt, Fehlercode 4254

Nichtfunktionale Anforderungen	Keine

Zugehörige Diagramme	Keine

Tabelle 47: TAB_KON_855 Fehlercodes TUC_KON_058 „Displaygröße ermitteln"

Fehlercode	ErrorType	Severity	Fehlertext
4254	Technical	Error	Keine Displaygröße für das Kartenterminal definiert

[<=]

4.1.4.5 Operationen an der Außenschnittstelle

TIP1-A_5411-02 - Basisdienst Kartenterminaldienst
Der Konnektor MUSS Clientsystemen den Basisdienst Kartenterminaldienst anbieten.

Tabelle 48: TAB_KON_722 Basisdienst Kartenterminaldienst

Name	CardTerminalService	
Version (KDV)	1.1.0 (WSDL-Version) 1.1.2 (XSD-Version)	
Namensraum	Siehe GitHub	
Namensraum-Kürzel	CT für Schema und CTW für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	RequestCard	Karte anfordern
	EjectCard	Karte auswerfen
WSDL	CardTerminalService.wsdl	
Schema	CardTerminalService.xsd	

[<=]

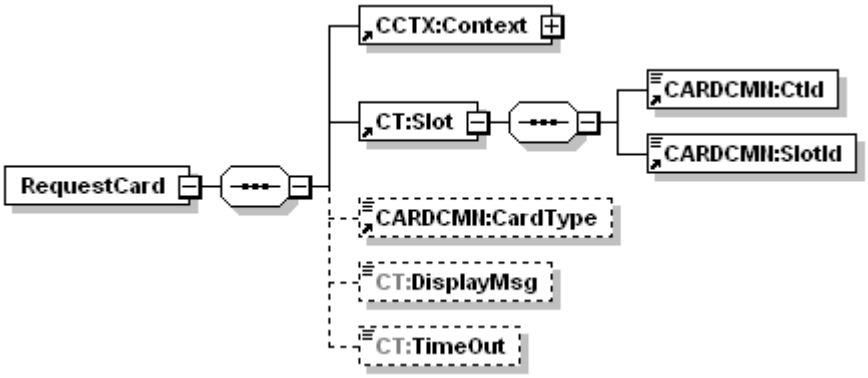
4.1.4.5.1 RequestCard

TIP1-A_5412 - Operation RequestCard
Der Konnektor MUSS an der Außenschnittstelle eine Operation RequestCard, wie in Tabelle TAB_KON_716 Operation RequestCard beschrieben, anbieten.

Tabelle 49: TAB_KON_716 Operation RequestCard

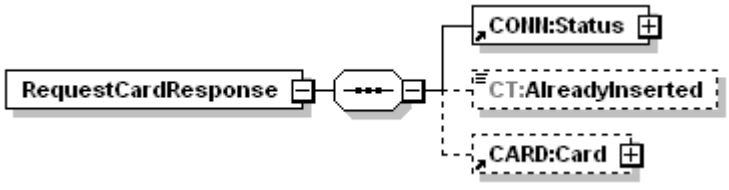
Name	**RequestCard**
Beschreibung	Liefert die Information zu einer Karte, die in dem Slot eines Kartenterminals steckt oder innerhalb einer bestimmten Zeit (Timeout) gesteckt wird.

Aufruf-parameter



Name	Beschreibung
`CCTX:Context`	MandantId, CsId, WorkplaceId verpflichtend
`CT:Slot`	Adressiert den Slot eines Kartenterminals über die Identifikation des Kartenterminal `CARDCMN:CtId` und die Nummer des Slots `CARDCMN:SlotId`
`CARDCMN: CardType`	Ein Kartentyp aus {EGK, KVK, HBAx, SM-B} als optionaler Filter. Wenn angegeben, werden nur Karten vom spezifizierten Typ zurückgegeben.
`CT:DisplayMsg`	Diese Nachricht wird am Display des Kartenterminals angezeigt, um den Nutzer zum Stecken der Karte aufzufordern.
`CT:TimeOut`	Die Zeit in sec, die maximal gewartet wird bis zum Stecken einer Karte. Wird dieser Parameter nicht übergeben, SOLL der Konnektor den Wert 20 sec verwenden. Optional KANN dieser Default-Wert im Konnektor konfigurierbar sein.

Rückgabe



Name	Beschreibung
`CONN:Status`	Enthält den Ausführungsstatus der Operation (siehe 3.5.2)
`CT:AlreadyInserted`	Dieses optionale Flag gibt an, ob die Karte bereits vor der Anfrage steckte (Wert true) oder erst auf Anforderung dieses Aufrufs gesteckt wurde (Wert false oder Element nicht vorhanden).

	CARD:Card	Falls eine Karte gesteckt ist, werden Information zur Karte zurückgegeben (siehe 4.1.6.5.2)
Vorbedingung	keine	
Nachbedingung	keine	

Der Ablauf der Operation RequestCard ist in Tabelle TAB_KON_717 Ablauf RequestCard beschrieben.

Tabelle 50: TAB_KON_717 Ablauf RequestCard

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab. Wird die Operation für einen nicht unterstützten Kartentypen aufgerufen, so bricht die Operation mit Fehler 4058 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		ctId = $Slot.CtId;
		needCardSession=false;
		allWorkplaces=false }
		Tritt bei der Prüfung ein Fehler auf, so bricht die Operation mit dem Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_056 „Karte anfordern“	Anforderung der Karte vom Kartenterminal durch Aufruf TUC_KON_056(
		ctId = $Slot.CtId;
		slotId = $Slot.SlotId;
		$cardType;
		displayMessage = $DisplayMsg;
		$timeOut)

Tabelle 51: TAB_KON_718 Fehlercodes „RequestCard“

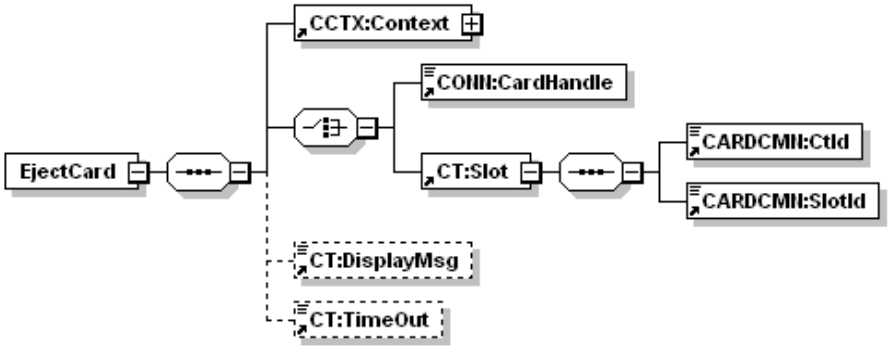
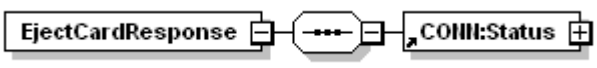
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4058	Security	Error	Aufruf nicht zulässig

[<=]

4.1.4.5.2 EjectCard

TIP1-A_5413 - Operation EjectCard
Der Konnektor MUSS an der Außenschnittstelle eine Operation EjectCard, wie in Tabelle TAB_KON_719 Operation EjectCard beschrieben, anbieten.

Tabelle 52: TAB_KON_719 Operation EjectCard

<table>
<tr><td>Name</td><td colspan="2">EjectCard</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Beendet die Kommunikation mit der Karte und wirft sie aus, falls das Kartenterminal eine solche mechanische Funktion hat.</td></tr>
<tr><td>Aufruf-parameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>Context</td><td>MandantId, CsId, WorkplaceId verpflichtend</td></tr>
<tr><td></td><td>CONN:CardHandle</td><td>Adressiert die Karte, die ausgeworfen werden soll.</td></tr>
<tr><td></td><td>CT:Slot</td><td>Adressiert alternativ den Slot eines Kartenterminals, aus dem die Karte ausgeworfen werden soll. Die Adressierung erfolgt über die Identifikation des Kartenterminal CARDCMN:CtId und die Nummer des Slots CARDCMN:SlotId.</td></tr>
<tr><td></td><td>CT:DisplayMsg</td><td>Diese Nachricht wird am Display des Kartenterminals angezeigt, um den Nutzer zum entnehmen der Karte aufzufordern.</td></tr>
<tr><td></td><td>CT:TimeOut</td><td>Die Zeit in msec, die maximal gewartet wird bis eine Karte gezogen ist. Wird dieser Parameter nicht übergeben, SOLL der Konnektor den Wert 20 sec verwenden. Optional KANN dieser Default-Wert im Konnektor konfigurierbar sein.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
</table>

	Name	Beschreibung
	`Status`	Enthält den Ausführungsstatus der Operation (siehe 3.5.2)
Vorbedingung	keine.	
Nachbedingung	keine.	

Der Ablauf der Operation EjectCard ist in Tabelle TAB_KON_720 Ablauf EjectCard beschrieben.

Tabelle 53: TAB_KON_720 Ablauf EjectCard

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Ist $cardHandle vorgegeben, so wird $ctId als Id des Kartenterminals bestimmt, in dem die Karte steckt.
		Prüfung der Zugriffsberechtigung durch den Aufruf
		TUC_KON_000 {
		mandantId = $Context.MandantId;
		clientSystemId = $Context.ClientSystemId;
		workplaceId = $Context.WorkplaceId;
		ctId = $Slot.CtId
		bzw. ctId = CM_CARD_LIST($CardHandle).CTID;
		needCardSession = false;
		allWorkplaces = false }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_057 „Karte auswerfen“	Wurde EjectCard mit dem Parameter Slot aufgerufen: Veranlassen des Kartenauswurfs am Kartenterminal durch Aufruf TUC_KON_057 {
		ctId = $Slot.CtId;
		slotId = $Slot.Slotid;
		displayMessage = $DisplayMsg;
		$timeOut }
		Wurde EjectCard mit dem Parameter CardHandle aufgerufen: Veranlassen des Kartenauswurfs am Kartenterminal durch Aufruf TUC_KON_057 {
		ctId = CM_CARD_LIST($CardHandle).CTID;
		slotId = CM_CARD_LIST ($CardHandle).SLOTNO; ;
		displayMessage = $DisplayMsg;
		$timeOut }

Tabelle 54: TAB_KON_721 Fehlercodes Operation „EjectCard"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4203	Technical	Error	Karte deaktiviert, aber nicht entnommen
4101	Technical	Error	Karten-Handle ungültig

[<=]

4.1.4.6 Betriebsaspekte

4.1.4.6.1 Allgemeine Betriebsaspekte

TIP1-A_4549 - Initialisierung Kartenterminaldienst
Während des Bootvorgangs, nach dem Einlesen der persistierten Informationen des Kartenterminaldienstes MUSS der Konnektor für jedes Kartenterminal CT in CTM_CT_LIST:

- die zugehörigen Attribute CT.SLOTS_USED, CT.VALID_VERSION und ggf. (bei dynamischer Adressvergabe) CT.IP_ADRESS aktualisieren
- für jedes CT mit CT.CORRELATION = „aktiv":
    - Wenn CT.VALID_VERSION = True: TUC_KON_050 „Beginne Kartenterminalsitzung" {ctId=CT.CtID; role=„User"} aufrufen
    - Wenn CT.VALID_VERSION = False: CT.CORRELATION=„gepairt" setzen

[<=]

Hinweis: Bei der Initialiserung des Kartenterminaldienstes liest der Konnektor noch nicht die Karten, um zu ermitteln, welche Karten gesteckt sind. Dies erfolgt erst bei Initialisierung des Kartendienstes.

TIP1-A_4550 - Konfigurationsparameter des Kartenterminaldienstes
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_527 vorzunehmen:

Tabelle 55: TAB_KON_527 Konfigurationswerte eines Kartenterminalobjekts

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
`CTM_SERVICE_DISCOVERY_PORT`	`Portnummer`	Der Administrator MUSS die Portnummer eingeben können, auf der die KTs im lokalen Netz auf Dienstanfragen hören. Default-Wert=4742

CTM_SERVICE_DISCOVERY_TIMEOUT	X Sekunden	Der Administrator MUSS die Anzahl Sekunden eingeben können, die der Konnektor auf Antworten zu Service-Discovery-Anfragen wartet. Default-Wert=3
CTM_SERVICE_ANNOUNCEMENT_PORT	Portnummer	Der Administrator MUSS die Portnummer eingeben können, auf der der Konnektor auf Dienstbeschreibungspakete hört. Default-Wert=4742
CTM_SERVICE_DISCOVERY_CYCLE	X Minuten	Der Administrator MUSS die Anzahl Minuten einstellen können, in denen der Konnektor wiederholt Service Discovery Nachrichten absetzt. Default-Wert=10, 0=Deaktiviert
CTM_KEEP_ALIVE_INTERVAL	X Sekunden	Intervall in Sekunden in den Keep-Alive-Nachrichten an das Kartenterminal gesendet werden Der Administrator MUSS diesen Wert im vorgegebenen Bereich anpassen können. Wertebereich:1-10 Default-Wert=10
CTM_KEEP_ALIVE_TRY_COUNT	Anzahl der Versuche	Anzahl von aufeinander folgenden, nicht beantworteten Keep-Alive-Nachrichten, nach denen ein Timeout der TLS-Verbindung festgestellt wird Der Administrator MUSS diesen Wert im vorgegebenen Bereich anpassen können. Wertebereich:3-10 Default-Wert=3
CTM_TLS_HS_TIMEOUT	X Sekunden	Der Administrator MUSS die Anzahl Sekunden eingeben können, die der Konnektor auf den TLS-Verbindungsaufbau zum Kartenterminal wartet (Handshake-Timeout). Wertebereich:1-60 Default-Wert=10

[<=]

TIP1-A_4986 - Informationsparameter des Kartenterminaldienstes
Die Managementschnittstelle MUSS es einem Administrator ermöglichen die Informationsparameter gemäß Tabelle TAB_KON_528 einzusehen:

Tabelle 56: TAB_KON_528 Informationsparamter des Kartenterminaldienstes

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
CTM_SUPPORTED_KT_VERSIONS	Liste von EHEALTH-	Der Administrator MUSS die Liste der vom Konnektor unterstützten modellunabhängigen

	Interface-Versionen	EHEALTH-Interface-Versionen einsehen können.

[<=]

4.1.4.6.2 Kartenterminals pflegen

Im Folgenden werden die Administratorinteraktionen beschrieben, die zum Hinzufügen, Pairen, Bearbeiten und Löschen von Kartenterminals innerhalb der CTM_CT_LIST angeboten werden müssen. Eine Aktualisierung der Kartenterminals mit neuer Firmware wird in Kapitel 4.3.9 beschrieben.

TIP1-A_4551 - Einsichtnahme von Kartenterminaleinträgen
Die Managementschnittstelle MUSS es einem Administrator ermöglichen die Liste der verwalteten und neu entdeckten Kartenterminals einzusehen (CTM_CT_LIST).
[<=]

TIP1-A_4552 - Manueller Verbindungsversuch zu Kartenterminals
Die Managementschnittstelle MUSS es einem Administrator ermöglichen zu jedem CT-Object-Eintrag in CTM_CT_LIST mit CT.CONNECTED=Nein und CT.CORRELATION=aktiv einen manuellen Verbindungsaufbau über TUC_KON_050 {ctId=CtID; role=„User"} auszulösen.
[<=]

TIP1-A_4553 - Einsichtnahme in und Aktualisierung der Kartenterminaleinträge
Die Managementschnittstelle MUSS es einem Administrator ermöglichen zu jedem CT-Object-Eintrag in CTM_CT_LIST die Werte gemäß Tabelle TAB_KON_529 einsehen zu können:
Zu jedem Eintrag MUSS der Administrator TUC_KON_055 „Befülle CT-Object" auslösen können.

Tabelle 57: TAB_KON_529 Anzeigewerte zu einem Kartenterminalobjekt

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
Geräte kenndaten		
CT.CTID	Identifier	Eindeutige, statische Identifikation des Kartenterminals
CT.IS_PHYSICAL	Ja/Nein	Kennzeichnung, ob es sich um ein logisches oder physisches Kartenterminal handelt (siehe auch TAB_KON_522 Parameterübersicht des Kartenterminaldienstes)
CT.MAC_ADRESS	MAC-Adresse	die MAC-Adresse des Kartenterminals
CT.HOSTNAME	String	SICCT-Terminalname des Kartenterminals, auch als FriendlyName bezeichnet
CT.IP_ADRESS	IP-Adresse	die IP-Adresse des Kartenterminals
CT.TCP_PORT	Portnummer	der TCP-Port des SICCT-Kommandointerpreters des Kartenterminals

CT.SLOTCOUNT	Nummer	Anzahl der Slots des Kartenterminals
CT.SLOTS_USED	Liste	Liste der mit Karten belegten Slots
CT.PRODUCT INFORMATION	Inhalt Product Information.xsd	die Herstellerinformationen zum Kartenterminal gemäß [gemSpec_OM]
CT.EHEALTH_ INTERFACE_ VERSION	Version	Die EHEALTH-Interface-Version des Kartenterminals, die mittels des SICCT-Kommandos GET STATUS aus dem Element VER des Discretionary Data Objects ermittelt wurde
CT.VALID_ VERSION	Boolean	True, wenn die Version des Kartenterminals (CT.EHEALTH_INTERFACE_VERSION) durch den Konnektor unterstützt wird, d.h. zu den in CTM_SUPPORTED_KT_VERSIONS passt
Pairingdaten		
CT.SMKT_AUT	X.509-Cert	C.SMKT.AUT-Zertifikat des Kartenterminals, gespeichert im Rahmen des Pairings
Verbindungs daten		
CT. CORRELATION	bekannt zugewiesen gepairt aktiv aktualisierend	Der Korrelationsstatus zum Konnektor: • bekannt (über Service Announcement/Service Discovery gelernte Kartenterminals), • zugewiesen (durch den Administrator aus dem Bereich der bekannten Kartenterminals oder manuell konfigurierte Kartenterminals), • gepairt (Pairing erfolgreich aber noch nicht zum Verbindungsaufbau freigegeben) • aktiv (durch Administrator zum Verbindungsaufbau freigegeben), • aktualisierend (ein laufender Updatevorgang, ausgelöst durch den Konnektor; Der Zustand tritt ein, wenn der Kartenterminaldienst das Event „KSR/UPDATE/START" fängt und endet mit dem Event „KSR/UPDATE/END"),
CT.CONNECTED	Ja/Nein	Der Verfügbarkeitsstatus des Kartenterminals (Ja = nach Aufbau der TLS-Verbindung und erfolgter zweiter Authentifizierung)
CT.ACTIVEROLE	User/Admin	Benutzerrolle, die für die aktuelle Session verwendet wird

KT-Admin-Credentials		
CT.ADMIN_USERNAME	String	Username des Administrators am KT

[<=]

TIP1-A_4554 - Bearbeitung von Kartenterminaleinträgen
Die Managementschnittstelle MUSS es einem Administrator ermöglichen zu jedem CT-Object-Eintrag in CTM_CT_LIST die Werte gemäß Tabelle TAB_KON_530 ändern zu können:
Zur Überprüfung der veränderten Parameter auf Korrektheit MUSS nach Änderung von CT.IP_ADRESS, TCP_PORT oder HOSTNAME TUC_KON_054 mit Mode= ManuallyModified und allen vorhandenen CT-Parametern aufgerufen werden. Endet der Aufruf von TUC_KON_054 mit einem Fehler, MUSS der Konnektor die geänderten Konfigurationswerte auf ihren Ausgangswert zurücksetzen.

Tabelle 58: TAB_KON_530 Konfigurationswerte eines Kartenterminalobjekts

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
CT.IP_ADRESS	IP-Adresse	Der Administrator MUSS für KTs mit CT.IS_PHYSICAL=Ja die IPv4-Adresse des Kartenterminals eingeben können.
CT.TCP_PORT	Portnummer	Der Administrator MUSS für KTs mit CT.IS_PHYSICAL=Ja den TCP-Port des SICCT-Kommandointerpreters des Kartenterminals eingeben können.
CT.HOSTNAME	String	Der Administrator MUSS den SICCT-Terminalnamen (Hostname) - auch als FriendlyName bezeichnet - des Kartenterminals eingeben können.
CT.ADMIN_USERNAME	String	Der Administrator MUSS für KTs mit CT.IS_PHYSICAL=Ja den Username des KT-Administrators des Kartenterminals eingeben können.
CT.ADMIN_PASSWORD	String	Der Administrator MUSS für KTs mit CT.IS_PHYSICAL=Ja das Password des KT-Administrators des Kartenterminals eingeben können.

[<=]

TIP1-A_6477 - Manuelles Service Discovery
Die Managementschnittstelle MUSS es einem Administrator ermöglichen, ein Service Discovery entsprechend [SICCT] auszulösen, um neue Kartenterminals hinzuzufügen.
[<=]

TIP1-A_4555 - Manuelles Hinzufügen eines Kartenterminals
Die Managementschnittstelle MUSS es einem Administrator ermöglichen für neue Kartenterminals CT-Objects manuell in CTM_CT_LIST aufzunehmen.

Hierzu MUSS der Administrator für das neue Kartenterminal folgende Werte eingeben können:

- IP-Adresse (Eingabe verpflichtend)
- TCP-Port (Eingabe optional)
- MAC-Adresse (Eingabe optional)
- Hostname (Eingabe optional)

Bestätigt der Administrator seine Eingaben, MUSS TUC_KON_054 mit Mode=ManuallyAdded und allen eingegebenen Parametern aufgerufen werden.
[<=]

Als Sicherung gegen den unbemerkten Austausch von Kartenterminals oder deren Identitäten wird das gSMC-KT über den Konnektor logisch an das eHealth-Kartenterminal gebunden. Dieser Vorgang wird als Pairing von Kartenterminal und gSMC-KT bezeichnet und ist ausführlich in [gemSpec_KT] beschrieben.

TIP1-A_4556 - Pairing mit Kartenterminal durchführen
Die Managementschnittstelle MUSS es einem Administrator ermöglichen alle Kartenterminals mit CT.CORRELATION = „zugewiesen" in einer Liste einzusehen und für einen ausgewählten Eintrag mit CT.VALID_VERSION=True TUC_KON_053 auslösen zu können.
[<=]

TIP1-A_4557 - Ändern der Korrelationswerte eines Kartenterminals
Die Managementschnittstelle MUSS es einem Administrator ermöglichen zu einem Kartenterminal aus CTM_CT_LIST für KTs mit CT.IS_PHYSICAL=Ja den Wert für CT.CORRELATION nach folgenden Mustern zu ändern:

- CT.CORRELATION = „bekannt"
  Das Kartenterminal gilt als nicht durch den Konnektor verwaltet.
- → „zugewiesen":
  Ein (per Service Announcement entdecktes) Kartenterminal dem Konnektor zuweisen.
  Folgende Schritte MUSS der Konnektor für diesen Zustandswechsel zuvor erfolgreich durchlaufen:
  - Rufe TUC_KON_055 **„**Befülle CT-Object**"**
  - Prüfen, ob CT.HOSTNAME bereits für ein anderes Kartenterminal in CTM_CT_LIST verwendet wird. Wenn ja MUSS dieser Änderungsversuch fehlschlagen (Prinzip der Eindeutigkeit verletzt). Der Administrator MUSS eine entsprechende Fehlermeldung erhalten.
- CT.CORRELATION = „zugewiesen"
  Das Kartenterminal gilt als durch den Konnektor verwaltet.
  - → „bekannt"
  - → „gepairt":
    Das Pairing wurde erfolgreich durchgeführt; die Werte CT.SMKT_AUT, CT.SHARED_SECRET sind im CT-Objekt eingetragen.
- CT.CORRELATION = „gepairt"
  Verbundenheit zwischen Kartenterminal und gesteckter gSMC-KT wurde nachgewiesen

- → „zugewiesen“:
  Die Werte CT.SMKT_AUT, CT.SHARED_SECRET werden gelöscht
- → „aktiv“:
  Wechsel nur möglich, wenn CT.VALID_VERSION=True. Dann Aufruf von TUC_KON_050 „Beginne Kartenterminalsitzung“ {ctId=CT.CtID; role=„User“}

- CT.CORRELATION = „aktiv"
  Das Kartenterminal kann fachlich genutzt werden
  - → „gepairt“:
    Eventuelle TLS-Verbindung wird beendet, CT.CONNECTED auf Nein gesetzt.

[<=]

TIP1-A_5698 - Löschen von Kartenterminaleinträgen
Die Managementschnittstelle MUSS einem Administrator die Möglichkeit bieten, Kartenterminals aus der Liste der Kartenterminals (CTM_CT_LIST) zu entfernen.
[<=]

4.1.4.6.3 Import der Kartenterminal-Informationen

Im Rahmen des Konnektormanagements müssen die Konfigurationsdaten des Konnektors ex- und importiert werden können (siehe Kapitel 4.3.3). Eine Sonderstellung nimmt dabei der Import von Kartenterminalinformationen ein, da hier im Rahmen des Imports folgende Interaktion mit dem Administrator erforderlich ist:

TIP1-A_5011 - Import von Kartenterminal-Informationen
Der Konnektor MUSS vor der endgültigen Aktivierung der importierten Kartenterminalkonfiguration folgende zusätzliche Schritte ausführen:

1. Die Liste der zu importierenden Kartenterminals MUSS dem Administrator angezeigt werden. Er MUSS die Möglichkeit erhalten, einzelne Kartenterminals aus dieser Liste zu löschen.
2. Erst nach Bestätigung durch den Administrator werden die Kartenterminalinformationen in die Kartenterminalverwaltung übernommen.
3. Sofern die Kartenterminal-Konfiguration in einen Konnektor mit neuer Identität importiert werden soll (neuer Konnektor oder neuer privater Schlüssel und neues Zertifikat C.SAK.AUT auf der gSMC-K), muss die neue Identität des Konnektors allen importierten Kartenterminals bekannt gemacht werden (Wartungs-Pairing, siehe auch [gemSpec_KT#2.5.2.4]).
   a. Dazu baut der Konnektor unter der Nutzung von C.SAK.AUT eine temporäre TLS-Verbindung auf und sendet das eHealth-Kartenterminal-Kommando EHEALTH TERMINAL AUTHENTICATE in der Variante „ADD” an jedes in der Liste aufgeführte Kartenterminal. Mit dem Kommando und P2=03 holt sich der Konnektor eine Challenge.
   b. Der eigentliche Austausch bzw. die Aufnahme des neuen Zertifikates erfolgt im KT erst, nachdem diese Challenge mit dem Kommando EHEALTH TERMINAL AUTHENTICATE im Modus P2=04 vom Konnektor korrekt beantwortet wurde. Dieses Kommando sowie die Erzeugung der Challenge-Antwort wird in [gemSpec_KT#3.7.2.4] und [gemSpec_KT#3.7.2] beschrieben.

c. Nach erfolgreicher Abarbeitung des Kommandos wird der Eintrag in die interne Liste der gepairten Kartenterminals übernommen und die temporäre Verbindung zum Kartenterminal abgebaut. Kann ein Kartenterminal nicht erreicht werden, so MUSS die Befehlskette nachgeholt werden, sobald das Kartenterminal vom Konnektor wieder als verfügbar erkannt wird.

4. Zur abschließenden Kontrolle und zur weiteren fachlichen Nutzung baut der Konnektor zu jedem der neu konfigurierten und aktiv gesetzten Kartenterminals via TUC_KON_050 eine Verbindung auf.

[<=]

4.1.5 Kartendienst

Innerhalb des Kartendienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „CARD"
- Konfigurationsparameter: „CM_"

Der Konnektor verwaltet eine Liste aller Karten (CM_CARD_LIST), die in die vom Konnektor verwalteten Kartenterminals gesteckt sind (CTM_CT_LIST). Alle Ereignisse und Operationen, die sich auf Karten beziehen, werden durch diesen Basisdienst gekapselt.

Für jede gesteckte Karte vergibt er einen eindeutigen Identifikator (im weiteren Text CardHandle bezeichnet), mit dem diese Karte adressiert werden kann, um zu diesen oder mit diesen Karten Operationen auszuführen. Dieses Handle ist gültig bis zum Entfernen der Karte aus dem Kartenterminal.

Um die in [gemSpec_Perf] geforderten Zeiten für kartenbezogene Operationen erreichen zu können, kann es erforderlich sein, dass der Konnektor möglichst viele Informationen der Karten cached. Hierzu gehören Steuerdaten wie Extended Length, Version etc. aber auch Zertifikate der Karte (X.509 und CVC). Da es sich bei Caching um einen internen Mechanismus handelt, der sich nicht auf das funktionale Außenverhalten von TUCs oder Operationen auswirkt, wird das Caching nicht weiter beschrieben oder explizit gefordert. Es kann aber Anforderungen aus Sicherheitssicht bezüglich des Cachings geben (insbesondere hinsichtlich der erlaubten Caching-Dauer). Die Einhaltung dieser Vorgaben wird im Rahmen der CC-Evaluierung geprüft werden.

Der Kartendienst verwaltet mindestens die in der informativen Tabelle TAB_KON_531 ausgewiesenen Parameter, weitere herstellerspezifische Parameter sind möglich. Die normative Festlegung wann welche Parameter wie belegt werden, erfolgt in den folgenden Abschnitten und Unterkapiteln.

Tabelle 59: TAB_KON_531 Parameterübersicht des Kartendienstes

ReferenzID	Belegung	Zustandswerte
CM_CARD_LIST	Liste von Card-Objekten	Eine Liste von Repräsentanzen (CardObjects) der dem Konnektor bekannten Karten. Die Attribute der Card-Objekte sind im Folgenden gelistet.
CARD.CARDHANDLE		vom Konnektor vergebenen eindeutigen Identifikator (Handle).

CARD.CTID		Kartenterminal, in dem die Karte steckt
CARD.SLOTNO		Slot, in dem die Karte steckt
CARD.ICCSN		ICCSN der Karte (sofern auslesbar),
CARD.TYPE		Typ der Karte gemäß Tabelle TAB_KON_500 Wertetabelle Kartentypen
CARD.CARDVERSION		die Versionsinformationen zum Produkttyp der Karte und den gespeicherten Datenstrukturen gemäß [gemSpec_Karten_Fach_TIP].
CARD.CARDVERSION. COSVERSION		Produkttypversion des COS
CARD.CARDVERSION. OBJECTSYSTEMVERSION		Produkttypversion des Objektsystems
CARD.CARDVERSION. CARDPTPERSVERSION		Produkttypversion der Karte bei Personalisierung
CARD.CARDVERSION. DATASTRUCTUREVERSION		Version der Speicherstrukturen (aus EF.Version)
CARD.CARDVERSION. LOGGINGVERSION		Version der Befüllvorschrift für EF.Logging
CARD.CARDVERSION. ATRVERSION		Version der Befüllvorschrift für EF.ATR
CARD.CARDVERSION. GDOVERSION		Version der Befüllvorschrift für EF.GDO
CARD.CARDVERSION. KEYINFOVERSION		Version der Befüllvorschrift für KeyInfo
CARD.INSERTTIME	Timestamp	Zeitpunkt, an dem die Karte gesteckt wurde
CARD.CARDHOLDERNAME	String	Name des Karteninhabers bzw. der Institution/Organisation (subject.commonName)
CARD.KVNR	String	Versicherten-ID (unveränderbarer Teil der KVNR)
CARD. CERTEXPIRATIONDATE		Ablaufdatum des AUT-Zertifikats der Karte
CARD.CARDSESSION_ LIST	Liste von CardSession-Objekten	Eine Liste von Repräsentanzen (CardSession-Objects) der pro Karte vorhandenen Kartensitzungen. Die Attribute der CardSession-Objekte sind im Folgenden gelistet. Das Tripel aus MandantID, CSID und

		UserID bildet den Kontext ab, in welchem diese Kartensitzung initiiert wurde.
CARDSESSION. AUTHSTATE	Liste von Einträgen aus a)C2C:KeyRef, Role oder b) CHV: PINRef	Liste von erreichten Sicherheitszuständen. Jeder einzelne Sicherheitszustand kann entweder über C2C gegen KeyRef (mit einer bestimmten Rolle gemäß [gemSpec_PKI_TI#Tab_PKI_918]) oder Card Holder Verification (CHV) gegen eine referenzierte PIN erreicht worden sein. Für eGK G2.0 wird der Zustand der MRPINs nicht in AuthState gespeichert.
CARDSESSION. MANDANTID		Mandant-ID
CARDSESSION.CSID		Clientsystem-ID
CARDSESSION.USERID		Nutzer-ID
CARDSESSION.AUTHBY	Referenz auf CardSession	Kartensitzung, über die diese Karte freigeschaltet wurde (nur für eGK belegt)
CARDSESSION. SIGNMODE	„PIN" oder „Comfort"	Signaturmodus „PIN": Komfortsignaturmodus ist für die Karte ausgeschaltet "Comfort": Komfortsignaturmodus ist eingeschaltet Default-Wert="PIN" Nur relevant für den HBA

4.1.5.1 Funktionsmerkmalweite Aspekte

TIP1-A_4988-02 - Unterstützung von Kartengenerationen

Der Konnektor MUSS für eGK, HBA, SMC-B, gSMC-KT und gSMC-K Karten der Generation 2 unterstützen. Karten der Generation 2 sind alle Karten, deren Version des dem aktiven Objektsystem zugrundeliegenden Produkttyps (Tag 'C1' in EF.Version2) den Wert 4.x.y hat, wobei x,y in {0, …, 255}.
Der Konnektor DARF eGKs der Generation < 2 (also 1 und 1+) NICHT unterstützen. eGKs der Generation < 2 werden im Konnektor als CARD.TYPE = UNKNOWN geführt.
Bei Karten der Generation 2

- MUSS der Konnektor die RSA-basierten Geräte-CV-Zertifikate unterstützen,
- MUSS der Konnektor die ECC-basierten Geräte-CV-Zertifikate unterstützen.

[<=]

Es kann notwendig sein, Karten der Generation 2 (G2) näher zu bezeichnen. In diesem Fall wird in G2.0- und G2.1-Karten unterschieden. Wird von Karten der Generation 2 gesprochen, so gilt die Festlegung für G2.x (G2.0, G2.1 und höher) des betrachteten Kartentyps.

TIP1-A_4558 - Caching-Dauer von Kartendaten im Konnektor
Sofern der Konnektor Daten gesteckter Karten cached, so DÜRFEN diese Daten von HBAx und SM-B NICHT länger als 24 Stunden gecached werden.
Der Konnektor DARF Daten der eGK NICHT über den Steckzyklus der Karte hinaus cachen.
Ausnahme: Eine Cachingdauer über den Steckzyklus der eGK hinaus wird von einer Fachanwendung gefordert und durch die Fachmodulspezifikation dieser Fachanwendung definiert.
[<=]

TIP1-A_6031 - Kein selbsttätiges Zurücksetzen der SM-B
Der Konnektor DARF NICHT selbsttätig die SM-B und deren Sicherheitszustände zurücksetzen, auch nicht, wenn die Daten der SM-B nach Ablauf der 24-Stunden-Frist neu in den Cache eingelesen werden.
[<=]

TIP1-A_4559 - Konnektorzugriffsverbot auf DF.KT
Der Konnektor DARF NICHT auf das DF.KT einer gSMC-KT zugreifen.
[<=]

TIP1-A_4560 - Rahmenbedingungen für Kartensitzungen
Der Konnektor MUSS alle Zugriffe auf Karten aus CM_CARD_LIST, die den Sicherheitszustand der Karte erhöhen können oder einen erhöhten Sicherheitszustand der Karte voraussetzen, im Kontext einer Kartensitzung zu dieser Karte durchführen (CARD.CARDSESSION). Ausgenommen hiervon ist der Zugriff durch das CMS (bzw. VSDD) auf die eGK.
Der Konnektor MUSS sicherstellen, dass in einer Kartensitzung nur dann auf einen erhöhten Sicherheitszustand einer Karte zugegriffen werden kann, wenn die zur Erreichung dieses Sicherheitszustandes erforderlichen Authentisierungen (PIN-Verifikation, C2C-Rollen-Authentisierung etc.) in dieser verwendeten Kartensitzung erfolgreich durchgeführt wurden.
Der Konnektor MUSS Authentisierungen in einer Kartensitzung so durchführen, dass in anderen Kartensitzungen vorhandene Sicherheitszustände nicht beeinflusst werden. (Eine falsche PIN-Eingabe in einer Kartensitzung darf den erhöhten Sicherheitszustand einer anderen Sitzung nicht zurücksetzen etc.).
Der Konnektor SOLL die Verwaltung der Sicherheitsstatus der Kartensitzungen so über die Nutzung logischer Kartenkanäle umsetzen, dass PIN-Verifikationen, die für die Dauer der Kartensitzung Gültigkeit haben, nicht unnötig wiederholt werden müssen.
[<=]

Für die TUCs zur PIN-Eingabe, Änderung und Entsperrung sind Festlegungen hinsichtlich der auf dem Kartenterminal anzuzeigenden Meldungen erforderlich. Die folgende Tabelle definiert diese Terminalanzeigen gemäß [SICCT#5.5.10.19].

TIP1-A_4561-02 - Terminal-Anzeigen für PIN-Operationen
Der Konnektor MUSS im Rahmen des interaktiven PIN-Handlings die folgenden Displaymessages für die Anzeige im Kartenterminal verwenden:

Tabelle 60: TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal

Karte/ Kontext	PIN-Referenz	I/ O	Terminal-Anzeige	ANW (max.Anz. Zeichen)

eGK /PIN-Eingabe für Vertreter-PIN	PIN.AMTS_REP	I	Vertreter-PIN•***0x0B***für•***0x0B***ANW ***0x0F***Vertr-PIN:	22
eGK /PIN-Eingabe für Vertreter-PIN ändern	PIN.AMTS_REP	I	Vertreter-PIN•***0x0B***ändern ***0x0F***PIN.eGK:	
eGK /PIN-Eingabe für Vertreter-PIN entsperren	PIN.AMTS_REP	I	Vertreter-PIN•***0x0***entsperren ***0x0F***PIN.eGK:	
eGK /PIN-Eingabe für PIN-Schutz einschalten	MRPIN.NFD, MRPIN.DPE, MRPIN.AMTS, MRPIN.GDD	I	PIN-Schutz•***0x0B***ANW•***0x0B***einschalten ***0x0F***PIN.eGK:	16
eGK /PIN-Eingabe für PIN-Schutz abschalten	MRPIN.NFD, MRPIN.DPE, MRPIN.AMTS, MRPIN.GDD	I	PIN-Schutz•***0x0B***ANW•***0x0B***abschalten ***0x0F***PIN.eGK:	16
eGK /Sonstige	ALLE (außer PIN.AMTS_REP)	I	PIN•***0x0B***für•***0x0B***ANW ***0x0F***PIN.eGK:	32
HBAx	PIN.CH	I	Eingabe•***0x0B***Freigabe-PIN•***0x0B***HBA ***0x0F***PIN.HBA:	
	PIN.QES (Signatur auslösen)	I	#*UVW-XYZ*•***0x0B***Eingabe•***0x0B***Signatur-PIN•***0x0B***HBA ***0x0F***PIN.QES:	
HBA	PIN.QES (Komfortsignatur aktivieren)	I	Komfortsignatur•***0x0B***aktivieren•***0x0B***HBA ***0x0F***PIN.QES:	
SMC-B	PIN.SMC	I	Eingabe•***0x0B***PIN•SMC-B•***0x0B***SLOT:*X* ***0x0F***PIN.SMC:	
ANDERE	BELIEBIG	I	Herstellerspezifisch	
Erfolgreiche PIN-Eingabe	ALLE	O	PIN•***0x0B***erfolgreich•***0x0B***verifiziert!	
Fehlerhafte PIN-Eingabe	ALLE	O	PIN•***0x0B***falsch•***0x0B***oder•***0x0B***gesperrt!	
PUK-Eingabe	eGK PUK.CH	I	Eingabe•***0x0B***Versicherten-***0x0B***PUK ***0x0F***PUK.eGK:	
	HBAx PUK.CH	I	Eingabe•***0x0B***Freigabe-PUK•***0x0B***HBA ***0x0F***PUK.HBA:	
	HBAx PUK.QES	I	Eingabe•***0x0B***Signatur-PUK•***0x0B***HBA ***0x0F***PUK.QES:	

	SMC-B PUK.SMC	I	Eingabe•***0x0B***PUK•SMC-B•***0x0B***SLOT:*X* ***0x0F***PUK.SMC:
Erfolgreiche PUK-Eingabe	ALLE	O	PIN•***0x0B***erfolgreich•***0x0B***entsperrt!
Fehlerhafte PUK-Eingabe	ALLE	O	PUK•***0x0B***falsch•***0x0B***oder•***0x0B***gesperrt!
Eingabe einer neuen PIN	eGK ALLE (außer PIN.AMTS_REP)	I	Eingabe• ***0x0B***neue•***0x0B***Versicherten-***0x0B***PIN• ***0x0B***(6-8•Ziffern) ***0x0F***PIN.eGK:
	eGK PIN.AMTS_REP	I	Eingabe• ***0x0B***neue•***0x0B***Vertreter-PIN• ***0x0B***(6-8•Ziffern) ***0x0F***Vertr-PIN:
	HBAx PIN.CH	I	Eingabe•***0x0B***neue•***0x0B***Freigabe-PIN•***0x0B***HBA•***0x0B***(6-8•Ziffern) ***0x0F***PIN.HBA:
	HBAx PIN.QES	I	Eingabe• ***0x0B***neue•***0x0B***Signatur-PIN•***0x0B***HBA•***0x0B***(6-8•Ziffern) ***0x0F***PIN.QES:
	SMC-B PIN.SMC	I	Eingabe•***0x0B***neue•***0x0B***PIN•SMC-B• ***0x0B***SLOT:*X*•***0x0B***(6-8•Ziffern) ***0x0F***PIN.SMC:
Eingabe einer Transport-PIN	eGK PIN.CH	I	Eingabe•***0x0B***Transport- ***0x0B***Versicherten-***0x0B***PIN ***0x0F***T-PIN.eGK:
	HBAx PIN.CH	I	Eingabe•***0x0B***Transport-***0x0B***PIN•***0x0B***HBA ***0x0F***T-PIN.HBA:
	HBAx PIN.QES	I	Eingabe•***0x0B***Transport-***0x0B***PIN•***0x0B***HBA ***0x0F***T-PIN.QES:
	SMC-B PIN.SMC	I	Eingabe•***0x0B***Transport-***0x0B***PIN•SMC-B•***0x0B***SLOT:*X* ***0x0F***T-PIN.SMC:
Wieder-holung einer neuen PIN	eGK PIN.CH	I	Eingabe•***0x0B***Versicherten-***0x0B***PIN•***0x0B*** wiederholen! ***0x0F***PIN.eGK:
	eGK PIN.AMTS_REP	I	Eingabe• ***0x0B***neue•***0x0B***Vertreter-PIN• ***0x0B*** wiederholen! ***0x0F***Vertr-PIN:
	HBAx PIN.CH	I	Eingabe•***0x0B***für•HBA•***0x0B***wiederholen! ***0x0F***PIN.HBA:
	HBAx PIN.QES	I	Eingabe•***0x0B***für•HBA•***0x0B***wiederholen! ***0x0F***PIN.QES:

	SMC-B PIN.SMC	I	Eingabe•***0x0B***PIN.SMC•***0x0B***SLOT:*X*•***0x0B***wiederholen!***0x0F***PIN.SMC:
Ungleichheit bei der Wieder-holung der Eingabe der neuen PIN	ALLE	O	PINs•***0x0B***nicht•***0x0B***identisch!•***0x0B***Abbruch!
Erfolgreiche PIN-Änderung	ALLE	O	PIN•0x0Berfolgreich•0x0Bgeändert!
Anzeigen am lokalen Terminal beim Remote-PIN-Verfahren für das Ergebnis der Verschlüsselung durch die gSMC-KT			
Erfolgreiche Verschlüsselung	ALLE	O	Eingabe•0x0Bwird•0x0Bbearbeitet.
Fehler bei der Verschlüsselung	ALLE	O	Eingabe•0x0Bfehlgeschlagen.

[<=]

Hinweise zu den Terminalanzeigen bei PIN-Eingaben und zu obiger Tabelle:

- ANW kennzeichnet den Anwendungsfall und wird durch den vom Aufrufer übergebenen String ersetzt (siehe z. B. TUC_KON_012 „PIN verifizieren“)
- Zu PIN.SMC: "Slot:X" im PIN-Prompt gibt die Slot-Nummer im Kartenterminal an, in der die SMC steckt, da in einem Kartenterminal mehr als eine SMC stecken kann.
- Variable Teile der Terminalanzeige (Job- und Slot-Nummer) sind kursiv formatiert.
- Zeichensatz gemäß ISO 646DE-/DIN 66003-Codierung
- max. 48 Zeichen Text + 10 Zeichen PIN-Prompt bei Input
- max. 48 Zeichen bei Output
- Leerzeichen werden als "•" dargestellt
- UVW-XYZ: zeigt die Jobnummer an (siehe Kapitel 4.1.8.1.4)
- #: Beginn der Jobnummer zur Verifizierung des korrekten Kartenterminals
- Weitere Details zur Gestaltung der Jobnummer finden sich im Kapitel 4.1.8.1.4.
- Die Zeilenumbrüche in der Spalte "Terminal-Anzeige" sind editorisch bedingt.
- 0x0B und 0x0F (Sollbruchstellen bzw. Trennung zwischen Nachricht und PIN-Prompt) sind Trennzeichen gemäß [SICCT#5.6.1].

In den Technischen Use Cases TUC_KON_012 „PIN verifizieren“, TUC_KON_019 „PIN ändern“, TUC_KON_021 „PIN entsperren“ wird das Remote-PIN-Verfahren verwendet, sofern die Zielkarte in einem als für den Arbeitsplatz entfernt definiertem Kartenterminal steckt (siehe Kap. 4.1.1.1, Relation [7]). In diesem Fall erfolgt die Nutzerinteraktion am

Remote-PIN-KT von workplaceId (PinInputKT). Dabei wendet der Konnektor das folgende Verfahren an:

TIP1-A_5012 - Remote-PIN-Verfahren
Der Konnektor MUSS das Remote-PIN Verfahren im Sinne der BSI TR-03114 unterstützen. Abweichend von der TR-03114 MUSS statt der SMC-A eine gSMC-KT verwendet werden.
Der Konnektor MUSS für die PIN-Objekte: HBA.PIN.CH, HBA.PUK.CH, HBA.PIN.QES, HBA.PUK.QES, SM-B.PIN.SMC und SM-B.PUK.SMC das Remote-PIN Verfahren unterstützen. Für alle anderen Karten und PIN-Objekte DARF das Verfahren NICHT verwendet werden.
Für die Interaktion mit dem Anwender MÜSSEN die Display Messages entsprechend TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal verwendet werden.
Der Ablauf für eine PIN-Operation gegen eine Zielkarte MUSS in diesen logischen Schritten erfolgen:

1. Aufruf TUC_KON_005 „Card-to-Card authentisieren" mit eigens für diesen Zweck erzeugten Cardsession sowohl für die „Sendekarte" im PinInputKT (gSMC-KT) sowie der Zielkarte. AuthMode ist „gegenseitig+TC"
2. Der Benutzer wird mit dem SICCT-Kommando PERFORM VERIFICATION bzw. MODIFY VERIFICATION DATA zur Eingabe der PIN am PinInputKT aufgefordert. Als Display Messages für die erfolgreiche Bearbeitung bzw. Fehler in der Bearbeitung dieser Kommandos müssen die Texte mitgesendet werden, die in TAB_KON_090 für die Ergebnisse der Verschlüsselung durch die gSMC-KT festgelegt sind.
3. Im PinInputKT verschlüsselt die gSMC-KT die eingegebene PIN mit dem zuvor erzeugten Sessionkey.
4. Die verschlüsselte PIN wird in das zur intendierten PIN-Operation passende Kommando eingebettet (PIN verifizieren, ändern oder entsperren - wird durch den eigentlichen PIN-TUC festgelegt) und das Kommando vom Konnektor an die Zielkarte zur Entschlüsselung und Verifikation übergeben. Dabei MUSS die Übertragung im gleichen Logischen Kanal wie die SM Vereinbarung erfolgen.
5. Der Konnektor zeigt das Resultat der Zielkarte mittels SICCT OUTPUT am lokalen Kartenterminal an. Er verwendet dabei den in TAB_KON_090 für die aktuelle PIN-Operation spezifizierten Ausgabetexte.
6. Das Result der Zielkarte wird an den Aufrufer zurückgegeben

Fehlermeldung: Ein Fehler in der Verarbeitung führt zum Abbruch mit Fehlercode 4053 „Remote-PIN nicht möglich" (Security, Error).
[<=]

Hinweis: Derzeit schlägt die Freischaltung der SMC-B durch Card-2-Card-Authentisierung ohne Fehlermeldung fehl. Der Sicherheitszustand der SMC-B wird nicht verändert. Diese Einschränkung betrifft TUC_KON_005 „Card-to-Card authentisieren" (TAB_KON_096).

4.1.5.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_4562 - Reaktion auf „Karte entfernt"
Empfängt der Kartendienst das Ereignis „CT/SLOT_FREE", so MUSS der Konnektor:

- das über die im Ereignis gemeldeten Parameter CtID und SlotNo in CM_CARD_LIST adressierte CardObject CARD identifizieren

- für dieses CardObject folgendes Ereignis absetzen:
  TUC_KON_256{
  topic = „CARD/REMOVED";
  eventType = Op;
  Severity = Info;
  parameters = <Params>}
  wobei <Params> mit folgenden Werten belegt werden MUSS:
  - „CardHandle=$CARD.CARDHANDLE,
  - Type=$CARD.TYP,
  - CardVersion=$CARD.VER,
  - ICCSN=$CARD.ICCSN,
  - CtID=$CARD.CTID,
  - SlotID=$CARD.SLOTID,
  - InsertTime=$CARD.INSERTTIME,
  - CardHolderName=$CARD.CARDHOLDERNAME,
  - KVNR=$CARD.KVNR"
- das zugehörige CardObject aus CM_CARD_LIST entfernen.

[<=]

TIP1-A_4563 - Reaktion auf „Karte gesteckt"
Empfängt der Kartendienst das Ereignis „CT/SLOT_IN_USE", so MUSS der Konnektor für die Karte, die über die im Ereignis gemeldeten Parameter CtID und SlotNo adressiert ist, über TUC_KON_001 ein neues CardObject in CM_CARD_LIST anlegen.
[<=]

4.1.5.3 Interne TUCs, nicht durch Fachmodule nutzbar

4.1.5.3.1 TUC_KON_001 „Karte öffnen"

TIP1-A_4565 - TUC_KON_001 „Karte öffnen"
Der Konnektor MUSS den technischen Use Case „Karte öffnen" gemäß TUC_KON_001 umsetzen.

Tabelle 61: TAB_KON_734 – TUC_KON_001 „Karte öffnen"

Element	Beschreibung
Name	TUC_KON_001 „Karte öffnen"
Beschreibung	Der TUC initialisiert ein Card-Object basierend auf einer physikalischen Karte und fügt es CM_CARD_LIST zu. Die Karte kann erst im Anschluss unter Verwendung des erzeugten CardHandles verwendet werden.
Auslöser	Der Kartenterminaldienst meldet das Belegen eines KT-Slots

Vorbedingungen	• In ctId/slotId steckt eine Karte
Eingangsdaten	• ctId (Kartenterminalidentifikator) • slotId (Nummer des Kartenslots)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	Keine

Standardablauf	1. Prüfe, ob unter ctId und slotId ein Eintrag in CM_CARD_LIST vorhanden ist. Wenn bereits ein Eintrag vorhanden ist, lösche diesen.
	2. Erzeuge neuen Card-Object-Eintrag in CM_CARD_LIST und
	a) Generiere CARD.CARDHANDLE. mit folgenden Anforderungen:
	- Das CardHandle MUSS innerhalb CM_CARD_LIST eindeutig sein.
	- Ein ungültig gewordenes CardHandle DARF innerhalb von 48h NICHT als neues CardHandle vergeben werden.
	b) Befülle CARD.CTID und CARD.SLOTNO mit den Eingangsdaten
	c) Ermittle und befülle (soweit durch Karte unterstützt) die folgenden Daten:
	- CARD.ICCSN
	- CARD.TYPE (mögliche Werte siehe Tabelle TAB_KON_500 Wertetabelle Kartentypen)
	- CARD.CARDVERSION
	- CARD.INSERTTIME (=aktuelle Systemzeit)
	- CARD.CARDHOLDERNAME (aus X.509-AUT-Zertifikat)
	- CARD.KVNR (nur für eGK, aus C.CH.AUT: unveränderbarer Teil der KVNR)
	- CARD.CERTEXPIRATIONDATE (=validity aus X.509-AUT-Zertifikat)
	
	X.509-AUT-Zertifikat bezeichnet für eGK das C.CH.AUT-Zertifikat, für HBAx das C.HP.AUT-Zertifikat und für SMC-B das C.HCI.AUT-Zertifikat.
	
	3. Rufe TUC_KON_256{
	topic = „CARD/INSERTED“;
	eventType = Op;
	severity = Info;
	parameters = <Params>}

	mit <Params> belegt aus dem CARD-Object: „CardHandle=$, CardType=$, CardVersion=$, ICCSN=$,CtID=$, SlotID=$, InsertTime=$, CardHolderName=$, KVNR=$, CertExpirationDate=$“ In CardVersion sind die Werte - COSVERSION und - OBJECTSYSTEMVERSION aus CARD.CARDVERSION einzutragen. Für eGK G1+ ist zusätzlich die - DATASTRUCTUREVERSION aus CARD.CARDVERSION einzutragen. CardVersion kann weitere Werte aus CARD.CARDVERSION enthalten.
Varianten/ Alternativen	Im Falle der KVK gibt es kein EF.ATR, EF.GDO und EF.DIR. Es wird daher lediglich der ATR ausgewertet, den das Kartenterminal beim Stecken der Karte liefert.
Fehlerfälle	(-> 2c) Karte/Kartenterminal antwortet mit einer spezifischen Fehlermeldung, Fehlercode <gemäß [gemSpec_COS]/[SICCT]> Auch im Fehlerfall wird Schritt 3 durchlaufen. Wenn nicht alle zu einem Kartentyp notwendigen Daten von der Karte gelesen werden konnten, dann wird Schritt 3 mit CardType=UNKNOWN ausgeführt. Auch im Fehlerfall wird Schritt 3 durchlaufen. Wenn nicht alle zu einem Kartentyp notwendigen Daten von der Karte gelesen werden konnten, dann wird Schritt 3 mit CardType=UNKNOWN ausgeführt.
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

[<=]

4.1.5.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.5.4.1 TUC_KON_026 „Liefere CardSession“

TIP1-A_4566 - TUC_KON_026 „Liefere CardSession“
Der Konnektor MUSS den technischen Use Case „Liefere CardSession“ gemäß TUC_KON_26 umsetzen.

Tabelle 62: TAB_KON_735 - TUC_KON_026

Element	Beschreibung
Name	TUC_KON_026 „Liefere CardSession“
Beschreibung	Dieser Use Case gibt auf Grund der übergebenen Parameter die zugehörige CardSession zurück. Ist für die Parameterkombination noch keine CardSession vorhanden, wird eine neue erzeugt und im zugehörigen Card-Object hinterlegt.
Auslöser	• Indirekter Aufruf über durch Clientsysteme ausgeführte Operationen. • Aufruf durch Fachmodul
Vorbedingungen	Keine
Eingangsdaten	• mandantId • clientSystemId • cardHandle • userId - *optional/verpflichtend, wenn cardType = HBAx*
Komponenten	Konnektor
Ausgangsdaten	• cardSession
Standardablauf	1. Ermittle Card in CM_CARD_LIST über cardHandle 2. Prüfe dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Ermittle cardSession in Card.CARDSESSION_LIST über mandantId, clientSystemId und userId

Varianten/ Alternativen	(→3) Wenn keine CardSession für diese Parameter vorhanden: 1. erzeuge neue CardSession in Card. CARDSESSION_LIST 2. Befülle CARDSESSION.MANDANTID, .CSID und .USERID mit Übergabeparametern
Fehlerfälle	(→2) Karte bereits reserviert, Fehlercode 4093
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 63: TAB_KON_824 Fehlercodes TUC_KON_026 „Liefere CardSession"

Fehlercode	ErrorType	Severity	Fehlertext
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet

[<=]

Hinweis zu TAB_KON_735 - TUC_KON_026: Die WorkplaceId wird als Eingangsparameter nicht benötigt. Bereits TUC_KON_000 stellt sicher, dass eine eGK jeweils nur von einem einzigen Arbeitsplatz aus angesprochen werden kann.

4.1.5.4.2 TUC_KON_012 „PIN verifizieren"

TIP1-A_4567 - TUC_KON_012 „PIN verifizieren"
Der Konnektor MUSS den technischen Use Case „PIN verifizieren" gemäß TUC_KON_012 umsetzen.

Tabelle 64: TAB_KON_087 – TUC_KON_012 „PIN verifizieren"

Element	Beschreibung
Name	TUC_KON_012 „PIN verifizieren"
Beschreibung	Dieser Use Case führt die Verifikation einer PIN einer Karte durch. Dabei wird der Anwender am Display des Kartenterminals aufgefordert, die PIN einzugeben. Dies erfolgt am PIN-Pad des Kartenterminals. Remote-PIN-Eingabe wird dabei automatisch unterstützt.
Auslöser	• Aufruf des Use Case durch Basisdienste des Konnektors • Aufruf des Use Cases durch ein Fachmodul im Konnektor

	• Aufruf der Operation VerifyPin des CardService (siehe 4.1.5.5.1) durch das Clientsystem.
Vorbedingungen	Karte unterstützt die übergebene pinRef
Eingangsdaten	• cardSession (Kartensitzung der Karte, deren PIN verifiziert werden soll) • workplaceId • pinRef (Referenz auf die zu verifizierende PIN, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps.) • actionName – *optional/verpflichtend, wenn cardType = eGK* (Zeichenkette, max. 32 bzw. 22 Zeichen PIN.AMTS_REP mit dem Namen der zugreifenden Fachanwendung bzw. des zu nutzenden Datenobjekts und der Zugriffsart, die mit dieser PIN freigeschaltet werden soll, z. B. für MRPIN.NFD: actionName = „Notfalldaten schreiben"; Positionen in der Zeichenkette, an denen ein Zeilenumbruch bei der Ausgabe am Kartenterminal erlaubt ist, werden mit `0x0B` gekennzeichnet. `0x0B` zählt bei der Länge der Zeichenkette nicht.) • verificationType [Mandatorisch | Sitzung] (Art der PIN-Verifikation: • Mandatorisch: PIN wird immer verifiziert. • Sitzung: PIN wird nicht erneut verifiziert, falls dies für die cardSession zuvor bereits geschehen ist und der dadurch erreichte Sicherheitszustand nicht zurückgesetzt wurde.)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• pinResult [PinResult] (Ergebnis der PIN-Verifikation) • leftTries – *optional/verpflichtend, wenn pinResult = REJECTED* (Anzahl der verbleibenden Versuche)
Standardablauf	1. Ermittle Card = CM_CARD_LIST(CardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Wenn PinTyp(pinRef) = PIN.QES oder VerificationType = Mandatorisch 6. 4. Wenn pinRef in CARDSESSION.AUTHSTATE vorhanden: pinResult = OK; 5. Prüfe TUC_KON_022 „Liefere PIN-Status" a. „VERIFYABLE"; b. „DISABLED": pinResult = OK; 6. Ermittle PinInputKT: Wenn Card.ctId ein dem Arbeitsplatz(workplaceId) lokal zugeordnetes Kartenterminal ist

	(siehe Relation [6], Kapitel 4.1.1.1)
	a. Setze PinInputKT = Card.CtID
	b. sonst „lokales Kartenterminal, das für die Remote-PIN-Eingabe zu verwenden ist“: PinInputKT = Arbeitsplatz(workplaceId).remote-PIN-KT(mandantId)
	7. Atomare Operation: PIN verifizieren inkl. Eventing und Ergebnisvermerk
	a. Rufe TUC_KON_256 {
	topic = „CARD/PIN/VERIFY_STARTED“;
	eventType = Op;
	severity =Info;
	parameters = („CardHandle=$, CardType=$, ICCSN=$, CtID=$, SlotID=$, PinRef=$, PinInputCtID=$PinInputKT“,
	doLog=false)}
	b. Pin-Verifikation über „Perform Verification“ ([SICCT]) mit
	Display Messages gemäß Kontext in TAB_KON_090
	Terminalanzeigen beim Eingeben der PIN am Kartenterminal, bei
	eGK ersetze „ANW“ durch actionName in Display Message.
	Wenn PinInputKT=Card.CtID dann PIN Verifikation direkt an
	Card.CtID, ansonsten Remote-PIN-Eingabe gemäß (TIP1-A_5012)
	c. Setze pinResult in Abhängigkeit von Ergebnis Perform
	Verification:
	- pinResult = OK für erfolgreiche Prüfung
	- pinResult = ERROR für Nutzer-Abbruch oder Bearbeitungsfehler (siehe Fehlerfälle)
	- pinResult = REJECTED für falsche PIN; leftTries = x (bei Kartenantwort ´63 Cx´, x > 0)
	- pinResult = BLOCKED für gesperrte PIN (bei Kartenantwort ´63 C0´)
	d. Rufe TUC_KON_256 {
	topic = „CARD/PIN/VERIFY_FINISHED“;
	eventType = Op;
	severity = Info;
	parameters = („CardHandle=$, CardType=$, ICCSN=$, CtID=$, SlotID=$, PinRef=$, PinInputCtID=$PinInputKT, Result=$pinResult“);
	doLog = false }
	e. befülle CARDSESSION.AUTHSTATE mit pinRef und Ergebnis der PIN-Prüfung
	8. Liefere pinResult zurück

Varianten/ Alternativen	Schritt 7e: Für eGK G2.0 wird der Zustand der MRPINs nicht in AuthState gespeichert.
Fehlerfälle	Schritt 7 wird mit allen Teilschritten durchlaufen. Ein als Fehler ausgewiesener Zustand führt erst nach Schritt 7e zum Abbruch des TUCs. Fehleingaben zählen explizit nicht zu den Fehlerzuständen, sondern werden auf das Ergebnis REJECTED

	oder BLOCKED abgebildet. * Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→5) Rückgabewert= - VERIFIED, Fehlercode 4001 - TRANSPORT_PIN oder EMPTY_PIN, Fehlercode 4065 - BLOCKED, Fehlercode 4063 (→6b) kein Remote-PIN-KT zugeordnet, Fehlercode 4092 (->6b) Card.TYP=eGK und Card.CtID ist nicht dem durch workplaceId bezeichneten Arbeitsplatz lokal zugeordnet: Fehlercode 4053 (→7) Timeout bei PIN Eingabe: Fehlercode 4043 (→7) Abbruch durch Nutzer: Fehlercode 4049 (→7) Sind das für die PIN-Eingabe benötigte Kartenterminal oder benötigte Teile davon (PIN Pad, Display) durch einen anderen zeitgleich im Konnektor ablaufenden Vorgang reserviert, so bricht der Use Case mit Fehler 4060 ab. (→7) Rückgabewert= - transportgeschützt (Transport-PIN oder Leer-PIN), Fehlercode 4065 (→7b) Ungültige PIN-Referenz; Fehlercode 4072 (→7b) Karte/Kartenterminal antwortet mit einer spezifischen Fehlermeldung, Fehlercode <gemäß [gemSpec_COS]/[SICCT]> Zusätzliche Fehlerfälle ergeben sich aus der Remote-PIN-Eingabe gemäß (TIP1-A_5012)
Nichtfunktionale Anforderungen	
Zugehörige Diagramme	Abbildung PIC_KON_111 Aktivitätsdiagramm zu „PIN verifizieren“

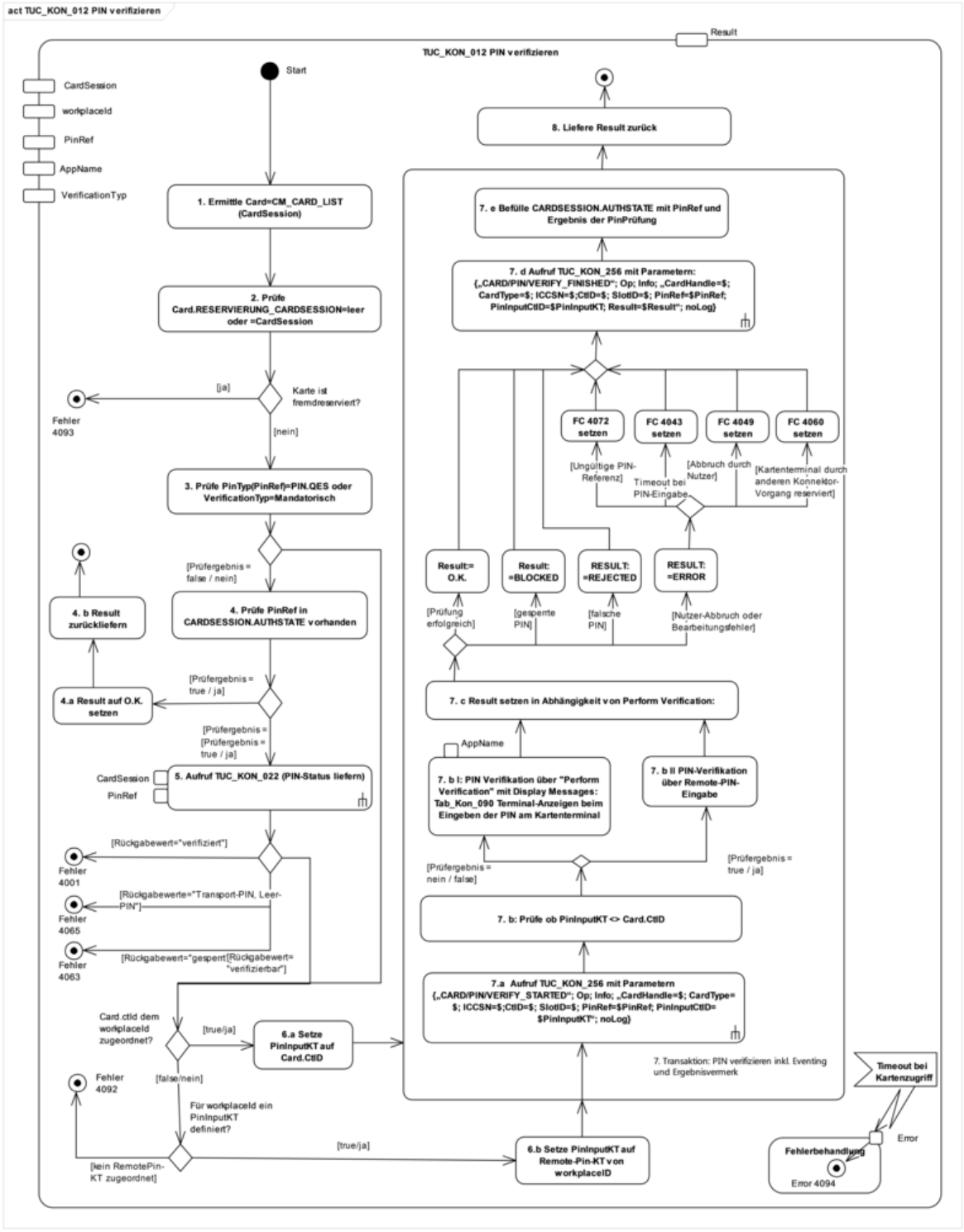


Abbildung 10: PIC_KON_111 Aktivitätsdiagramm zu „PIN verifizieren"

Tabelle 65: TAB_KON_089 Fehlercodes TUC_KON_012 „PIN verifizieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			

4001	Technical	Error	Interner Fehler
4043	Technical	Warning	Timeout bei der PIN-Eingabe
4049	Technical	Error	Abbruch durch den Benutzer
4053	Security	Error	Remote-PIN nicht möglich
4060	Technical	Error	Ressource belegt
4063	Security	Error	PIN bereits gesperrt (BLOCKED)
4065	Technical	Warning	PIN ist transportgeschützt, Änderung erforderlich
4072	Technical	Error	Ungültige PIN-Referenz `PinRef`
4092	Technical	Error	Remote-PIN-KT benötigt aber für diesen Arbeitsplatz nicht definiert
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.3 TUC_KON_019 „PIN ändern"

TIP1-A_4568 - TUC_KON_019 „PIN ändern"
Der Konnektor MUSS den technischen Use Case „PIN ändern" gemäß TUC_KON_019 umsetzen.

Tabelle 66: TAB_KON_736 – TUC_KON_019 „PIN ändern"

Element	Beschreibung
Name	TUC_KON_019 „PIN ändern"
Beschreibung	Dieser Use Case führt die Änderung einer PIN einer Karte durch. Dabei wird der Anwender am Display des Kartenterminals aufgefordert, alte und neue PIN einzugeben. Remote-PIN-Eingabe wird dabei automatisch unterstützt.
Auslöser	• Aufruf der Operation ChangePin des CardService (siehe 4.1.5.5.2) durch das Clientsystem. • Aufruf durch Fachmodul
Vorbedingungen	Karte unterstützt die übergebene pinRef
Eingangsdaten	• cardSession • workplaceId (Arbeitsplatz-Identifikator)

	• pinRef (Referenz auf die zu ändernde PIN, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps) • sourceCardSession – *optional/verpflichtend, wenn C2C erforderlich ist* (CardSession der Karte, die für die Card-to-Card-Authentisierung bei Änderung der PIN einer eGK der Generation 1+ verwendet werden soll.)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• pinResult [PinResult] (Ergebnis der PIN-Verifikation) • leftTries – *optional/verpflichtend, wenn pinStatus = REJECTED* (verbleibende Versuche)
Standardablauf	1. Ermittle Card = CM_CARD_LIST(CardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Prüfe TUC_KON_022 „Liefere PIN-Status“ {cardSession; pinRef}<>BLOCKED 4. Wenn pinRef=PIN.AMTS_REP, dann rufe TUC_KON_012 „PIN verifizieren“ { cardSession; workplaceId; pinRef=PIN.CH; actionName= „“; mandatorisch} 5. Wenn Card.TYP=eGK UND Card.Version=Generation1+, dann Aufruf TUC_KON_005 „Card-to-Card authentisieren“{ sourceCardSession; targetCardSession=cardSession; AuthMode =einseitig}. Falls keine sourceCardSession angegeben ist, kann die CardSession der für den Mandanten verwalteten SMC-B verwendet werden. 6. Ermittle PinInputKT: Wenn Card.ctId ein dem Arbeitsplatz (workplaceId) lokal zugeordnetes Kartenterminal ist (siehe Relation [6], Kapitel 4.1.1.1) a. Setze PinInputKT = Card.CtID b. sonst „lokales Kartenterminal, das für die Remote-PIN-Eingabe zu verwenden ist“: PinInputKT = Arbeitsplatz(workplaceId).remote-PIN-KT(mandantId) 7. Atomare Operation: PIN ändern inkl. Eventing und Ergebnisvermerk a. Rufe TUC_KON_256 { topic = „CARD/PIN/CHANGE_STARTED“;

	eventType = Op;
	severity = Info;
	parameters = („CardHandle=$, CardType=$,
	ICCSN=$, CtID=$, SlotID=$, PinRef=$,
	PinInputCtID=$PinInputKT“);
	doLog = false }
	b. Pin-Änderung über „MODIFY VERIFICATION DATA“ ([SICCT]) mit Display Messages entsprechend TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal. Bei Änderung der Versicherten-PIN der eGK ist dabei der Platzhalter „ANW“ durch den String „Änderung“ zu ersetzen. Der Platzhalter "#UVW-XYZ" entfällt für die PIN.QES des HBA.
	Wenn PinInputKT=Card.CtID, dann PIN-Änderung direkt an Card.CtID, ansonsten Remote-PIN-Eingabe gemäß (TIP1-A_5012)
	Dabei sowohl Unterstützung normaler PIN-Änderung als auch Umsetzens eines Transportschutzes (alle Varianten gemäß Kartenspec sind zu unterstützen)
	c. Setze pinResult in Abhängigkeit von Ergebnis MODIFY VERIFICATION DATA:
	- pinResult = OK für erfolgreiche Änderung
	- pinResult = ERROR für Nutzer-Abbruch oder Bearbeitungsfehler (siehe Fehlerfälle)
	pinResult = REJECTED für falsche PIN-Eingaben;
	leftTries = x
	(bei Kartenantwort ´63 Cx´, x > 0)
	- pinResult = BLOCKED für gesperrte PIN (bei Kartenantwort ´63 C0´)
	d. Rufe TUC_KON_256 {
	topic = „CARD/PIN/CHANGE_FINISHED“;
	eventType = Op;
	severity = Info;
	parameters = („CardHandle=$, CardType=$,
	ICCSN=$;CtID=$, SlotID=$, PinRef=$,
	PinInputCtID=$PinInputKT,
	Result=pinStatus“);
	doLog = false}
	e. Wenn Result = REJECTED oder BLOCKED , dann entferne PinRef aus CARDSESSION.AUTHSTATE
	8. Liefere pinResult und ggf. leftTries zurück

Varianten/ Alternativen	Schritt 4: Für eGK G2.0 gilt:
	Wenn pinRef=PIN.AMTS_REP, dann
	rufe TUC_KON_012 „PIN verifizieren“ {
	cardSession;
	workplaceId;
	pinRef=MRPIN.AMTS;
	actionName= „“;
	mandatorisch}

	Schritt 7e: Für eGK G2.0 wird der Zustand der MRPINs nicht in AuthState gespeichert.
Fehlerfälle	Schritt 7 wird mit allen Teilschritten durchlaufen. Ein als Fehler ausgewiesener Zustand führt erst nach Schritt 7e zum Abbruch des TUCs. * Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→3) pinStatus=BLOCKED: Fehlercode 4063 (→5) sourceCardSession benötigt aber leer, Fehlercode 4071 (→6b) kein Remote-PIN-KT zugeordnet, Fehlercode 4092 (->6b) Card.TYP=eGK und Card.CtID ist nicht dem durch workplaceId bezeichneten Arbeitsplatz lokal zugeordnet: Fehlercode 4053 (→7b) neue PIN zu kurz/lang: Fehlercode 4068 (→7b) zweite neue PIN<> erste neue PIN: Fehlercode 4067 (→7b) Timeout bei PIN-Eingabe: Fehlercode 4043. (→7b) Abbruch durch Nutzer: Fehlercode 4049. (→7b) Ist das Kartenterminal oder Teile davon (PIN-Pad, Display) durch einen anderen Vorgang reserviert: Fehlercode 4060 (→7b) kein PIN-Pad am Kartenterminal verfügbar: Fehlercode 4066 (→7b) Ungültige PIN-Referenz; Fehlercode 4072 (→7b) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]> Zusätzliche Fehlerfälle ergeben sich aus der Remote-PIN-Eingabe gemäß (TIP1-A_5012)
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 67: TAB_KON_093 Fehlercodes TUC_KON_019 „PIN ändern"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4043	Technical	Warning	Timeout bei der PIN-Eingabe
4049	Technical	Error	Abbruch durch den Benutzer
4053	Security	Error	Remote-PIN nicht möglich
4060	Technical	Error	Ressource belegt

4063	Security	Error	PIN bereits blockiert (BLOCKED)
4066	Technical	Error	PIN Pad nicht verfügbar
4067	Security	Error	neue PIN nicht identisch
4068	Security	Error	neue PIN zu kurz/zu lang
4071	Technical	Error	keine Karte für C2C-Auth gesetzt
4072	Technical	Error	ungültige PIN-Referenz `PinRef`
4092	Technical	Error	Remote-PIN-KT benötigt aber für diesen Arbeitsplatz nicht definiert
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.4 TUC_KON_021 „PIN entsperren"

TIP1-A_4569-02 - TUC_KON_021 „PIN entsperren"
Der Konnektor MUSS den technischen Use Case „PIN entsperren" gemäß TUC_KON_021 umsetzen.

Tabelle 68: TAB_KON_236 – TUC_KON_021 „PIN entsperren"

Element	Beschreibung
Name	TUC_KON_021 „PIN entsperren"
Beschreibung	Dieser Use Case setzt den Fehlbedienungszähler für diese PIN in der Karte auf seinen Anfangswert zurück und es wird optional eine neue PIN gesetzt. Remote-PIN-Eingabe wird dabei automatisch unterstützt.
Auslöser	• Aufruf der Operation UnblockPin des CardService (siehe 4.1.5.5.4) durch das Clientsystem.
Vorbedingungen	Karte unterstützt die übergebene pinRef
Eingangsdaten	• cardSession CardSession der Karte, deren PIN entsperrt werden soll) • workplaceId

	• pinRef (Referenz auf die zu entsperrende PIN, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps) setNewPin (true/false) - Angabe, ob eine neue PIN gesetzt oder die aktuelle weiterverwendet werden soll. Default = false sourceCardSession - *optional/wenn eGK G1+* (CardSession der Karte, die für die Card-to-Card-Authentisierung bei Entsperrung der PIN einer eGK der Generation 1+ verwendet werden soll)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• result [PukResult]) (Ergebnis der PIN-Entsperrung durch PUK-Eingabe) • leftTries – *optional/verpflichtend, wenn pukStatus = REJECTED* (verbleibende Versuche des PUKs)
Standardablauf	1. Ermittle Card = CM_CARD_LIST(Target.CardHandle) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Wenn TUC_KON_022 „Liefere PIN-Status“ { cardSession; pinRef } <>(„BLOCKED“ oder "TRANSPORT_PIN") dann beende TUC erfolgreich. 4. Wenn pinRef=PIN.AMTS_REP, dann a. setNewPin = true b. rufe TUC_KON_012 „PIN verifizieren“ { cardSession; workplaceId; pinRef=PIN.CH; actionName= „“; mandatorisch} 5. Wenn Card.TYP=eGK UND Card.Version=Generation1+, dann Aufruf TUC_KON_005 „Card-to-Card authentisieren“ { sourceCardSession; targetCardSession=cardSession; AuthMode =einseitig }. Falls keine sourceCardSession angegeben ist, kann die CardSession der für den Mandanten verwalteten SMC-B verwendet werden. 6. Ermittle PinInputKT: Wenn Card.ctId ein dem Arbeitsplatz(workplaceId) lokal zugeordnetes Kartenterminal ist (siehe Relation [6], Kapitel 4.1.1.1) a. Setze PinInputKT = Card.CtID

b. sonst „lokales Kartenterminal, das für die Remote-PIN-Eingabe zu verwenden ist“: PinInputKT = Arbeitsplatz(workplaceId).remote-PIN-KT(mandantId)

7. Atomare Operation: PIN entsperren inkl. Eventing und Ergebnisvermerk

a. Rufe TUC_KON_256 {
topic = „CARD/PIN/CHANGE_STARTED“;
eventType = Op;
severity = Info;
parameters = („CardHandle=$, CardType=$,
ICCSN=$, CtID=$; SlotID=$, PinRef=$,
PinInputCtID=$PinInputKT“);
doLog=false}

b. PIN-Entsperrung mit Display Messages entsprechend TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal.
Wenn PinInputKT=Card.CtID, dann PIN-Änderung direkt an Card.CtID, ansonsten Remote-PIN-Eingabe gemäß (TIP1-A_5012)

- Für pinRef == PIN.QES über „PERFORM VERIFICATION“ [SICCT] mit dem eingebetteten Kommando Reset Retry Counter in der Variante P1=1 (keine neue PIN setzen).
- Für pinRef<>PIN.QES
wenn setNewPin = false,
dann über PERFORM VERIFICATION“ [SICCT],
sonst über „MODIFY VERIFICATION DATA“ [SICCT].
Das mit dem SICCT-Kommando als Command-To-Perform mitgesandte „Reset Retry Counter“ wird entsprechend dem Wert von setNewPIN parametrisiert.

c. Setze result in Abhängigkeit von Ergebnis Perform Verification bzw. Modify VerificationData:

- result = OK für erfolgreiche Entsperrung
- result = ERROR für Nutzer-Abbruch oder Bearbeitungsfehler (siehe Fehlerfälle)
- result = REJECTED für falsche PUK;
- result = BLOCKED für gesperrte PUK; (bei Kartenantwort ´63 C0´)

d. Rufe TUC_KON_256 {
topic=„CARD/PIN/CHANGE_FINISHED“;
eventType=Op; severity=Info;
parameters = („CardHandle=$; CardType=$;
ICCSN=$;CtID=$; SlotID=$; PinRef=$;
PinInputCtID=$PinInputKT; Result=$“);
doLog=false }

8. Liefere result und ggf. leftTries zurück

Varianten/ Alternativen	Schritt 4: Für eGK G2.0 gilt:
	Wenn pinRef=PIN.AMTS_REP, dann
	rufe TUC_KON_012 „PIN verifizieren“ {
	cardSession;
	workplaceId;
	pinRef=MRPIN.AMTS;
	actionName= „“;
	mandatorisch}

Fehlerfälle	Schritt 7 wird mit allen Teilschritten durchlaufen. Ein als Fehler ausgewiesener Zustand führt erst nach Schritt 7d zum Abbruch des TUCs. * Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte wird in einer anderen Kartensitzung exklusiv verwendet, Fehlercode 4093 (→5) sourceCardSession benötigt aber leer, Fehlercode 4071 (→6b) kein Remote-PIN-KT zugeordnet, Fehlercode 4092 (→6b) Card.TYP=eGK und Card.CtID ist nicht dem durch workplaceId bezeichneten Arbeitsplatz lokal zugeordnet: Fehlercode 4053 (→7b) blockierte PUK: Fehlercode 4064 (→7b) neue PIN zu kurz/lang: Fehlercode 4068 (→7b) zweite neue PIN<> erste neue PIN: Fehlercode 4067 (→7b) Timeout bei PIN Eingabe: Fehlercode 4043. (→7b) Abbruch durch Nutzer: Fehlercode 4049. (→7b) Ist das Kartenterminal oder Teile davon (PIN-Pad, Display) durch einen anderen Vorgang reserviert: Fehlercode 4060 (→7b) Karte/Kartenterminal antwortet mit einer spezifischen Fehlermeldung, Fehlercode <gemäß [gemSpec_COS]/[SICCT]> (→7b) Ungültige PIN-Referenz; Fehlercode 4072. Zusätzliche Fehlerfälle ergeben sich aus der Remote-PIN-Eingabe gemäß (TIP1-A_5012)
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 69: TAB_KON_193 Fehlercodes TUC_KON_021 „PIN entsperren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			

4043	Technical	Warning	Timeout bei der PIN-Eingabe
4049	Technical	Error	Abbruch durch den Benutzer
4053	Security	Error	Remote-PIN nicht möglich
4060	Technical	Error	Ressource belegt
4064	Security	Error	alte PIN bereits blockiert (hier: PUK)
4067	Security	Error	neue PIN nicht identisch
4068	Security	Error	neue PIN zu kurz/zu lang
4072	Technical	Error	ungültige PIN-Referenz `PinRef`
4092	Technical	Error	Remote-PIN-KT benötigt aber für diesen Arbeitsplatz nicht definiert
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.5 TUC_KON_022 „Liefere PIN-Status"

TIP1-A_4570 - TUC_KON_022 „Liefere PIN-Status"
Der Konnektor MUSS den technischen Use Case „Liefere PIN-Status" gemäß TUC_KON_022 umsetzen.

Tabelle 70 TAB_KON_532 – TUC_KON_022 „Liefere PIN-Status"

Element	Beschreibung

Name	TUC_KON_022 „Liefere PIN-Status“
Beschreibung	Dieser Use Case prüft den Zustand eines PIN-Objekts einer Karte im Kontext einer CardSession.
Auslöser	• Aufruf des Use Case im Rahmen von technischen Use Cases der Basisdienste des Konnektors • Aufruf des Use Cases durch ein Fachmodul im Konnektor • Aufruf der Operation GetPinStatus des CardService (siehe 4.1.5.5.1) durch das Clientsystem.
Vorbedingungen	Karte unterstützt die übergebene pinRef
Eingangsdaten	• cardSession • pinRef (Pin-Referenz der angefragten PIN, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• pinStatus [PinStatus] • leftTries – *optional/verpflichtend, wenn pinStatus = VERIFYABLE* (Anzahl der verbleibenden Versuche für die Verifikation der PIN)
Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. pinRef in CardSession.AUTHSTATE vorhanden: a) Ja: Setze pinStatus = VERIFIED oder DISABLED (wie in AUTHSTATE) b) Nein:Aufruf der Kartenoperation „GET PIN STATUS“, Antwort der Karte wird ausgewertet: a. ´90 00´ (NoError: Verifiziert): pinStatus = VERIFYABLE (da nicht in dieser CardSession verifiziert) b. ´62 C1´: pinStatus = TRANSPORT_PIN c. ´62 C7´: pinStatus = EMPTY_PIN (Leer-PIN) d. ´63 Cx´: pinStatus = VERIFYABLE (mit 1<= x <= 3); LeftTries=x e. ´63 C0´: pinStatus = BLOCKED; leftTries=0 f. ´62 D0´: pinStatus = DISABLED (Verifikation nicht erforderlich, da PIN-Schutz ausgeschaltet); cardSession.AUTHSTATE aktualisieren g. Antwortet die Karte mit einer Fehlermeldung, bricht der TUC ab.

	Liefere leftTries nur in den Fällen d und e zurück.
Varianten/ Alternativen	
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→3b) pinRef nicht gefunden: Fehlercode 4072
Zugehörige Diagramme	keine

Tabelle 71: TAB_KON_091 Fehlercodes TUC_KON_022 „Liefere PIN-Status"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4072	Technical	Error	ungültige PIN-Referenz `PinRef`
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.6 TUC_KON_027 „PIN-Schutz ein-/ausschalten"

TIP1-A_5486 - TUC_KON_027 „PIN-Schutz ein-/ausschalten"
Der Konnektor MUSS den technischen Use Case TUC_KON_027 „PIN-Schutz ein-/ausschalten" umsetzen.

Tabelle 72: TAB_KON_240 - TUC_KON_027 „PIN-Schutz ein-/ausschalten"

Element	Beschreibung
Name	TUC_KON_027 „PIN-Schutz ein-/ausschalten"
Beschreibung	Schaltet das Erfordernis, die PIN zu verifizieren, ein bzw. aus. Diese Operation wird nur unterstützt für PINs der EGK G2 gemäß [gemSpec_eGK_ObjSys]; für sie können folgende Kommandos auf das Passwortobjekt angewendet werden: • DISABLE VERIFICATION REQUIREMENT • ENABLE VERIFICATION REQUIREMENT
Auslöser	• Aufruf durch ein Fachmodul • Aufruf der Operationen EnablePin und DisablePin des CardService durch das Clientsystem.
Vorbedingungen	Karte unterstützt die übergebene pinRef

Eingangsdaten	• cardSession (CardSession einer EGK G2) • pinRef (PIN-Referenz der ab-/anzuschaltenden PIN, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps) • enable [Boolean] (enable = true: Erfordernis der Benutzerverifikation einschalten; enable = false: Erfordernis der Benutzerverifikation abschalten)
Komponenten	Konnektor
Ausgangsdaten	• pinResult [PinResult] (Ergebnis von PIN-Schutz ein-/ausschalten durch PIN-Eingabe) • leftTries – *optional/verpflichtend nach fehlerhafter PIN* (verbleibende Versuche)

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession)
	2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer im Besitz des Karten-Locks ist.
	3. Prüfe Card.Type = EGK und Generation ≥ 2
	
	4. Prüfe pinRef = MRPIN.AMTS und Card.Type = EGK und Generation > 2.0
	5. Wenn enable
	A: =true:
	Atomare Operation: PIN bearbeiten inkl. Eventing und Ergebnisvermerk
	a. Rufe TUC_KON_256 {
	topic = „CARD/PIN/ENABLE_STARTED“;
	eventType = Op;
	severity = Info;
	parameters = („CardHandle=$, CardType=$, ICCSN=$, CtID=$, SlotID=$, PinRef=$, PinInputCtID=$PinInputKT“);
	doLog = false }
	b. Aufruf des Kartenterminalkommandos „SICCT PERFORM VERIFICATION“ mit der Kartenoperation „ENABLE VERIFICATION REQUIREMENT” als Command-To-Perform. Es ist der Parameter P1=’00’ (mit Benutzerverifikation) zu verwenden. Die Anzeige am KT erfolgt entsprechend TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal. Ersetze in displayMessage „ANW“ entsprechend ANW(pinRef) gemäß Tabelle TAB_KON_838.
	c. Setze pinResult in Abhängigkeit von Ergebnis Perform Verification:
	- pinResult = OK für erfolgreiche Änderung
	- pinResult = ERROR für Nutzer-Abbruch oder Bearbeitungsfehler (siehe Fehlerfälle)
	- pinResult = REJECTED für falsche PIN;
	leftTries = x (bei Kartenantwort ´63 Cx´, x > 0)

	- pinResult = BLOCKED für gesperrte PIN (bei Kartenantwort ´63 C0´)
	d. Rufe TUC_KON_256 {
	topic =„CARD/PIN/ENABLE_FINISHED“;
	eventType = Op;
	severity = Info;
	parameters = („CardHandle=$, CardType=$, ICCSN=$, CtID=$, SlotID=$, PinRef=$, PinInputCtID=$PinInputKT”);
	doLog = false }
	B: =false:
	Atomare Operation: PIN bearbeiten inkl. Eventing und Ergebnisvermerk
	a. Rufe TUC_KON_256 {
	topic = „CARD/PIN/DISABLE_STARTED“;
	eventType = Op;
	severity = Info;
	parameters = („CardHandle=$, CardType=$, ICCSN=$, CtID=$, SlotID=$, PinRef=$, PinInputCtID=$PinInputKT“);
	doLog = false }
	b. Aufruf des Kartenterminalkommandos „SICCT PERFORM VERIFICATION“ mit der Kartenoperation „DISABLE VERIFICATION REQUIREMENT” als Command-To-Perform. Es ist der Parameter P1=’00’ (mit Benutzerverifikation) zu verwenden. Die Anzeige am KT erfolgt entsprechend TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal. Ersetze in displayMessage „ANW“ entsprechend ANW(pinRef) gemäß Tabelle TAB_KON_838.
	c. Setze pinResult in Abhängigkeit von Ergebnis Perform Verification:
	- pinResult = OK für erfolgreiche Änderung
	- pinResult = ERROR für Nutzer-Abbruch oder Bearbeitungsfehler (siehe Fehlerfälle)
	- pinResult = REJECTED für falsche PIN; leftTries = x (bei Kartenantwort ´63 Cx´, x > 0)
	- pinResult = BLOCKED für gesperrte PIN
	
	d. Rufe TUC_KON_256 {
	topic = „CARD/PIN/DISABLE_FINISHED“;
	eventType = Op;
	severity = Info;
	parameters = („CardHandle=$, CardType=$, ICCSN=$;CtID=$, SlotID=$, PinRef=$, PinInputCtID=$PinInputKT”);
	doLog=false}
	6. Liefere pinResult und leftTries zurück

Varianten/ Alternativen	(->3) zur Optimierung kann vor Schritt 5 der PIN-Schutz geprüft werden: a. pinStatus=TUC_KON_022 „Liefere PIN-Status" { cardSession; pinRef } b. Wenn pinStatus<>DISABLED und enable=true, dann pinResult=OK und -> weiter in Schritt 6 c. Wenn pinStatus=DISABLED und enable=false, dann pinResult=OK und -> weiter in Schritt 6
Fehlerfälle	(→2) Karte ist fremd reserviert: Fehlercode 4093 (→3) Karte ist keine eGK der Generation 2 oder höher: Fehlercode 4209 (→4) PIN nicht gefunden; Karte ist eGK G2.0: Die Operation „PIN-Schutz ein-/ausschalten" wird für MRPIN.AMTS nicht unterstützt: Fehlercode 4072 (→5) Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden: Fehlercode 4094 (→5) PIN nicht gefunden: Fehlercode 4072 (→5) PIN gesperrt: Fehlercode 4063 (→5) Zugriffsbedingung nicht erfüllt (PIN nicht abschaltbar): Fehlercode 4085
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	keine

Tabelle 73: TAB_KON_838 Mapping von pinRef auf ANW

pinRef	ANW (max. 16 Zeichen)
MRPIN.NFD	Notfalldaten
MRPIN.DPE	Pers.Erklärungen
MRPIN.AMTS	Medikationsdaten
MRPIN.GDD	PIN•GDD

Hinweis zu TAB_KON_838: Leerzeichen werden als "•" dargestellt.

Tabelle 74: TAB_KON_241 Fehlercodes TUC_KON_027 „PIN-Schutz ein/ausschalten"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			

4063	Security	Error	PIN bereits blockiert (BLOCKED)
4072	Technical	Error	ungültige PIN-Referenz `PinRef`
4085	Security	Error	Zugriffsbedingung nicht erfüllt
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=]

4.1.5.4.7 TUC_KON_023 „Karte reservieren"

TIP1-A_4571 - TUC_KON_023 „Karte reservieren"
Der Konnektor MUSS den technischen Use Case „Karte reservieren" gemäß TUC_KON_023 umsetzen.

Tabelle 75: TAB_KON_533 - TUC_KON_023 „Karte reservieren"

Element	Beschreibung
Name	TUC_KON_023 „Karte reservieren" Dem Aufrufer des TUC_KON_023 wird beim Reservieren (DoLock=Ja) der Karte zur ausschließlichen Nutzung ein Lock zugeordnet. Wird der TUC-KON_023 mit diesem Lock zum Freigeben der Reservierung (DoLock=Nein) aufgerufen, dann erlischt das Lock und die ausschließliche Nutzung wird beendet. Der Scope der Kartenreservierung wird vom Aufrufer des TUC_KON_023 gesteuert. Das Lock ist Konnektor-intern. Es darf nicht außerhalb des Konnektors referenzierbar sein. Zwei verschiedene Operationsaufrufe am Konnektor dürfen nie ein identisches Lock haben. Der Konnektor MUSS sicherstellen, dass auch im Fehlerfall die Reservierung zu einem Lock aufgehoben wird. Ein Lock darf nicht dauerhaft bestehen.
Beschreibung	Reservierung der Karte
Auslöser	• Aufruf des Use Case im Rahmen von technischen Use Cases der Basisdienste des Konnektors • Aufruf des Use Cases durch ein Fachmodul im Konnektor
Vorbedingungen	Keine
Eingangsdaten	• cardSession

	• doLock [Boolean] (Zielzustand der Karte; true = reserviert, false = freigegeben)
Komponenten	Konnektor
Ausgangsdaten	Keine
Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Wenn doLock A: = true: i. Prüfe, dass der zur cardSession gehörenden Karte kein Lock zugeordnet ist ii. Dem Aufrufer wird ein Lock auf die zur cardSession gehörende Karte zugeordnet. Es wird nicht explizit als Ausgangsdatum modelliert, sondern der Aufrufer hat das Lock durch die Zuordnung, muss es aber nicht verwalten. B: = false: i. Prüfe, dass der Aufrufer für die zur cardSession gehörende Karte ein Lock hat. ii. Das der Karte zugeordnete Lock wird gelöscht.
Varianten/ Alternativen	Keine
Fehlerfälle	(→2Ai) Karte bereits reserviert, Fehlercode 4093 (→2Bi) Karte nicht durch Aufrufer reserviert, Fehlercode 4001
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 76: TAB_KON_534 Fehlercodes TUC_KON_023 „Karte reservieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4001	Technical	Error	interner Fehler
4093	Technical	Error	Karte bereits reserviert

[<=]

4.1.5.4.8 TUC_KON_005 „Card-to-Card authentisieren"

Die C2C-Authentisierung erfolgt konform zu den in [gemSpec_COS#15] festgelegten Authentisierungsprotokollen.

Definition Quellkarte/Zielkarte:

Bei einseitiger Card-to-Card-Authentisierung ohne Aushandlung eines Session Key ist die Quellkarte diejenige, die die Rolle des Karteninhabers bzw. der Organisation gemäß [gemSpec_PKI_TI#Tab_PKI_254] gegenüber der anderen Karte nachweist, z. B. der HBA bei der Freischaltung einer eGK.

Bei gegenseitiger Card-to-Card-Authentisierung ohne Aushandlung eines Session Key erfolgen nach einander zwei einseitige Card-to-Card-Authentisierungen mit vertauschten Rollen. Quell- und Zielkarte habe daher für den Gesamtablauf keine nähere Bedeutung.

Bei Card-to-Card-Authentisierung mit Aushandlung eines Session Key ist die Quellkarte diejenige, die die SM-APDUs produzieren kann, also die SMC (-KT oder -K).

Die Zielkarte ist jeweils die Karte, die nicht die Quellkarte ist.

TIP1-A_4572 - TUC_KON_005 „Card-to-Card authentisieren"
Der Konnektor MUSS den technischen Use Case „Card-to-Card authentisieren" gemäß TUC_KON_005 umsetzen.
Die Card-to-Card-Authentisierung zwischen zwei Karten, bei der eine Karte der Generation 1+ angehört MUSS das RSA-Verfahren verwenden.
Die Card-to-Card-Authentisierung zwischen zwei Karten der Generation 2 MUSS das Verfahren der elliptischen Kurven verwenden.

Tabelle 77: TAB_KON_096 – TUC_KON_005 „Card-to-Card authentisieren"

Element	Beschreibung
Name	TUC_KON_005 „Card-to-Card authentisieren"
Beschreibung	Durchführung einer Card-to-Card-Authentisierung
Auslöser	• Aufruf des Use Case im Rahmen von technischen Use Cases der Basisdienste des Konnektors • Aufruf des Use Cases durch ein Fachmodul im Konnektor
Vorbedingungen	Wert von Source_CARDSESSION.AUTHSTATE: wenn Quellkarte a) ein HBA ist: CHV; PIN.CH, verifiziert b) eine SMC-B ist: CHV; PIN.SMC verifiziert
Eingangsdaten	• sourceCardSession (Quellkarte) • targetCardSession (Zielkarte) • authMode (gemäß Tabelle TAB_KON_673)
Komponenten	Karten, Konnektor, Kartenterminal
Ausgangsdaten	Keine
Standardablauf	1. Ermittle sCard = CM_CARD_LIST(sourceCardSession) 2. Ermittle tCard = CM_CARD_LIST(targetCardSession) 3. Prüfe, dass der Quellkarte entweder kein Lock zugeordnet ist oder der Aufrufer im Besitz auf das Lock der Quellkarte ist. Prüfe, dass der Zielkarte entweder kein Lock zugeordnet

	ist oder der Aufrufer im Besitz auf das Lock der Zielkarte ist. 4. Prüfe Aufrufparameter auf erlaubte Kombination gemäß Tabelle TAB_KON_674 5. Wenn das zu verwendende CV-Zertifikat der Quellkarte ein CV-Zertifikat der Generation 2 oder höher ist, dann prüfe sein Ausstellungsdatum (CED) gegen die aktuelle Zeit 6. Wenn Card.TYP=eGK UND Card.Version=Generation1+, dann prüfe, ob aktuelles System-Datum < 01.01.2019 ist 7. Wähle Key-Referenzen gemäß Tabelle TAB_KON_674 8. Prüfe pinRef/keyRef in sCard.CARDSESSION.AUTHSTATE und tCard.CARDSESSION.AUTHSTATE für adressierte Schlüssel wie in Zugriffsbedingung der Karten definiert vorhanden 9. Durchführung der Authentisierung gemäß Tabelle TAB_KON_673 mit Key-Referenzen gemäß Tabelle TAB_KON_674 10. Ergänze targetCardSession.AUTHSTATE mit tKeyRef und Rolle aus sKeyRef (CHA bzw. CHAT aus dem EndEntity-CV-Zertifikat der Quellkarte)
Varianten/ Alternativen	(→9) Wenn der für die CA-Zertifikatsprüfung zu selektierende CVC-Root-Key auf der Zielkarte nicht vorhanden ist (Returncode des Kartenkommandos „MANAGE SECURITY ENVIRONMENT" ist '6A 88'), dann muss der Konnektor: a) das oder die passenden Cross-CV-Zertifikate aus dem Truststore auswählen b) mit dem Kartenkommando „PSO Verify Certificate" jedes ausgewählte Cross-CV-Zertifikat durch die Zielkarte prüfen lassen. Dadurch wird der im Cross-CV-Zertifikat enthaltene öffentliche Schlüssel an die Zielkarte übertragen. Die Zielkarte speichert den darin enthaltenen neuen CVC-Root-Key. c) den neuen CVC-Root-Key auf der Zielkarte selektieren d) den Standardablauf der C2C-Authentisierung fortsetzen (→9) Wenn tCard.TYPE=EGK und AuthMode=gegenseitig, dann Echtheitsprüfung der eGK durch den Konnektor: a) Freischaltung der EGK durch den HBA/die SMC-B: Durchführen der Authentisierung gemäß Tabelle TAB_KON_673 mit Key-Referenzen gemäß Tabelle TAB_KON_674 aber mit AuthMode=einseitig b) Konnektor liest das CA-Zertifikat EF.C.CA_eGK.CS (G1+) bzw. C.CA_eGK.CS.E256 (G2) c) Konnektor liest das End-Entity-Zertifikat der EGK EF.C.eGK.AUT_CVC (G1+) bzw. EF.C.eGK.AUT_CVC.E256 (G2) d) Konnektor prüft das CVC-EE-Zertifikat mit TUC_KON_042

	„CV-
	Zertifikat prüfen“ {
	certificate = C.eGK.AUT_CVC/C.eGK.AUT_CVC.E256;
	caCertificate = C.CA_eGK.CS/C.CA_eGK.CS.E256 }
	e) Konnektor erzeugt Zufallszahl
	f) Konnektor selektiert den PrK.eGK.AUT_CVC (G1+) bzw. PrK.eGK.AUT_CVC.E256 (G2) und stellt abhängig von der Version
	der eGK den Algorithmus auf der eGK ein (MSE Set)
	g) Konnektor sendet Konkatenation aus Zufallszahl und CARD.ICCSN mit dem Befehl „INTERNAL AUTHENTICATE“ an die
	eGK
	h) Konnektor wertet das von der Karte erhaltene Chiffrat aus

Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→3) Eine Karte ist fremd reserviert, Fehlercode 4093 (→5) Zertifikat der Quellkarte fehlerhaft. Ausstellungsdatum liegt in der Zukunft; Fehlercode 4233 (→6) eGK G1+ ausgealtert, Fehlercode 4192 (→8) Nötige PIN, bzw. KeyRef ist nicht verifiziert, Fehlercode 4085 (→9) Je nachdem, welche Karte den Fehler verursachte, wird zum ursprünglichen Fehler (Fehlercode gemäß [gemSpec_COS]) im Error-Trace (welcher an erster Stelle im Falle des HBA z. B. bereits ein Fehler bezüglich PIN-Verifikation enthalten kann) noch ein weiterer mit Code 4056 oder 4057 hinzugefügt. Kann der Fehler nicht eindeutig einer der beiden Karten zugeordnet so wird Error-Code 4048 verwendet.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	Keine

Tabelle 78: TAB_KON_673 AuthMode für C2C

AuthMode	Definition des Ablaufs
einseitig	Externe oder Interne Authentisierung ([gemSpec_COS#15.1] oder [gemSpec_COS#15.2], passend zu den Zugriffsregeln der beteiligten CVC)
gegenseitig	Card-2-Card-Authentisierung ohne Sessionkey-Aushandlung ([gemSpec_COS#15.3])
gegenseitig+TC	Card-2-Card-Authentisierung mit Sessionkey-Aushandlung zur Etablierung eines Trusted Channels ([gemSpec_COS#15.4])

Tabelle 79: TAB_KON_674 Erlaubte Parameterkombinationen und resultierende CV-Zertifikate für C2C

Quellkarte	Zielkarte	AuthMode	sKeyRef	tKeyRef	Fachlicher UseCase
HBA oder SM-B	eGK G1+	einseitig	{HPC.AUTR_ CVC.R2048 | SMC.AUTR_ CVC.R2048}		Freischaltung eGK
HBA oder SM-B	eGK G1+	gegen seitig	{HPC.AUTR_ CVC.R2048 | SMC.AUTR_ CVC.R2048}	eGK.AUT_ CVC.R2048	Freischaltung eGK mit Echtheits prüfung eGK
HBA oder SM-B	eGK G2	einseitig	{HPC.AUTR_ CVC.E256 | SMC.AUTR_ CVC.E256}		Freischaltung eGK
HBA oder SM-B	eGK G2	gegen seitig	{HPC.AUTR_ CVC.E256 | SMC.AUTR_ CVC.E256}	eGK.AUT_ CVC.E256	Freischaltung eGK mit Echtheits prüfung eGK
SMC-K	HBA	gegen seitig+TC	SAK.AUTD_ CVC.E256	HPC.AUTD_ SUK_CVC.E256	DTBS-Übertragung bei QES
SMC-KT	HBA	gegen seitig+TC	SMC.AUTD_ RPS_CVC.E256	HPC.AUTD_ SUK_CVC.E256	Remote-PIN
SMC-KT	SM-B	gegen seitig+TC	SMC.AUTD_ RPS_CVC.E256	SMC.AUTD_ RPE_CVC.E256	Remote-PIN

Tabelle 80: TAB_KON_535 Fehlercodes TUC_KON_005 „Card-to-Card authentisieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4048	Technical	Error	Fehler bei der C2C-Authentisierung
4056	Technical	Error	Fehler bei der C2C-Authentisierung, Quellkarte
4057	Technical	Error	Fehler bei der C2C-Authentisierung, Zielkarte
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4192	Security	Error	C2C mit eGK G1+ ab 01.01.2019 nicht mehr gestattet
4233	Security	Error	Ausstellungsdatum des Zertifikats liegt in der Zukunft;

[<=]

4.1.5.4.9 TUC_KON_202 „LeseDatei"

TIP1-A_4573 - TUC_KON_202 „LeseDatei"
Der Konnektor MUSS den technischen Use Case „LeseDatei" gemäß TUC_KON_202 umsetzen.

Tabelle 81: TAB_KON_218 – TUC_KON_202 „LeseDatei"

Element	Beschreibung
Name	TUC_KON_202 „LeseDatei"
Beschreibung	Transparente Datei oder Teile davon lesen
Auslöser	• Aufruf des Use Case im Rahmen von technischen Use Cases der Basisdienste des Konnektors • Aufruf des Use Cases durch ein Fachmodul im Konnektor
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • offset – *optional/nur verwendbar, wenn fileIdentifier angegeben ist* (Startposition innerhalb der Datei) • length – *optional* (Längenangabe, um den Zugriff auf Teile einer Datei einzuschränken)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• content (Gelesene Daten)

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 4. Selektiere Verzeichnis und Datei 5. Lies Daten über Kartenkommando „READ BINARY" unter Berücksichtigung von Offset- und Längenangaben 6. Die gelesenen Daten werden an den Aufrufer zurückgegeben
Varianten/ Alternativen	Wenn Card.TYPE = KVK, sendet der Konnektor in diesem Fall ein "Read Binary" im Sinne von SICCT 1.2.1, 5.5.8.1 "Kommandos für synchrone Chipkarten".
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→3) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5) Verzeichnis deaktiviert, Fehlercode 4086 (→4-5) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 82: TAB_KON_536 Fehlercodes TUC_KON_202 „LeseDatei"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4086	Technical	Error	Verzeichnis deaktiviert
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.10 TUC_KON_203 „SchreibeDatei"

TIP1-A_4574 - TUC_KON_203 „SchreibeDatei„
Der Konnektor MUSS den technischen Use Case „SchreibeDatei" gemäß TUC_KON_203 umsetzen.

Tabelle 83: TAB_KON_219 – TUC_KON_203 „SchreibeDatei"

Element	Beschreibung
Name	TUC_KON_203 „SchreibeDatei"
Beschreibung	Daten in transparente Datei schreiben
Auslöser	• Aufruf durch Fachmodul
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen.
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • offset– *optional* (Startposition innerhalb der Datei, default: 0) • length – *optional* (Längenangabe, um den Zugriff auf Teile einer Datei einzuschränken; default: alles ab offset) • dataToBeWritten (Zu schreibende Daten)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	Keine

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe Card.TYPE <> KVK 3. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 4. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 5. Selektiere Verzeichnis 6. Selektiere Datei mittels SELECT mit P2='04' (Selektieren einer Datei, Antwortdaten mit FCP) 7. Ermittle size (Größe der selektierten Datei in Byte) mit size = numberOfOctet aus FCP 8. Wenn size – offset >= Größe von dataToBeWritten in Byte, dann schreibe dataToBeWritten mittels Kartenkommando "UPDATE BINARY" unter Berücksichtigung von Offset- und Längenangaben
Varianten/ Alternativen	keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Schreibzugriff auf KVK nicht gestattet, Fehlercode 4085 (→3) Karte ist fremd reserviert, Fehlercode 4093 (→4) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5) Verzeichnis oder Datei existiert nicht, Fehlercode 4087 (→4-5) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]> (→6) Ausgewählte Datei ist nicht transparent, Fehlercode 4089 (→6) Verzeichnis deaktiviert, Fehlercode 4086 (→8) dataToBeWritten sind größer als der zur Verfügung stehende Speicherplatz, Fehlercode 4247
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 84: TAB_KON_537 Fehlercodes TUC_KON_203 „Schreibe Datei"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt

4086	Technical	Error	Verzeichnis deaktiviert
4087	Technical	Error	Datei nicht vorhanden
4089	Technical	Error	Datei ist vom falschen Typ
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4247	Technical	Error	Speicherplatz auf der Karte nicht ausreichend

[<=]

4.1.5.4.11 TUC_KON_204 „LöscheDateiInhalt"

TIP1-A_5476 - TUC_KON_204 „LöscheDateiInhalt"
Der Konnektor MUSS den technischen Use Case „LöscheDateiInhalt" gemäß TUC_KON_204 umsetzen.

Tabelle 85: TAB_KON_204 – TUC_KON_204 „LöscheDateiInhalt"

Element	Beschreibung
Name	TUC_KON_204 „LöscheDateiInhalt"
Beschreibung	Inhalt einer transparenten Datei löschen
Auslöser	• Aufruf des Use Cases durch ein Fachmodul im Konnektor
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu bearbeitenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • offset – *optional* (Position, ab der der Inhalt gelöscht werden soll. Default: 0)
Komponenten	Karte, Kartenterminal, Konnektor

Ausgangsdaten	keine
Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe Card.TYPE <> KVK 3. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer im Besitz des Karten-Locks ist. 4. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 5. Selektiere Verzeichnis und Datei 6. Lösche Inhalt der selektierten Datei über Kartenkommando „ERASE BINARY", ggf. ab angegebenem Offset, sonst ab Anfang
Varianten/ Alternativen	keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist keine eGK der Generation 2 oder höher: Fehlercode 4209 (→3) Karte ist fremd reserviert, Fehlercode 4093 (→4) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5) Verzeichnis oder Datei existiert nicht, Fehlercode 4087 (→6) Ausgewählte Datei ist nicht transparent, Fehlercode 4089 (→6) Verzeichnis deaktiviert, Fehlercode 4086 (→5-6) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 86: TAB_KON_785 Fehlercodes TUC_KON_204 „LöscheDateiInhalt"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4086	Technical	Error	Verzeichnis deaktiviert
4087	Technical	Error	Datei nicht vorhanden
4089	Technical	Error	Datei ist vom falschen Typ

4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=]

4.1.5.4.12 TUC_KON_209 „LeseRecord"

TIP1-A_4575 - TUC_KON_209 „LeseRecord"
Der Konnektor MUSS den technischen Use Case „LeseRecord" gemäß TUC_KON_209 umsetzen.

Tabelle 87: TAB_KON_538 – TUC_KON_209 „LeseRecord"

Element	Beschreibung
Name	TUC_KON_209 „LeseRecord"
Beschreibung	Daten aus strukturierter Datei lesen
Auslöser	• Aufruf durch Fachmodul
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • recordNumber
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• content (Inhalt des Records)

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt 3. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 4. Selektiere Verzeichnis und ggf. Datei 5. Lies Daten über Kartenkommando „READ RECORD" unter Berücksichtigung von recordNumber 6. Rückgabe der Daten an den Aufrufer
Varianten/ Alternativen	keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→3) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→4) Verzeichnis oder Datei oder Record existiert nicht, Fehlercode 4087 (→5) Wenn Karte WrongFileType liefert, Fehlercode 4089 (→5) Verzeichnis deaktiviert, Fehlercode 4086 (→4-5) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 88: TAB_KON_539 Fehlercodes TUC_KON_209 „LeseRecord"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4086	Technical	Error	Verzeichnis deaktiviert
4087	Technical	Error	Datei nicht vorhanden
4089	Technical	Error	Datei ist vom falschen Typ
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.13 TUC_KON_210 „SchreibeRecord"

TIP1-A_4576 - TUC_KON_210 „SchreibeRecord"
Der Konnektor MUSS den technischen Use Case „SchreibeRecord" gemäß TUC_KON_210 umsetzen.

Tabelle 89: TAB_KON_224 – TUC_KON_210 „SchreibeRecord"

Element	Beschreibung
Name	TUC_KON_210 „SchreibeRecord"
Beschreibung	Daten in lineare Datei schreiben
Auslöser	• Aufruf durch Fachmodul
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu bearbeitenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • recordNumber • dataToBeWritten (Zu schreibende Daten)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	keine

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe Card.TYPE <> KVK 3. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 4. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 5. Selektiere Verzeichnis und ggf. Datei 6. Schreibe Daten über Kartenkommando „UPDATE RECORD" unter Berücksichtigung von recordNummer
Varianten/ Alternativen	keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Schreibzugriff auf KVK nicht gestattet, Fehlercode 4085 (→3) Karte ist fremd reserviert, Fehlercode 4093 (→4) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5) Verzeichnis, Datei existiert nicht, Fehlercode 4087 (→5-6) Verzeichnis deaktiviert, Fehlercode 4086 (→4-6) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 90: TAB_KON_540 Fehlercodes TUC_KON_210 „SchreibeRecord"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4086	Technical	Error	Verzeichnis deaktiviert
4087	Technical	Error	Datei nicht vorhanden
4088	Technical	Error	Datensatz zu groß

4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.14 TUC_KON_211 „LöscheRecordInhalt"

TIP1-A_5477 - TUC_KON_211 „LöscheRecordInhalt"
Der Konnektor MUSS den technischen Use Case „LöscheRecordInhalt" gemäß TUC_KON_211 umsetzen.

Tabelle 91: TAB_KON_211 – TUC_KON_211 „LöscheRecordInhalt"

Element	Beschreibung
Name	TUC_KON_211 „LöscheRecordInhalt"
Beschreibung	Inhalt eines Records einer strukturierten Datei löschen
Auslöser	• Aufruf des Use Case im Rahmen von technischen Use Cases der Basisdienste des Konnektors • Aufruf des Use Cases durch ein Fachmodul im Konnektor
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu bearbeitenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • recordNumber
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	keine

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe Card.TYPE <> KVK 3. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer im Besitz des Karten-Locks ist. 4. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 5. Selektiere Verzeichnis und Datei 6. Lösche Recordinhalt (identifiziert durch recordNumber) der selektierten Datei über Kartenkommando „ ERASE RECORD“
Varianten/ Alternativen	keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist keine eGK der Generation 2 oder höher: Fehlercode 4209 (→3) Karte ist fremd reserviert, Fehlercode 4093 (→4) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5) Verzeichnis, Datei oder Record existiert nicht, Fehlercode 4087 (→6) Verzeichnis deaktiviert, Fehlercode 4086 (→6) Record nicht vorhanden, Fehlercode 4091 (→5-6) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 92: TAB_KON_786 Fehlercodes TUC_KON_211 „LöscheRecordInhalt“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4086	Technical	Error	Verzeichnis deaktiviert
4087	Technical	Error	Datei nicht vorhanden
4091	Technical	Error	Record nicht vorhanden
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=]

4.1.5.4.15 TUC_KON_214 „FügeHinzuRecord"

TIP1-A_4577 - TUC_KON_214 „FügeHinzuRecord"
Der Konnektor MUSS den technischen Use Case „FügeHinzuRecord" gemäß TUC_KON_214 umsetzen.

Tabelle 93: TAB_KON_228 – TUC_KON_214 „FügeHinzuRecord"

Element	Beschreibung
Name	TUC_KON_214 „FuegeHinzuRecord"
Beschreibung	Daten in lineare Datei anfügen
Auslöser	• Aufruf durch Fachmodul • TUC_KON_006
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu bearbeitenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • dataToBeWritten (Zu schreibende Daten)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	keine

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe Card.TYPE <> KVK 3. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 4. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 5. Selektiere Verzeichnis und ggf. Datei 6. Schreibe Daten über Kartenkommando „APPEND RECORD"
Varianten/ Alternativen	keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Schreibzugriff auf KVK nicht gestattet, Fehlercode 4085 (→3) Karte ist fremd reserviert, Fehlercode 4093 (→4) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5-6) Verzeichnis, Datei existiert nicht, Fehlercode 4087 (→6) Verzeichnis deaktiviert, Fehlercode 4086 (→5-6) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 94: TAB_KON_541 Fehlercodes TUC_KON_214 „FügeHinzuRecord"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4086	Technical	Error	Verzeichnis deaktiviert
4087	Technical	Error	Datei nicht vorhanden
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.16 TUC_KON_215 „SucheRecord"

TIP1-A_4578 - TUC_KON_215 „SucheRecord"
Der Konnektor MUSS den technischen Use Case „SucheRecord" gemäß TUC_KON_215 umsetzen.

Tabelle 95: TAB_KON_229 – TUC_KON_215 „SucheRecord"

Element	Beschreibung
Name	TUC_KON_215 „SucheRecord"
Beschreibung	Daten in linearer Datei suchen
Auslöser	• Aufruf durch Fachmodul
Vorbedingungen	Die Karte MUSS den nötigen Sicherheitszustand für den Zugriff besitzen
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu bearbeitenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich die Datei befindet) • pattern (SuchMuster) • recordNumber – *optional; default = 1* (Recordnummer, bei der Suche beginnen soll) ()
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• numbersFound (Liste: Nummern der Records, die dem SuchMuster entsprechen)

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Prüfe PinRef/KeyRef in CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden 4. Selektiere Verzeichnis und ggf. Datei 5. Sende Kartenkommando „SEARCH RECORD" mit SuchMuster *pattern* unter Berücksichtigung von recordNumber 6. Liefere Antwort der Karte zurück
Varianten/ Alternativen	Keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD-Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→3) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→4-5) Verzeichnis, Datei existiert nicht, Fehlercode 4087 (→5) Verzeichnis deaktiviert, Fehlercode 4086 (→4-5) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 96: TAB_KON_542 Fehlercodes TUC_KON_215 „SucheRecord"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4086	Technical	Error	Verzeichnis deaktiviert
4087	Technical	Error	Datei nicht vorhanden
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.17 TUC_KON_018 „eGK-Sperrung prüfen"

TIP1-A_4579 - TUC_KON_018 „eGK-Sperrung prüfen"
Der Konnektor MUSS den technischen Use Case „eGK-Sperrung prüfen" gemäß TUC_KON_018 umsetzen.

Tabelle 97: TAB_KON_110 - TUC_KON_018 „eGK-Sperrung prüfen"

Element	Beschreibung
Name	TUC_KON_018 „eGK-Sperrung prüfen"
Beschreibung	Es wird geprüft, dass DF.HCA (Health Care Application) der eGK nicht gesperrt ist und optional, dass das AUT-Zertifikat im DF.ESIGN gültig ist. Für eine Karte ab der Generation G2.1 wird das AUT-Zertifikat (ECC) geprüft. Für eine Karte der Generation G2.0 wird das AUT-Zertifikat (RSA) geprüft.
Auslöser	Aufruf durch Fachmodul im Konnektor
Vorbedingungen	keine
Eingangsdaten	• cardSession • checkHcaOnly [Boolean] - *optional; default = false* (Prüfung auf die Frage beschränken, ob auf DF.HCA zugegriffen werden kann)
Komponenten	Konnektor, Kartenterminal, eGK

Ausgangsdaten	• Karte gesperrt: true | false
	• Status – *optional/wenn checkHcaOnly = false*
	  • DF.HCA gesperrt: true | false
	  • Ergebnis der Offline-Prüfung des C.CH.AUT-Zertifikats:
	  gültig | ungültig
	  • Sperrstatus des C.CH.AUT-Zertifikats:
	  gut | gesperrt | nicht ermittelbar

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession)
	2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt.
	3. Selektiere DF.HCA :
	  a. Wenn die Karte '90 00' zurückmeldet, war das Selektieren möglich: DF.HCA gesperrt = false
	  b. In allen anderen Fällen war das Selektieren nicht fehlerfrei möglich: DF.HCA gesperrt = true
	4. Wenn checkHcaOnly = true
	Beende TUC, liefere Status.
	5. Ermittle Zertifikatsobjekt (fileIdentifier und folder) für C.AUT der Karte unter Berücksichtigung des kryptographischen Verfahrens crypt

	gemäß TAB_KON_858. Für eine Karte ab der Generation G2.1 setze crypt=ECC. Für eine Karte der Generation G2.0 setze crypt=RSA. Rufe Cert = TUC_KON_216 „LeseZertifikat“ {cardSession; fileIdentifier; folder} 6. Bestimme per Aufruf von TUC_KON_037 „Zertifikat prüfen“ a. das Ergebnis der Offline-Prüfung des C.CH.AUT-Zertifikats (gültig | ungültig) sowie b. den Sperrstatus des C.CH.AUT-Zertifkats (gut | gesperrt | nicht ermittelbar). 7. Die Karte ist gesperrt = true, wenn a. DF.HCA gesperrt = true oder b. Ergebnis der Offline-Prüfung des C.CH.AUT-Zertifikats = ungültig oder c. Sperrstatus des C.CH.AUT-Zertifkats = gesperrt. In allen anderen Fällen ist die Karte gesperrt = false.
Varianten/ Alternativen	keine
Fehlerfälle	(→2) Karte ist fremd reserviert, Fehlercode 4093
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 98: TAB_KON_239 Fehlercodes TUC_KON_018 „eGK-Sperrung prüfen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet

[<=]

4.1.5.4.18 TUC_KON_006 „Datenzugriffsaudit eGK schreiben“

TIP1-A_4580 - TUC_KON_006 „Datenzugriffsaudit eGK schreiben“
Der Konnektor MUSS den technischen Use Case „Datenzugriffsaudit eGK schreiben“ gemäß TUC_KON_006 umsetzen.

Tabelle 99: TAB_KON_108 - TUC_KON_006 „Datenzugriffsaudit eGK schreiben"

Element	Beschreibung
Name	TUC_KON_006 „Datenzugriffsaudit eGK schreiben"
Beschreibung	Zugriff auf eGK in EF.Logging protokollieren.
Auslöser	Aufruf durch ein Fachmodul
Vorbedingungen	Keine
Eingangsdaten	• cardSession (CardSession einer eGK) • sourceCardSession (HBA/SMC-B, der/die für den eGK-Zugriff verwendet wird) • dataType (zugreifende Anwendung, siehe [gem_Spec_Karten_Fach_TIP#4.1 – Tabelle Tab_Karten_Fach_TIP_010 _StrukturEF.Logging – Struktur der Rekords der Datei EF.Logging] • accessType (Zugriffsart, siehe ebenda)
Komponenten	eGK, HBA/SMC, Konnektor, Kartenterminal
Ausgangsdaten	Keine
Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe Card.TYPE = EGK 3. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 4. Wenn KeyRef in CARDSESSION.AUTHSTATE für DF.HCA.EF.LOGGING nicht mit passender Rolle vorhanden: Rufe TUC_CON_005 „Card-to-Card authentisieren" { sourceCardSession; targetCardSession = cardSession; authMode = einseitig} 5. Erzeuge Loggingdaten gemäß [gem_Spec_Karten_Fach_TIP#4.1 – Tabelle Tab_Karten_Fach_TIP_010_StrukturEF.Logging – Struktur der Rekords der Datei EF.Logging] 6. Rufe TUC_KON_214 „FügeHinzuRecord" { cardSession =$cardSession; folder = MF; *fileIdentifier* = DF.HCA/EF.Logging; dataToBeWritten = Loggingdaten }
Varianten/ Alternativen	Keine
Fehlerfälle	(→2) Protokoll nur für eGK gestattet, Fehlercode 4251 (→3) Karte ist fremd reserviert, Fehlercode 4093

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 100: TAB_KON_238 Fehlercodes TUC_KON_006 „Datenzugriffsaudit eGK schreiben"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4251	Technical	Error	Protokoll nur für eGK gestattet

[<=]

4.1.5.4.19 TUC_KON_218 „Signiere"

TIP1-A_4581 - TUC_KON_218 „Signiere"
Der Konnektor MUSS den technischen Use Case „Signiere" gemäß TUC_KON_218 umsetzen.

Tabelle 101: TAB_KON_231 – TUC_KON_218 „Signiere"

Element	Beschreibung
Name	TUC_KON_218 „Signiere"
Beschreibung	Dieser Use Case beschreibt das Anwenden eines privaten Schlüssels einer Karte zur Signatur oder Authentisierung.
Auslöser	• Aufruf einer der Operationen SignDocument des Signaturdienstes oder ExternalAuthenticate des Authentifizierungsdienstes durch das Clientsystem. • Aufruf durch Fachmodul
Vorbedingungen	Zugriffsbedingung für referenzierten Schlüssel MUSS erfüllt sein

Eingangsdaten	• cardSession • pinRef (PIN-Referenz, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps) • keyRef (Referenz auf den privaten Schlüssel, mit dem signiert werden soll, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps) • algorithmusId (einer der laut Objektspezifikation für diesen Schlüssel zulässigen algorithmIdentifier) • dataToBeSigned (Zu signierende Daten, Hashwert)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• chiffrat (Signatur)
Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Prüfe pinRef in CARDSESSION.AUTHSTATE vorhanden: 4. Setze keyRef und algorithmusId der Karte 5. Sende „PSO: COMPUTE DS“ mit dataToBeSigned an Karte 6. Gib chiffrat an den Aufrufer zurück
Varianten/ Alternativen	Keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→3) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 102: TAB_KON_543 Fehlercodes TUC_KON_218 „Signiere"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4085	Security	Error	Zugriffsbedingungen nicht erfüllt
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.20 TUC_KON_219 „Entschlüssele"

TIP1-A_4582 - TUC_KON_219 „Entschlüssele"
Der Konnektor MUSS den technischen Use Case „Entschlüssele" gemäß TUC_KON_219 umsetzen.

Tabelle 103: TAB_KON_232 – TUC_KON_219 „Entschlüssele"

Element	Beschreibung
Name	TUC_KON_219 „Entschlüssele"
Beschreibung	Dieser Use Case beschreibt das Anwenden eines privaten Schlüssels einer Karte zur Entschlüsselung.
Auslöser	• Aufruf durch Fachmodul
Vorbedingungen	Zugriffsbedingung für referenzierten Schlüssel muss erfüllt sein
Eingangsdaten	• cardSession • pinRef (Referenz auf die PIN, mit der der Entschlüsselungsschlüssel freigeschaltet werden kann, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps) • keyRef (Referenz auf den privaten Schlüssel, mit dem entschlüsselt werden soll, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps.) • algorithmusId (einer der für diesen Schlüssel zulässigen algorithmIdentifier) • encryptedData (Zu entschlüsselnde Daten, Chiffrat)
Komponenten	Karte(n), Kartenterminal, Konnektor

Ausgangsdaten	• plainData (Entschlüsselte Daten)
Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt. 3. Prüfe pinRef in CARDSESSION.AUTHSTATE vorhanden: 4. Selektiere DF, in dem der private Schlüssel (keyRef) liegt, falls er noch nicht selektiert ist. 5. Setze Schlüssel (keyRef) und algorithmusId. 6. Sende encryptedData mittels Kommandos PSO: DECIPHER. 7. gib plainData an den Aufrufer zurück
Varianten/ Alternativen	Keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→3) Nötige PIN ist nicht verifiziert, Fehlercode 4085 (→5) Schlüssel nicht vorhanden, Fehlercode 4079 (→6) Fehler im Chiffrat: Fehlercode 4069 (→4, 6) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Varianten/ Alternativen	Keine
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 104: TAB_KON_210 Fehlercodes TUC_KON_219 „Entschlüssele"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4069	Technical	Error	korruptes Chiffrat bei asymmetrischer Entschlüsselung
4079	Technical	Error	Schlüsseldaten fehlen
4085	Security	Error	Zugriffsbedingungen nicht erfüllt

4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten

[<=]

4.1.5.4.21 TUC_KON_200 „SendeAPDU"

TIP1-A_4583 - TUC_KON_200 „SendeAPDU"
Der Konnektor MUSS den technischen Use Case „SendeAPDU" gemäß TUC_KON_200 umsetzen.

Tabelle 105: TAB_KON_215 TUC_KON_200 „SendeAPDU"

Element	Beschreibung
Name	TUC_KON_200 „SendeAPDU"
Beschreibung	Dieser Use Case beschreibt das Senden einer APDU an eine Chipkarte bzw. an ein Kartenterminal und das Empfangen der Antwort.
Auslöser	• Aufruf durch Fachmodul
Vorbedingungen	Zugriffsbedingungen für das Kommando müssen in der Karte erfüllt sein und Karte muss für exklusiven Zugriff reserviert worden sein
Eingangsdaten	• cardSession – *optional/verpflichtend, wenn die APDU an die Karte gerichtet ist* • ctId – *optional/verpflichtend, wenn die APDU an das Kartenterminal gerichtet ist* (Kartenterminalidentifikator für Kommandos an das Kartenterminal) • commandAPDU (versandfertige APDU (Bytefolge), in dem die Parameter {CLA, INS, P1,P2, Data (*optional*) Le(*optional*) } gesetzt sind.)
Komponenten	Karte(n), Kartenterminal, Konnektor
Ausgangsdaten	• responseAPDU (Antwort der Chipkarte oder des Kartenterminals, Bytefolge)
Standardablauf	A. cardSession ist gegeben 1. Ermittle Card = CM_CARD_LIST(cardSession) 2. Prüfe, dass der Aufrufer für die zur cardSession gehörenden Karte ein Lock hat. 3. commandAPDU wird über das Kartenterminal an die Zielkarte gesendet

	4. die Antwort (responseAPDU) der Zielkarte wird an den Aufrufer zurückgegeben. B. ctId ist gegeben 1. Sende commandAPDU an das Kartenterminal ctId 2. gib die Antwort responseAPDU des Kartenterminals an den Aufrufer zurück
Varianten/Alternativen	• Soll Secure Messaging verwendet werden, MUSS vorher TUC_KON_023 „Karte reservieren" aufgerufen werden
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Der Aufrufer ist nicht im Besitz des Karten-Locks, Fehlercode 4232 (→3) Kommunikationsfehler mit dem Kartenterminal: Fehlercode 4044. (→3) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 106: TAB_KON_216 Fehlercodes TUC_KON_200 „SendeAPDU"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4044	Technical	Error	Fehler beim Zugriff auf das Kartenterminal
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4232	Technical	Error	der Aufrufer besitzt nicht das Karten-Lock

[<=]

4.1.5.4.22 TUC_KON_024 „Karte zurücksetzen"

TIP1-A_4584 - TUC_KON_024 „Karte zurücksetzen"
Der Konnektor MUSS den technischen Use Case „Karte zurücksetzen" gemäß TUC_KON_024 umsetzen.

Tabelle 107: TAB_KON_737 – TUC_KON_024 „Karte zurücksetzen"

Element	Beschreibung
Name	TUC_KON_024 „Karte zurücksetzen"
Beschreibung	Der technische Use Case setzt die gewählte Karte zurück (alle erreichten Sicherheitszustände werden auf der Karte und in der

	Verwaltung des Konnektors zurückgesetzt; auf der Karte wird MF selektiert). Ein eventuell laufendes C2C wird dabei abgebrochen.
Auslöser	Fachmodul
Vorbedingungen	keine
Eingangsdaten	• ctId – *optional/verpflichtend, wenn keine cardSession angegeben ist* (Kartenterminalidentifikator) • slotId – *optional/verpflichtend, wenn keine cardSession angegeben ist* (Nummer des Slots, in dem die Karte steckt) • cardSession – *optional/verpflichtend, wenn ctId und slotId nicht angegeben sind* (Angabe der CardSession alternativ zur Angabe von ctId und slotId)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	Keine
Standardablauf	1. Wenn cardSession gegeben, dann ermittle ctId und slotId 2. Der Konnektor prüft, dass entweder die Karte nicht reserviert ist oder der Aufrufer im Besitz des Karten-Locks ist. 3. Brich eventuell parallel laufenden TUC_KON_005 ab 4. Sende SICCT RESET ICC für slotId an das Kartenterminal CtID, um einen Warm Reset auszulösen 5. Lösche alle Sicherheitszustände aus `CARDSESSION.AUTHSTATE` und den Inhalt von `CARDSESSION.AUTHBY`.
Varianten/ Alternativen	Keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Der Aufrufer ist nicht im Besitz des Karten-Locks, Fehlercode 4232 (→4) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 108: TAB_KON_544 Fehlercodes TUC_KON_024 „Karte zurücksetzen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			

4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4232	Technical	Error	der Aufrufer ist nicht im Besitz des Karten-Locks

[<=]

4.1.5.4.23 TUC_KON_216 „LeseZertifikat"

TIP1-A_4585 - TUC_KON_216 „LeseZertifikat"
Der Konnektor MUSS den technischen Use Case „LeseZertifikat" gemäß TUC_KON_216 umsetzen.

Tabelle 109: TAB_KON_230 – TUC_KON_216 „LeseZertifikat"

Element	Beschreibung
Name	TUC_KON_216 „LeseZertifikat"
Beschreibung	Dieser Use Case beschreibt das Lesen eines Zertifikates von einer Karte
Auslöser	• Aufruf der Operation ReadCardCertificate des Zertifikatsdienstes durch das Clientsystem. • Aufruf durch Fachmodul • Aufruf im Rahmen von technischen Use Cases der Basisdienste des Konnektors
Vorbedingungen	Keine
Eingangsdaten	• cardSession • fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei) • sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei) • folder (Verzeichnis/Applikation auf der Karte, in dem sich das Zertifikat befindet)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	• certificate (gelesenes Zertifikat)

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession)
	2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer das Karten-Lock besitzt.
	3. Prüfe CARDSESSION.AUTHSTATE für adressierte Datei wie in Zugriffsbedingung der Karte definiert vorhanden
	4. Rufe TUC_KON_202 „LeseDatei" {
	cardSession;
	fileIdentifier;
	folder }
	oder TUC_KON_202 „LeseDatei" {
	cardSession;
	sfid;
	folder }
	5. Das Zertifikat wird an den Aufrufer zurückgegeben.

Varianten/ Alternativen	Keine
Fehlerfälle	(→2) Karte ist fremd reserviert, Fehlercode 4093 (->4) Es wurde versucht, ein Zertifikat von der Karte zu lesen, welches auf der Karte nicht vorhanden ist (Fehlercode 4256). Hierbei kann es sich um ein fehlendes Zertifikatsobjekt (z.B. adressiertes ECC-Zertifikat auf HBA G2.0) oder ein leeres Zertifikatsobjekt (z.B. adressiertes ECC-Zertifikat auf gSMC-K G2.0, welches aber nicht personalisiert wurde) handeln.
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 110: TAB_KON_209 Fehlercodes TUC_KON_216 „LeseZertifikat"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4256	Technical	Warning	Zertifikat auf Karte nicht vorhanden

[<=]

4.1.5.4.24 TUC_KON_036 „LiefereFachlicheRolle"

TIP1-A_5478 - TUC_KON_036 „LiefereFachlicheRolle"
Der Konnektor MUSS den technischen Use Case TUC_KON_036 „LiefereFachlicheRolle" umsetzen.

Tabelle 111: TAB_KON_827 TUC_KON_036 „LiefereFachlicheRolle"

Element	Beschreibung
Name	TUC_KON_036 „LiefereFachlicheRolle"
Beschreibung	Dieser TUC liefert die fachliche Rolle, die der OID aus dem X.509Zertifikat der gesteckten Karte zugeordnet ist. Es werden nur folgende Karten unterstützt: HBAx, SM-B, EGK, KVK Es werden nur die AUT-Zertifikate ausgelesen. Für eine Karte ab der Generation G2.1 wird das AUT-Zertifikat (ECC) geprüft. Für eine Karte der Generation G2.0 wird das AUT-Zertifikat (RSA) geprüft.
Auslöser	• Aufruf durch ein Fachmodul oder eine Basisanwendung des Konnektors
Vorbedingungen	Keine
Eingangsdaten	• cardSession
Komponenten	Konnektor, Karte
Ausgangsdaten	• role (fachliche Rolle gemäß [gemSpec_PKI#Tab_PKI_254 Zugriffsprofile für eine Rollenauthentisierung]
Nachbedingungen	Keine

Standardablauf	1. Ermittle Card = CM_CARD_LIST(cardSession)
	2. Prüfe, dass der Karte entweder kein Lock zugeordnet ist oder der Aufrufer im Besitz des Karten-Locks ist.
	3. Wenn CARD.TYPE = KVK, dann setze fachliche Rolle = „Versicherter" und springe zu Schritt 8.
	4. Ermittle *fileIdentifier* und folder des C.AUT-Zertifikates unter Berücksichtigung des kryptographischen Algorithmus crypt für die Karte, die durch die cardSession referenziert wird.
	Für eine Karte ab der Generation G2.1 setze crypt=ECC.
	Für eine Karte der Generation G2.0 setze crypt=RSA.
	Welches Zertifikat gelesen wird, ist in TAB_KON_858 beschrieben.
	5. Lies Zertifikat:
	Rufe TUC_KON_216 "LeseZertifikat" {
	cardSession;
	fileIdentifier = *fileIdentifier* (AUT-Zertifikat);
	folder = folder(AUT-Zertifikat)}

	6. Ermittle ProfessionOIDs aus Extension Admission des Zertifikates: Rufe TUC_PKI_009 „Rollenermittlung" {certificate} 7. Ermittle die fachliche Rolle, die den ProfessionOIDs entspricht, gemäß [gemSpec_PKI# Tab_PKI_254 Zugriffsprofile für eine Rollenauthentisierung]. 8. Rückgabe $role (fachliche Rolle) an den Aufrufer
Varianten/ Alternativen	Keine
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Karte ist fremd reserviert, Fehlercode 4093 (→7) ProfessionOIDs mappen nicht auf die gleiche Rolle, Fehlercode 4210
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 112: TAB_KON_829 Fehlercodes TUC_KON_036 „LiefereFachlicheRolle"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4093	Technical	Error	Karte wird in einer anderen Kartensitzung exklusiv verwendet
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4210	Technical	Error	ProfessionOIDs nicht eindeutig auf eine Rolle abbildbar

[<=]

4.1.5.5 Operationen an der Außenschnittstelle

TIP1-A_4586-03 - Basisanwendung Kartendienst

Der Konnektor MUSS für Clients eine Basisanwendung Kartendienst mit den Operationen VerifyPin, ChangePin, UnblockPin, GetPinStatus an der Außenschnittstelle anbieten.

Tabelle 113: TAB_KON_038 Basisanwendung Karten- und Kartenterminaldienst

Name	CardService

Version (KDV)	8.1.0 (WSDL- und XSD-Version) 8.1.1 (WSDL- und XSD-Version) 8.1.2 (WSDL-Version) 8.1.3 (XSD-Version)	
Namensraum	Siehe GitHub	
Namensraum-Kürzel	`CARD für Schema und CARDW für WSDL`	
Operationen	**Name**	**Kurzbeschreibung**
	`VerifyPin`	PIN prüfen
	`ChangePin`	PIN ändern
	`UnblockPin`	PIN entsperren
	`GetPinStatus`	PIN-Status ermitteln
	`EnablePin`	Erfordernis der PIN-Verifikation einschalten
	`DisablePin`	Erfordernis der PIN-Verifikation abschalten

WSDL	`CardService.wsdl (WSDL-Version 8.1.0)`
	`CardService_v8_1_1.wsdl`
	`CardService_v8_1_2.wsdl`	

Schema	`CardService.xsd (XSD-Version 8.1.0)`
	`CardService_v8_1_1.xsd`
	`CardService_v8_1_3.xsd`	

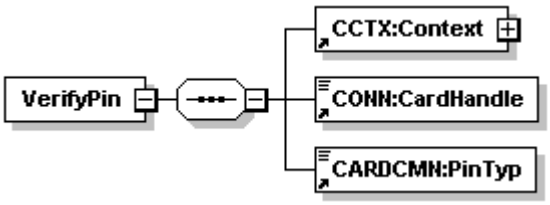
[<=]

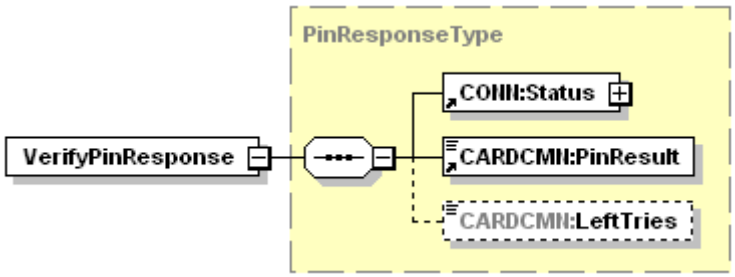
4.1.5.5.1 VerifyPin

TIP1-A_4587 - Operation VerifyPin
Der Konnektor MUSS an der Außenschnittstelle eine Operation VerifyPin, wie in Tabelle TAB_KON_047 Operation VerifyPin beschrieben, anbieten.

Tabelle 114: TAB_KON_047 Operation VerifyPin

Name	**`VerifyPin`**
Beschreibung	Stößt die sichere Eingabe einer PIN am PIN-Pad des Kartenterminals für eine Karte an. Das Ergebnis der PIN-Prüfung gibt Auskunft darüber, ob die PIN richtig oder falsch war oder aufgrund zu vieler Fehlversuche blockiert ist. Eine Karte kann potentiell mehrere PINs haben. Insbesondere für die qualifizierte elektronische Signatur existiert eine separate PIN. Diese PIN darf nur über das PIN-Pad eingegeben werden. Die PIN-Verifikation und die damit verbundene Änderung des Sicherheitsstatus der Karte erfolgt nur für die durch den Aufrufkontext adressierte Kartensitzung. Falls die Karte in einem zentralen Kartenterminal steckt, auf das der Arbeitsplatz Zugriff hat, erfolgt eine Remote-PIN-Eingabe. Das Kartenterminal für die PIN-Eingabe ergibt sich dabei aus der im Aufrufkontext

<table>
<tr><td></td><td>angegebenen Mandanten-ID und Arbeitsplatz-ID
Diese Operation entspricht dem Aufruf von TUC_KON_012 „PIN verifizieren“. Dort sind auch die Display Messages definiert, die bei PIN-Eingabe am Kartenterminal anzuzeigen sind (TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal). Die beim Aufruf von TUC_KON_012 anzugebende PIN-Art lautet „mandatorisch“. Die PIN-Verifikation wird also unabhängig vom erreichten Sicherheitsstatus in der Karte immer durchgeführt.</td></tr>
<tr><td>Aufrufparameter</td><td> 

<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>Context</td><td>MandantId, CsId, WorkplaceId verpflichtend; UserId verpflichtend für HBAx</td></tr>
<tr><td>CardHandle</td><td>Adressiert die Karte, für die die PIN verifiziert werden soll.
Die Operation DARF die PIN-Verifikation mit der eGK NICHT unterstützen. Unterstützt werden die Kartentypen HBAx und SM-B. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4209 abbrechen.</td></tr>
<tr><td>PinTyp</td><td>Gibt an, welche PIN der Karte verifiziert werden soll.
Erlaubte Belegung von PinTyp in Abhängigkeit der durch Cardhandle referenzierten Karte:
HBAx: PIN.CH
SM-B: PIN.SMC</td></tr>
</table></td></tr>
<tr><td>Rückgabe</td><td> 

<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
</table></td></tr>
</table>

	Status	Enthält den Ausführungsstatus der Operation (siehe 3.5.2)	
	PinResult	Wert	Bedeutung
		OK	Prüfung war erfolgreich
		REJECTED	PIN war falsch. Die Anzahl der verbleibenden Versuche ist im Element LeftTries
		ERROR	Ein Bearbeitungsfehler ist aufgetreten. Fehlerursache im Feld Error
		WASBLOCKED	PIN war zum Aufrufzeitpunkt bereits gesperrt
		NOWBLOCKED	PIN ist durch aktuellen Fehlversuch gesperrt
		TRANSPORT_PIN	PIN ist mit Transportschutz versehen
	LeftTries	Im Falle von Result=REJECTED wird hier die Anzahl der verbleibenden möglichen Versuche zurückgegeben.	
Vorbedingung	Keine		
Nachbedingung	keine		

Der Ablauf der Operation VerifyPin ist in Tabelle TAB_KON_738 Ablauf VerifyPin beschrieben.

Tabelle 115: TAB_KON_738 Ablauf VerifyPin

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle cardSession über TUC_KON_026 {
		mandatId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		userId = $context.userId;
		cardHandle }

4.	TUC_KON_012 „PIN verifizieren“	Verifiziere PIN über TUC_KON_012 {
		cardSession;
		workplaceId = $context.workplaceId;
		pinRef = PinRef(PinTyp);
		appName = „“ (Leerstring);
		verificationType = Mandatorisch }

5.	Verifikationsergebnis auswerten	Wenn TUC_KON_012 mit Fehler 4065 abbricht, wird dies als erfolgreicher Operationsdurchlauf mit PinResult= TRANSPORT_PIN abgefangen. Wenn TUC_KON_012 den Returncode BLOCKED liefert, wird dies als erfolgreicher Operationsdurchlauf mit PinResult= NOWBLOCKED zurückgegeben. Wenn TUC_KON_012 mit Fehler 4063 abbricht, wird dies als erfolgreicher Operationsdurchlauf mit PinResult= WASBLOCKED zurückgegeben

Tabelle 116: TAB_KON_545 Fehlercodes „VerifyPin“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4078	Security	Error	PIN-Eingabe über das Clientsystem ist nicht zugelassen
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

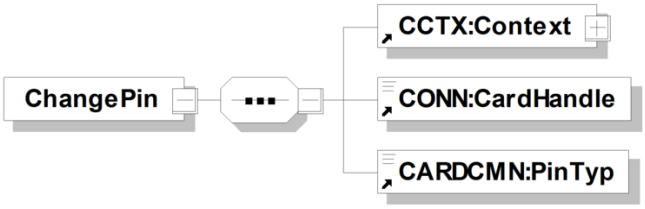
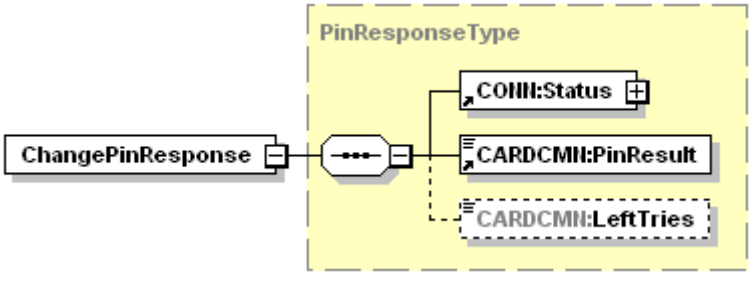
[<=]

4.1.5.5.2 ChangePin

TIP1-A_4588 - Operation ChangePin
Der Konnektor MUSS an der Außenschnittstelle eine Operation ChangePin, wie in Tabelle TAB_KON_049 Operation ChangePin beschrieben, anbieten.

Tabelle 117: TAB_KON_049 Operation ChangePin

Name	`ChangePin`
Beschreibung	Ändert eine PIN einer Karte. Alte und neue PIN werden dabei am PIN-Pad des Kartenterminals eingegeben. Falls die Karte in einem zentralen Kartenterminal steckt, auf das

<table>
<tr><td></td><td colspan="2">der im Aufrufkontext angegebene Arbeitsplatz Zugriff hat, erfolgt eine Remote-PIN-Eingabe.
Diese Operation entspricht dem Aufruf TUC_KON_019 „PIN ändern“ .</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>Context</td><td>MandantId, CsId, WorkplaceId verpflichtend; UserId optional (verpflichtend beim HBA)</td></tr>
<tr><td></td><td>CardHandle</td><td>Adressiert die Karte, für die die PIN geändert werden soll. Unterstützt werden die Kartentypen EGK, HBAx und SM-B. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4209 abbrechen.</td></tr>
<tr><td></td><td>PinTyp</td><td>Gibt an, welche PIN der Karte geändert werden soll.
Erlaubte Belegung von PinTyp in Abhängigkeit der durch CardHandle referenzierten Karte:
eGK G1+: PIN.CH,
eGK G2: PIN.CH, MRPIN.NFD, MRPIN.NFD_READ, MRPIN.DPE, MRPIN.GDD, MRPIN.OSE, MRPIN.AMTS, PIN.AMTS_REP
zusätzlich eGK G2.0: MRPIN.DPE_READ
HBAx: PIN.CH, PIN.QES
SM-B: PIN.SMC</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
</table>

	LeftTries	Im Falle von REJECTED wird hier die Anzahl der verbleibenden möglichen Versuche zurückgegeben.	
	Status	Enthält den Ausführungsstatus der Operation, siehe 3.5.2	
	PinResult	Wert	Bedeutung
		OK	PIN-Änderung war erfolgreich
		ERROR	Ein Bearbeitungsfehler ist aufgetreten. Fehlerursache im Feld Error
		REJECTED	OldPIN war falsch Die Anzahl der verbleibenden Versuche ist im Element LeftTries
		WASBLOCKED	PIN war zum Aufrufzeitpunkt bereits gesperrt
		NOWBLOCKED	PIN ist durch aktuellen Fehlversuch gesperrt
Vorbedingung	Keine		
Nachbedingung	keine		

Tabelle 118: TAB_KON_546 Ablauf ChangePin

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 { mandantId = $context.mandantId; clientSystemId = $context.clientsystemId; workplaceId = $context.workplaceId; userId = $context.userId; cardHandle } Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle cardSession über TUC_KON_026 {
		mandantId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		cardHandle;
		userId = $context.userId }

4.	TUC_KON_019 „PIN ändern“	Ändere PIN über TUC_KON_019 {
		cardSession;
		workplaceId = $context.workplaceId;
		pinRef = PinRef(PinTyp) }

5.	Verifikationsergebnis auswerten	Wenn TUC_KON_019 den Returncode BLOCKED liefert, wird dies als erfolgreicher Operationsdurchlauf mit PinResult= NOWBLOCKED zurückgegeben. Wenn TUC_KON_019 mit Fehler 4063 abbricht, wird dies als erfolgreicher Operationsdurchlauf mit PinResult= WASBLOCKED zurückgegeben.

Tabelle 119: TAB_KON_547 Fehlercodes „ChangePin“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4072	Technical	Error	Ungültige PIN-Referenz `PinRef`
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=]

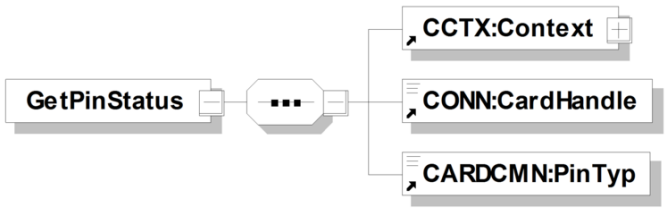
4.1.5.5.3 GetPinStatus

TIP1-A_4589 - Operation GetPinStatus
Der Konnektor MUSS an der Außenschnittstelle eine Operation GetPinStatus, wie in Tabelle TAB_KON_051 Operation GetPinStatus beschrieben, anbieten.

Tabelle 120: TAB_KON_051 Operation GetPinStatus

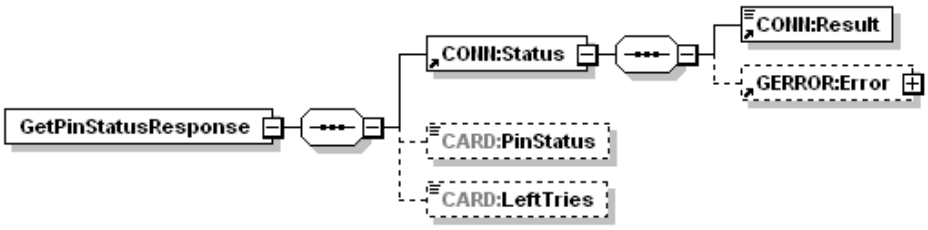
Name	`GetPinStatus`
Beschreibung	Diese Operation gibt Auskunft über den PIN-Zustand einer Karte. Für transportgeschützte PIN gibt die Operation die Art des Transportschutzes an. Für Echt-PINs kann mit dieser Operation die Anzahl der noch verbleibenden Versuche für PIN-Verifikationen ermittelt werden oder ob die PIN bereits verifiziert wurde.

Aufruf-parameter



Name	Beschreibung
Context	MandantId, CsId, WorkplaceId; UserId
`CardHandle`	Adressiert die Karte, für die der PIN-Status ermittelt werden soll. Unterstützt werden die Kartentypen EGK, HBAx und SM-B. Eine KVK ist nicht zulässig. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4209 abbrechen.
`PinTyp`	Gibt an, für welche PIN der Karte der PIN-Status ermittelt werden soll. Erlaubte Belegung von PinTyp in Abhängigkeit der durch Cardhandle referenzierten Karte: • eGK G1+: PIN.CH • eGK G2: PIN.CH, MRPIN.NFD, MRPIN.NFD_READ, MRPIN.DPE, MRPIN.GDD, MRPIN.OSE, MRPIN.AMTS, PIN.AMTS_REP • zusätzlich eGK G2.0: MRPIN.DPE_READ • HBAx: PIN.CH, PIN.QES • SM-B: PIN.SMC

Rückgabe



Name	Beschreibung	
`Status`	Enthält den Ausführungsstatus der Operation siehe 3.5.2	
`PinStatus`	Status der PIN. Die folgenden Werte sind verpflichtend:	
	Wert	Bedeutung

		VERIFIED	Bereits verifiziert (in CARDSESSION.AUTHSTATE vorhanden)
		TRANSPORT_PIN	Transport-PIN
		EMPTY_PIN	Leer-PIN
		BLOCKED	gesperrt
		VERIFIABLE	Echt-PIN, noch nicht verifiziert
		DISABLED	PIN-Schutz ausgeschaltet (Verifikation nicht erforderlich)
	LeftTries	Bei einer Echt-PIN wird hier bei PinStatus = VERIFIABLE die Anzahl der verbleibenden möglichen Versuche für die Verifikation der PIN zurückgegeben, bei einer gesperrten PIN 0.	
Vorbedingung	keine		
Nachbedingung	keine		

Tabelle 121: TAB_KON_548 Ablauf GetPinStatus

Nr .	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 { mandantId = $context.mandantId; clientSystemId = $context.clientsystemId; workplaceId = $context.workplaceId; userId = $context.userId; // falls angegeben cardHandle } Tritt bei der Prüfung ein Fehler auf, so bricht die Operation mit dem Fehlercode aus TUC_KON_000 ab.
3.	TUC_KON_026 „Liefere CardSession“	Ermittle cardSession über TUC_KON_026 { mandantId = $context.mandantId; clientSystemId = $context.clientsystemId; userId = $context.userId; // falls angegeben ; cardHandle }
4.	TUC_KON_022 „Liefere PIN-	Ermittle PinStatus über TUC_KON_022 { cardSession;

	Status“	pinRef = PinRef(PinTyp) }

Tabelle 122: TAB_KON_549 Fehlercodes „GetPinStatus“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4001	Technical	Error	interner Fehler
4072	Technical	Error	ungültige PIN-Referenz `PinRef`
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

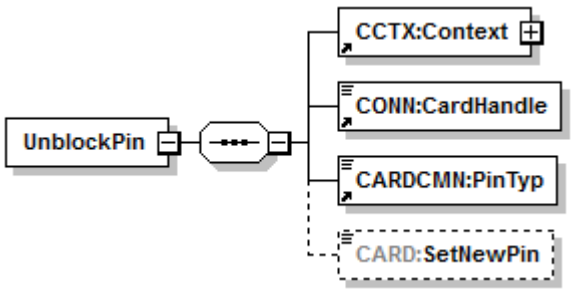
[<=]

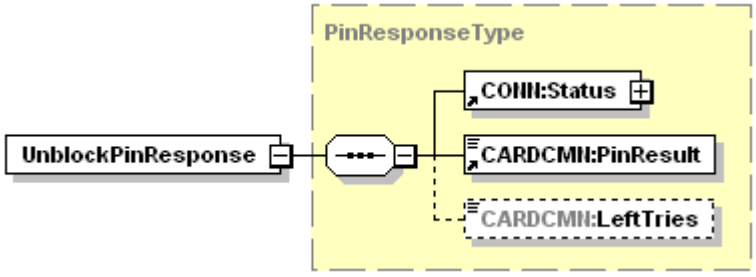
4.1.5.5.4 UnblockPin

TIP1-A_4590 - Operation UnblockPin
Der Konnektor MUSS an der Außenschnittstelle eine Operation UnblockPin, wie in Tabelle TAB_KON_053 Operation UnblockPin beschrieben, anbieten.

Tabelle 123: TAB_KON_053 Operation UnblockPin

Name	`UnblockPin`
Beschreibung	Mit diesem Kommando kann eine blockierte PIN wieder freigeschaltet werden. Dabei wird der Wiederholungszähler für diese PIN in der Karte auf seinen Anfangswert zurückgesetzt und es KANN eine neue PIN gesetzt werden. Um diese Aktion durchführen zu können, muss eine PUK (auch als Resetting Code bezeichnet) präsentiert werden. PIN und PUK werden am PIN-Pad des Kartenterminals eingegeben. Falls die Karte in einem zentralen Kartenterminal steckt, auf das der im Aufrufkontext angegebene Arbeitsplatz Zugriff hat, erfolgt eine Remote-PIN-Eingabe. Das Kartenterminal für die PIN-Eingabe ergibt sich dabei aus der im Aufrufkontext angegebenen Mandanten-ID und Arbeitsplatz-ID. Diese Operation entspricht dem Aufruf von TUC_KON_021 „PIN entsperren“.

<table>
<tr><td>Aufruf-parameter</td><td>


<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>Context</td><td>MandantId, CsId, WorkplaceId verpflichtend; UserId (optional, für HBA verpflichtend)</td></tr>
<tr><td>CardHandle</td><td>Adressiert die Karte, für die die Blockierung der PIN aufgehoben werden soll. Unterstützt werden die Kartentypen EGK, HBAx und SM-B. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4209 abbrechen.</td></tr>
<tr><td>PinTyp</td><td>Gibt an, für welche PIN der Karte die Blockierung aufgehoben werden soll.
Erlaubte Belegung von PinTyp in Abhängigkeit der durch Cardhandle referenzierten Karte:
- eGK G1+: PIN.CH
- eGK G2: PIN.CH, MRPIN.NFD, MRPIN.NFD_READ,
MRPIN.DPE, MRPIN.GDD, MRPIN.OSE, MRPIN.AMTS,
PIN.AMTS_REP
- zusätzlich eGK G2.0: MRPIN.DPE_READ
- HBAx: PIN.CH, PIN.QES
- SM-B: PIN.SMC</td></tr>
<tr><td>SetNewPin</td><td>Dieses Flag zeigt an, ob eine neue PIN gesetzt werden soll. Wird dieses Flag nicht angegeben, so wird FALSE angenommen.
Das Flag ist notwendig, um bei Eingabe am PIN-Pad eindeutig festzulegen, ob eine neue PIN gesetzt werden soll.</td></tr>
</table>
</td></tr>
<tr><td>Rückgabe</td><td>


</td></tr>
</table>

	Name	Beschreibung	
	LeftTries	Im Falle von REJECTED wird hier die Anzahl der verbleibenden möglichen Versuche für die Eingabe der PUK zurückgegeben.	
	Status	Enthält den Ausführungsstatus der Operation siehe 3.5.2	
	PinResult	Wert	Bedeutung
		OK	Prüfung war erfolgreich.
		ERROR	Ein Bearbeitungsfehler ist aufgetreten. Fehlerursache im Feld Error.
		REJECTED	PUK war falsch. Die Anzahl der verbleibenden Versuche ist im Element LeftTries.
		WASBLOCKED	PUK war zum Aufrufzeitpunkt bereits gesperrt
		NOWBLOCKED	PUK ist durch aktuellen Fehlversuch gesperrt
Vorbe-dingungen	keine		
Nachbe-dingungen	keine		

Tabelle 124: TAB_KON_550 Ablauf UnblockPIN

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 { mandantId = $context.mandantId; clientSystemId = $context.clientsystemId; workplaceId = $context.workplaceId; userId = $context.userId; // falls angegeben cardHandle }
3.	TUC_KON_026 „Liefere CardSession“	Ermittle cardSession über TUC_KON_026 { mandantId = $context.mandantId; clientSystemId = $context.clientsystemId; userId = $context.userId; // falls angegeben

		cardHandle }

4.	TUC_KON_021 „PIN entsperren“	Rücksetzen des Fehlbedienungszählers über TUC_KON_021 {
		cardSession;
		workplaceId = $context.workplaceId;
		pinRef = pinRef(PinTyp);
		setNewPIN = SetNewPIN }

5.	Verifikationsergebnis auswerten	Wenn TUC_KON_021 den Returncode BLOCKED liefert, wird dies als erfolgreicher Operationsdurchlauf mit PinResult= NOWBLOCKED zurückgegeben. Wenn TUC_KON_021 mit dem Fehlercode 4064 abbricht, wird dies als erfolgreicher Operationsdurchlauf mit PinResult= WASBLOCKED zurückgegeben.

Tabelle 125: TAB_KON_551 Fehlercodes „UnblockPin“

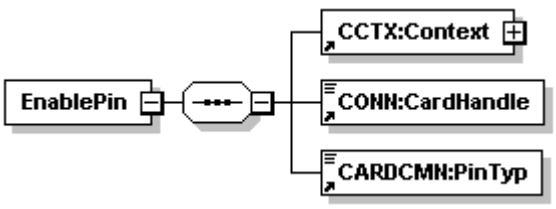
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

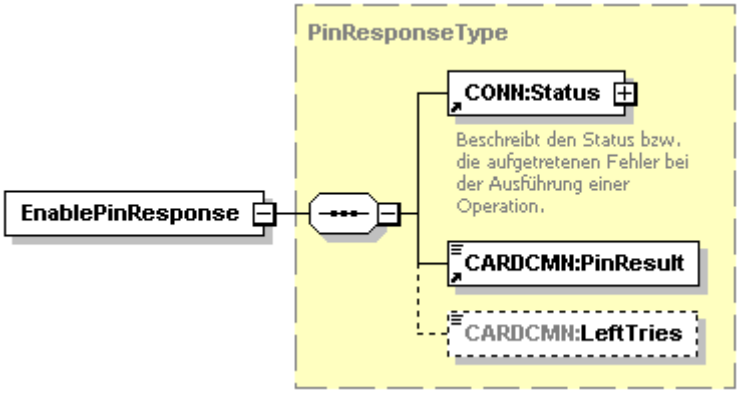
[<=]

4.1.5.5.5 EnablePin

TIP1-A_5487 - Operation EnablePin
Der Konnektor MUSS an der Außenschnittstelle eine Operation EnablePin, wie in Tabelle TAB_KON_242 Operation EnablePin beschrieben, anbieten.

Tabelle 126: TAB_KON_242 Operation EnablePin

Name	`EnablePin`
Beschreibung	Schaltet für eine Multireferenz-PIN das Erfordernis, das Nutzergeheimnis zu verifizieren, ein, so dass der Sicherheitszustand nur durch eine erfolgreiche Benutzerverifikation gesetzt werden kann.
Aufrufparameter	

<table>
<tr><td></td><td>Name</td><td colspan="2">Beschreibung</td></tr>
<tr><td></td><td>Context</td><td colspan="2">MandantId, ClientSystemId, WorkplaceId verpflichtend;</td></tr>
<tr><td></td><td>CardHandle</td><td colspan="2">Adressiert die Karte, deren MRPIN bearbeitet werden soll. Es werden nur eGKs ab Generation 2 unterstützt.</td></tr>
<tr><td></td><td>PinTyp</td><td colspan="2">Gibt an, auf welche MRPIN der Karte die Operation angewendet werden soll.
Erlaubte Werte:
eGK G2: MRPIN.NFD, MRPIN.DPE, MRPIN.GDD
zusätzlich ab eGK G2.1: MRPIN.AMTS</td></tr>
<tr><td>Rückgabe</td><td colspan="3"> 
</td></tr>
<tr><td></td><td>Name</td><td colspan="2">Beschreibung</td></tr>
<tr><td></td><td>Status</td><td colspan="2">Enthält den Ausführungsstatus der Operation, siehe 3.5.2</td></tr>
<tr><td></td><td>PinResult</td><td>Wert</td><td>Bedeutung</td></tr>
<tr><td></td><td></td><td>OK</td><td>Aktivierung war erfolgreich</td></tr>
<tr><td></td><td></td><td>REJECTED</td><td>PIN war falsch. Die Anzahl der verbleibenden Versuche ist im Element LeftTries</td></tr>
<tr><td></td><td></td><td>ERROR</td><td>Ein Bearbeitungsfehler ist aufgetreten. Fehlerursache im Feld Error</td></tr>
<tr><td></td><td></td><td>WASBLOCKED</td><td>PIN war zum Aufrufzeitpunkt bereits gesperrt</td></tr>
<tr><td></td><td></td><td>NOWBLOCKED</td><td>PIN ist durch aktuellen Fehlversuch gesperrt</td></tr>
<tr><td></td><td></td><td>TRANSPORT_PIN</td><td>Dieser Wert wird nicht verwendet</td></tr>
</table>

	LeftTries	Im Falle von Result=REJECTED wird hier die Anzahl der verbleibenden möglichen Versuche zurückgegeben.
Vorbedingung	keine	
Nachbedingung	Für das Erreichen des Sicherheitszustands der MRPIN ist eine Nutzereingabe erforderlich	

Tabelle 127: TAB_KON_243 Ablauf EnablePin

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle }
		Tritt bei der Prüfung ein Fehler auf, so bricht die Operation mit dem Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle cardSession über TUC_KON_026 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		userId = $context.userId;
		cardHandle }

4.	TUC_KON_027 „PIN-Schutz ein-/ausschalten“	Aktiviere das Erfordernis der Benutzerverifikation der MRPIN durch Aufruf des TUC_KON_027 „PIN-Schutz ein-/ausschalten“ {
		cardSession;
		pinRef = PinRef(PinType);
		enable = true}

5.	Verifikationsergebnis auswerten	Als erfolgreicher Operationsdurchlauf wird nur PinResult=OK gewertet. Alle anderen Resultate sind Fehlerfälle, und neben dem Status ist auch PinResult entsprechend zu setzen. Dabei gelten folgende Regeln: Wenn TUC_KON_027 den PIN-Status BLOCKED liefert, wird auf PinResult=NOWBLOCKED abgebildet.

		Wenn TUC_KON_027 mit Fehler 4063 abbricht, wird dies auf PinResult=WASBLOCKED abgebildet.

Tabelle 128: TAB_KON_244 Fehlercodes „EnablePin"

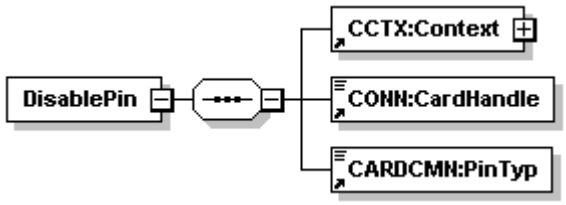
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4072	Technical	Error	Ungültige PIN-Referenz `PinRef`
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=]

4.1.5.5.6 DisablePin

TIP1-A_5488 - Operation DisablePin
Der Konnektor MUSS an der Außenschnittstelle eine Operation DisablePin, wie in Tabelle TAB_KON_245 Operation DisablePin beschrieben, anbieten.

Tabelle 129: TAB_KON_245 Operation DisablePin

<table>
<tr><td>Name</td><td colspan="2">DisablePin</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Schaltet für eine Multireferenz-PIN das Erfordernis, das Nutzergeheimnis zu verifizieren, ab. Die MRPIN verhält sich danach bei allen Zugriffen auf die durch sie geschützten Objekte, als wäre sie freigeschaltet.</td></tr>
<tr><td rowspan="4">Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>Context</td><td>MandantId, ClientSystemId, WorkplaceId verpflichtend;</td></tr>
<tr><td>CardHandle</td><td>Adressiert die Karte, deren MRPIN bearbeitet werden soll. Es werden nur eGKs ab Generation 2 unterstützt.</td></tr>
<tr><td></td><td>PinTyp</td><td>Gibt an, auf welche MRPIN der Karte die Operation angewendet werden soll.
Erlaubte Werte:</td></tr>
</table>

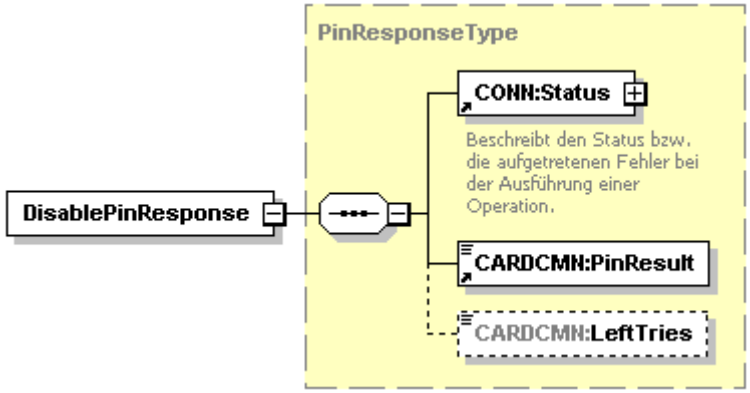
<table>
<tr><td></td><td></td><td colspan="2">eGK G2: MRPIN.NFD, MRPIN.DPE, MRPIN.GDD
zusätzlich ab eGK G2.1: MRPIN.AMTS</td></tr>
<tr><td>Rückgabe</td><td colspan="3"> 
</td></tr>
<tr><td></td><td>Name</td><td colspan="2">Beschreibung</td></tr>
<tr><td></td><td>Status</td><td colspan="2">Enthält den Ausführungsstatus der Operation siehe 3.5.2</td></tr>
<tr><td></td><td>PinResult</td><td>Wert</td><td>Bedeutung</td></tr>
<tr><td></td><td></td><td>OK</td><td>Aktivierung war erfolgreich</td></tr>
<tr><td></td><td></td><td>REJECTED</td><td>PIN war falsch. Die Anzahl der verbleibenden Versuche ist im Element LeftTries</td></tr>
<tr><td></td><td></td><td>ERROR</td><td>Ein Bearbeitungsfehler ist aufgetreten. Fehlerursache im Feld Error</td></tr>
<tr><td></td><td></td><td>WASBLOCKED</td><td>PIN war zum Aufrufzeitpunkt bereits gesperrt</td></tr>
<tr><td></td><td></td><td>NOWBLOCKED</td><td>PIN ist durch aktuellen Fehlversuch gesperrt</td></tr>
<tr><td></td><td></td><td>TRANSPORT_PIN</td><td>Dieser Wert wird nicht verwendet</td></tr>
<tr><td></td><td>LeftTries</td><td colspan="2">Im Falle von Result=REJECTED wird hier die Anzahl der verbleibenden möglichen Versuche zurückgegeben.</td></tr>
<tr><td>Vorbedingung</td><td colspan="3">keine</td></tr>
<tr><td>Nachbedingung</td><td colspan="3">Der Sicherheitszustand der PIN ist dauerhaft (bis zur expliten Aktivierung mit EnablePin) gesetzt, ohne dass eine Nutzereingabe erforderlich wäre</td></tr>
</table>

Tabelle 130: TAB_KON_246 Ablauf DisablePin

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle }
		Tritt bei der Prüfung ein Fehler auf, so bricht die Operation mit dem Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle cardSession über TUC_KON_026 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientSystemId;
		userId = $context.userId;
		cardHandle }

4.	TUC_KON_027 „PIN-Schutz ein-/ausschalten“	Deaktiviere das Erfordernis der Benutzerverifikation der MRPIN durch Aufruf des TUC_KON_027 „PIN-Schutz ein-/ausschalten“ {
		cardSession;
		pinRef = PinRef(PinType);
		enable = false}

5.	Verifikations-ergebnis auswerten	Als erfolgreicher Operationsdurchlauf wird nur PinResult=OK gewertet. Alle anderen Resultate sind Fehlerfälle, und neben dem Status ist auch PinResult entsprechend zu setzen. Dabei gelten folgende Regeln: Wenn TUC_KON_027 den PIN-Status BLOCKED liefert, wird auf PinResult=NOWBLOCKED abgebildet. Wenn TUC_KON_027 mit Fehler 4063 abbricht, wird dies auf PinResult=WASBLOCKED abgebildet.

Tabelle 131: TAB_KON_247 Fehlercodes „DisablePin“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			

4000	Technical	Error	Syntaxfehler
4072	Technical	Error	ungültige PIN-Referenz `PinRef`
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=]

4.1.5.6 Betriebsaspekte

TIP1-A_4592 - Konfigurationswerte des Kartendienstes
Der Konnektor MUSS es einem Administrator ermöglichen, Konfigurationsänderungen gemäß Tabelle TAB_KON_554 vorzunehmen.

Tabelle 132: TAB_KON_554 Konfiguration des Kartendienstes

ReferenzID	Belegung	Bedeutung
`CARD_TIMEOUT_CARD`	`Sekunden`	Maximale Zeit, die ein Aufruf einer Kartenoperation dauern darf, bevor der Aufruf abgebrochen wird. Der Konnektor MUSS sicherstellen, dass dieser Parameter einen Wert besitzt, mit dem ein reibungsloser Betrieb gewährleistet ist, und MUSS dem Administrator die Möglichkeit bieten, diesen Parameter zu konfigurieren.

[<=]

4.1.5.6.1 TUC_KON_025 "Initialisierung Kartendienst"

TIP1-A_4593 - TUC_KON_025 „Initialisierung Kartendienst"
Der Konnektor MUSS den technischen Use Case „Initialisierung Kartendienst" gemäß TUC_KON_025 umsetzen.

Tabelle 133: TAB_KON_555 - TUC_KON_025 „Initialisierung Kartendienst"

Element	Beschreibung
Name	TUC_KON_025 „Initialisierung Kartendienst"
Beschreibung	Nach dem Start des Kartendienstes MUSS der Konnektor für alle gesteckten Karten den TUC_KON_001 {ctId, slotId } aufrufen und CM_CARD_LIST befüllen.
Auslöser	der Kartendienst wird gestartet
Vorbedingungen	Kartenterminaldienst wurde gestartet
Eingangsdaten	CTM_CT_LIST
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	Aktuelle CM_CARD_LIST

Standardablauf	1. Rufe TUC_KON_001 „Karte öffnen“ 2. Wiederhole, bis für alle gesteckten Karten ein Eintrag in CM_CARD_LIST existiert.
Varianten/Alternativen	keine
Fehlerfälle	keine
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

[<=]

4.1.5.6.2 Kartenübersicht und PIN-Management

TIP1-A_5110 - Übersicht über alle verfügbaren Karten
Die Administrationsoberfläche MUSS dem Administrator eine Übersichtsseite anbieten, die alle in CM_CARD_LIST enthaltenen Karten listet.
In dieser Übersichtsseite muss zu jeder Karte dargestellt werden:

- CARD.CTID
- CT(CARD.CTID).HOSTNAME
- CARD.SLOTNO
- CARD.TYPE
- CARD.INSERTTIME
- CARD.CARDHOLDERNAME

Ferner MÜSSEN auf Verlangen des Administrators zu jeder Karte neben den obigen Informationen auch folgende Details angezeigt werden:

- CARD.ICCSN
- CARD.CARDVERSION
- CARD.CERTEXPIRATIONDATE

[<=]

TIP1-A_5111 - PIN-Management der SM-Bs für den Administrator
Über die Administrationsoberfläche MUSS der Administrator für jede in der Übersichtsseite angezeigte Karte vom Typ SM-B die folgenden TUCs für die PIN.SMC auslösen können.
Für diese MUSS er einen der gemäß Kapitel 4.1.1.6 [TIP1-A_4526] definierten Mandanten auswählen können:

- TUC_KON_012 „PIN verifizieren“
- TUC_KON_019 „PIN ändern“
- TUC_KON_021 „PIN entsperren“
- TUC_KON 022 „Liefere PIN-Status“

Die Eingabe der PIN SOLL von jedem vom Informationsmodell her zulässigen Kartenterminal aus möglich sein.
[<=]

Der Konnektor kann den Administrator zur Laufzeit entscheiden lassen, an welchem Kartenterminal die PIN eingegeben werden soll, indem er ihn wählen lässt, ob er die PIN am Kartenterminal eingibt, in dem die betroffene SM-B steckt, oder ihn den Arbeitsplatz wählen lässt, von dem aus er die Remote-PIN eingibt.

4.1.6 Systeminformationsdienst

Der Systeminformationsdienst stellt Basisdiensten, Fachmodulen und Clientsystemen sowohl aktiv (Push-Mechanismus) wie passiv (Pull-Mechanismus) Informationen zur Verfügung. Dabei erhebt er selbst keine Daten, sondern dient nur als zentraler Mechanismus, der von anderen Basisdiensten und Fachmodulen zur Verteilung und Bereitstellung derer Informationen verwendet werden kann.

Innerhalb des Systeminformationsdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „EVT"
- Konfigurationsparameter: „EVT_"

Push-Mechanismus

Der Push-Mechanismus des Systeminformationsdienstes hat die Aufgabe, Ereignisse von internen Ereignisquellen im Konnektor (z. B. von anderen Basisdiensten wie Kartendienst, Kartenterminaldienst oder Fachmodulen) an alle Basisdienste und Fachmodule sowie an die bei ihm registrierten Ereignisempfänger (Clientsysteme) weiterzuleiten.

Die Namen der Ereignisse, die Topics, sind als Baumstruktur organisiert und werden mittels „/"-getrennter Liste angegeben (z. B. „Auslöser/Ereigniskategorie1/…/Ereignis1"). Die konkreten Topics werden innerhalb der einzelnen Funktionsmerkmale kontextbezogen definiert und im Anhang in einer zentralen Liste übersichtlich dargestellt.

Clientsysteme können sich für den Empfang bestimmter Ereigniskategorien (Topics) anmelden. Der Systeminformationsdienst übernimmt dementsprechend bei der Verteilung der Ereignisse auch eine Filterfunktion für die weiterzuleitenden Ereignisse.

Die Zustellung der Ereignisse erfolgt unidirektional über eine Netzschnittstelle, deren Kommunikationsendpunkt („Ereignissenke") vom Clientsystem realisiert werden muss. Zur Übertragung der Daten wird ein konnektoreigenes Protokoll cetp (Connector Event Transport Protocol) verwendet.

Pull-Mechanismus

Der Pull-Mechanismus des Systeminformationsdienstes hat die Aufgabe sowohl Zustandswerte als auch statische Informationen des Konnektors selbst als auch von den über ihn verwalteten Ressourcen durch Fachmodule und Clientsysteme abrufbar zu machen. Dabei können entweder Listen von Ressourcen oder Details zu einzelnen Ressourcen abgerufen werden.

Die folgenden Unterkapitel regeln:

- Das Kommunikationsprotokoll cetp
- Die Struktur der Ereignisnachricht
- Die TUCs für die Ereignisverteilung (PUSH)

- Die TUCs und Operationen der Außenschnittstelle für den Abruf von Informationen (PULL)
- Einstellungen, die der Administrator zur Steuerung des Verhaltens vornehmen kann.

4.1.6.1 Funktionsmerkmalweite Aspekte

TIP1-A_4594 - Richtung bei Verbindungsaufbau des Systeminformationsdienstes
Der Konnektors MUSS zur Übertragung von Ereignissen eine cetp-Verbindung zu der Ereignissenke des Clientsystems aufbauen, die das Clientsystem beim Operationsaufruf Subscribe per `EventTo` angegeben hatte.
[<=]

TIP1-A_5536 - Connector Event Transport Protocol über TCP
Der Konnektor MUSS das Anwendungsprotokoll cetp (Connector Event Transport Protocol) ausschließlich über das Transportprotokoll TCP (gebenenfalls TLS gesichert) anbieten.
[<=]

TIP1-A_4595 - Gesicherte Übertragung von Ereignissen
Der Konnektor MUSS zur Übertragung der Ereignisse eine gesicherte Verbindung (TLS) verwenden, die vom Konnektor als TLS-Client initiiert wurde, wenn ANCL_TLS_MANDATORY=Enabled.
Der Konnektor muss sich beim Aufbau der TLS-Sitzung gegenüber dem Clientsystem authentisieren, wenn dieses eine Authentisierung im Rahmen des TLS-Handshakes anfordert.
Die Schalter ANCL_CAUT_MODE und ANCL_CAUT_MANDATORY wirken für die Übertragung der Ereignisse nicht.
[<=]

Für die Übermittlung der Ereignisse wurde ein leichtgewichtiges Protokoll gewählt, da vom Clientsystem keine Antwort auf Anwendungsebene erwartet wird.

TIP1-A_4596 - Nachrichtenaufbau und -kodierung des Systeminformationsdienstes
Der Konnektors MUSS Ereignisse an Ereignissenken mittels Nachrichten verteilen, die gemäß TAB_KON_030 „Ereignisnachricht" aufgebaut sind. Der Konnektor MUSS die Nachrichten mit der Zeichenkette „CETP" beginnen, gefolgt von der Länge der folgenden Ereignisnachricht in Anzahl Bytes. Das vier Byte lange Längenfeld MUSS in der Byte-Reihenfolge Big-Endian codiert werden (das hochwertigste Byte wird als erstes übertragen).

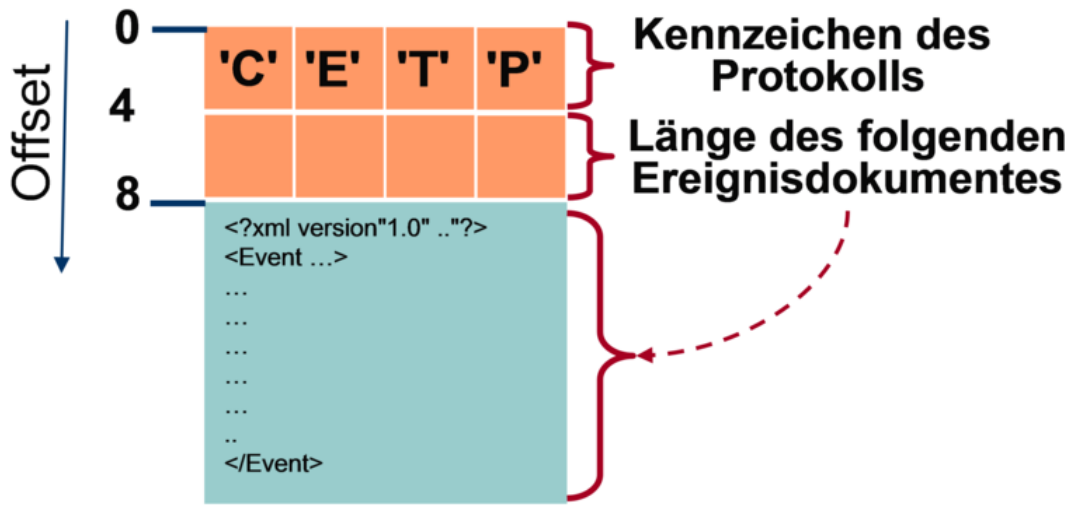
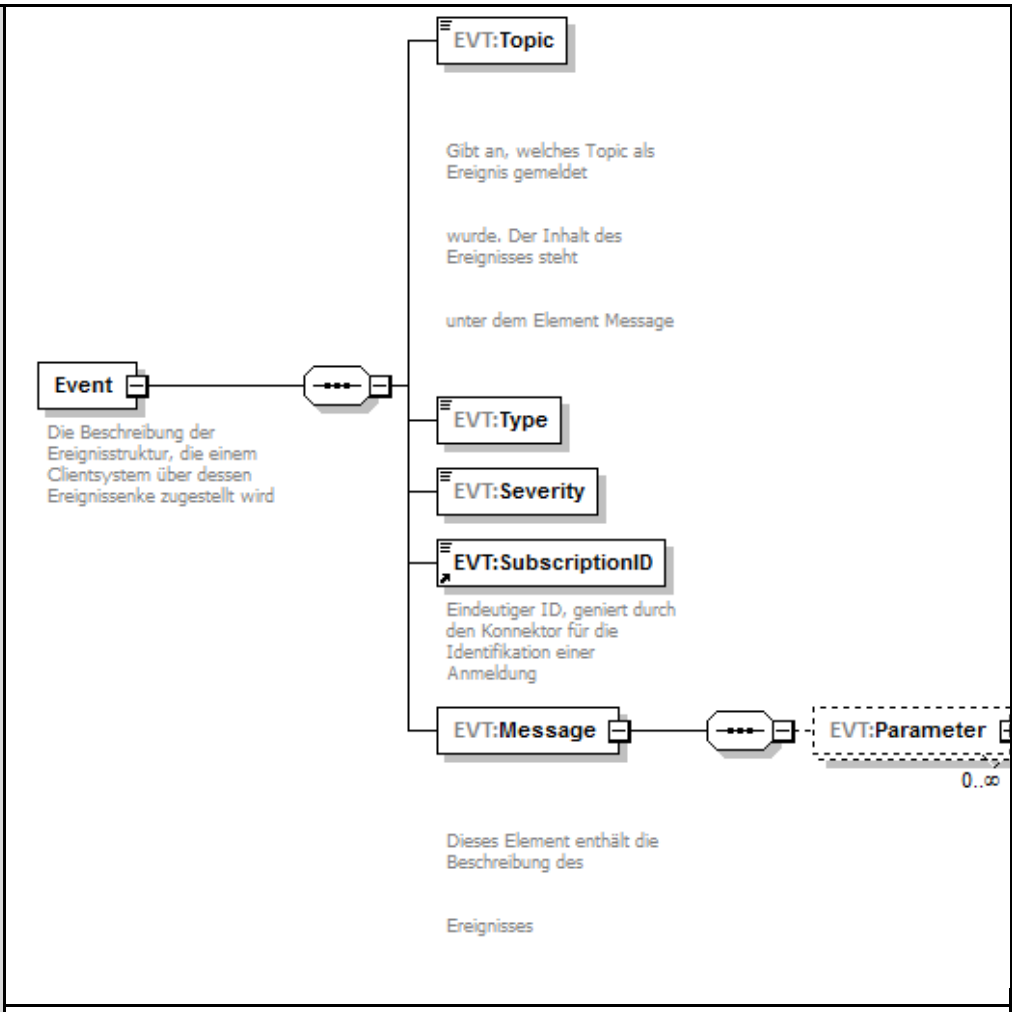
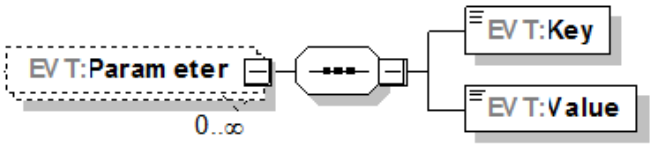


Abbildung 11: PIC_KON_022 Grundsätzlicher Aufbau der Ereignisnachricht

Tabelle 134: TAB_KON_030 Ereignisnachricht

Beschreibung	Die Ereignisnachricht, die zur Ereignissenke gesendet wird, ist ein XML-Dokument. Die Ereignisnachricht wird in den „Umschlag" Event gepackt. Wenn ein mandantenfähiges Clientsystem mehrere Anwendungskonnektoren verwendet, dann kann es die erhaltenen Ereignisbenachrichtigungen anhand der Subscription-ID einem Mandanten zuordnen.





`Topic`	Topic der Ereignisnachricht
`Type`	Typ der Ereignisnachricht (Security, Operation, Infrastructure)
`Severity`	Schwere der Ereignisnachricht (Info, Warning, Error, Fatal)
`SubscriptionID`	Identifikator der Anmeldung, der vom Konnektor bei der Operation `Subscribe` für die Anmeldung des jeweiligen Clientsystems vergeben wurde.
`Message`	Dieses Element enthält die Ereignisnachricht. Der Inhalt

		ist abhängig vom Topic und wird mittels „Key-Value"-Parametern übertragen.
	`Message/Parameter/Key`	Name des Parameters (String), case sensitiv
	`Message/Parameter/Value`	Wert des Parameters (String)
Hinweise	Das XML-Dokument MUSS UTF-8-codiert sein.	

[<=]

4.1.6.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.1.6.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.1.6.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.6.4.1 TUC_KON_256 „Systemereignis absetzen"

TIP1-A_4598 - TUC_KON_256 „Systemereignis absetzen"
Der Konnektor MUSS für den PUSH-Mechanismus des Systeminformationsdienstes den technischen Use Case TUC_KON_256 „Systemereignis absetzen" umsetzen.

Tabelle 135: TAB_KON_556 - TUC_KON_256 „Systemereignis absetzen"

Element	Beschreibung
Name	TUC_KON_256 „Systemereignis absetzen"
Beschreibung	Dieser TUC verteilt ein Ereignis im Konnektor intern (d.h. an Basisdienste und Fachmodule) sowie an Clientsysteme, die sich für den Empfang angemeldet haben (Operation `Subscribe`). Zusätzlich wird, bei gesetztem Flag, das Ereignis durch den Protokollierungsdienst protokolliert.
Auslöser	Aufruf durch Basisdienst oder Fachmodul
Vorbedingungen	Fall Topic = „BOOTUP/BOOTUP_COMPLETE": Zu allen URLs von clientseitigen Endpunkten, wie sie bei der `Subscribe`-Operation angegeben wurden, ist in der Subscription-Verwaltung des Konnektors eine TerminationTime gespeichert. Sie wird jeweils auf den Wert der TerminationTime der am längsten gültigen Subscription zu dem jeweiligen Endpunkt gesetzt. Die URLs von clientseitigen Endpunkten müssen bis zum Ablauf ihrer TerminationTime auch über Bootups hinweg gespeichert werden. Vor dem Versenden des BOOTUP_COMPLETE-Events werden sämtliche Subscriptions, jedoch nicht die URLs gelöscht. Bei

	Ablauf ihrer TerminationTime werden nach dem Versenden des BOOTUP_COMPLETE-Events auch die URLs gelöscht.
Eingangsdaten	Attribute des zu versendenden Ereignisses: • topic (Name des Ereignisses) • eventType [EventType] (Wenn statt eines EventType ein ErrorType übergeben wird, so wird der EventType daraus abgeleitet. Typ des Events: Op = Operation, Sec = Security, Infra = Infrastructure) • severity [EventSeverity] (Schwere des Ereignisses: Info = Information, Warn = Warning, Err = Error, Fatal) • parameters (weitere Parameter als key-value-Paare) Arbeitsanweisungen: • doLog [Boolean] – *optional; default = true* (Schalter „Schreibe Protokolleintrag“) • doDisp [Boolean] – *optional; default = true* (Schalter „An Clientsysteme versenden“) Die Bezeichnungen Op, Sec, Infra, Info, Warning, Err, Fatal werden nur in diesem Dokument verwendet und sind als Abkürzungen für die Werte Operation, Security, Infrastructure, Information, Warning, Error, Fatal in den jeweiligen Ereignisnachrichten gemäß Schema EventService.xsd zu verstehen.
Komponenten	Konnektor
Ausgangsdaten	Keine
Nachbedingungen	

Standardablauf	Für das übergebene Ereignis werden folgende Schritte durchgeführt:
	1. Das Ereignis wird an alle Basisdienste und Fachmodule des Konnektors gesendet.
	2. Wenn doLog = true, erfolgt der Aufruf von TUC_KON_271 {
	eventType = $Event.eventType
	(mit eventType = „Op“, wenn $Event.eventType in {„Op“, „Infra“}
	mit eventType = „Sec“, wenn $Event.eventType gleich "Sec")
	severity=$Event. severity;
	parameters= ($Event.topic, $Event.parameters) }
	Die Einschränkungen zur Protokollierung personenbezogener Daten
	gemäß TIP1-A_4710 müssen beim Aufruf von TUC_KON_271 beachtet werden.

	3. Falls doDisp = false ist, wird an dieser Stelle abgebrochen.
	4. Das für den Versand an Clientsysteme benötigte XML-Dokument des
	Ereignisses wird aufgebaut (Element „Event" gemäß EventService.XSD).
	5. Setze $subscriptionList = Liste der Clientsystem-Abonnements, die
	durch die Operationen Subscribe/Unsubscribe gepflegt werden und
	deren TerminationTime > Systemzeit.
	Im Folgenden durchläuft diese Liste der Reihe nach drei Filter. Nach
	dem letzten Filterschritt enthält $subscriptionList nur noch die Abonnements, an die das Ereignis versendet werden soll.
	a. Filtern nach Topics:
	für jede $subscription in $subscriptionList {
	wenn $event.topic nicht mit $subscription.topic beginnt oder übereinstimmt (case insensitiver Vergleich),
	dann entferne $subscription aus $subscriptionList
	}
	b. Filtern nach Zugriffsberechtigung:
	für jede $subscription in $subscriptionList {
	wenn TUC_KON_000 mit einem Fehler abgeschlossen wird, dann entferne $subscription aus $subscriptionList.
	Wenn cardHandle in parameters übergeben wurde, dann
	TUC_KON_000 {
	mandantId = $subscription.context.mandantId;
	clientSystemId = $subscription.context.clientsystemId;
	workplaceId = $subscription.context.workplaceId;
	ctId = $parameters.value für $parameters.key = „ctId"
	cardHandle = $parameters.value für $parameters.key = „cardHandle";
	needCardSession = false;
	allWorkplaces = false
	}
	oder im Fall nicht gegebenes cardHandle
	TUC_KON_000 {
	mandantId = $subscription.context.mandantId;
	clientSystemId = $subscription.context.clientsystemId;
	workplaceId = $subscription.context.workplaceId;
	ctId = $parameters.value für $parameters.key = „ctId"
	needCardSession = false;
	allWorkplaces = false
	} }
	c. Filtern nach XPath-Filter in Subscription ([XPATH]):
	für jede $subscription in $subscriptionList {
	wenn der XPath-Ausdruck $subscription.filter angewandt auf das als XML-Dokument dargestellte Ereignis
	ein leeres Ergebnis liefert,

	dann entferne $subscription aus $subscriptionList
	}
	6. Versenden:
	für jede $subscription in $subscriptionList {
	versende das Ereignis an $subscription.eventTo
	}
	Für das versendete Ereignis wird keine Antwort durch das Clientsystem erwartet.

Varianten/ Alternativen	Fall Topic = „BOOTUP/BOOTUP_COMPLETE":
	4. Das für den Versand an Clientsysteme benötigte XML-Dokument
	des Ereignisses wird aufgebaut (Element „Event" gemäß EventService.XSD, SubscriptionID als leeres Element).
	5. Setze $urlList = Liste der URLs von clientseitigen Endpunkten, wie
	sie bei der `Subscribe`-Operation angegeben wurden. Clientsysteme,
	deren Subscription-URL beim Einschalten des Konnektors noch nicht
	gelöscht waren, müssen benachrichtigt werden, auch wenn dann
	bereits deren TerminationTime < Systemzeit ist.
	Versenden:
	für jede $url in $urlList {
	versende das Ereignis an $url
	}
	Für das versendete Ereignis wird keine Antwort durch das Clientsystem erwartet. Dadurch wird bei einer Nichtzustellung auch kein erneuter
	Versand des Ereignisses angestoßen, da der Konnektor keine Kenntnis
	über den Erfolg einer Ereignisübermittlung hat.

Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (→5c) Fehler bei der Auswertung des XPath-Ausdrucks: Fehlercode: 4095, nur für die jeweilige Abonnement-Prüfung.
Fachliche Fehlermeldung	Keine
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Abbildung PIC_KON_112 Aktivitätsdiagramm zu „Systemereignis absetzen"

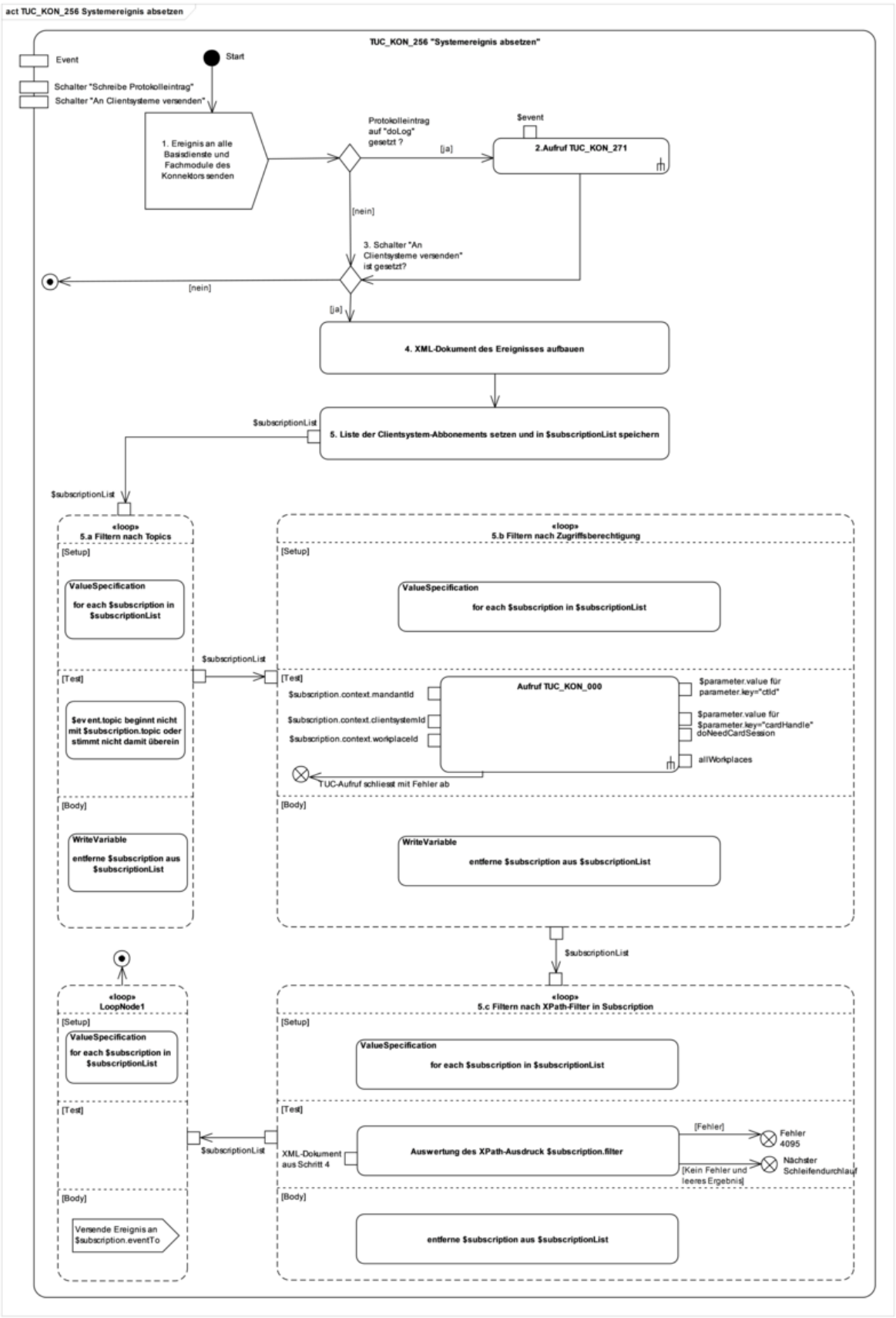

act TUC_KON_256 Systemereignis absetzen
TUC_KON_256 "Systemereignis absetzen"
Event
Start
Schalter "Schreibe Protokolleintrag"
Schalter "An Clientsysteme versenden"
1. Ereignis an alle Basisdienste und Fachmodule des Konnektors senden
Protokolleintrag auf "doLog" gesetzt ?
[ja]
$event
2.Aufruf TUC_KON_271
[nein]
3. Schalter "An Clientsysteme versenden" ist gesetzt?
[nein]
[ja]
4. XML-Dokument des Ereignisses aufbauen
$subscriptionList
5. Liste der Clientsystem-Abbonements setzen und in $subscriptionList speichern
$subscriptionList
«loop»
5.a Filtern nach Topics
[Setup]
ValueSpecification
for each $subscription in $subscriptionList
[Test]
$event.topic beginnt nicht mit $subscription.topic oder stimmt nicht damit überein
[Body]
WriteVariable
entferne $subscription aus $subscriptionList
«loop»
5.b Filtern nach Zugriffsberechtigung
[Setup]
ValueSpecification
for each $subscription in $subscriptionList
$subscriptionList
[Test]
$subscription.context.mandantId
$subscription.context.clientsystemId
$subscription.context.workplaceId
Aufruf TUC_KON_000
$parameter.value für parameter.key="ctId"
$parameter.value für $parameter.key="cardHandle"
doNeedCardSession
allWorkplaces
TUC-Aufruf schliesst mit Fehler ab
[Body]
WriteVariable
entferne $subscription aus $subscriptionList
$subscriptionList
«loop»
LoopNode1
[Setup]
ValueSpecification
for each $subscription in $subscriptionList
[Test]
[Body]
Versende Ereignis an $subscription.eventTo
$subscriptionList
«loop»
5.c Filtern nach XPath-Filter in Subscription
[Setup]
ValueSpecification
for each $subscription in $subscriptionList
[Test]
XML-Dokument aus Schritt 4
Auswertung des XPath-Ausdruck $subscription.filter
[Fehler]
Fehler 4095
[Kein Fehler und leeres Ergebnis]
Nächster Schleifendurchlauf
[Body]
entferne $subscription aus $subscriptionList

Abbildung 12: PIC_KON_112 Aktivitätsdiagramm zu „Systemereignis absetzen“

Tabelle 136: TAB_KON_557 Fehlercodes TUC_KON_256 „Systemereignis absetzen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4095	Technical	Error	Fehler bei der Auswertung eines XPath-Ausdruck

[<=]

4.1.6.4.2 TUC_KON_252 „Liefere KT_Liste“

TIP1-A_4599 - TUC_KON_252 „Liefere KT_Liste”
Der Konnektor MUSS den technischen Use Case TUC_KON_252 „Liefere KT_Liste” umsetzen.

Tabelle 137: TAB_KON_558 – TUC_KON_252 „Liefere KT_Liste“

Element	Beschreibung
Name	TUC_KON_252 „Liefere KT_Liste“
Beschreibung	Dieser TUC liefert eine Liste der Kartenterminals, die unter Beachtung der Eingangsdaten verfügbar/erreichbar sind.
Auslöser	Aufruf durch ein Clientsystem (Operation `GetCardTerminals`) oder ein Fachmodul
Vorbedingungen	Keine
Eingangsdaten	• workplaceId - *optional* (Arbeitsplatz ID) • clientSystemId (Clientystem ID) • mandantId (Mandanten ID)
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	• cardTerminals (Liste der Kartenterminals, die den angegebenen Arbeitsplätzen, Mandanten und Clientsystemen zugeordnet sind bzw. auf die diese zugreifen dürfen (siehe Zugriffsberechtigungsdienst), sowie deren aktuelle Verfügbarkeit. Verfügbarkeit bedeutet im Falle eines eHealth-Kartenterminals, dass der Konnektor eine Verbindung zum Kartenterminal aktuell hält.)
Nachbedingungen	• Der Zustand der Kartenterminals bleibt unverändert

Standardablauf	1. Erstellen der Liste aller Kartenterminals, auf die der angegebene Mandant und das angegebene Clientsystem zugreifen dürfen (siehe Zugriffsberechtigungsdienst) a. Wurde der optionale Parameter workplaceId ID übergeben, so werden nur die Kartenterminals in die Liste aufgenommen, die diesem Arbeitsplatz zugeordnet sind (siehe Zugriffsberechtigungsdienst). Dazu zählen insbesondere nicht die als entfernte Kartenterminals bezeichneten KT. b. Fehlt dieser Parameter, so werden alle Kartenterminals in die Liste aufgenommen, die sowohl dem Clientsystem als auch dem Mandanten zugeordnet sind. 2. Rückgabe der Liste cardTerminals (der in Schritt 1 erstellten Liste) mit Angaben zu jedem Kartenterminal gemäß Schema „`Eventservice.xsd`“.
Varianten/Alternativen	Keine
Fehlerfälle	Keine
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

[<=]

4.1.6.4.3 TUC_KON_253 „Liefere Karten_Liste“

TIP1-A_4600 - TUC_KON_253 „Liefere Karten_Liste”
Der Konnektor MUSS den technischen Use Case TUC_KON_253 „Liefere Karten_Liste” umsetzen.

Tabelle 138: TAB_KON_559 – TUC_KON_253 „Liefere Karten_Liste“

Element	Beschreibung
Name	TUC_KON_253 „Liefere Karten_Liste“
Beschreibung	Dieser TUC liefert eine Liste der gesteckten Karten, die unter Beachtung der Eingangsdaten verfügbar/erreichbar sind.
Auslöser	Aufruf durch ein Clientsystem (Operation `GetCards`) oder ein Fachmodul
Vorbedingungen	Keine
Eingangsanforderung	Keine
Eingangsdaten	• workplaceId – *optional* (Arbeitsplatz-ID)

	• clientSystemId (Clientystem ID) • cardTerminalId - *optional; verpflichtend, wenn slotId übergeben wird* (Kartenterminalidentifikator) • slotId – *optional* (Nummer des Slots, beginnend bei 1) • mandantId (Mandanten ID) • cardType – *optional* (Kartentyp gemäß Tabelle TAB_KON_500)
Komponenten	Konnektor, Kartenterminal, Karte
Ausgangsdaten	• cards (Liste der gesteckten Karten einschließlich der Informationen für CARD:card, auf die der Mandant und das Clientsystem von dem Arbeitsplatz aus zugreifen dürfen (siehe Zugriffsberechtigungsdienst)). Wird workplaceId nicht übergeben, so werden alle vom Clientsystem und dem Mandant erreichbaren Kartenterminals in die Liste aufgenommen. Die Eingangsdaten dienen als Filter, welche Karten in cards zurückgegeben werden. Beispiel: Falls cardTerminalId angegeben ist, werden nur Karten in die Liste aufgenommen, die im entsprechenden Kartenterminal stecken.)
Nachbedingungen	• Der Zustand der Kartenterminals und der Karten bleibt unverändert
Standardablauf	1. Erstellen der Liste aller Karten, auf die der angegebene Mandant und das angegebene Clientsystem zugreifen dürfen (siehe Zugriffsberechtigungsdienst). a. Wurde cardTerminalId übergeben, dann nur Karten berücksichtigen, die dem dadurch referenziertem Kartenterminal zugeordnet sind. b. Wurde außer cardTerminalId auch slotId übergeben, so ist nur die Karte zu berücksichtigen, die in dem angegebenen Slot steckt. c. Wurde workplaceId übergeben, so werden nur die Karten in die Liste aufgenommen, auf die von diesem Arbeitsplatz aus zugegriffen werden darf (siehe „Zugriffsberechtigung Ressourcen“). d. Wurde cardType übergeben, so werden nur die Karten in die Liste aufgenommen, die dem Kartentyp in CardType entsprechen.

	2. Rückgabe cards, der in Schritt 1 erstellten Liste mit Angaben zu jeder Karte gemäß Schema „Eventservice.xsd“.
Varianten/ Alternativen	Keine
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (→1 a) Ungültige Kartenterminal-ID: Fehlercode: 4007
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 139: TAB_KON_560 Fehlercodes TUC_KON_253 „Liefere Karten_Liste“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4007	Technical	Error	ungültige Kartenterminal-ID

[<=]

4.1.6.4.4 TUC_KON_254 „Liefere Ressourcendetails“

TIP1-A_4602 - TUC_KON_254 „Liefere Ressourcendetails”
Der Konnektor MUSS den technischen Use Case TUC_KON_254 „Liefere Ressourcendetails” umsetzen.

Tabelle 140: TAB_KON_561 - TUC_KON_254 „Liefere Ressourcendetails“

Element	Beschreibung
Name	TUC_KON_254 „Liefere Ressourcendetails“
Beschreibung	Dieser TUC liefert Detailinformationen zu einer Ressource (KT, Karte) oder dem Konnektor
Auslöser	Aufruf durch ein Clientsystem (Operation GetResourceInformation) oder ein Fachmodul
Vorbedingungen	Keine
Eingangsanforderung	Keine
Eingangsdaten	• clientSystemId (Clientsystem ID) • mandantId (Mandanten ID) • workplaceId – *optional* (Arbeitsplatz ID)

	• cardTerminalId – *optional* (Kartenterminal ID) • slotId – *optional/zulässig nur, wenn auch cardTerminalId angegeben ist* (Kartenslot-Nummer) • cardHandle – *optional* • iccsn – *optional*
Komponenten	Konnektor, Kartenterminal, Karte, HSM
Ausgangsdaten	• resource (Informationsobjekt einer Ressource (Kartenterminal, Karte, HSM))
Nachbedingungen	• Der Zustand der Kartenterminals, Karten und HSM bleibt unverändert
Standardablauf	1. Falls cardTerminalId und slotId übergeben wurde oder in den Eingangsparametern iccsn oder cardHandle enthalten ist, wird ein Informationsobjekt der Karte, die sich in dem angegebenen Slot befindet bzw. die über die Iccsn oder das CardHandle identifiziert werden kann, zurückgegeben. 2. Falls cardTerminalId, aber keine slotId übergeben wurde, wird ein Informationsobjekt des Kartenterminals zurückgegeben. 3. Wurde weder iccsn, cardHandle, cardTerminalId noch eine slotId übergeben, so wird ein Informationsobjekt des Konnektors zurückgegeben. Für das Element ErrorCondition ist aus der Tabelle TAB_KON_503 Betriebszustand_Fehlerzustandsliste der Text aus der Spalte ErrorCondition zu übernehmen, ggf. mit den in dieser Spalte angegebenen Parameterwerten. Vor der Rückgabe der Informationen über eine Ressource wird geprüft, ob der angegebene Mandant und das angegebene Clientsystem darauf zugreifen dürfen (siehe Zugriffsberechtigungsdienst). Wurde zusätzlich der optionale Parameter workplaceId übergeben, so wird auch geprüft, ob die Ressource diesem Arbeitsplatz zugeordnet ist. Die Rückgabe der Informationen erfolgt gemäß dem Schema „`Eventservice.xsd`“.
Varianten/ Alternativen	Keine
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (→1) Ungültige Kartenterminal-ID: Fehlercode: 4007 (→1) Ungültige Kartenslot-ID: Fehlercode: 4097 (→1) Keine Karte im angegebenen Slot: Fehlercode: 4098 (→1) Keine Karte mit angegebener Iccsn gefunden: Fehlercode: 4099

	(→1) Karten-Handle ungültig: Fehlercode: 4101 (→2) Ungültige Kartenterminal-ID: Fehlercode: 4007
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 141: TAB_KON_562 Fehlercodes TUC_KON_254 „Liefere Ressourcendetails"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4007	Technical	Error	ungültige Kartenterminal-ID
4097	Technical	Error	ungültige Kartenslot-ID
4098	Technical	Error	keine Karte im angegebenen Slot gefunden
4099	Technical	Error	keine Karte zur angegebenen Iccsn gefunden
4101	Technical	Error	Karten-Handle ungültig

[<=]

4.1.6.5 Operationen an der Außenschnittstelle

TIP1-A_4603-02 - Basisanwendung Systeminformationsdienst

Der Konnektor MUSS für Clients eine Basisanwendung Systeminformationsdienst anbieten.

Tabelle 142 TAB_KON_029 Basisanwendung Systeminformationsdienst

Name	**`EventService`**	
Version	7.2.0 (WSDL-Version) 7.2.1 (XSD-Version)	
Namensraum	Siehe GitHub	
Namensraum-Kürzel	`EVT für Schema und EVTW für WSDL`	
Operationen	**Name**	**Kurzbeschreibung**
	`GetCardTerminals`	Auflistung der verfügbaren Kartenterminals
	`GetCards`	Auflistung der gesteckten Karten

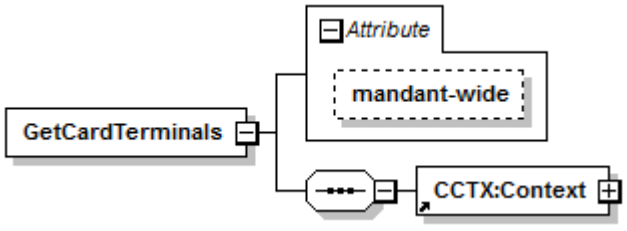
	GetResourceInformation	Liefert Details zu einer Ressource (Kartenterminal, Karte, HSM)
	Subscribe	Anmeldung der Zustellung von Ereignissen
	Unsubsribe	Abmelden von der Zustellung von Ereignissen
	RenewSubscriptions	Gültigkeit bestehender Subscriptions verlängern
	GetSubscriptions	Abfrage der angemeldeten Zustellungen von Ereignissen
WSDL	EventService.wsdl	
Schema	EventService.xsd	

[<=]

4.1.6.5.1 GetCardTerminals

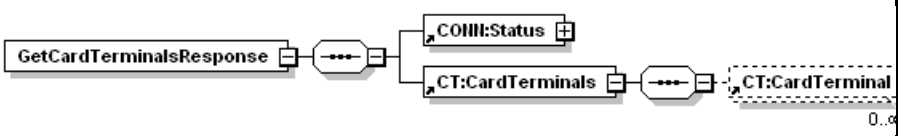
TIP1-A_4604 - Operation GetCardTerminals
Der Konnektors MUSS an der Außenschnittstelle eine Operation GetCardTerminals, wie in Tabelle TAB_KON_563 „Operation GetCardTerminals" beschrieben, anbieten.

Tabelle 143: TAB_KON_563 Operation GetCardTerminals

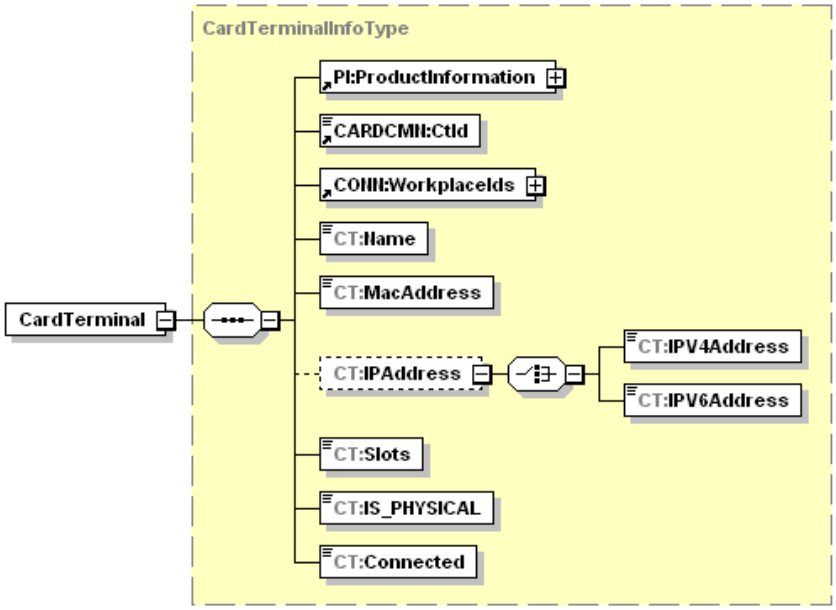
Name	GetCardTerminals	
Beschrei-bung	Liefert die Liste der Kartenterminals, auf die der aufrufende Mandant und das aufrufende Clientsystem zugreifen dürfen (siehe Zugriffsberechtigungsdienst) sowie deren aktuelle Verfügbarkeit. Verfügbarkeit bedeutet im Falle eines eHealth-Kartenterminals, dass der Konnektor eine Verbindung zum Kartenterminal aktuell hält.	
Aufruf-parameter		
	Name	Beschreibung
	@mandant-wide	Wenn „true", werden alle Kartenterminals zurückgegeben, auf die der Mandant und das

		aufrufende Clientsystem zugreifen dürfen. Wenn „false" (Standardbelegung), werden nur Kartenterminals zurückgegeben, auf die von dem im Aufrufkontext spezifizierten Arbeitsplatz zugegriffen werden darf.
	Context	Aufrufkontext

Rückgabe



Name	Beschreibung
Status	Ergebnis der Operation



Die Liste der Kartenterminals

Name	Beschreibung
Product Information	Produktinformationen gemäß [gemSpec_OM] und dem Schema „ProductInformation.xsd" zu formatieren.
CtId	Eineindeutige Identifikation des Kartenterminals
WorkplaceIds	Die Liste der Arbeitsplätze, denen das Kartenterminal als lokales Kartenterminal zugeordnet ist. Insbesondere für Entfernte Kartenterminals kann diese Liste leer sein (siehe TUC_KON_252).
Name	Sprechender Name des Kartenterminals
MacAddress	MAC-Adresse des Kartenterminals

	IPAddress	IP-Adresse des Kartenterminals
	Slots	Anzahl der Slots des Kartenterminals
	IS_PHYSICAL	Attribut des Kartenterminals das anzeigt ob es sich um ein physisches oder logisches Kartenterminal handelt (siehe auch TAB_KON_522 Parameterübersicht des Kartenterminaldienstes)
	Connected	Zeigt an, ob dieses Kartenterminal aktuell verfügbar ist. TRUE – ist verfügbar FALSE – ist nicht verfügbar
Vorbedingungen	Keine	
Nachbedingungen	Der Zustand der Kartenterminals bleibt unverändert.	
Hinweise	Der Aufruf DARF nur den im Konnektor verwalteten, aktuellen Zustand des Kartenterminals liefern und DARF NICHT Abfragen an die Kartenterminals absetzen.	

Der Ablauf der Operation GetCardTerminals ist in Tabelle TAB_KON_564 Ablauf GetCardTerminals beschrieben:

Tabelle 144: TAB_KON_564 Ablauf GetCardTerminals

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession = false;
		allWorkplaces = @mandant-wide }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_252 „Liefere KT_Liste“	Die Liste der Kartenterminals wird erstellt und zurückgegeben. Tritt hierbei ein Fehler auf, so bricht die Operation mit dem Fehler des TUCs ab.
		Wenn @mandant-wide=true dann ermittle die Liste der Kartenterminals für alle Arbeitsplätze des Mandanten für das angegebene Clientsystem durch den Aufruf von TUC_KON_252{
		clientSystemId = $context.ClientSystemId;
		mandantId = $context.mandantId }
		Wenn @mandant-wide=false dann ermittle die Liste der

		Kartenterminals für den Arbeitsplatz des Mandanten für das angegebene Clientsystem entsprechend $context durch den Aufruf von TUC_KON_252{
		workplaceId = $context.workplaceId;
		clientSystemId = $context.ClientSystemId;
		mandantId = $context.mandantId }

Tabelle 145: TAB_KON_823 Fehlercodes „GetCardTerminals"

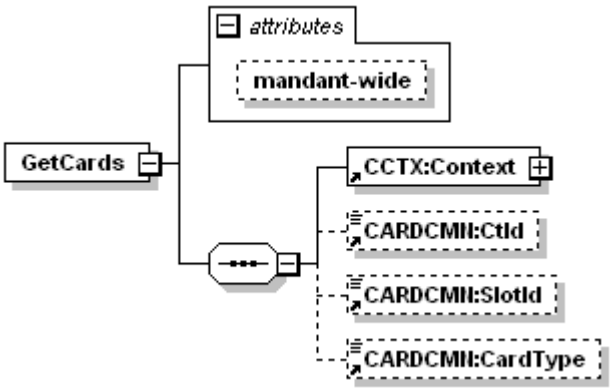
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

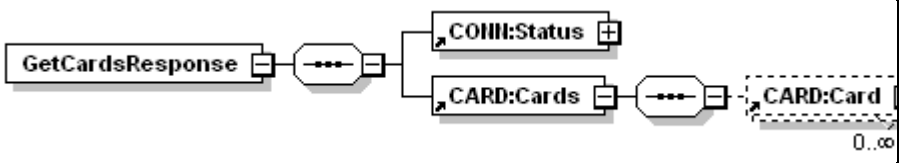
[<=]

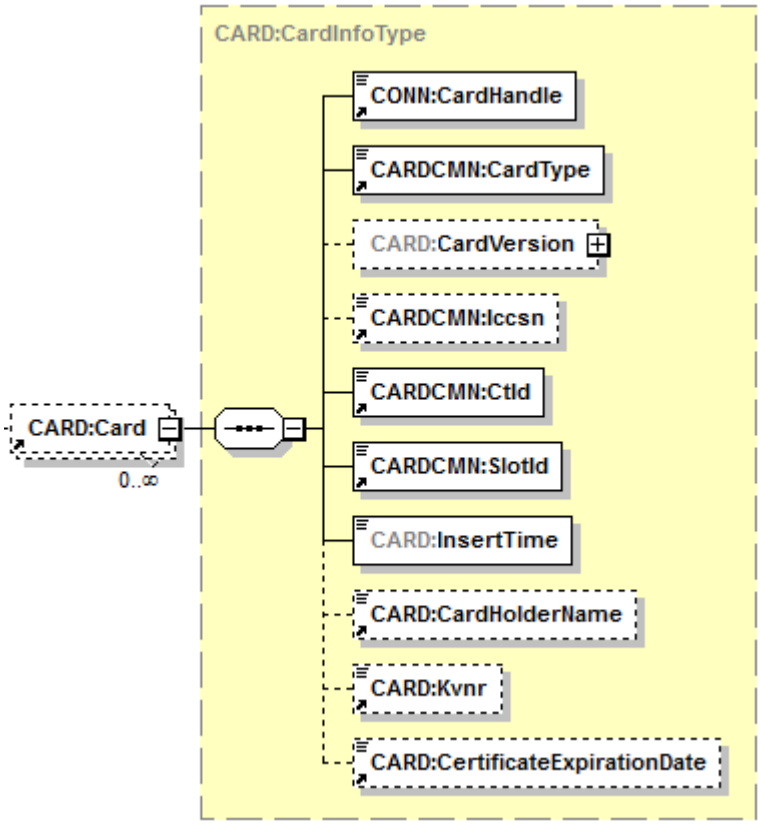
4.1.6.5.2 GetCards

TIP1-A_4605 - Operation GetCards
Der Konnektors MUSS an der Außenschnittstelle eine Operation GetCards, wie in Tabelle TAB_KON_565 „Operation GetCards" beschrieben, anbieten und MUSS dabei Kartentypen aus Tabelle TAB_KON_500 Wertetabelle Kartentypen unterscheiden.

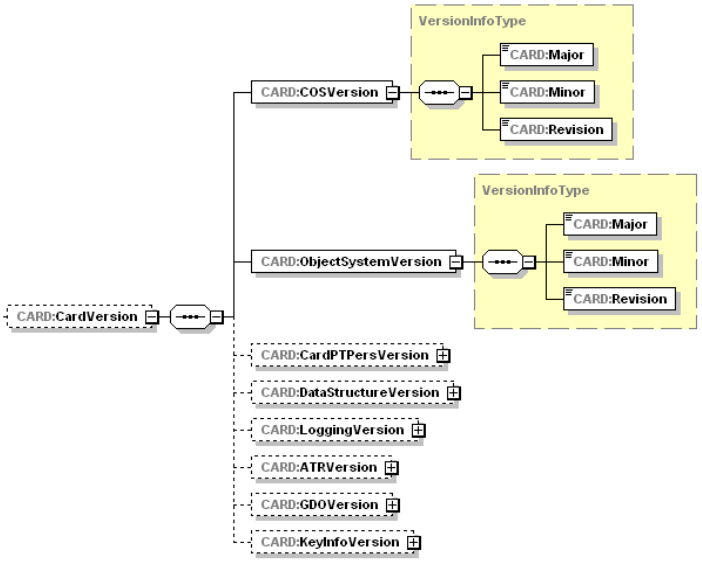
Tabelle 146: TAB_KON_565 Operation GetCards

Name	`GetCards`
Beschreibung	Liefert Informationen zu den in den Kartenterminals verfügbaren Karten zurück, die in Kartenterminals stecken, auf die Mandant und Clientsystem zugreifen dürfen. Insbesondere umfasst die Information die sog. Karten-Handles. Die Karten-Handles können bei anderen Konnektoraufrufen zur Adressierung von Karten genutzt werden.
Aufrufparameter	

	Name	Beschreibung
	@mandant-wide	Wenn „true", werden alle Karten zurückgegeben, auf die der Mandant und das aufrufende Clientsystem zugreifen dürfen. Wenn „false" (Standardbelegung), werden nur Karten zurückgegeben, auf die von dem im Aufrufkontext spezifizierten Arbeitsplatz zugegriffen werden darf.
	Context	Aufrufkontext
	CtId	Identifikation des Kartenterminals. Wenn angegeben, werden nur die Karten zurückgeliefert, die in diesem Kartenterminal verfügbar sind.
	SlotId	Nummer des Slots, beginnend bei 1. Wird zusätzlich zur CtId auch SlotId übergeben, so wird die Karte zurückgegeben, die in dem angegebenen Slot des mit CtId adressierten Kartenterminals steckt.
	CardType	Ein Kartentyp gemäß Tabelle TAB_KON_500 „Wertetabelle Kartentypen" als optionaler Filter. Wenn angegeben, werden nur Karten vom spezifizierten Typ zurückgegeben. Unterstützt werden die Kartentypen EGK, KVK, HBAx, SM-B, SMC-KT und UNKNOWN.
Antwort		
	Name	**Beschreibung**
	Status	Ergebnis der Operation
	Im Element Cards wird die Liste der gesteckten Karten zurückgegeben. Für jede Karte wird dabei ein Card-Element angegeben. Leere Slots der Kartenterminals sind in dieser Liste nicht enthalten.	



Name	Beschreibung
`Card Handle`	Handle, mit dem die Karte in Folgeaufrufen adressiert werden kann. Der Konnektor garantiert, dass dieses Handle die gesteckte Karte eindeutig identifiziert und bei Entfernen der Karte aus dem Kartenterminal ungültig wird. Auch für nicht erkannte Karten (z. B. bei falscher Steckrichtung der Karte) SOLL der Konnektor gültige Handles liefern (sofern das Kartenterminal in diesem Fall in der Lage ist, das entsprechende Ereignis „Karte wurde gesteckt" zu liefern), damit diese Karten z. B. zum Auswurf adressiert werden können.
`CardType`	Erkannter Typ der Karte. Siehe Tabelle TAB_KON_500 Wertetabelle Kartentypen,

	Card Version	 Der Konnektor MUSS in CardVersion bei eGK, HBA und SM-B/SMC-KT der Generation 2 die Versionsinformationen gemäß [gemSpec_Karten_Fach_TIP] übergeben, für G1+ aus /MF/EF.Version. Bei KVK, HBA-VK und unbekannten Karten MUSS das Element weggelassen werden.
	Iccsn	Falls auslesbar, die ICC-Serial-Number der Karte. Im Fall der KVK wird das optionale Element Iccsn nicht zurückgegeben.
	CtId	Identifikation des Kartenterminals, in dem die Karte steckt.
	SlotId	Nummer des Slots (beginnend bei 1), in dem die Karte steckt.
	InsertTime	Gibt den Zeitpunkt an, zu dem der Konnektor die Karte erkannt hat. Die Zeit wird mit dem Datentyp `DateTime` in folgendem Format angegeben: `yyyy-mm-ddThh:mm:ss+hh:mm` Es ist also – gemäß ISO 8601 [ISO8601] – die lokale Zeit und die Differenz zur UTC anzugeben.
	CardHolderName	Name des Karteninhabers bzw. der Institution/Organisation (subject.commonName). Bei KVK und unbekannten Karten MUSS das Element weggelassen werden.

	Kvnr	KVNR (Unveränderbarer Teil) MUSS bei eGK belegt werden. Bei allen anderen Karten MUSS das Element weggelassen werden.
	Certificate Expiration Date	Ablaufdatum des Zertifikates (AUT bzw. OSIG). Bei KVK und unbekannten Karten MUSS das Element weggelassen werden.
Vorbedingungen	Keine.	
Nachbedingungen	Der Zustand der Karten und der Kartenterminals bleibt unverändert.	
Hinweise	Der Aufruf darf nur den im Konnektor verwalteten aktuellen Zustand der Karte liefern und keine Abfragen an die Kartenterminals absetzen.	

Der Ablauf der Operation GetCards ist in Tabelle TAB_KON_566 Ablauf GetCards beschrieben:

Tabelle 147: TAB_KON_566 Ablauf GetCards

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffsberechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession = false;
		allWorkplaces = @mandant-wide}
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_253 „Liefere Karten_Liste“	Die Liste der Karten wird erstellt und zurückgegeben. Tritt hierbei ein Fehler auf, so bricht die Operation mit dem Fehler des TUCs ab.
		Wenn `@mandant-wide=true` dann ermittle die Liste der Karten für alle Arbeitsplätze des Mandanten für das angegebene Clientsystem durch den Aufruf
		TUC_KON_253 „Liefere Karten_Liste“ {
		clientSystemId = $context.clientsystemId;
		cardTerminalId = CtId;
		slotId = SlotId;

		mandantId = $context.mandantId;
		cardType = CardType }
		Wenn `@mandant-wide=false` dann ermittle die Liste der Karten für den Arbeitsplatz des Mandanten für das angegebene Clientsystem entsprechend $context durch den Aufruf
		TUC_KON_253 „Liefere Karten_Liste“ {
		workplaceId= $context.workplaceId;
		clientSystemId = $context.clientsystemId;
		cardTerminalId = CtId;
		slotId = SlotId;
		mandantId = $context.mandantId;
		cardType = CardType }

Die Fehlerfälle der Operation GetCards sind in Tabelle TAB_KON_567 Fehlercodes „GetCards dargestellt:

Tabelle 148: TAB_KON_567 Fehlercodes „GetCards“

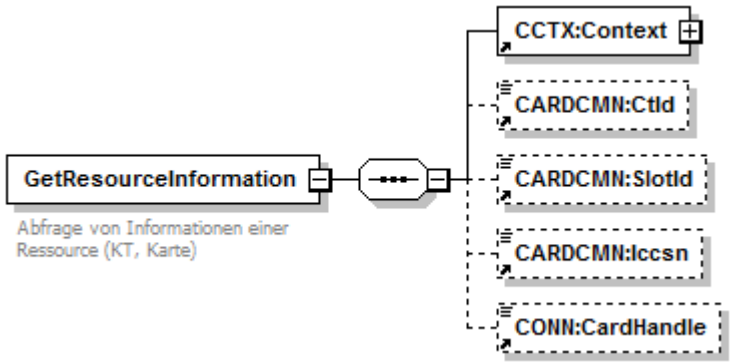
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

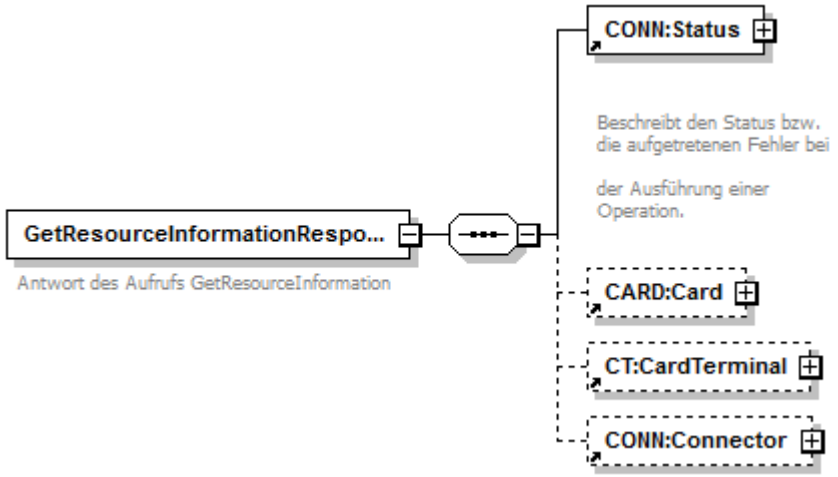
[<=]

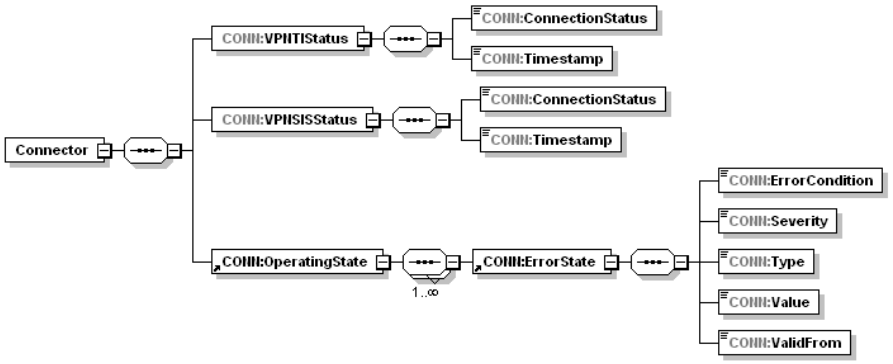
4.1.6.5.3 GetResourceInformation

TIP1-A_4607 - Operation GetResourceInformation
Der Konnektors MUSS an der Außenschnittstelle eine Operation GetResourceInformation, wie in Tabelle TAB_KON_568 „Operation GetResourceInformation“ beschrieben, anbieten.

Tabelle 149: TAB_KON_568 Operation GetResourceInformation

Name	`GetResourceInformation`	
Beschreibung	Gibt Informationen zu einer Ressource (Karte, KT) oder dem Konnektor selbst zurück	
Aufrufparameter		
	Name	**Beschreibung**

	`Context`	Aufrufkontext
	`CtId`	Identifikator eines Kartenterminals
	`SlotId`	Kartenslot-Nummer (in Kombination mit `CtId`)
	`Iccsn`	Iccsn einer Karte
	`CardHandle`	CardHandle einer Karte. Unterstützt werden die Kartentypen EGK, KVK, HBAx, SM-B, SMC-KT und UNKNOWN.
	Wurde keines der Elemente `CtId`, `SlotId`, `Iccsn` übergeben, so wird davon ausgegangen, dass der Aufrufer Informationen zum Konnektor selbst abfragen möchte.	
Rückgabe		
	Name	**Beschreibung**
	`Status`	Ergebnis der Operation
	`Card`	Informationen einer Karte (siehe `GetCards`)
	`CardTerminal`	Informationen eines Kartenterminals (siehe `GetCardTerminals`)
	`Connector`	Informationen zum Konnektor



VPNTIStatus	
VPNTIStatus/ ConnectionStatus	Status der VPN-Verbindung zur TI (Online oder Offline)
VPNTIStatus/ Timestamp	Zeitstempel der letzten Statusänderung
VPNSISSTatus	
VPNSISStatus/ ConnectionStatus	Status der VPN-Verbindung des SIS (Online oder Offline)
VPNSISStatus/ Timestamp	Zeitstempel der letzten Statusänderung
OperatingState	Betriebszustand (siehe Kapitel 3.3)
OperatingState/ ErrorState	Eine Zeile der Fehlerzustandsliste gemäß Tabelle TAB_KON_503 Betriebszustand_Fehlerzustandsliste
OperatingState/ ErrorState/ ErrorCondition	ErrorCondition gemäß Tabelle TAB_KON_503 Betriebszustand_Fehlerzustandsliste
OperatingState/ ErrorState/Severity	Schwere des Fehlerzustandes gemäß Tabelle TAB_KON_503 Betriebszustand_Fehlerzustandsliste
OperatingState/ ErrorState/Type	Fehlertyp gemäß Tabelle TAB_KON_503 Betriebszustand_Fehlerzustandsliste
OperatingState/ ErrorState/Value	Fehlerzustandswert
OperatingState/ ErrorState/ValidFrom	Zeitstempel der letzten Änderung des Fehlerzustands

Vorbedingung	
Nachbedingung	Der Zustand der Ressource bleibt unverändert.

Hinweise	

Der Ablauf der Operation GetResourceInformation ist in Tabelle TAB_KON_569 Ablauf GetResourceInformation beschrieben:

Tabelle 150: TAB_KON_569 Ablauf GetResourceInformation

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Insbesondere wird geprüft, dass eine `SlotId` nur in Verbindung mit einer `CtId` übergeben werden kann (Abfrage einer Karte). Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt,
		falls die Ressource der Konnektor ist, durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		ctId = null;
		cardHandle = null;
		needCardSession = false;
		allWorkplaces = true
		}
		falls die Ressource ein Kartenterminal ist, durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		ctId = $ctId;
		cardHandle = null;
		needCardSession = false;
		allWorkplaces = true
		}
		falls die Ressource eine Karte ist, durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		ctId = null;
		cardHandle = $cardHandle;
		needCardSession = false;
		allWorkplaces = false
		}

		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
3.	TUC_KON_254 „Liefere Ressourcendetails“	Die Informationen zu der Ressource werden zusammengetragen und zurückgegeben. Tritt hierbei ein Fehler auf, so bricht die Operation mit dem Fehler des TUCs ab.

Die Fehlerfälle der Operation GetResourceInformation sind in Tabelle TAB_KON_570 Fehlercodes „GetResourceInformation dargestellt:

Tabelle 151: TAB_KON_570 Fehlercodes „GetResourceInformation“

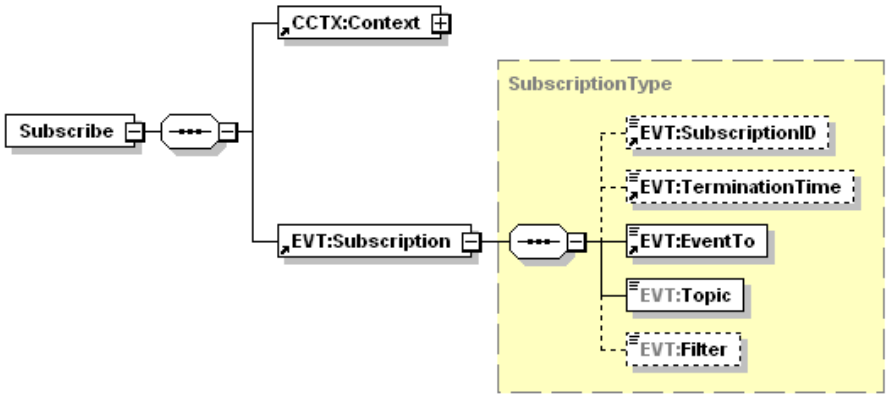
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

[<=]

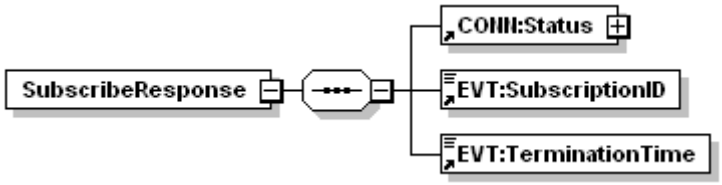
4.1.6.5.4 Subscribe

TIP1-A_4608 - Operation Subscribe
Der Konnektors MUSS an der Außenschnittstelle eine Operation Subscribe, wie in Tabelle TAB_KON_571 Operation Subscribe beschrieben, anbieten.

Tabelle 152: TAB_KON_571 Operation Subscribe

<table>
<tr><td>Name</td><td>Subscribe</td></tr>
<tr><td>Beschreibung</td><td>Clientsysteme melden mit dieser Operation ihr Interesse an bestimmten Topics (Ereignissen) an.</td></tr>
<tr><td>Aufrufparameter</td><td> 

<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td>Context</td><td>Aufrufkontext</td></tr>
</table></td></tr>
</table>

	SubscriptionID	Darf nicht verwendet werden
	TerminationTime	Darf nicht verwendet werden
	EventTo	URL des Endpunkts, wo die Ereignisse zugestellt werden sollen. Syntax: cetp://*host*:*port* *host*: IP-Adresse oder FQDN des Clientsystems. *port*: Portnummer des Kommunikationsendpunkts, an dem das Clientsystem auf die Zustellung der Ereignisse wartet.
	Topic	Ein Topic, für das das Clientsystem sein Interesse anmeldet.
	Filter	Filter für die Ereignisnachricht (X-Path Ausdruck im Kontext mit Default Namespace gleich "http://ws.gematik.de/conn/EventService/v7.2" ohne Verwendung eines Namespace-Präfixes sowie Namensraum mit Präfix EVT gleich "http://ws.gematik.de/conn/EventService/v7.2", der beim Versand von Ereignissen in TUC_KON_256 ausgewertet wird. Ermöglicht die Filterung auf Basis der Elemente einer XML-Ereignisnachricht)
Rückgabe		
	Name	**Beschreibung**
	Status	Ergebnis der Operation
	SubscriptionID	Ein Identifikator, der die Anmeldung für die Topics eindeutig identifiziert. Bei den Operationen Unsubscribe, GetSubscription und RenewSubsriptions MUSS dieser SubscriptionID angegeben werden.
	TerminationTime	Maximaler Gültigkeitszeitpunkt der Subscription. Sie MUSS auf Systemzeit + 25 h gesetzt werden.
Vorbedingung	Das Clientsystem muss die Ereignissenke realisieren.	
Nachbedingung	Nach erfolgreicher Anmeldung vermerkt der Konnektor das angemeldete Topic unter dem SubscriptionID. Der Konnektor muss die Anmeldungen so lange als gültig behandeln, bis entweder das Clientsystem diese explizit abmeldet	

	oder der Konnektor das Clientsystem als nicht mehr erreichbar erkennt (siehe nächsten Punkt) oder der Konnektor neu gestartet oder ausgeschaltet wird oder die TerminationTime kleiner als die Systemzeit ist. Der Konnektor muss die Anmeldung aus seiner Verwaltung entfernen („Auto-Unsubscribe"), wenn EVT_MAX_TRY Verbindungsaufbauversuche oder zählbare Zustellungsversuche (z.B. durch Zählung beim Absenden der Zustellversuche) in Folge fehlgeschlagen sind. Wenn die Ereignissenke Zustellungen grundsätzlich nicht beantwortet, so sind nur die Verbindungsaufbauversuche zu zählen.
Hinweise	

Der Ablauf der Operation Subscribe ist in Tabelle TAB_KON_572 Ablauf Subscribe beschrieben:

Tabelle 153: TAB_KON_572 Ablauf Subscribe

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung"	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 _000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession = false;
		allWorkplaces = true }
		Tritt bei der Prüfung ein Fehler auf, so bricht die Operation mit dem Fehlercode aus TUC_KON_000 ab.

3.	saveSubscription	Die Anmeldung wird im Konnektor hinterlegt. Vorgehalten werden folgende Daten: • SubscriptionId (wird generiert) • TerminationTime (Systemzeit + 25h) • MandantId • ClientsystemId • WorkplaceId • Ereignissenke (Feld `EventTo`) • Abonnierter Topic (Feld `Topic`) • Filterausdruck (Feld `Filter`) Bei der Übernahme wird eine eindeutige `SubscriptionId` generiert, die dem aufrufenden System

		zurückgegeben wird, falls die Subscription noch nicht existiert. Existiert sie bereits, wird die bestehende `SubscriptionID` zurückgegeben.

Die Fehlerfälle der Operation Subscribe sind in Tabelle TAB_KON_573 Fehlercodes „Subscribe dargestellt:

Tabelle 154 TAB_KON_573 Fehlercodes „Subscribe"

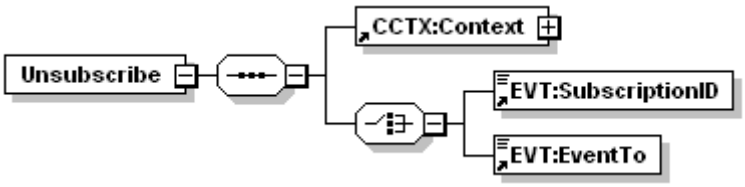
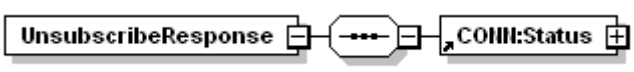
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

[<=]

4.1.6.5.5 Unsubscribe

TIP1-A_4609 - Operation Unsubscribe
Der Konnektors MUSS an der Außenschnittstelle eine Operation Unsubscribe, wie in Tabelle TAB_KON_574 Operation Unsubscribe beschrieben, anbieten.

Tabelle 155: TAB_KON_574 Operation Unsubscribe

<table>
<tr><td>Name</td><td colspan="2"><code>Unsubscribe</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Löscht eine Anmeldung mit dem angegebenen <code>SubscriptionID</code> oder alle Anmeldungen zu einem Endpunkt <code>EventTo</code>.</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td><code>Context</code></td><td>Aufrufkontext</td></tr>
<tr><td></td><td><code>SubscriptionID</code></td><td>Der Identifikator, der bei der <code>Subscribe</code>-Operation geliefert wurde.</td></tr>
<tr><td></td><td><code>EventTo</code></td><td>URL des clientseitigen Endpunkts, wie er bei der <code>Subscribe</code>-Operation angegeben wurde.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td><code>Status</code></td><td>Ergebnis der Operation</td></tr>
</table>

Vorbedingung	Die Anmeldung `Subscribe` muss vor dieser Operation aufgerufen worden sein.
Nachbedingung	Der Konnektor entfernt aus seiner Verwaltung die Subscription zur `Subscription-ID` bzw. alle Subscriptions zur clientseitigen URL des Endpunkts `EventTo`.
Hinweise	Keine

Der Ablauf der Operation Unsubscribe ist in Tabelle TAB_KON_575 Ablauf Unsubscribe beschrieben:

Tabelle 156: TAB_KON_575 Ablauf Unsubscribe

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession = false;
		allWorkplaces = true }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	removeSubscription	Entfernen der Subscriptions aus der Liste aller Subscriptions.

Die Fehlerfälle der Operation Unsubscribe sind in Tabelle TAB_KON_576 Fehlercodes „Unsubscribe dargestellt:

Tabelle 157: TAB_KON_576 Fehlercodes „Unsubscribe“

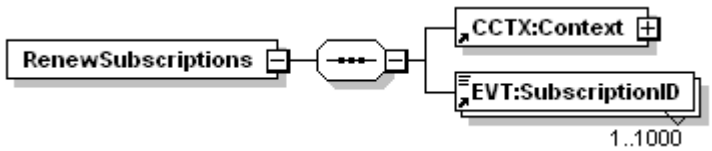
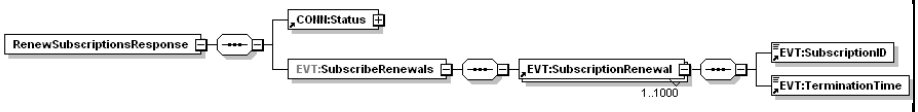
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4102	Technical	Error	ungültige SubscriptionId

[<=]

4.1.6.5.6 RenewSubscriptions

TIP1-A_5112 - Operation RenewSubscriptions
Der Konnektors MUSS an der Außenschnittstelle eine Operation RenewSubscriptions, wie in Tabelle TAB_KON_792 Operation RenewSubscriptions beschrieben, anbieten.

Tabelle 158: TAB_KON_792 Operation RenewSubscriptions

<table>
<tr><td>Name</td><td colspan="2">RenewSubscriptions</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Verlängert die Gültigkeit einer Liste von Anmeldungen, die jeweils per SubscriptionID identifiziert werden.</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>Context</td><td>Aufrufkontext</td></tr>
<tr><td></td><td>Subscription ID</td><td>Der Identifikator, der bei der Subscribe-Operation geliefert wurde.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>Status</td><td>Ergebnis der Operation</td></tr>
<tr><td></td><td>Subscription ID</td><td>Ein Identifikator, der die Anmeldung für die Topics eindeutig identifiziert. Bei den Operationen Unsubscribe, GetSubscription und RenewSubsriptions MUSS diese SubscriptionID angegeben werden.</td></tr>
<tr><td></td><td>Termination Time</td><td>Maximaler Gültigkeitszeitpunkt der Subscription. Sie MUSS auf Systemzeit + 25 h gesetzt werden.</td></tr>
<tr><td>Vorbedingung</td><td colspan="2"></td></tr>
<tr><td>Nachbedingung</td><td colspan="2">Der Konnektor speichert jede neu vergebene TerminationTime in seiner Verwaltung der Subscriptions.</td></tr>
<tr><td>Hinweise</td><td colspan="2">Keine</td></tr>
</table>

Der Ablauf der Operation RenewSubscriptions ist in Tabelle TAB_KON_793 Ablauf RenewSubscriptions beschrieben:

Tabelle 159: TAB_KON_793 Ablauf RenewSubscriptions

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession = false;
		allWorkplaces = true }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	renewSubscriptions	Es wird eine neue `SubscribeRenewals`-Liste angelegt. Alle Subscriptions, deren `TerminationTime` kleiner als die Systemzeit sind, muss der Konnektor aus der Verwaltung entfernen. Für jede `SubscriptionID`, die in der Verwaltung der Subscriptions existiert und deren `TerminationTime` größer als die Systemzeit ist, wird eine neue `TerminationTime = Systemzeit + 25h` bestimmt. Diese wird zusammen mit der `SubscriptionID` als `SubscribeRenewal` der `SubscribeRenewals`-Liste hinzugefügt. Kommt es zu keiner Subscription-Verlängerung, weil nur ungültige SubscriptionIDs im Aufruf angegeben waren, wird der Fehler 4102 zurückgeliefert. Kommt es zu mindestens einer Subscription-Verlängerung, sind aber auch ungültige SubscriptionIDs im Aufruf, wird eine RenewSubscriptionsResponse zurückgeliefert, mit CONN:Status/CONN:Result = "Warning", GERROR:Trace mit {Fehlercode: 4102, ErrorType: Technical, Severity: Error, Fehlertext: "Ungültige SubscriptionId"}, und der Information, welche SubscriptionsIDs ungültig waren.

Die Fehlerfälle der Operation RenewSubscriptions sind in Tabelle TAB_KON_794 Fehlercodes „RenewSubscriptions dargestellt:

Tabelle 160: TAB_KON_794 Fehlercodes „RenewSubscriptions“

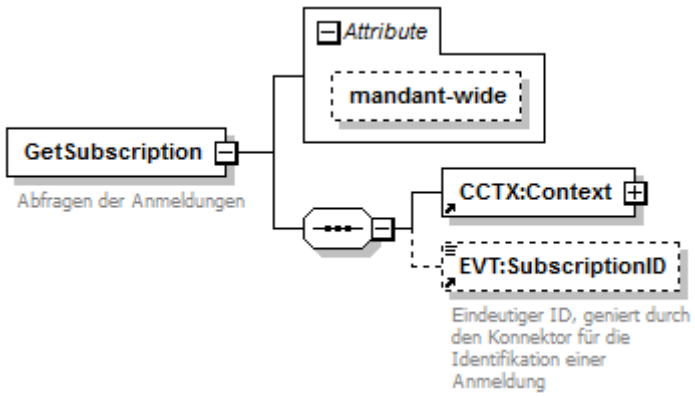
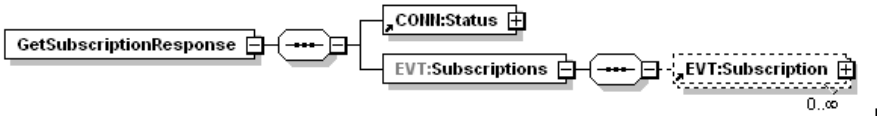
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4102	Technical	Error	Ungültige SubscriptionId

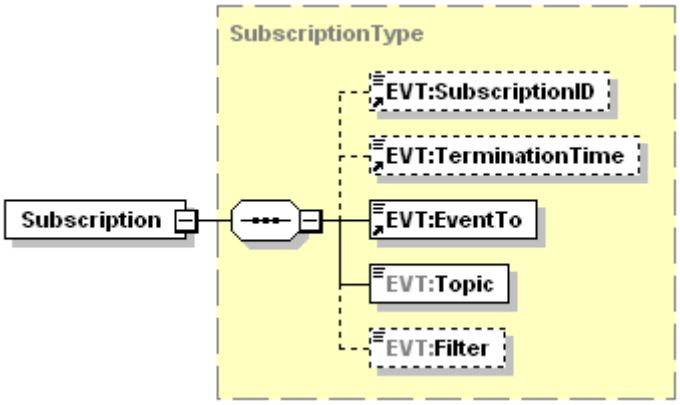
[<=]

4.1.6.5.7 GetSubscription

TIP1-A_4610 - Operation GetSubscription
Der Konnektor MUSS an der Außenschnittstelle eine Operation GetSubscription, wie in Tabelle TAB_KON_577 Operation GetSubscription beschrieben, anbieten.

Tabelle 161: TAB_KON_577 Operation GetSubscription

<table>
<tr><td>Name</td><td colspan="2">GetSubscription</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Gibt die Liste der angemeldeten Topics zurück</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>@mandant-wide</td><td>Wenn „true“, werden alle Subscriptions zurückgegeben, die Mandant und Clientsystem zugeordnet sind.
Wenn „false“ (Standardbelegung) werden alle Subscriptions zurückgegeben, die dem im Aufrufkontext spezifizierten Tripel aus Clientsystem, Mandanten und Arbeitsplatz zugeordnet sind.</td></tr>
<tr><td></td><td>Context</td><td>Aufrufkontext</td></tr>
<tr><td></td><td>SubscriptionID</td><td>Der Identifikator, der bei der Subscribe-Operation geliefert wurde.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>Status</td><td>Ergebnis der Operation</td></tr>
<tr><td></td><td>Subscriptions</td><td>Die Liste Subscriptions (vgl. Operation Subscribe)</td></tr>
</table>

	
	`Subscription` — Angefordertes Subscription-Element
	`Subscription/SubscriptionID` — Identifikator der Subscription
	`Subscription/TerminationTime` — Maximaler Gültigkeitszeitpunkt der Subscription.
	`Subscription/EventTo` — URL des Endpunkts, wo die Ereignisse zugestellt werden sollen (Ereignissenke)
	`Subscription/Topic` — Angemeldeter Topic
	`Subscription/Filter` — Filterausdruck (falls vorhanden)
Vorbedingung	Keine
Nachbedingung	Die Liste der Subscriptions bliebt unverändert
Hinweise	Keine

Der Ablauf der Operation GetSubscription ist in Tabelle TAB_KON_578 Ablauf GetSubscription beschrieben:

Tabelle 162: TAB_KON_578 Ablauf GetSubscription

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession = false;
		allWorkplaces = @mandant-wide }

		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
3.	getSubscriptions	Rückgabe der Subscription, die durch `SubscriptionId` identifiziert wird. Wurde keine `SubscriptionId` angegeben und @mandant-wide="true", werden alle Subscriptions zurückgegeben, die dem angegebenen Clientsystem und Mandanten zugeordnet werden können. Wurde keine SubscriptionId angegeben und @mandant-wide="false", werden alle Subscriptions zurückgegeben, die dem angegebenen Clientsystem, Mandanten und Arbeitsplatz zugeordnet werden können.

Die Fehlerfälle der Operation GetSubscription sind in Tabelle TAB_KON_579 Fehlercodes „GetSubscription dargestellt:

Tabelle 163: TAB_KON_579 Fehlercodes „GetSubscription"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4102	Technical	Error	ungültige SubscriptionId

[<=]

4.1.6.6 Betriebsaspekte

TIP1-A_4611 - Konfigurationswerte des Systeminformationsdienstes
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_580 vorzunehmen:

Tabelle 164: TAB_KON_580 Konfigurationswerte des Systeminformationsdienstes (Administrator)

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
`EVT_MAX_TRY`	`Nummer`	Der Administrator MUSS über diesen Konfigurationsparameter die Anzahl der Fehlversuche bzgl. Verbindungsversuche bzw. Ereigniszustellungen festlegen können. Ist diese maximal zulässige Anzahl der Fehlversuche überschritten, muss der Konnektor automatisch ein „Auto-Unsubscribe" (analog Operation „Unsubscribe" mit „EventTo gleich der URL des clientseitigen Endpunkts") durchführen.

[<=]

TIP1-A_4612 - Maximale Anzahl an Subscriptions
Der Konnektor MUSS eine Mindestmenge von 999 Subscriptions insgesamt unterstützen. Der Konnektorhersteller kann jedoch die Anzahl der maximal möglichen Subscriptions (insgesamt und/oder pro Ziel) festlegen.
[<=]

TIP1-A_4613 - Initialisierung Subscriptions-Liste beim Bootup
Der Konnektor MUSS beim Bootup mit einer leeren Liste an Subscriptions starten.
[<=]

4.1.7 Verschlüsselungsdienst

Der Verschlüsselungsdienst bietet Schnittstellen zum hybriden und symmetrischen Ver- und Entschlüsseln von Dokumenten an.

Der Verschlüsselungsdienst bietet für alle `Alle_DocFormate` die hybride und symmetrische Ver- und Entschlüsselung nach dem Cryptographic Message Syntax (CMS) Standard an [RFC5652].

Darüber hinaus werden folgende formaterhaltende Ver-/Entschlüsselungsmechanismen unterstützt:

- hybride Ver-/Entschlüsselung von XML-Dokumenten nach der W3C Recommendation „XML Encryption Syntax and Processing" [XMLEnc]
- hybride Ver-/Entschlüsselung von MIME-Dokumenten nach dem S/MIME-Standard [S/MIME]

Der Konnektor muss bezüglich der zur Ver- und Entschlüsselung von Dokumenten verwendeten Verfahren und Algorithmen die Vorgaben in [gemSpec_Krypt#3.1.4] sowie in [gemSpec_Krypt#3.1.5] und hinsichtlich ECC-Migration die Vorgaben aus [gemSpec_Krypt#5] erfüllen.

4.1.7.1 Funktionsmerkmalweite Aspekte

TIP1-A_4614 - Missbrauchserkennung Verschlüsselungsdienst
Der Konnektors MUSS zur Unterstützung von Missbrauchserkennungen die in Tabelle TAB_KON_581 gelisteten Operationen als Einträge in EVT_MONITOR_OPERATIONS berücksichtigen.

Tabelle 165: TAB_KON_581 Verschlüsselungsdienst-Operationen für EVT_MONITOR_OPERATIONS

Operationsname	OK_Val	NOK_Val	Alarmwert (Default-Grenzwert 10 Minuten-Σ)
`EncryptDocument`	1	5	101
`DecryptDocument`	1	5	101

[<=]

TIP1-A_5434 - Verschlüsselung/Entschlüsselung eines XML Dokuments ergibt unverändertes XML-Dokument
Der Konnektor MUSS das Operationspaar Verschlüsselung/Entschlüsselung so implementieren, dass Dokumente vom Typ XML unverändert bleiben, wobei zwei XML-Dokumente als identisch zu betrachten sind, wenn sie gemäß Canonical XML 1.1 gleich sind [CanonXML1.1].**[<=]**

A_17746 - Einsatzbereich und Vorgaben für Ver- und Entschlüsselung (ECC-Migration)
Der Konnektor MUSS für die kartenbasierte Ver- und Entschlüsselung die Zertifikate und Schlüssel in Abhängigkeit vom kryptographischen Verfahren unter Berücksichtigung des Einsatzbereiches aus TAB_KON_747 ermitteln.**[<=]**

Tabelle 166: TAB_KON_747 KeyReference für Encrypt-/DecryptDocument

Karte	KeyReference	Crypt	Zertifikat (Encrypt) ...in DF.ESIGN	Schlüssel (Decrypt) ...in DF.ESIGN	Einsatzbereich	
					Außen-schnittstelle	Fachmodul-schnittstelle
HBA	C.ENC	RSA_ECC	EF.C.HP.ENC.R2048 EF.C.HP.ENC.E256	PrK.HP.ENC.R2048 PrK.HP.ENC.E256	Ja	Ja
		ECC	EF.C.HP.ENC.E256	PrK.HP.ENC.E256	Ja	Ja
		RSA	EF.C.HP.ENC.R2048	PrK.HP.ENC.R2048	Ja	Ja
SM-B	C.ENC	RSA_ECC	EF.C.HCI.ENC.R2048 EF.C.HCI.ENC.E256	PrK.HCI.ENC.R2048 PrK.HP.ENC.E256	Ja	Ja
		ECC	EF.C.HCI.ENC.E256	PrK.HP.ENC.E256	Ja	Ja
		RSA	EF.C.HCI.ENC.R2048	PrK.HCI.ENC.R2048	Ja	Ja
HBA-VK	C.ENC	RSA_ECC RSA	EF.C.HP.ENC	PrK.HP.ENC	Ja	Ja

eGK	C.ENC	ECC	C.CH.ENC.E256	PrK.CH.ENC.E256	Nein	Ja
	C.ENC	RSA	C.CH.ENC.R2048	PrK.CH.ENC.R2048	Nein	Ja

Tabelle 167: TAB_KON_859 Werteliste und Defaultwert des Parameters crypt bei hybrider Verschlüsselung

Typname	Werteliste	Defaultwert	Bedeutung
ENC_CRYPT	RSA ECC RSA_ECC	RSA	Werteliste des Parameters crypt bei der hybriden Verschlüsselung RSA: Es wird RSA-2048 basiert verschlüsselt. ECC: Es wird ECC-256 basiert verschlüsselt. RSA_ECC: Es wird dual RSA-2048 basiert und ECC-256 basiert verschlüsselt. Es wird als Fehlerfall gewertet, wenn weder RSA- noch ECC-Zertifikat von der Karte geladen werden konnten, und als Warnung, wenn nur ein Zertifikat geladen werden konnte.

4.1.7.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.1.7.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.1.7.4 Interne TUCs, auch durch Fachmodule nutzbar

Die in diesem Kapitel beschriebenen TUCs zur hybriden Ver- und Entschlüsselung werden den Fachmodulen und Außenoperationen angeboten. Die TUCs zur symmetrischen Ver-/Entschlüsselung werden den Fachmodulen angeboten. Es gibt keine Aufrufhierarchie innerhalb der hier beschriebenen TUCs zur hybriden und symmetrischen Ver-/Entschlüsselung.

4.1.7.4.1 TUC_KON_070 „Daten hybrid verschlüsseln"

TIP1-A_4616-02 - TUC_KON_070 „Daten hybrid verschlüsseln"
Der Konnektor MUSS den technischen Use Case TUC_KON_070 „Daten hybrid verschlüsseln" umsetzen.

Tabelle 168: TAB_KON_739 - TUC_KON_070 „Daten hybrid verschlüsseln"

Element	Beschreibung

Name	TUC_KON_070 „Daten hybrid verschlüsseln“
Beschreibung	Dieser TUC verschlüsselt ein Dokument oder Teile eines Dokumentes. Die Verschlüsselung erfolgt zweistufig, d. h. die Daten werden symmetrisch mit einem generierten Schlüssel verschlüsselt und anschließend wird dieser Schlüssel mit einem asymmetrischen Verfahren verschlüsselt. Die asymmetrische Verschlüsselung des symmetrischen Schlüssels kann für mehrere Identitäten, repräsentiert durch X.509-Zertifikate oder öffentliche Schlüssel, erfolgen. Das Ergebnis sind entsprechend viele Verschlüsselungen desselben symmetrischen Schlüssels. Es werden die folgenden formaterhaltenden Verschlüsselungsverfahren für die genannten Dokumententypen unterstützt: • XML mit [XMLEnc] • MIME mit [S/MIME] Des Weiteren ist für alle unterstützten Dokumentformate (`Alle_DocFormate`) die Verschlüsselung mit CMS [RFC5652] möglich.
Auslöser	Aufruf durch einen Fachmodul-TUC oder durch die Operation EncryptDocument des Verschlüsselungsbasisdienstes
Vorbedingungen	Falls mit einem öffentlichen Schlüssel auf einer Karte verschlüsselt werden soll, muss die Karte gesteckt sein.
Eingangsdaten	• documentToBeEncrypted (Zu verschlüsselndes Dokument) • encryptionCertificates – *optional/entfällt, wenn encryptionKeys übergeben wird* (X.509v3-Zertifikate) • encryptionKeys – *optional/entfällt, wenn* encryptionCertificates *übergeben wird* (öffentliche Schlüssel; unterstützte Karten sind SM-B, HBAx und eGK) • encryptionType [EncryptionType] (Angaben zum einzusetzenden Verschlüsselungsverfahren (CMS, XMLEnc oder S/MIME)). • cardSession – *optional/verpflichtend, wenn ein Zertifikat von einer Karte gelesen werden soll* (Kartensitzung; unterstützte Karten sind SM-B, HBAx und eGK.) • certificateReference – *optional/verpflichtend, wenn ein Zertifikat von einer Karte gelesen werden soll* (Zertifikatsreferenz; unterstützte Karten sind SM-B, HBAx und eGK).

	• crypt [ENC_CRYPT] - *optional;* *default und Wertebereich siehe TAB_KON_859* *(Wenn das Verschlüsselungszertifikat von einer Karte kommt, steuert* `crypt`*, mit welchen kryptographischen Verfahren die Verschlüsselung der Hybridschlüssel erfolgt.)* • xmlElements – *optional/verpflichtend, wenn encryptionType = XMLEnc* (Festlegung der zu verschlüsselnden Teile des Dokumentes durch Spezifikation eines Xpath-Ausdruckes (XML-Elements). • keyInfoMode [embedded | separate] – *optional/verpflichtend, wenn encryptionType = XMLEnc* (Angabe, ob die KeyInfo in das XML-Dokument eingebettet oder separat an den Aufrufer zurückgegeben werden soll)
Komponenten	Konnektor, Kartenterminal, Karte
Ausgangsdaten	• encryptedDocument (Verschlüsseltes Dokument) • encryptedKeys – *optional/verpflichtend, wenn diese nicht im verschlüsselten Dokument enthalten sind* (Verschlüsselte symmetrische Schlüssel) • keyInfo – *optional/verpflichtend, wenn encryptionType = XMLEnc und keyInfoMode = separate* (KeyInfo, falls nicht ins Dokument eingebettet)
Standardablauf	1. Das Verschlüsselungsverfahren wird anhand des Eingangsparameters EncryptionType gewählt. 2. Nur für XMLEnc: Die zu verschlüsselnden XML-Elemente werden lokalisiert. Falls kein zu verschlüsselndes XML-Element gefunden wurde, wird Fehler 4103 gemeldet. Die zu verschlüsselnden XML-Elemente dürfen nicht ineinander verschachtelt sein. Sind die zu verschlüsselnden XML-Elemente ineinander verschachtelt, so wird Fehler 4104 gemeldet. 3. Für jedes von der Karte zu lesende Zertifikat, wird TUC_KON_216 „Lese Zertifikat" aufgerufen. Welches Zertifikat von der Karte gelesen werden soll, wird durch den Parameter `crypt` über Tabelle TAB_KON_747 gesteuert. In den Fällen crypt = RSA und crypt = ECC bricht der TUC ab, wenn dabei ein Fehler auftritt. Im Fall crypt = RSA_ECC bricht der TUC im Fehlerfall dann ab, wenn weder RSA- noch ECC-Zertifikat geladen werden konnte, und läuft mit einer Warnung durch, wenn nur ein Zertifikat geladen werden konnte.

4. Falls Zertifikate übergeben oder von der Karte gelesen wurden, werden diese durch Aufruf von TUC_KON_037 „Zertifikat prüfen“ geprüft.
   Als Parameter des TUC-Aufrufs gilt für Zertifikate, die mit Zertifikaten aus CERT_IMPORTED_CA_LIST geprüft werden:
   TUC_KON_037 „Zertifikat prüfen“ {
   certificate =Zertifikat;
   qualifiedCheck = not_required;
   offlineAllowNoCheck = true;
   intendedKeyUsage= intendedKeyUsage(Zertifikate aus CERT_IMPORTED_CA_LIST);
   validationMode = NONE }

   Für alle anderen Zertifikate gilt: {
   certificate = [C.CH.ENC];
   qualifiedCheck=not_required;
   offlineAllowNoCheck=false;
   policyList =[oid_egk_enc];
   intendedKeyUsage= intendedKeyUsage(C.CH.ENC);
   validationMode=OCSP }
   oder
   {
   certificate = [C.CH.ENCV];
   qualifiedCheck=not_required;
   offlineAllowNoCheck=false;
   policyList =[oid_egk_encv];
   intendedKeyUsage= intendedKeyUsage(C.CH.ENCV);
   validationMode=OCSP }
   oder
   {
   certificate = [C.HCI.ENC];
   qualifiedCheck=not_required;
   offlineAllowNoCheck=false;
   policyList =[oid_smc_b_enc];
   intendedKeyUsage= intendedKeyUsage(C.HCI.ENC);
   validationMode=OCSP }
   oder
   {
   certificate = [C.HP.ENC];
   qualifiedCheck=not_required;
   offlineAllowNoCheck=false;
   policyList =[oid_hba_enc, oid_vk_pt_enc, oid_vk_eaa_enc];
   intendedKeyUsage= intendedKeyUsage(C.HP.ENC);
   validationMode=OCSP }

5. Die öffentlichen Schlüssel werden aus den Zertifikaten extrahiert, falls sie nicht direkt übergeben wurden.
   Falls ein Schlüssel keinen der zugelassenen Verschlüsselungsalgorithmen gemäß [gemSpec_Krypt#3.5.2] bzw. [gemSpec_Krypt#5.8] erlaubt, wird Fehler 4200 gemeldet.

	6. Der Konnektor generiert einen symmetrischen Schlüssel. Dabei muss der symmetrische Schlüssel den Kriterien aus [gemSpec_Krypt#2.4] entsprechen. 7. Der Konnektor verschlüsselt das Dokument oder Teile des Dokuments mit dem generierten symmetrischen Schlüssel. a. CMS: Es MÜSSEN die Vorgaben aus [gemSpec_Krypt#3.5.1] beachtet werden. b. XMLEnc: Alle zu verschlüsselnden XML-Elemente werden mit demselben symmetrischen Schlüssel verschlüsselt. Dabei MÜSSEN die Vorgaben aus [gemSpec_Krypt#3.1.4] beachtet werden. 8. Der symmetrische Schlüssel wird asymmetrisch für die einzelnen Identitäten verschlüsselt. Dabei müssen die Vorgaben aus [gemSpec_Krypt#3.1.5; 3.5.2; 5.8] beachtet werden. 9. Das Zieldokument wird erstellt. XMLEnc Format und Inhalt des verschlüsselten Dokuments SOLLEN dem XML Encryption Format in [COMMON_PKI#Part 8] folgen. Zum Format des verschlüsselten XML-Dokumentes siehe auch Tabelle TAB_KON_073 Vorgaben zum Format verschlüsselter XML-Dokumente. Die verschlüsselten Datenelemente (EncryptedData) werden erstellt. EncryptedData ersetzt jeweils das zu verschlüsselnde Element des XML-Dokuments. In [COMMON_PKI] wird die Verwendung des Attributs Type in EncryptedData ausgeschlossen; diese Spezifikation sieht jedoch dessen Verwendung für verschlüsselte XML-Bestandteile (element, content) wie in [XMLEnc] beschrieben vor. Der Namespace von EncryptedData ist als http://www.w3.org/2001/04/xmlenc# anzugeben. Für das Element EncryptedData wird das Sub-Element EncryptionMethod mit Angaben zum Verschlüsselungsalgorithmus als obligatorisch vorgegeben, ebenso die Elemente KeyInfo und CipherData. Das Element EncryptedData/KeyInfo hat den Namespace "http://www.w3.org/2000/09/xmldsig#". Es muss pro Hybridschlüssel ein Element EncryptedKey enthalten. In jedem EncryptedKey-Element wird neben dem eigentlichen Hybridschlüssel ein Element zur EncryptionMethod der asymmetrischen Verschlüsselung und ein KeyInfo-Element mit dem Zertifikat angelegt, das für die Verschlüsselung des symmetrischen Schlüssels verwendete wurde. Das Zertifikat wird jeweils im Element EncryptedKey/KeyInfo/X509Data/X509Certificate

	base64-kodiert und darin DER-kodiert abgelegt. Hybridschlüssel (RSA): Das Element EncryptedData/KeyInfo/EncryptedKey muss die Verschlüsselungsmethode im Element EncryptionMethod angeben, den hybridSchlüssel im Element CipherData speichern und das Zertifikat, mit dem der symmetrische Schlüssel zum Hybridschlüssel verschlüsselt wurde, im Element EncryptedKey/KeyInfo/X509Data/X509Certificate base64-kodiert und darin DER-kodiert ablegen. Hybridschlüssel (ECC): Es gelten die Vorgaben aus [gemSpec_Krypt#5.8] CMS: Es ist CMS mit Authenticated-Enveloped-Data Content Type gemäß [RFC-5083] und der AES-GCM-Encryption gemäß [RFC-5084] zu verwenden. Bei der Verschlüsselung des „content-encryption key" wird die Technik „key transport" eingesetzt. Pro Empfänger wird eine Instanz vom Typ KeyTransRecipientInfo erzeugt. Dabei ist für RecipientIdentifier die Option IssuerAndSerialNumber zu wählen. ContentType = OID {… authEnvelopedData} = 1.2.840.113549.1.9.16.1.23 Im Fall ECC sind die Vorgaben aus [gemSpec_Krypt#5.8] zur Erzeugung des Hybridschlüssels zu beachten. Im Fall RSA sind die Vorgaben aus [gemSpec_Krypt#3.5.2] zur Erzeugung des Hybridschlüssels zu beachten. 10. Der symmetrische Schlüssel wird aktiv gelöscht (überschrieben).
Varianten/ Alternativen	Zur Rückgabe der Hybridschlüssel MUSS auch die Variante vorgesehen werden, dass die Hybridschlüssel („KeyInfo") nicht eingebettet im Zieldokument zurückgegeben werden, sondern separat. Im Fall des Verschlüsselungsverfahrens S/MIME wird der Standardablauf des CMS Verschlüsselungsverfahrens durch einen vorgelagerten S/MIME-Vorbereitungsschritt und einen nachgelagerten S/MIME-Nachbereitungsschritt ergänzt. Das S/MIME-Verfahren MUSS konform [S/MIME] und SOLL konform [COMMON_PKI#Part 3] erfolgen. Der S/MIME-Vorbereitungsschritt bereitet das übergebene MIME-Dokument gemäß [S/MIME#3.1] auf die nachfolgende CMS-Verschlüsselung durch eine Kanonisierung für Text [S/MIME#3.1.1] vor. Eine weitere Kanonisierung oder eine Anpassung des Transfer Encodings [S/MIME#3.1.2] erfolgt nicht. Im S/MIME-Nachbereitungsschritt wird das im Standardablauf erzeugt CMS-Objekt in eine MIME-Nachricht vom Typ „application/pkcs7-mime" eingebettet. Sämtliche Header-Felder der Nachricht MÜSSEN in die Header-Felder der S/MIME-Nachricht übernommen werden. Die im Folgenden explizit zu setzenden Header-Felder überschreiben die betroffenen Header-Felder.

	Es MUSS ein neues message-id Element für den S/MIME-Header generiert werden. "MIME-Version: 1.0" MUSS definiert sein. Das Feld "Subject" MUSS mit "Subject: Verschlüsselte Nachricht" überschrieben werden. Die Codierung des verschlüsselten Inhalts der Nachricht MUSS in "base64" erfolgen. Entsprechend ist das zugehörige Header-Feld zu füllen: "Content-Transfer-Encoding: base64". Das Feld "Content-Type:" ist als "application/pkcs7-mime" zu definieren. Die weiteren Attribute dieses Feldes sind: • "smime-type=enveloped-data;" • "name=$dateiname", wobei $dateiname auf ".p7m" endet. Das Feld "Content-Disposition" definiert den Inhalt der Nachricht als Dateianhang: "Content-Disposition: attachment; filename=$dateiname" Zu Schritten 5 und 8 für TI-fremde X.509-Zertifikate Der Konnektor MUSS beim asymmetrischen Anteil der hybriden Verschlüsselung auch TI-fremde X.509-Zertifikate unterstützen, wenn diese von einem CA-Zertifikat aus CERT_IMPORTED_CA_LIST ausgestellt wurden und die kryptographischen Vorgaben aus Tabelle [gemSpec_Krypt#Tab_KRYPT_002] erfüllen. Der Konnektor MUSS Anfragen zur Hybridverschlüsselung mit einer Fehlermeldung (Fehler 4200) abweisen, wenn hierfür TI-fremde X509-Zertifikate vorgegeben werden, die nicht die kryptographischen Vorgaben aus Tabelle [gemSpec_Krypt#Tab_KRYPT_002] oder [gemSpec_Krypt#Tab_KRYPT_002a] erfüllen.
Fehlerfälle	Siehe Tabelle TAB_KON_740 Fehlercodes TUC_KON_070 „Daten hybrid verschlüsseln". Wenn im Ablauf des TUCs ein anderer Fehler als die in Tabelle TAB_KON_740 beschriebenen Fehler auftritt, wird Fehler 4105 gemeldet. (->4) Schritt 4 – Zertifikatsprüfung „für alle anderen Zertifikate" Für MGM_LU_ONLINE=Enabled gilt: Liefert die Zertifikatsprüfung (OCSP-Abfrage) mdt. eine der folgenden Warnungen gemäß [gemSpec_PKI#Tab_PKI_274] • CERT_REVOKED • CERT_UNKNOWN dann wird der TUC mit Fehler 4105 abgebrochen, Ausnahme: Falls im Falle crypt=RSA_ECC der Hybridschlüssel nur für eines der beiden Zertifikate erzeugt werden konnte, dann wird die Warnung 4259 mit <Zertifikat> gemäß TAB_KON_747 in der Response zurückgegeben.

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	Abbildung PIC_KON_058 Aktivitätsdiagramm „Daten hybrid verschlüsseln“ Das Diagramm dient nur der Veranschaulichung und ist nicht vollständig. Beispielsweise enthält es nicht die Steuerung durch den Parameter `crypt`.

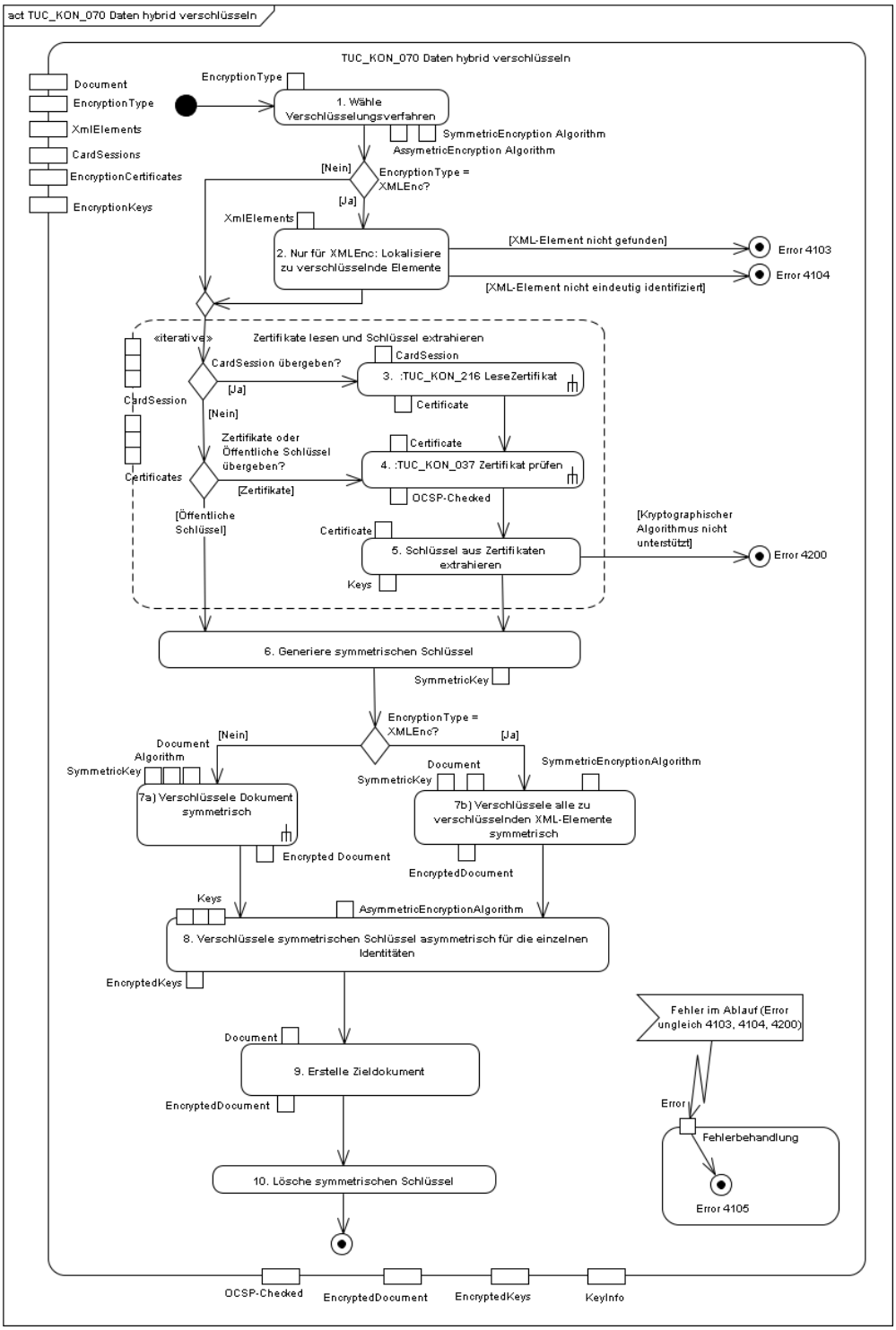


Abbildung 13: PIC_KON_058 Aktivitätsdiagramm „Daten hybrid verschlüsseln“

Tabelle 169: TAB_KON_073 Vorgaben zum Format verschlüsselter XML-Dokumente

#	Beschreibung

	xenc:EncryptedData MUSS ein ds:KeyInfo Element enthalten, welches wiederum ein xenc:EncryptedKey Element enthält.
	Der xenc:EncryptedKey MUSS [XMLEnc] konform sein.
	Die xenc:EncryptionMethod für den Schlüssel MUSS gemäß [gemSpec_Krypt#3.1.5] gewählt werden
	Der xenc:EncryptedKey MUSS ein ds:KeyInfo Element mit ds:X509Data und ds:X509Certificate Subelement enthalten, in dem das X.509-Zertifikat hinterlegt wird.

Tabelle 170: TAB_KON_740 Fehlercodes TUC_KON_070 „Daten hybrid verschlüsseln“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4103	Technical	Error	XML-Element nicht gefunden
4104	Technical	Error	XML-Element nicht eindeutig identifiziert. (Überschneidung)
4105	Technical	Error	hybride Verschlüsselung konnte nicht durchgeführt werden
4200	Security	Error	Schlüssel erlaubt keinen zugelassenen Verschlüsselungsalgorithmus
4259	Technical	Warning	Verschlüsselung für Zertifikat <Zertifikat> nicht möglich

[<=]

4.1.7.4.2 TUC_KON_071 „Daten hybrid entschlüsseln”

TIP1-A_4617-02 - TUC_KON_071 „Daten hybrid entschlüsseln“
Der Konnektor MUSS den technischen Use Case TUC_KON_071 „Daten hybrid entschlüsseln” umsetzen.

Tabelle 171: TAB_KON_140 – TUC_KON_071 „Daten hybrid entschlüsseln"

Element	Beschreibung
Name	TUC_KON_071 „Daten hybrid entschlüsseln"
Beschreibung	Ein hybrid verschlüsseltes Dokument, das konform zu TUC_KON_070 erstellt wurde, wird entschlüsselt. Es muss eine asymmetrische Verschlüsselung vorliegen, zu der der Schlüssel auf einer Karte vorliegt.
Auslöser	Aufruf in einem fachlichen Use Case oder des Verschlüsselungsbasisdienstes.
Vorbedingungen	Die Karte mit dem privaten Schlüssel muss gesteckt sein und der Sicherheitszustand zur Nutzung des privaten Schlüssels muss gesetzt sein. Ein konform zu TUC_KON_070 hybrid verschlüsseltes Dokument liegt vor. Bei XML-Dokumenten: Das Dokument enthält EncryptedData Elemente. Falls mehrere Elemente des Dokumentes zu entschlüsseln sind, müssen diese alle mit demselben symmetrischen Schlüssel verschlüsselt sein.
Eingangsdaten	• encryptedDocument (Zu entschlüsselndes Dokument) • cardSession (Kartensitzung; unterstützt werden SM-B, HBAx und eGK mit der jeweiligen C.ENC-KeyReference. • privateKeyReference (Referenz auf den privaten Schlüssel; unterstützt werden SM-B, HBAx und eGK mit der jeweiligen C.ENC-KeyReference). • encryptionCertificate – *optional* (Verschlüsselungszertifikat passend zur Schlüsselreferenz). • encryptionCertificateReference – *optional* (Referenz auf das Zertifikat auf obiger Karte passend zur Schlüsselreferenz). • encryptedKey – *optional, falls nicht in encryptedDocument enthalten* (asymmetrisch verschlüsselter symmetrischer Schlüssel) Darüber hinaus werden die folgenden, vom Dokumentformat und dem Verschlüsselungsverfahren abhängigen Eingangsdaten benötigt: Bei XML-Dokumenten: • xmlElements – *optional/verpflichtend, wenn encryptionType = XMLEnc*

	(bei XML-Dokumenten Angabe der zu entschlüsselnden Teile des XML-Dokuments)
Komponenten	Konnektor, Kartenterminal, Karte
Ausgangsdaten	• plainDocument (Unverschlüsseltes Dokument. Bei XML-Dokumenten: Das EncryptedData-Element ist durch das entschlüsselte ersetzt.)
Standardablauf	1. Das Verfahren zum Entschlüsseln wird entsprechend dem Format des übergebenen zu entschlüsselnden Dokuments (EncryptedDocument) gewählt. Der Konnektor MUSS beim asymmetrischen Anteil der Entschlüsselung hybrid verschlüsselter Dokumente die in [gemSpec_Krypt] beschriebenen Verfahren unterstützen. 2. XMLEnc: Das EncryptedData Element (oder mehrere Elemente) werden im Dokument lokalisiert. Falls sie nicht oder nicht eindeutig gefunden werden können wird Fehler 4103 bzw. 4104 gemeldet. Ist in einem EncryptedData Element ein vom Konnektor nicht unterstützter Mechanismus spezifiziert, wird Fehler 4201 gemeldet. 3. Falls erforderlich, wird TUC_KON_216 „Lese Zertifikat" aufgerufen, um das Zertifikat von der Karte zu lesen. 3.1 Die Kenntnis des Zertifikats kann erforderlich sein, um im Zertifikat kodierte Verschlüsselungsparameter auszulesen. (Zur Zeit der Erstellung dieser Spezifikation werden zur Laufzeit keine zusätzlichen Parameter aus dem Zertifikat benötigt, da alle nötigen Informationen aus den PKI- und Kartenspezifikationen abgeleitet werden können.) 4. XMLEnc: Es wird geprüft, ob die Verschlüsselungsparameter (EncryptionMethod in EncryptedKey) zum referenzierten privaten Schlüssel auf der Karte passen. Ist dies nicht der Fall, bricht der Use Case mit Fehler 4106 ab. 5. Es wird TUC_KON_219 „Entschlüssele" aufgerufen, um den symmetrischen Schlüssel mit Hilfe des angegebenen privaten Schlüssels zu entschlüsseln. 6. Mit dem symmetrischen Schlüssel wird der unverschlüsselte Dateninhalt wiederhergestellt. 6.1 XMLEnc: Das EncrpytedData Element wird durch die entschlüsselten Daten ersetzt. 7. Der symmetrische Schlüssel wird aktiv gelöscht (überschrieben).
Varianten/Alternativen	Zu 6.: Zur Unterstützung von Bestandssystemen werden, neben den für den symmetrischen Teil der hybriden Verschlüsselung vorgeschriebenen kryptographischen Algorithmen, für den

	symmetrischen Teil der hybriden Entschlüsselung auch folgende Algorithmen unterstützt (siehe [gemSpec_Krypt#3.5.1]): • AES-128 GCM • AES-192 GCM RSA- und ECC-basierter Hybridschlüssel: Wenn sowohl ein RSA- als auch ein ECC-basierter Hybridschlüssel vorliegen, muss zuerst die Entschlüsselung des ECC-basierten Hybridschlüssels erfolgen. Falls dabei ein Fehler auftritt, muss der Fehler protokolliert werden, und dann - ohne Abbruch - mit der Entschlüsselung des RSA-basierten Hybridschlüssels fortgefahren werden.
Fehlerfälle	Siehe Tabelle TAB_KON_142 Fehlercodes TUC_KON_071 „Daten hybrid entschlüsseln". Wenn im Ablauf des TUCs ein anderer Fehler als die in Tabelle TAB_KON_142 Fehlercodes TUC_KON_071 „Daten hybrid entschlüsseln" beschriebenen Fehler auftritt, wird Fehler 4107 gemeldet.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	Abbildung PIC_KON_059 Aktivitätsdiagramm „Daten hybrid entschlüsseln"

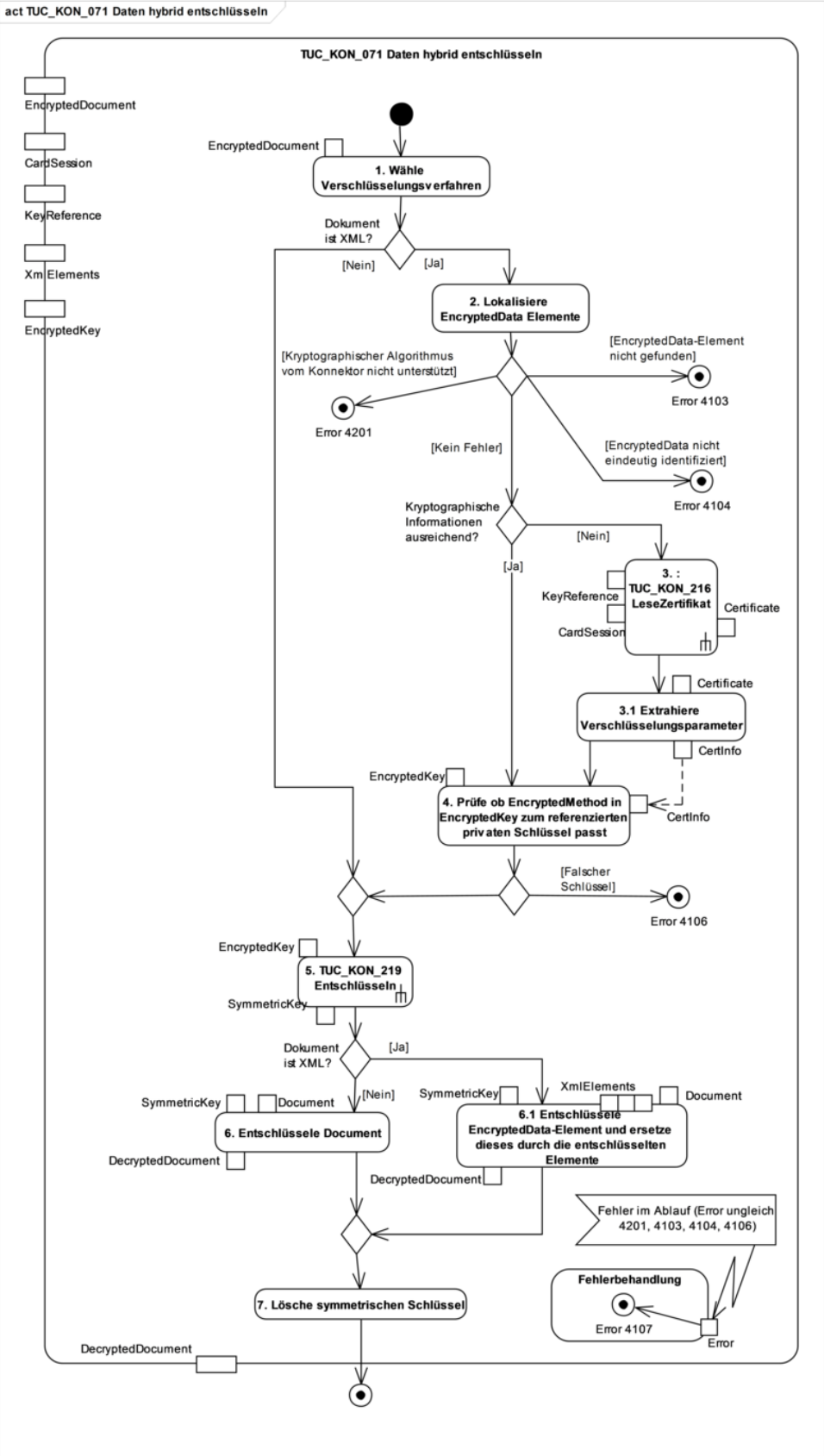

act TUC_KON_071 Daten hybrid entschlüsseln
TUC_KON_071 Daten hybrid entschlüsseln
EncryptedDocument
CardSession
KeyReference
Xml Elements
EncryptedKey
EncryptedDocument
1. Wähle Verschlüsselungsverfahren
Dokument ist XML?
[Nein]
[Ja]
2. Lokalisiere EncryptedData Elemente
[Kryptographischer Algorithmus vom Konnektor nicht unterstützt]
Error 4201
[EncryptedData-Element nicht gefunden]
Error 4103
[Kein Fehler]
[EncryptedData nicht eindeutig identifiziert]
Error 4104
Kryptographische Informationen ausreichend?
[Nein]
[Ja]
3. : TUC_KON_216 LeseZertifikat
KeyReference
CardSession
Certificate
Certificate
3.1 Extrahiere Verschlüsselungsparameter
CertInfo
EncryptedKey
4. Prüfe ob EncryptedMethod in EncryptedKey zum referenzierten privaten Schlüssel passt
CertInfo
[Falscher Schlüssel]
Error 4106
EncryptedKey
5. TUC_KON_219 Entschlüsseln
SymmetricKey
Dokument ist XML?
[Ja]
[Nein]
SymmetricKey
Document
6. Entschlüssele Document
DecryptedDocument
SymmetricKey
XmlElements
Document
6.1 Entschlüssele EncryptedData-Element und ersetze dieses durch die entschlüsselten Elemente
DecryptedDocument
Fehler im Ablauf (Error ungleich 4201, 4103, 4104, 4106)
Fehlerbehandlung
Error 4107
Error
7. Lösche symmetrischen Schlüssel
DecryptedDocument

Abbildung 14: PIC_KON_059 Aktivitätsdiagramm „Daten hybrid entschlüsseln“

Tabelle 172: TAB_KON_142 Fehlercodes TUC_KON_071 „Daten hybrid entschlüsseln“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4106	Technical	Error	falscher Schlüssel
4107	Technical	Error	hybride Entschlüsselung konnte nicht durchgeführt werden
4103	Technical	Error	XML-Element nicht gefunden
4104	Technical	Error	XML-Element nicht eindeutig identifiziert
4201	Technical	Error	kryptographischer Algorithmus vom Konnektor nicht unterstützt

[<=]

4.1.7.4.3 TUC_KON_072 „Daten symmetrisch verschlüsseln”

TIP1-A_4618 - TUC_KON_072 „Daten symmetrisch verschlüsseln”
Der Konnektor MUSS den technischen Use Case TUC_KON_072 „Daten symmetrisch verschlüsseln“ umsetzen.

Tabelle 173: TAB_KON_741 – TUC_KON_072 „Daten symmetrisch verschlüsseln“

Element	Beschreibung
Name	TUC_KON_072 „Daten symmetrisch verschlüsseln“
Beschreibung	Es wird ein Dokument symmetrisch verschlüsselt. Dabei kann der zu verwendende symmetrische Schlüssel optional übergeben werden.
Auslöser	Aufruf durch ein Fachmodul in einem fachlichen Use Case
Vorbedingungen	keine
Eingangsdaten	• documentToBeEncrypted (zu verschlüsselndes Dokument.) • symmetricKey – *optional* (zu verwendender symmetrischer Schlüssel)
Komponenten	Konnektor
Ausgangsdaten	• encryptedDocument (Verschlüsseltes Dokument) • symmetricKey – *optional/verpflichtend, wenn Schlüssel durch den TUC erzeugt wurde* (erzeugter symmetrischer Schlüssel)
Standardablauf	1. Wurde kein symmetrischer Schlüssel übergeben, so wird ein Schlüssel erzeugt. Die Qualität des

	Schlüssels muss den Vorgaben in [gemSpec_Krypt#2.2] genügen. 2. Das Dokument wird mit dem erzeugten oder übergebenen symmetrischen Schlüssel verschlüsselt. Als Verfahren zum Verschlüsseln wird CMS gewählt ([RFC5652]). Die Content Type Option „Encrypted-data Content Type" ist zu verwenden. ContentType = OID{… pkcs-7 encryptedData} = 1.2.840.113549.1.7.6 Die symmetrische Verschlüsselung binärer Daten erfolgt nach den Vorgaben gemäß [gemSpec_Krypt#GS-A_5016]. Falls die Verschlüsselung fehlschlägt, wird Fehler 4108 gemeldet. 3. Das verschlüsselte Dokument und der symmetrische Schlüssel (falls dieser erzeugt wurde) werden zurückgeliefert.
Varianten/Alternativen	keine
Fehlerfälle	Siehe Standardablauf.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 174: TAB_KON_742 Fehlercodes TUC_KON_072 „Daten symmetrisch verschlüsseln"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4108	Technical	Error	Symmetrische Verschlüsselung konnte nicht durchgeführt werden

[<=]

4.1.7.4.4 TUC_KON_073 „Daten symmetrisch entschlüsseln"

TIP1-A_4619 - TUC_KON_073 „Daten symmetrisch entschlüsseln"
Der Konnektor MUSS den technischen Use Case TUC_KON_073 „Daten symmetrisch entschlüsseln" umsetzen.

Tabelle 175: TAB_KON_743 - TUC_KON_073 „Daten symmetrisch entschlüsseln"

Element	Beschreibung
Name	TUC_KON_073 „Daten symmetrisch entschlüsseln"

Beschreibung	Es wird ein Dokument symmetrisch entschlüsselt. Der zu verwendende symmetrische Schlüssel wird übergeben.
Auslöser	Aufruf durch ein Fachmodul in einem fachlichen Use Case
Vorbedingungen	keine
Eingangsdaten	• encryptedDocument (Verschlüsseltes Dokument) • symmetricKey (zu verwendender symmetrischer Schlüssel)
Komponenten	Konnektor
Ausgangsdaten	• plainDocument (Entschlüsseltes Dokument)
Standardablauf	Das verschlüsselte Dokument wird mit dem symmetrischen Schlüssel entschlüsselt. Als Verfahren zum Entschlüsseln wird CMS gewählt ([RFC5652]). Das entschlüsselte Dokument wird zurückgeliefert.
Varianten/Alternativen	keine
Fehlerfälle	Bei Auftreten eines Fehlers im Standardablauf wird Fehlercode 4109 gemeldet.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 176: TAB_KON_744 Fehlercodes TUC_KON_073 „Daten symmetrisch entschlüsseln“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4109	Technical	Error	symmetrische Entschlüsselung konnte nicht durchgeführt werden

[<=]

4.1.7.4.5 TUC_KON_075 „Symmetrisch verschlüsseln“

A_18001 - TUC_KON_075 „Symmetrisch verschlüsseln”
Der Konnektor MUSS den technischen Use Case TUC_KON_075 „Symmetrisch verschlüsseln“ umsetzen.

Tabelle 177: TAB_KON_860 – TUC_KON_075 „Symmetrisch verschlüsseln“

Element	Beschreibung

Name	TUC_KON_075 „Symmetrisch verschlüsseln“
Beschreibung	Es werden binäre Daten symmetrisch verschlüsselt. Optional können der zu verwendende symmetrische Schlüssel und AssociatedData für Authenticated Encryption with Associated Data (AEAD) übergeben werden.
Auslöser	Aufruf durch ein Fachmodul in einem fachlichen Use Case
Vorbedingungen	keine
Eingangsdaten	• dataToBeEncrypted (zu verschlüsselnde Daten) • symmetricKey – optional (zu verwendender symmetrischer Schlüssel) • associatedData - optional (Parameter für den Verschlüsselungsalgorithmus)
Komponenten	Konnektor
Ausgangsdaten	• encryptedData (Verschlüsselte Daten mit der Struktur gemäß Punkt 2 aus A_18004) • symmetricKey – optional/verpflichtend, wenn Schlüssel durch den TUC erzeugt wurde (erzeugter symmetrischer Schlüssel)
Standardablauf	1. Wurde kein symmetrischer Schlüssel übergeben, so wird ein Schlüssel erzeugt. Die Qualität des Schlüssels muss den Vorgaben in GS-A_4367 genügen. 2. dataToBeEncrypted wird mit dem erzeugten oder übergebenen symmetrischen Schlüssel unter Berücksichtigung der optional übergebenen associatedData verschlüsselt. Die symmetrische Verschlüsselung binärer Daten erfolgt nach den Vorgaben gemäß A_17872. 3. encryptedData wird erzeugt mit der Struktur gemäß Punkt 2 aus A_18004. 4. Das verschlüsselte Dokument und der symmetrische Schlüssel (falls dieser erzeugt wurde) werden zurückgeliefert.
Varianten/Alternativen	keine
Fehlerfälle	-> 2: Falls die Verschlüsselung fehlschlägt, wird Fehler 4108 gemäß TAB_KON_742 gemeldet.

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

[<=]

4.1.7.4.6 TUC_KON_076 „Symmetrisch entschlüsseln"

A_18002 - TUC_KON_076 „Symmetrisch entschlüsseln"
Der Konnektor MUSS den technischen Use Case TUC_KON_076 „Symmetrisch entschlüsseln" umsetzen.

Tabelle 178: TAB_KON_861 - TUC_KON_076 „Symmetrisch entschlüsseln"

Element	Beschreibung
Name	TUC_KON_076 „Symmetrisch entschlüsseln"
Beschreibung	Es werden verschlüsselte Daten symmetrisch entschlüsselt. Für Authenticated Encryption with Associated Data (AEAD) kann AssociatedData optional übergeben werden. Der zu verwendende symmetrische Schlüssel wird übergeben.
Auslöser	Aufruf durch ein Fachmodul in einem fachlichen Use Case
Vorbedingungen	keine
Eingangsdaten	• encryptedData (Verschlüsselte Daten mit der Struktur gemäß Punkt 2 aus A_18004) • symmetricKey (zu verwendender symmetrischer Schlüssel) • associatedData - optional (Parameter für den Verschlüsselungsalgorithmus)
Komponenten	Konnektor
Ausgangsdaten	• plainData (Entschlüsselte Daten)
Standardablauf	Das verschlüsselte Dokument wird mit dem symmetrischen Schlüssel und associatedData unter Verwendung der kryptographischen Verfahren aus A_17872 entschlüsselt. Die entschlüsselten Daten werden zurückgeliefert.

Varianten/Alternativen	keine
Fehlerfälle	Bei Auftreten eines Fehlers im Standardablauf wird Fehlercode 4109 gemäß TAB_KON_744 gemeldet.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

[<=]

4.1.7.5 Operationen an der Außenschnittstelle

TIP1-A_4620-03 - Basisdienst Verschlüsselungsdienst
Der Konnektor MUSS für Clients einen Basisdienst Verschlüsselungsdienst anbieten.

Tabelle 179: TAB_KON_745 Basisdienst Verschlüsselungsdienst

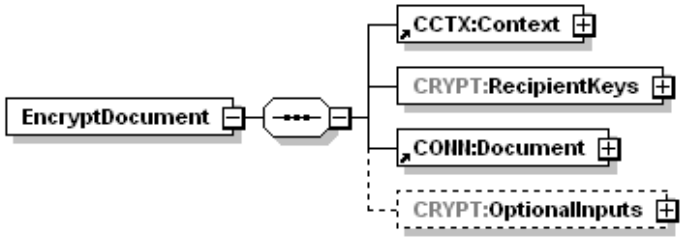
Name	`EncryptionService`	
Version (KDV)	6.1.0 (WSDL-Version), 6.1.1 (XSD-Version) 6.1.1 (WSDL-Version), 6.1.2 (XSD-Version)	
Namensraum	Siehe GitHub	
Namensraum-Kürzel	`CRYPT` für Schema und `CRYPTW` für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	`EncryptDocument`	Dokument hybrid verschlüsseln
	`DecryptDocument`	Dokument hybrid entschlüsseln
WSDL	`EncryptionService.wsdl` (WSDL-Version 6.1.0) `EncryptionService_v6_1_1.wsdl`	
Schema	`EncryptionService.xsd` (XSD-Version 6.1.1) `EncryptionService_v6_1_2.xsd`	

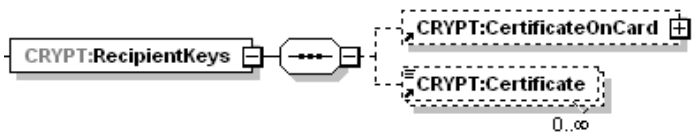
[<=]

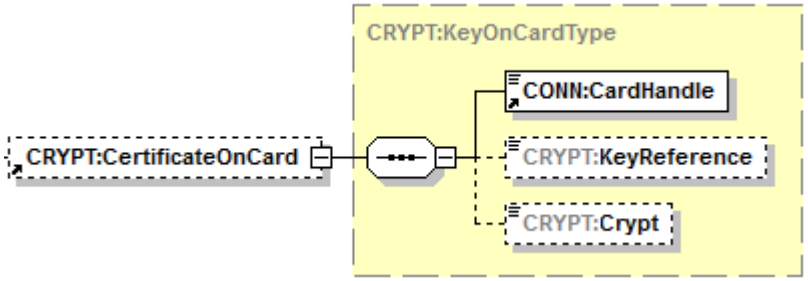
4.1.7.5.1 EncryptDocument

TIP1-A_4621-02 - Operation EncryptDocument
Der Basisdienst Verschlüsselungsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation EncryptDocument anbieten.

Tabelle 180: TAB_KON_071 Operation EncryptDocument

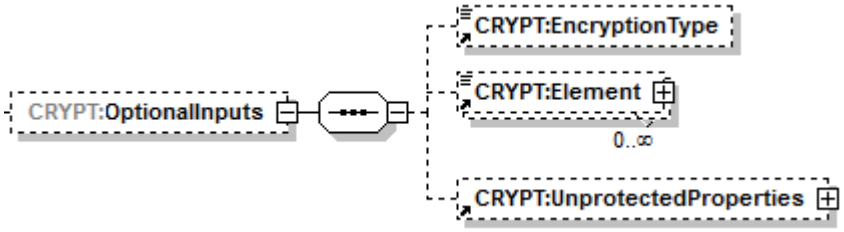
<table>
<tr><th>Name</th><td>EncryptDocument</td></tr>
<tr><td>Beschreibung</td><td>Diese Operation verschlüsselt ein übergebenes Dokument hybrid.
Es werden die Dokumententypen <code>Alle_DocFormate</code> unterstützt.
Für die hybride Verschlüsselung wird ein asymmetrischer Schlüssel aus einem X.509v3-Zertifikat genutzt. Dieses Zertifikat kann von einer Karte kommen oder als Parameter übergeben werden. Pro Operationsaufruf können mehrere Hybridschlüssel erzeugt werden. Übergibt der Aufrufer die Zertifikate beim Aufruf, steuert er durch die Wahl der Zertifikate, ob RSA-basierte oder ECC-basierte Hybridschlüssel erzeugt werden. Wenn das Verschlüsselungszertifikat von einer Karte kommt, kann der Aufrufer durch Angabe des Kryptoverfahrens <code>crypt</code> steuern, ob Hybridschlüssel für RSA oder für ECC oder beide erzeugt werden. Das Defaultverhalten ist die Hybridschlüsselerzeugung für RSA und entspricht dem Verhalten aus der Version 6.1.0 der Schnittstelle.
Es werden die folgenden Karten unterstützt: HBAx und SM-B. Die Operation EncryptDocument DARF das Verschlüsseln mit der eGK NICHT unterstützen.
Für alle Dokumenttypen wird immer das gesamte Dokument verschlüsselt.</td></tr>
<tr><td></td><td> 

<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>Context</code></td><td>Aufrufkontext:
MandantID, ClientSystemID, WorkplaceId verpflichtend
UserID verpflichtend bei HBAx, bei SM-B nicht ausgewertet</td></tr>
</table>

</td></tr>
</table>



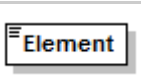
Das RecipientKeys-Element identifiziert die Empfänger der zu verschlüsselnden Nachricht über X.509-Zertifikate (öffentliche Schlüssel). Quelle für die Zertifikate kann eine gesteckte Karte sein, die per CertificateOnCard-Element referenziert wird, oder der Aufrufer, der X.509-Zertifikate im Certificate-Element übergibt.

`CardHandle`	Identifiziert die zu verwendende Karte mit dem (öffentlichen) Schlüssel. Ist das Element nicht vorhanden, so werden nur Zertifikate per Element `Certificate` übergeben.
`KeyReference`	Der Wert dieses Parameters ist in Tabelle TAB_KON_747 KeyReference für Encrypt-/DecryptDocument spezifiziert. Ist der Parameter nicht angegeben, gilt der Default-Wert C.ENC.
`Crypt`	Optional; Default: siehe TAB_KON_859 Wertebereich: [ENC_CRYPT] Gibt den Typ von Zertifikaten vor, die von der per CardHandle referenzierten Karte für die Erzeugung der Hybridschlüssel gemäß Tabelle TAB_KON_747 verwendet werden.
`Certificate`	`Certificate` ist ein Base64-kodiertes XML-Element, in dem das Zertifikat, das den asymmetrischen Schlüssel enthält (öffentlicher Schlüssel), DER-kodiert übergeben wird. Es kann eine Liste von Zertifikaten übergeben werden. Kommt das Zertifikat ausschließlich von einer Karte, dann kann dieser Parameter weggelassen werden.
Document	
`CONN:Document`	Dieses entsprechend [OASIS-DSS] Section 2.4.2 spezifizierte Element enthält das zu verschlüsselnde Dokument, wobei die Kindelemente CONN:`Base64XML` und `dss:Base64Data` verwendet werden. Im Fall `dss:Base64Data` wird ein etwaig übergebenes MIME-Type-Attribut nicht ausgewertet.

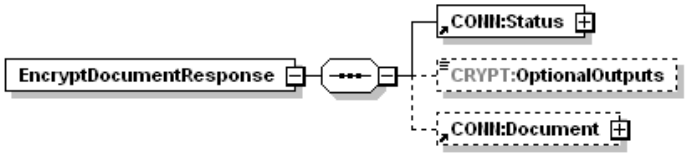


CRYPT: Optional Inputs	Enthält eine Auswahl der folgenden unten näher erläuterten (optionalen) Eingabeparameter:

Encryption Type	Zu wählendes Verschlüsselungsverfahren, wobei folgende URI vorgesehen sind: • XMLEnc: „http://www.w3.org/TR/xmlenc-core/" • CMS: „urn:ietf:rfc:5652" • S/MIME: „urn:ietf:rfc:5751" Im Fall XMLEnc wird ein Base64-codiertes XML-Dokument im Element `CONN:Document/CONN:Base64XML` übergeben. In den Fällen CMS und S/MIME wird ein Base64-codiertes Binär-Dokument im Element `CONN:Document/dss:Base64Data` übergeben . Ist der Parameter EncryptionType nicht gesetzt, dann gilt folgendes Default-Verhalten: Für ein im Element `CONN:Document/CONN:Base64XML` übergebenes XML-Dokument wird als Verschlüsselungsverfahren [XMLEnc] angewandt, und für ein im Element `CONN:Document/dss:Base64Data` übergebenes Dokument wird das Verschlüsselungsverfahren CMS angewandt. XML-Documente werden nach Type=http://www.w3.org/2001/04/xmlenc#Element verschlüsselt. Im Fall S/MIME ist das in `CONN:Document/dss:Base64Data` übergebene Dokument eine MIME-Nachricht.



Element	Der Parameter wird nicht ausgewertet.

	CRYPT: Unprotected Properties	Dieses optionale Element wird im CMS-Fall (EncryptionType = urn:ietf:rfc:5652) ausgewertet. Die Elemente ./UnprotectedProperties/Property/Value/CMSAttribute müssen base64/DER-kodiert ein vollständiges ASN.1-Attribute enthalten, definiert in [CMS# 9.1.AuthenticatedData Type]. Es muss bei der Erstellung des CMS-Containers unter "unauthAttrs" aufgenommen werden. Das zugehörige Element ./UnprotectedProperties/Property/Identifier wird nicht ausgewertet.
Rückgabe		
	Status	Enthält den Ausführungsstatus der Operation.
	CRYPT: Optional Outputs	Kann – in zukünftigen Versionen der Spezifikation – optionale Ausgabeparameter enthalten.
	CONN: Document	Enthält das verschlüsselte Dokument in base64-codierter Form, wenn die Verschlüsselung erfolgreich durchgeführt wurde. Im Fall XMLEnc wird das Base64-codierte verschlüsselte XML-Dokument im Element CONN:Document/CONN:Base64XML zurückgegeben. Im Fall CMS wird das Base64-codierte Binär-Dokument im Element CONN:Document/dss:Base64Data zurückgegeben. Im Fall S/MIME wird die Base64-codierte S/MIME-Nachricht im Element CONN:Document/dss:Base64Data zurückgegeben. Das Attribut CONN:Document/dss:Base64Data/@MimeType wird auf „application/pkcs7-mime“ gesetzt. Die S/MIME-Nachricht hat Content-Transfer-Encoding: base64.
Fehler	Bei Auftreten eines Fehlers im Standardablauf werden Fehlercodes entsprechend TAB_KON_141 gemeldet.	
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Der Ablauf der Operation EncryptDocument ist in Tabelle TAB_KON_746 Ablauf EncryptDocument beschrieben:

Tabelle 181: TAB_KON_746 Ablauf EncryptDocument

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab. Wird die Operation für einen nicht unterstützten Kartentypen aufgerufen, so bricht die Operation mit Fehler 4058 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 {
		mandatId =$context..mandantId;
		clientSystemId = $context.clientSystemId;
		cardHandle = $context..cardHandle;
		userId = $context.userId }

4.	TUC_KON_070 „Daten hybrid verschlüsseln“	Die hybride Verschlüsselung wird durchgeführt. Tritt hierbei ein Fehler auf, bricht die Operation ab. Die KeyInfo, d.h. die Liste der Hybridschlüssel inklusive des bei ihrer Erzeugung verwendeten Zertifikates, sind dabei in das Dokument einzubetten.

Tabelle 182: TAB_KON_141 Fehlercodes „EncryptDocument“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4001	Security	Error	Interner Fehler

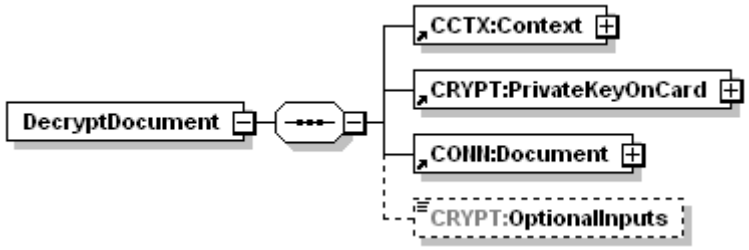
4058	Security	Error	Aufruf nicht zulässig

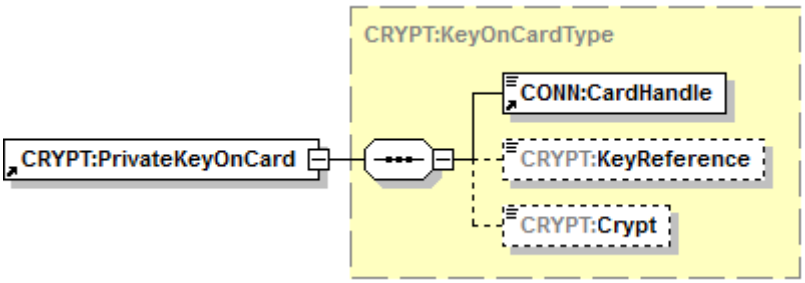
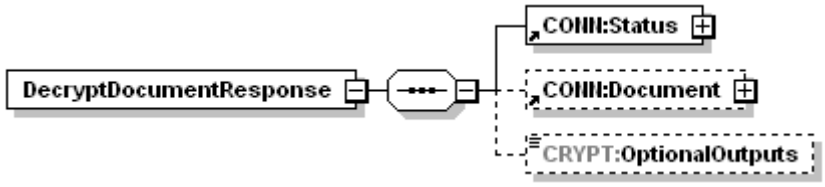
[<=]

4.1.7.5.2 DecryptDocument

TIP1-A_4622-02 - Operation DecryptDocument
Der Basisdienst Verschlüsselungsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation DecryptDocument anbieten.

Tabelle 183: TAB_KON_075 Operation DecryptDocument

<table>
<tr><td>Name</td><td colspan="2">DecryptDocument</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Die Operation entschlüsselt alle hybrid verschlüsselten Dokumente, die mit der Operation EncryptDocument erzeugt wurden. Es werden die Dokumententypen Alle_DocFormate unterstützt.
Für die Entschlüsselung wird ein asymmetrischer Schlüssel zu einem X.509v3-Zertifikat genutzt. Dieses Zertifikat und der Schlüssel müssen von einer Karte kommen.
Das bei der Entschlüsselung verwendete Kryptoverfahren (RSA oder ECC) wird durch den Hybridschlüssel bestimmt, der durch die Karte entschlüsselt werden soll. Sind sowohl RSA- als auch ECC-Hybridschlüssel für die referenzierte Karte vorhanden, versucht der Konnektor die Entschlüsselung des ECC-Hybridschlüssels, und wenn das nicht erfolgreich war, die Entschlüsselung des RSA-Hybridschlüssels.</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>Context</td><td>Aufrufkontext:
MandantId, ClientSystemId, WorkplaceId verpflichtend
UserId verpflichtend bei HBAx, bei SM-B nicht ausgewertet</td></tr>
</table>

<table>
<tr><td></td><td colspan="2"> 
</td></tr>
<tr><td></td><td>PrivateKeyOnCard</td><td>Identifiziert die zu verwendende Karte mit dem (privaten) Schlüssel.
Es werden die folgenden Karten unterstützt: HBAx und SM-B. Die Operation DecryptDocument DARF das Entschlüsseln mit der eGK NICHT unterstützen.</td></tr>
<tr><td></td><td>CardHandle</td><td>Identifiziert die gesteckte Karte.</td></tr>
<tr><td></td><td>KeyReference</td><td>Der Wert dieses Parameters ist in der Tabelle TAB_KON_747 KeyReference für Encrypt-/DecryptDocument spezifiziert. Ist der Parameter nicht angegeben, gilt der Default-Wert C.ENC.</td></tr>
<tr><td></td><td>Crypt</td><td>Ist nicht enthalten.</td></tr>
<tr><td></td><td>CONN:Document</td><td>Enthält das base64-codierte Dokument, das entschlüsselt werden soll.</td></tr>
<tr><td></td><td>CRYPT:OptionalInputs</td><td>Kann – in zukünftigen Versionen der Spezifikation – optionale Aufrufparameter enthalten.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Status</td><td>Enthält den Ausführungsstatus der Operation.</td></tr>
<tr><td></td><td>CRYPT:OptionalOutputs</td><td>Kann – in zukünftigen Versionen der Spezifikation – optionale Ausgabeparameter enthalten.</td></tr>
<tr><td></td><td>CONN:Document</td><td>Enthält das entschlüsselte Dokument in base64-codierter Form</td></tr>
<tr><td>Fehler</td><td colspan="2">Bei Auftreten eines Fehlers im Standardablauf werden Fehlercodes entsprechend TAB_KON_145 gemeldet.</td></tr>
</table>

Vorbe-dingungen	Keine
Nachbe-dingungen	Keine

Der Ablauf der Operation DecryptDocument ist in Tabelle TAB_KON_076 Ablauf DecryptDocument beschrieben:

Tabelle 184: TAB_KON_076 Ablauf DecryptDocument

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1. 2.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab. Wird die Operation für einen nicht unterstützten Kartentypen aufgerufen, so bricht die Operation mit Fehler 4058 ab.

2. 1.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über 026 {
		mandatId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		cardHandle = $context.cardHandle;
		userId = $context.userId }

4. 4.	TUC_KON_071 Daten hybrid entschlüsseln	Die Entschlüsselung wird durchgeführt. Im Fall eines XML-Dokuments mit mehreren verschlüsselten Elementen sind alle mit dem angegebenen Schlüssel entschlüsselbaren Elemente zu entschlüsseln.

Tabelle 185: TAB_KON_145 Fehlercodes „DecryptDocument“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4001	Security	Error	interner Fehler
4058	Security	Error	Aufruf nicht zulässig

[<=]

4.1.7.6 Betriebsaspekte

keine

4.1.8 Signaturdienst

Der Signaturdienst bietet Clientsystemen und Fachmodulen eine Schnittstelle zum Signieren von Dokumenten und Prüfen von Dokumentensignaturen

Innerhalb des Signaturdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): keine Events vorhanden
- Konfigurationsparameter: „SAK_"

4.1.8.1 Funktionsmerkmalweite Aspekte

4.1.8.1.1 Dokumentensignatur

Der Signaturdienst umfasst die Funktionalität der nicht-qualifizierten elektronischen Signatur (nonQES) mit der SM-B, sowie die qualifizierte elektronische Signatur (QES) mit dem HBA und den HBA-Vorläuferkarten HBA-qSig und ZOD_2.0 (=HBAx).

In der Abbildung fachlicher Abläufe kann es nötig sein, ein Dokument mehrfach parallel zu signieren, oder existierende Signaturen gegenzusignieren. Der Konnektor unterstützt **parallele Signaturen** (QES und nonQES). Ebenso unterstützt er Gegensignaturen (QES und nonQES), die jeweils alle bestehenden Signaturen gegensignieren. Die angebotene Möglichkeit des Gegensignierens bezieht sich dabei auf das Signieren aller vorhandenen parallelen Signaturen, während ein Gegensignieren von Gegensignaturen nicht angeboten wird. Der Konnektor unterstützt ausschließlich eine **dokumentexkludierende Gegensignatur**, bei der alle Signaturen gegensigniert werden, aber nicht der fachliche Inhalt des Dokumentes selbst.

TIP1-A_4623 - Unterstützte Signaturverfahren nonQES
Der Signaturdienst MUSS für die Erstellung und Prüfung von nicht-qualifizierten elektronischen Signaturen (nonQES) für die `nonQES_DocFormate` die Signaturverfahren entsprechend Tabelle TAB_KON_582 – Signaturverfahren unterstützen.
[<=]

TIP1-A_4627 - Unterstützte Signaturverfahren QES
Der Signaturdienst MUSS für die Erstellung und Prüfung von qualifizierten elektronischen Signaturen (QES) für die `QES_DocFormate` die Signaturverfahren entsprechend Tabelle TAB_KON_582 – Signaturverfahren unterstützen.
[<=]

Tabelle 186: TAB_KON_582 – Signaturverfahren Dokumentensignatur

Signaturformat	Standard	Dokumentformate	QES/ nonQES	Bemerkung
XMLDSig (XAdES)	[RFC3275] [XMLDSig]	XML	QES, nonQES	Hierdurch können abgesetzte (detached), umschließende

	[XAdES] [RFC6931]			(enveloping) und eingebettete (enveloped) Signaturen erzeugt werden.
CMS (CAdES)	[RFC5652] [CAdES]	QES_DocFormate nonQES_DocFormate	QES, nonQES	Hierdurch können abgesetzte (detached) und umschließende (enveloping) Signaturen erzeugt werden.
PDF/A (PAdES)	[PAdES-3]	PDF/A	QES, nonQES	Hierdurch können CMS-basierte Signaturen in PDF/A-Dokumente eingefügt und dadurch eingebettete Signaturen erzeugt werden.
S/MIME	[RFC5751]	nonQES_DocFormate	nonQES	Es werden MIME-Nachrichten signiert.

Zu den Begriffen detached, enveloping und enveloped Signaturen siehe beispielsweise auch [HüKo06#Abs. 4.3.3. und 4.3.1.5].

TIP1-A_5447 - Einsatzbereich der Signaturvarianten
Der Signaturdienst MUSS für die Erstellung und Prüfung von nicht-qualifizierten elektronischen Signaturen (nonQES) und qualifizierten elektronischen Signaturen (QES) die Vorgaben zum Einsatzbereich gemäß Tabelle TAB_KON_778 umsetzen.

Tabelle 187: TAB_KON_778 – Einsatzbereich der Signaturvarianten für XAdES, CAdES und PAdES

Signaturvarianten				Einsatzbereich		
Signatur verfahren	Signatur-variante	WAS wird signiert?	WO wird die Signatur abgelegt?	nonQES	QES Außen-schnittstelle	QES Fachmodul-schnittstelle
XAdES	detached	beliebiges (Binär)-Dokument	außerhalb des Dokuments in der SignResponse	Nein	Nein	Nein

XAdES	detached	gesamtes Input XML-Dokument (= Root-Element mit Subelementen)	außerhalb des Dokuments in der SignResponse	Nein	Nein	Nein
XAdES	detached	ausgewähltes nicht Root-Element mit Subelementen im Input XML-Dokument	außerhalb des Dokuments in der SignResponse	Nein	Nein	Nein
XAdES	detached	ausgewähltes nicht Root-Element mit Subelementen im Input XML-Dokument	Innerhalb des Dokuments, aber außerhalb des signierten Subbaums	Nein	Bedingt	Bedingt
XAdES	enveloped	gesamtes Input XML-Dokument (= Root-Element mit Subelementen)	Als direktes Child des Root-Elements	Ja	Bedingt	Bedingt
XAdES	enveloped	ausgewähltes nicht Root-Element mit Subelementen im Input XML-Dokument	Als direktes Child des ausgewählten Elements	Nein	Nein	Bedingt
XAdES	enveloping	gesamtes Input XML-Dokument (= Root-Element mit Subelementen)	Im Dokument, das Root-Element umschließend	Ja	Bedingt	Bedingt
XAdES	enveloping	ausgewähltes nicht Root-Element mit Subelementen im Input XML-Dokument	Im Dokument, das ausgewählte Element umschließend	Nein	Nein	Nein

CAdES	detached	gesamtes Binärdokument	außerhalb des Dokuments in der SignResponse	Ja	Ja	Ja
CAdES	envelopingg	gesamtes Binär-Dokument	innerhalb des CMS-Dokuments	Ja	Ja	Ja
PAdES	-	gesamtes PDF-Dokument	Im PDF-Dokument	Ja	Ja	Ja

Legende:
Ja: Die Signaturvariante ist für den Einsatzbereich erlaubt.
Nein: Die Signaturvariante ist für den Einsatzbereich nicht erlaubt.
Bedingt: Die Signaturvariante ist für den Einsatzbereich nicht erlaubt, es sei denn es wird durch eine im Konnektor integrierte Signaturrichtlinie explizit gefordert.
Die Spalten mit gelber Kopfzeile definieren die Signaturvarianten, die mit grauer, den Einsatzbereich. Beim Einsatzbereich wird zwischen nonQES und QES unterschieden und im Fall QES nach der Bereitstellung an der Außenschnittstelle oder intern für Fachmodule.
Die benötigten Signaturvarianten werden für XAdES über die Aufrufparameter IncludeObject und SignaturePlacement gemäß [OASIS-DSS] gesteuert.
Für CAdES erfolgt die Steuerung welche Signaturvariante gewählt wird, über den Aufrufparameter IncludeEContent.
[<=]

A_18756 - Optionalität von nonQES-XAdES Signatur
Der Konnektor KANN alle Aufrufe zu Signaturerstellung einer nonQES-XAdES Signatur mit Fehler 4111 und alle Aufrufe zur Signaturprüfung einer nonQES-XAdES Signatur mit Fehler 4112 beantworten. Die Signaturvarianten aus TAB_KON_778 werden damit weiter eingeschränkt. Wird die nonQES-XAdES Signatur umgesetzt, so ist diese in der Sicherheitszertifizierung zu betrachten.**[<=]**

TIP1-A_5402 - Baseline-Profilierung der AdES-EPES-Profile
Der Konnektor MUSS von den AdES-Profilen die AdES-EPES-Profile umsetzen, ergänzt um

- RevocationValues gemäß AdES-X-L,
- SignatureTimeStamp (für Signaturprüfung, nicht für Signaturerstellung) gemäß AdES-T

  Dabei MUSS der Konnektor die Baseline-Profilierung gemäß Kapitel 6 in [XAdES Baseline Profile] für XAdES, Kapitel 6 in [CAdES Baseline Profile] für CAdES und Kapitel 6 in [PAdES Baseline Profile] für PAdES umsetzen.

[<=]

Durch die Baseline-Profilierung der AdES-BES-Profile wird festgelegt, wie der Signaturzeitpunkt, gemessen als Systemzeit des Konnektors, in die Signatur eingebracht wird.

TIP1-A_5403 - Common PKI konforme Profile
Der Konnektor SOLL die signierten Dokumente konform zu [COMMON_PKI#Part 3] und [COMMON_PKI#Part 8] erstellen.
[<=]

TIP1-A_4624 - Default-Signaturverfahren nonQES
Bei fehlender expliziter Angabe durch den Aufrufer MUSS der Signaturdienst bei der Erstellung von nicht-qualifizierten elektronischen Signaturen (nonQES) die Default-Signaturverfahren entsprechend TAB_KON_583 Default-Signaturverfahren wählen.
[<=]

TIP1-A_4628 - Default-Signaturverfahren QES
Bei fehlender expliziter Angabe durch den Aufrufer MUSS der Signaturdienst bei der Erstellung von qualifizierten elektronischen Signaturen (QES) die Default-Signaturverfahren entsprechend TAB_KON_583 – Default-Signaturverfahren wählen.
[<=]

Tabelle 188: TAB_KON_583 – Default-Signaturverfahren

Dokument-Format	Signaturverfahren (und -variante)			
	Signaturverfahren	Signaturvariante	WAS wird signiert?	WO wird die Signatur abgelegt?
XML	XAdES	enveloped	gesamtes Input XML-Dokument (= Root-Element mit Subelementen)	als direktes Child des Root-Elements
PDF/A	PAdES	-	gesamtes PDF-Dokument	im PDF-Dokument
alle anderen	CAdES	detached	gesamtes Binärdokument	außerhalb des Dokuments in der SignResponse

TIP1-A_5387 - Erweiterte Nutzung der AdES-Profile
Der Konnektor MUSS auf eine vollständige Nutzung der AdES-Profile erweiterbar sein.
[<=]

TIP1-A_5033 - Missbrauchserkennung Signaturdienst (nonQES)
Der Konnektor MUSS zur Unterstützung von Missbrauchserkennungen die in Tabelle TAB_KON_584 gelisteten Operationen als Einträge in EVT_MONITOR_OPERATIONS berücksichtigen.

Tabelle 189: TAB_KON_584 nonQES-Operationen für EVT_MONITOR_OPERATIONS

Operationsname	OK_Val	NOK_Val	Alarmwert (Default-Grenzwert 10 Minuten-Σ)
SignDocument (nonQES)	1	5	41
VerifyDocument (nonQES)	1	5	61

[<=]

TIP1-A_4629 - Unterstützte Karten QES-Erstellung
Der Signaturdienst MUSS für die QES-Erstellung die Kartentypen HBA, HBA-qSig und ZOD_2.0 unterstützen.
[<=]

TIP1-A_5436 - XML Dokument nach Entfernen der Signatur unverändert
Der Konnektor MUSS die Operation SignDocument für XML-Dokumente so implementieren, dass das Dokument nach Entfernen der Signatur, insbesondere auch einer Teilsignatur, als Ganzes unverändert ist, wobei zwei XML-Dokumente als identisch zu betrachten sind, wenn sie gemäß Canonical XML 1.1 gleich sind [CanonXML1.1].
[<=]

TIP1-A_5682 - XML Nicht geeignete Algorithmen im VerificationReport
Der Konnektor MUSS im VerificationReport einer QES-Signaturprüfung ausweisen, wenn die für die Signatur verwendeten Algorithmen nach dem Algorithmenkatalog [ALGCAT] als nicht geeignet eingestuft werden.
[<=]

A_17768 - Zertifikate und Schlüssel für Signaturerstellung und Signaturprüfung (QES und nonQES)
Der Konnektor MUSS bei der Signaturerstellung und Signaturprüfung (QES und nonQES) die Zertifikate und Schlüssel gemäß den Vorgaben in TAB_KON_900 ermitteln.

Tabelle 190: TAB_KON_900 Zertifikate und private Schlüssel für Signaturerstellung und Signaturprüfung (QES und nonQES)

Karte	Crypt	Zertifikat (Verify)	Schlüssel (Sign)	Einsatzbereich	
				Außen-schnittstelle	Fachmodul-schnittstelle
QES		**...in DF.QES**			
HBA	RSA	EF.C.HP.QES.R2048	PrK.HP.QES.R2048	ja	ja
	ECC	EF.C.HP.QES.E256	PrK.HP.QES.E256	ja	ja
	RSA_ECC	**[**ab **G2.1]**: EF.C.HP.QES.E256 **[G2.0]**: EF.C.HP.QES.R2048	**[**ab **G2.1]**: PrK.HP.QES.E256 **[G2.0]**: PrK.HP.QES.R2048	ja	ja
HBA-VK	RSA	EF.C.HP.QES	PrK.HP.QES	ja	ja
nonQES		**...in DF.ESIGN**			
SM-B	RSA	EF.C.HCI.OSIG.R2048	PrK.HCI.OSIG.R2048	ja	ja
	ECC	EF.C.HCI.OSIG.E256	PrK.HCI.OSIG.E256	ja	ja

	RSA_ECC	**[**ab **G2.1]**: EF.C.HCI.OSIG.E256 **[G2.0]**: EF.C.HCI.OSIG.R2048	**[**ab **G2.1]**: PrK.HCI.OSIG.E256 **[G2.0]**: PrK.HCI.OSIG.R2048	ja	ja
eGK	RSA	EF.C.CH.AUT.R2048	PrK.CH.AUT.R2048	nein	ja
	ECC	EF.C.CH.AUT.E256	PrK.CH.AUT.E256	nein	ja
	RSA_ECC	**[**ab **G2.1]**: EF.C.CH.AUT.E256 **[G2.0]**: EF.C.CH.AUT.R2048	**[**ab **G2.1]**: PrK.CH.AUT.E256 **[G2.0]**: PrK.CH.AUT.R2048	nein	ja

[<=]

Tabelle 191: TAB_KON_862-01 Werteliste und Defaultwert des Parameters crypt bei QES-Erzeugung

Typname	Werteliste	Defaultwert	Bedeutung
SIG_CRYPT_QES	RSA ECC RSA_ECC	RSA	Werteliste des Parameters crypt bei der bei der Erzeugung einer QES-Signatur RSA: Es wird eine RSA-2048 Signatur erzeugt. ECC: Es wird eine ECC-256 Signatur erzeugt. RSA_ECC: In Abhängigkeit von der Kartengeneration wird eine RSA-2048 bzw. eine ECC-256 Signatur erzeugt (siehe TAB_KON_900).

Tabelle 192: TAB_KON_863 Werteliste und Defaultwert des Parameters crypt bei nonQES-Erzeugung

Typname	Werteliste	Defaultwert	Bedeutung
SIG_CRYPT_nonQES	RSA ECC RSA_ECC	RSA	Werteliste des Parameters crypt bei der bei der Erzeugung einer nonQES-Signatur RSA: Es wird eine RSA-2048 Signatur erzeugt. ECC: Es wird eine ECC-256 Signatur erzeugt. RSA_ECC: In Abhängigkeit von der Kartengeneration wird eine RSA-2048 bzw. eine ECC-256 Signatur erzeugt (siehe TAB_KON_900).

4.1.8.1.2 Signaturrichtlinien

Signaturrichtlinien dienen der Profilierung von Signaturerstellung und -prüfung. Beim Aufruf der Operation SignDocument kann eine URI übergeben werden, die eine im Konnektor hinterlegte Signaturrichtlinie referenziert. Die Plattform des Konnektors stellt selbst keine Signaturrichtlinien bereit. Fachanwendungen, die Signaturrichtlinien erfordern, definieren diese im Fachmodul des Konnektors. Für XML-Dokumentenformate aus der Menge von `QES_DocFormate` können die nachfolgenden Aspekte über eine Signaturrichtlinie gekapselt festgelegt werden:

- XML-Schemas für die Typkonformitätsprüfung (im Konnektor zu hinterlegen)
- Constraints für den Aufruf der Schnittstelle SignDocument und VerifyDocument, die zur Profilierung der Schnittstelle dienen.

TIP1-A_5538 - Signaturrichtlinien bei QES für XML-Dokumentenformate
Der Konnektor MUSS Signaturrichtlinien für XML-Dokumentenformate aus der Menge von `QES_DocFormate` bei die Signaturerstellung und -prüfung umsetzen.
Der Konnektor MUSS den für jede Signaturrichtlinie definierten Bezeichner (URI) bei der Signatur als SigPolicyId im Feld SignaturePolicyIdentifier einbetten. Bei der Signaturprüfung MUSS der Konnektor über eine etwaig vorhandene SigPolicyId die Signaturrrichtlinie identifizieren.
Die gemäß AdES erforderliche Hash-Referenz über die Policy (SigPolicyHash) MUSS Schema-konform leer gelassen werden. Bei der Signaturprüfung DARF die Hash-Referenz über die Policy NICHT geprüft werden.
[<=]

4.1.8.1.3 Signaturzeitpunkt

Bezogen auf den vom Konnektor für die Signaturprüfung anzunehmenden Signaturerstellungszeitpunkt werden in dieser Spezifikation die Bezeichner Ermittelter_Signaturzeitpunkt und Benutzerdefinierter_Zeitpunkt verwendet.

Ermittelter_Signaturzeitpunkt: Vom Konnektor ermittelter Zeitpunkt, zu dem eine Signatur geprüft wird. Es werden folgende Signaturzeitpunkte ermittelt:

1. `Ermittelter_Signaturzeitpunkt_Eingebettet`:
   in der Signatur eingebetteter Zeitpunkt (falls vorhanden)
2. `Ermittelter_Signaturzeitpunkt_System`:
   Systemzeit des Konnektors bei Signaturprüfung

Anmerkung: Bei vom Konnektor selbst erstellten Signaturen ist immer ein in der Signatur eingebetteter Zeitpunkt vorhanden, jedoch kein qualifizierter Zeitstempel, da in der TI keine qualifizierten Zeitstempel ausgestellt werden. Sollte ein Dokument mit einem qualifizierten Zeitstempel versehen sein, so wird dieser nicht für die Ermittlung des Signaturzeitpunktes herangezogen.

Benutzerdefinierter_Zeitpunkt: Vom Benutzer beim Aufruf der Signaturprüfoperation als Parameter an den Konnektor übergebener Zeitpunkt, zu dem eine Signatur geprüft werden soll.

4.1.8.1.4 Jobnummer

Da die eHealth-Kartenterminals dezentral über eine Netzwerkschnittstelle am Konnektor betrieben werden, fehlt die Möglichkeit zur direkten physischen und vom Anwender kontrollierbaren Zuordnung eines solchen Terminals zu einem Arbeitsplatz, auf dem sich das Clientsystem befindet.

Daher ist es bei einer fehlerhaften Zuordnung eines eHealth-Kartenterminals zu einem Arbeitsplatz möglich, dass die PIN-Eingabeaufforderung – beispielsweise zu einem Signaturauftrag – an ein entferntes Kartenterminal weitergeleitet wird. Diese fehlerhafte Zuordnung kann durch einen Fehler des Clientsystems oder den Versuch eines Angriffes hervorgerufen werden.

Die Jobnummern werden vom Konnektor erzeugt und können durch Clientsystem oder Signaturproxy abgerufen werden. Der Konnektor stellt jedoch keine Verbindung zwischen erzeugten und verwendeten Jobnummern her. Es wird also nicht geprüft, ob nur Jobnummern verwendet werden, die vorher vom Konnektor erzeugt wurden, oder ob alle Jobnummern verwendet werden, die vom Konnektor erzeugt wurden.

TIP1-A_4639 - Generierung von Jobnummern für PIN-Eingaben
Um Fehler- und Angriffsmöglichkeiten auszuschließen, MUSS der Konnektor bei bestimmten PIN-Verifikationen vor der Aufforderung zur PIN-Eingabe an einem eHealth-Kartenterminal eine hinreichend eindeutige Nummer – die Jobnummer – generieren, welche den Auftrag kennzeichnet, für dessen Verarbeitung die PIN-Eingabe erfolgen soll. Bei welchen PIN-Verifikationen dies der Fall ist, kann den PIN-Prompts in TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal entnommen werden.
[<=]

TIP1-A_4640 - Anzeige der Jobnummern für PIN-Eingaben
Diese Jobnummer MUSS vom Konnektor im Display des eHealth-Kartenterminals neben der PIN-Eingabeaufforderung angezeigt werden.
[<=]

TIP1-A_4992 - Guidance zur Jobnummer
Das Handbuch des Konnektors MUSS den Benutzer über den korrekten Gebrauch der Jobnummer informieren. Es MUSS ihm verdeutlichen, dass er seine PIN über die Tastatur des eHealth-Kartenterminals nur eingeben darf, wenn am Signaturproxy bzw. Primärsystem und am Display des Kartenterminals die gleiche Jobnummer angezeigt wird. Stimmen die beiden Nummern nicht überein, so soll der Benutzer seine PIN nicht eingeben und stattdessen weitergehende Schritte zur Klärung des aufgetretenen Fehlverhaltens einleiten.
[<=]

TIP1-A_4642 - Ableitung der Jobnummer von einem Zufallswert
Zur hinreichend eindeutigen Kennzeichnung des Vorganges MUSS eine Jobnummer von einem Zufallswert abgeleitet sein, wobei die Vorgaben an einen solchen Zufallswert beachtet werden MÜSSEN [gemSpec_Krypt#2.2].
[<=]

TIP1-A_4643 - Beschaffenheit der Jobnummer
Zur Wahrung der Benutzerfreundlichkeit MUSS eine Reduzierung der Jobnummer auf eine Länge von sechs Zeichen erfolgen. Diese sechs Zeichen MÜSSEN in zwei Zeichengruppen mit je drei Zeichen, getrennt durch einen Bindestrich (0x2D), dargestellt werden. Die erste Zeichengruppe MUSS ausschließlich die Zeichen "A-Z" beinhalten, die zweite Zeichengruppe MUSS aus Ziffern "0-9" bestehen. Die Länge der resultierenden, reduzierten Jobnummer ist sieben und wird durch den Umfang der darstellbaren Zeichen auf dem Display des eHealth-Kartenterminals beschränkt.
[<=]

TIP1-A_4644 - Jobnummer über 1.000 Vorgänge eindeutig
Der Konnektor MUSS die Eindeutigkeit einer Jobnummer sicherstellen:

- Bei Aufruf der Operation GetJobnumber MUSS der Konnektor innerhalb von 1000 Aufrufen eine eindeutige Jobnummer generieren. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext.

- Wird die Operation SignDocument mit einer Jobnummer aufgerufen, die innerhalb der vorangegangenen 1.000 Vorgänge verwendet wurde, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4252 abbrechen. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext.

[<=]

TIP1-A_4645 - Zeichen der Jobnummer
Die einzelnen Zeichen der Jobnummer MÜSSEN für die Anzeige am Kartenterminal gemäß dem Zeichensatz ISO 646DE/DIN66003, bzw. ISO 646 US codiert werden. Aus diesem Zeichensatz dürfen nur die Zeichen „A-Z" (0x41 bis 0x5A) und die Ziffern „0-9" (0x30 bis 0x39) für die Anzeige der Jobnummer verwendet werden.
[<=]

Beispiele für eine Jobnummer sind ABC-475 und HZF-696.

Die Einbettung der Jobnummer in den Nachrichtentext für den Bildschirm des Kartenlesers wird in TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal beschrieben.

4.1.8.1.5 Komfortsignatur

Für die QES unterstützt der Konnektor die Komfortsignaturfunktion. In diesem Modus können für ein- und denselben HBA mehrere vom Clientsystem initiierte Signaturaufträge (Einzel- oder Stapelsignatur) abgearbeitet werden, ohne dass der Inhaber des HBA für jeden einzelnen dieser Signaturaufträge die PIN.QES am Kartenterminal eingegeben muss.

Im Auslieferungszustand ist die Komfortsignaturfunktion ausgeschaltet (`SAK_COMFORT_SIGNATURE = Disabled`), d. h. mit dem Konnektor können zunächst keine Komfortsignaturen durchgeführt werden. Die Komfortsignaturfunktion kann vom Administrator eingeschaltet werden. Dies ist nur möglich, wenn an der Clientsystemschnittstelle des Konnektors verpflichtend TLS mit Clientauthentisierung (Konfigurationsvariante SOAP1 und SOAP2 in TAB_KON_852) konfiguriert ist. Das Einschalten der Komfortsignaturfunktion im Konnektor hat zur Folge, dass alle Operationen an der Clientsystemschnittstelle nur über TLS mit Clientauthentisierung angesprochen werden können (außer ggf. Dienstverzeichnisdienst).

Bei eingeschalteter Komfortsignaturfunktion können potentiell alle HBAs in der Umgebung, in der der Konnektor eingesetzt ist, Komfortsignaturen durchführen. Die eigentliche Aktivierung der Komfortsignatur muss separat für jeden einzelnen HBA erfolgen.

Durch Aufruf der Operation ActivateComfortSignature des Konnektors durch das Primärsystem wird die Nutzung der Komfortsignatur für einen HBA (Komfortsignaturmodus) aktiviert. Dazu muss der HBA-Inhaber die `PIN.QES` eingeben.

Der Konnektor merkt sich für die Cardsession des HBA, dass die Komfortsignatur aktiviert wurde. Bei den folgenden Aufrufen von `SignDocument`werden dann Komfortsignaturen ausgeführt, solange bis eines der folgenden Abbruchkriterien eintritt:

- Die vom HBA (entsprechend Personalisierung) oder die vom Konnektor (entsprechend Konfiguration `SAK_COMFORT_SIGNATURE_MAX`) durchgesetzte maximale Anzahl von Signaturen wurde erreicht.
- Das konfigurierte Zeitintervall für die Komfortsignatur (entsprechend Konfiguration `SAK_COMFORT_SIGNATURE_TIMER`) ist für die Cardsession abgelaufen.
- Der Komfortsignaturmodus wurde für die betroffene Cardsession deaktiviert.

- Der HBA wurde gezogen.
- Der Sicherheitszustand des HBA wurde zurückgesetzt.
- Die Komfortsignaturfunktion wurde für den Konnektor durch den Administrator deaktiviert.

A_19945 - Unterstützte Signaturvarianten bei Komfortsignatur
Der Signaturdienst MUSS bei der Komfortsignatur die Signaturvarianten für die QES gemäß TAB_KON_778 unterstützen.**[<=]**

A_18597 - Sicherheitszustand der PIN.QES bei Komfortsignatur
Bei der Komfortsignatur DARF der Konnektor den Sicherheitszustand der `PIN.QES` NICHT selbsttätig zurücksetzen, außer wenn dies explizit spezifikatorisch gefordert wird.**[<=]**

A_18597 kann z. B. umgesetzt werden, indem

- ein dedizierter logischer Kanal des HBA für die Komfortsignatur verwendet wird und
- im dedizierten logischen Kanal des HBA die Selektion von `DF.QES` solange beibehalten wird, bis ein Verlassen von `DF.QES` durch die Spezifikation explizit gefordert wird.

A_18686-01 - Komfortsignatur-Timer
Der Konnektor MUSS für jede HBA-Kartensitzung mit eingeschalteter Komfortsignatur einen Komfortsignatur-Timer gemäß konfiguriertem Zeitintervall `SAK_COMFORT_SIGNATURE_TIMER` einrichten.
Der Konnektor DARF nach Erreichen des Maximalwerts des Timers NICHT weitere Signaturaufträge annehmen.
Der Konnektor MUSS den Sicherheitszustand des HBA nach Erreichen des Maximalwertes des Timers zurücksetzen, nachdem Signaturaufträge, die bis zu diesem Zeitpunkt bereits zur Bearbeitung angenommen wurden, vollständig abgearbeitet wurden.
[<=]

A_19100 - Komfortsignatur-Zähler
Der Konnektor MUSS für jeden gesteckten HBA mit eingeschalteter Komfortsignatur die an die Karte gesendeten Signaturaufträge zählen und nach Erreichen des Maximalwerts den Sicherheitszustand des HBA zurücksetzen.**[<=]**

A_19258 - Secure Messaging bei Komfortsignatur
Bei der Komfortsignatur MUSS der Signaturdienst die zu signierenden Daten (DTBS) über Secure Messaging vom Konnektor zum HBA übertragen. Dieser Secure Messaging-Kanal MUSS über die gSMC-K zum HBA mittels C.SAK.AUTD_CVC aufgebaut werden.**[<=]**

A_20073-01 - Prüfung der Länge der UserId
Der Konnektor MUSS die beim Aktivieren des Komfortsignaturmodus vom PS übermittelte UserId für die Kartensitzung des HBA, für den der Modus aktiviert wird, auf die ausreichende Länge von 128 Bit im Format einer UUID nach RFC4122 prüfen und die Aktivierung mit Fehler 4272 ablehnen, wenn die UserId nicht ausreichend lang ist.**[<=]**

A_20074 - UserId über 1.000 Vorgänge eindeutig
Der Konnektor MUSS die Eindeutigkeit der UserId sicherstellen. Wird die Operation ActivateComfortSignature mit einer UserId im Aufrufkontext aufgerufen, die innerhalb der vorangegangenen 1.000 Vorgänge bereits verwendet wurde, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4270 abbrechen. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext.**[<=]**

A_19101 - Handbuch-Hinweis zu Nutzerauthentisierung am Clientsystem bei Komfortsignatur
Das Handbuch des Konnektors MUSS einen Hinweis enthalten, dass die Authentifizierung des HBA-Inhabers für die Komfortsignatur vom Clientsystem vorgenommen wird und dass die Authentifizierung des Nutzers am Clientsystem einen unverzichtbaren Beitrag zur Sicherheit der Lösung leistet.**[<=]**

4.1.8.2 Durch Ereignisse ausgelöste Reaktionen

keine

4.1.8.3 Interne TUCs, nicht durch Fachmodule nutzbar

Abbildung PIC_KON_103 Use Case Diagramm Signaturdienst (nonQES) beschreibt die Aufrufbeziehungen der nonQES-TUCs des Signaturdienstes. Die TUCs des Signaturdienstes sind weiß dargestellt. Genutzte TUCs anderer Basisdienste sind grau hinterlegt.

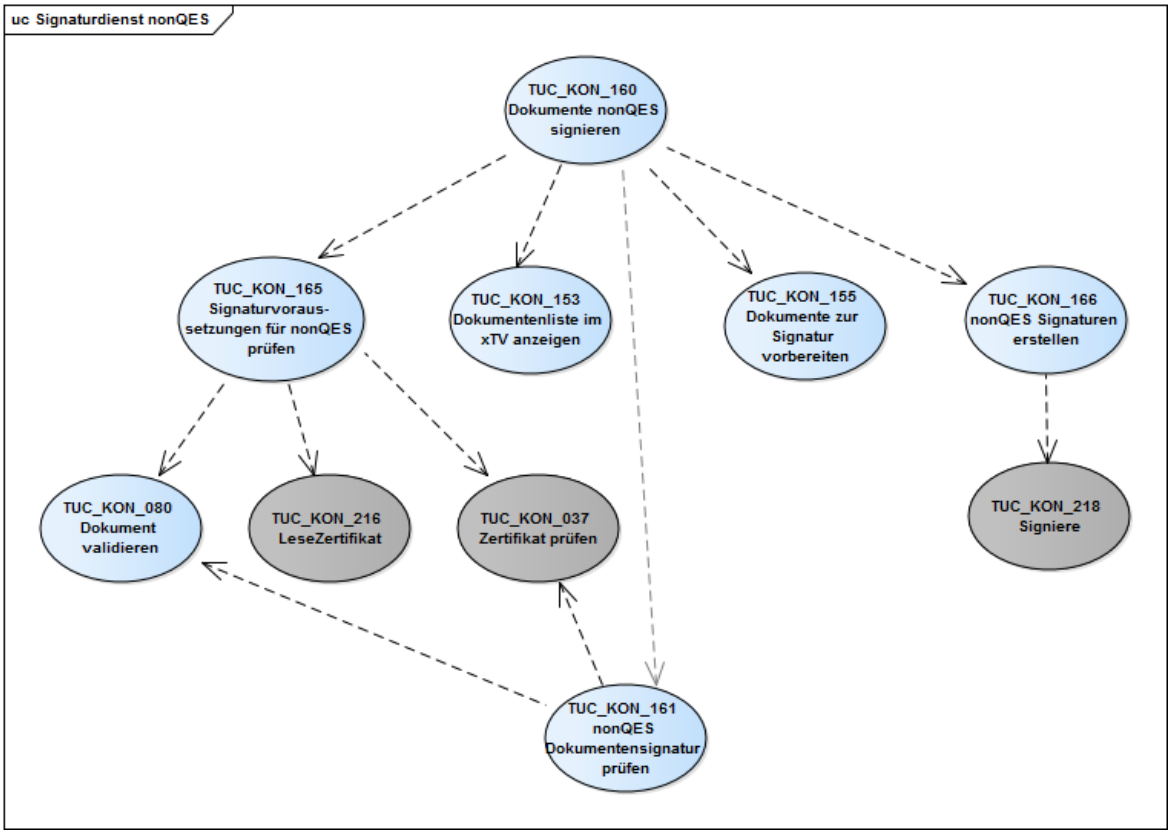


Abbildung 15: PIC_KON_103 Use Case Diagramm Signaturdienst (nonQES)

Abbildung PIC_KON_104 Use Case Diagramm Signaturdienst (QES) beschreibt die Aufrufbeziehungen der QES-TUCs des Signaturdienstes.

Abbildung 16: PIC_KON_104 Use Case Diagramm Signaturdienst (QES)

Abbildung PIC_KON_102 Use Case Diagramm Signaturdienst (Komfortsignatur) beschreibt die Aufrufbeziehungen der TUCs des Signaturdienstes für die Komfortsignatur.

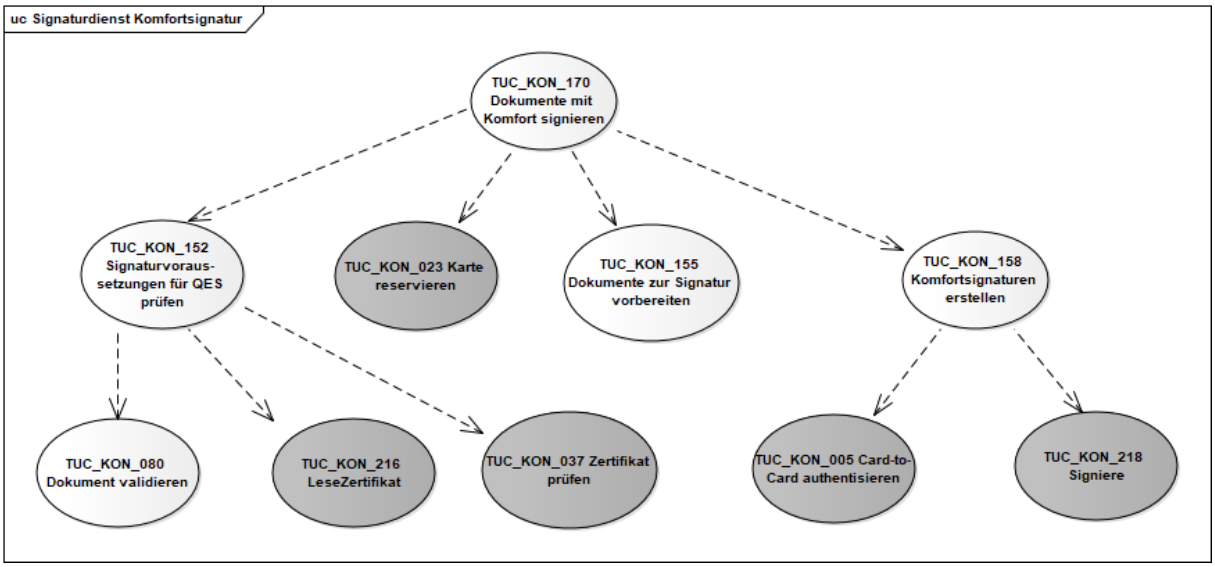


Abbildung 17: PIC_KON_102 Use Case Diagramm Signaturdienst (Komfortsignatur)

4.1.8.3.1 TUC_KON_155 „Dokumente zur Signatur vorbereiten"

TIP1-A_4646-02 - ab PTV4: TUC_KON_155 „Dokumente zur Signatur vorbereiten"

Der Konnektor MUSS den technischen Use Case TUC_KON_155 „Dokumente zur Signatur vorbereiten" umsetzen.

Tabelle 193: TAB_KON_748 - TUC_KON_155 „Dokumente zur Signatur vorbereiten"

Element	Beschreibung
Name	TUC_KON_155 "Dokumente zur Signatur vorbereiten"
Beschreibung	Die zu signierenden Dokumente werden entsprechend den Erfordernissen der Signaturverfahren für die QES oder nonQES vorbereitet.
Anwendungsumfeld	Erstellung von qualifizierten elektronischen Signaturen (QES) und nicht-qualifizierten elektronischen Signaturen (nonQES)
Auslöser	Aufruf durch TUC_KON_150 „Dokumente QES signieren" oder TUC_KON_160 „Dokumente nonQES signieren"
Vorbedingungen	keine
Eingangsdaten	- signatureMode (Signaturart: QES | nonQES) - documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente) und pro Dokument: - documentFormat (Formatangabe für das zu signierende Dokument) - optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur (siehe Operation SignDocument, Parameter dss:OptionalInputs), darin u.a. -signatureType (URI für den Signaturtyp XML-, CMS-, S/MIME-o PDF-Signatur) - certificate (Signaturzertifikat) - ocspResponses – *optional* (OCSP-Response des EE-Zertifikats, das bei der Signaturerstellung in die Signatur eingebettet wird.)
Komponenten	Konnektor
Ausgangsdaten	• preProcessedDocuments (Aufbereitetes zu signierendes Dokument bzw. aufbereitete zu signierende Dokumente)
Standardablauf	signatureType = XMLDSig (XAdES) Entsprechend den Regeln für die QES und die nonQES werden zunächst weitere Signatureigenschaften zum jeweiligen Dokument in Form von `QualifyingProperties` (siehe [XAdES]) hinzugefügt. Die Systemzeit des Anwendungskonnektors muss in das XML-Element

	`SigningTime` (siehe [XAdES]) eingetragen werden. Die Signatur wird anschließend entsprechend [XMLDSig] vorbereitet. D. h., es wird je Dokument nach Erzeugung der Reference Elemente das SignedInfo Element aufgebaut. Dessen Inhalt ergibt dann nach erfolgter XML-Kanonisierung und Hashing die DTBS (Data To Be Signed), die später zur Karte gesendet werden. certificate wird im Element `ds:KeyInfo/ds:X509Data` gespeichert. Im Fall signatureMode = QES können neben den reinen Nutzdaten auch alle weiteren Elemente in die Signatur einbezogen werden, die für die Rekonstruktion der ursprünglich dargestellten Daten in der sicheren Anzeige erforderlich sind. Für XML-Dokumente sind das, falls vorhanden, das/die XML-Schema(ta). Für diese werden Referenzen (Hash + URI) in die Signatur eingebettet. Die URI ist im Fall übergebener XML-Schemata der übergebene *signatureType* - Parameter. Die URI ist im Fall der im Konnektor im Rahmen einer Signaturrichtlinie hinterlegten XML-Schemata/XSL- Stylesheets die URI der Signaturrichtlinie, ergänzt um den Dateinamen mit Pfad, wie in der Signaturrichtlinie festgelegt. (Beispiel: URI für Schemadatei NFD_Document.xsd der Signaturrichtlinie SR_DF_NFDM_NOTFALLDATEN lautet: urn:gematik:fa:sak:nfdm:r1:v1:NFD_Document.xsd) Das Einbetten der Referenzen erfolgt über das XML-Element ds:object/ds:manifest (XMLDSig) mit eingebetteten XML-Elementen ds:Reference, die eine URI (RefURI) als Identifier für die jeweilige Datei und einen Hash über die jeweilige Resource enthalten. Der ShortTextClientsystem muss in die Signatur in das `DataObjectFormat/Description`-Element gemäß [XAdES] (Abschnitt 7.2.5) eingebettet werden. Falls durch den Aufrufparameter `SIG:IncludeRevocationInfo` angefordert, wird die für die Offline-Prüfung notwendige OCSP-Antwort im Sinne vom ES-X-L vom Konnektor in die Signatur eingebettet: Die base-64 kodierte OCSP-Response wird im Feld

	`QualifyingProperties/UnsignedProperties /UnsignedSignatureProperties/RevocationValues /OCSPValues/EncapsulatedOCSPValue` (selbst DER-kodiert) gespeichert. signatureType = CMS (CAdES) Etwaig einzubettende XML-Schemata werden zunächst wie für XAdES definiert in ein ds:manifest-Element eingebettet. Die so erzeugte Zeichenkette wird als genau ein ASN.1 Character String vom Typ UTF8String verpackt. Dieser wird als contentDescription in einen Content-Hints Attributwert vom Typ ContentHints verpackt, wobei der contentType=id-data gemäß [CAdES]. Der ShortTextClientsystem muss in die Signatur in das `content-hints.ContentDescription`-Attribut gemäß [CAdES] (Abschnitt 5.10.3) eingebettet werden. Ist die Einbettung von OCSP-Responses gefordert, wird die für die Offline-Prüfung notwendige OCSP-Antwort des EE-Zertifikats im Attribut SingedData.crls.other abgelegt. signatureType = PDF/A (PAdES) Der ShortTextClientsystem muss bei einer PDF-Signatur in das `Reason`-Feld eingebettet werden. OCSP-Responses werden bei PAdES nicht eingebettet. Es sind die Vorgaben zum Signaturprofil gemäß Tabelle TAB_KON_779 „Profilierung der Signaturformate“ zu erfüllen. Die aufbereiteten zu signierenden Dokumente werden an den Aufrufer zurückgegeben.
Varianten/ Alternativen	keine
Fehlerfälle	Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_586 Fehlercodes TUC_KON_155 „Dokumente zur Signatur vorbereiten“ „PDF/A (PAdES)“ Es ist nicht genügend Speicherplatz im PDF-Dokument verfügbar: 4205

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 194: TAB_KON_586 Fehlercodes TUC_KON_155 „Dokumente zur Signatur vorbereiten"

Fehlercode	ErrorType	Severity	Fehlertext
4205	Technical	Error	Es ist nicht genügend Speicherplatz im PDF-Dokument verfügbar.

[<=]

4.1.8.3.2 TUC_KON_165 „Signaturvoraussetzungen für nonQES prüfen"

TIP1-A_4647-02 - TUC_KON 165 „Signaturvoraussetzungen für nonQES prüfen"
Der Konnektor MUSS den technischen Use Case „Signaturvoraussetzungen für nonQES prüfen" umsetzen.

Tabelle 195: TAB_KON_749 – TUC_KON_165 „Signaturvoraussetzungen für nonQES prüfen"

Element	Beschreibung
Name	TUC_KON_165 „Signaturvoraussetzungen für nonQES prüfen"
Beschreibung	Es werden die Voraussetzungen an die zu signierenden Dokumente und das Signaturzertifikat geprüft. Es werden die `nonQES_DocFormate` unterstützt.
Auslöser	TUC_KON_160 „Dokumente nonQES signieren"
Vorbedingungen	keine
Eingangsdaten	• Zu signierende Dokumente • optionale Eingabeparameter zur Steuerung der Details der Signaturerstellung • cardSession Signaturkarte • zu verwendende Identität (Zertifikatsreferenz)
Komponenten	Konnektor, Kartenterminal, Signaturkarte
Ausgangsdaten	• Prüfergebnis • Signaturzertifikat
Standardablauf	1. Abhängig von seinem Typ werden für jedes Dokument die typabhängigen Validierungsschritte (ohne Prüfung auf sichere Anzeigbarkeit) durchgeführt. Dies geschieht durch Aufruf von TUC_KON_080 „Dokument validieren".

	Wird der Aufruf von TUC_KON_080 mit einem Fehler beendet, wird die Prüfung im laufenden TUC mit diesem Fehler abgebrochen. 2. Durch Aufruf von TUC_KON_216 „Lese Zertifikat" wird das Signaturzertifikat von der Signaturkarte gelesen. 3. Das Signaturzertifikat wird durch Aufruf von TUC_KON_037 „Zertifikat prüfen"{ certificate = Zertifikatsreferenz; qualifiedCheck = not_required; offlineAllowNoCheck = true; validationMode = OCSP} geprüft.
Varianten/Alternativen	Keine
Fehlerfälle	Keine
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 196: TAB_KON_587 Fehlercodes TUC_KON_165 „Signaturvoraussetzungen für nonQES prüfen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können keine weiteren Fehlercodes auftreten.			

[<=]

4.1.8.3.3 TUC_KON_166 „nonQES Signaturen erstellen"

TIP1-A_4648 - TUC_KON_166 „nonQES Signaturen erstellen"
Der Konnektor MUSS den technischen Use Case TUC_KON_166 „nonQES Signaturen erstellen" umsetzen.

Tabelle 197: TAB_KON_750 – TUC_KON_166 „nonQES Signaturen erstellen"

Element	Beschreibung
Name	TUC_KON_166 „nonQES Signaturen erstellen"
Beschreibung	Die Data To Be Signed (DTBS) werden erzeugt, an die Signaturkarte gesendet und dort signiert.
Auslöser	TUC_KON_160 „Dokumente nonQES signieren"
Vorbedingungen	keine
Eingangsdaten	• Liste der zu signierenden Dokumente • cardSession Signaturkarte • zu verwendende Identität (Zertifikatsreferenz)

	• crypt [SIG_CRYPT_nonQES]: *optional*; *default und* Wertebereich: SIG_CRYPT_DEFAULT siehe TAB_KON_863 (Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden.)
Komponenten	Konnektor, Kartenterminal, Signaturkarte
Ausgangsdaten	• Signierte Dokumente
Standardablauf	Die folgenden Schritte werden für jedes Dokument der Liste durchgeführt. 1. Das zu signierende Dokument wird, soweit noch nicht erfolgt, für die XML-Signatur vorbereitet. Ein Ergebnis dieser Vorbereitung sind die Data To Be Signed (DTBS): der Hash-Wert (Digest des SignedInfo-Elementes), der zur Signatur an die Karte gesendet werden soll. Für XML-Signaturen müssen die Vorgaben aus [gemSpec_Krypt#3.1.1] beachtet werden. 2. Für das zu signierende Dokument werden die DTBS zur Signatur an die Signaturkarte übermittelt (Aufruf von TUC_KON_218 „Signiere“). Dabei werden der Schlüssel und der Algorithmusidentifier über die Tabelle TAB_KON_900 bestimmt. 3. Die erstellte Signatur wird mathematisch geprüft. 4. Der ermittelte Signaturwert wird in die zuvor vorbereitete XML-Signatur eingefügt. 5. Der Konnektor löst TUC_KON_256 {"SIG/SIGNDOC/NEXT_SUCCESSFUL"; Op; Info; „$Jobnummer"; noLog; $doInformClients } aus.
Varianten/Alternativen	keine
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (→3) Fehlgeschlagene mathematische Prüfung der Signatur: 4120
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 198: TAB_KON_120 Fehlercodes TUC_KON_166 „nonQES Signaturen erstellen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4120	Security	Error	Kartenfehler

[<=]

4.1.8.3.4 TUC_KON_152 "Signaturvoraussetzungen für QES prüfen"

TIP1-A_4649 - TUC_KON_152 „Signaturvoraussetzungen für QES prüfen"
Der Konnektor MUSS den technischen Use Case TUC_KON_152 „Signaturvoraussetzungen für QES prüfen" umsetzen.

Tabelle 199: TAB_KON_751 – TUC_KON_152 „Signaturvoraussetzungen für QES prüfen"

Element	Beschreibung
Name	TUC_KON_152 „Signaturvoraussetzungen für QES prüfen"
Beschreibung	Es werden die Voraussetzungen an die zu signierenden Dokumente und das Signaturzertifikat geprüft. Es werden die `QES_DocFormate` unterstützt.
Auslöser	TUC_KON_150 „Dokumente QES signieren"
Vorbedingungen	keine
Eingangsdaten	• Zu signierende Dokumente • optionale Eingabeparameter zur Steuerung der Details der Signaturerstellung • cardSession Signaturkarte • zu verwendende Identität (Zertifikatsreferenz) • includeRevocationInfo [Boolean] - optional; Default: true (Dieser Parameter steuert die Einbettung von OCSP-Responses in die Signatur. true: Die Sperrinformationen werden in ocspResponses zurückgegeben.)
Komponenten	Konnektor, Kartenterminal, Signaturkarte
Ausgangsdaten	• Prüfergebnis • Signaturzertifikat • ocspResponses - optional/nur wenn includeRevocationInfo = true (OCSP-Response des EE-Zertifikats, die beim Aufruf von TUC_KON_037 „Zertifikat prüfen" zurückgegeben wird)
Standardablauf	1. Abhängig von seinem Typ werden für jedes Dokument die typabhängigen Dokumentvalidierungsschritte durchgeführt (Aufruf TUC_KON_080 „Dokument validieren"). Wird der Aufruf von TUC_KON_080 mit einem Fehler beendet, wird die Prüfung im laufenden TUC mit diesem Fehler abgebrochen.

	2. Durch Aufruf von TUC_KON_216 „Lese Zertifikat“ wird das Signaturzertifikat von der Signaturkarte gelesen.
	3. Das Signaturzertifikat wird durch Aufruf von TUC_KON_037 „Zertifikat prüfen“{
	certificate = Zertifikatsreferenz;
	qualifiedCheck = required;
	offlineAllowNoCheck = true;
	validationMode = OCSP;
	getOCSPResponses = includeRevocationInfo}
	geprüft.

Varianten/Alternativen	keine
Fehlerfälle	(->3) Für MGM_LU_ONLINE=Enabled gilt: Liefert die Zertifikatsprüfung (OCSP-Abfrage) die Warnung CERT_REVOKED oder CERT_UNKNOWN gemäß [gemSpec_PKI#Tab_PKI_274], dann wird der TUC mit Fehler 4123 abgebrochen.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 200: TAB_KON_588 Fehlercodes TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können keine weiteren Fehlercodes auftreten.			

[<=]

4.1.8.3.5 TUC_KON_154 "QES Signaturen erstellen"

Der TUC_KON_154 stellt den Standardsignaturfall in der TI, die Stapelsignatur dar (auch für Stapel der Größe 1). Da die Stapelsignatur auf der Zielkarte passende CVC voraussetzt, die auf den HBA-Vorläuferkarten nicht vorhanden sind, kann dieser TUC nur den HBA unterstützen. Für HBA-Vorläuferkarten kann TUC_KON_168 verwendet werden.

TIP1-A_4651-02 - TUC_KON_154 „QES Signaturen erstellen“
Der Konnektor MUSS den technischen Use Case TUC_KON_154 „QES Signaturen erstellen” umsetzen.

Tabelle 201: TAB_KON_752 – TUC_KON_154 „QES Signaturen erstellen"

Element	Beschreibung
Name	TUC_KON_154 „QES Signaturen erstellen“

Beschreibung	Die Data To Be Signed (DTBS) werden erzeugt, an die Signaturkarte gesendet und dort signiert.
Auslöser	TUC_KON_150 „Dokumente QES signieren“
Vorbedingungen	Die Ressourcen Signaturkarte und Kartenterminal sind für den Vorgang reserviert. DF.QES ist selektiert. PIN.QES ist initial verifiziert
Eingangsdaten	• Zu signierendes Dokument bzw. zu signierende Dokumente • cardSession (nur HBA erlaubt) • zu verwendende Identität (Zertifikatsreferenz) • crypt [SIG_CRYPT_QES] - *optional*; *default und* Werte*bereich*: siehe TAB_KON_862-01 (Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden.) • WorkplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBA)
Ausgangsdaten	• Signierte Dokumente

Standardablauf	Basierend auf SAK.AUTD_CVC und HPC.AUTD_SUK_CVC und den zugehörigen privaten Schlüsseln wird ein sicherer Kanal zwischen der gSMC-K des Konnektors und dem HBA aufgebaut mittels Aufruf TUC_KON_005 „Card-to-Card authentisieren“ {
	sourceCardSession = gSMC-K;
	targetCardSession = CardSession;
	authMode = „gegenseitig+TC“}
	
	Die folgenden Schritte werden für jedes Dokument des Stapels durchgeführt.
	1. Das zu signierende Dokument wird, soweit noch nicht erfolgt, für die Signatur gemäß des entsprechenden Formats vorbereitet. Ein Ergebnis dieser Vorbereitung sind die Data To Be Signed (DTBS): der Hash-Wert (Digest des SignedInfo-Elementes), der zur Signatur an die Karte gesendet werden soll.
	2. Für das zu signierende Dokument werden die DTBS zur Signatur im sicheren Kanal an den HBA übermittelt (Aufruf TUC_KON_218 „Signiere“). Dabei werden der Schlüssel und der Algorithmusidentifier über die Tabelle TAB_KON_900 bestimmt.
	3. Falls Schritt 3 fehlgeschlagen ist, weil der PIN.QES-Nutzungszähler abgelaufen ist (erkennbar z. B. daran, dass die Karte einen Autorisierungsfehler zurückmeldet), wird die

	PIN.QES verifiziert (Aufruf TUC_KON_012 „PIN verifizieren“, nachdem der im Konnektor verwaltetete Sicherheitszustand (CARDSESSION.AUTHSTATE) aktualisiert wurde). Am Display des Kartenterminals wird dabei die Jobnummer für den Signaturvorgang angezeigt. Aus der WorkplaceId geht hervor, ob es sich um eine Remote-PIN-Eingabe handelt. Nach der PIN-Verifikation wird erneut die zuvor fehlgeschlagene Signatur in Schritt 3 ausgeführt. 4. Die erstellte Signatur wird mathematisch geprüft. 5. Der ermittelte Signaturwert wird in den zuvor vorbereiteten Signaturprototypen eingefügt. 6. Der Konnektor löst TUC_KON_256 {"SIG/SIGNDOC/NEXT_SUCCESSFUL"; Op; Info; „$Jobnummer"; noLog; $doInformClients } aus.
Varianten/ Alternativen	Alternativ zum Standardablauf kann zu Beginn die maximal erlaubte Stapelgröße SSEC durch Auslesen von EF.SSEC ermittelt werden. Der zu signierende Dokumentenstapel wird in Teilstapel von maximaler Größe SSEC zerlegt. Für jeden Teilstapel wird die PIN.QES verifiziert. Die Dokumente des Teilstapels werden wie im Standardablauf beschrieben signiert. Der Nutzer kann den Vorgang der PIN-Eingabe abbrechen.
Fehlerfälle	(->2) Fehler im Signaturvorgang führen zum Abbruch des gesamten Signaturvorgangs, Fehlercode 4123 (->3) Fehler bei der PIN-Eingabe führen zum Abbruch des Signaturvorgangs (->4) Fehler in mathematischer Prüfung der Signatur führen zum Abbruch des Signaturvorgangs, Fehlercode 4120 Das Verhalten des TUCs bei einem Fehlerfall, einem Timeout der PIN-Eingabe oder beim Abbruch durch den Benutzer ist in Tabelle TAB_KON_192 Verhalten des Konnektors beim Abbruch einer Stapelsignatur beschrieben.
Sicherheitsanforderungen	Zum Aufbau des sicheren Kanals bzw. zur Aushandlung des symmetrischen Schlüssels DARF DF.QES NICHT verlassen werden. Benötigte CVCs des HBA MÜSSEN also bereits vor dem Signaturvorgang eingelesen und gecacht werden. Dies KANN bereits beim Stecken des HBA geschehen. Die in [gemSpec_Krypt#3.1.2] angegebenen Festlegungen der zu unterstützenden Algorithmen MÜSSEN berücksichtigt werden.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	Abbildung PIC_KON_113 Aktivitätsdiagramm zu „QES Signaturen erstellen“

	Das Diagramm dient nur der Veranschaulichung und ist nicht vollständig. Beispielsweise enthält es nicht die Steuerung durch den Parameter crypt.

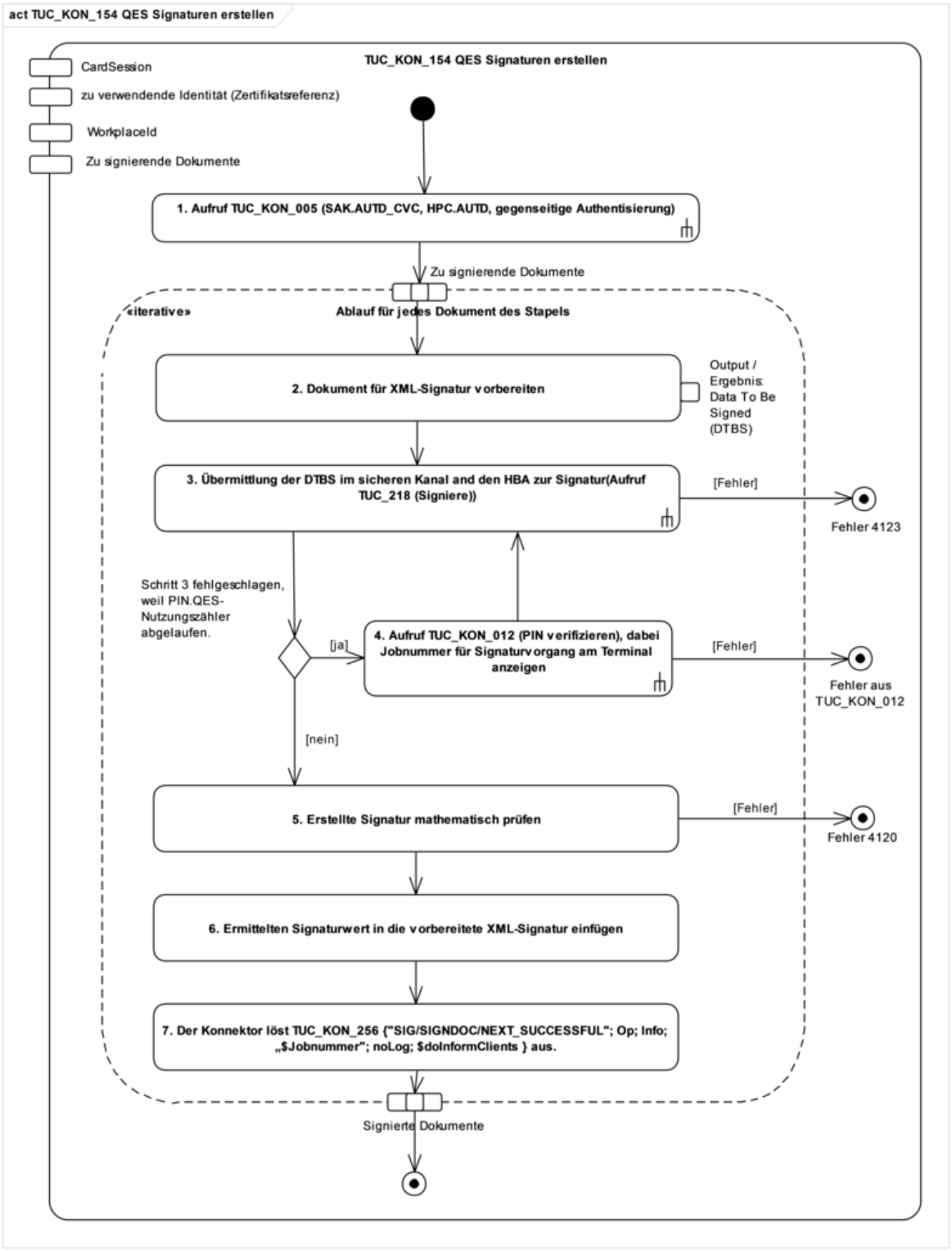


Abbildung 18: PIC_KON_113 Aktivitätsdiagramm zu „QES Signaturen erstellen"

Tabelle 202: TAB_KON_126 Fehlercodes TUC_KON_154 „QES Signaturen erstellen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4120	Security	Error	Kartenfehler
4123	Security	Error	Fehler bei Signaturerstellung

[<=]

4.1.8.3.6 TUC_KON_168 „Einzelsignatur QES erstellen“

TIP1-A_4652-02 - TUC_KON_168 „Einzelsignatur QES erstellen“
Der Konnektor MUSS den technischen Use Case TUC_KON_168 „Einzelsignatur QES erstellen“ umsetzen.

Tabelle 203: TAB_KON_293 - TUC_KON_168 „Einzelsignatur QES erstellen“

Element	Beschreibung
Name	TUC_KON_168 ”Einzelsignatur QES erstellen”
Beschreibung	Es wird ein Dokument technisch mit einer Signatur versehen. Im Gegensatz zum TUC_KON_154 „QES Signaturen erstellen“ wird hier nur eine einzelne Signatur ohne vorhergehendes C2C erstellt. Die Übertragung der DTBS erfolgt ohne Secure Messaging.
Auslöser	TUC_KON_150 Dokumente QES signieren
Vorbedingungen	Die Ressourcen Signaturkarte und Kartenterminal sind für den Vorgang reserviert. DF.QES ist selektiert.
Eingangsdaten	• zu signierendes Dokument • CardSession (HBAx) • zu verwendende Identität (Zertifikatsreferenz) • crypt: [SIG_CRYPT_QES] - *optional*; *default und* Werte*bereich*: siehe TAB_KON_862-01 Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden. • WorkplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBAx)
Ausgangsdaten	• Signiertes Dokument
Standardablauf	1. Das zu signierende Dokument wird, soweit noch nicht erfolgt, für die Signatur vorbereitet. Ein Ergebnis dieser Vorbereitung sind die DTBS: der Hash-Wert (Digest des

	SignedInfo-Elementes), der zur Signatur an die Karte gesendet werden soll. 2. Für das zu signierende Dokument werden die DTBS zur Signatur an den HBAx übermittelt (Aufruf TUC_KON_218 „Signiere“). Dabei werden der Schlüssel und der Algorithmusidentifier über die Tabelle TAB_KON_900 bestimmt. Jeder Fehler führt zum Abbruch des Signaturvorgangs 3. Die erstellte Signatur wird mathematisch geprüft. Der ermittelte Signaturwert wird in den zuvor gemäß des entsprechenden Signaturformates vorbereiteten Signaturprototypen eingefügt.
Varianten/ Alternativen	keine
Fehlerfälle	Das Verhalten des TUCs bei einem Fehlerfall, einem Timeout der PIN-Eingabe oder beim Abbruch durch den Benutzer ist in Tabelle TAB_KON_192 Verhalten des Konnektors beim Abbruch einer Stapelsignatur beschrieben. (→3) Fehler in mathematischer Prüfung der Signatur: Abbruch mit 4120
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 204: TAB_KON_590 Fehlercodes TUC_KON_168 „Einzelsignatur QES erstellen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten.			
4120	Security	Error	Kartenfehler

[<=]

4.1.8.3.7 TUC_KON_158 "Komfortsignaturen erstellen"

Der TUC_KON_158 führt die Komfortsignatur für ein Dokument oder mehrere Dokumente eines Stapels aus. Da die Komfortsignatur auf der Zielkarte passende CVC voraussetzt, die auf den HBA-Vorläuferkarten nicht vorhanden sind, unterstützt dieser TUC nur den HBA.

A_19102-04 - TUC_KON_158 „Komfortsignaturen erstellen“
Der Konnektor MUSS den technischen Use Case TUC_KON_158 „Komfortsignaturen erstellen” umsetzen.

Tabelle 205: TAB_KON_870 – TUC_KON_158 „Komfortsignaturen erstellen"

Element	Beschreibung
Name	TUC_KON_158 „Komfortsignaturen erstellen"
Beschreibung	Die Data To Be Signed (DTBS) werden erzeugt, an die Signaturkarte gesendet und dort signiert. Die Übertragung der DTBS erfolgt mit Secure Messaging. Die Abarbeitung der Signatur erfolgt im SE#2.
Auslöser	TUC_KON_170 „Dokumente mit Komfort signieren"
Vorbedingungen	Die Ressource Signaturkarte ist für den Vorgang reserviert. DF.QES ist selektiert. PIN.QES ist initial verifiziert
Eingangsdaten	• Zu signierendes Dokument bzw. zu signierende Dokumente • cardSession (nur HBA erlaubt) • zu verwendende Identität (Zertifikatsreferenz) • crypt [SIG_CRYPT_QES] - *optional*; *default und* Werte*bereich*: siehe TAB_KON_862-01 (Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden.) • WorkplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBA)
Ausgangsdaten	• Signierte Dokumente

Standardablauf	1. Wenn noch nicht erfolgt, wird basierend auf SAK.AUTD_CVC und HPC.AUTD_SUK_CVC und den zugehörigen privaten Schlüsseln ein sicherer Kanal zwischen der gSMC-K des Konnektors und dem HBA aufgebaut mittels Aufruf TUC_KON_005 „Card-to-Card authentisieren" {
	sourceCardSession = gSMC-K;
	targetCardSession = CardSession;
	authMode = „gegenseitig+TC"}
	
	Die folgenden Schritte werden für jedes Dokument des Stapels durchgeführt.
	2. Das zu signierende Dokument wird, soweit noch nicht erfolgt, für die Signatur gemäß des entsprechenden Formats vorbereitet. Ein Ergebnis dieser Vorbereitung sind die Data To Be Signed (DTBS): der Hash-Wert (Digest des SignedInfo-Elementes), der zur Signatur an die Karte gesendet werden soll.
	3. Es wird geprüft, ob der Komfortsignatur-Zähler der cardSession den Wert SAK_COMFORT_SIGNATURE_MAX überschritten hat .

	4. Für das zu signierende Dokument werden die DTBS zur Signatur im sicheren Kanal an den HBA übermittelt (Aufruf TUC_KON_218 „Signiere"). Dabei werden der Schlüssel und der Algorithmusidentifier über die Tabelle TAB_KON_900 bestimmt. 5. Der Komfortsignatur-Zähler der cardSession wird um 1 erhöht. 6. Die erstellte Signatur wird mathematisch geprüft. 7. Der ermittelte Signaturwert wird in den zuvor vorbereiteten Signaturprototypen eingefügt. 8. Der Konnektor löst TUC_KON_256 {"SIG/SIGNDOC/NEXT_SUCCESSFUL"; Op; Info; „$Jobnummer"; noLog; $doInformClients } aus.
Varianten/ Alternatien	Keine
Fehlerfälle	In den Fehlerfällen, die zum Abbruch des Komfortsignaturmodus mit Fehlercode 4271 führen, wird vor dem Abbruch TUC_KON_172 für das cardHandle des HBA ausgeführt. (->3) Der Komfortsignatur-Zähler der cardSession hat den Maximalwert überschritten: Fehlercode 4271 (->4) Der PIN.QES-Nutzungszähler der Karte ist abgelaufen (erkennbar z. B. daran, dass die Karte einen Autorisierungsfehler zurückmeldet): Fehlercode 4271 (->4) Fehler im Signaturvorgang führen zum Abbruch des gesamten Signaturvorgangs: Fehlercode 4123 (->6) Fehler in mathematischer Prüfung der Signatur führen zum Abbruch des Signaturvorgangs: Fehlercode 4120 Das weitere Verhalten des TUCs bei einem Fehlerfall oder beim Abbruch durch den Benutzer ist in TAB_KON_192, Verhalten des Konnektors beim Abbruch einer Stapelsignatur, beschrieben.
Sicherheitsanforderungen	Zum Aufbau des sicheren Kanals bzw. zur Aushandlung des symmetrischen Schlüssels DARF DF.QES NICHT verlassen werden. Benötigte CVCs des HBA MÜSSEN also bereits vor dem Signaturvorgang eingelesen und gecached werden. Dies KANN bereits beim Stecken des HBA geschehen. Komfortsignaturen MÜSSEN im SE#2 abgearbeitet werden. Die in [gemSpec_Krypt] angegebenen Festlegungen der zu unterstützenden Algorithmen MÜSSEN berücksichtigt werden.

Tabelle 206: TAB_KON_873 Fehlercodes TUC_KON_158 „Komfortsignaturen erstellen"

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4120	Security	Error	Kartenfehler
4123	Security	Error	Fehler bei Signaturerstellung

[<=]

4.1.8.4 Interne TUCs, auch durch Fachmodule nutzbar

A_20478 - Zusätzliche Dokumentformate für nonQES-Signatur
Der Konnektor KANN für die nonQES-Signaturerstellung an der Schnittstelle zu Fachmodulen zusätzliche Dokumentformate unterstützen.**[<=]**

Die in der obigen Anforderung benannten Signaturen von Dokumentenformaten umfassen beispielsweise die Signatur von Token nach SAML2 für das Fachmodul ePA entsprechend [gemSpec_FM_ePA#A_14927].

4.1.8.4.1 TUC_KON_160 „Dokumente nonQES signieren"

TIP1-A_4653 - TUC_KON_160 „Dokumente nonQES signieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_160 „Dokumente nonQES signieren" umsetzen.

Tabelle 207: TAB_KON_753 – TUC_KON_160 „Dokumente nonQES signieren"

Element	Beschreibung
Name	TUC_KON_160 „Dokumente nonQES signieren"
Beschreibung	Im Rahmen von Fachanwendungen werden ein oder mehrere Dokumente mit einer nicht-qualifizierten elektronischen Signatur (nonQES) versehen. Es werden die nonQES_DocFormate unterstützt.
Auslöser	Aufruf durch ein Clientsystem (Operation SignDocument) oder ein Fachmodul.
Vorbedingungen	Die Signaturkarte muss gesteckt sein.
Eingangsdaten	• cardSession (Kartensitzung; zulässig sind SM-B, oder bei Aufruf durch Fachmodul auch zusätzlich eGK) • signRequests (Liste von Signaturaufträgen.) Jeder Signaturauftrag (SignRequest) kapselt:

	• documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente); darin u.a. documentFormat (Formatangabe für das zu signierende Dokument) • optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur, siehe Operation SignDocument, Parameter dss:OptionalInputs); darin u.a. signatureType (URI für den Signaturtyp XML-, CMS-, S/MIME-, PDF-Signatur) • includeRevocationInfo: – *optional; default: true* (Dieser optionale Parameter steuert die Einbettung von OCSP-Antworten in die Signatur: nur wirksam bei der Prüfung von enthaltenen Parallelsignaturen, wenn eine Gegensignatur erstellt werden soll. Die OCSP-Antworten werden in die jeweils geprüfte Parallelsignatur eingebettet.) • workplaceId (Identifikator des Arbeitsplatzes)
Komponenten	Konnektor, Kartenterminal, Signaturkarte bzw. HSM-B
Ausgangsdaten	• signedDocuments (Liste der signierten Dokumente)
Standardablauf	Der Konnektor KANN die Schritte 1 bis 4 in einer beliebigen Reihenfolge durchführen. 1. Der signatureType und die Signaturvariante werden für jedes Dokument der Liste entsprechend SignatureType und SignatureVariant festgelegt. Wenn signatureType oder SignatureVariant nicht übergeben wurden (als Element von optionalInputs), wird das dem dokumentformat entsprechende Default-Verfahren gewählt (siehe TAB_KON_583 – Default-Signaturverfahren). 2. Für alle Dokumente des Stapels wird die Zulässigkeit des Kartentyps geprüft. Das für die Signatur zu nutzende Zertifikat wird anhand des Kartentyps implizit ausgewählt. 3. Es werden die Voraussetzungen für die Signatur geprüft. Dies erfolgt durch Aufruf von TUC_KON_165 „Signaturvoraussetzungen für nonQES prüfen". 4. Zum Vorbereiten der Dokumente für die Signatur wird TUC_KON_155 „Dokumente zur Signatur vorbereiten" mit ocspResponses aufgerufen.

	5. Die Signaturen werden durch den Aufruf von TUC_KON_166 erstellt. 6. Die signierten Dokumente werden an den Aufrufer zurückgegeben.
Varianten/ Alternativen	Im Fall signatureType=S/MIME-Signatur wird der Standardablauf des CMS Signaturverfahrens durch einen vorgelagerten S/MIME-Vorbereitungsschritt und einen nachgelagerten S/MIME-Nachbereitungsschritt ergänzt. Das S/MIME-Verfahren MUSS konform [S/MIME] und SOLL konform [COMMON_PKI], Part 3, erfolgen. Der S/MIME-Vorbereitungsschritt bereitet das übergebene MIME-Dokument gemäß [S/MIME], Kapitel 3.1, auf die nachfolgende CMS-Signatur durch eine Kanonisierung für Text [S/MIME], Kapitel 3.1.1, vor. Eine weitere Kanonisierung oder eine Anpassung des Transfer Encodings [S/MIME], Kapitel 3.1.2, erfolgt nicht. Im S/MIME-Nachbereitungsschritt wird das im Standardablauf erzeugte CMS-Objekt in eine MIME-Nachricht vom Typ „application/pkcs7-mime“ eingebettet. Sämtliche Header-Felder der Nachricht MÜSSEN in die Header-Felder der S/MIME-Nachricht übernommen werden. "MIME-Version: 1.0" MUSS definiert sein. Das Feld "Content-Type:" ist als "application/pkcs7-mime" zu definieren. Die weiteren Attribute dieses Feldes sind: • "smime-type=signed-data;" • "name=$dateiname", wobei $dateiname auf ".p7m" endet. Die Codierung des signierten Inhalts der Nachricht MUSS in "base64" erfolgen. Entsprechend ist das zugehörige Header-Feld zu füllen: "Content-Transfer-Encoding: base64". Das Feld "Content-Disposition" definiert den Inhalt der Nachricht als Dateianhang: "Content-Disposition: attachment; filename=$dateiname"
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (→2) Ungültige Angabe des Signaturverfahrens: Fehlercode 4111 Übergabe eines für die nonQES nicht unterstützten Dokumentformats: Fehlercode 4110 (→3) Kartentyp nicht zulässig für Signatur: Fehlercode 4126
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 208: TAB_KON_127 Fehlercodes TUC_KON_160 „Dokumente nonQES signieren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4110	Technical	Error	ungültiges Dokumentformat (%Format%) Der Parameter Format enthält das übergebene Dokumentformat.
4111	Technical	Error	ungültiger Signaturtyp oder Signaturvariante
4126	Security	Error	Kartentyp nicht zulässig für Signatur

[<=]

TIP1-A_4653-02 - ab PTV4: TUC_KON_160 „Dokumente nonQES signieren“
Der Konnektor MUSS den technischen Use Case TUC_KON_160 „Dokumente nonQES signieren” umsetzen.

Tabelle 209: TAB_KON_753 – TUC_KON_160 „Dokumente nonQES signieren“

Element	Beschreibung
Name	TUC_KON_160 „Dokumente nonQES signieren“
Beschreibung	Im Rahmen von Fachanwendungen werden ein oder mehrere Dokumente mit einer nicht-qualifizierten elektronischen Signatur (nonQES) versehen. Es werden die nonQES_DocFormate unterstützt.
Auslöser	Aufruf durch ein Clientsystem (Operation SignDocument) oder ein Fachmodul.
Vorbedingungen	Die Signaturkarte muss gesteckt sein.
Eingangsdaten	• cardSession (Kartensitzung; zulässig sind SM-B, oder bei Aufruf durch Fachmodul auch zusätzlich eGK) • crypt [SIG_CRYPT_nonQES] - *optional*; default und Wertebereich: siehe TAB_KON_863 Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden. • signRequests (Liste von Signaturaufträgen. Jeder Signaturauftrag (SignRequest) kapselt: • documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente); darin u.a.

	documentFormat (Formatangabe für das zu signierende Dokument) • optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur, siehe Operation SignDocument, Parameter dss:OptionalInputs); darin u.a. signatureType (URI für den Signaturtyp XML-, CMS-, S/MIME-, PDF-Signatur) • includeRevocationInfo: – *optional; default: true* (Dieser optionale Parameter steuert die Einbettung von OCSP-Antworten in die Signatur: nur wirksam bei der Prüfung von enthaltenen Parallelsignaturen, wenn eine Gegensignatur erstellt werden soll. Die OCSP-Antworten werden in die jeweils geprüfte Parallelsignatur eingebettet.) • workplaceId (Identifikator des Arbeitsplatzes)
Komponenten	Konnektor, Kartenterminal, Signaturkarte bzw. HSM-B
Ausgangsdaten	• signedDocuments (Liste der signierten Dokumente)
Standardablauf	Der Konnektor KANN die Schritte 1 bis 4 in einer beliebigen Reihenfolge durchführen. 1. Der signatureType und die Signaturvariante werden für jedes Dokument der Liste entsprechend SignatureType und SignatureVariant festgelegt. Wenn signatureType oder SignatureVariant nicht übergeben wurden (als Element von optionalInputs), wird das dem dokumentformat entsprechende Default-Verfahren gewählt (siehe TAB_KON_583 – Default-Signaturverfahren). 2. Für alle Dokumente des Stapels wird die Zulässigkeit des Kartentyps geprüft. Das für die Signatur zu nutzende Zertifikat wird anhand des Kartentyps und des Parameters crypt ausgewählt. 3. Es werden die Voraussetzungen für die Signatur geprüft. Dies erfolgt durch Aufruf von TUC_KON_165 „Signaturvoraussetzungen für nonQES prüfen". 4. Zum Vorbereiten der Dokumente für die Signatur wird TUC_KON_155 „Dokumente zur Signatur vorbereiten" mit ocspResponses aufgerufen. 5. Die Signaturen werden durch den Aufruf von TUC_KON_166 erstellt. 6. Die signierten Dokumente werden an den Aufrufer zurückgegeben.

Varianten/ Alternativen	Im Fall signatureType=S/MIME-Signatur wird der Standardablauf des CMS Signaturverfahrens durch einen vorgelagerten S/MIME-Vorbereitungsschritt und einen nachgelagerten S/MIME-Nachbereitungsschritt ergänzt. Das S/MIME-Verfahren MUSS konform [S/MIME] und SOLL konform [COMMON_PKI], Part 3, erfolgen. Der S/MIME-Vorbereitungsschritt bereitet das übergebene MIME-Dokument gemäß [S/MIME], Kapitel 3.1, auf die nachfolgende CMS-Signatur durch eine Kanonisierung für Text [S/MIME], Kapitel 3.1.1, vor. Eine weitere Kanonisierung oder eine Anpassung des Transfer Encodings [S/MIME], Kapitel 3.1.2, erfolgt nicht. Im S/MIME-Nachbereitungsschritt wird das im Standardablauf erzeugte CMS-Objekt in eine MIME-Nachricht vom Typ „application/pkcs7-mime“ eingebettet. Sämtliche Header-Felder der Nachricht MÜSSEN in die Header-Felder der S/MIME-Nachricht übernommen werden. "MIME-Version: 1.0" MUSS definiert sein. Das Feld "Content-Type:" ist als "application/pkcs7-mime" zu definieren. Die weiteren Attribute dieses Feldes sind: • "smime-type=signed-data;" • "name=$dateiname", wobei $dateiname auf ".p7m" endet. Die Codierung des signierten Inhalts der Nachricht MUSS in "base64" erfolgen. Entsprechend ist das zugehörige Header-Feld zu füllen: "Content-Transfer-Encoding: base64". Das Feld "Content-Disposition" definiert den Inhalt der Nachricht als Dateianhang: "Content-Disposition: attachment; filename=$dateiname"
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (→2) Ungültige Angabe des Signaturverfahrens: Fehlercode 4111 Übergabe eines für die nonQES nicht unterstützten Dokumentformats: Fehlercode 4110 (→3) Kartentyp nicht zulässig für Signatur: Fehlercode 4126
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 210: TAB_KON_127 Fehlercodes TUC_KON_160 „Dokumente nonQES signieren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			

4110	Technical	Error	ungültiges Dokumentformat (%Format%) Der Parameter Format enthält das übergebene Dokumentformat.
4111	Technical	Error	ungültiger Signaturtyp oder Signaturvariante
4126	Security	Error	Kartentyp nicht zulässig für Signatur

Die zulässigen Zertifikate und Schlüssel sind in TAB_KON_900 aufgelistet.**[**<=**]**

4.1.8.4.2 TUC_KON_161 „nonQES Dokumentsignatur prüfen "

TIP1-A_4654-03 - TUC_KON_161 „nonQES Dokumentsignatur prüfen"
Der Konnektor MUSS den technischen Use Case TUC_KON_161 „nonQES Dokumentsignatur prüfen" umsetzen.

Tabelle 211: TAB_KON_121 - TUC_KON_161 „nonQES Dokumentsignatur prüfen"

Element	Beschreibung
Name	TUC_KON_161 „nonQES Dokumentsignatur prüfen"
Beschreibung	Es wird die nicht-qualifizierte elektronische Signatur (nonQES) eines Dokuments geprüft. Dabei werden die Signaturverfahren laut Tabelle TAB_KON_582 – Signaturverfahren unterstützt. Sind mehrere Signaturen vorhanden, so werden alle geprüft. Auch Parallel- und Gegensignaturen MÜSSEN unterstützt werden.
Auslöser	Aufruf durch ein Clientsystem (Operation VerifyDocument) oder ein Fachmodul
Vorbedingungen	keine
Eingangsdaten	• signedDocument (Signiertes Document vom Typ `nonQES_DocFormate`) • signature – *optional/falls detached Signatur* (Signatur. Es werden Parallel- und Gegensignaturen unterstützt.) • optionalInputs (optionale Eingabeparameter, siehe Operation VerifyDocument, Parameter SIG:OptionalInputs) • certificate – *optional/verpflichtend, wenn das Zertifikat nicht im signierten Dokument enthalten ist* (X.509-Zertifikat, gegen das die Signatur geprüft werden soll) ocspGracePeriod (OCSP-Grace Period: maximal zulässiger Zeitraum, den die letzte OCSP-Antwort aus dem Cache bezüglich des Referenzzeitpunkts zurückliegen darf)

	• xmlSchemas – *optional/nur für XML-Dokumente* (XMLSchema und ggf. weitere vom Hauptschema benutzte Schemata) • includeRevocationInfo: – *optional; Default = false* (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur)
Komponenten	Konnektor
Ausgangsdaten	• verificationResult [VerificationResult] (Ergebnis der Signaturprüfung) • optionalOutput – *optional* (weitere Ausgabedaten gemäß SIG:OptionalOutput)
Standardablauf	1. **„DocumentValidation"**: Falls die Signatur im Dokument eingebettet ist, wird das signierte Dokument validiert durch Aufruf TUC_KON_080 „Dokument validieren" { CheckDisplayability=false; … } Treten dabei Fehler bei Validierung der Typkonformität auf, wird die Prüfung mit einem Fehler abgebrochen. 2. **„CoreValidation"**: Es erfolgt die mathematische Prüfung der Signatur bestehend aus der Prüfung der Hash-Kette bis zum signierten Hashwert und der Prüfung der kryptographischen Signatur unter Verwendung des öffentlichen Schlüssels, des Signaturwertes und des signierten Hashwertes. XML-Signatur: Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation. CMS-Signatur: Die Core Validation erfolgt entsprechend Cryptographic Message Syntax (CMS) Kapitel 5.6 Signature Verification Process [RFC5652]. PDF-Signatur: Die Core Validation erfolgt entsprechend [PAdES-3] Kapitel 4.6 Signature Validation aus PAdES-BES Part 3. Auch wenn die Validierung fehlschlägt, werden die folgenden Prüfschritte durchgeführt, so dass ein vollständiges Prüfprotokoll erstellt werden kann. 3. **„CheckSignatureCertificate"**: **Teil 1: Signaturzertifikat ermitteln** XML-Signatur: Das Signaturzertifikat ist im XMLDSig Element `ds:KeyInfo/ds:X509Data` gespeichert [XMLDSig] oder wird als Eingangsparameter übergeben.

	CMS-Signatur: Das Signaturzertifikat für CAdES ist im Feld `certificates` im `SignedData` Container gespeichert [CAdES] oder wird als Eingangsparameter übergeben. PDF-Signatur: Das PDF Signaturzertifikat für PAdES ist im Feld SignedData.certificates entsprechend Kapitel 6.1.1 „Placements of the signing certificate" [PAdES Baseline Profile] gespeichert oder wird als Eingangsparameter übergeben. **Teil 2: Signaturzeitpunkt bestimmen** Der Signaturzeitpunkt Ermittelter_Signaturzeitpunkt_Eingebettet wird wie folgt selektiert: XML-Signatur: Das XML element SigningTime spezifiziert den Signaturzeitpunkt entsprechend Kapitel 7.2.1 XAdES [XAdES]. CMS-Signatur: Das Attribut SigningTime spezifiziert den Signaturzeitpunkt entsprechend Kapitel 11.3 CMS [CMS]. PDF-Signatur: Der Signaturzeitpunkt kann dem M Eintrag des Signature Dictionary entnommen werden [PAdES Baseline Profile] Kapitel 6.2.1 Signing time. Der Signaturzeitpunkt `Benutzerdefinierter_Zeitpunkt` liegt gegebenenfalls als Aufrufparameter vor. Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_System` wird ermittelt. **Teil 3: Signaturzertifikatsprüfung:** Bei der folgenden Signaturzertifikatsprüfung sind die Signaturzeitpunkte gemäß [TIP1-A_5545] zu berücksichtigen. Die Signaturzertifikatsprüfung erfolgt durch Aufruf von TUC_KON_037 „Zertifikat prüfen", und zwar: Wenn es sich um das X.509-Zertifikat einer eGK handelt (PolicyList = oid_egk_aut bzw. oid_egk_autn), dann: TUC_KON_037 „Zertifikat prüfen" {

	certificate;
	qualifiedCheck = not_required;
	baseTime = Signaturzeitpunkt;
	offlineAllowNoCheck = true;
	policyList = [oid_egk_aut | oid_egk_autn];
	intendedKeyUsage=
	intendedKeyUsage(C.CH.AUT|C.CH.AUTN);
	intendedExtendedKeyUsage = id-kp-clientAuth;
	ocspResponses = OCSP-Response;
	gracePeriod = ocspGracePeriod;
	validationMode = OCSP;
	getOCSPResponses = includeRevocationInfo }
	Wenn es ein X.509-Zertifikat der SM-B ist (PolicyList = oid_smc_b_osig), dann:
	TUC_KON_037 „Zertifikat prüfen“ {
	certificate;
	qualifiedCheck = not_required;
	baseTime = Signaturzeitpunkt;
	offlineAllowNoCheck = true;
	policyList = oid_smc_b_osig;
	intendedKeyUsage = intendedKeyUsage(C.HCI.OSIG);
	ocspResponses = OCSP-Response;
	gracePeriod = ocspGracePeriod;
	validationMode = OCSP ;
	getOCSPResponses = includeRevocationInfo }
	Sind OCSP-Responses in der Signatur eingebettet, ist die jüngste OCSP-Response, die für die Zertifikatsprüfung notwendig ist, beim Aufruf von TUC_KON_037 zu übergeben.
	Sofern der Aufruf von TUC_KON_037 ocspResponsesRenewed zurückgibt, wird die Liste der OCSP-Responses in die Signatur eingebettet.
	Auch wenn die Zertifikatsprüfung fehlschlägt, werden die folgenden Prüfungen durchgeführt.
	4. **“CheckPolicyConstraints“**
	In diesem Schritt wird das signierte Dokument entsprechend der Profilierung der Signaturformate (siehe Anhang B.2) geprüft.
	Es sind die Vorgaben für die Prüfung von Signaturen aus den Standards für AdES [XAdES], [XAdES Baseline], [CAdES], [CAdES Baseline], [PAdES-3] und [PAdES Baseline] umzusetzen. Dabei sind die Vorgaben aus Tabelle TAB_KON_779 „Profilierung der Signaturformate“ und Tabelle TAB_KON_778 „Einsatzbereich der Signaturvarianten“ zu erfüllen.
	Auch wenn nicht alle Anforderungen an das Format des signierten Dokuments erfüllt werden, wird die Prüfung

	mit den folgenden Schritten fortgesetzt, um ein vollständiges Prüfungsprotokoll zu erhalten. 5. Das **Prüfergebnis** (verificationResult, optionalOutput wird an den Aufrufer zurückgegeben (siehe TAB_KON_754 Übersicht Status für Prüfung einer Dokumentensignatur).
Varianten/ Alternativen	Im Fall, dass die Online-Prüfung des Sperrzustands des Signaturzertifikats nicht möglich ist und eine möglicherweise gecachte OCSP-Response nicht vorhanden ist oder nicht mehr verwendet werden darf, wird das Prüfergebnis mit der entsprechenden Warnung zurückgegeben. Im Fall einer PKCS#1-Signatur ist das verwendete Signaturverfahren, RSASSA-PSS bzw. RSASSA-PKCS1-v1_5, aus der Signatur zu bestimmen.
Fehlerfälle	Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_124 Fehlercodes TUC_KON_161 „nonQES Dokumentensignatur prüfen" beschrieben. (->1) keine Signatur in signedDocument und signature vorhanden: 4253 (→2 **„CoreValidation"**) Interner Fehler: 4001, Signatur des Dokument ungültig: 4115. Signatur umfasst nicht das gesamte Dokument: 4262. (→3 „**CheckSignatureCertificate"**) Interner Fehler: 4001, Signaturzertifikat ermitteln fehlgeschlagen: 4206. (→4 **„CheckPolicyConstraints"**) Interner Fehler: 4001, Dokument nicht konform zu Regeln für nonQES: 4112.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 212: TAB_KON_124 Fehlercodes TUC_KON_161 „nonQES Dokumentensignatur prüfen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten.			
4001	Technical	Error	Interner Fehler
4206	Technical	Error	Signaturzertifikat ermitteln ist fehlgeschlagen
4112	Technical	Error	Dokument nicht konform zu Regeln für nonQES

4115	Security	Error	Signatur des Dokuments ungültig. Der SignatureValue des Dokuments ist falsch oder für mindestens eine Reference ist der DigestValue falsch.
4253	Technical	Error	Keine Signatur im Aufruf
4262	Technical	Error	Signatur umfasst nicht das gesamte Dokument
4264	Technical	Warning	Ein oder mehrere Zertifikate ignoriert

Das Gesamtergebnis (VerificationResult) für die Prüfung einer Dokumentensignatur fasst die Ergebnisse aller Prüfungsschritte in einem einzelnen Statuswert zusammen.

Tabelle 213: TAB_KON_754 Übersicht Status für Prüfung einer Dokumentensignatur

VerificationResult für gesamtes Dokument (VerificationResult/HighLevelResult)	
Wert	**Bedeutung**
VALID	Wenn VerificationResult für alle Signaturen zum Dokument VALID
INVALID	Wenn VerificationResult für eine Signatur zum Dokument INVALID
INCONCLUSIVE	in allen anderen Fällen
VerificationResult pro Signatur (VerificationReport/IndividualReport/Result)	
Wert	**Bedeutung** **mögliche Ausprägungen im VerificationReport**
VALID	Die Signatur wurde gemäß den Regeln für die nonQES geprüft und für gültig befunden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:HasManifestResults
INVALID	Die Signatur ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.

	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidSignatureTimestamp
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:ResponderError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:CertificateChainNotComplete
INCONCLUSIVE	Die Signatur wurde gemäß den Regeln für die nonQES geprüft. Allerdings konnten eine oder mehrere Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail. Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:OcspNotAvailiable Hinweis: Das Erreichen dieses Zustandes hängt davon ab, ob eine OCSP-Abfrage nicht durchgeführt werden konnte, unabhängig davon, ob die Ursache dafür die Offlineschaltung des Konnektors (MGM_LU_ONLINE = Disabled) oder die Nichterreichbarkeit des OCSP-Responders im Online-Betrieb (MGM_LU_ONLINE = Enabled) ist.

[<=]

TIP1-A_5545 - nonQES-Signaturprüfergebnis bezogen auf Signaturzeitpunkt
Der Konnektor MUSS zur nonQES-Signaturprüfung ein Prüfergebnis das sich auf genau einen angenommenen Signaturzeitpunkt bezieht, an den Aufrufer zurückgeben.
Die Auswahl des angenommenen Signaturzeitpunkts, auf den sich das Signaturergebnis bezieht, erfolgt hierarchisch:

- `Benutzerdefinierter_Zeitpunkt`
  falls vorhanden, sonst
- `Ermittelter_Signaturzeitpunkt_Eingebettet`
  falls vorhanden, sonst
- `Ermittelter_Signaturzeitpunkt_System`

[<=]

4.1.8.4.3 TUC_KON_162 „Kryptographische Prüfung der XML-Dokumentensignatur"

TIP1-A_5505 - TUC_KON_162 „Kryptographische Prüfung der XML-Dokumentensignatur"
Der Konnektor MUSS den technischen Use Case TUC_KON_162 „Kryptographische Prüfung der XML-Dokumentensignatur" umsetzen.

Tabelle 214: TAB_KON_430 – TUC_KON_162 „Kryptographische Prüfung der XML-Dokumentensignatur"

Element	Beschreibung
Name	TUC_KON_162 „Kryptographische Prüfung der XML-Dokumentensignatur"
Beschreibung	Es wird die mathematische Korrektheit der elektronischen Signatur eines XML-Dokuments geprüft. Sind mehrere Signaturen vorhanden, so werden alle geprüft.
Auslöser	Aufruf durch ein Fachmodul
Vorbedingungen	• signedDocument ist ein XML-Dokument • signedDocument hat TUC_KON_080 erfolgreich durchlaufen
Eingangsdaten	• signedDocument – *optional* (QES-signiertes XML-Dokument -> siehe Definition in Operation VerifyDocument mit `SIG:Document`) • signatureObject– *optional* (-> siehe Definition in Operation VerifyDocument mit `dss:SignatureObject`)
Komponenten	Konnektor
Ausgangsdaten	• result (Ergebnis der Signaturprüfung)
Standardablauf	**„CoreValidation"**: Es erfolgt die mathematische Prüfung der Signatur, bestehend aus der Prüfung der Hash-Kette bis zum signierten Hashwert und der Prüfung der kryptographischen Signatur unter Verwendung des öffentlichen Schlüssels aus dem Zertifikat, des Signaturwertes und des signierten Hashwertes. XML-Signatur: Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation. a) CoreValidation erfolgreich -> result = true b) CoreValidation fehlerhaft -> result = false
Varianten/Alternativen	keine

Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zu den ausgewiesenen Fehlercodes: Interner Fehler: 4001
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 215: TAB_KON_431 Fehlercodes TUC_KON_162 „Kryptographische Prüfung der XML-Dokumentensignatur"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten.			
4001	Technical	Error	Interner Fehler

[<=]

4.1.8.4.4 TUC_KON_150 „Dokumente QES signieren"

TIP1-A_4655-02 - TUC_KON_150 „Dokument QES signieren„
Der Konnektor MUSS den technischen Use Case TUC_KON_150 „Dokumente QES signieren" umsetzen.

Tabelle 216: TAB_KON_755 – TUC_KON_150 „Dokumente QES signieren"

Element	Beschreibung
Name	TUC_KON_150 "Dokumente QES signieren"
Beschreibung	Im Rahmen von Fachanwendungen werden ein oder mehrere Dokumente mit einer qualifizierten elektronischen Signatur versehen. Es werden die QES_DocFormate unterstützt.
Auslöser	Aufruf durch ein Clientsystem (Operation SignDocument) oder ein Fachmodul.
Vorbedingungen	Die Signaturkarte muss gesteckt sein.
Eingangsdaten	• signRequests (Liste von Signaturaufträgen) Jeder Signaturauftrag (SignRequest) kapselt: • documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente); darin u.a. documentFormat (Formatangabe für das zu signierende Dokument) • optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur, siehe

	Operation SignDocument, Parameter dss:OptionalInputs); darin u.a. signatureType (URI für den Signaturtyp XML-, CMS-, S/MIME-, PDF-Signatur) • includeRevocationInfo [Boolean]: – optional; Default: true (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur; siehe Operation SignDocument, Parameter SIG:IncludeRevocationInfo) • cardSession (Kartensitzung. Unterstützte Kartentypen: HBAx) • crypt [SIG_CRYPT_QES] - *optional*; default und Wertebereich: siehe TAB_KON_862-01 (Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden.) • workplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBAx)
Ausgangsdaten	• signedDocuments (Liste der signierten Dokumente)
Standardablauf	Der Konnektor KANN die Schritte 1 bis 4 in einer beliebigen Reihenfolge durchführen. 1. Der Signaturtyp und die Signaturvariante werden für jedes Dokument der Liste entsprechend signatureType und SignatureVariant festgelegt (ggf. in optionalInputs enthalten). Wenn SignatureType oder SignatureVariant nicht übergeben wurden, wird das dem Dokumentformat entsprechende Default-Verfahren gewählt (siehe TAB_KON_583 – Default-Signaturverfahren). 2. Für alle Dokumente des Stapels wird die Zulässigkeit des Kartentyps geprüft. Das für die Signatur zu nutzende Zertifikat wird anhand des Kartentyps und des Parameters crypt ausgewählt. 3. Es werden die Voraussetzungen für die Signatur geprüft. Dies erfolgt im TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“. Wenn includeRevocationInfo=true, dann setze ocspResponses auf Rückgabewert von TUC_KON_152. 4. Die am Signaturvorgang beteiligten Ressourcen (Signaturkarte sowie PIN Pad und Display des PIN-Eingabe-Kartenterminals) werden für die exklusive Nutzung durch diesen Signaturvorgang reserviert. Die Reservierung der Signaturkarte erfolgt durch Aufruf von TUC_KON_023 „Karte reservieren“ { cardSession; doLock = true }.

	5. Zum Vorbereiten der Dokumente für die Signatur wird TUC_KON_155 „Dokumente zur Signatur vorbereiten" mit ocspResponses aufgerufen. Die Zugriffe auf die Signaturkarte in den Schritten 6 bis 7 müssen im DF.QES erfolgen. 6. Die Signaturerstellung wird durch den Anwender autorisiert. Dies erfolgt durch Aufruf von TUC_KON_012 „PIN verifizieren" { cardSession; workplaceId; pinRef = PIN.QES; verificationType = Mandatorisch } Wenn nur ein zu signierendes Dokument vorhanden ist und der Einfachsignaturmodus aktiviert ist (siehe Konfigurationsparameter SAK_SIMPLE_SIGNATURE_ MODE), wird in Schritt 7 Variante a) durchgeführt, ansonsten Variante b). 7. Variante a) Die Signatur wird erstellt. Dies erfolgt gemäß TUC_KON_168 „Einzelsignatur QES erstellen". Variante b) Die Signaturen werden erstellt. Dies erfolgt gemäß TUC_KON_154 „QES-Signaturen erstellen". 8. Es wird DF.QES verlassen, um den PIN-Status der PIN.QES zurückzusetzen. Der im Konnektor verwaltetete Sicherheitszustand (CARDSESSION.AUTHSTATE) ist zu aktualisieren. 9. Die reservierten Ressourcen (Signaturkarte sowie PIN-Pad und Display des PIN-Eingabe-Kartenterminals) werden wieder freigegeben. Zur Freigabe der Signaturkarte wird TUC_KON_023 „Karte reservieren" cardSession; doLock = false } aufgerufen. 10. Die signierten Dokumente werden an den Aufrufer zurückgegeben.
Varianten/ Alterna-tiven	Der Nutzer kann den Vorgang bei der Autorisierung (Schritt 6) abbrechen. Hierbei sind die gleichen Regeln anzuwenden wie im Fehlerfall (s. Fehlerfälle).
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (->1) Ungültige Angabe des Signaturtyps oder Signaturvariante: Fehlercode 4111 Übergabe eines für die QES nicht unterstützten Dokumentformats: Fehlercode 4110 (->2) Kartentyp nicht zulässig für Signatur: Fehlercode 4126 (->5) Fehler bei der Reservierung von Ressourcen: Fehlercode 4060 (->7b) Karte ist kein HBA, sondern HBA-Vorläuferkarte: Fehlercode 4118

	Im Fehlerfall, inklusive Timeout bei der PIN-Eingabe, oder bei Abbruch durch den Benutzer (Fehler 4049): a) … MUSS DF.QES verlassen werden b) … MÜSSEN alle reservierten Ressourcen freigegeben werden c) … MUSS der Fehler immer an das Clientsystem zurückgemeldet werden
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	Abbildung PIC_KON_114 Aktivitätsdiagramm zu „Dokument QES signieren“

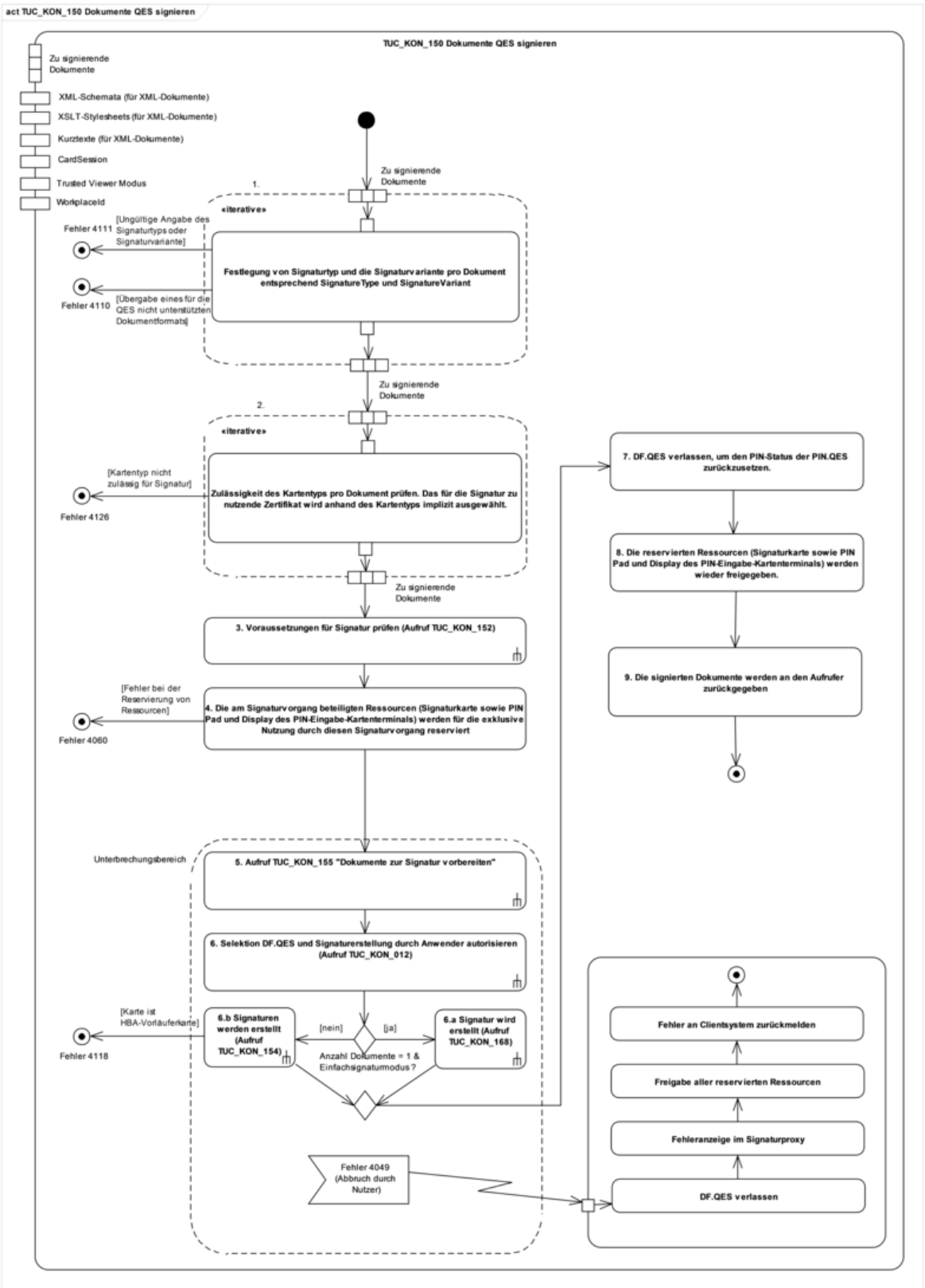


Abbildung 19: PIC_KON_114 Aktivitätsdiagramm zu „Dokument QES signieren“

Tabelle 217: TAB_KON_128 Fehlercodes TUC_KON_150 „Dokument QES signieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4060	Technical	Error	Ressource belegt
4110	Technical	Error	ungültiges Dokumentformat (%Format%) Der Parameter Format enthält das übergebene Dokumentformat.
4111	Technical	Error	ungültiger Signaturtyp oder Signaturvariante
4118	Technical	Error	Stapelsignaturen werden nur für den HBA unterstützt. Mit HBA-Vorläuferkarten sind nur Einzelsignaturen möglich.
4126	Security	Error	Kartentyp nicht zulässig für Signatur
4049	Technical	Error	Abbruch durch den Benutzer

[<=]

Anforderungen zur XML-Sicherheit:

TIP1-A_5113 - Abwehr von XML-Signature-Wrapping Angriffen
Der Konnektor MUSS XML-Signature-Wrapping-Angriffe (XSW) abwehren.
[<=]

4.1.8.4.5 Anforderungen an die Stapelsignatur

Eine Stapelsignatur definiert sich als „Erstellung einer begrenzten Anzahl Signaturen nach den zeitlich unmittelbar aufeinander folgenden Prozessen der Anzeige der zu signierenden Daten und der einmaligen Authentisierung des Signaturschlüssel-Inhabers gegenüber der qualifizierten elektronischen Signaturerstellungseinheit" (siehe [BSI-TR03114]).

TIP1-A_4669 - QES-Stapelsignatur
Der Signaturdienst MUSS die Möglichkeit bieten, Dokumente eines Stapels einzeln qualifiziert elektronisch zu signieren. Der Signaturdienst MUSS als qualifizierte elektronische Signaturerstellungseinheit für die Stapelsignatur den HBA unterstützen.
[<=]

TIP1-A_5664 - Reihenfolge der Dokumente bei Stapelsignatur
Die zu signierenden Dokumente einer Stapelsignatur MÜSSEN vom Signaturdienst im Konnektor in derselben Reihenfolge signiert, in der sie im Signaturauftrag vom Clientsystem geschickt werden.
[<=]

TIP1-A_4670 - Secure Messaging für die DTBS
Bei der Stapelsignatur MUSS der Signaturdienst die zu signierenden Daten (DTBS) über Secure Messaging vom Konnektor zum HBA übertragen. Dieser Secure Messaging-Kanal MUSS über die gSMC-K zum HBA mittels C.SAK.AUTD_CVC aufgebaut werden.
[<=]

TIP1-A_4671 - Verhalten des Konnektors beim Abbruch einer Stapelsignatur
Der Signaturdienst MUSS dem Benutzer während und nach einer PIN-Eingabe die Möglichkeit zum Abbruch einer Stapelsignatur anbieten.

Das geforderte Verhalten des Konnektors beim Abbruch einer Stapelsignatur wird in der folgenden Tabelle beschrieben. Hierbei werden die beiden Punkte „Abbruch, während die erneute PIN-Eingabe angefordert wird" (Nummer 1 bis 4) und „Abbruch, während der Vorgang der Signaturerstellung läuft" (Nummer 5 bis 6) unterschieden. Zeile Nummer 7 beschreibt alle sonstigen Fehlerfälle.
Ein Teilstapel einer Stapelsignatur ist durch die maximale Anzahl der Dokumente definiert, welche nach der Eingabe der Signatur-PIN durch den Signaturschlüssel-Inhaber signiert werden kann.

Tabelle 218: TAB_KON_192 Verhalten des Konnektors beim Abbruch einer Stapelsignatur

Nummer		Problem/Fehler/Ereignis	Verhalten des Konnektors
Während die erneute PIN-Eingabe angefordert wird	1	Timeout bei der PIN-Eingabe am KT	Der Signaturvorgang (Stapel) wird beendet: Kein „Fehler" Die Signaturen des/der vorherigen Teilstapel(s) bleiben erhalten und werden an das Clientsystem zurückgegeben. Keine weiteren Signaturen des neuen Teilstapels werden erstellt. (Die Weiterverarbeitung bereits erstellter Signaturen des letzten Teilstapels (sofern vorhanden) wird noch abgeschlossen).
	2	PIN gesperrt (nach mehrfacher Fehleingabe)	Siehe Verhalten unter Nummer 1
	3	Abbruchkommando „StopSignature" zur Jobnummer wird empfangen	Der Signaturvorgang (Stapel) wird beendet. Kein „Fehler" Keine weiteren Signaturen des neuen Teilstapels werden erstellt. (Die Weiterverarbeitung bereits erstellter Signaturen des letzten Teilstapels (sofern vorhanden) wird noch abgeschlossen).
	4	Abbruchtaste am Kartenterminal wird gedrückt	Siehe Verhalten unter Nummer 1

während der Vorgang der Signaturerstellung läuft	5	Abbruchkommando „StopSignature“ zur Jobnummer wird empfangen	Signaturvorgang (Stapel) wird abgebrochen. Kein „Fehler“ Keine weiteren Signaturen des Stapels werden erstellt. Keine weiteren Signaturen des Teilstapels werden erstellt. Bisher erstellte Signaturen des aktuellen Teilstapels werden verworfen.
	6	Abbruchtaste am Kartenterminal wird gedrückt.	Die „Abbruch“-Taste wird nicht vom Signaturdienst fortlaufend überwacht → Keine Aktion seitens des Signaturdienstes.
	7	Bei allen anderen Fehlerfällen (z. B.: es kommen zu viele Signaturen zurück, der Hash-Wert einer der Signaturen stimmt nicht, Karte gezogen, etc)	Signaturvorgang (Stapel) wird abgebrochen. Schwerer Fehler. Keine weiteren Signaturen des Stapels werden erstellt. Keine weiteren Signaturen des aktuellen Teilstapels werden erstellt. Bisher erstellte Signaturen aller Teilstapel werden verworfen. Es handelt sich um Probleme/Fehlerfälle, die bei typischen Angriffen auftreten können.

[<=]

4.1.8.4.6 TUC_KON_151 „QES Dokumentensignatur prüfen“

TIP1-A_4672-02 - TUC_KON_151 „QES-Dokumentensignatur prüfen“
Der Konnektor MUSS den technischen Use Case TUC_KON_151 „QES-Dokumentensignatur prüfen” umsetzen.

Tabelle 219: TAB_KON_591 - TUC_KON_151 „QES-Dokumentensignatur prüfen“

Element	Beschreibung
Name	TUC_KON_151 „QES-Dokumentensignatur prüfen”
Beschreibung	Es wird die QES eines Dokuments geprüft. Dabei werden die Signaturverfahren laut Tabelle TAB_KON_582 – Signaturverfahren unterstützt. Sind mehrere Signaturen vorhanden, so werden alle geprüft. Auch Parallel- und Gegensignaturen MÜSSEN unterstützt werden.
Eingangsanforderung	keine
Auslöser	Aufruf durch ein Clientsystem (Operation VerifyDocument) oder durch ein Fachmodul im Konnektor

Vorbedingungen	keine
Eingangsdaten	• signedDocument – *optional* (QES-signiertes Dokument vom Typ `QES_DocFormate` -> siehe Definition in Operation `VerifyDocument` mit `SIG:Document`) • signatureObject – *optional* (-> siehe Definition in Operation VerifyDocument mit dss:SignatureObject. Es werden Parallel- und Gegensignaturen unterstützt.) • optionalInputParams (optionale Eingabeparameter, siehe Operation VerifyDocument, Parameter SIG:OptionalInputs) • certificates – *optional/falls diese nicht im signierten Dokument enthalten sind, sondern nur referenziert werden* (X.509-Zertifikate). • xmlSchemas – *optional/nur für XML-Dokumente* (XMLSchema und ggf. weitere vom Hauptschema benutzte Schemata) • includeRevocationInfo [Boolean]: – *optional; Default: false* (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur)
Komponenten	Konnektor
Ausgangsdaten	• verificationResult [VerificationResult] (Ergebnis der Signaturprüfung) • optionalOutput – *optional* (weitere Ausgabedaten gemäß SIG:OptionalOutput)
Standardablauf	**1. „DocumentValidation"**: Das signierte Dokument wird validiert mit Aufruf TUC_KON_080 „Dokument validieren"{ … }. Treten Fehler bei der Validierung der Typkonformität auf, wenn die Signatur im Dokument eingebettet ist, wird die Prüfung mit einem Fehler abgebrochen. Treten bei der Typkonformität, wenn die Signatur nicht im Dokument eingebettet ist, Fehler auf, so bricht der TUC nicht ab, sondern führt die folgenden Schritte soweit sinnvoll möglich durch. (Die Entscheidung über das sinnvoll Durchführbare liegt beim Hersteller des Konnektors.) **2. „CoreValidation"**: Es erfolgt die mathematische Prüfung der Signatur, bestehend aus der Prüfung der Hash-Kette bis zum signierten Hashwert und der Prüfung der Signatur unter Verwendung des öffentlichen Schlüssels, des

	Signaturwertes und des signierten Hashwertes. XML-Signatur: Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation. CMS-Signatur: Die Core Validation erfolgt entsprechend Cryptographic Message Syntax (CMS) Kapitel 5.6 Signature Verification Process [RFC5652]. PDF-Signatur: Die Core Validation erfolgt entsprechend [PAdES-3] Kapitel 4.6 Signature Validation aus PAdES-BES Part 3. Auch wenn die Validierung fehlschlägt, werden die folgenden Prüfschritte durchgeführt, so dass ein vollständiges Prüfprotokoll erstellt werden kann. **3. „CheckSignatureCertificate":** **Teil 1: Signaturzertifikat ermitteln** XML-Signatur: Das Signaturzertifikat ist im XMLDSig Element `ds:KeyInfo/ds:X509Data` gespeichert [XMLDSig] oder wird als Eingangsparameter übergeben. CMS-Signatur: Das Signaturzertifikat für CAdES ist im Feld `certificates` im `SignedData` Container gespeichert [CAdES] oder wird als Eingangsparameter übergeben. PDF-Signatur: Das PDF Signaturzertifikat für PAdES ist im Feld SignedData.certificates entsprechend Kapitel 6.1.1 „Placements of the signing certificate" [PAdES Baseline Profile] gespeichert oder wird als Eingangsparameter übergeben. **Teil 2: Signaturzeitpunkt bestimmen** Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_Eingebettet` wird wie folgt selektiert: XML-Signatur: Das XML element `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 7.2.1 XAdES [XAdES]. CMS-Signatur: Das Attribut `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 11.3 CMS [CMS]. PDF-Signatur: Der Signaturzeitpunkt kann dem M Eintrag des Signature

	Dictionary
	entnommen werden [PAdES Baseline Profile] Kapitel 6.2.1 Signing
	time.
	
	Der Signaturzeitpunkt `Benutzerdefinierter_Zeitpunkt` liegt gegebenenfalls als Aufrufparameter vor. Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_System` wird ermittelt.
	
	Teil 3: Signaturzertifikatsprüfung:
	Bei der folgenden Signaturzertifikatsprüfung sind die Signaturzeitpunkte gemäß [TIP1-A_5540] zu berücksichtigen.
	Die Signaturzertifikatsprüfung erfolgt durch Aufruf von TUC_KON_037 „Zertifikat prüfen“ {
	certificate = C.HP.QES;
	qualifiedCheck = required;
	baseTime = Signaturzeitpunkt;
	offlineAllowNoCheck = true;
	validationMode = OCSP;
	ocspResponses = OCSP-Response;
	getOCSPResponses = includeRevocationInfo
	}.
	Sind OCSP-Responses in der Signatur eingebettet, ist die jüngsten OCSP-Response des EE-Zertifikats, die für die Zertifikatsprüfung notwendig ist, beim Aufruf von TUC_KON_037 zu übergeben.
	Sofern der Aufruf von TUC_KON_037 ocspResponses zurückgibt, wird die OCSP-Response des EE-Zertifikats in die Signatur eingebettet.
	Auch wenn die Zertifikatsprüfung fehlschlägt, werden die folgenden Prüfungen durchgeführt.
	
	4. „CheckPolicyConstraints“:
	
	In diesem Schritt wird das signierte Dokument entsprechend der Profilierung der Signaturformate (siehe Anhang B.2) geprüft. Es sind die Vorgaben für die Prüfung von Signaturen aus den Standards für AdES [XAdES], [XAdES Baseline], [CAdES], [CAdES Baseline], [PAdES-3] und [PAdES Baseline] umzusetzen. Dabei sind die Vorgaben aus Tabelle TAB_KON_779 „Profilierung der Signaturformate“ und Tabelle TAB_KON_778 „Einsatzbereich der Signaturvarianten“ zu erfüllen.
	
	Auch wenn nicht alle Anforderungen an das Format des signierten Dokuments erfüllt werden, wird die Prüfung mit den folgenden Schritten fortgesetzt, um ein vollständiges Prüfungsprotokoll zu erhalten.

	5. Das **Prüfergebnis** (VerificationResult, OptionalOutput) wird an den Aufrufer zurückgegeben (siehe TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur).
Varianten/Alternativen	Keine
Fehlerfälle	Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_592 Fehlercodes TUC_KON_151 „QES Dokumentensignatur prüfen" beschrieben. (->1) keine Signatur in signedDocument und signatureObject vorhanden: 4253. (→ 2 **„CoreValidation"**) Interner Fehler: 4001, Signatur des Dokuments ungültig: 4115, Signatur umfasst nicht das gesamte Dokument: 4262 (→3 **„CheckSignatureCertificate"**) Interner Fehler: 4001, Signaturzertifikat ermitteln ist fehlgeschlagen: 4206. (→4 **„CheckPolicyConstraints"**) Interner Fehler: 4001, Dokument nicht konform zu Regeln für QES: 4124, Dokument nicht konform zu Profilierung der Signaturformate: 4208.
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	keine

Tabelle 220: TAB_KON_592 Fehlercodes TUC_KON_151 „QES Dokumentensignatur prüfen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4001	Technical	Error	interner Fehler
4115	Security	Error	Signatur des Dokuments ungültig. Prüfung der Hashwertkette bzw. Prüfung der kryptographischen Signatur fehlgeschlagen.
4124	Technical	Error	Dokument nicht konform zu Regeln für QES
4206	Technical	Error	Signaturzertifikat ermitteln ist fehlgeschlagen
4208	Technical	Error	Dokument nicht konform zu Profilierung der Signaturformate
4253	Technical	Error	Keine Signatur im Aufruf
4262	Technical	Error	Signatur umfasst nicht das gesamte Dokument
4264	Technical	Warning	Ein oder mehrere Zertifikate ignoriert

Das Gesamtergebnis (VerificationResult) für die Prüfung einer Dokumentensignatur fasst die Ergebnisse
aller Prüfungsschritte in einem einzelnen Statuswert zusammen.

Tabelle 221: TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur

VerificationResult für gesamtes Dokument (VerificationResult/HighLevelResult)	
Wert	Bedeutung
VALID	Wenn VerificationResult für alle Signaturen zum Dokument VALID
INVALID	Wenn VerificationResult für eine Signatur zum Dokument INVALID
INCONCLUSIVE	in allen anderen Fällen
VerificationResult pro Signatur (VerificationReport/IndividualReport/Result)	
Wert	Bedeutung mögliche Ausprägungen im VerificationReport
VALID	Die Signatur wurde gemäß den Regeln für die QES geprüft und für gültig befunden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:HasManifestResults
INVALID	Die Signatur ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:ResponderError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation

	ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:CertificateChainNotComplete
INCONCLUSIVE	Die Signatur wurde gemäß den Regeln für die QES geprüft. Allerdings konnten eine oder mehrere Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail. Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:OcspNotAvailiable Hinweis: Das Erreichen dieses Zustandes hängt davon ab, ob eine OCSP-Abfrage nicht durchgeführt werden konnte, unabhängig davon, ob die Ursache dafür die Offlineschaltung des Konnektors (MGM_LU_ONLINE = Disabled) oder die Nichterreichbarkeit des OCSP-Responders im Online-Betrieb (MGM_LU_ONLINE = Enabled) ist.

[<=]

TIP1-A_5540-01 - QES-Signaturprüfergebnis bezogen auf Signaturzeitpunkt
Der Konnektor MUSS zur QES-Signaturprüfung ein Prüfergebnis, das sich auf genau einen angenommenen Signaturzeitpunkt bezieht, an den Aufrufer zurückgeben.
Die Auswahl des angenommenen Signaturzeitpunkts, auf den sich das Signaturergebnis bezieht, erfolgt hierarchisch:

- `Benutzerdefinierter_Zeitpunkt`
  falls vorhanden, sonst
- `Ermittelter_Signaturzeitpunkt_Eingebettet`
  falls vorhanden, sonst
- `Ermittelter_Signaturzeitpunkt_System`

[<=]

4.1.8.4.7 TUC_KON_170 „Dokumente mit Komfort signieren"

A_19103-06 - TUC_KON_170 "Dokumente mit Komfort signieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_170 „Dokumente mit Komfort signieren" umsetzen.

Tabelle 222: TAB_KON_871 – TUC_KON_170 „Dokumente mit Komfort signieren"

Element	Beschreibung
Name	TUC_KON_170 "Dokumente mit Komfort signieren"

Beschreibung	Im Rahmen von Fachanwendungen werden ein oder mehrere Dokumente mit einer Komfortsignatur versehen. Es werden die QES_DocFormate unterstützt.
Auslöser	Aufruf durch ein Clientsystem (Operation SignDocument) oder ein Fachmodul.
Vorbedingungen	Die Signaturkarte muss gesteckt sein.
Eingangsdaten	• signRequests (Liste von Signaturaufträgen) Jeder Signaturauftrag (SignRequest) kapselt: • documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente); darin u.a. documentFormat (Formatangabe für das zu signierende Dokument) • optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur, siehe Operation SignDocument, Parameter dss:OptionalInputs); darin u.a. signatureType (URI für den Signaturtyp XML-, CMS-, PDF-Signatur) • includeRevocationInfo [Boolean]: – optional; Default: true (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur; siehe Operation SignDocument, Parameter SIG:IncludeRevocationInfo) • cardSession (Kartensitzung. Unterstützte Kartentypen: HBA) • crypt [SIG_CRYPT_QES] - *optional*; default und Wertebereich: siehe TAB_KON_862-01 (Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden.) • workplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBA)
Ausgangsdaten	• signedDocuments (Liste der signierten Dokumente)
Standardablauf	Der Konnektor KANN die Schritte 1 bis 5 in einer beliebigen Reihenfolge durchführen. 1. Prüfe `SAK_COMFORT_SIGNATURE = Enabled` 2. Prüfe, ob der Komfortsignatur-Timer der cardSession (`SAK_COMFORT_SIGNATURE_TIMER`) abgelaufen ist. 3. Der Signaturtyp und die Signaturvariante werden für jedes Dokument der Liste entsprechend signatureType und SignatureVariant festgelegt (ggf. in optionalInputs enthalten). Wenn SignatureType oder SignatureVariant nicht übergeben wurden, wird das dem Dokumentformat entsprechende Default-

	Verfahren gewählt (siehe TAB_KON_583 – Default-Signaturverfahren). 4. Für alle Dokumente des Stapels wird die Zulässigkeit des Kartentyps geprüft. Das für die Signatur zu nutzende Zertifikat wird anhand des Kartentyps und des Parameters crypt ausgewählt. 5. Es werden die Voraussetzungen für die Signatur geprüft. Dies erfolgt im TUC_KON_152 „Signaturvoraussetzungen für QES prüfen". Wenn includeRevocationInfo=true, dann setze ocspResponses auf Rückgabewert von TUC_KON_152. 6. Die am Signaturvorgang beteiligte RessourceSignaturkarte wird für die exklusive Nutzung durch diesen Signaturvorgang reserviert. Die Reservierung der Signaturkarte erfolgt durch Aufruf von TUC_KON_023 „Karte reservieren" { cardSession; doLock = true }. 7. Zum Vorbereiten der Dokumente für die Signatur wird TUC_KON_155 „Dokumente zur Signatur vorbereiten" mit ocspResponses aufgerufen. Die Zugriffe auf die Signaturkarte im Schritt 8 müssen im DF.QES erfolgen. DF.QES darf am Ende des TUCs nicht verlassen werden. 8. Die Signaturen werden erstellt. Dies erfolgt gemäß TUC_KON_158 „Komfortsignaturen erstellen". 9. Die reservierte Ressource Signaturkarte wird wieder freigegeben. Zur Freigabe der Signaturkarte wird TUC_KON_023 „Karte reservieren" cardSession; doLock = false } aufgerufen. 10. Die signierten Dokumente werden an den Aufrufer zurückgegeben.
Varianten/ Alternativen	keine
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes. In den Fehlerfällen, die zum Abbruch des Komfortsignaturmodus mit Fehlercode 4271 führen, wird vor dem Abbruch TUC_KON_172 für das cardHandle des HBA ausgeführt. (->1) Komfortsignaturfunktion im Konnektor nicht aktiviert: Fehlercode 4263 (->2) Der Komfortsignatur-Timer der cardSession ist abgelaufen: Fehlercode 4271 (->3) Ungültige Angabe des Signaturtyps oder Signaturvariante: Fehlercode 4111 Übergabe eines für die QES nicht unterstützten

	Dokumentformats: Fehlercode 4110 (->4) Kartentyp nicht zulässig für Signatur: Fehlercode 4126 (->6) Fehler bei der Reservierung der Signaturkarte: Fehlercode 4060 (->8) Karte ist kein HBA, sondern HBA-Vorläuferkarte: Fehlercode 4274 Im Fehlerfall: a) … DARF DF.QES NICHT verlassen werden b) … MÜSSEN alle reservierten Ressourcen freigegeben werden c) … MUSS der Fehler immer an das Clientsystem zurückgemeldet werden
Sicherheits-anforderungen	Der Konnektor MUSS sicherstellen, dass der erhöhte Sicherheitszustand der PIN.QES nur für die Komfortsignatur mittels TUC_KON_170 innerhalb einer Kartensitzung nachgenutzt werden darf.

Tabelle 223: TAB_KON_872 Fehlercodes TUC_KON_170 „Dokumente mit Komfort signieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4060	Technical	Error	Ressource belegt
4110	Technical	Error	ungültiges Dokumentformat (%Format%) Der Parameter Format enthält das übergebene Dokumentformat.
4111	Technical	Error	ungültiger Signaturtyp oder Signaturvariante
4126	Security	Error	Kartentyp nicht zulässig für Signatur
4049	Technical	Error	Abbruch durch den Benutzer
4263	Technical	Error	Komfortsignaturfunktion nicht aktiviert
4271	Technical	Error	Komfortsignaturmodus abgebrochen
4274	Technical	Error	Komfortsignaturen werden nur für den HBA unterstützt

[<=]

4.1.8.4.8 TUC_KON_171 „Komfortsignatur einschalten"

A_19104-03 - TUC_KON_171 „Komfortsignatur einschalten"
Der Konnektor MUSS den technischen Use Case TUC_KON_171 „Komfortsignatur einschalten" umsetzen.

Tabelle 224: TAB_KON_883 – TUC_KON_171 „Komfortsignatur einschalten“

Element	Beschreibung
Name	TUC_KON_171 „Komfortsignatur einschalten“
Beschreibung	Zum Einschalten des Komfortsignaturmodus wird die PIN.QES verifiziert und der Signaturmodus „Comfort“ für die cardSession gesetzt.
Auslöser	• Operation ActivateComfortSignature • Aufruf durch ein Fachmodul
Vorbedingungen	Der Karte muss gesteckt sein.
Eingangsdaten	• cardSession (nur HBA erlaubt)
Komponenten	Konnektor, Kartenterminal, Karte (HBA)
Ausgangsdaten	• signatureMode
Standardablauf	1. Prüfe `SAK_COMFORT_SIGNATURE = Enabled` 2. Die am Vorgang beteiligten Ressourcen (Karte sowie PIN-Pad und Display des PIN-Eingabe-Kartenterminals) werden für die exklusive Nutzung durch diesen Vorgang reserviert. Die Reservierung der Karte erfolgt durch Aufruf von TUC_KON_023 „Karte reservieren“ { cardSession; doLock = true } Der Zugriff auf die Karte im Schritt 3 muss im DF.QES erfolgen. Das DF.QES darf danach nicht verlassen werden, damit der PIN-Status der PIN.QES erhalten bleibt. 3. Die Einschaltung der Komfortsignatur wird durch den Anwender autorisiert. Dies erfolgt durch Aufruf von TUC_KON_012 „PIN verifizieren“ { cardSession; workplaceId; pinRef = PIN.QES; verificationType = Mandatorisch } Für die Anzeige am Kartenterminal ist die Displaymessage für „Komfortsignatur aktivieren“ aus TAB_KON_090 zu verwenden. 4. Setze `CARDSESSION.SIGNMODE` = Comfort 5. Starte Komfortsignatur-Timer für die cardSession bei „0“ 6. Die reservierten Ressourcen (Karte sowie PIN-Pad und Display des PIN-Eingabe-Kartenterminals) werden wieder freigegeben. Zur Freigabe der Karte wird TUC_KON_023 „Karte reservieren“ cardSession; doLock = false } aufgerufen.

Varianten/ Alternativen	Keine
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (->1) Komfortsignaturfunktion im Konnektor nicht aktiviert: Fehlercode 4263 (->2) Fehler bei der Reservierung von Ressourcen: Fehlercode 4060 (->3) Karte ist kein HBA, sondern HBA-Vorläuferkarte: Fehlercode 4274 (->3) pinResult = BLOCKED: Fehlercode 4275 (->3) pinResult = REJECTED: Fehlercode 4276 (->4) Fehler beim Setzen des Signaturmodus: Fehlercode 4267 (->5) Fehler beim Starten des Komfortsignatur-Timers: Fehlercode 4267 Im Fehlerfall, inklusive Timeout bei der PIN-Eingabe, oder bei Abbruch durch den Benutzer (Fehler 4049): a) … MUSS (ab Schritt 5) `CARDSESSION.SIGNMODE` = PIN gesetzt werden b) … MUSS (ab Schritt 3) DF.QES verlassen werden c) … MÜSSEN alle reservierten Ressourcen freigegeben werden d) … MUSS der Fehler immer an das Clientsystem zurückgemeldet werden

Tabelle 225: TAB_KON_886 Fehlercodes TUC_KON_171 „Komfortsignatur einschalten“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4049	Technical	Error	Abbruch durch den Benutzer
4060	Technical	Error	Ressource belegt
4263	Technical	Error	Komfortsignaturfunktion nicht aktiviert
4267	Technical	Error	Fehler beim Aktivieren des Komfortsignaturmodus <cardHandle>
4274	Technical	Error	Komfortsignaturen werden nur für den HBA unterstützt
4275	Technical	Error	Security Error PIN jetzt gesperrt (BLOCKED)
4276	Technical	Error	Security Error PIN falsch (REJECTED)

[<=]

4.1.8.4.9 TUC_KON_172 „Komfortsignatur ausschalten”

A_19105 - TUC_KON_172 „Komfortsignatur ausschalten“
Der Konnektor MUSS den technischen Use Case TUC_KON_172 „Komfortsignatur ausschalten“ umsetzen.

Tabelle 226: TAB_KON_884 – TUC_KON_172 „Komfortsignatur ausschalten"

Element	Beschreibung
Name	TUC_KON_172 „Komfortsignatur ausschalten"
Beschreibung	Zum Ausschalten des Komfortsignaturmodus werden die Sicherheitszustände der Karte(n), die im Konnektor verwalteten Sicherheitszustände und der Signaturmodus der cardSession(s) zurückgesetzt.
Auslöser	• Operation DeactivateComfortSignature • TUC_KON_158 • Der Administrator setzt SAK_COMFORT_SIGNATURE = Disabled • Aufruf durch ein Fachmodul
Vorbedingungen	Die Karten müssen gesteckt sein.
Eingangsdaten	Bei Auslösen des TUCs durch den Administrator: • Keine Ansonsten: • cardHandles : Liste von cardHandles (nur HBA erlaubt)
Komponenten	Konnektor, Kartenterminal, Karte (HBA)
Ausgangsdaten	Keine
Standardablauf	1. Wenn der TUC nicht durch den Administrator ausgelöst wurde: Prüfe `SAK_COMFORT_SIGNATURE = Enabled` 2. Wenn der TUC durch den Administrator ausgelöst wurde: Ermittle die cardHandles aller gesteckten HBA. 3. Für jedes übergebene bzw. ermittelte cardHandle: 4. Ermittle cardSessions zu cardHandle 5. Für jede ermittelte cardSession: a. Setze den PIN-Status der PIN.QES zurück (z. B. durch Verlassen von DF.QES für alle logischen Kanäle der Karte) b. Lösche den im Konnektor verwalteten Sicherheitszustand aus `CARDSESSION.AUTHSTATE (PINRef=PIN.QES)` c. Setze `CARDSESSION.SIGNMODE = PIN` d. Stoppe Komfortsignatur-Timer für die cardSession
Varianten/ Alternativen	Keine
Fehlerfälle	(->1) Komfortsignaturfunktion im Konnektor nicht aktiviert: Fehlercode 4263 Fehler und Warnungen in den folgenden Schritten werden

	über alle cardHandle akkumuliert und die <komma-separierte Liste von cardHandle> für den jeweiligen Fehlertext erzeugt. (->3) Bei einem ungültigen cardHandle wird mit dem nächsten cardHandle aus cardHandles fortgesetzt. Fehlercode 4265 (->4) Ist zu einem cardHandle keine cardSession vorhanden wird mit dem nächsten cardHandle fortgesetzt. Fehlercode 4266 (->5) Tritt in Schritt 4 ein Fehler auf wird mit dem nächsten cardHandle fortgesetzt. Fehlercode 4268

Tabelle 227: TAB_KON_887 Fehlercodes TUC_KON_172 „Komfortsignatur ausschalten"

Fehlercode	ErrorType	Severity	Fehlertext
4263	Technical	Fehler	Komfortsignaturfunktion nicht aktiviert
4265	Technical	Warning	Karten-Handle ungültig <komma-separierte Liste von cardHandle>
4266	Technical	Warning	Keine Kartensitzung vorhanden <komma-separierte Liste von cardHandle>
4268	Technical	Fehler	Fehler beim Deaktivieren des Komfortsignaturmodus <komma-separierte Liste von cardHandle>

[<=]

4.1.8.4.10 TUC_KON_173 „Liefere Signaturmodus"

A_19106-01 - TUC_KON_173 „Liefere Signaturmodus"
Der Konnektor MUSS den technischen Use Case TUC_KON_173 „Liefere Signaturmodus" umsetzen.

Tabelle 228: TAB_KON_885 – TUC_KON_173 „Liefere Signaturmodus"

Element	Beschreibung
Name	TUC_KON_173 „Liefere Signaturmodus"
Beschreibung	Der aktuell konfigurierte Status der Komfortsignaturfunktion im Konnektor und, falls vorhanden, Informationen zu der aktuell im Konnektor existierenden Komfortsignatursession werden ermittelt und an den Aufrufer zurückgegeben.
Auslöser	• Operation GetSignatureMode • Aufruf durch ein Fachmodul
Vorbedingungen	Keine
Eingangsdaten	• cardSession (Kartensitzung. Unterstützte Kartentypen: HBA)

Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBA)
Ausgangsdaten	• comfortSignatureStatus • comfortSignatureMax • comfortSignatureTimer • sessionInfo (optional): Struktur aus signatureMode, countRemaining, timeRemaining
Standardablauf	1. Ermittle den Status der Komfortsignaturfunktion: comfortSignatureStatus=SAK_COMFORT_SIGNATURE 2. Ermittle comfortSignatureMax= SAK_COMFORT_SIGNATURE_MAX 3. Ermittle comfortSignatureTimer= SAK_COMFORT_SIGNATURE_Timer 4. Ermittle sessionInfo a. Ermittle den Signaturmodus (signatureMode) aus CARDSESSION.SIGNMODE b. Ermittle Differenz von SAK_COMFORT_SIGNATURE_MAX und Komfortsignatur-Zähler der cardSession (countRemaining) c. Ermittle verbleibende Zeit aus SAK_COMFORT_SIGNATURE_TIMER und Komfortsignatur-Timer der cardSession (timeRemaining) 5. Wenn signatureMode = "Comfort" wird sessionInfo an den Aufrufer zurückgegeben.
Varianten/ Alternativen	Keine
Fehlerfälle	Wenn im Standardablauf ein Fehler auftritt, wird mit Fehler 4269 abgebrochen.

Tabelle 229: TAB_KON_888 Fehlercodes TUC_KON_173 „Liefere Signaturmodus“

Fehlercode	ErrorType	Severity	Fehlertext
4269	Technical	Error	Fehler beim Ermitteln des Signaturmodus <cardHandle>

[<=]

4.1.8.5 Operationen an der Außenschnittstelle

TIP1-A_4676-08 - Basisdienst Signaturdienst (nonQES und QES)
Der Konnektor MUSS Clientsystemen den Basisdienst Signaturdienst (nonQES und QES) anbieten.

Tabelle 230: TAB_KON_197 Basisdienst Signaturdienst (nonQES und QES)

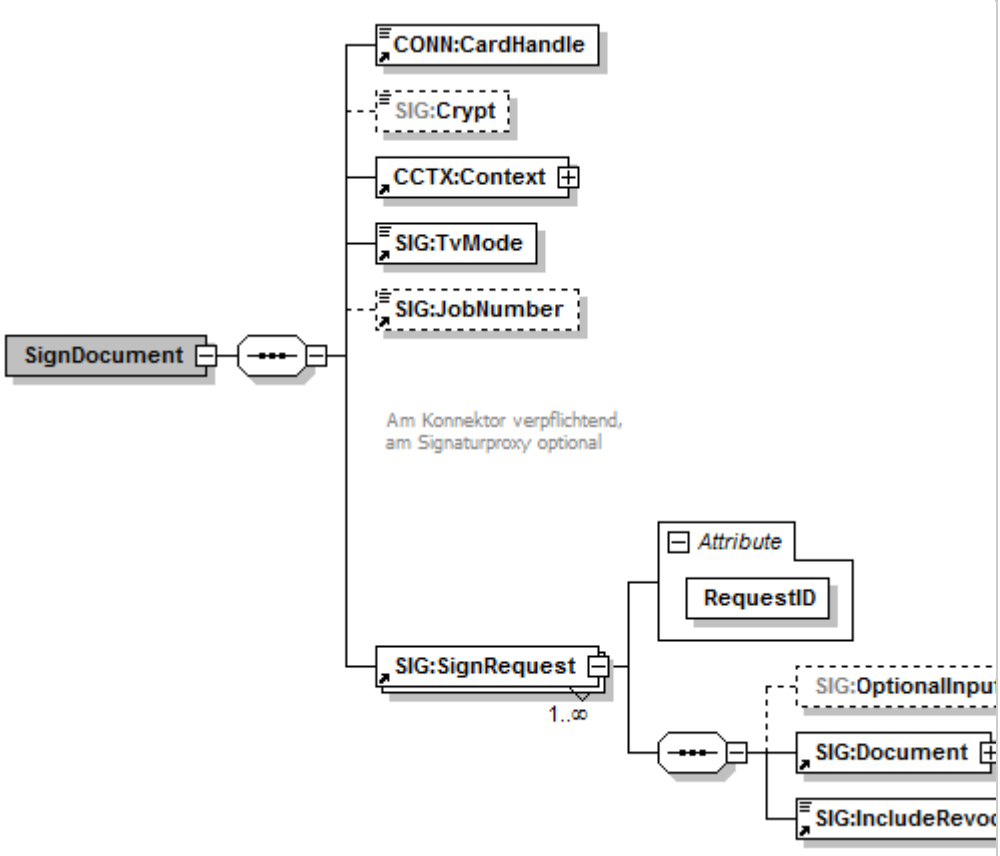
Name	SignatureService	
Version (KDV)	7.4.0 (WSDL-Version), 7.4.2 (XSD-Version) 7.4.2 (WSDL-Version), 7.4.4 (XSD-Version) 7.5.5 (WSDL- und XSD-Version) Siehe Anhang D	
Namensraum	Siehe Anhang D	
Namensraum-Kürzel	SIG für Schema und SIGW für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	SignDocument	Dokument signieren
	VerifyDocument	Signatur verifizieren
	StopSignature	Signieren eines Dokumentenstapels abbrechen
	GetJobNumber	Liefert eine Jobnummer für den nächsten Signiervorgang
	ActivateComfortSignature	Aktiviert die Komfortsignatur für einen HBA
	DeactivateComfortSignature	Deaktiviert die Komfortsignatur für einen oder mehrere HBA
	GetSignatureMode	Liefert den Status der Komfortsignaturfunktion und Informationen zur Komfortsignatursession eines HBA
WSDL	SignatureService_V7_5_5.wsdl SignatureService_V7_4_2.wsdl SignatureService.wsdl (WSDL-Version 7.4.0)	
Schema	SignatureService_V7_5_5.xsd SignatureService_V7_4_4.xsd SignatureService.xsd (XSD-Version 7.4.2)	

[<=]

4.1.8.5.1 SignDocument (nonQES und QES)

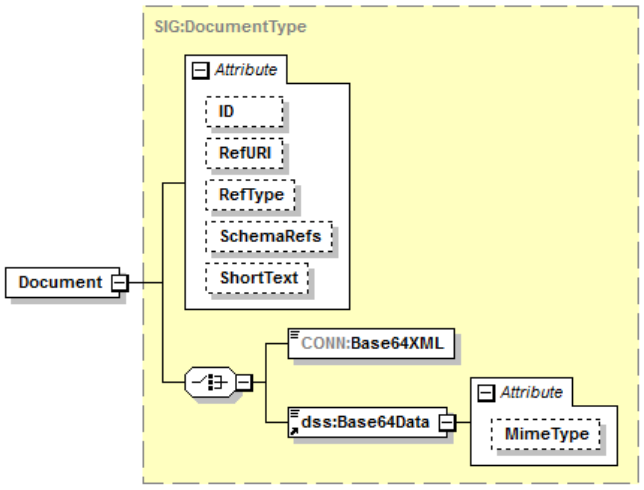
TIP1-A_5010-07 - Operation SignDocument (nonQES und QES)
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation SignDocument anbieten**.**

Tabelle 231: TAB_KON_065 Operation SignDocument (nonQES und QES)

Name	`SignDocument`
Beschreibung	Diese Operation lehnt sich an [OASIS-DSS] an. Sie enthält voneinander unabhängige SignRequests. Jeder SignRequest erzeugt eine Signatur für ein Dokument. Für die qualifizierte elektronische Signatur (QES) werden die `QES_DocFormate` unterstützt. Für nicht-qualifizierte elektronische Signaturen (nonQES) werden die `nonQES_DocFormate` unterstützt. Zur Signaturerzeugung werden Schlüssel und Zertifikate einer Chipkarte benutzt. Unterstützte Karten sind für die QES der HBAx mit dem QES-Zertifikat. Für die nonQES wird für die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur" die SM-B mit dem OSIG–Zertifikat unterstützt. Bei der Erstellung von XML-Signaturen MUSS Canonical XML 1.1 verwendet werden [CanonXML1.1]. Es soll der Common-PKI-Standard eingesetzt werden, siehe [Common-PKI]. In Summe für die Größe der Dokumente in allen SignRequests innerhalb einer SignDocument-Anfrage MUSS der Konnektor eine Gesamtgröße von <= 250 MB unterstützen.
Aufruf-parameter	

	Name	Beschreibung
	CONN: Card Handle	Identifiziert die zu verwendende Signaturkarte. Die Operation DARF die Signatur mit der eGK NICHT unterstützen. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4126 abbrechen.
	SIG: Crypt	Der Parameter crypt steuert die Auswahl der Zertifikate und Schlüssel für die Signaturerstellung abhängig von der durch cardHandle adressierten Karte gemäß TAB_KON_900. Defaultwert: • gemäß TAB_KON_862-01 für die QES • gemäß TAB_KON_863 für die nonQES.
	CCTX: Context	Aufrufkontext QES mit HBAx: MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend Aufrufkontext nonQES mit SM-B: MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
	TvMode	Der Parameter wird im Konnektor nicht ausgewertet.
	SIG: JobNumber	Die Nummer des Jobs, unter der der nächste Signaturvorgang gestartet wird. Parameter ist verpflichtend.
	SIG: Sign Request	Ein SignRequest kapselt den Signaturauftrag für ein Dokument. Das verpflichtende XML-Attribut RequestID identifiziert einen SignRequest innerhalb eines Stapels von SignRequests eindeutig. Es dient der Zuordnung der SignResponse zum jeweiligen SignRequest. Enthält der Aufruf mehr als die unterstützte Anzahl von SignRequests, bricht die Operation mit Fehler 4000 ab. Es sind mindestens 50 SignRequests zu unterstützen.

	SIG: Optional Inputs	Enthält optionale Eingangsparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7): 

	SIG: Document	 Dieses an das dss:Document Element aus [OASIS-DSS] Section 2.4.2 angelehnte Element enthält das zu signierende Dokument, wobei die Kindelemente CONN:Base64XML und dss:Base64Data auftreten können. Bei den als dss:Base64Data übergebenen Dokumenten werden folgende (Klassen von) MIME-Types unterschieden: • ”application/pdf-a” – für PDF/A-Dokumente, • ”text/plain”, ”text/plain; charset=iso-8859-15” oder ”text/plain; charset=utf-8” – für Text-Dokumente, • ”image/tiff” – für TIFF-Dokumente und • ein beliebiger anderer MIME-Type für nicht näher unterschiedene Binärdaten des spezifizierten Typs. Der MIME-Type „text/plain“ wird interpretiert als „text/plain; charset=iso-8859-15”. Das Element enthält ein Attribut ShortText Es muss für QES-Signaturen bei jedem Aufruf vom Clientsystem übergeben werden, für nonQES-Signaturen ist es optional. Über das Attribut RefURI kann gemäß [OASIS-DSS] (Abschnitt 2.4.1) ein zu signierender Teilbaum eines XML-Dokuments ausgewählt werden. Wenn die Signatur eines Teilbaums für die Signaturvariante nicht unterstützt wird, muss der Signaturauftrag mit Fehler 4111 abgelehnt werden.

	SIG: Include Revocation Info	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturerstellung vorliegenden Sperrinformationen anfordern. Es wird ausschließlich die zu erstellende Signatur betrachtet, d.h. es erfolgt keine Einbettung von Sperrinformationen für bereits enthaltene Signaturen. Für nicht-qualifizierte elektronische Signaturen (nonQES) wird diese Funktionalität nicht unterstützt. Für PDF-Signaturen werden keine Sperrinformationen eingebettet.

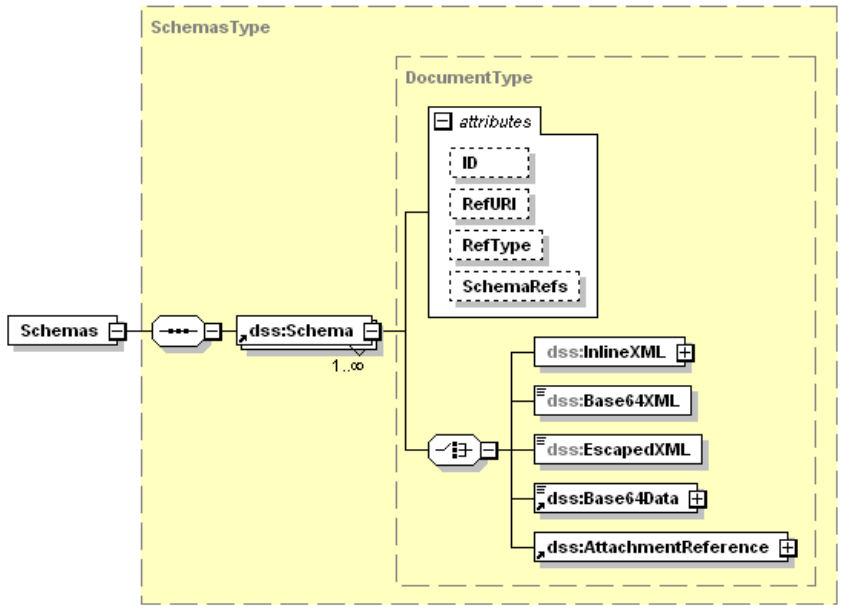
	`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element kann der generelle Typ der zu erzeugenden Signaturen spezifiziert werden. Hierbei MÜSSEN folgende Signaturtypen unterstützt werden: • **XML-Signatur** Durch Übergabe der URI urn:ietf:rfc:3275 wird die Erstellung von XML-Signaturen gemäß [RFC3275], [XMLDSig] angestoßen. Das zu verwendende Profil ist XAdES-BES ([XAdES]). Die Rückgabe einer solchen Signatur erfolgt als `ds:Signature`-Element. • **CMS-Signatur** Durch Übergabe der URI urn:ietf:rfc:5652 wird eine CMS-Signatur gemäß [RFC5652] angestoßen. Das zu verwendende Profil ist CAdES-BES ([CAdES]). Die Signatur wird als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert. • **S/MIME-Signatur** Durch Übergabe der URI „urn:ietf:rfc:5751" wird eine S/MIME-Signatur gemäß [RFC5751] angestoßen. Die CMS-Signatur der übergebenen MIME-Nachricht erfolgt konform der Vorgaben zur CMS-Signatur. Das Rückgabedokument ist eine MIME-Nachricht vom Typ „application/pkcs7-mime" mit einer CMS-Struktur vom Typ_SignedData. Ist das übergebene Dokument keine MIME-Nachricht, so wie der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. • **PDF-Signatur** Durch Übergabe der URI http://uri.etsi.org/02778/3 wird die Erzeugung einer PAdES-Basic Signatur gemäß [PAdES-3]angestoßen, wobei das Dokument mit der integrierten Signatur als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert wird. Handelt es sich beim übergebenen Dokument nicht um ein `Base64Data`-Element mit MIME-Type „application/pdf-a", so wird ein Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder Signaturvariante).

		Die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur“ DÜRFEN für QES der HBAx nur mit dem QES-Zertifikat erfolgen, für nonQES nur mit dem OSIG-Zertifikat der SM-B. In jedem diese Anforderung verletzenden Fall MUSS der Fehler 4058 (Aufruf nicht zulässig) zurückgeliefert werden. Fehlt dieses Element, so wird der Signaturtyp gemäß TAB_KON_583 – Default-Signaturverfahren aus dem Dokumententyp abgeleitet.

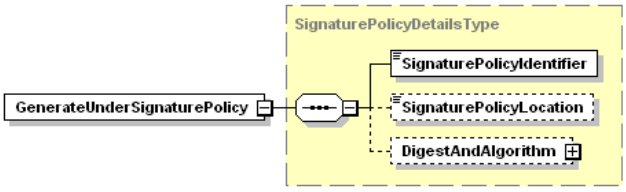
	dss:
	Properties
		Durch dieses in [OASIS-DSS] (Abschnitt 3.5.5) definierte Element können zusätzliche signierte und unsignierte Eigenschaften (Properties) bzw. Attribute in die Signatur eingefügt werden.
		Unterstützt werden genau folgende Attribute:
		Im CMS-Fall (SignatureType = urn:ietf:rfc:5652) kann es XML-Elemente
		`./SignedProperties/Property/Value/CMSAttribute`
		und
		`./UnsignedProperties/Property/Value/CMSAttribute`
		enthalten. Ein solches XML-Element `CMSAttribute` muss ein vollständiges, base64/DER-kodiertes ASN.1-`Attribute` enthalten, definiert in [CMS#5.3.SignerInfo Type]. Es muss bei der Erstellung des CMS-Containers unverändert unter `SignedAttributes bzw. UnsignedAttributes` aufgenommen werden.
		Die Übergabe der Attribute
		• `ContentType`
		• `SigningTime`
		• `MessageDigest`
		• `SigningCertificate` und `SigningCertificateV2`
		wird ignoriert und es wird die Warnung 4273 zurück gegeben.

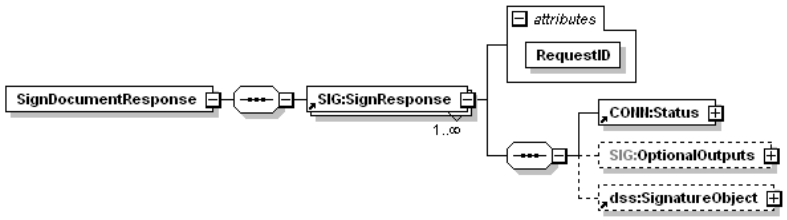
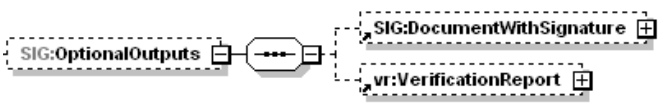
	SIG: Include EContent	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.7), definierte Element kann bei einer CMS-basierten Signatur das Einfügen des signierten Dokumentes in die Signatur angefordert werden. Die Verwendung dieses Parameters bei anderen Signaturtypen führt zu einem Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante).
	SIG: Include Object	Dieses Element enthält zum Anfordern einer Enveloping XML Signatur ein dss:IncludeObject-Element gemäß [OASIS-DSS] (Abschnitt 3.5.6). Ist das Element vorhanden und ein anderer Signaturtyp als eine XML-Signatur angefordert, so wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.

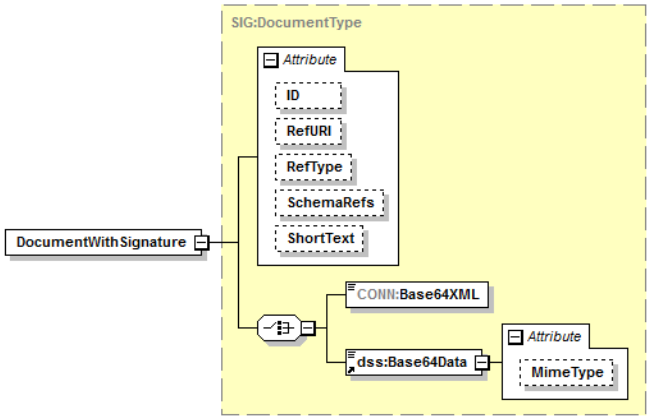
	dss:SignaturePlacement	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.8) definierte Element kann bei XML-basierten Signaturen gemäß [RFC3275] die Platzierung der Signatur im Dokument angegeben werden. Die in [OASIS-DSS] (Abschnitt 2.5, XPath c) beschriebene Deklaration von Namespace-Prefixes im dss:SignaturePlacement-Element muss nicht unterstützt werden. Bei anderen Signaturtypen wird das Element ignoriert und eine Warnung (Fehlercode 4197, Parameter SignaturePlacement wurde ignoriert) zurückgeliefert. dss:SignaturePlacement darf nur zusammen mit einer unterstützten Signaturrichtlinie verwendet werden (sp:SignaturePolicyIdentifier muss entsprechend gesetzt sein).
	dss:ReturnUpdatedSignature	Durch dieses in [OASIS-DSS] (Abschnitt 4.5.8) definierte Element kann eine übergegebene XML- oder CMS-Signatur mit zusätzlichen Informationen und Signaturen (Parallel- und Gegensignaturen) versehen werden. Hierbei sind folgende Ausprägungen für das Type-Attribut vorgesehen: • http://ws.gematik.de/conn/sig/sigupdate/parallel Hierdurch wird eine Parallelsignatur zu einer bereits existierenden Signatur erzeugt und entsprechend zurückgeliefert. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt. Bei anderen Type-Attributen wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.
	dss:Schemas	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schemata übergeben werden, die zur Validierung der übergebenen XML-Dokumente verwendet werden können.



`dss:Schema`	Dieses Element enthält ein XML-Schema zur Validierung des übergebenen XML-Dokuments. Das Attribut `RefURI` ist verpflichtend. Es kennzeichnet dabei den Namensraum des XML-Schemas entsprechend [OASIS-DSS] (Abschnitt 2.8.5)

`sp:GenerateUnderSignaturePolicy`	
	Über dieses in [OASIS-SP], Kapitel 2.2.1.1.1.1 Optional Input <GenerateUnderSignaturePolicy>, definierte Element wird die erforderliche Singnaturrichlinie ausgewählt.
	Die im Element `sp:SignaturePolicyIdentifier` übergebene URI identifiziert die Signaturrichtlinie.
	Die XML-Elemente
	`SignaturePolicyLocation`
	`DigestAndAlgorithm`
	werden nicht verwendet.
	Wenn eine nach TAB_KON_778 notwendige Signaturrichtlinie fehlt oder die übergebene Signaturrichtlinie unbekannt ist, wird Fehler

		4111 zurückgeliefert.
	SIG: Viewer Info	Enthält optional die vom Konnektor in die Signatur einzubeziehende Referenzen für die Stylesheets zur Anzeige.
Rückgabe		
	SIG: SignResponse	Eine SignResponse kapselt den ausgeführten Signaturauftrag pro Dokument. Die Zuordnung zwischen SignRequest und SignResponse erfolgt über die RequestID.
	CONN: Status	Enthält den Status der ausgeführten Operation pro SignRequest.
	SIG: OptionalOutputs	Enthält (angelehnt an dss:OptionalOutputs) optionale Ausgangsparameter: 

	SIG: Document With Signature	 Pro SignResponse wird ein Element SIG:DocumentWithSignature gemäß [OASIS-DSS] (Abschnitt 3.5.8) zurückgeliefert, in dem das Dokument mit Signatur enthalten ist. Dabei werden die XML-Attribute des Elements SIG:Document auf dem zugehörigen SignRequest übernommen. Ist die Signatur nicht im Dokument enthalten, wird ein leeres Element Base64XML oder Base64Data zurückgegeben. Die Signatur wird dann im Element dss:SignatureObject abgelegt. Wenn die Signatur im Dokument enthalten ist, wird das signierte Dokument im Feld Base64XML bzw. Base64Data zurückgeliefert. In diesem Fall MUSS die dss:SignaturePtr-Alternative in dss:SignatureObject (vgl. [OASIS-DSS] Abschnitt 2.5) dazu genutzt werden, auf die in den Dokumenten enthaltenen Signaturen zu verweisen.
	vr: Verification Report	Vom Konnektor nicht befüllt.
	dss: SignatureObject	Enthält im Erfolgsfall die erzeugte Signatur pro SignRequest in Form eines dss:SignatureObject-Elementes gemäß [OASIS-DSS] (Abschnitt 3.2).
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Der Ablauf der Operation SignDocument ist in Tabelle TAB_KON_756 Ablauf Operation SignDocument (nonQES und QES) beschrieben:

Tabelle 232: TAB_KON_756 Ablauf Operation SignDocument (nonQES und QES)

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Anhand des Kartentyps wird ermittelt, ob eine QES oder eine nonQES erzeugt werden soll. Alle übergebenen Parameterwerte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 {
		mandatId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		cardHandle = $context.cardHandle;
		userId = $context.userId }

Im Fall QES wird Schritt 4 ausgeführt. Im Fall nonQES wird Schritt 5 ausgeführt.		
4a)	Prüfe Signaturdienst-Modul	Prüfe, ob MGM_LU_SAK=Enabled. Ist dies nicht der Fall, so bricht die Operation mit Fehler 4125 ab.
Wenn für die CardSession die Komfortsignatur aktiviert ist (CARDSESSION.SIGNMODE = Comfort) wird Schritt 4 c) ausgeführt. Andernfalls wird Schritt 4 b) ausgeführt.		
4b)	TUC_KON_150 „Dokumente QES signieren“	Die QES wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.
4c)	TUC_KON_170 „Dokumente mit Komfort signieren“	Eine Komfortsignatur wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.

5)	TUC_KON_160 „Dokumente nonQES signieren“	Die nonQES wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.

Tabelle 233: TAB_KON_757 Fehlercodes „SignDocument (nonQES und QES)“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4111	Technical	Error	ungültiger Signaturtyp oder Signaturvariante
4126	Security	Error	Kartentyp nicht zulässig für Signatur
4125	Technical	Error	LU_SAK nicht aktiviert
4197	Technical	Warning	Parameter SignaturePlacement wurde ignoriert
4252	Technical	Error	Jobnummer wurde in den letzten 1.000 Aufrufen bereits verwendet und ist nicht zulässig
4273	Technical	Warning	Attribute im Parameter dss:Properties wurden ignoriert

Die zulässigen Zertifikate und Schlüssel sind in TAB_KON_900 aufgelistet.
[<=]

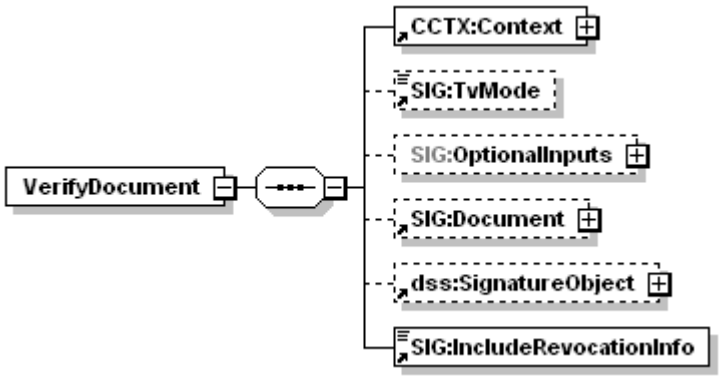
4.1.8.5.2 VerifyDocument (nonQES und QES)

TIP1-A_5034-04 - Operation VerifyDocument (nonQES und QES)
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation VerifyDocument (nonQES und QES) anbieten.

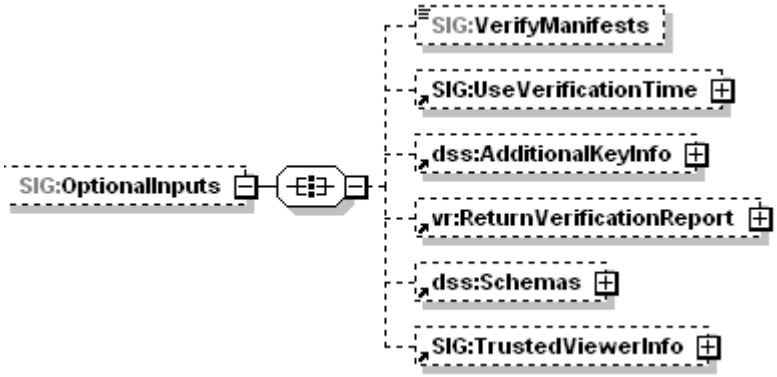
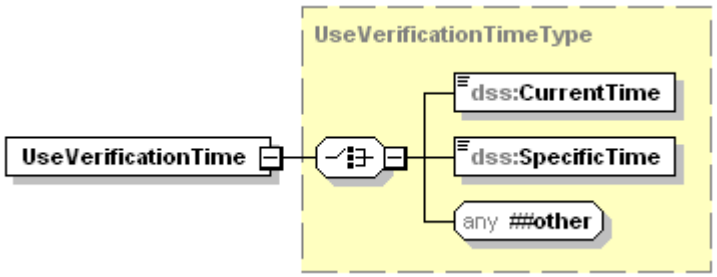
Tabelle 234: TAB_KON_066 Operation VerifyDocument (nonQES und QES)

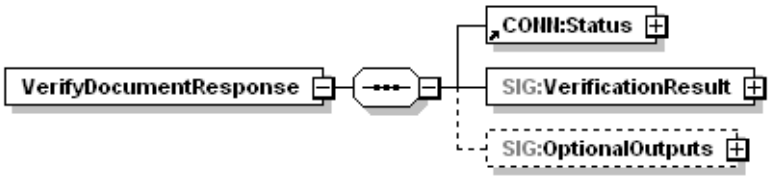
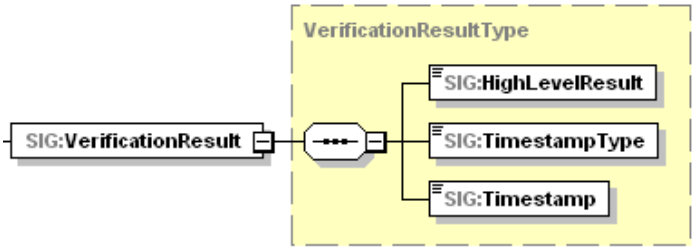
Name	`VerifyDocument`
Beschreibung	Diese Operation verifiziert die Signatur eines Dokumentes. Der Konnektor MUSS jede konform zur Außenschnittstelle SignDocument erzeugte Signatur durch VerifyDocument prüfen können. Das Ergebnis der Prüfung wird, wenn gefordert, in Form eines standardisierten Prüfberichts in einer `VerificationReport`-Struktur gemäß [OASIS-VR] zurückgeliefert.

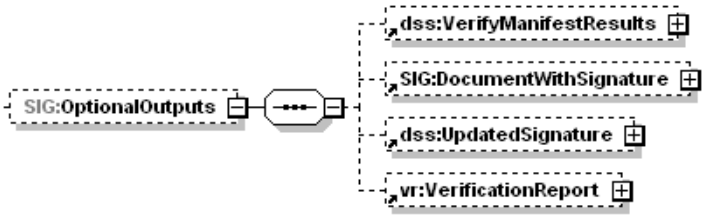
Aufruf-parameter



Name	Beschreibung
`CCTX: Context`	MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
`TvMode`	Der Parameter wird im Konnektor nicht ausgewertet.
`SIG: OptionalInputs`	Enthält optionale Eingabeparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7): Die zulässigen optionalen Eingabeparameter sind unten erläutert.
`SIG: Document`	Enthält im Fall der Prüfung von detached oder enveloped Signaturen das zur Signatur gehörende bzw. das diese umschließende Dokument (siehe [OASIS-DSS] Section 2.4.2 und oben).
`dss: SignatureObject`	Enthält die zu prüfende Signatur, wenn sie nicht im Dokument selbst eingebettet ist ([OASIS-DSS] Kapitel 4.1). Hierbei werden **XML-Signaturen** als `ds:Signature` Element und alle anderen Signaturen als `dss:Base64Signature` mit entsprechend gesetztem `Type`-Attribut (siehe `SignatureType`, Operation SignDocument) übergeben, wobei die nachfolgenden Werte unterstützt werden müssen: • **CMS-Signatur** urn:ietf:rfc:5652 • **S/MIME-Signatur** urn:ietf:rfc:5751 • **PDF-Signatur** http://uri.etsi.org/02778/3
`SIG: IncludeRevocationInfo`	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturprüfung vorliegenden Sperrinformationen anfordern. Ist bereits eine Sperrinformation eingebettet, so wird die neue Sperrinformation zusätzlich

		eingebettet. Für in einer Gegensignatur enthaltene Signaturen erfolgt keine Einbettung von Sperrinformationen. Für PDF-Signaturen erfolgt keine Einbettung von Sperrinformationen. Der Konnektor nimmt die Warnung 4261 in die Antwort auf.
		
	SIG: Verify Mani fests	Durch das in [OASIS-DSS] (Abschnitt 4.5.1) definierte Element kann die Prüfung eines ggf. vorhandenen Manifests angefordert werden.
		
	SIG: Use Verifi cation Time	Durch das in [OASIS-DSS] (Abschnitt 4.5.2) spezifizierte Element kann die Prüfung der Signatur bezüglich eines durch den Aufrufer bestimmten Zeitpunktes (`Benutzerdefinierter_Zeitpunkt`) erfolgen.
	dss: Addit ional KeyInfo	Durch das in [OASIS-DSS] (Abschnitt 4.5.4) spezifizierte Element kann zusätzliches, für die Prüfung benötigtes, Schlüsselmaterial übergeben werden.
	vr: Return Verifi cation Report	Durch dieses in [OASIS-VR] spezifizierte Element kann die Erstellung eines ausführlichen Prüfberichtes angefordert werden. Der Konnektor MUSS die Anforderungen der Konformitätsstufe 2 („Comprehensive“) erfüllen und die Profilierung aus Anhang B3 beachten.

	dss:Schemas	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schematas übergeben werden, die zur Validierung des übergebenen XML-Dokumentes verwendet werden können. Zur Struktur dieses Elements siehe Beschreibung des Parameters dss:Schemas der Operation SignDocument.
	SIG:ViewerInfo	Der Parameter wird im Konnektor nicht ausgewertet.
Rückgabe		
	Status	Enthält den Ausführungsstatus der Operation.
	SIG:VerificationResult	 Das Element Sig:VerificationResult enthält das Ergebnis der Prüfung als Ampel, den Typ des zugehörigen angenommenen Signaturzeitpunkts und der angenommene Signaturzeitpunkt selbst.
	SIG:HighLevelResult	Das Ergebnis der Prüfung (Ampelschaltung) mit folgenden Werten: • VALID: alle Signaturen sind gültig • INVALID: mindestens eine der Signaturen ist ungültig • INCONCLUSIVE: in allen anderen Fällen
	SIG:TimestampType	Der Typ des angenommenen Signaturzeitpunkts mit folgenden Werten: • SIGNATURE_EMBEDDED_TIMESTAMP: in der Signatur eingebetter Zeitpunkt Ermittelter_Signaturzeitpunkt_Eingebettet • • SYSTEM_TIMESTAMP: Systemzeit des Konnektors bei

		Signaturprüfung `Ermittelter_Signaturzeitpunkt_System` • USER_DEFINED_TIMESTAMP: benutzerdefinierter Zeitpunkt `Benutzerdefinierter_Zeitpunkt` Als Format darf jedes zum XML-Typ "dateTime" konforme Format verwendet werden (<element name="Timestamp" type="dateTime"/>). Wenn mehrere Signaturen im Dokument vorhanden sind, wird hier der angenommene Signaturzeitpunkt der jüngsten Signatur angegeben.
	`SIG: Time stamp`	Im Element SIG:Timestamp wird der zu SIG:TimestampType gehörende Zeitstempel zurückgegeben.
	`SIG: Optio nal Outputs`	Enthält (angelehnt an dss:OptionalOutputs, wie in Abschnitt 2.7 von [OASIS-DSS] beschrieben) optionale Ausgangselemente: 
	`dss: Verify Manifest Results`	Dieses in Abschnitt 4.5.1 von [OASIS-DSS] definierte Element enthält Informationen zur Prüfung eines ggf. vorhandenen Signaturmanifests und wird zurückgeliefert, sofern beim Aufruf das `dss:VerifyManifest`-Element, aber nicht das `RequestVerificationReport` als optionales Eingabeelement übergeben wurde.
	`SIG: Document With Signa ture`	Dieses in Abschnitt 4.5.8 von [OASIS-DSS] spezifizierte Element wird zurückgeliefert, falls eine in dem Dokument enthaltene Signatur (Enveloped Signature) in Verbindung mit dem `SIG:IncludeRevocationInfo`-Element geprüft wurde.
	`dss: Updated Signa ture`	Dieses in Abschnitt 4.5.8 von [OASIS-DSS] spezifizierte Element wird zurückgeliefert, falls eine abgesetzte (Detached Signature) oder umschließende (Enveloping Signature) in Verbindung mit dem `SIG:IncludeRevocationInfo-` Element geprüft wurde.
	`vr: Verifi cation Report`	Dieses in [OASIS-VR] spezifizierte Element wird zurückgeliefert, falls das `ReturnVerificationReport`-Element als

		Eingabeparameter verwendet wurde. Die Profilierung von Anhang B3 MUSS beachtet werden.
Vorbe-dingungen	Keine	
Nachbe-dingungen	Keine	

Tabelle 235: TAB_KON_760 Ablauf Operation VerifyDocument (nonQES und QES)

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession= false;
		}
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	prüfe, ob QES oder nonQES	Ist im jeweiligen Signaturzertifikat mindestens ein QCStatement mit dem OID id-etsi-qcs-QcCompliance (0.4.0.1862.1.1) enthalten, handelt es sich um eine QES-Signatur, andernfalls liegt eine nonQES-Signatur vor.
Für QES-Signaturen wird Schritt 4 ausgeführt. Für nonQES-Signaturen wird Schritt 5 ausgeführt.		
4.a	Prüfe Signaturdienst-Modul	Prüfe, ob MGM_LU_SAK=Enabled. Ist dies nicht der Fall, so bricht die Operation mit Fehler 4125 ab.
4.b	TUC_KON_151 „QES Dokumentensignatur prüfen“	Die QES wird geprüft. Tritt hierbei ein Fehler auf, bricht die Operation ab.
5.	TUC_KON_161 „nonQES Dokumentensignatur prüfen“	Die nonQES wird geprüft. Tritt hierbei ein Fehler auf, bricht die Operation ab.

Tabelle 236: TAB_KON_761 Fehlercodes „VerifyDocument (nonQES und QES)“

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen TUCs (siehe Tabelle TAB_KON_760 Ablauf Operation VerifyDocument) können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4261	Technical	Warning	Einbettung von Revocation-Informationen nicht unterstützt
4125	Technical	Error	LU_SAK nicht aktiviert

[<=]

4.1.8.5.3 StopSignature

TIP1-A_5666 - Operation StopSignature (nonQES und QES)
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation StopSignature anbieten.

Tabelle 237: TAB_KON_840 Operation StopSignature

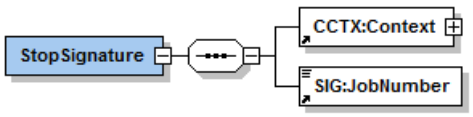
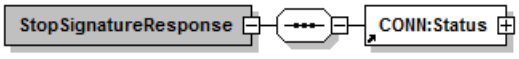
<table>
<tr><td>Name</td><td colspan="2">StopSignature</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Diese Operation unterbricht die Signatur eines Dokumentenstapels.
Der Konnektor MUSS jede Signaturerstellung für ein Dokumentenstapel unterbrechen können.</td></tr>
<tr><td rowspan="4">Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>CCTX:Context</td><td>MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet</td></tr>
<tr><td>SIG:
JobNumber</td><td>Die Nummer des Jobs, der gestoppt werden soll.</td></tr>
<tr><td rowspan="2">Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td>CONN:Status</td><td>Enthält den Ausführungsstatus der Operation.</td></tr>
<tr><td>Vorbedingungen</td><td colspan="2">Keine</td></tr>
<tr><td>Nachbedingungen</td><td colspan="2">Keine</td></tr>
</table>

Tabelle 238: TAB_KON_841 Ablauf Operation StopSignature

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	Stoppe die Stapelsignaturverarbeitung	Die Verarbeitung der Stapelsignatur wird abgebrochen

Tabelle 239: TAB_KON_842 Fehlercodes „StopSignature"

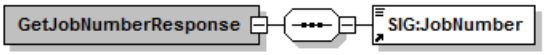
Fehlercode	ErrorType	Severity	Fehlertext
Folgende Fehlercodes können auftreten:			
4000	Technical	Error	Syntaxfehler
4243	Technical	Error	Jobnummer unbekannt

[<=]

4.1.8.5.4 GetJobNumber

TIP1-A_5667 - Operation GetJobNumber
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation GetJobNumber anbieten.

Tabelle 240: TAB_KON_843 Operation GetJobNumber

Name	GetJobNumber	
Beschreibung	Diese Operation liefert eine Jobnummer zur Verwendung in der Operation SignDocument. Die Jobnummer MUSS nach den Vorgaben von Kapitel 4.1.8.1.4 erstellt werden.	
Aufrufparameter		
	Name	Beschreibung
	CCTX:Context	MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
Rückgabe		
	SIG: JobNumber	Jobnummer zur Verwendung in „SignDocument"
Vorbedingungen	Keine	

Nachbedingungen	Keine

Tabelle 241: TAB_KON_844 Ablauf Operation GetJobNumber

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	Generiere und liefere eine Jobnummer	Eine innerhalb von 1000 Aufrufen eindeutige Jobnummer wird generiert und geliefert. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext.

Tabelle 242: TAB_KON_845 Fehlercodes „GetJobNumber"

Fehlercode	ErrorType	Severity	Fehlertext
Folgende Fehlercodes können auftreten:			
4000	Technical	Error	Syntaxfehler

[<=]

4.1.8.5.5 ActivateComfortSignature

A_19107 - Operation ActivateComfortSignature
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation ActivateComfortSignature anbieten.

Tabelle 243: TAB_KON_874 ActivateComfortSignature

Name	`ActivateComfortSignature`
Beschreibung	Diese Operation aktiviert die Komfortsignatur für einen HBA bezogen auf einen Aufrufkontext.

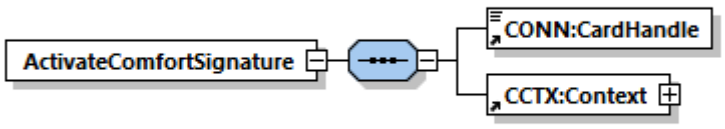
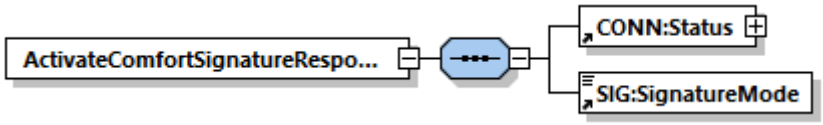
Aufrufparameter		
	Name	Beschreibung
	`CONN: CardHandle`	Identifiziert die zu adressierende Karte. Es wird nur der HBA unterstützt.
	`CCTX:Context`	MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend zu übergeben; MandantId, WorkplaceId nicht ausgewertet
Rückgabe		
	`CONN:Status`	Enthält den Ausführungsstatus der Operation.
	`SIG: SignatureMode`	Signaturmodus des HBA Enthält bei erfolgreicher Ausführung der Operation den Wert „COMFORT“
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Tabelle 244: TAB_KON_877 Ablauf ActivateComfortSignature

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffsberechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 { mandatId =$context.mandantId;

		clientsystemId = $context.clientsystemId;
		cardHandle = $context.cardHandle;
		userId = $context.userId }

4.	TUC_KON_171 „Komfortsignatur einschalten“	Der Komfortsignaturmodus wird für das Tupel (CardHandle, CardSession) eingeschaltet. Tritt hierbei ein Fehler auf, bricht die Operation ab.

Tabelle 245: TAB_KON_879 Fehlercodes ActivateComfortSignature

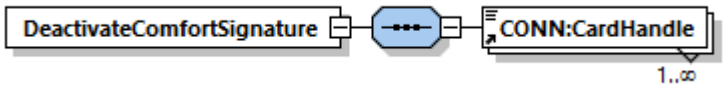
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4270	Technical	Error	UserId wurde in den letzten 1.000 Vorgängen bereits verwendet
4272	Technical	Error	UserId nicht zulässig

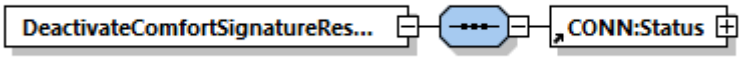
[<=]

4.1.8.5.6 DeactivateComfortSignature

A_19108 - Operation DeactivateComfortSignature
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation DeactivateComfortSignature anbieten.

Tabelle 246: TAB_KON_875 DeactivateComfortSignature

<table>
<tr><td>Name</td><td colspan="2">DeactivateComfortSignature</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Diese Operation deaktiviert die Komfortsignatur für einen oder mehrere HBA.</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 

<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>CONN:
Card
Handle</td><td>Identifiziert die zu adressierende Karte.
Es wird nur der HBA unterstützt.</td></tr>
</table></td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
</table>

	CONN:Status	Enthält den Ausführungsstatus der Operation.
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Tabelle 247: TAB_KON_878 Ablauf DeactivateComfortSignature

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	TUC_KON_172 „Komfortsignatur ausschalten“	Der Komfortsignaturmodus wird für alle Karten aus der CardHandle-Liste ausgeschaltet.

Tabelle 248: TAB_KON_880 Fehlercodes DeactivateComfortSignature

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

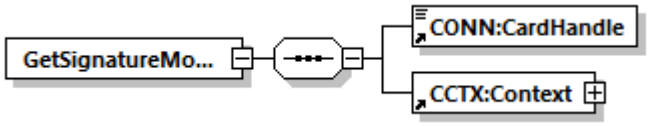
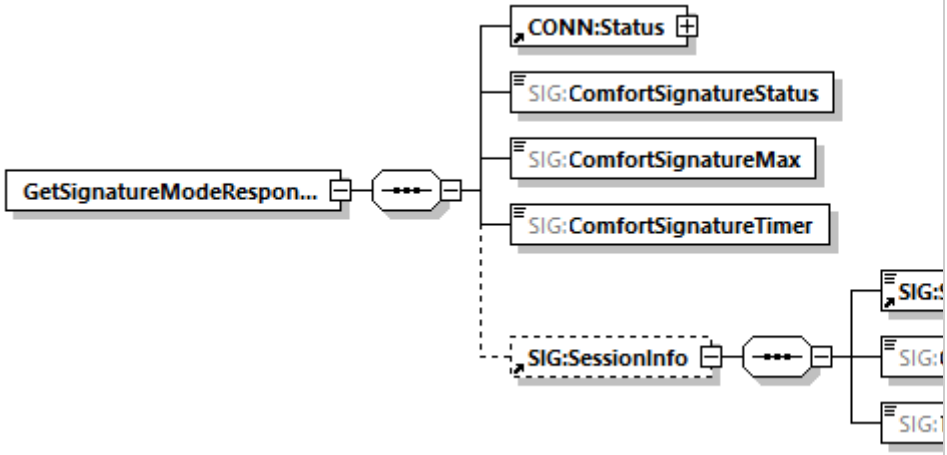
[<=]

4.1.8.5.7 GetSignatureMode

A_19109-01 - Operation GetSignatureMode
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation GetSignatureMode anbieten.

Tabelle 249: TAB_KON_876 GetSignatureMode

Name	GetSignatureMode
Beschreibung	Diese Operation liefert den aktuell konfigurierten Status der Komfortsignaturfunktion im Konnektor und, falls vorhanden, Informationen zu der aktuell im Konnektor existierenden Komfortsignatursession für das CardHandle und den Aufrufkontext.

Aufrufparameter		
	Name	Beschreibung
	CONN:CardHandle	Identifiziert die zu adressierende Karte. Es wird nur der HBA unterstützt.
	CCTX:Context	MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend zu übergeben
Rückgabe		
	CONN:Status	Enthält den Ausführungsstatus der Operation.
	SIG: ComfortSignatureStatus	Komfortsignatur-Konfigurationsstatus des Konnektors
	SIG:ComfortSignatureMax	Im Konnektor konfigurierte Anzahl von Komfortsignaturen, die ohne erneute PIN-Eingabe ausgeführt werden dürfen
	SIG: ComfortSignatureTimer	Im Konnektor konfiguriertes Zeitintervall, in dem Komfortsignaturen ohne erneute PIN-Eingabe ausgeführt werden dürfen, Format: "PTnHnMnS" (gemäß Datenttyp xsd:duration)

	SIG:SessionInfo	Falls vorhanden, Informationen zu der aktuell im Konnektor existierenden Komfortsignatursession für das CardHandle und den Aufrufkontext
	SIG:SignatureMode	Signaturmodus der Komfortsignatursession (="Comfort")
	SIG:CountRemaining	Verbleibende Anzahl von Komfortsignaturen, die ohne erneute PIN-Eingabe ausgeführt werden dürfen
	SIG:TimeRemaining	Verbleibende Zeit, in der Komfortsignaturen ohne erneute PIN-Eingabe ausgeführt werden dürfen Format: "PTnHnMnS" (gemäß Datenttyp xsd:duration)
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Tabelle 250: TAB_KON_882 Ablauf GetSignatureMode

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffsberechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 {
		mandatId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		cardHandle = $context.cardHandle;
		userId = $context.userId }

4.	TUC_KON_173 „Liefere Signaturmodus“	Der Komfortsignatur-Konfigurationsstatus des Konnektors und der im Konnektor hinterlegte Signaturmodus werden für den dem Konnektor bekannten Aufrufkontext des HBA aus dem übergebenen CardHandle zurück geliefert.

Tabelle 251: TAB_KON_881 Fehlercodes GetSignatureMode

Fehlercode	ErrorType	Severity	Fehlertext
Folgende Fehlercodes können auftreten:			
4000	Technical	Error	Syntaxfehler

[<=]

4.1.8.6 Betriebsaspekte

TIP1-A_4680-03 - Konfigurationswerte des Signaturdienstes
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_596 vorzunehmen:

Tabelle 252: TAB_KON_596 Konfigurationswerte des Signaturdienstes (Administrator)

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
SAK_SIMPLE_ SIGNATURE_ MODE	`SE#1` `SE#2`	Aktivierung/Deaktivierung des „Einfachsignaturmodus“ für alle HBAx für die Durchführung von Einfachsignaturen im SecurityEnvironment #1 (SE#1) für Dokumentenstapel der Größe 1 anstelle der Verwendung des SE#2. Default-Wert = SE#1
SAK_COMFORT_ SIGNATURE	`Enabled/` `Disabled`	Aktivierung/Deaktivierung der Komfortsignaturfunktion im Konnektor Default-Wert = Disabled Die Komfortsignaturfunktion darf nur aktiviert sein, wenn ANCL_TLS_MANDATORY = Enabled und ANCL_CAUT_MANDATORY = Enabled

SAK_COMFORT_ SIGNATURE_MAX	[1 – 250]	Anzahl von Komfortsignaturen, die ohne erneute PIN-Eingabe ausgeführt werden dürfen Default-Wert = 100 Der Parameter ist nur relevant, wenn die Komfortsignaturfunktion aktiviert ist (SAK_COMFORT_SIGNATURE = Enabled).
SAK_COMFORT_ SIGNATURE_TIMER	[1 – 24 h]	Zeitintervall, in dem Komfortsignaturen ohne erneute PIN-Eingabe ausgeführt werden dürfen Der Timer startet mit Eingabe der PIN.QES für die Komfortsignatur. Default-Wert = 6 h Der Parameter ist nur relevant, wenn die Komfortsignaturfunktion aktiviert ist (SAK_COMFORT_SIGNATURE = Enabled).

[<=]

4.1.9 Zertifikatsdienst

Der Zertifikatsdienst bietet eine Schnittstelle zur Überprüfung der Gültigkeit von Zertifikaten an. Dies geschieht auf Grundlage des durch den Vertrauensanker (TSL-CA-Signer-Zertifikat und eine aktuelle, gültige TSL aufgespannten Vertrauensraums sowie unter Berücksichtigung von aktuellen Statusinformationen (OCSP, CRL). Die Zertifikatsprüfung wird sowohl für nonQES- als auch für QES-Zertifikate unterstützt.

Die für die QES-Zertifikatsprüfung notwendigen QES-Signer-Zertifikate werden durch die Vertrauensliste der Bundesnetzagentur (BNetzA-VL) bereitgestellt. Das Signer-Zertifikat der BNetzA-VL ist in der TSL enthalten.

Im Rahmen der ECC-Migration muss der Konnektor neben RSA auch ECC unterstützen. Hierfür wird eine TSL bereitgestellt, die sowohl die neuen ECC-basierten Zertifikate als auch aus Rückwärtskompatibilitätsgründen die weiterhin benötigten RSA-basierten Zertifikate enthält. Diese neue TSL wird auch als „TSL(ECC-RSA)“ bezeichnet. In dieser Spezifikation wird außerhalb der Regelungen zur ECC-Migration nicht zwischen „TSL(ECC-RSA)" und „TSL(RSA)" unterschieden, da die Anforderungslage keine Unterscheidung erfordert.

Innerhalb des Zertifikatsdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „CERT“
- Konfigurationsparameter: „CERT_“

4.1.9.1 Funktionsmerkmalweite Aspekte

Bei der Zertifikatsprüfung wird im Rahmen eines Anwendungsfalls u.a. auch der Verwendungszweck des Zertifikats geprüft. Der Verwendungszweck (intendedKeyUsage) wird als Parameter an TUC_KON_037 übergeben. Der konkrete Wert von intendedKeyUsage ist abhängig vom kryptographischen Verfahren, auf welchem das Zertifikat basiert. Die Parametrisierung von intendedKeyUsage wird in TAB_KON_853

in Abhängigkeit vom zu prüfenden Zertifikat, dem Anwendungsfall und dem kryptographischen Verfahren definiert.

A_17295 - Verwendung der intendedKeyUsage bei der Zertifikatsprüfung (ECC-Migration)
Der Konnektor MUSS bei der Zertifikatsprüfung die intendedKeyUsage in Abhängigkeit vom zu prüfenden Zertifikat, dem Anwendungsfall und dem kryptographischen Verfahren gemäß TAB_KON_853 prüfen.

Tabelle 253: TAB_KON_853- intendedKeyUsage bei Zertifikatsprüfung

Zertifikat	Anwendungsfall	intendedKeyUsage bei	
		RSA	ECC
C.SMKT.AUT	TUC_KON_050 „Beginne Kartenterminalsitzung“ TUC_KON_053 „Paire Kartenterminal“	digitalSignature &keyEncipherment	digitalSignature
C.CH.AUT C.CH.AUTN	TUC_KON_161 „nonQES Dokumentsignatur prüfen”	digitalSignature &keyEncipherment	digitalSignature
C.CH.ENC C.CH.ENCV C.HCI.ENC C.HP.ENC Zertifikate aus CERT_IMPORTED_CA_LIST	TUC_KON_070 „Daten hybrid verschlüsseln“	keyEncipherment	keyAgreement
C.HCI.OSIG	TUC_KON_161 „nonQES Dokumentsignatur prüfen”	nonRepudiation	nonRepudiation
C.FD.TLS-S	TUC_KON_110 „Kartenbasierte TLS-Verbindung aufbauen“	digitalSignature&keyEncipherment	digitalSignature
C.ZD.TLS-S	TUC_KON_290 „LDAP-Verbindung aufbauen“	digitalSignature	digitalSignature
C.ZD.TLS-S	TIP1-A_5662 - Gesicherte Übertragung von	digitalSignature&keyEncipherment	digitalSignature

	BNetzA-VL und Hashwert TUC_KON_282 „UpdateInformationen beziehen“ TUC_KON_283 Infrastruktur Konfiguration aktualisieren TUC_KON_285 „UpdateInformationen für Fachmodul beziehen“ TUC_KON_286 „Paket für Fachmodul laden“		
C.FD.AUT	A_17225	digitalSignature&keyEncipherment	digitalSignature

[<=]

Bei der Zertifikatsprüfung wird ein übergebenes Zertifikat oder ein Zertifikat einer referenzierten Karte geprüft. Das konkrete Zertifikatsobjekt einer Karte ist abhängig vom Kartentyp und dem gewählten kryptographischen Verfahren. Die folgende Tabelle führt auf, welche Zertifikatsobjekte einer Karte in Abhängigkeit vom kryptographischen Verfahren für die jeweilige Zertifikatsreferenz ausgewählt werden.

Tabelle 254: TAB_KON_858 Kartenobjekt in Abhängigkeit vom kryptographischen Verfahren

CertRef	Kartentyp	Objekt der Karte in Abhängigkeit vom kryptographischen Verfahren (Crypt)	
		RSA	ECC
C.AUT	HBA-VK	EF.C.HP.AUT	-
	HBA	EF.C.HP.AUT.R2048	EF.C.HP.AUT.E256
	SM-B	EF.C.HCI.AUT	EF.C.HCI.AUT.E256
	eGK G2	EF.C.CH.AUT.R2048	EF.C.CH.AUT.E256
C.ENC	HBA-VK	EF.C.HP.ENC	-
	HBA	EF.C.HP.ENC.2048	EF.C.HP.ENC.E256
	SM-B	EF.C.HCI.ENC.R2048	EF.C.HCI.ENC.E256

C.SIG	SM-B	EF.C.HCI.OSIG.R2048	EF.C.HCI.OSIG.E256
C.QES	HBA-VK	EF.C.HP.QES	-
	HBA	EF.C.HP.QES.R2048	EF.C.HP.QES.E256

TIP1-A_4682 - Sicheres Einbringen des TI-Vertrauensankers
Der Vertrauensanker der TI MUSS zum Auslieferungszeitpunkt des Konnektors integritätsgeschützt im Konnektor hinterlegt sein. Zur Sicherstellung dieser Integrität MUSS die Dateiablage EF.C.TSL.CA_1 der Anwendung DF.Sicherheitsanker der gSMC-K [gemSpec_gSMC-K_ObjSys#5.7.2] verwendet werden.
[<=**]**

TIP1-A_4684 - Regelmäßige Aktualisierung der CRL und der TSL
Falls Parameter MGM_LU_ONLINE=Enabled, MUSS der Zertifikatsdienst einmal täglich die Aktualisierung der TSL durch Aufruf von TUC_KON_032 „TSL aktualisieren" durchführen und anschließend TUC_KON_040 „CRL aktualisieren" aufrufen.
[<=**]**

TIP1-A_4685 - Vermeidung von Spitzenlasten bei TSL- und CRL-Download
Der Konnektor MUSS Spitzenlasten durch paralleles Herunterladen der TSL und der CRL vermeiden. Dazu MÜSSEN die im Einsatz befindlichen Konnektoren eines Herstellers ihre Download-Versuche gleichmäßig über den Tag verteilen.
[<=**]**

Dadurch wird gleichzeitig die Spitzenlast bei OCSP-Anfragen begrenzt.

A_17572 - Nutzung der Hash-Datei für TSL (ECC-Migration)
Falls die TSL(ECC-RSA) verwendet wird, MUSS der Konnektor vor deren Aktualisierung mit TUC_KON_032 „TSL aktualisieren" die Hash-Datei der TSL(ECC-RSA) herunterladen, um zu prüfen, ob die am TSL-Downloadpunkt verfügbare TSL(ECC-RSA) eine andere ist, als die schon zuvor heruntergeladene und bereits ausgewertete TSL(ECC-RSA). Entspricht der Hash-Wert am Download-Punkt der bereits heruntergeladenen und ausgewerteten TSL(ECC-RSA), MUSS der Konnektor auf den Download verzichten.**[**<=**]**

A_17661 - Gesicherte Übertragung der Hash-Datei für TSL (ECC-Migration)
Der Konnektor MUSS für den Download der Hash-Datei der TSL(ECC-RSA) die Verbindung zum TSL-Dienst durch TLS absichern. Der Konnektor MUSS das vom TSL-Dienst beim TLS-Verbindungsaufbau präsentierte Zertifikat C.ZD.TLS-S prüfen. Die Prüfung erfolgt durch Aufruf von TUC_KON_037 „Zertifikat prüfen" {

certificate = C.ZD.TLS-S;
qualifiedCheck = not_required;
offlineAllowNoCheck = true;
policyList = oid_zd_tls_s;
intendedKeyUsage = intendedKeyUsage(C.ZD.TLS-S);
intendedExtendedKeyUsage = id-kp-serverAuth;
validationMode = OCSP } .

Falls Fehler im TLS-Verbindungsaufbau bzw. bei der Zertifikatsprüfung auftreten MUSS der Konnektor den TLS-Verbindungsaufbau mit Fehlercode 4235 gemäß TAB_KON_825 abbrechen.
[<=**]**

A_17781 - Aktualisierung der TSL ohne Hash-Datei für TSL (ECC-Migration)
Falls im Rahmen der TSL-Aktualisierung beim Download der Hash-Datei der TSL(ECC-RSA) ein Fehler auftritt MUSS der Konnektor die Aktualisierung der TSL mit TUC_KON_032 „TSL aktualisieren" ohne einen ermittelten Hashwert aufrufen.**[<=]**

TIP1-A_6730 - Regelmäßige Aktualisierung der BNetzA-VL
Falls Parameter MGM_LU_ONLINE=Enabled, MUSS der Zertifikatsdienst die Aktualisierung der BNetzA-VL im Zeitintervall `CERT_BNETZA_VL_UPDATE_INTERVAL` durch Aufruf von TUC_KON_031 „BNetzA-VL aktualisieren" durchführen.
[<=]

TIP1-A_6731 - Regelmäßige Prüfung der BNetzA-VL
Der Zertifikatsdienst MUSS einmal täglich die zeitliche Gültigkeit der BNetzA-VL prüfen. Wenn das Element NextUpdate in der Vergangenheit liegt MUSS der Konnektor den Betriebszustand EC_ BNetzA_VL_not_valid auslösen.
[<=]

TIP1-A_6732 - Vermeidung von Spitzenlasten bei BNetzA-VL-Download
Der Konnektor MUSS Spitzenlasten durch Herunterladen der BNetzA-VL vermeiden. Dazu MÜSSEN die im Einsatz befindlichen Konnektoren den Zeitpunkt für den Download zufällig wählen unter Beachtung des konfigurierten Zeitintervalls `CERT_BNETZA_VL_UPDATE_INTERVAL`.
[<=]

TIP1-A_5662 - Gesicherte Übertragung von BNetzA-VL und Hashwert
Der Konnektor MUSS für den Download der BNetzA-VL und deren Hashwert die Verbindung zum TSL-Dienst durch TLS absichern. Der Konnektor MUSS das vom TSL-Dienst beim TLS-Verbindungsaufbau präsentierte Zertifikat ID.ZD.TLS_S prüfen. Die Prüfung erfolgt durch Aufruf von TUC_KON_037 „Zertifikat prüfen" {

certificate = ID.ZD.TLS_S;
qualifiedCheck = not_required;
offlineAllowNoCheck = true;
policyList = oid_zd_tls_s;
intendedKeyUsage = intendedKeyUsage(C.ZD.TLS-S);
intendedExtendedKeyUsage = id-kp-serverAuth;
validationMode = OCSP } .

Fehler im TLS-Verbindungsaufbau bzw. bei der Zertifikatsprüfung führen zum Abbruch des TLS-Verbindungsaufbaus mit Fehlercode 4235 gemäß TAB_KON_825.

Tabelle 255: TAB_KON_825 Fehlercodes „TLS-Verbindungsaufbau zum TSL-Dienst"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4235	Security	Error	TSL-Dienst konnte bei TLS-Verbindungsaufbau nicht authentisiert werden

[<=]

TIP1-A_5663 - Prüfung der technischen Rolle bei TLS-Verbindungsaufbau zum TSL-Dienst
Der Konnektor MUSS beim TLS-Verbindungsaufbau zum TSL-Dienst prüfen, dass die vom TSL-Dienst in ID.ZD.TLS_S übergebene technische Rolle gemäß [gemSpec_OID#GS-

A_4446] dem Wert „oid_tsl_ti“ entspricht.
Ein Fehler bei der Prüfung der techischen Rolle führt zum Abbruch des TLS-Verbindungsaufbaus mit Fehlercode 4236 gemäß TAB_KON_826.

Tabelle 256: TAB_KON_826 Fehlercodes „TLS-Verbindungsaufbau zum TSL-Dienst bei Prüfung der technischen Rolle“

Fehlercode	ErrorType	Severity	Fehlertext
4236	Security	Error	Rollenprüfung bei TLS-Verbindungsaufbau zum TSL-Dienst fehlgeschlagen

[<=]

TIP1-A_4686 - Warnung vor und bei Ablauf der TSL
Steht der Ablauf der TSL innerhalb von 7 Tagen an, MUSS der Konnektor den Betriebszustand EC_TSL_Expiring annehmen.
Mit Ablauf der Gültigkeit der TSL MUSS der Konnektor den Betriebszustand EC_TSL_Out_Of_Date_Within_Grace_Period annehmen.
Mit Ablauf der Graceperiod der TSL MUSS der Konnektor den kritischen Betriebszustand EC_TSL_Out_Of_Date_Beyond_Grace_Period annehmen.
[<=]

TIP1-A_4687 - Warnung vor und bei Ablauf des TI-Vertrauensankers
Steht der Ablauf der Gültigkeit des TI-Vertrauensankers innerhalb von 30 Tagen an, MUSS der Konnektor den Betriebszustand EC_TSL_Trust_Anchor_Expiring annehmen.
Mit Ablauf der Gültigkeit des Vertrauensankers MUSS der Konnektor den kritischen Betriebszustand EC_TSL_Trust_Anchor_Out_Of_Date annehmen.
[<=]

TIP1-A_4994 - Warnung vor und bei Ablauf der CRL
Steht der Ablauf der Gültigkeit der CRL innerhalb von 3 Tagen an, MUSS der Konnektor den Betriebszustand EC_CRL_Expiring annehmen.
Mit Ablauf der Gültigkeit der CRL MUSS der Konnektor den kritischen Betriebszustand EC_CRL_Out_Of_Date annehmen.
[<=]

TIP1-A_4688 - OCSP-Forwarding
Der Konnektor MUSS alle OCSP-Anfragen über den OCSP-Forwarder (HTTP-Proxy) des Zugangsdienst-Providers schicken, der durch die Konfigurationswerte (CERT_OCSP_FORWARDER_ADDRESS, CERT_OCSP_FORWARDER_PORT) festgelegt ist.
[<=]

TIP1-A_4689 - Caching von OCSP-Antworten
Der Zertifikatsdienst MUSS erhaltene OCSP-Antworten für eine durch `CERT_OCSP_DEFAULT_GRACE_PERIOD_NONQES` angegebene Anzahl an Minuten (nonQES-Zertifikate) zwischenspeichern.
[<=]

TIP1-A_4690 - Timeout und Graceperiod für OCSP-Anfragen
Bei Ausführung von TUC_PKI_006 „OCSP-Abfrage“ [gemSpec_PKI#8.3.2.2] MÜSSEN folgende Parameter verwendet werden:

OCSP-Graceperiod = `CERT_OCSP_DEFAULT_GRACE_PERIOD_NONQES`

- Timeout-Parameter = `CERT_OCSP_TIMEOUT_NONQES` bzw. `CERT_OCSP_TIMEOUT_QES`

[<=]

TIP1-A_4691 - Ablauf der gSMC-K und der gesteckten Karten regelmäßig prüfen
Für die gSMC-K sowie für jede gesteckte Karte außer eGK MUSS der Konnektor im Intervall `CERT_EXPIRATION_CARD_CHECK_DAYS` genau einmal TUC_KON_033 aufrufen. Der Konnektor MUSS die Gültigkeitsdauer der Zertifikate prüfen mittels Aufruf von:
für gSMC-K
TUC_KON_033{checkSMCK; doInformClients=Ja; crypt = ECC}
TUC_KON_033{checkSMCK; doInformClients=Ja; crypt = RSA}
für jede gesteckte G2.0 Karte außer eGK und außer gSMC-K
TUC_KON_033{cardSession; doInformClients=Ja; crypt = RSA}
für jede gesteckte ab G2.1 Karte außer eGK
TUC_KON_033{cardSession; doInformClients=Ja; crypt = ECC}
TUC_KON_033{cardSession; doInformClients=Ja; crypt = RSA}
[<=]

TIP1-A_4692 - Missbrauchserkennung, zu kontrollierende Operationen
Der Konnektor MUSS zur Unterstützung von Missbrauchserkennungen die in Tabelle TAB_KON_597 gelisteten Operationen als Einträge in EVT_MONITOR_OPERATIONS berücksichtigen.

Tabelle 257: TAB_KON_597 Operationen in EVT_MONITOR_OPERATIONS

Operationsname	OK_Val	NOK_Val	Alarmwert (Default-Grenzwert 10 Minuten- Σ)
`VerifyCertificate`	`1`	`5`	401

[<=]

4.1.9.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.1.9.3 Interne TUCs, nicht durch Fachmodule nutzbar

4.1.9.3.1 TUC_KON_032 „TSL aktualisieren"

TIP1-A_4693-02 - TUC_KON_032 „TSL aktualisieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_032 „TSL aktualisieren" umsetzen.

Tabelle 258: TAB_KON_766 TUC_KON_032 „TSL aktualisieren"

Element	Beschreibung
Name	TUC_KON_032 „TSL aktualisieren"
Beschreibung	Dieser TUC prüft die Aktualität der TSL und initialisiert ggf. den TSL-spezifischen Bereich des TrustStores neu. Zusätzlich wird bei einem Wechsel des TI-Vertrauensankers das neue TSL-Signer-CA-Zertifikat in einem sicheren Speicherort im Konnektor hinterlegt. Im Fall der Veröffentlichung eines CVC-

	Root-CA-Zertifikats werden das CVC-Root-CA-Zertifikat und die Cross-CV-Zertifikate aus der TSL in den Truststore eingestellt.
Auslöser	• Aufruf durch andere TUCs
Vorbedingungen	• Ein gültiger TI-Vertrauensanker ist vorhanden • Das XML-Schema der TSL-Datei liegt vor
Eingangsdaten	• importedTSL – *optional* (TSL aus aus manuellem Import) (Optional) • baseTime – *optional; default: aktuelles Datum* (Referenzzeitpunkt) () • onlineMode [ENABLED | DISABLED] (Flag „MGM_LU_ONLINE“ für Offline/Online-Modus) • hashTSL - *optional* (Hashwert-Datei der TSL im System; gilt nur für TSL(ECC-RSA))
Komponenten	Konnektor
Ausgangsdaten	• result (Status der Prüfung) • newHashTSL - *optional; verpflichtend für TSL(ECC-RSA)* (Hashwert-Datei der TSL im System; gilt nur für TSL(ECC-RSA))
Nachbedingungen	• Aktuelle TSL-Informationen inkl. des Vertrauensankers der BNetzA VL und sämtlicher CVC-Root-CA- und Cross-CV-Zertifikate liegen im Truststore vor. • Ein ggf. gelieferter neuer Vertrauensanker der TI ist in einem sicheren Speicherort gespeichert
Standardablauf	1. Der Konnektor prüft und aktualisiert ggf. die TSL durch Aufruf von TUC_PKI_001. Der Konnektor verwendet bei der Aktualisierung der TSL standardmäßig die Download-Punkte in der TI. Der durch den dort aufgerufenen TUC_PKI_011 „Prüfung des TSL-Signer-Zertifikates“ benötigte aktuelle TI-Vertrauensanker befindet sich auf der gSMC-K in der Datei EF.C.TSL_CA_1 oder in einem sicheren Speicherort im Konnektor. Es ist dasjenige Zertifikat zu verwenden, welches zum Referenzzeitpunkt gültig ist und ab dem Aktivierungsdatum (`StatusStartingTime` des neuen TSL-Signer-CA-Zertifikats) aktiviert ist. 2. Ggf. vorhandene CVC-Root-CA-Zertifikat und Cross-CV-Zertifikate werden genauso wie und zusammen mit den anderen CA-Zertifikaten aus der TSL extrahiert. 3. Alle Informationen aus der TSL werden im TSL-spezifischen Bereich des TrustStores gespeichert 4. Der Konnektor löst TUC_KON_256 { topic = „CERT/TSL/UPDATED”;

	eventType = Op;
	severity = Info;
	doLog = true;
	doDisp = false }
	aus.
	5. CERT_CRL_DOWNLOAD_ADDRESS wird mit den CRL-Download-Adressen aus der TSL überschrieben.

Varianten/ Alternativen	(-> 1) Wenn der Download der TSL aus der TI fehlschlägt oder wenn der Konnektor im Fall*onlineMode* = ENABLED keine Verbindung zur TI hat, muss der Konnektor die TSL vom Download-Punkt im Internet (CERT_TSL_DOWNLOAD_ADDRESS_INTERNET) gemäß [gemSpec_TSL#A_21182] beziehen. Im Fall onlineMode = DISABLED wird abgebrochen. Wenn kein aktiver VPN-Tunnel SIS vorhanden ist, muss der Konnektor den Downloadversuch der TSL aus dem Internet direkt über das IAG initiieren, auch wenn ANLW_INTERNET_MODUS=SIS konfiguriert ist. Im Fall ANLW_INTERNET_MODUS=KEINER wird abgebrochen. Wenn die Namensauflösung für CERT_TSL_DOWNLOAD_ADDRESS_INTERNET fehlschlägt, muss der Konnektor die TSL über CERT_TSL_IP_ADDRESS_INTERNET beziehen. Wenn keiner der vorigen Downloadversuche erfolgreich war, muss der Konnektor die TSL von der konfigurierbaren Adresse CERT_TSL_DOWNLOAD_ADDRESS_INTERNET_BU beziehen. Wenn die Namensauflösung für CERT_TSL_DOWNLOAD_ADDRESS_INTERNET_BU fehlschlägt, muss der Konnektor die TSL über CERT_TSL_IP_ADDRESS_INTERNET_BU beziehen. Für eine aus dem Internet bezogene TSL muss der Konnektor auch die vom TSL-Dienst gemäß [gemSpec_TSL#A_21182] bereitgestellte detached-Signatur der TSL herunterladen. Der Konnektor muss dann immer zunächst die detached-Signatur der TSL prüfen, einschließlich vollständiger Prüfung der Zertifikatskette bis zum TI-Vertrauensanker. Die kryptographische Prüfung der Signatur muss entsprechend A_21185 durchgeführt werden. Bezüglich der Prüfung des Sperrstatus des TSL-Signer-Zertifikats muss der Konnektor eines der folgenden Verfahren umsetzen: 1. Der Konnektor lädt die vorgefertigte OCSP-Response für das TSL-Signer Zertifikat aus dem Internet herunter (vgl. [gemSpec_TSL#A_21182]). Bei der Prüfung dieser OCSP-Response entfällt die Auswertung gegen die im System konfigurierte OCSP-Graceperiod. Der Konnektor prüft, dass die vorgefertigte OCSP-Response nicht älter als 61 Minuten ist. Die OCSP-Abfrage für das TSL-Signer Zertifikat in TUC_PKI_001, Schritt 4 entfällt. oder

	2. Standard OCSP-Abfrage für das TSL-Signer Zertifikat in TUC_PKI_001, Schritt 4, jedoch unter Verwendung des im Internet verfügbaren OCSP-Responders entsprechend [gemSpec_TSL#TIP1-A_4076-01]. (→1) Wird die *importedTSL* manuell übergeben (in den Eingangsdaten), so wird diese statt des Downloads verwendet und an TUC_PKI_001 übergeben. Innerhalb der PKI TUCs findet dann kein Download der TSL statt. (→1) Falls *onlineMode* = DISABLED, kann der Sperrstatus des TSL-Signer-Zertifikats nicht überprüft werden. In diesem Fall wird die Aktivierung der *importedTSL* auch ohne Prüfung des Sperrstatus durchgeführt. (→1) Wird durch den von TUC_PKI_001 aufgerufenen TUC_PKI_013 „Import neuer Vertrauensanker" ein neuer TI-Vertrauensanker (ein neues TSL-Signer-CA-Zertifikat) in der *importedTSL* gefunden, so wird dieser, wie dort beschrieben, extrahiert und in einem sicheren Speicherort gespeichert. Vor Erreichen des Aktivierungsdatums werden die TSLs ausschließlich mit dem alten TSL-Signer-Zertifikat signiert. Ab dem Aktivierungsdatum werden die TSLs mit einem TSL-Signer-Zertifikat signiert, das von der neuen TSL-Signer-CA ausgestellt wurde.

Fehlerfälle	(→1) Tritt beim manuellen Import der Datei ein Fehler auf, wird TUC_KON_256 {
	topic = „CERT/TSL/IMPORT";
	eventType = Op;
	severity = Error;
	parameters = „$Fehlerbeschreibung";
	doLog = true;
	doDisp = false }
	ausgelöst. Fehlercode 4128.
	(→1) Tritt beim periodischen Update der TSL beim Aufruf des TUC_PKI_001 oder eines durch ihn aufgerufenen TUCs ein Fehler auf, geht der Konnektor in den Betriebszustand EC_TSL_Update_Not_Successful. Der Konnektor geht erst in den Betriebszustand EC_TSL_Update_Not_Successful, wenn weder der Downloadversuch aus der TI noch der Downloadversuch aus dem Internet erfolgreich war. Die vorhandenen TSL-Vertrauensanker werden weiter verwendet. Fehlercode 4127.

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 259: TAB_KON_598 Fehlercodes TUC_KON_032 „TSL aktualisieren"

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4127	Security	Error	Import der TSL-Datei fehlgeschlagen
4128	Technical	Error	der manuelle Import der TSL-Datei schlägt fehl

[<=]

Für den Download der TSL über einen HTTP-Server im Internet wird zusätzlich zu der bereits mit einer XML-Signatur versehenen TSL eine detached-Signatur als separate Datei auf dem Download-Punkt zur Verfügung gestellt. Diese detached-Signatur umfasst die TSL in ihrerer Gänze, das heißt die TSL-XML-Datei wird inkl. der dort bereits enthaltenen XML-Signatur nochmal durch den TSL-Signer signiert. Ein Konnektor verwendet dann bei der Signaturprüfung einer TSL, die über das Internet bezogen wurde, die detached-Signatur für die Signaturprüfung. Hintergrund ist die aus Sicherheitsperspektive einfachere, im Sinne von sicherer prüfbare detached-Signatur. Das heißt, die TSL muss nicht als XML-File verarbeitet und die relativ komplexe XML-Signatur - die potentiell von einem Angreifer modifiziert sein könnte - nicht ausgewertet werden. Deshalb wird der Weg gewählt, der auch für die Signatur von X.509-Zertifikaten und OCSP-Responses verwendet wird.

A_21185 - Prüfung der detached Signatur der TSL bei Download aus dem Internet

Der Konnektor MUSS beim Download der TSL aus dem Internet ebenfalls deren detached-Signatur (vgl. [gemSpec_TSL#A_21182]) mit herunterladen und immer zunächst folgende Prüfungen durchführen:

1. Prüfung, dass die heruntergeladene detached-Signatur-Datei den folgenden Aufbau aufweist:
   Sequence aus drei Elementen:
   ```
   SEQUENCE {
     a
     b
     c}
   ```
   Mit *a*, *b* und *c* wie folgt:
   a. OID für den Signaturtyp, bestehend aus
      i. im Falle ECDSA:
      ```
      SEQUENCE {OBJECT IDENTIFIER ecdsaWithSHA256 (1 2 840 10045 4 3 2)}
      ```
      ii. im Falle RSASSA-PSS:
      ```
      SEQUENCE {OBJECT IDENTIFIER rsaPSS (1 2 840 113549 1 1 10)
          SEQUENCE {
            [0] {SEQUENCE {OBJECT IDENTIFIER sha-256 (2 16 840 1 101 3 4 2 1)}}
            [1] {SEQUENCE {
                      OBJECT IDENTIFIER pkcs1-MGF (1 2 840 113549 1 1 8)
                      SEQUENCE {OBJECT IDENTIFIER sha-256 (2 16 840 1 101 3 4 2 1)}}}
            [2] {INTEGER 32}}}
      ```
   b. Kryptographische Signatur

i. im Falle ECDSA:
einer ECDSA-Signatur nach [BSI-TR-03111#5.2.2.]

ii. im Falle RSASSA-PSS:
eine RSASSA-PSS-Signatur nach [RFC-8017] (reiner ASN.1-kodierter Signaturwert – die OID ist schon in Teil a.ii. aufgeführt)

c. Zertifikat des Signierenden (TSL-Signer)

i. im Falle ECDSA:
genau nur das ECC-TSL-Signer-Zertifikat

ii. im Falle RSASSA-PSS:
genau nur das RSA-TSL-Signer-Zertifikat

2. Prüfung der Signatur (1b) gegen das TSL-Signer-Zertifikat (1c).

Schlägt eine der Prüfungen fehl, MUSS der Import abgebrochen werden.
Ist die Prüfung erfolgreich, KANN die Prüfung der XML-Signatur der TSL im weiteren fachlichen Ablauf der TSL-Aktualisierung entfallen.
[<=]

Eine erweiterte Übersicht zum Aufbau der detached-Signatur-Datei inkl. Beispiel finden sie unter [gemGitHub_tslSig].

A_20750 - Hinweis auf Betreiber-Verantwortung bei automatischer TSL-Aktualisierung
Der Hersteller des Konnektors MUSS den Betreiber des Konnektors in geeigneter Weise (mindestens per Handbuch-Eintrag und per Hinweis innerhalb der UpdateInformation eines FirmwareUpdates am KSR) darüber informieren, dass im Fall, dass eine TSL-Aktualisierung innerhalb der TI fehlschlägt, automatisch versucht wird, eine TSL-Aktualisierung aus dem Internet vorzunehmen.**[<=]**

4.1.9.3.2 TUC_KON_031 „BNetzA-VL aktualisieren"

TIP1-A_6729 - TUC_KON_031 „BNetzA-VL aktualisieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_031 „BNetzA-VL aktualisieren" umsetzen.

Tabelle 260: TAB_KON_618 TUC_KON_031 „BNetzA-VL aktualisieren"

Element	Beschreibung
Name	TUC_KON_031 „BNetzA-VL aktualisieren"
Beschreibung	Dieser TUC prüft die Aktualität der BNetzA-VL. Wenn eine neuere BNetzA-VL vorliegt, wird diese heruntergeladen, geprüft und im Truststore gespeichert.
Auslöser	• Aufruf durch andere TUCs • TIP1-A_6728
Vorbedingungen	• Aktuell gültige TSL im Truststore vorhanden
Eingangsdaten	• BNetzA-VL aus manuellem Import (Optional)

	• Flag „MGM_LU_ONLINE“ für Offline-/Online-Modus • Flag „MGM_LU_SAK“ für Signaturdienst-Modus
Komponenten	Konnektor
Ausgangsdaten	• Status der Prüfung
Nachbedingungen	• Aktuelle BNetzA-VL und deren Hashwert liegen im Truststore vor.
Standardablauf	1. Der Konnektor prüft und aktualisiert ggf. die BNetzA-VL durch Aufruf von TUC_PKI_036. 2. Der Konnektor löst TUC_KON_256 {"CERT/BNETZA_VL/UPDATED"; Op; Info; „"; doLog = true; doDisp = false} aus.
Varianten/Alternativen	(→1) Wird eine zu importierende BNetzA-VL manuell übergeben (in den Eingangsdaten), so wird diese statt des Downloads verwendet und an TUC_PKI_036 {BNetzA-VL Datei} übergeben. Innerhalb der PKI TUCs findet dann kein Download der BNetzA-VL statt. (→1) Ist MGM_LU_SAK=disabled, so wird der TUC ohne Fehler beendet.
Fehlerfälle	(→1) Tritt beim manuellen Import der Datei ein Fehler auf, wird TUC_KON_256 {„CERT/BNETZA_VL/IMPORT“; Op; Error; „$Fehlerbeschreibung“; doLog = true; doDisp = false} ausgelöst. Fehlercode 4129. (→1) Tritt beim periodischen Update der BNetzA-VL beim Aufruf des TUC_PKI_036 oder eines durch ihn aufgerufenen TUCs ein Fehler auf, geht der Konnektor in den Betriebszustand EC_BNetzA_VL_Update_Not_Successful. Fehlercode 4133. In beiden Fällen wird eine vorhandene gültige BNetzA-VL weiter verwendet.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 261: TAB_KON_619 Fehlercodes TUC_KON_031 „BNetzA-VL aktualisieren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4129	Technical	Error	der manuelle Import der BNetzA-Vertrauensliste schlägt fehl
4133	Security	Error	Import der BNetzA-Vertrauensliste fehlgeschlagen

[<=]

4.1.9.3.3 TUC_KON_040 „CRL aktualisieren"

TIP1-A_4694 - TUC_KON_040 „CRL aktualisieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_040 „CRL aktualisieren" umsetzen.

Tabelle 262: TAB_KON_767 TUC_KON_040 „CRL aktualisieren"

Element	Beschreibung
Name	TUC_KON_040 „CRL aktualisieren"
Beschreibung	Dieser TUC aktualisiert die CRL
Auslöser	• Aufruf durch andere TUCs
Vorbedingungen	• Ein gültiger Vertrauensraum
Eingangsdaten	• importedCRL – *optional* (Manuell importierte CRL)
Komponenten	Konnektor
Ausgangsdaten	Keine
Nachbedingungen	• Eine aktuelle, gültige CRL liegt vor

Standardablauf	1. Der Konnektor lädt die aktuelle CRL von `CERT_CRL_DOWNLOAD_ADDRESS` herunter.
	2. Die Prüfung der CRL-Signatur mit dem CRL-Signer-Zertifikat setzt sich aus folgenden Teilschritten zusammen
	a. Prüfung auf zeitliche Gültigkeit des CRL-Signer-Zertifikats mittels TUC_PKI_002 "Gültigkeitsprüfung des Zertifikats" mit Referenzzeitpunkt = aktuelle Systemzeit
	b. Auswahl des öffentlichen Schlüssels des CRL-Signer-Zertifikats (CRL-Signer-Zertifikat im Truststore)
	c. Die Signatur und der verwendete Algorithmus werden aus der CRL ausgelesen
	d. Verifikation der Signatur und Hashwert-Vergleich (Verfahren siehe [RFC5280]).
	Falls die Prüfung ein negatives Ergebnis erbracht hat, löst der
	Konnektor das Ereignis TUC_KON_256 {
	topic = „CERT/CRL/INVALID";
	eventType = Op;
	severity = Error;
	parameters = „„;
	doLog = true;
	doDisp = false }
	aus.
	3. Nach einer erfolgreichen Prüfung speichert der Konnektor die neue CRL und löst das Ereignis TUC_KON_256{

	topic = „CERT/CRL/UPDATED“;
	eventType = Op;
	severity = Error;
	parameters = „„;
	doLog = true;
	doDisp=false}
	aus.
	4. Falls die aktuelle Systemzeit den Wert NextUpdate aus der CRL erreicht oder überschritten hat, geht der Konnektor in den Betriebszustand EC_CRL_Out_Of_Date.

Varianten/ Alternativen	(→1) Wird eine manuell importierte CRL übergeben, so wird diese verwendet.

Fehlerfälle	(→1) Tritt beim manuellen Import der Datei ein Fehler auf, wird TUC_KON_256 {
	topic = „CERT/CRL/IMPORT“;
	eventType = Op;
	severity = Error;
	parameters = „${Fehlerbeschreibung}“;
	doLog = true;
	doDisp=false}
	ausgelöst.
	(→2) Signaturprüfung der CRL fehlgeschlagen: Fehlercode 4130

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 263: TAB_KON_599 Fehlercodes TUC_KON_040 „CRL aktualisieren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4130	Security	Error	Signatur- oder Gültigkeitsprüfung der CRL fehlgeschlagen

[<=]

4.1.9.3.4 TUC_KON_033 „Zertifikatsablauf prüfen“

TIP1-A_4695 - TUC_KON_033 „Zertifikatsablauf prüfen“
Der Konnektor MUSS den technischen Use Case TUC_KON_033 „Zertifikatsablauf prüfen” umsetzen.

Tabelle 264: TAB_KON_768 TUC_KON_033 „Zertifikatsablauf prüfen“

Element	Beschreibung
Name	TUC_KON_033 „Zertifikatsablauf prüfen“

Beschreibung	Dieser TUC prüft und meldet das zeitliche Ablaufen eines X.509-Zertifikats einer Karte.
Auslöser	• Aufruf durch andere TUCs des Konnektors oder • über die Managementschnittstelle
Vorbedingungen	Keine
Eingangsdaten	• cardSession – *optional*/für eGK, HBA, SM-B, gSMC-KT • checkSMCK [Boolean] – *optional/für gSMC-K;* (Referenz auf eine/die gSMC-K, alternativ zu cardSession) • doInformClients [Boolean] (Angabe, ob ein Event an die Clients gesendet werden soll) • crypt - *optional; default = RSA* (kryptographischer Algorithmus, für welchen das Zertifikat ermittelt wird; Wertebereich: ECC, RSA)
Komponenten	Konnektor
Ausgangsdaten	• expirationDate (Ablaufdatum des untersuchten Zertifikats)

Standardablauf	1. TUC_KON_216 „LeseZertifikat“ für:
	• Bei checkSMCK das Zertifikat der gSMC-K (C.NK.VPN) gemäß TAB_KON_856
	• bei CardSession die Zertifikate der identifizierten Karte.
	i. Für die eGK: C.CH.AUT
	ii. Für den HBAx: C.HP.AUT
	iii. Für SM-B: C.HCI.AUT
	• Das konkrete Zertifikatsobjekt der Karte gemäß TAB_KON_858 wird vom Eingangsparameter `crypt` abgeleitet.
	2. Das Ablaufdatum *expirationDate* wird aus dem Feld validity ausgelesen.
	3. Falls das Zertifikat abgelaufen ist, Systemereignis absetzen:
	• gSMC-K:
	TUC_KON_256 {
	topic = „CERT/CARD/EXPIRATION”;
	eventType = Op;
	severity = Warning;
	parameters = („CARD_TYPE=gSMC-K,
	ICCSN=$ICCSN,
	Konnektor=$MGM_KONN_HOSTNAME,
	ZertName=$Name des Zertifikatsobjekts gemäß

	TAB_KON_856,
	ExpirationDate=$validity");
	doLog = true;
	doDisp = $doInformClients }
	• Sonstige Karten (mit CARD(CardSession)):
	TUC_KON_256 {
	topic = „CERT/CARD/EXPIRATION";
	eventType = Op;
	severity = Warning;
	parameters = („CARD_TYPE=$Type,
	ICCSN=$ICCSN,
	CARD_HANDLE=$CardHandle,
	CardHolderName=$CardHolderName,
	ZertName=$Name des Zertifikatsobjekts aus Schritt 1,
	ExpirationDate=$validity");
	doLog=false;
	doDisp = $doInformClients }
	4. Alternativ bei Ablauf des Zertifikats innerhalb von CERT_EXPIRATION_WARN_DAYS Systemereignis absetzen:
	• gSMC-K:
	TUC_KON_256 {
	topic = „CERT/CARD/EXPIRATION";
	eventType = Op;
	severity = Info;
	parameters = („CARD_TYPE=gSMC-K,
	ICCSN=$ICCSN,
	Konnektor=$MGM_KONN_HOSTNAME,
	ZertName=$Name des Zertifikatsobjekts gemäß TAB_KON_856,
	ExpirationDate=$validity,
	DAYS_LEFT=$validity-$Today");
	doLog = false;
	doDisp = $doInformClients}
	• Sonstige mit CARD(CardSession)):
	TUC_KON_256 {
	topic = „CERT/CARD/EXPIRATION";
	eventType = Op;
	severity = Info;
	parameters = („CARD_TYPE=$Type,
	ICCSN = $ICCSN,
	CARD_HANDLE = $CardHandle,
	CardHolderName = $CardHolderName,
	ZertName=$Name des Zertifikatsobjekts aus Schritt 1,
	ExpirationDate = $validity,
	DAYS_LEFT = $validity-$Today");
	doLog = false;
	doSisp = $doInformClients}

	5. expirationDate wird zurückgegeben.
Varianten/ Alternativen	Keine
Fehlerfälle	(→1) Zur angegebenen CardSession keine Karte gefunden: Fehlercode 4131. (→1) Für eGK, HBA, SM-B gilt: Wenn crypt=ECC und Kartengeneration<G2.1, bricht der TUC mit Warnung 4257 ab. (→1) Für gSMC-K gilt: Wenn crypt=ECC und beim Aufruf von TUC_KON_216 wird die Warnung 4256 zurückgegeben, dann wird der TUC nach Schritt 1 beendet und die Warnung 4257 an den Aufrufer zurückgegeben. (→2) Extraktion des Ablaufsdatums fehlgeschlagen: Fehlercode 4132.
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 265: TAB_KON_600 Fehlercodes TUC_KON_033 „Zertifikatsablauf prüfen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4131	Technical	Error	Zum angegebenen CardHandle keine Karte gefunden.
4132	Security	Error	Extraktion des Ablaufsdatums fehlgeschlagen
4257	Technical	Warning	ECC-Zertifikate nicht vorhanden auf Karte: <cardHandle>

[<=]

4.1.9.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.9.4.1 TUC_KON_037 „Zertifikat prüfen"

TIP1-A_4696-02 - TUC_KON_037 „Zertifikat prüfen"
Der Konnektor MUSS den technischen Use Case „Zertifikat prüfen" gemäß TUC_KON_037 „Zertifikat prüfen" umsetzen.

Tabelle 266: TAB_KON_769 TUC_KON_037 „Zertifikat prüfen"

Element	Beschreibung
Name	TUC_KON_037 „Zertifikat prüfen"
Beschreibung	Der TUC beschreibt • die Prüfung eines X.509-Zertifikats gegen den Vertrauensraum
Auslöser	• Aufruf in einem Fachmodul oder • technischen Use Case
Vorbedingungen	• aktuelle TSL-Informationen im Truststore vorhanden • für QES X.509-Prüfung: eine aktuell gültige BNetzA-VL
Eingangsdaten	• certificate (ein X.509-Zertifikat (nonQES- oder QES-X.509-Zertifikat)) • EECertificateContainedInTSL - *optional; default: false* (true: Prüfung, ob ein EE-Zertifikat in der TSL vorhanden und zeitlich gültig ist; EE-Zertifikat wird in der TSL innerhalb eines "TSPService"-Eintrags ServiceTypeIdentifier "http://uri.etsi.org/TrstSvc/Svctype/unspecified" erwartet. false: vollständige Prüfung eines X.509-Zertifikats mit TUC_PKI_018 bzw. TUC_PKI_030) • qualifiedCheck [not_required | required | if_QC_present] – (Art der Zertifikatsprüfung) • baseTime – *optional/verpflichtend, wenn ein Zeitpunkt zur Prüfung vorgegeben werden soll; default: Verwendung der Systemzeit des Konnektors* (Referenzzeitpunkt: Zeitpunkt, für den das Zertifikat geprüft werden soll) • offlineAllowNoCheck [Boolean] – *optional; default: false* (Angabe, ob es als Fehler (false) oder als Warnung (true) interpretiert werden soll, wenn eine OCSP-Prüfung nicht durchgeführt werden konnte.) • intendedKeyUsage – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist; wird bei QES nicht ausgewertet* (Vorgesehene KeyUsage) • nur für nonQES-Zertifikate:

	• policyList (Liste der zugelassenen Zertifikatstyp-OIDs gemäß [gemSpec_OID#GS-A_4445]) • intendedExtendedKeyUsage – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist; wird bei QES nicht ausgewertet* *(*Vorgesehene ExtendedKeyUsage) • gracePeriod – *optional/nur für nonQES-X.509-Zertifikat und wenn vom Standard abgewichen werden soll; wird bei QES nicht ausgewertet*; *default*: *CERT_OCSP_DEFAULT_GRACE_PERIOD_NONQES* (OCSP-GracePeriod: maximal zulässiger Zeitraum, den letzte OCSP-Antwort aus dem Cache bezüglich des Referenzzeitpunkts zurückliegen darf;) • validationMode [OCSP | CRL | NONE] – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist* (Prüfmodus: • OCSP: Es wird mittels OCSP geprüft. Dabei wird, falls die OCSP-GracePeriod noch nicht abgelaufen ist, die OCSP-Antwort aus dem Cache des Konnektors verwendet. Für QES einzig erlaubter validationMode. • CRL: Es wird gegen die aktuelle CRL auf dem Konnektor geprüft. • NONE: Keine Prüfung von Statusinformationen) • ocspResponse – *optional* (OCSPResponse des EE-Zertifikats) • getOCSPResponses [Boolean]– *optional; default: false* (true – OCSPResponse des geprüften Zertifikats soll an den Aufrufer zurückgegeben werden)
Komponenten	Konnektor
Ausgangsdaten	• Status und Liste von Warnungen/Fehlern bei der Zertifikatsprüfung • role (aus dem Zertifikate ermittelte Rolle oder Berufsgruppe; siehe „Tab_PKI_406 OID-Festlegung technische Rolle in X.509-Zertifikaten“ oder „Tab_PKI_402 OID-Festlegung Rolle im X.509-Zertifikat für Berufsgruppen“ oder Tab_PKI_403 OID-Festlegung Institutionen im X.509-Zertifikat der SMC-B [gemSpec_OID]) • qcStatement – *optional/verpflichtend, wenn certificate ein QES-X.509-Zertifikat ist*; *nicht relevant bei EECertificateContainedInTSL=true.* (QCStatements des Zertifikats) • ocspResponsesRenewed – *optional/verpflichtend, wenn Eingabeparameter getOCSPResponses = true;*

	nicht relevant bei EECertificateContainedInTSL=true. (OCSP-Response des geprüften Zertifikats)

Standardablauf	1. Falls EECertificateContainedInTSL=false: a. Wenn das X.509-Zertifikat von einem CA-Zertifikat ausgestellt wurde, das in CERT_IMPORTED_CA_LIST enthalten ist, erfolgt eine Zertifikatsprüfung analog zu den Festlegungen in TUC_PKI_018 „Zertifikatsprüfung". Dabei sind zu prüfen: - Zeitliche Gültigkeit, - Gültigkeit des EE-Zertifikats nach Kettenmodell (analog zu z. B. [gemKPT_PKI_TIP#2.4.3]) - mathematische Prüfung der Zertifikatssignatur, - die Prüfung der Zweckbindung gemäß der im Zertifikat hinterlegten keyUsage TSL-bezogene Prüfungen im TUC_PKI_018 werden in diesem Fall nicht durchgeführt. Ebenso erfolgt keine OCSP-Prüfung. b. Wenn das zum X.509-Zertifikat gehörende CA-Zertifikat nicht in CERT_IMPORTED_CA_LIST enthalten ist, werden, abhängig vom Parameter *qualifiedCheck* folgende TUCs unter Weitergabe aller Eingangsparameter sowie der Negation des Werts von MGM_LU_ONLINE als Parameter „Offline-Modus" aufgerufen: i. Für qualifiedCheck = not_required: TUC_PKI_018 „Zertifikatsprüfung in der TI" Ist der Eingangsparameter ocspResponses mit einer OCSP-Antwort gefüllt, so wird dieser übergeben. Die aktuell aus der OCSP-Abfrage resultierte OCSP-Antwort, falls vorhanden, wird an den Aufrufer weitergegeben. ii. Für qualifiedCheck = required: TUC_PKI_030 „QES-Zertifikatsprüfung" Dabei wird das Basiszertifikat übergeben. Ist Eingangsparamter ocspResponses mit einer OCSP-Response gefüllt, so wird dieser übergeben. Die aktuell aus der OCSP-Abfrage resultierende OCSP-Response, falls vorhanden, wird an den Aufrufer weitergegeben. iii. Für qualifiedCheck = if_QC_present: Ist im jeweiligen Signaturzertifikat mindestens ein QCStatement mit dem OID id-etsi-qcs-QcCompliance (0.4.0.1862.1.1) enthalten, handelt es sich um eine QES-Zertifikatsprüfung mittels TUC_PKI_030 „QES-Zertifikatsprüfung", sonst um eine nonQES-Zertifikatsprüfung mittels TUC_PKI_018 „Zertifikatsprüfung". Als Timeout wird beim Aufruf von TUC_PKI_018 der Wert von CERT_OCSP_TIMEOUT_NONQES bzw. beim Aufruf von TUC_PKI_030 der Wert von CERT_OCSP_TIMEOUT_QES übergeben (siehe auch Eingangsdaten von diesen TUCs in [gemSpec_PKI]).

	Für die QES-Zertifikatsprüfung wird das zu prüfende QES-Zertifikat an TUC_PKI_030 „QES-Zertifikatsprüfung“ übergeben. Wird im Aufruf der Eingangsparameter getOCSPResponses = false mit übergeben, wird keine OCSP-Response an den Aufrufer zurückgegeben. Als TOLERATE_OCSP_FAILURE wird beim Aufruf von TUC_PKI_018 offlineAllowNoCheck verwendet. Wenn der Eingangsparameter validationMode („Prüfmodus“) den Wert NONE hat, werden die TUC_PKI_018-Eingangsparameter • „Offline-Modus“ unabhängig von MGM_LU_ONLINE auf „ja“ gesetzt und • „Prüfmodus“ auf „OCSP“. 2. Falls EECertificateContainedInTSL=true a. Prüfe, ob das in certificate übergebene X.509-Zertifikat in der TSL innerhalb eines "TSPService"-Eintrags mit dem ServiceTypeIdentifier "http://uri.etsi.org/TrstSvc/Svctype/unspecified" aufgeführt ist. b. Prüfe zeitliche Gültigkeit von certificate zum Prüfzeitpunkt aktuelle Systemzeit durch Aufruf von TUC_PKI_002. c. Ermittle role von certificate durch Aufruf von TUC_PKI_009. 3. Der Status der Prüfung und die ermittelten Ausgangsdaten werden zurückgegeben.

Varianten/ Alternativen	
Fehlerfälle	TUC_KON_037 im kritischen Betriebszustand EC_TSL_Out_Of_Date_Beyond_Grace_Period aufgerufen: Fehlercode 4002. -> 2a) certificate ist nicht in der TSL enthalten
Nichtfunktionale Anforderungen	Der Konnektor MUSS unter Einhaltung aller anderen Anforderungen an die Zertifikatsprüfung die Anzahl der OCSP-Abfragen minimieren. Dies MUSS durch Caching (unter Berücksichtigung der Grace Period) und DARF NICHT durch Bündelung von OCSP-Anfragen geschehen.
Zugehörige Diagramme	keine

Tabelle 267: TAB_KON_601 Fehlercodes TUC_KON_037 „Zertifikat prüfen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases treten folgende Fehlercodes auf.			
4002	Security	Fatal	Der Konnektor befindet sich in einem kritischen Betriebszustand
4260	Security	Error	Zertifikat nicht vorhanden in TSL

[<=]

4.1.9.4.2 TUC_KON_042 „CV-Zertifikat prüfen"

TIP1-A_5482 - TUC_KON_042 „CV-Zertifikat prüfen"
Der Konnektor MUSS den technischen Use Case „CV-Zertifikat prüfen" gemäß TUC_KON_042 „CV-Zertifikat prüfen" umsetzen.
[<=]

Tabelle 268: TAB_KON_818 TUC_KON_042 „CV-Zertifikat prüfen"

Element	Beschreibung
Name	TUC_KON_042 „CV-Zertifikat prüfen"
Beschreibung	Die Gültigkeit eines (EndEntity-)CV-Zertifikats wird geprüft. Es werden folgende Prüfungen durchgeführt: Kryptographische Prüfung der Signaturen des End-Entity-CV-Zertifikats und des CVC-CA-Zertifikats

	• Zeitliche Gültigkeit nach dem Schalenmodell (nur CV-Zertifikate der Generation 2).
Auslöser	• Aufruf in einem Fachmodul oder • Technischen Use Case
Vorbedingungen	• keine
Eingangsdaten	• eeCertificate (zu prüfendes kartenindividuelles CV-Zertifikat) • caCertificate (das CVC-CA-Zertifikat mit dem öffentlichen Schlüssel der zugehörigen ausstellenden CA)
Komponenten	Konnektor
Ausgangsdaten	• status [Boolean] (Ergebnis der Prüfung; true: CV-Zertifikat ist gültig false: CV-Zertifikat ist ungültig)
Standardablauf	1. Abhängig von der Zertifikats-Generation wird Vorgehen A oder B gewählt. A. Prüfung von CV-Zertifikaten der Generation 1: Die CVC-Prüfung setzt sich gemäß GS-A_4668 [gemSpec_PKI#8.7] aus folgenden Schritten zusammen. i. Prüfe die Signatur des CA-Zertifikats caCertificate mit dem öffentlichen Root-Schlüssel der ausstellenden CVC-Root-CA. Der benötigte Root-Key befindet sich sich auf der gSMC-K in der Datei EF.PuK.RCA.CS.R2048. ii. Prüfe die Signatur des (EndEntity-)CV-Zertifikats eeCertificate mit dem öffentlichen Schlüssel der ausstellenden CVC-CA (aus dem CVC-CA-Zertifikat extrahiert). B. Prüfung von CV-Zertifikaten der Generation 2: Die CVC-Prüfung setzt sich gemäß GS-A_5009, … GS-A_5012 [gemSpec_PKI#8.8] aus folgenden Schritten zusammen: i. Prüfe die kryptographische Korrektheit der Signatur des CA-Zertifikats caCertificate mit dem öffentlichen Root-Schlüssel der ausstellenden CVC-Root-CA. Der benötigte Root-Key befindet sich im Truststore des Konnektors. ii. Prüfe die kryptographische Korrektheit der Signatur des (EndEntity-)CV-Zertifikats certificate mit dem öffentlichen Schlüssel der ausstellenden CVC-CA (aus dem

	CVC-CA- Zertifikat extrahiert). iii. Prüfe die zeitliche Gültigkeit des (EndEntity-)CV-Zertifikates, des CVC-CA-Zertifikates und CVC-Root-CA-Zertifikates nach dem Schalenmodell. 2. Der Status *status* der Prüfung wird zurückgegeben.
Varianten/Alternativen	(→ B.i) Mathematische Korrektheitsprüfung CV-Zertifikate mit Cross-CV-Zertifikat (vgl. Varianten/Alternativen von TUC_KON_005)
Fehlerfälle	(→ A.i) kryptographische (mathematische) Prüfung des CVC-CA-Zertifikats fehlgeschlagen, Fehlercode 4196. (→ A.ii) kryptographische (mathematische) Prüfung des (EndEntity-) CV-Zertifikats fehlgeschlagen, Fehlercode 4196. (→ B.i) das benötigte Cross-CV-Zertifikat ist nicht vorhanden, Fehlercode 4228 (→ B.i) kryptographische (mathematische) Prüfung des CVC-CA-Zertifikats fehlgeschlagen, Fehlercode 4196. (→ B.ii) kryptographische Prüfung des (EndEntity-)CV-Zertifikats fehlgeschlagen, Fehlercode 4196. (→ B.iii) zeitliche Gültigkeit eines der CV-Zertifikate ist abgelaufen, Fehlercode 4196.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 269: TAB_KON_819 Fehlercodes TUC_KON_042 „CV-Zertifikat prüfen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4196	Technical	Error	Fehler bei der CV-Zertifikatsprüfung
4228	Technical	Error	das benötigte Cross-CV-Zertifikat ist nicht vorhanden

4.1.9.4.3 TUC_KON_034 „Zertifikatsinformationen extrahieren"

TIP1-A_4697 - TUC_KON_034 „Zertifikatsinformationen extrahieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_034 „Zertifikatsinformationen extrahieren" umsetzen.

Tabelle 270: TAB_KON_770 TUC_KON_034 „Zertifikatsinformationen extrahieren"

Element	Beschreibung

Name	TUC_KON_034 „Zertifikatsinformationen extrahieren“
Beschreibung	Dieser TUC beschreibt die Extraktion der fachlich zentralen Informationen aus bestimmten Zertifikaten einer gesteckten Karte eines Mandanten.
Auslöser	• Aufruf durch ein Fachmodul oder eine Basisanwendung des Konnektors
Vorbedingungen	Keine
Eingangsdaten	• cardSession – *optional/verpflichtend für den Zugriff auf eGK, HBA, SM-B oder gSMC-KT* • checkSMCK [Boolean] – *optional/verpflichtend für gSMC-K;* (Referenz auf eine/die gSMC-K, alternativ zu cardSession) • qes [Boolean] - *optional; default: false* – (Angabe, ob die QES-Identität oder die nonQES-Identität der Karte interessiert) • crypt - *optional; default = RSA* (kryptographischer Algorithmus, für welchen das Zertifikat ermittelt wird; Wertebereich: ECC, RSA)
Komponenten	Konnektor
Ausgangsdaten	• certType [C.CH.AUT | C.HP.AUT | C.HCI.AUT | C.HP.QES] (Zertifikatstyp) • certInfo (Zertifikatsinformationen, bestehend aus SerialNumber, Issuer, Subject, Rollen, registrationNumber und ggf. id-etsi-qcs-QcCompliance, siehe Standardablauf) • qcStatments – *optional/nur wenn certType = C.HP.QES* (QCStatements)
Nachbedingungen	Keine
Standardablauf	1. Je nach Kartentyp wird aus der Karte das passende Zertifikat über TUC_KON_216 "LeseZertifikat" {selektiertes Zertifikat} ausgelesen. Das Zertifikatsobjekt (fileIdentifier und folder)/Zertifikatsbezeichnung wird für die jeweilige Karte unter Berücksichtigung des kryptographischen Verfahrens crypt gemäß TAB_KON_858 bzw. TAB_KON_856 ermittelt.

	a. Bei qes = false: i. Für die eGK: C.CH.AUT ii. Für den HBAx: C.HP.AUT iii. Für SM-B: C.HCI.AUT iv. Für gSMC-K: C.NK.VPN b. Bei qes = true: i. Für den HBAx: C.HP.QES 2. Die Zertifikatsbezeichnung aus Schritt 1 („C.XXX.YYY.ZZZZ") wird als Ausgangsdatum „certType" zurückgegeben. 3. Zusätzlich werden aus dem Zertifikat folgende Informationen extrahiert und zurückgegeben: a. X509SerialNumber b. Issuer (DistinguishedName) nach RFC 2253 c. Subject (DistinguishedName) nach RFC 2253 d. Aus der Extension Admission: i. eine Liste von Rollen durch Aufruf von TUC_PKI_009 „Rollenermittlung" ii. registrationNumber (=Telematik-ID; falls vorhanden) e. id-etsi-qcs-QcCompliance (0.4.0.1862.1.1) in QCStatements, falls vorhanden f. Restriction, falls vorhanden g. validity
Varianten/Alternativen	Keine
Fehlerfälle	(→1) Wenn im Aufrufkontext (also erreichbar durch den Mandanten) zum angegebenen CardHandle keine Karte gefunden werden kann, bricht der TUC mit Fehlercode 4146 ab. (→1b) Ist bei Angabe von QES=true auf der Karte keine QES-Identität zu finden, bricht der TUC mit Fehlercode 4147 ab. Für die Kombination QES=true mit einer eGK bricht der TUC mit Fehlercode 4148 ab (QES-Zertifikate der eGK werden noch nicht unterstützt). (→1) Für eGK, HBA, SM-B gilt: Wenn crypt=ECC und Karte vom Typ <G2.1, bricht der TUC mit Warnung 4257 ab. (→1) Für gSMC-K gilt: Wenn crypt=ECC und TUC_KON_216 Warnung 4256 liefert, bricht der TUC mit Warnung 4257 ab. (→1) Wenn aus anderen Gründen die Extraktion der Zertifikatsinformationen fehlschlägt, bricht der TUC mit Fehlercode 4148 ab.

Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 271: TAB_KON_602 Fehlercodes TUC_KON_034 „Zertifikatsinformationen extrahieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4146	Technical	Error	Kartenhandle existiert nicht
4147	Technical	Error	Zertifikat nicht vorhanden (z. B. kein QES-Zertifikat in SM-B)
4148	Technical	Error	Fehler beim Extrahieren von Zertifikatsinformationen
4257	Technical	Warning	ECC-Zertifikate nicht vorhanden auf Karte: <cardHandle>

[<=]

4.1.9.5 Operationen an der Außenschnittstelle

TIP1-A_4698-03 - Basisanwendung Zertifikatsdienst

Der Konnektor MUSS Clientsystemen eine Basisanwendung Zertifikatsdienst zur Verfügung stellen

Tabelle 272: TAB_KON_771 Basisanwendung Zertifikatsdienst

Name	`CertificateService`	
Version (KDV)	6.0.0 (WSDL-Version), 6.0.1 (XSD-Version) 6.0.1 (WSDL-Version), 6.0.2 (XSD-Version)	
Namensraum	Siehe GitHub	
Namensraum-Kürzel	`CERT für Schema und CERTW für WSDL`	
Operationen	Name	Kurzbeschreibung
	`ReadCardCertificate`	Zertifikat von einer Karte lesen
	`CheckCertificateExpiration`	Ablaufdatum von Zertifikaten erfragen

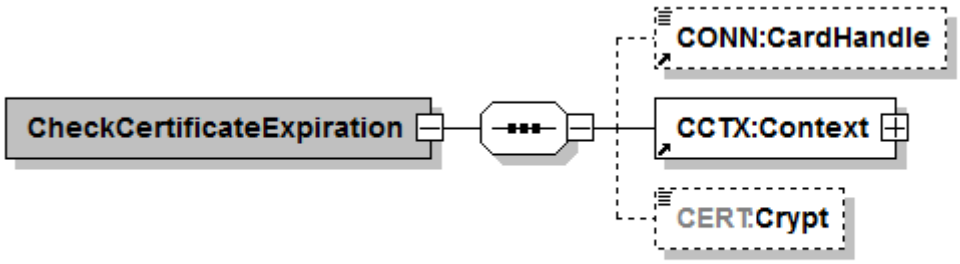
	VerifyCertificate	Prüfung des Status eines Zertifikats
WSDL	CertificateService.wsdl (WSDL-Verion 6.0.0) CertificateService_v6_0_1.wsdl	
Schema	CertificateService.xsd (XSD-Version 6.0.1) CertificateService_v6_0_2.xsd	

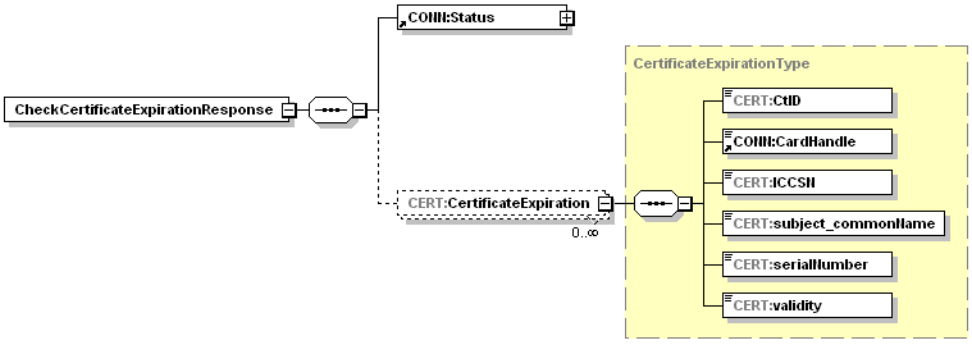
[<=]

4.1.9.5.1 CheckCertificateExpiration

TIP1-A_4699-02 - Operation CheckCertificateExpiration
Die Basisanwendung Zertifikatsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation CheckCertificateExpiration anbieten.

Tabelle 273: TAB_KON_676 Operation CheckCertificateExpiration

Name	CheckCertificateExpiration	
Beschreibung	Gibt das Datum des Ablaufs eines bestimmten Zertifikats oder gesammelt des Zertifikats der gSMC-K, der gSMC-KT's sowie aller gesteckten HBAx und SM-B des Mandanten zurück.	
Aufruf-parameter		
	Name	**Beschreibung**
	CardHandle	Optional. Identifiziert die Karte, deren Zertifikate geprüft werden sollen. Wird der Parameter nicht angegeben, so werden alle für den Konnektor erreichbaren Karten (inkl. gSMC-K und aller gSMC-KT's), die zum Mandanten passen, berücksichtigt. Die Operation CheckCertificateExpiration DARF das Lesen von Zertifikaten der eGK NICHT unterstützen.
	Context	MandantId, CsId, WorkplaceId verpflichtend; UserId optional
	Crypt	Optional; Default: RSA Gibt den kryptopraphischen Algorithmus vor, für

		den das Zertifikat ermittelt werden soll. Wertebereich: RSA, ECC • RSA: Zertifikat für RSA-2048 • ECC: Zertifikat für ECC-256
Rückgabe		
	`Status`	Enthält den Ausführungsstatus der Operation.
	`CertificateExpiration`	Eine Liste von Tupeln aus (CtID, CardHandle, ICCSN, subject.CommonName, serialNumber, validity) der Zertifikate der Karten. Für die gSMC-K soll in CertificateExpiration/CtID und CertificateExpiration/CardHandle jeweils ein Leerstring zurückgegeben werden.
Vorbe-dingungen	Keine	
Nachbe-dingungen	Keine	

Der Ablauf der Operation CheckCertificateExpiration ist in Tabelle TAB_KON_677 beschrieben:

Tabelle 274: TAB_KON_677 Ablauf CheckCertificateExpiration

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab. Wird die Operation für einen nicht unterstützten Kartentypen aufgerufen, so bricht die Operation mit Fehler 4058 ab.

2.	TUC_KON_000 „Prüfe Zugriffsberechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientSystemId = $context.clientsystemId;

		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle
		}
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	enumerateCardHandles	Wenn der Parameter CardHandle übergeben wurde, wird dieser als einziges Element in eine Liste gepackt. Wenn der Parameter CardHandle leer war, wird eine Liste der CardHandles aller für den Konnektor erreichbaren Karten (inkl. gSMC-K und gSMC-KT's), die zum Mandanten passen, erstellt.
Für jedes CardHandle der in Schritt 3 erzeugten Liste werden folgende Schritte ausgeführt, für die gSMC-Ks die Schritte 5 und 6: Falls Schritt 5 der TUC_KON_033 die Warnung 4257 zurückgibt, wird Schritt 6 nicht ausgeführt und die Schritte für das CardHandle der in Schritt 3 erzeugten Liste weiter ausgeführt. Die Warnung 4257 wird über alle CardHandle akkumuliert und <komma-separierte List von cardHandle> für den Fehlertext erzeugt.		

4.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 {
		mandatId =MandantId;
		clientSystemId = ClientSystemId;
		cardHandle = CardHandle;
		userId = UserId }

5.	TUC_KON_033 „Zertifikatsablauf prüfen“	Das Gültigkeitsdatum des Zertifikats wird geprüft mit TUC_KON_033 {
		cardSession;
		doInformClients = false;
		Crypt;
		}
		bzw. TUC_KON_033 {
		checkSMCK = true;
		doInformClients = false;
		Crypt;
		}

6.	TUC_KON_034 „Zertifikatsinformationen extrahieren”	Beim Aufruf des TUC_KON_034 ist der Parameter qes = false zu setzen. Aus den jeweiligen Rückgabewerten entsteht eine Liste aus Tupeln (CtID, CardHandle, ICCSN, subject.CommonName, serialNumber, validity). Diese wird von der Operation zurückgegeben.

Tabelle 275: TAB_KON_603 Fehlercodes „CheckCertificateExpiration"

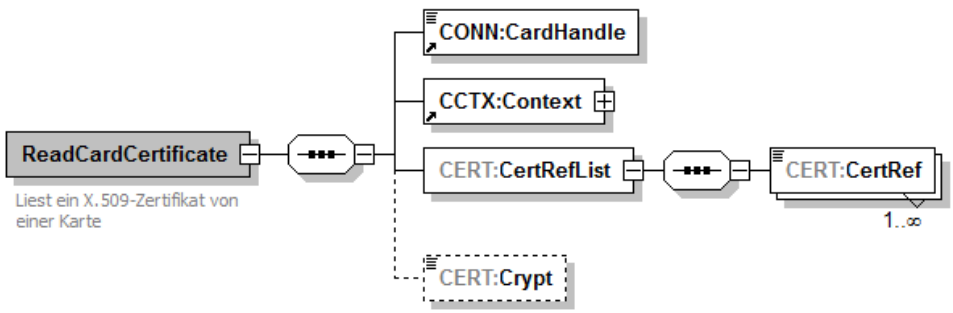
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4058	Security	Error	Aufruf nicht zulässig
4257	Technical	Warning	ECC-Zertifikate nicht vorhanden auf Karte: <komma-separierte List von cardHandle>

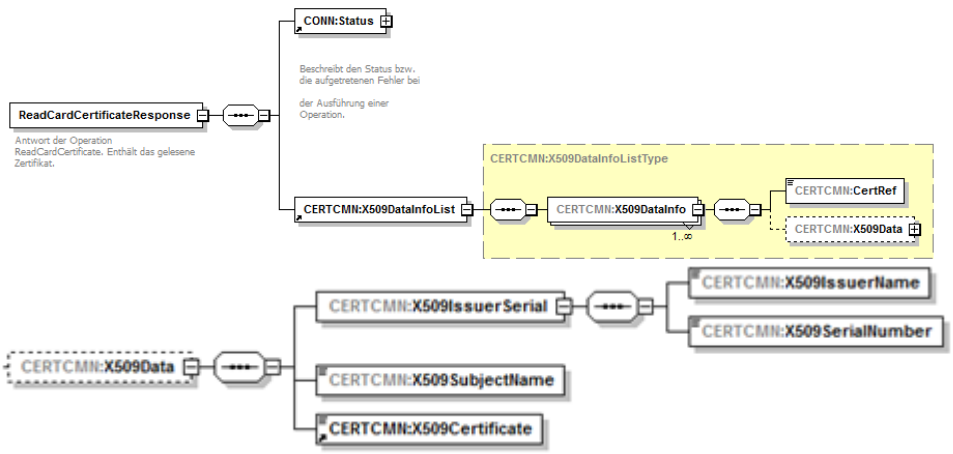
[<=]

4.1.9.5.2 ReadCardCertificate

TIP1-A_4700 - Operation ReadCardCertificate
Die Basisanwendung Zertifikatsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation ReadCardCertificate wie in Tabelle TAB_KON_678 Operation ReadCardCertificate beschrieben anbieten.

Tabelle 276: TAB_KON_678 Operation ReadCardCertificate

Name	`ReadCardCertificate`
Beschreibung	Liest X.509-Zertifikate von einer Karte.
Aufruf-parameter	

	Name	Beschreibung
	CardHandle	Gibt die Karte an, von der das Zertifikat gelesen werden soll. Es können Zertifikate von HBAx (HBA, HBA-VK), SM-B ausgelesen werden. Die Operation ReadCardCertificate DARF das Lesen von Zertifikaten der eGK NICHT unterstützen.
	Context	Aufrufkontext (Mandant)
	CertRefList	Gibt an, welche(s) Zertifikat(e) gelesen werden soll. Mögliche Werte für CertRef sind: C.AUT, C.ENC, C.SIG, C.QES
	Crypt	Optional; Default: RSA Gibt den kryptographischen Algorithmus vor, für den das Zertifikat ermittelt werden soll. Wertebereich: RSA, ECC • RSA: Zertifikat für RSA-2048 • ECC: Zertifikat für ECC-256
Rückgabe		
	Status	Enthält den Ausführungsstatus der Operation.
	CertRef	Dieses Element beinhaltet die Referenz des Zertifikats, welches bei der Anfrage übergeben

		wurde.	
	X509Data	Inhalt des über die CertRef referenzierten Zertifikats. Ist das referenzierte Zertifikat nicht vorhanden, so wird dieses Element nicht vom Konnektor gefüllt.	
		X509IssuerName	Enthält den Issuer-Name des Zertifikats. Bezüglich des Encodings sind die in [XMLDSig#4.4.4.4.1] angegebenen Regeln zu beachten (Escaping von Sonderzeichen etc.)
		X509SerialNumber	Enthält die serialNumber des Zertifikats.
		X509SubjectName	Enthält das Feld subject.CommonName. Bezüglich des Encodings sind die in [XML DSig#4.4.4.4.1] angegebenen Regeln zu beachten (Escaping von Sonderzeichen etc.)
		X509Certificate	Enthält das base64-codierte Zertifikat, dessen Binärstruktur wiederum ASN.1-codiert (gemäß [COMMON_PKI]) vorliegt.
Vorbedingungen	Keine		
Nachbedingungen	Keine		

Der Ablauf der Operation ReadCardCertificate ist in Tabelle TAB_KON_679 Ablauf ReadCardCertificate beschrieben:

Tabelle 277: TAB_KON_679 Ablauf ReadCardCertificate

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung

1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab. Wurde als Zielkarte eine eGK adressiert, wird Fehlercode 4090 zurückgeliefert.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle
		}
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 {
		mandatId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		cardHandle = $context.cardHandle;
		userId = $context.userId }

4.	getEF	Für jedes Paar von CertRef und CardHandle wird in Abhängigkeit des Parameters Crypt gemäß Tabelle TAB_KON_858 das zu lesende File (EF) bestimmt: Ist die übergebene Zertifikatsreferenz ungültig, wird Fehlercode 4149 zurückgegeben. Das Lesen von Zertifikaten der eGK ist aus Sicherheitsgründen für Clientsysteme nicht zulässig.
	TUC_KON_216 „LeseZertifikat“	Für jedes Paar von CardHandle und EF wird nun durch Aufruf von TUC_KON_216 „LeseZertifikat“ das Zertifikat ausgelesen. Falls TUC_KON_216 die Warnung 4256 zurückgibt, wird die Operation abgebrochen und Fehler 4258 zurückgegeben.
6.	Zertifikatsattribute extrahieren	Aus jedem Zertifikat werden die zu liefernden Attribute extrahiert. Die Ergebnisstruktur wird mit den erhaltenen Rückgabewerten gefüllt.

Tabelle 278: TAB_KON_604 Fehlercodes „ReadCardCertificate“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			

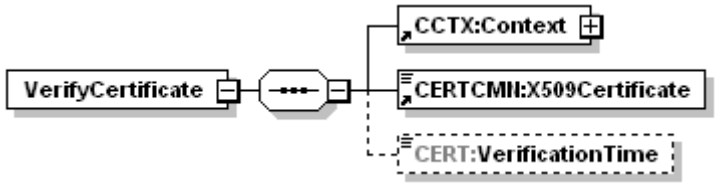
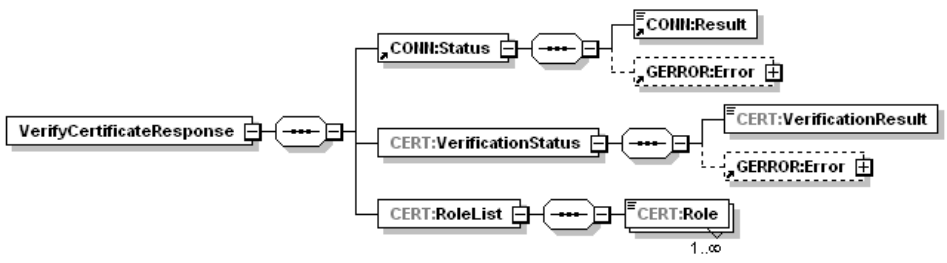
4000	Technical	Error	Syntaxfehler
4149	Technical	Error	Ungültige Zertifikatsreferenz
4090	Security	Error	Zugriff auf eGK nicht gestattet
4258	Technical	Error	ECC-Zertifikate nicht vorhanden auf Karte: <cardHandle>

[<=]

4.1.9.5.3 VerifyCertificate

TIP1-A_5449 - Operation VerifyCertificate
Die Basisanwendung Zertifikatsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation VerifyCertificate wie in Tabelle TAB_KON_795 Operation VerifyCertificate beschrieben anbieten.

Tabelle 279: TAB_KON_795 Operation VerifyCertificate

<table>
<tr><td>Name</td><td colspan="2">VerifyCertificate</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Prüft den Status eines Zertifikats.</td></tr>
<tr><td>Aufruf-parameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>CCTX:Context</td><td>Aufrufkontext (Mandant)</td></tr>
<tr><td></td><td>CERTCMN:
X509Certificate</td><td>Zu prüfendes Zertifikat (base64 kodiert), wie in Response zur Operation ReadCardCertificate enthalten.</td></tr>
<tr><td></td><td>CERT:
VerificationTime</td><td>Der für die Prüfung zu verwendende Referenzzeitpunkt. Falls der Parameter nicht angegeben ist, wird als Referenzzeitpunkt die Systemzeit verwendet.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>CONN:Status</td><td>Enthält den Ausführungsstatus der Operation.</td></tr>
</table>

	CERT: Verification Status	Enthält eines der drei möglichen Prüfungsergebnisse in CERT:VerificationResult • VALID • INCONCLUSIVE • INVALID sowie weiter Details zu den Zuständen „INCONCLUSIVE" und „INVALID" in GERROR:Error.
	CERT:RoleList	OIDs der im Zertifikat gespeicherten Rollen.
Vorbe-dingungen	Keine	
Nachbe-dingungen	Keine	

Der Ablauf der Operation VerifyCertificate ist in Tabelle TAB_KON_797 Ablauf VerifyCertificate beschrieben:

Tabelle 280: TAB_KON_797 Ablauf VerifyCertificate

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_037 „Zertifikat prüfen"	Die Zertifikatsprüfung erfolgt durch Aufruf von TUC_KON_037. Als Parameter des TUC-Aufrufs gilt für Zertifikate aus CERT_IMPORTED_CA_LIST: {
		certificate = CERTCMN:X509Certificate
		qualifiedCheck = not_required;
		baseTime = CERT:VerificationTime;
		offlineAllowNoCheck = true;
		policyList= keine Einschränkung;
		intendedKeyUsage=empty;
		intendedExtendedKeyUsage=empty;
		gracePeriod = empty;
		validationMode = NONE;
		ocspResponses (OCSP-Response/Liste von OCSP-Responses = empty }
		für alle anderen Zertifikate gilt: {
		certifiacate = CERTCMN:X509Certificate
		qualifiedCheck =if_QC_present;
		baseTime = CERT:VerificationTime;
		offlineAllowNoCheck = true;
		policyList = alle zugelassenen Zertifikatstyp-OIDs;
		intendedKeyUsage = empty;

		intendedExtendedKeyUsage = empty;
		gracePeriod = empty;
		validationMode = OCSP;
		ocspResponses = empty}.

3.		Wenn der Prüfprozess fehlerhaft war und nicht zu einem Ergebnis im Sinne eines VerificationResult führt, wird eine FaultMessage erzeugt. War der Prüfprozess erfolgreich, wird eine VerifyCertificateResponse mit CONN:Status/CONN:Result=OK, dem VerificationStatus (als Ergebnis der Zertifikatsprüfung) und den ermittelten Rollen-OIDs erzeugt. Ein Prüfergebnis „INCONCLUSIVE" bzw. „INVALID" wird in CERT:VerificationStatus/GERROR:Error mit den zugehörigen Fehlermeldungen detailliert (in diesem Fall kann CONN:Status/CONN:Result=OK oder CONN:Status/CONN:Result=Warning gesetzt sein).

Tabelle 281: TAB_KON_800 Fehlercodes „VerifyCertificate"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

[<=]

4.1.9.6 Betriebsaspekte

4.1.9.6.1 TUC_KON_035 „Zertifikatsdienst initialisieren"

TIP1-A_4701 - TUC_KON_035 „Zertifikatsdienst initialisieren"
In der Bootup-Phase MUSS der Konnektor den Zertifikatsdienst durch Aufruf des TUC_KON_035 „Zertifikatsdienst initialisieren" initialisieren.

Tabelle 282: TAB_KON_772 TUC_KON_035 „Zertifikatsdienst initialisieren"

Element	Beschreibung
Name	TUC_KON_035 „Zertifikatsdienst initialisieren"
Beschreibung	Der TUC beschreibt den gesamten Ablauf der Initialisierung des TrustStore im Rahmen der betrieblichen Prozesse: Prüfung der Aktualität, Integrität und Authentizität der Einträge im TrustStore.
Auslöser	• Bootup des Konnektors
Vorbedingungen	keine
Eingangsdaten	keine
Komponenten	Konnektor

Ausgangsdaten	• Status der Initialisierung des TrustStore
Nachbedingungen	Keine
Standardablauf	Für den übergebenen Status der Initialisierung des TrustStore werden folgende Schritte durchgeführt: 1. Durch eine DNS-Anfrage an den DNS-Forwarder zur Auflösung der SRV-RR mit dem Bezeichner "_ocsp._tcp.<DOMAIN_SRVZONE_TI>„ erhält der Konnektor Adressen des http-Forwarders des VPN-Zugangsdienststandortes. 2. Falls in den letzten 24 Stunden keine Aktualisierung der TSL und CRL im Truststore stattgefunden hat, aktualisiert der Konnektor die TSL durch den Aufruf von TUC_KON_032 „TSL aktualisieren“ und die CRL durch den Aufruf von TUC_KON_040 „CRL aktualisieren“. 3. Falls im Zeitraum von CERT_BNETZA_VL_UPDATE_INTERVAL keine Aktualisierung der BNetzA VL stattgefunden hat, aktualisert der Konnektor die BNetzA VL durch den Aufruf von TUC_KON_031 „BNetzA-VL aktualisieren“. 4. Der Konnektor prüft die Gültigkeitsdauer der Zertifikate aller gesteckten Karten (inkl. gSMC-K) mittels Aufruf von: für gSMC-K TUC_KON_033{checkSMCK; doInformClients=Ja; crypt = ECC} TUC_KON_033{checkSMCK; doInformClients=Ja; crypt = RSA} für jede gesteckte G2.0 Karte TUC_KON_033{cardSession; doInformClients=Ja; crypt = RSA} für jede gesteckte ab G2.1 Karte TUC_KON_033{cardSession; doInformClients=Ja; crypt = ECC} TUC_KON_033{cardSession; doInformClients=Ja; crypt = RSA} 5. Der Konnektor liest von der gSMC-K den öffentlichen Schlüssel des CVC-Root-Zertifikats und speichert diesen im TrustStore [gemSpec_gSMC-K_ObjSys#5.3.10].
Varianten/ Alternativen	Keine
Fehlerfälle	Keine
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 283: TAB_KON_605 Fehlercodes TUC_KON_035 „Zertifikatsdienst initialisieren“

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen technischen Use Cases können keine weiteren Fehlercodes auftreten.

[<=]

TIP1-A_4702-03 - Konfigurierbarkeit des Zertifikatsdienstes
Der Administrator MUSS die in TAB_KON_606 aufgelisteten Parameter über die Managementschnittstelle konfigurieren und die in TAB_KON_733 aufgelisteten Parameter ausschließlich einsehen können.

Tabelle 284: TAB_KON_606 Konfiguration des Zertifikatsdienstes

ReferenzID	Belegung	Bedeutung
CERT_TSL_DEFAULT_GRACE_PERIOD_DAYS	X Tage	Default Grace Period TSL in Tagen Gibt an, wie viele Tage der Konnektor mit einer zeitlich abgelaufenen TSL weiter betrieben werden kann. Der Wert MUSS zwischen 1 und 30 Tagen liegen. Default-Wert = 30 Tage *Hinweis: Vor dem zeitlichen Ablauf einer TSL wird mit ausreichendem Vorlauf eine neue TSL verteilt. Sollte die TSL dennoch ablaufen und der Konfigurationswert überschritten werden, kann eine neue TSL immer noch lokal geladen werden (TIP1-A_4705 „TSL manuell importieren").*
CERT_OCSP_DEFAULT_GRACE_PERIOD_NONQES	X Minuten	Default Grace Period OCSP für nonQES in Minuten. Der Wert MUSS zwischen 0 und 20 Minuten liegen. Default-Wert = 10 Minuten
CERT_OCSP_TIMEOUT_NONQES	X Sekunden	Timeout für OCSP-Abfragen bei der Prüfung von nonQES-Zertifikaten. Der Wert MUSS zwischen 1 und 120 Sekunden liegen. Default-Wert = 10 Sekunden
CERT_OCSP_TIMEOUT_QES	X Sekunden	Timeout für OCSP-Abfragen bei der Prüfung von QES-Zertifikaten. Der Wert muss zwischen 1 und

		120 Sekunden liegen. Default-Wert = 10 Sekunden
CERT_EXPIRATION_ WARN_DAYS	X Tag(e)	Warnung X Tage vor Ablauf von Zertifikaten im Managementinterface und per Ereignis. Der Wert muss zwischen 0 und 180 Tagen (0=keine Warnung) liegen. Default-Wert = 90 Tage
CERT_EXPIRATION_ CARD_CHECK_DAYS	X Tag(e)	Alle X Tage wird der Ablauf aller gesteckten Karten überprüft. Der Wert muss zwischen 0 und 365 liegen (0=kein Check). Default-Wert = 1 Tag
CERT_IMPORTED_ CA_LIST	Liste von manuell importierten CA-Zertifikaten	Der Administrator MUSS CA-Zertifikate importieren, anzeigen und löschen können. Der Konnektor DARF CA-Zertifikate zur Ableitung von QES-Zertifikaten NICHT importieren. Default-Wert = leere Liste
CERT_BNETZA_VL_ UPDATE_INTERVAL	X Stunden	Intervall, in dem die BNetzA VL auf Aktualität geprüft werden muss. Der Wert MUSS zwischen 1 Stunde und 168 Stunden (7 Tage) liegen. Default-Wert = 24 Stunden
CERT_TSL_DOWNLOAD_ADDRESS_INTERNET_BU	1 URI	Konfigurierbare Backup Adresse der TSL im Internet
CERT_ECC_RSA_TSL_SIGNER_CA_CERTIFICATE_INTERNET_BU	1 URI	Konfigurierbare Backup Adresse der TSL-Signer-CA Zertifikate im Internet (gemäß gemSpec_TSL#A_17680-02 und gemSpec_PKI#(5.15.3 X.509 Zertifikatsprofil der TSL-Signer-CA)
CERT_ECC_RSA_TSL_SIGNER_CA_CROSS_CERTIFICATE_INTERNET_BU	1 URI	Konfigurierbare Backup Adresse der TSL-Signer-CA-Cross Zertifikate im Internet (gemäß gemSpec_TSL#A_17680-02

		und gemSpec_PKI#(5.15.3 X.509 Zertifikatsprofil der TSL-Signer-CA)
CERT_TSL_IP_ADDRESS_INTERNET_BU	1 URI	Konfigurierbare Backup Adresse der TSL im Internet (enthält IP-Adresse des Hosts statt FQDN). Wird verwendet, falls Auflösen der FQDN mittels DNS bei CERT_TSL_DOWNLOAD_ADDRESS_INTERNET_BU fehlschlägt.

Tabelle 285: TAB_KON_733 Einsehbare Konfigurationsparameter des Zertifikatsdienstes

ReferenzID	Belegung	Bedeutung
CERT_CRL_DOWNLOAD_ADDRESS	2 URIs	Download-Adressen für die CRL
CERT_OCSP_FORWARDER_ADDRESS	2 FQDNs	Adressen der OCSP-Forwarder (HTTPS-Proxy) beim Zugangsdienstprovider Der Administrator muss in geeigneter Weise einen Test auslösen können, ob einer der Server per ICMP-Echo (ping) erreichbar ist und ob ein (beliebiger) OCSP-Request zu einer erhaltenen OCSP-Antwort führt.
CERT_OCSP_FORWARDER_PORT	TCP-Port	TCP-Port des OCSP-Forwarders (HTTPS-Proxy) beim Zugangsdienstprovider
CERT_TSL_DOWNLOAD_ADDRESS_INTERNET	1 URI	Adresse der TSL im Internet gemäß gemSpec_TSL
CERT_ECC_RSA_TSL_SIGNER_CA_CERTIFICATE_INTERNET	URIs	Adresse der TSL-Signer-CA Zertifikate im Internet (gemäß gemSpec_TSL#A_17680-02 und gemSpec_PKI#(5.15.3 X.509 Zertifikatsprofil der TSL-Signer-CA)
CERT_ECC_RSA_TSL_SIGNER_CA_CROSS_CERTIFICATE_INTERNET	URIs	Adresse der TSL-Signer-CA-Cross Zertifikate im Internet (gemäß gemSpec_TSL#A_17680-02 und

		gemSpec_PKI#(5.15.3 X.509 Zertifikatsprofil der TSL-Signer-CA)
CERT_TSL_IP_ADDRESS_INTERNET	1 URI	Adresse der TSL im Internet gemäß gemSpec_TSL (enthält IP-Adresse des Hosts statt FQDN). Wird verwendet, falls Auflösen der FQDN mittels DNS bei CERT_TSL_DOWNLOAD_ADDRESS_INTERNET fehlschlägt.

[<=]

TIP1-A_4703-01 - Vertrauensraumstatus anzeigen
Das Managementinterface des Konnektors MUSS einem authentisierten Administrator die Anzeige des Status des Vertrauensraums in Form folgender Daten anbieten: Sequenznummer der aktuellen TSL, StatusStartingTime (des TSPService (TSL-Signer-CA-Dienst) zum aktuell gültigen, aktiven TI-Vertrauensanker), NextUpdate, Gültigkeit der TSL, Typ der TSL (RSA oder ECC-RSA)) sowie optional für den Administrator einsehbar der Fingerprint des TSL-Signer-Zertifikats.**[<=]**

Der Typ der TSL liefert dem Administrator die Information, ob es sich um eine TSL handelt, die den TI-Vertrauensraum ausschließlich für Zertifikate mit kryptographischen Verfahren nach RSA-2048 (TSL(RSA)) oder für Zertifikate mit kryptographischen Verfahren nach RSA-2048 und ECC-256 (TSL(ECC-RSA)) bereitstellt. Die Information kann aus der Signatur der TSL ermittelt werden.

TIP1-A_6733 - Aktive BNetzA-VL anzeigen
Das Managementinterface des Konnektors MUSS einem authentisierten Administrator die Anzeige des Status der BNetzA-VL in Form folgender Daten anbieten: Sequenznummer, NextUpdate, Gültigkeitsstatus und Zeitpunkt der letzten Prüfung der Aktualität durch TUC_KON_031.
[<=]

TIP1-A_4704 - Zertifikatsablauf anzeigen
Der Administrator MUSS einen Prüflauf auf den innerhalb von CERT_EXPIRATION_WARN_DAYS Tagen bevorstehenden Ablauf von Zertifikaten aller für den Konnektor erreichbaren Karten (inkl. gSMC-K) an zentraler Stelle in der Managementschnittstelle auslösen können und das Ergebnis angezeigt bekommen.
Der Konnektor MUSS die Gültigkeitsdauer der Zertifikate aller gesteckten Karten (inkl. gSMC-K) prüfen mittels Aufruf von:
für gSMC-K
TUC_KON_033{checkSMCK; doInformClients=Nein; crypt = ECC}
TUC_KON_033{checkSMCK; doInformClients=Nein; crypt = RSA}
für jede gesteckte G2.0 Karte außer gSMC-K
TUC_KON_033{cardSession; doInformClients=Nein; crypt = RSA}
für jede gesteckte ab G2.1 Karte außer gSMC-K

TUC_KON_033{cardSession; doInformClients=Nein; crypt = ECC}
TUC_KON_033{cardSession; doInformClients=Nein; crypt = RSA}**[**<=**]**

A_18931 - Anzeige Personalisierungs-Status gSMC-K-X.509-Zertifikate
Der Konnektor MUSS dem Administrator die X.509-Zertifikate der verbauten gSMC-Ks gemäß TIP1-A_4506 anzeigen. Aus der Anzeige MUSS der Personalisierungs-Status der X.509-Zertifikate ersichtlich sein (dual RSA- und ECC-personalisiert oder nur RSA-personalisiert).
[<=**]**

TIP1-A_4705 - TSL manuell importieren
Der Konnektor MUSS es dem Administrator ermöglichen, eine TSL manuell von lokaler Datenquelle einzuspielen. Dabei MUSS der Konnektor TUC_KON_032{TSL-Datei} mit der manuell importierten TSL aufrufen.
Der Konnektor MUSS den manuellen Import einer TSL auch ermöglichen, wenn er sich im kritischen Betriebszustand EC_TSL_Out_Of_Date_Beyond_Grace_Period befindet.
Der Konnektor MUSS den manuellen Import einer zeitlich abgelaufenen TSL zulassen. **[**<=**]**

Auch im Fall des automatischen Imports der TSL muss dies im kritischen Betriebszustand `EC_TSL_Out_Of_Date_Beyond_Grace_Period` unterstützt werden.

A_20536 - TSL im kritischen Betriebszustand
Der Konnektor MUSS den automatischen Import einer TSL auch ermöglichen, wenn er sich im kritischen Betriebszustand `EC_TSL_Out_Of_Date_Beyond_Grace_Period` befindet. **[**<=**]**

A_20748 - Automatischer TSL Download im kritischen Zustand
Falls der Konnektor im kritischen Betriebszustand EC_TSL_Out_Of_Date_Beyond_Grace_Period ist, und falls *onlineMode* = ENABLED ist, MUSS der Konnektor periodisch und in der BootUp Phase die TSL aktualisieren.**[**<=**]**

TIP1-A_6728 - BNetzA-VL manuell importieren
Der Konnektor MUSS es dem Administrator ermöglichen, die BNetzA-VL manuell von lokaler Datenquelle einzuspielen. Dabei MUSS der Konnektor TUC_KON_031{BNetzA-VL-Datei} mit der manuell importierten BNetzA-VL-Datei aufrufen.
[<=**]**

TIP1-A_4706 - CRL manuell importieren
Der Konnektor SOLL es dem Administrator ermöglichen, eine CRL manuell von einer lokalen Datenquelle einzuspielen. In dem Fall MUSS der Konnektor TUC_KON_040{CRL-Datei} mit der manuell importierten CRL aufrufen.**[**<=**]**

Für die ECC-Migration ist es notwendig den ECC-RSA-Vertrauensraum zu etablieren. Dies erfolgt durch das Einspielen eines TSL-Signer-CA Cross-Zertifikats und das neue TSL-Signer-CA-Zertifikat, wodurch der ECC-Vertrauensanker im Konnektor im sicheren Datenspeicher gespeichert wird. Die Prüfung des Cross-Zertifikats erfolgt durch A_17821 - Wechsel des Vertrauensraumes mittels Cross-Zertifikaten (ECC-Migration). Danach kann die TSL(ECC-RSA) importiert werden. Das Ergebnis ist ein etablierter TI-Vertrauensraum für ECC und RSA.
Konnektoren müssen den ECC-Vertrauensraum automatisiert und im Rahmen des Upgrades auf PTV4 etablieren. Manuelle Schritte durch den Administrator sind für den Regelfall zu vermeiden und sollten nur im Fehlerfall nötig werden. Als Fallback-Lösung muss das manuelle Verfahren dennoch unterstützt werden.

A_20469-01 - Automatisierte Etablierung des ECC-RSA-Vertrauensraums (ECC-Migration)
In der BootUp-Phase MUSS ein Konnektor, der den RSA-Vertrauensraum (RSA) verwendet, überprüfen, ob die TSL(ECC-RSA) und die entsprechenden TSL-Signer-CA Cross-Zertifikate sowie TSL-Signer-CA-Zertifikate verfügbar sind und MUSS sie im positiven Fall automatisiert herunterladen, nach erfolgreicher Prüfung verwenden und dadurch den ECC-Vertrauensraum (ECC-RSA) etablieren.
Der Konnektor MUSS hierzu die Downloadpunkte, die mit A_17680-02 in [gemSpec_TSL#6.3.1.2] definiert sind, verwenden. Dabei MUSS der Konnektor zunächst die Downloadpunkte innerhalb der TI verwenden. Wenn der Download aus der TI fehlschlägt, MUSS der Konnektor einen der definierten Downloadpunkte im Internet verwenden.
Falls beim Wechsel auf den ECC-RSA Vertrauensraum ein Fehler auftritt, MUSS der Konnektor weiterhin den RSA-Vertrauensraum (RSA) verwenden.

[<=**]**

A_20508-01 - Protokollierung der Etablierung des ECC-RSA-Vertrauensraums (ECC-Migration)
Der Konnektor MUSS alle Schritte, die er zur Etablierung des ECC-RSA-Vertrauensraums durchläuft, im Systemprotokoll des Konnektors mit dem Log-Level "Warning" protokollieren.**[**<=**]**

A_17345 - TSL-Signer-CA Cross-Zertifikat manuell importieren (ECC-Migration)
Der Konnektor MUSS es dem Administrator ermöglichen, ein TSL-Signer-CA Cross-Zertifikat und das TSL-Signer-CA-Zertifikat für den neuen TI-Vertrauensanker manuell von lokaler Datenquelle einzuspielen. **[**<=**]**

A_17837-01 - Nutzung von Cross-Zertifikaten für Vertrauensraum-Wechsel nach ECC-RSA (ECC-Migration)
Um auf Basis des bereits etablierten Vertrauensankers (RSA) in den Vertrauensraum (ECC-RSA) zu wechseln MUSS der Konnektor bei der Initialisierung des neuen Vertrauensankers (ECC-RSA) Cross-Zertifikate verwenden. Das Ergebnis ist ein neuer etablierter TI-Vertrauensanker (ECC-RSA).**[**<=**]**

A_17548-01 - TSL-Signer-CA Zertifikat sicher speichern (ECC-Migration)
Der Konnektor MUSS den neuen TI-Vertrauensanker im sicheren Datenspeicher speichern.**[**<=**]**

A_17549-01 - TSL-Signer-CA Cross-Zertifikat im kritischen Betriebszustand (ECC-Migration)
Der Konnektor MUSS den Import des TSL-Signer-CA Cross-Zertifikats auch ermöglichen, wenn er sich im kritischen Betriebszustand EC_TSL_Out_Of_Date_Beyond_Grace_Period befindet. **[**<=**]**

A_17550-01 - TSL-Signer-CA Cross-Zertifikat importieren - Fehler (ECC-Migration)
Falls der Import des TSL-Signer-CA Cross-Zertifikats nicht erfolgreich durchgeführt werden konnte, MUSS der Konnektor den Vorgang abbrechen und einen Fehler gemäß TAB_KON_857 dem Administrator zur Anzeige bringen und protokollieren.

Tabelle 286: TAB_KON_857 - Fehlercodes beim Import des Cross-Zertifikats für TI-Vertrauensanker ECC

Fehlercode	ErrorType	Severity	Fehlertext

4255	Security	Error	Fehler beim Import des TSL-Signer-CA Cross-Zertifikats

[<=]

TIP1-A_5700 - Ereignisbasiert http-Forwarder Adressen ermitteln
Beim Auftreten des Events NETWORK/VPN_TI/UP MUSS der Konnektor über DNS die Adressen des http-Forwarders des VPN-Zugangsdienststandortes ermitteln (SRV-RR mit Bezeichner "_ocsp._tcp.<DOMAIN_SRVZONE_TI>).
[<=]

A_21158 - Mindestanzahl von CV-Crosszertifikaten
Der Konnektor MUSS mindestens 20 CV-Crosszertifikate verwalten und verarbeiten können.**[<=]**

4.1.10 Protokollierungsdienst

Der Protokollierungsdienst protokolliert system- und sicherheitsrelevante Ereignisse, sowie Ereignisse im Kontext der Performancemessung (siehe [gemSpec_Perf#4.1.2), innerhalb des Konnektors. Auch Ereignisse von Fachmodulen können protokolliert werden. Im Sicherheitsprotokoll werden alle Ereignisse eingetragen, die Auswirkungen auf Sicherheitsmerkmale des Konnektors haben können (Änderungen an der Firewall-Konfiguration, Authentisierungsfehler etc.). Ereignisse im Kontext der Performancemessung innerhalb des Konnektors werden in das Konnektor-Performanceprotokoll geschrieben. Ereignisse im Kontext der Performancemessung von Fachmodulen werden in das Fachmodul-Performanceprotokoll geschrieben. Alle anderen Ereignisse werden in das Systemprotokoll oder die Fachmodulprotokolle geschrieben (grundsätzlich trifft die Entscheidung über den zu verwendenden Protokollspeicher der Aufrufer des Protokolldienstes).

Die Protokolle werden persistiert.

Hinweis:
Ereignisse im Protokollierungsdienst adressieren nicht nur zu protokollierende Events im Sinne des Systeminformationsdienstes sondern alles, was zu einem Protokolleintrag führen soll (z.B. Fehler, Informationen zu Ablauf, Debug, Performance).

Innerhalb des Protokollierungsdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „LOG“
- Konfigurationsparameter: „LOG_“

4.1.10.1 Funktionsmerkmalweite Aspekte

TIP1-A_4708 - Protokollierungsfunktion
Der Konnektor MUSS einen Protokollierungsdienst anbieten. Dabei MUSS der Konnektor zwischen System- und Sicherheitsprotokoll, sowie Fachmodulprotokollen unterscheiden. Je Fachmodul MUSS ein getrenntes Protokoll vorhanden sein.
Die Protokolleinträge MÜSSEN durch den Konnektor lokal persistiert werden.
[<=]

TIP1-A_5654 - Sicherheits-Protokollierung
Der Konnektor MUSS herstellerspezifische Fehler, die Auswirkungen auf Sicherheitsmerkmale des Konnektors haben, in das Sicherheitsprotokoll schreiben.
[<=]

TIP1-A_4709 - Integrität des Sicherheitsprotokolls
Der Konnektor MUSS sicherstellen, dass Einträge in das Sicherheitsprotokoll nicht von außen und nicht durch den Administrator verändert und gelöscht werden können.
[<=]

TIP1-A_4710 - Protokollierung personenbezogener und medizinischer Daten
Der Konnektor DARF medizinische Daten NICHT in die Protokolldateien schreiben.
Personenbezogene Daten DÜRFEN NICHT in Protokolleinträgen gespeichert werden.
KVNR, ICCSN und CardHolderName MÜSSEN als personenbezogene Daten behandelt werden.
Die ICCSN DARF Im Fehlerfall durch Fachmodule in Protokolleinträgen gespeichert werden. Die ICCSN DARF NICHT im Sicherheitsprotokoll gespeichert werden.
[<=]

TIP1-A_6479 - Keine Protokollierung vertraulicher Daten
Der Konnektor DARF vertrauliche Daten NICHT in die Protokolldateien schreiben.
[<=]

TIP1-A_4711 - Kapazität der Protokolldateien
Der Konnektor MUSS über eine Speichergröße für Protokolldateien verfügen, so dass Einträge (protokollierte Ereignisse ab der Schwere „Warning“) über einen Zeitraum von bis zu einem Jahr darin vorgehalten werden können.
[<=]

Da sich die Menge an Einträgen nach der Größe der Einsatzumgebung richtet, ist die Speichergröße nach den in [gemSpec_Perf#3.1.1] beschriebenen Einsatzumgebungen (LE-Ux, x=1,2,3,4) ausreichend zu wählen.

TIP1-A_4712 - Protokollierung erfolgreicher Kryptooperationen
Wenn `LOG_SUCCESSFUL_CRYPTO_OPS` = Enabled MUSS der Konnektor die folgenden erfolgreich durchlaufenen Außenoperationen protokollieren:
- SignDocument,
- VerifyDocument,
- ExternalAuthenticate,
- EncryptDocument,
- DecryptDocument.

Dazu MUSS er

```
TUC_KON_256 {
    topic = „LOG/CRYPTO_OP“;
    eventType = Sec;
    severity = Info;
    parameters = („Operation=$Operationsname,
                  <für alle betroffenen Schlüssel:>
                      Karte=$ICCSN,
                      Keyref=<Referenz auf den Schlüssel>,
                      CARD_HANDLE=$CardHandle,
                      CardHolderName=$CardHolderName“);
                      doDisp = false}
```

aufrufen.

[<=]

TIP1-A_4713 - Herstellerspezifische Systemprotokollierung
Wenn `LOG_LEVEL_SYSLOG` = Info MUSS der Konnektor herstellerspezifische Informationen über den laufenden Betrieb in das Systemprotokoll eintragen, um im Bedarfsfall das

Verhalten des Konnektors analysieren zu können (Unterstützung der Fehlersuche etc.). Die Häufigkeit und der Inhalt der protokollierten Informationen sind herstellerspezifisch.
[<=]

TIP1-A_4714 - Art der Protokollierung
Der Konnektor MUSS Protokolleinträge so anlegen, dass eine Analyse der Einträge unterstützt wird:

- Die Protokolleinträge MÜSSEN eine patternbasierte Filterung unterstützen. Protokollwert/-texte sowie Attribute MÜSSEN in ihren Namensstrukturen hierauf abgestimmt sein.
- „;" MUSS als Trennzeichen zwischen Key/Value-Paaren verwendet werden.
- „=" MUSS als Trennzeichen zwischen Key und Value in einem Key/Value-Paar verwendet werden.
- Es MUSS durchgängig dasselbe Zeitstempelformat verwendet werden, entweder „timestamp=%d{dd.MM.yyyy HH:mm:ss.SSS}" (Beispiel „30.08.2017 13:44:12.436") und als Wert die gesetzliche Zeit (§4 EinhZeitG)
  oder
  „timestamp=%d{dd.MM.yyyy HH:mm:ss.SSSZ}", wobei „Z" die Zeitzonenangabe nach RFC 822 mit ("+" / "-") 4DIGIT bezeichnet (Beispiel „30.08.2017 13:44:12.436+0200").

[<=]

4.1.10.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.1.10.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.1.10.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.10.4.1 TUC_KON_271 „Schreibe Protokolleintrag"

TIP1-A_4715 - TUC_KON_271 „Schreibe Protokolleintrag"
Der Konnektor MUSS den technischen Use Case TUC_KON_271 „Schreibe Protokolleintrag" umsetzen.

Tabelle 287: TAB_KON_607 – TUC_KON_271 „Schreibe Protokolleintrag"

Element	Beschreibung
Name	TUC_KON_271 „Schreibe Protokolleintrag"
Beschreibung	Dieser TUC schreibt einen Eintrag in ein Protokoll.
Auslöser	Aufruf durch Basisdienst, Fachmodul oder TUC_KON_256 „Systemereignis absetzen"

Vorbedingungen	Im Fall eines zu protokollierenden Ereignisses des Systeminformationsdienstes wird • eventType = "Op" gesetzt, wenn Event.Type gleich "Operation", "Infrastructure ", "Business " oder "Other" bzw. • eventType = "Sec", wenn Event.Type gleich "Security". Die Schwere entspricht der Event.Severity gemäß Schema EventService.xsd. Im Fall eines zu protokollierenden Fehlers wird • eventType = "Op" gesetzt, wenn ErrorType gleich "Technical", "Business", "Infrastructure" oder "Other" bzw. eventType = "Sec", wenn ErrorType gleich "Security". Die Schwere entspricht der Severity des Fehlers.
Eingangs anforderung	Keine
Eingangsdaten	• Zu protokollierendes Ereignis • fmName – *optional/verpflichtend für Aufruf durch Fachmodule; default = „“* (Name des aufrufenden Fachmoduls; das Ereignis wird in das entsprechende Konnektor-Protokoll geschrieben) • eventType [EventType] definiert den Protokolltyp, in welchen das Ereignis geschrieben wird; Sec = Security: Ereignis wird in das Securityprotokoll geschrieben Op = Operation: Wenn fmName =„“ wird das Ereignis in das Systemprotokoll geschrieben. Wenn fmName gesetzt ist, wird das Ereignis in das durch fmName definierte Fachmodul-Protokoll geschrieben. Perf = Performance: Wenn fmName =„“ wird das Ereignis in das Konnektor-Performanceprotokoll geschrieben. Wenn fmName gesetzt ist, wird das Ereignis in das durch fmName definierte Fachmodul-Performance protokoll geschrieben. • severity { [EventSeverity] , Debug} (Schwere mit: Debug = Debug Information, Info = Information, Warn = Warning, Err = Error, Fatal • parameters beinhaltet die Daten des Ereignisses, die im Protokolleintrag geschrieben werden
Komponenten	Konnektor

Ausgangsdaten	Keine
Nachbedingungen	
Standardablauf	1. Wenn eventType = Sec, so wird das Ereignis in das Sicherheitsprotokoll geschrieben. Falls fmName angegeben ist, wird er dem Eintrag hinzugefügt. 2. fmName ist angegeben (durch ein Fachmodul aufgerufen) und eventType = „Op", so wird das Ereignis in das zugehörige Fachmodulprotokoll geschrieben. a. Gemäß den Festlegungen in den jeweiligen Fachmodulspezifikationen (FM_<fmName>_LOG_LEVEL), werden nur Ereignisse in das Fachmodulprotokoll geschrieben, deren severity mindestens dem jeweils dort festgelegten Wert entsprechen. 3. fmName ist nicht angegeben (Aufruf durch ein Fachmodul) und eventType = „Op", dann wird das Ereignis in das Systemprotokoll geschrieben. a. Gemäß den Festlegungen in LOG_LEVEL_SYSLOG werden nur Ereignisse in das Systemprotokoll geschrieben, deren Schwere mindestens dem Wert von LOG_LEVEL_SYSLOG entsprechen. 4. Wurde der TUC durch ein Fachmodul aufgerufen (fmName ist angegeben) und ist eventType = Perf, so wird das Ereignis in das zugehörige Fachmodul-Performanceprotokoll geschrieben. 5. Wurde der TUC nicht durch ein Fachmodul aufgerufen (fmName ist nicht angegeben) und ist eventType = Perf, so wird das Ereignis in das Konnektor-Performanceprotokoll geschrieben. Die geschriebenen Protokolleinträge MÜSSEN mindestens folgende Attribute beinhalten: • Datum und Uhrzeit • Übergebenes Ereignis Die Speicherung erfolgt rollierend. Übersteigt die Anzahl der Einträge im Sicherheitsprotokoll `SECURITY_LOG_SIZE`, so werden ältere Einträge überschrieben. Für die anderen Protokolle beginnt das Überschreiben, wenn der jeweilige Speicherplatz für das Protokoll erschöpft ist. Dabei werden die nach der Reihenfolge der Erstellung ältesten Einträge überschrieben, unabhängig vom Zeitstempel des Logeintrags. Ist der Zeitstempel eines überschriebenen Logeintrags jünger als LOG_DAYS bzw. FM_<fmName>_LOG_DAYS bzw. SECURITY_LOG_DAYS, so wird der Fehlerzustand EC_LOG_OVERFLOW ausgelöst.
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zu: a) Aufruf von TUC_KON_256 {

	topic = „LOG/ERROR“;
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („Error=$Fehlercode, Bedeutung=$Fehlertext“);
	doLog = false }
	b) Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes
	
	(→1) In das Sicherheitsprotokoll kann nicht geschrieben werden: Fehlercode: 4152
	(→2) In das Fachmodulprotokoll kann nicht geschrieben werden: Fehlercode: 4151
	(→3) In das Systemprotokoll kann nicht geschrieben werden: Fehlercode: 4150
	(→4) In das Fachmodul-Performanceprotokoll kann nicht geschrieben werden: Fehlercode: 4217
	(→5) In das Konnektor-Performanceprotokoll kann nicht geschrieben werden: Fehlercode: 4216

Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 288: TAB_KON_608 Fehlercodes TUC_KON_271 „Schreibe Protokolleintrag“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4150	Technical	Fatal	Fehler beim Schreiben des Systemprotokolls
4151	Technical	Fatal	Fehler beim Schreiben eines Fachmodulprotokolls
4152	Security	Error	Fehler beim Schreiben des Sicherheitsprotokolls
4216	Technical	Fatal	Fehler beim Schreiben des Konnektor-Performanceprotokolls
4217	Technical	Fatal	Fehler beim Schreiben eines Fachmodul-Performanceprotokolls

[<=]

Die Darstellung PIC_KON_118 veranschaulicht den Aufbau der Protokolle für Plattform und Fachmodule und die Steuerung der Protokolleinträge in TUC_KON_271 „Schreibe

Protokolleintrag".

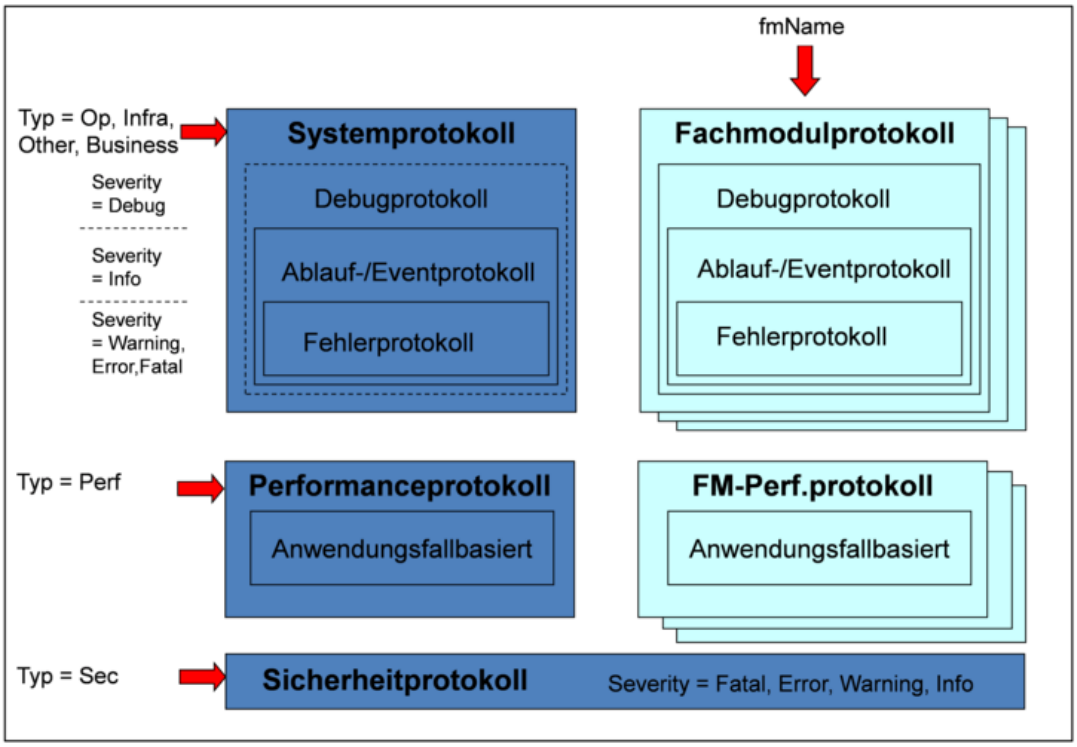


Abbildung 20: PIC_KON_118 Aufbau und Struktur der Protokolldateien für Plattform und Fachmodule

4.1.10.5 Operationen an der Außenschnittstelle

Keine

4.1.10.6 Betriebsaspekte

TIP1-A_4716 - Einsichtnahme und Veränderung der Protokolle

Der Administrator MUSS die durch den Protokollierungsdienst geschriebenen Protokolle über die Managementschnittstelle einsehen können.
Eine Veränderung des Sicherheitsprotokolls DARF für den Administrator NICHT möglich sein.
Das Löschen folgender Protokolle MUSS für den Administrator möglich sein:

- Systemprotokoll
- das jeweils durch <fmName> spezifizierte Fachmodulprotokoll
- Konnektor-Performanceprotokoll
- das jeweils durch <fmName> spezifizierte Fachmodul-Performanceprotokoll

Der Konnektor MUSS den Export von Protokolleinträgen oder ganzen Protokolldateien unterstützen.
Der Konnektor SOLL das Sortieren und Filtern der Protokolleinträge sowie das Suchen in den Protokolleinträgen unterstützen.
[<=]

TIP1-A_4996 - Hinweis auf neue Sicherheitsprotokolleinträge
Nachdem sich der Administrator an der Managementschnittstelle angemeldet hat, MUSS der Konnektor ihn automatisch auf Sicherheitsprotokolleinträge hinweisen, die seit dem Ausloggen dieses Administrator aufgelaufen sind.
[<=]

TIP1-A_4717 - Konfigurationswerte des Protokollierungsdienstes
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_609 vorzunehmen:

Tabelle 289: TAB_KON_609 Konfigurationswerte des Protokollierungsdienstes (Administrator)

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
LOG_LEVEL_ SYSLOG	Info, Warning, Error, Fatal	Der Administrator MUSS den Detaillierungsgrad des Systemprotokolls durch Festlegung der Mindest-Schwere zu protokollierender Einträge festlegen können. Default-Wert: Warning
FM_<fmName>_ LOG_LEVEL	Debug, Info, Warning, Error, Fatal	Der Administrator MUSS den Detaillierungsgrad des Fachmodulprotokolls durch Festlegung der Mindest-Schwere zu protokollierender Einträge festlegen können. Default-Wert: Warning
SECURITY_LOG_SIZE	X Einträge	Der Administrator MUSS die Größe des Sicherheitsprotokolls angeben können (Anzahl der Einträge im Ringbuffer). Mindestgröße: >= 10.000 Maximalgröße: herstellerspezifisch Default-Wert: >= 50.000
SECURITY_LOG_DAYS	X Tage	Der Administrator MUSS die erwartete Anzahl der im Sicherheitsprotokoll gespeicherten Tage im Bereich 10 bis 365 konfigurieren können. Default-Wert: 180
LOG_DAYS	X Tage	Der Administrator MUSS die Anzahl der gespeicherten Tage für das Systemprotokoll und das Performanceprotokoll festlegen können. Der Konnektor kann Protokolleinträge, die älter als LOG_DAYS sind, zyklisch löschen. Dabei DARF der eingestellte Wert NICHT unter der Mindestgröße von 10 Tagen oder über der Maximalgröße von einem Jahr (365 Tage) liegen. Default-Wert: 180

FM_<fmName>_ LOG_DAYS	X Tage	Der Administrator MUSS die Anzahl der gespeicherten Tage für die fachmodulspezifischen Protokolle festlegen können. Es kann je Fachmodul einen Konfigurationsparameter für LOG_DAYS geben, der gemeinsam für das Fachmodulprotokoll und das Fachmodul-Performanceprotokoll gilt. Der Konnektor kann Protokolleinträge, die älter als FM_<fmName>LOG_DAYS sind, zyklisch löschen. Dabei DARF der eingestellte Wert NICHT unter der Mindestgröße von 10 Tagen oder über der Maximalgröße von einem Jahr (365 Tage) liegen. Default-Wert: 180 Die Definition des fachmodulspezifischen Konfigurationswertes ist Bestandteil der entsprechenden Fachmodulspezifikation. Ist kein fachmodulspezifischer Konfigurationsparameter spezifiziert, dann gilt LOG_DAYS.
LOG_SUCCESSFUL_ CRYPTO_OPS	Enabled/Disabled	Der Administrator MUSS festlegen können, ob auch erfolgreich ausgeführte Kryptooperationen im Sicherheitslog protokolliert werden sollen. Default-Wert: Disabled

[<=]

4.1.10.6.1 TUC_KON_272 „Initialisierung Protokollierungsdienst

TIP1-A_4718 - TUC_KON_272 „Initialisierung Protokollierungsdienst"
Der Konnektor MUSS den technischen Use Case TUC_KON_272 „Initialisierung Protokollierungsdienst" umsetzen.

Tabelle 290: TAB_KON_610 – TUC_KON_272 „Initialisierung Protokollierungsdienst"

Element	Beschreibung
Name	TUC_KON_272 „Initialisierung Protokollierungsdienst"
Beschreibung	Der Konnektor muss zum Bootup den Protokollierungsdienst starten und die Existenz und Schreibbarkeit der Protokolle sicherstellen.
Eingangs anforderung	Keine
Auslöser und Vorbedingungen	Bootup
Eingangsdaten	Keine
Komponenten	Konnektor

Ausgangsdaten	Keine
Standardablauf	1. Prüfen, ob Schreib-/Lesezugriff auf Sicherheitsprotokoll möglich ist 2. Prüfen, ob Schreib-/Lesezugriff auf Systemprotokoll möglich ist 3. Prüfen, ob Schreib-/Lesezugriff auf Fachmodulprotokolle möglich ist 4. Prüfen, ob Schreib-/Lesezugriff auf Konnektor-Performanceprotokoll möglich ist 5. Prüfen, ob Schreib-/Lesezugriff auf Fachmodul-Performanceprotokolle möglich ist
Varianten/ Alternativen	Keine

Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zu:
	a) Aufruf von TUC_KON_256 mit folgenden Parametern {
	topic = „LOG/ERROR“;
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („Error=$Fehlercode, Bedeutung=$Fehlertext“);
	doLog = false }
	b) Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes
	(→1) Zugriff nicht möglich: Fehlercode: 4153
	(→2) Zugriff nicht möglich: Fehlercode: 4154
	(→3) Zugriff nicht möglich: Fehlercode: 4155
	(→4) Zugriff nicht möglich: Fehlercode: 4218
	(→5) Zugriff nicht möglich: Fehlercode: 4219

Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 291: TAB_KON_611 Fehlercodes TUC_KON_272 „Initialisiere Protokollierungsdienst“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4153	Technical	Fatal	Zugriff auf Sicherheitsprotokoll nicht möglich
4154	Technical	Fatal	Zugriff auf Systemprotokoll nicht möglich
4155	Technical	Fatal	Zugriff auf Fachmodulprotokolle nicht möglich
4218	Technical	Fatal	Zugriff auf Konnektor-Performanceprotokoll nicht möglich
4219	Technical	Fatal	Zugriff auf Fachmodul-Performanceprotokoll nicht möglich

[<=]

4.1.11 TLS-Dienst

Fachmodule müssen gesicherte Verbindungen zu Fachdiensten in der TI aufbauen können. Dabei sollen sie sich mit einer Organisationsidentität (einer SM-B) authentisieren können. Der TLS-Dienst stellt hierfür TUCs für einen TLS-Verbindungsaufbau und -Verbindungsabbau zur Verfügung. Die gesicherte Kommunikation selbst erfolgt dann durch das Fachmodul unter Nutzung der etablierten Verbindung.

Die Funktionalität steht nur zur Verfügung, wenn MGM_LU_ONLINE aktiv ist (siehe Kapitel 4.3.6)

4.1.11.1 Funktionsmerkmalweite Aspekte

4.1.11.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_4719 - TLS-Dienst reagiert auf Veränderung LU_ONLINE
Tritt das Ereignis „MGM/LU_CHANGED/LU_ONLINE" ein, so MUSS

- wenn „Active=Enabled" der Dienst bereitgestellt werden
- wenn „Active=Disabled" der Dienst gestoppt werden.
  Sind TLS-Verbindungen aktiv, so MUSS für jede TUC_KON_111 "Kartenbasierte TLS-Verbindung abbauen" gerufen werden.

[<=]

4.1.11.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.1.11.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.11.4.1 TUC_KON_110 „Kartenbasierte TLS-Verbindung aufbauen"

TIP1-A_4720 - TUC_KON_110 „Kartenbasierte TLS-Verbindung aufbauen"
Der Konnektor MUSS den technischen Use Case "Kartenbasierte TLS-Verbindung aufbauen" gemäß TUC_KON_110 umsetzen.

Tabelle 292: TAB_KON_773 – TUC_KON_110 „Kartenbasierte TLS-Verbindung aufbauen"

Element	Beschreibung
Name	TUC_KON_110 „Kartenbasierte TLS-Verbindung aufbauen"
Beschreibung	Der TUC_KON_110 „Kartenbasierte TLS-Verbindung aufbauen" baut eine TLS-Verbindung zur angegebenen Zieladresse auf. Dabei kann für eine gegenseitige Authentisierung eine SM-B verwendet werden.
Auslöser	Aufruf durch ein Fachmodul
Vorbedingungen	Die für die Authentisierung adressierte Karte muss freigeschaltet sein
Eingangsdaten	• roleToMatch – *optional/verpflichtend, wenn Rollenprüfung durchgeführt werden soll*

	• cardSession – *optional/verpflichtend, wenn Clientauthentisierung durchgeführt werden soll* (CardSession einer SM-B) • targetUri (URI des Verbindungsziels)
Komponenten	Konnektor, eHealth-Kartenterminal, Karte, Server des Fachdienstes
Ausgangsdaten	• tlsConnectiontId (ConnectionIdentifier der TLS-Verbindung)

Standardablauf	Der Konnektor MUSS folgende Schritte durchlaufen:
	1. Auflösen des FQDN im targetUri per 'TUC_KON_361 „DNS Namen auflösen"
	2. TLS-Verbindung mit ermittelter Adresse aufbauen:
	a) Prüfe Server-Zertifikat mittels TUC_KON_037 „Zertifikat prüfen" {
	certificate = C.FD.TLS-S;
	qualifiedCheck = not_required;
	offlineAllowNoCheck = true;
	policyList = oid_fd_tls_s;
	intendedKeyUsage= intendedKeyUsage(C.FD.TLS-S);
	intendedExtendedKeyUsage = id-kp-serverAuth;
	validationMode = OCSP}
	Das Server-Zertifikat MUSS C.FD.TLS-S sein
	b) Prüfe in a) zurückgegebene Rolle („ermittelte Rolle") == roleToMatch
	c) Wenn cardSession übergeben: Clientauthentisierung mittels ID.HCI.AUT
	3. tlsConnectiontId der erzeugten Verbindung zurückgeben

Varianten/ Alternativen	• Keine
Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu einem Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes (→1) Der Name der Gegenstelle kann nicht aufgelöst werden (→2b) Rollenprüfung fehlgeschlagen: Fehlercode 4220 (→2) Server konnte nicht authentisiert werden: Fehlercode 4156 (→2) Clientauthentisierung fehlgeschlagen: Fehlercode 4157
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 293: TAB_KON_612 Fehlercodes TUC_KON_110 „Kartenbasierte TLS-Verbindung aufbauen"

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4156	Security	Error	Server konnte bei TLS-Verbindungsaufbau nicht authentisiert werden
4157	Security	Error	Clientauthentisierung bei TLS-Verbindungsaufbau fehlgeschlagen
4220	Security	Error	Rollenprüfung bei TLS-Verbindungsaufbau fehlgeschlagen

[<=]

4.1.11.4.2 TUC_KON_111 „Kartenbasierte TLS-Verbindung abbauen"

TIP1-A_4721 - TUC_KON_111 „Kartenbasierte TLS-Verbindung abbauen"
Der Konnektor MUSS den technischen Use Case "Kartenbasierte TLS-Verbindung abbauen" gemäß TUC_KON_111 umsetzen.

Tabelle 294: TAB_KON_774 - TUC_KON_111 „Kartenbasierte TLS-Verbindung abbauen"

Element	Beschreibung
Name	TUC_KON_111 „Kartenbasierte TLS-Verbindung abbauen"
Beschreibung	Der TUC_KON_111 „Kartenbasierte TLS-Verbindung abbauen" dient der geregelten Beendigung einer TLS-Verbindung, die zuvor über TUC_KON_110 aufgebaut wurde.
Auslöser	Aufruf durch ein Fachmodul
Vorbedingungen	Mittels TUC_KON_110 wurde eine TLS-Verbindung aufgebaut
Eingangsdaten	• tlsConnectiontId (ConnectionIdentifier der TLS-Verbindung)
Komponenten	Konnektor, Server des Fachdienstes
Ausgangsdaten	Keine
Standardablauf	Der Konnektor MUSS folgende Schritte durchlaufen: 1. Trennen der über tlsConnectiontId adressierten TLS-Verbindung
Varianten/ Alternativen	keine
Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu einem Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes: (→1) Keine Verbindung mit angegebenem TLSConnectionIdentifier vorhanden: Fehlercode 4158
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 295: TAB_KON_613 Fehlercodes TUC_KON_111 „Kartenbasierte TLS-Verbindung abbauen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4158	Technical	Error	Adressierte TLS-Verbindung nicht vorhanden

[<=]

4.1.11.5 Operationen an der Außenschnittstelle

Keine.

4.1.11.6 Betriebsaspekte

TIP1-A_4722 - TLS-Dienst initialisieren
Wenn MGM_LU_ONLINE = „Enabled", MUSS der Basisdienst TLS-Dienst nach dem Bootup zur Nutzung zur Verfügung stehen.
Wenn MGM_LU_ONLINE = „Disabled", DARF der Basisdienst TLS-Dienst nach dem Bootup NICHT zur Nutzung zur Verfügung stehen.
[<=]

4.1.12 LDAP-Proxy

Der Konnektor ermöglicht es Clientsystemen und Fachmodulen durch Nutzung des LDAP-Proxies Daten aus dem Verzeichnisdienst der TI-Plattform (VZD) abzufragen. Die Kommunikation erfolgt über das LDAPv3 Protokoll.

Die Funktionalität steht nur zur Verfügung, wenn MGM_LU_ONLINE=Enabled ist (siehe Kapitel 4.3.6)

4.1.12.1 Funktionsmerkmalweite Aspekte

Keine.

4.1.12.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_5516 - LDAP-Proxy reagiert auf Veränderung LU_ONLINE
Tritt das Ereignis „MGM/LU_CHANGED/LU_ONLINE" ein, so MUSS

- wenn „Active=Enabled" der Dienst bereitgestellt werden
- wenn „Active=Disabled" der Dienst gestoppt werden.
  Ist eine Verbindung zum VZD aktiv, so MUSS diese abgebaut werden.

[<=]

4.1.12.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.1.12.4 Interne TUCs, auch durch Fachmodule nutzbar

4.1.12.4.1 TUC_KON_290 „LDAP-Verbindung aufbauen"

TIP1-A_5517-02 - Konnektor, TUC_KON_290 „LDAP-Verbindung aufbauen"
Der Konnektor MUSS den technischen Use Case TUC_KON_290 „LDAP-Verbindung aufbauen" gemäß TAB_KON_805 umsetzen.

Tabelle 296: TAB_KON_805 - TUC_KON_290 „LDAP-Verbindung aufbauen"

Element	Beschreibung
Name	TUC_KON_290 „LDAP-Verbindung aufbauen"
Beschreibung	Initiiert durch einen Verbindungsaufbau des LDAP-Clients zum Konnektor baut der Konnektor eine TLS-gesicherte Verbindung zum VZD auf.
Auslöser	LDAP (oder LDAPS wenn ANCL_TLS_MANDATORY=Enabled) Verbindungsaufbau von einem Fachmodul oder einem Clientsystem ist abgeschlossen. Bei Verwendung von LDAPS authentisiert sich der Konnektor beim LDAP-Client mit der Identität ID.AK.AUT.
Vorbedingungen	• MGM_LU_ONLINE=Enabled
Eingangsdaten	keine
Komponenten	Konnektor, VZD
Ausgangsdaten	keine
Standardablauf	1. Der Konnektor ermittelt den FQDN und Port des VZD durch eine DNS-SD Namensauflösung gemäß [RFC6763] mit dem Bezeichner "_ldap._tcp.vzd.<DNS_TOP_LEVEL_DOMAIN_TI>." 2. Der Konnektor baut eine LDAPS-Verbindung zum VZD auf. Dabei wird das Serverzertifikat des Verzeichnisdienst C.ZD.TLS-S nach TUC_PKI_018 geprüft (PolicyList: oid_zd_tls_s (gemäß gemSpec_OID), intendedKeyUsage: intendedKeyUsage(C.ZD.TLS-S), ExtendedKeyUsages: serverAuth (1.3.6.1.5.5.7.3.1), Offlinemodus: nein, TOLERATE_OCSP_FAILURE: false , Prüfmodus: OCSP)
Varianten/Alternativen	keine
Fehlerfälle	
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

[<=]

4.1.12.4.2 TUC_KON_291 „Verzeichnis abfragen"

TIP1-A_5518 - Konnektor, TUC_KON_291 „Verzeichnis abfragen"
Der Konnektor MUSS den technischen Use Case TUC_KON_291 „Verzeichnis abfragen" gemäß TAB_KON_815 umsetzen.

Tabelle 297: TAB_KON_815 – TUC_KON_291 „Verzeichnis abfragen"

Element	Beschreibung
Name	TUC_KON_291 „Verzeichnis abfragen"
Beschreibung	Der Konnektor leitet als LDAP-Proxy einen Search Request des LDAP-Clients an den VZD weiter. Vom VZD empfängt der Konnektor eine Search Response und leitet diese an den LDAP-Client weiter.
Auslöser	Aufruf durch einen LDAPv3 Search Request von einen Fachmodul-TUC oder einem Clientsystem
Vorbedingungen	• MGM_LU_ONLINE=Enabled • Eine LDAPv3-Verbindung LDAP-Client sowie eine LDAPv3 Verbindung vom Konnektor zum VZD wurden aufgebaut (TUC_KON_290 „LDAP-Verbindung aufbauen")
Eingangsdaten	• LDAPv3 Search Request gemäß [RFC4511]#4.5.1
Komponenten	Konnektor, VZD
Ausgangsdaten	• LDAPv3 Search Response gemäß [RFC4511]#4.5.2
Standardablauf	1. Der Konnektor führt TUC_VZD_0001 „search_Directory" mit dem vom LDAP-Client empfangenen Search Request als Eingangsdaten aus und empfängt die LDAPv3 Search Response vom VZD (entspricht den Ausgangsdaten).
Varianten/Alternativen	keine
Fehlerfälle	Auftretende Fehler werden gemäß [RFC4511]# Appendix A behandelt.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

[<=]

4.1.12.4.3 TUC_KON_292 „LDAP-Verbindung trennen"

TIP1-A_5519 - Konnektor, TUC_KON_292 „LDAP-Verbindung trennen"
Der Konnektor MUSS den technischen Use Case „LDAP-Verbindung trennen" gemäß TAB_KON_816 umsetzen.

Tabelle 298: TAB_KON_816 – TUC_KON_292 „LDAP-Verbindung trennen"

Element	Beschreibung
Name	TUC_KON_292 „LDAP-Verbindung trennen"
Beschreibung	Der Konnektor beendet die Verbindung zum VZD und zum LDAP-Client.
Auslöser	Aufruf durch einen LDAPv3 Unbind Request von einen Fachmodul-TUC oder einem Clientsystem
Vorbedingungen	• MGM_LU_ONLINE=Enabled • Eine LDAPv3-Verbindung LDAP-Client sowie eine LDAPv3 Verbindung vom Konnektor zum VZD wurden aufgebaut (TUC_KON_290 „Verbindungsaufbau zum VZD“)
Eingangsdaten	keine
Komponenten	Konnektor, VZD
Ausgangsdaten	keine
Standardablauf	1. Der Konnektor empfängt vom LDAP-Client einen Unbind Request gemäß [RFC4511]#4.3. 2. Der Konnektor sendet zum VZD einen Unbind Request. 3. Der Konnektor beendet die Verbindung zum VZD und zum LDAP Client gemäß [RFC4511]#5.3.
Varianten/Alternativen	keine
Fehlerfälle	Auftretende Fehler werden gemäß [RFC4511]# Appendix A behandelt.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

[<=]

4.1.12.4.4 TUC_KON_293 „Verzeichnisabfrage abbrechen"

TIP1-A_5520 - Konnektor, TUC_KON_293 „Verzeichnisabfrage abbrechen"
Der Konnektor MUSS den technischen Use Case TUC_KON_293 „Verzeichnisabfrage abbrechen" gemäß TAB_KON_817 umsetzen.

Tabelle 299: TAB_KON_817 – TUC_KON_293 „Verzeichnisabfrage abbrechen"

Element	Beschreibung
Name	TUC_KON_293 „Verzeichnisabfrage abbrechen"
Beschreibung	Der Konnektor bricht einen unbeantworteten Search Request ab.
Auslöser	Aufruf durch einen LDAPv3 Abandon Request von einen Fachmodul-TUC oder einem Clientsystem
Vorbedingungen	• MGM_LU_ONLINE=Enabled • Ein Search Request wurde vom Konnektor empfangen und an den VZD weitergeleitet (TUC_KON_291 „Verzeichnis Abfragen"). Der Request wurde vom VZD noch nicht beantwortet.
Eingangsdaten	keine
Komponenten	Konnektor, VZD
Ausgangsdaten	keine
Standardablauf	1. Der Konnektor empfängt vom LDAP-Client einen Abandon Request gemäß [RFC4511]#4.11. 2. Der Konnektor sendet zum VZD einen Abandon Request gemäß [RFC4511]#4.11
Varianten/Alternativen	keine
Fehlerfälle	Auftretende Fehler werden gemäß [RFC4511]# Appendix A behandelt.
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

[<=]

4.1.12.5 Operationen an der Außenschnittstelle

4.1.12.5.1 Unterstützte LDAPv3 Operationen

TIP1-A_5521 - Konnektor, LDAPv3 Operationen
Der Konnektor MUSS an der Client-Schnittstelle die folgenden LDAPv3 Operationen gemäß [RFC4511] anbieten.

- Bind Operation
- Unbind Operation

- Search Operation
- Abandon Operation

Andere LDAPv3 Operationen werden mit dem LDAP-Fehler unwillingToPerform (53) beantwort.
Wenn ANCL_TLS_MANDATORY=Enabled, muss der Konnektor sicherstellen, dass nur über eine LDAPS-Verbindung (Voreinstellung TCP Port 636) Daten abgefragt werden können.
Wenn ANCL_TLS_MANDATORY=Disabled, muss der Konnektor sicherstellen, dass über eine LDAP-Verbindung (Voreinstellung TCP Port 389) und über eine LDAPS-Verbindung (Voreinstellung TCP Port 636) Daten abgefragt werden können.
Fehler müssen gemäß [RFC4511]#Appendix A behandelt werden.
[<=]

4.1.12.6 Betriebsaspekte

keine

4.1.13 Authentifizierungsdienst

Der Authentifizierungsdienst bietet Clientsystemen und Fachmodulen eine Schnittstelle zum Signieren von Binärstrings zum Zweck der externen Authentisierung.

Innerhalb des Authentifizierungsdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): *keine Events vorhanden*
- Konfigurationsparameter: *keine Konfigurationsparameter vorhanden*

Eine Prüfung der Signatur bietet der Konnektor nicht an.

4.1.13.1 Funktionsmerkmalweite Aspekte

4.1.13.1.1 Externe Authentisierung

TIP1-A_5437-02 - Signaturverfahren für externe Authentisierung
Der Signaturdienst MUSS für die Operation ExternalAuthenticate die Signaturverfahren entsprechend TAB_KON_780 – Signaturverfahren Externe Authentisierung unterstützen.

Tabelle 300: TAB_KON_780 – Signaturverfahren Externe Authentisierung

Signaturformat	Standard	Dokument formate	QES/ nonQES	Bemerkung
PKCS#1 (V2.1)	[RFC3447]	Binär	nonQES	Die Low-Level-Signatur von Bitstrings DARF NUR in Verbindung mit dem zur Authentisierung vorgesehenen Schlüssel des HBAx und des SM-B genutzt werden. Die Nutzung ist auf Bitstrings (Hash-Werte) von maximal 512 bit
ECDSA	[BSI-TR-03111]	Binär	nonQES	

				Länge beschränkt.

[<=]

TIP1-A_5149-01 - ExternalAuthenticate nur für Authentisierung mit HBAx und SM-B nutzen
Der Hersteller des Konnektors MUSS den Anwender (Clientsystem) im Handbuch des Konnektors geeignet und ausreichend darüber informieren, dass die Operation ExternalAuthenticate nur zu Authentisierungszwecken mit dem Authentisierungsschlüssel des HBAx und des SM-B verwendet werden darf.**[<=]**

4.1.13.2 Durch Ereignisse ausgelöste Reaktionen

keine

4.1.13.3 Interne TUCs

keine

4.1.13.4 Operationen an der Außenschnittstelle

TIP1-A_5665-03 - Basisdienst Authentifizierungsdienst
Der Konnektor MUSS Clientsystemen den Basisdienst Authentifizierungsdienst anbieten.

Tabelle 301: TAB_KON_839 Basisdienst Authentifizierungsdienst

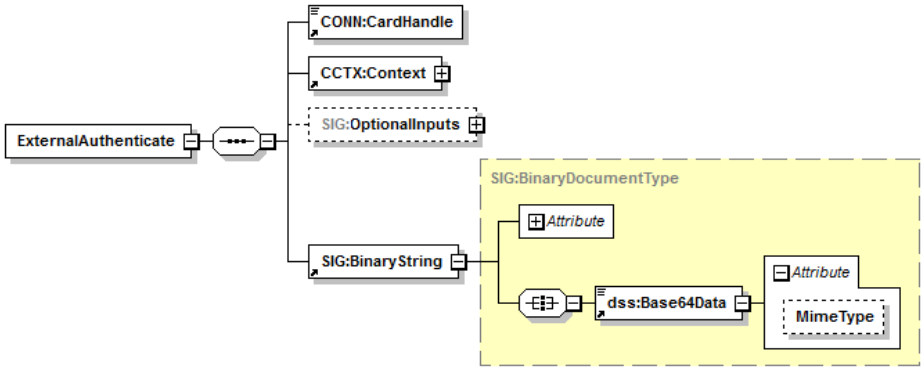
Name	`AuthSignatureService`	
Version (KDV)	7.4.0 (WSDL-Version) 7.4.1 (WSDL-Version)	
Namensraum	Siehe GitHub	
Namensraum-Kürzel	`SIG für Schema und SIGW für WSDL`	
Operationen	**Name**	**Kurzbeschreibung**
	`ExternalAuthenticate`	Binärstring signieren (nonQES)
WSDL	`AuthSignatureService_V7_4_1.wsdl` `AuthSignatureService.wsdl (WSDL-Version 7.4.0)`	
Schema	`Kein`	

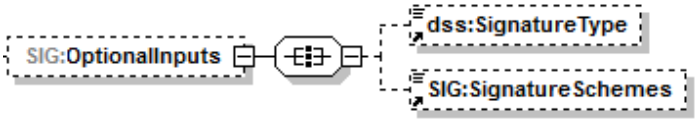
[<=]

4.1.13.4.1 ExternalAuthenticate

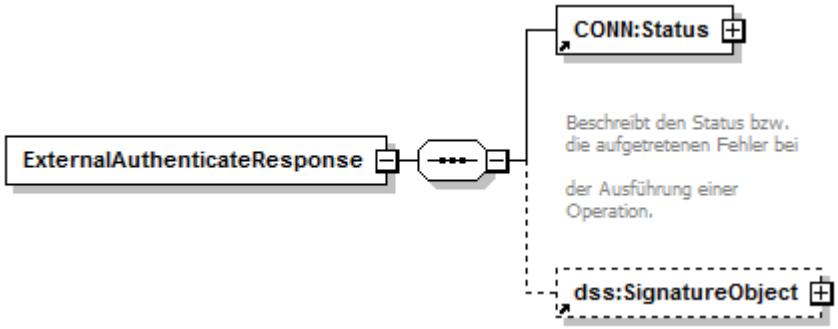
TIP1-A_5439 - Operation ExternalAuthenticate
Der Authentifizierungsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation ExternalAuthenticate anbieten**.**

Tabelle 302: TAB_KON_781 Operation ExternalAuthenticate

<table>
<tr><td>Name</td><td colspan="2">ExternalAuthenticate</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Diese Operation versieht einen Binärstring der maximalen Länge 512 Bit mit einer nicht-qualifizierten elektronischen Signatur (nonQES). Dazu wird das Signaturverfahren PKCS#1 oder ECDSA verwendet. Das AUT-Zertifikat der SM-B und das AUT-Zertifikat des HBAx werden unterstützt.</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 

<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>CONN:
CardHandle</td><td>Identifiziert die zu verwendende Signaturkarte.
Die Operation unterstützt HBAx und SM-B.</td></tr>
<tr><td>CCTX:
Context</td><td>Aufrufkontext für HBAx:
MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend
Aufrufkontext für SM-B:
MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet</td></tr>
<tr><td>SIG:
Optional
Inputs</td><td>Enthält optionale Eingangsparameter:
 
</td></tr>
<tr><td>SIG:
Binary
String</td><td>Dieses Element enthält im Kindelement dss:Base64Data den zu signierenden Binärstring.
Das XML Attribut SIG:BinaryString/dss:Base64Data/@MimeType MUSS den Wert "application/octet-stream" haben.
Die maximale Länge des Binärstrings beträgt 512 Bit entsprechend der maximal zu erwartenden Hash-Größe.
Aus der Länge des Binärstrings wird auf das verwendete Hashverfahren geschlossen. Es werden folgende Längen unterstützt:
• 256 Bit: SHA-256 (OID 2.16.840.1.101.3.4.2.1)</td></tr>
</table></td></tr>
</table>

		• 384 Bit: SHA-384 (OID 2.16.840.1.101.3.4.2.2) • 512 Bit: SHA-512 (OID 2.16.840.1.101.3.4.2.3) Im Falle des Signaturverfahrens RSASSA-PKCS1-v1_5 werden SHA-256, SHA-384 und SHA-512 unterstützt. Im Falle des Signaturverfahrens RSASSA-PSS wird SHA-256 unterstützt. Im Falle des Signaturverfahrens ECDSA wird SHA-256 unterstützt. Für die Signaturerstellung gilt: • Im Falle des Signaturverfahrens RSASSA-PKCS1-v1_5 beginnt der Konnektor die Ausführung der Methode EMSA-PKCS1-v1_5-ENCODE nach [RFC3447], Abschnitt 9.2, mit Schritt 2, Erstellung des DigestInfo-Datenfeldes. • Im Falle des Signaturverfahrens RSASSA-PSS beginnt der Konnektor die Ausführung der Methode EMSA-PSS-ENCODE nach [RFC3447], Abschnitt 9.1.1, mit Schritt 3. • Im Falle des Signaturverfahrens ECDSA erfolgt die Signaturerstellung gemäß [BSI-TR-03111]#4.2.1. Als Eingangsparameter wird der Hash vom Aufrufer in SIG: BinaryString übergeben.
	`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element wird der Typ der zu erzeugenden Signatur bestimmt. Als Signaturtyp wird unterstützt : • **PKCS#1-Signatur** Durch Übergabe der URI urn:ietf:rfc:3447 wird eine PKCS#1 (Version 2.1) Signatur gemäß [RFC3447] erzeugt, die als `dss:Base64Signature` mit der oben genannten URI zurückgeliefert wird. • **ECDSA-Signatur** Durch Übergabe der URI urn:bsi:tr:03111:ecdsa wird eine ECDSA-Signatur gemäß [BSI-TR-03111]#4.2.1 erzeugt, die als `dss:Base64Signature` mit der oben genannten URI zurückgeliefert wird. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder Signaturvariante). Fehlt dieses Element, so wird ebenfalls der Signaturtyp PKCS#1-Signatur verwendet.

	SIG: Signature Schemes	Durch dieses Element wird für PKCS#1-Signaturen zwischen den folgenden SignatureScheme-Optionen unterschieden: • RSASSA-PSS • RSASSA-PKCS1-v1_5 Fehlt dieses Element, so wird als Default-SignatureScheme RSASSA-PSS gewählt.
Rückgabe		
	CONN: Status	Enthält den Status der ausgeführten Operation.
	dss: Signature Object	Enthält im Erfolgsfall die erzeugte Signatur in Form eines dss:SignatureObject-Elements gemäß [OASIS-DSS] (Abschnitt 3.2). Der Signaturwert wird im XML-Element dss:SignatureObject/dss:Base64Signature übergeben. Die Signatur wird binär gemäß [BSI-TR-03111]#5.2.2 in der ASN.1 Struktur ECDSA-Sig-Value zurückgegeben. Das XML-Attribut dss:SignatureObject/dss:Base64Signature/@Type kennzeichnet durch den Wert: • urn:ietf:rfc:3447 den Signatur-Typ PKCS#1 bzw. • urn:bsi:tr:03111:ecdsa den Signatur-Typ ECDSA. Die XML-Elemente dss:SignatureObject/ds:Signature dss:SignatureObject/dss:Timestamp dss:SignatureObject/dss:SignaturePtr dss:SignatureObject/dss:Other werden nicht verwendet.
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Der Ablauf der Operation ExternalAuthenticate ist in Tabelle TAB_KON_782 beschrieben:

Tabelle 303: TAB_KON_782 Ablauf Operation ExternalAuthenticate

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Alle übergebenen Parameterwerte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf TUC_KON_000 {$context.mandantId; $context.clientsystemId; $context.workplaceId; $context.userId; $cardHandle} Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 { MandantId, CsId, CardHandle, UserId }

4.	TUC_KON_218 „Signiere“	Signaturberechnung durch Aufruf des TUC_KON_218 {
		PinRef = PIN.CH bzw. PIN.SMC;
		KeyRef = PrK.HP.AUT bzw. PrK.HCI.AUT;
		AlgorithmusID = signPKCS1_V1_5 oder signPSS oder signECDSA;
		DTBS = Binärstring
		}

Tabelle 304: TAB_KON_783 Übersicht Fehler Operation ExternalAuthenticate

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4058	Security	Error	Aufruf nicht zulässig

Die folgende Tabelle führt die zulässigen privaten Schlüssel für die Operation ExternalAuthenticate auf:

Tabelle 305: TAB_KON_784 Privater Schlüssel je Karte für ExternalAuthenticate

Karte	Schlüssel
SM-B	PrK.HCI.AUT in DF.ESIGN
HBAx	PrK.HP.AUT in DF.ESIGN

[<=]

4.1.13.5 Betriebsaspekte

Keine

4.1.14 Betriebsdatenmeldedienst

A_21136 - Konnektor:Betriebsdaten - Nutzung der Operation I_Registration_Service#sendData
Der Konnektor MUSS die Operation I_Registration_Service#sendData benutzen, um die Betriebsdaten täglich zu versenden.
[<=]

A_21137 - Konnektor:Betriebsdaten - Formatierung der Betriebsdaten
Der Konnektor MUSS die Betriebsdaten als XML-Dokument gemäß dem Schema „conn/OperatingData.xsd" mit

- dem MimeType "text/xml" und
- dem Type "OperatingDataConnector"

senden.**[<=]**

A_21225 - Konnektor:Betriebsdaten - Annotations im XML-Schema
Der Konnektor MUSS die XML Elemente der Betriebsdaten gemäß der Annotations im Schema OperatingData.xsd befüllen.**[<=]**

A_21138 - Konnektor:Betriebsdaten - Fehlerbehandlung
Liefert I_Registration_Service einen Fehler, MUSS der Konnektor das Senden der Betriebsdaten 3 Mal im Abstand von 5 Minuten erneut versuchen.**[<=]**

A_21139 - Konnektor:Betriebsdaten - Soap Operation sendData nicht vorhanden
Meldet I_Registration_Service, dass die Soap Operation sendData nicht vorhanden ist, DARF der Konnektor die Operation NICHT sofort wiederholen. Erst zum nächsten regulären Termin soll wieder gesendet werden.**[<=]**

A_21140 - Konnektor:Betriebsdaten - Keine personenbezogenen und medizinischen Daten senden
Der Konnektor DARF NICHT personenbezogene, personenbeziehbare oder medizinische Daten senden.**[<=]**

4.2 Netzkonnektor

4.2.1 Anbindung LAN/WAN

Unter Anbindung LAN/WAN werden die Mechanismen beschrieben, mit denen der Konnektor auf der einen Seite in das lokale Netz der Einsatzumgebung, auf der anderen Seite in die TI bzw. die Bestandsnetze angebunden wird. Diese wesentlichen Aspekte betreffen Routing und Firewall.

Innerhalb des Kapitels Anbindung LAN/WAN werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „ANLW"
- Konfigurationsparameter: „ANLW_"

4.2.1.1 Funktionsmerkmalweite Aspekte

TIP1-A_4723 - Verhalten als IPv4 Router
Der Konnektor MUSS sich nach den in [RFC1812#1.1.3] definierten Rahmenbedingungen als IP Version 4 (IPv4) Router verhalten.

Hiervon ausgenommen sind die in den folgenden Kapiteln aufgeführten Anforderungen des [RFC1812]:

- 7.2 INTERIOR GATEWAY PROTOCOLS
- 7.3 EXTERIOR GATEWAY PROTOCOLS
- 7.5 FILTERING OF ROUTING INFORMATION
- 7.6 INTER-ROUTING-PROTOCOL INFORMATION EXCHANGE
- 8. APPLICATION LAYER - NETWORK MANAGEMENT PROTOCOLS
- 9. APPLICATION LAYER - MISCELLANEOUS PROTOCOLS
- 10. OPERATIONS AND MAINTENANCE

Die in [RFC2644] geforderten Aktualisierungen zum [RFC1812] müssen vom Konnektor umgesetzt werden.
[<=]

TIP1-A_5406 - IP-Pakete mit Source Route Option
Der Konnektor DARF NICHT IP-Pakete mit gesetzter Source Route Option gemäß [RFC791] erzeugen oder weiterleiten.
[<=]

In der folgenden Anforderung wird die Terminologie gemäß [RFC2663] verwendet.

TIP1-A_5407 - NAT-Umsetzung im Konnektor
Der Konnektor MUSS für die Kommunikation aus den Adressbereichen NET_LEKTR-Umgebung mit den Adressbereichen NET_TI_OFFENE_FD und ANLW_BESTANDSNETZE eine Network Address Port Translation (NAPT) gemäß [RFC3022#2.2, 3, 4.1-4.3] vornehmen.
Für die Umsetzung der Private Local Address aus den Adressbereichen der Einsatzumgebung MUSS die IP-Adresse VPN_TUNNEL_TI_INNER_IP als Global Address genutzt werden.
Der Konnektor MUSS für die Kommunikation aus den Adressbereichen der NET_LEKTR-Umgebung mit dem Internet über den VPN-Tunnel SIS eine Network Address Port Translation (NAPT) gemäß RFC3022#2.2, 3, 4.1-4.3 vornehmen. Für die Umsetzung der Local Address MUSS die IP-Adresse VPN_TUNNEL_SIS_INNER_IP als Global Address genutzt werden.
[<=]

TIP1-A_4724 - LAN-Adapter
Der Konnektor MUSS sicherstellen, dass nur über den LAN-Adapter (Adressen aus ANLW_LAN_NETWORK_SEGMENT oder Adressen aus einem der Netzwerksegmente in ANLW_LEKTR_INTRANET_ROUTES) mit den Clientsystemen und den Kartenterminals kommuniziert werden kann.
[<=]

TIP1-A_4725 - WAN-Adapter
Für den Betrieb in Reihe (ANLW_ANBINDUNGS_MODUS=InReihe) MUSS der Konnektor den WAN-Adapter für den Zugang zum Internet über das IAG der Einsatzumgebung verwenden.
[<=]

TIP1-A_4726 - Internet Anbindung nur bei MGM_LU_ONLINE
Der Hersteller des Konnektors MUSS sicherstellen, dass eine Anbindung an das Transportnetz/Internet nur möglich ist, wenn (MGM_LU_ONLINE=Enabled) gesetzt ist.
[<=]

TIP1-A_4728 - Nur IPv4. IPv6 nur hardwareseitig vorbereitet
Der Konnektor MUSS IP Version 4 (IPv4) für alle seine IP-Schnittstellen unterstützen.
Die Hardware des Konnektors MUSS für den Einsatz von IPv4 und IPv6 im Dual-Stack-Mode geeignet sein.
Bis zu einer Migration von IPv4 auf IPv6 MUSS der Konnektor sämtliche empfangene IP-Pakete der Version 6 (IPv6) verwerfen.
[<=]

TIP1-A_4728-01 - IPv4 und IPv6 (Option IPv6)
Der Konnektor MUSS IP Version 4 (IPv4) und IP Version 6 (IPv6) für alle seine physikalischen Adapter unterstützen.
Die Hardware des Konnektors MUSS für den Einsatz von IPv4 und IPv6 im Dual-Stack-Mode geeignet sein.
[<=]

TIP1-A_4729 - Es darf kein dynamisches Routing verwendet werden
Dynamische Routing-Protokolle dürfen vom Konnektor nicht eingesetzt werden. Wird in einem der an den Konnektor angeschlossenen Netzwerke ein dynamisches Routing genutzt, so DÜRFEN Routing Updates vom Konnektor NICHT akzeptiert werden und keine Routen eingetragen werden.
[<=]

TIP1-A_5152 - Aktualisieren der Infrastrukturinformationen aus der TI
Falls Parameter MGM_LU_ONLINE=Enabled, MUSS der Konnektor einmal täglich TUC_KON_283 „Infrastruktur Konfiguration aktualisieren“ aufrufen.
[<=]

4.2.1.1.1 Netzwerksegmentierung

In Anlehnung an die in der [gemSpec_Net#2.3.3] definierten Netzwerksegmente werden in der Konnektorspezifikation die folgenden Bezeichner verwendet:

Tabelle 306: TAB_KON_680 Mapping der Netzwerksegmente

ReferenzID im Konnektor	Adressbereich für die TI-Produktivumgebung	Adressbereich für die TI-Testumgebung	Adressbereich für die TI-Referenzumgebung
NET_SIS	TI_Dezentral_SIS - Konnektoren	TI_Test_Dezentral_SIS - Konnektoren	Ist durch den Testbetriebsverantwortlichen zu definieren.
NET_TI_ DEZENTRAL	TI_Dezentral - Konnektoren	TI_Test_Dezentral - Konnektoren	Ist durch den Testbetriebsverantwortlichen zu definieren.
NET_TI_ ZENTRAL	TI_Zentral - Zentrale Dienste	TI_Test_Zentral - Zentrale Dienste	Ist durch den Testbetriebsverantwortlichen zu definieren.

NET_TI_ OFFENE_FD	Anwendungsdienste - Offene Fachdienste - WANDA Smart	Test_Anwendungsdienste - Offene Fachdienste - WANDA Smart	Ist durch den Testbetriebsverantwortlichen zu definieren.
NET_TI_ GESICHERTE_ FD	Anwendungsdienste - Gesicherte Fachdienste	Test_Anwendungsdienste - Gesicherte Fachdienste	Ist durch den Testbetriebsverantwortlichen zu definieren.
NET_LEKTR	Liste der Netzwerke die in der Einsatzumgebung über den Konnektor erreichbar sind. Ein Eintrag der Liste enthält die Netzwerkadresse und den Netzwerkprefix.		
ANLW_ BESTANDS NETZE	Liste der an die TI angeschlossenen Bestandsnetze (u. a. das Sichere Netz der KVen). Ein Eintrag der Liste enthält die Netzwerkadresse und den Netzwerkprefix.		
ANLW_ AKTIVE_ BESTANDS NETZE	Liste der an die TI angeschlossenen und aktivierten Bestandsnetze		

Tabelle 307: TAB_KON_681 Definition der vom Konnektor verwendeten VPN-Tunnel

ReferenzID	Bedeutung/Belegung
VPN_TI	Logischer Adapter des VPN-Tunnel zur TI mit dessen VPN_TUNNEL_TI_INNER_IP aus dem Adresssegment NET_TI_DEZENTRAL
VPN_SIS	Logischer Adapter des VPN-Tunnel zur SIS mit dessen VPN_TUNNEL_SIS_INNER_IP aus dem Addresssegment NET_SIS

Tabelle 308: TAB_KON_682 Definition der Konnektor IP-Adressen

ReferenzID	Bedeutung/Belegung
ANLW_LAN_IP_ADDRESS	Dies ist die IP-Adresse des LAN-Adapters. Aus dem Netz der Einsatzumgebung (ANLW_LAN_NETWORK_SEGMENT) die vom Konnektor verwendete IP-Adresse. Unter dieser Adresse werden die Dienste des Konnektor im lokalen Netzwerk bereitgestellt. Diese Adresse entspricht dem in Tabelle TAB_KON_683 LAN-Adapter IP-

	Konfiguration definierten Parameter ANLW_LAN_IP_ADDRESS.
ANLW_WAN_IP_ADDRESS	Dies ist die IP-Adresse des WAN-Adapters.

4.2.1.1.2 Routing und Firewall

Darstellung der Kommunikationsregeln des Konnektors

Diese Abbildung dient der Veranschaulichung der im Konnektor verwendeten Kommunikationsregeln welche in den nachfolgenden Afo definiert werden.

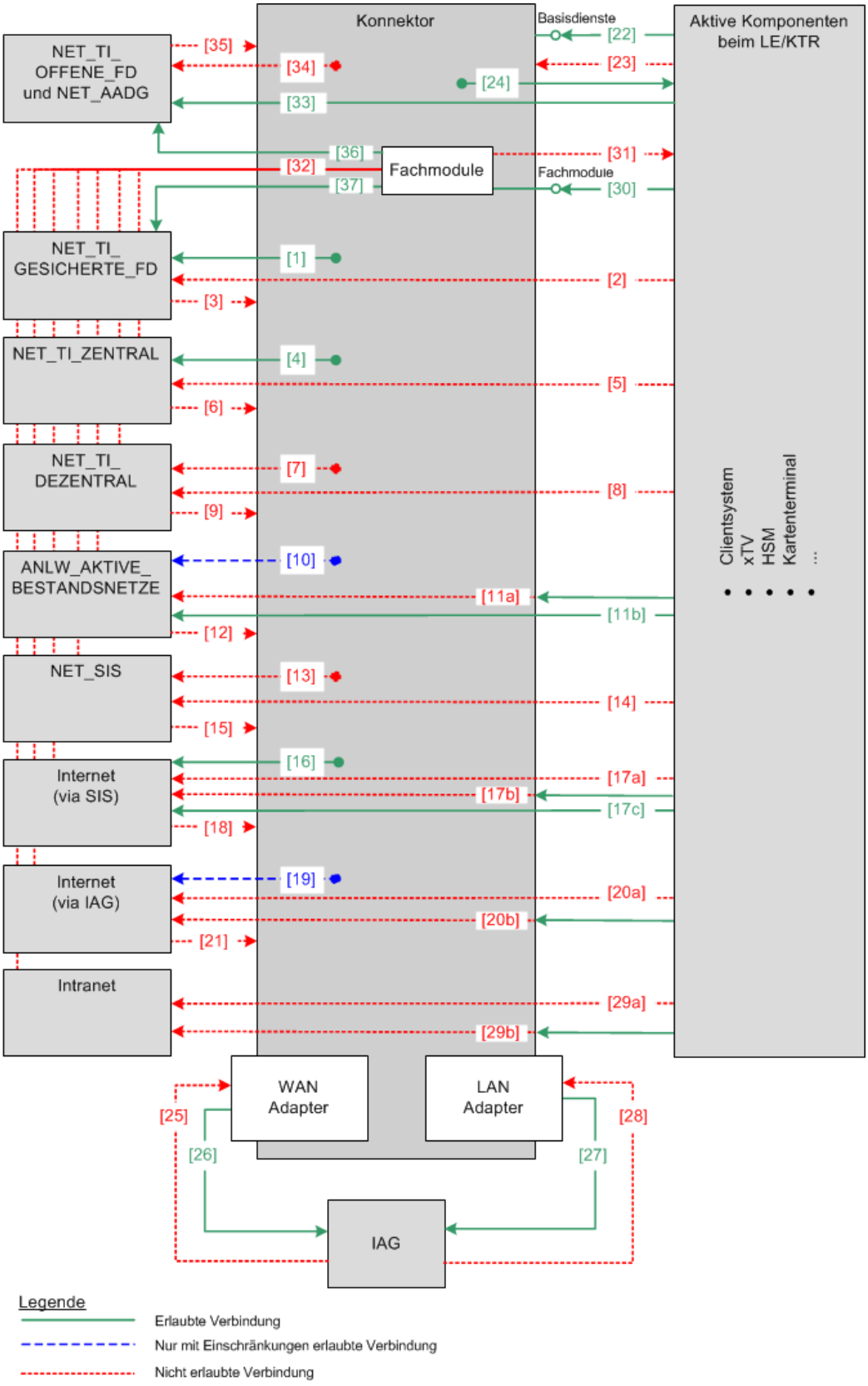


Abbildung 21: PIC_KON_115 Kommunikationsregeln Konnektor

TIP1-A_4730 - Kommunikation mit NET_TI_GESICHERTE_FD
Der Konnektor MUSS sicherstellen, dass IP-Pakete mit einer Absenderadresse aus dem Adressbereich NET_TI_GESICHERTE_FD verworfen werden, wenn sie nicht aus dem VPN-Tunnel der TI (VPN_TI) stammen.
Der Konnektor MUSS die Kommunikation mit Systemen des Netzwerksegments NET_TI_GESICHERTE_FD für folgende Fälle unterstützen:

- [1] vom Konnektor kommend
- [37] wenn (MGM_LU_ONLINE=Enabled) vom Fachmodul kommend

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit Systemen des Netzwerksegments NET_TI_GESICHERTE_FD für folgende Fälle blockieren:

- [2] von „Aktive Komponenten" kommend
- [3] in Richtung Konnektor gehend

Der Konnektor MUSS sicherstellen, dass die aus einer unterstützten Kommunikation mit Systemen aus dem Netzwerksegment NET_TI_GESICHERTE_FD bestimmten IP-Pakete ausschließlich in den VPN-Tunnel der TI (VPN_TI) geleitet werden.
[<=]

TIP1-A_5530 - Kommunikation mit NET_TI_OFFENE_FD
Der Konnektor MUSS sicherstellen, dass IP-Pakete mit einer Absenderadresse aus dem Adressbereich NET_TI_OFFENE_FD verworfen werden, wenn sie nicht aus dem VPN-Tunnel der TI (VPN_TI) stammen.
Der Konnektor MUSS die Kommunikation mit Systemen des Netzwerksegments NET_TI_OFFENE_FD für folgende Fälle unterstützen:

- [33] von „Aktive Komponenten" kommend
- [36] vom Fachmodul kommend

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit Systemen des Netzwerksegments NET_TI_OFFENE_FD für folgende Fälle blockieren:

- [34] vom Konnektor kommend
- [35] in Richtung Konnektor gehend

Der Konnektor MUSS sicherstellen, dass die aus einer unterstützten Kommunikation mit Systemen aus dem Netzwerksegment NET_TI_OFFENE_FD bestimmten IP-Pakete ausschließlich in den VPN-Tunnel der TI (VPN_TI) geleitet werden.
[<=]

TIP1-A_4731 - Kommunikation mit NET_TI_ZENTRAL
Der Konnektor MUSS sicherstellen, dass IP-Pakete mit einer Absenderadresse aus dem Adressbereich NET_TI_ZENTRAL verworfen werden, wenn sie nicht aus dem VPN-Tunnel der TI (VPN_TI) stammen.
Der Konnektor MUSS die Kommunikation mit Systemen des Netzwerksegments NET_TI_ZENTRAL für folgende Fälle unterstützen:

- [4] vom Konnektor kommend

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit Systemen des Netzwerksegments NET_TI_ZENTRAL für folgende Fälle blockieren:

- [5] von „Aktive Komponenten" kommend
- [6] in Richtung Konnektor gehend

Der Konnektor MUSS sicherstellen, dass die aus einer unterstützten Kommunikation mit Systemen aus dem Netzwerksegment NET_TI_ZENTRAL bestimmten IP-Pakete

ausschließlich in den VPN-Tunnel der TI (VPN_TI) geleitet werden.
[<=]

TIP1-A_4732 - Kommunikation mit NET_TI_DEZENTRAL
Der Konnektor MUSS sicherstellen, dass die Adressen aus dem Adressbereich NET_TI_DEZENTRAL nur für die Kommunikation mit der TI/den weiteren Anwendungen des Gesundheitswesens in Form der inner IP (VPN_TUNNEL_TI_INNER_IP) des VPN-Tunnel der TI (VPN_TI) verwendet wird.
Der Konnektor MUSS die Kommunikation mit Systemen des Netzwerksegments NET_TI_DEZENTRAL für folgende Fälle unterstützen:

- keine

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit Systemen des Netzwerksegments NET_TI_DEZENTRAL für folgende Fälle blockieren:

- [7] vom Konnektor kommend (zur Verhinderung des Zugriffs auf fremde Konnektoren)
- [8] von „Aktive Komponenten"
- [9] in Richtung Konnektor gehend

[<=]

TIP1-A_4733 - Kommunikation mit ANLW_AKTIVE_BESTANDSNETZE
Der Konnektor MUSS sicherstellen, dass IP-Pakete mit einer Absenderadresse aus dem Adressbereich ANLW_AKTIVE_BESTANDSNETZE verworfen werden, wenn sie nicht aus dem VPN-Tunnel der TI (VPN_TI) stammen.
Der Konnektor MUSS die Kommunikation mit Systemen des Netzwerksegments ANLW_AKTIVE_BESTANDSNETZE für folgende Fälle unterstützen:

- [10] wenn (MGM_LU_ONLINE=Enabled) vom Konnektor kommend nur für die DNS-Namensauflösung mittels DNS_SERVERS_BESTANDSNETZE
- [11b] wenn (MGM_LU_ONLINE=Enabled) von „Aktive Komponenten" kommend

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit Systemen des Netzwerksegments ANLW_AKTIVE_BESTANDSNETZE für folgende Fälle blockieren:

- [11a] für nicht freigegebene angeschlossene Netze des Gesundheitswesens mit WANDA Basic (ANLW_BESTANDSNETZE abzüglich ANLW_AKTIVE_BESTANDSNETZE) von „Aktive Komponenten" kommend;
- [12] in Richtung Konnektor gehend (und den dahinterliegenden „Aktive Komponenten")

Der Konnektor MUSS sicherstellen, dass die aus einer unterstützten Kommunikation mit Systemen aus dem Netzwerksegment ANLW_AKTIVE_BESTANDSNETZE bestimmten IP-Pakete ausschließlich in den VPN-Tunnel der TI (VPN_TI) geleitet werden.
[<=]

TIP1-A_4734 - Kommunikation mit NET_SIS
Der Konnektor MUSS sicherstellen, dass eine Adresse aus dem Adressbereich NET_SIS nur für die Kommunikation mit dem Internet (via SIS) in Form der inner IP (VPN_TUNNEL_SIS_INNER_IP) des VPN-Tunnel der SIS (VPN_SIS) verwendet wird.
Der Konnektor MUSS insbesondere die Kommunikation mit Systemen des Netzwerksegments NET_SIS für folgende Fälle unterstützen:

- keine

Der Konnektor MUSS die Kommunikation an seinen Außenschnittstellen mit NET_SIS für folgende Fälle blockieren:

- [13] vom Konnektor kommend
- [14] von „Aktive Komponenten" kommend
- [15] in Richtung Konnektor gehend (und den dahinterliegenden „Aktiven Komponenten")

[<=]

TIP1-A_4735 - Kommunikation mit dem Internet (via SIS)
Der Konnektor MUSS sicherstellen, dass IP-Pakete mit einer Absenderadresse aus dem Adressbereich NET_TI_ZENTRAL, NET_TI_GESICHERTE_FD; NET_TI_OFFENE_FD, NET_TI_DEZENTRAL, ANLW_AKTIVE_BESTANDSNETZE, ANLW_LAN_ADDRESS_SEGMENT, aus einem der Netzwerksegmente in ANLW_LEKTR_INTRANET_ROUTES oder ANLW_WAN_NETWORK_SEGMENT verworfen werden, wenn sie aus dem VPN-Tunnel der SIS (VPN_SIS) stammen.
Der Konnektor MUSS die Kommunikation mit Systemen des Netzwerksegments Internet (via SIS) für folgende Fälle unterstützen:

- [16] wenn (MGM_LU_ONLINE=Enabled und ANLW_INTERNET_MODUS=SIS) vom Konnektor kommend
- [17c] wenn (MGM_LU_ONLINE=Enabled und ANLW_INTERNET_MODUS=SIS) von „Aktive Komponenten" kommend

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit Internet (via SIS) für folgende Fälle blockieren oder umleiten:

- [17a] blockieren, wenn (MGM_LU_ONLINE=Enabled und ANLW_INTERNET_MODUS=KEINER) von „Aktive Komponenten" kommend
- [17b] umleiten, wenn (MGM_LU_ONLINE=Enabled und ANLW_INTERNET_MODUS=IAG) von „Aktive Komponenten" kommend;
  ➔ Der Konnektor MUSS an Hosts im Internet gerichtete IP-Pakete gemäß [RFC792] umleiten (ICMP Redirect).
- [18] blockieren, wenn von SIS kommend in Richtung Konnektor (und die dahinterliegenden „Aktive Komponenten")

  Der Konnektor MUSS sicherstellen, dass die für die Kommunikation mit dem Internet (via SIS) bestimmten IP-Pakete ausschließlich in den VPN-Tunnel des SIS (VPN_SIS) geleitet werden.

[<=]

TIP1-A_4736-02 - Kommunikation mit dem Internet (via IAG)
Der Konnektor MUSS sicherstellen, dass eingehende IP-Pakete von der Kommunikation mit dem Internet mit der Empfängeradresse ungleich (ANLW_LAN_IP_ADDRESS oder aus einem der Netzwerksegmente in ANLW_LEKTR_INTRANET_ROUTES wenn ANLW_WAN_ADAPTER_MODUS=DISABLED) oder (ANLW_WAN_IP_ADDRESS wenn ANLW_WAN_ADAPTER_MODUS=ENABLED) verworfen werden.
Der Konnektor MUSS sicherstellen, dass ausgehende IP-Pakete für die Kommunikation mit dem Internet mit der Absenderadresse ungleich (ANLW_LAN_IP_ADDRESS oder aus einem der Netzwerksegmente in ANLW_LEKTR_INTRANET_ROUTES wenn ANLW_WAN_ADAPTER_MODUS=DISABLED) oder (ANLW_WAN_IP_ADDRESS wenn ANLW_WAN_ADAPTER_MODUS=ENABLED) verworfen werden.
Der Konnektor MUSS die Kommunikation mit Systemen des Netzwerksegments Internet (via IAG) für folgende Fälle unterstützen:

- [19] vom Konnektor kommend zu den folgenden Systemen für das Protokoll IPsec
  - VPN_KONZENTRATOR_TI_IP_ADDRESS
  - VPN_KONZENTRATOR_SIS_IP_ADDRESS
- [19] vom Konnektor kommend zu den folgenden Systemen für HTTP und HTTPS
  - CERT_CRL_DOWNLOAD_ADDRESS
  - TSL-Download-Punkt des TSL-Dienstes
  - hash&URL-Server
  - Registrierungsserver
  - Remote-Managementserver
  - DNS_ROOT_ANCHOR_URL (benötigte IP-Adressen um den DNSSEC Trust Anchor im Namensraum Internet zu verifizieren)
- [19] vom Konnektor kommend zu den folgenden Systemen für das Protokoll DNS
  - beliebige Hosts

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit Internet (via IAG) für folgende Fälle blockieren oder umleiten:

- [20a] blockieren, wenn (ANLW_INTERNET_MODUS=KEINER oder MGM_LU_ONLINE=Disabled) von „Aktive Komponenten" kommend
- [20b] mittels ICMP Redirect gemäß [RFC792] zum Default Gateway umleiten, wenn die Zieladresse des IP-Pakets nicht innerhalb der Adressbereiche (NET_TI_ZENTRAL, NET_TI_OFFENE_FD, NET_TI_GESICHERTE_FD und ANLW_AKTIVE_BESTANDSNETZE) ist und ANLW_INTERNET_MODUS=IAG und von „Aktive Komponenten" kommend.
- [21] blockieren, wenn von IAG kommend in Richtung Konnektor (und die dahinterliegenden „Aktive Komponenten")

[<=]

TIP1-A_4737 - Kommunikation mit „Aktive Komponenten"
Der Konnektor MUSS sicherstellen, dass ausgehende IP-Pakete für die Kommunikation mit „Aktive Komponenten" mit einer Absenderadresse ungleich ANLW_LAN_IP_ADDRESS, einer Adresse aus einem Netzwerksegment in ANLW_LEKTR_INTRANET_ROUTES oder 0.0.0.0 verworfen werden.
Der Konnektor MUSS die Kommunikation mit „Aktive Komponenten" für folgende Fälle unterstützen:

- [22] auf den Konnektor (mittels der Schnittstelle Basisdienste)
- [24] vom Konnektor kommend

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit „Aktive Komponenten" für folgende Fälle blockieren:

- [23] zum Konnektor eingehend (direkt – ohne eine der Schnittstellen Fachmodule oder Basisdienste zu nutzen)

[<=]

TIP1-A_4738 - Route zum IAG
Der Konnektor MUSS die Kommunikation mit dem IAG der Einsatzumgebung für folgende Fälle unterstützen:

- [26] wenn (ANLW_WAN_ADAPTER_MODUS=ENABLED) vom WAN-Adapter kommend
- [27] wenn (ANLW_WAN_ADAPTER_MODUS=DISABLED) vom LAN-Adapter kommend

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit dem IAG der Einsatzumgebung für folgende Fälle blockieren:

- [25] wenn (ANLW_WAN_ADAPTER_MODUS=ENABLED) zum WAN-Adapter eingehend
- [28] wenn (ANLW_WAN_ADAPTER_MODUS=DISABLED) zum LAN-Adapter eingehend

[<=]

TIP1-A_4740 - Admin Defined Firewall Rules
Die Firewall des Konnektor MUSS alle vom Administrator in ANLW_FW_SIS_ADMIN_RULES definierten Firewall-Regeln als zusätzliche Einschränkung übernehmen.
[<=]

TIP1-A_4741 - Kommunikation mit dem Intranet
Der Konnektor MUSS die Kommunikation mit Systemen aus einem Intranet-VPN (einem der Netzwerksegmente ANLW_LEKTR_INTRANET_ROUTES) für folgende Fälle unterstützen:

- [22] wenn von Aktive Komponenten aus dem Netzwerksegment ANLW_LEKTR_INTRANET_ROUTES kommend zum Konnektor mittels der Schnittstelle Basisdienste
- [24] wenn vom Konnektor kommend zu ANLW_LEKTR_INTRANET_ROUTES
- Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit einem der Intranet Netzwerksegmente für folgende Fälle blockieren bzw. umleiten:
- [29a] blockieren, wenn (ANLW_INTRANET_ROUTES_MODUS=BLOCK) vom „Aktive Komponenten" kommend;
- [29b] umleiten, wenn (ANLW_INTRANET_ROUTES_MODUS=REDIRECT) vom „Aktive Komponenten" kommend;
  ➔ Der Konnektor MUSS an ANLW_LEKTR_INTRANET_ROUTES gerichtete IP-Pakete gemäß [RFC792] umleiten (ICMP Redirect).

[<=]

TIP1-A_4742 - Kommunikation mit den Fachmodulen
Der Konnektor MUSS die Kommunikation mit den Fachmodulen für folgende Fälle unterstützen:

- [30] von „Aktive Komponenten" über Schnittstelle Fachmodule

Der Konnektor MUSS insbesondere die Kommunikation an seinen Außenschnittstellen mit den Fachmodulen für folgende Fälle blockieren:

- [31] zu „Aktive Komponenten"

- [32] zu den Netzwerksegmenten, NET_TI_ZENTRAL, NET_TI_DEZENTRAL, ANLW_AKTIVE_BESTANDSNETZE, Internet (via SIS), Internet (via IAG) und Intranet

[<=]

TIP1-A_4744 - Firewall - Drop statt Reject
Die Firewall des Konnektor MUSS alle abgelehnten IP-Pakete verwerfen (DROP) ohne ein ICMP-Destination-Unreachable (Type 3) zu schicken.
[<=]

TIP1-A_4746 - Firewall – Abwehr von IP-Spoofing, DoS/DDoS-Angriffe und Martian Packets
Der Konnektor MUSS geeignete technische Funktionen zur Abwehr von IP-Spoofing und DoS/DDoS-Angriffen implementieren.
Der Konnektor MUSS Martian Packets (Absender– oder Empfängeradressen aus den von der IETF als Special-Purpose definierten Netzbereichen), mindestens jedoch aus folgenden Netzbereichen 0.0.0.0/8, 127.0.0.0/8, 169.254.0.0/16, 192.0.0.0/24, 192.0.2.0/24, 198.18.0.0/15, 198.51.100.0/24, 203.0.113.0/24, 224.0.0.0/4, 240.0.0.0/4 verwerfen. Die in [RFC1918] und [RFC 6598] definierten Netzbereiche sind hiervon ausgenommen.
[<=]

TIP1-A_4745 - Eingeschränkte Nutzung von „Ping“
Die Firewall des Konnektor MUSS TCP-Port-7(Echo)-Pakete verwerfen.
Die Firewall des Konnektor MUSS ICMP-Echo-Request (Typ 8) und ICMP-Echo-Response (Typ 0) ausschließlich für die folgenden Kommunikationen zulassen:

- vom Konnektor zu den VPN-Konzentratoren für SIS und TI über das Transportnetz (via IAG)
- vom Konnektor zu dem CRL-Webservern (im Transportnetz) über das Internet (via SIS) und das Transportnetz (via IAG)
- vom Konnektor zu dem IAG der Einsatzumgebung
- vom Konnektor zu NET_TI_ZENTRAL
- vom Konnektor zu NET_TI_GESICHERTE_FD
- vom Konnektor zu NET_TI_OFFENE_FD
- vom Konnektor zum lokalen Netzwerk (Adressen aus ANLW_LAN_NETWORK_SEGMENT oder Adressen aus einem der Netzwerksegmente in ANLW_LEKTR_INTRANET_ROUTES)
- vom lokalen Netzwerk (Adressen aus ANLW_LAN_NETWORK_SEGMENT (jedoch ohne die ANLW_LAN_IP_ADDRESS) oder Adressen aus einem der Netzwerksegmente in ANLW_LEKTR_INTRANET_ROUTES) zum Konnektor
- vom lokalen Netzwerk in ANLW_AKTIVE_BESTANDSNETZE (die freigegebenen angeschlossenen Netze des Gesundheitswesens mit WANDA Basic)
- vom lokalen Netzwerk in das Internet (via SIS)

  Die Firewall des Konnektors MUSS für alle anderen Kommunikationen ein ICMP-Echo-Request (Typ 8) verwerfen.

[<=]

TIP1-A_4747 - Firewall – Einschränkungen der IP-Protokolle
Der Konnektor MUSS alle IP-Protokolle außer 1 (ICMP), 4 (IP in IP (encapsulation)), 17 (UDP), 6 (TCP), 50 (ESP) und 108 (IPComp) für alle ein- oder ausgehenden Pakete an

allen seinen Adaptern verwerfen.
[<=]

TIP1-A_4748 - Firewall – Routing-Regeln
Der Konnektor DARF seine Routing-Regeln NICHT durch IP-Kommunikation beeinflussen lassen, weder mittels eines Routing-Protokolls (wie BGP oder RIP) noch mittels ICMP-Kommandos (wie Redirect (5), Router Advertisement (9/10) oder auch Mobile Host Redirect (32)) sondern MUSS diese ausschließlich durch TUC_KON_304 „Netzwerk-Routen einrichten" setzen.
Die Firewall des Konnektor MUSS alle aus einem der Tunnel (VPN_TI oder VPN_SIS) kommenden DHCP-Pakete verwerfen.
Die Firewall des Konnektors MUSS an den Konnektor gerichtete IPsec-Pakete (IKE, ESP und IPsec NAT-T) verwerfen, sofern sie nicht einer vom Konnektor initiierten IPsec-Verbindung (VPN_TI und VPN_SIS) zugeordnet werden können.
[<=]

TIP1-A_4749 - Firewall Restart
Der Konnektor MUSS gewährleisten, dass unmittelbar nach einer Änderung der Parameter eines Adapters (LAN-Adapter, WAN-Adapter, virtueller Adapter VPN_TI oder virtueller Adapter VPN_SIS) die Firewall des Konnektor neu erstellt und geladen wird.
Wenn der WAN-Adapter verwendet wird (ANLW_WAN_ADAPTER_MODUS=ENABLED) DARF die Firewall des Konnektor bei einer Änderung der ANLW_WAN_IP_ADDRESS NICHT die Verbindungen über den LAN-Adapter durch einen Restart der Firewall beeinflussen.
Wenn der WAN-Adapter verwendet wird (ANLW_WAN_ADAPTER_MODUS=ENABLED), DARF die Firewall des Konnektor bei einer Änderung der ANLW_LAN_IP_ADDRESS NICHT die Verbindungen über die Adapter WAN, VPN_TI oder VPN_SIS durch einen Restart der Firewall beeinflussen.
[<=]

Umsetzungshinweis für den Hersteller: Es können zwei getrennten Firewall-Regelsets für den LAN- bzw. für den WAN-Adapter verwendet werden.

TIP1-A_4750 - Firewall-Protokollierung
Der Konnektor MUSS bei Start und Stopp der Firewall einen Protokolleintrag mit der Schwere „Warning" und dem Typ „Operations" sowie mindestens folgenden Informationen generieren:

- Zeitstempel, Aktion (Start/Stop), Ergebnis (Erfolg/Fehler), Auslöser (Prozess/User)

Der Konnektor MUSS bei Konfigurationsänderungen der Firewall einen Protokolleintrag mit der Schwere „Warning" und dem Typ „Operations" sowie mindestens folgenden Informationen generieren:

- Zeitstempel, Aktion (Add/Delete/Change), Details (Beschreibung der Änderung), Auslöser (Prozess/User)

Der Konnektor MUSS für alle vom Konnektor ausgehenden, nicht zugelassenen Kommunikationsversuche einen Protokolleintrag mit der Schwere „Warning" und dem Typ „Security" sowie mindestens folgenden Informationen generieren:

- Zeitstempel, Aktion (Drop, Reject), Absender-IP-Adresse, Empfänger-IP-Adresse, Protokoll, Absender-Port und Empfänger-Port, Interface über das das Paket empfangen wurde

Der Konnektor MUSS für alle verworfenen IP-Spoofing- und Martian-Packets einen Protokolleintrag mit der Schwere „Warning" und dem Typ „Security" sowie mindestens folgenden Informationen generieren:

- Zeitstempel, Aktion (Drop, Reject), Absender-IP-Adresse, Empfänger-IP-Adresse, Protokoll, Absender-Port und Empfänger-Port, Interface über das das Paket empfangen wurde

Der Konnektor MUSS für alle von der Firewall verworfenen IP-Pakete einen Protokolleintrag mit der Schwere „Info" und dem Typ „Security" sowie mindestens folgenden Informationen generieren, wobei Layer 3 Broadcasts von der Protokollierung ausgenommen werden können:

- Zeitstempel, Aktion (Drop, Reject), Absender-IP-Adresse, Empfänger-IP-Adresse, Protokoll, Absender-Port und Empfänger-Port, Interface über das das Paket empfangen wurde

Der Konnektor MUSS für die Firewall-Protokollierung den TUC_KON_271 „Schreibe Protokolleintrag" nutzen.
[<=]

4.2.1.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_4751 - Reagiere auf LAN_IP_Changed
Beim Auftreten eines der nachfolgenden Events MUSS der Konnektor den TUC_KON_305 „LAN-Adapter initialisieren" starten.

- Event ANLW/LAN/IP_CHANGED
- Event DHCP/LAN_CLIENT/RENEW

[<=]

TIP1-A_4752 - Reagiere auf WAN_IP_Changed
Beim Auftreten eines der nachfolgenden Events MUSS der Konnektor TUC_KON_306 „WAN-Adapter initialisieren" starten.

- Event ANLW/WAN/IP_CHANGED
- Event DHCP/WAN_CLIENT/RENEW

[<=]

TIP1-A_4753 - Ereignisbasiert Netzwerkrouten einrichten
Beim Auftreten eines der nachfolgenden Events MUSS der Konnektor den TUC_KON_304 „Netzwerk-Routen einrichten" aufrufen.

- Event NETWORK/VPN_TI/UP
- Event NETWORK/VPN_TI/DOWN
- Event NETWORK/VPN_SIS/UP
- Event NETWORK/VPN_SIS/DOWN
- Event MGM/LU_CHANGED/LU_ONLINE

[<=]

4.2.1.3 Interne TUCs, nicht durch Fachmodule nutzbar

4.2.1.3.1 TUC_KON_305 „LAN-Adapter initialisieren"

TIP1-A_4754 - TUC_KON_305 „LAN-Adapter initialisieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_305 „LAN-Adapter initialisieren" umsetzen.

Tabelle 309: TAB_KON_614 - TUC_KON_305 „LAN-Adapter initialisieren"

Element	Beschreibung
Name	TUC_KON_305 LAN-Adapter initialisieren
Beschreibung	Initialisieren der LAN-Netzwerkschnittstelle
Auslöser	• Event ANLW/LAN/IP_CHANGED • Event DHCP/LAN_CLIENT/RENEW; BOOTUP
Vorbedingungen	• Wenn die IP-Konfiguration des LAN-Adapters statisch (DHCP_CLIENT_LAN_STATE=Disabled) gesetzt wird, MUSS der Konnektor gewährleisten, dass alle Konfigurationsparameter gemäß „Tabelle TAB_KON_683 LAN-Adapter IP-Konfiguration„ vorab über die Managementschnittstelle gesetzt wurden. • Wenn die IP-Konfiguration des LAN-Adapters dynamisch per DHCP (DHCP_CLIENT_LAN_STATE=Enabled) gesetzt wird, MUSS der DHCP-Client diese vorab gesetzt haben.
Eingangsdaten	Keine
Komponenten	Konnektor
Ausgangsdaten	Keine
Standardablauf	1) Die in „Tabelle TAB_KON_683 LAN-Adapter IP-Konfiguration„ und „Tabelle TAB_KON_684 LAN-Adapter Erweiterte Parameter „ gesetzten Werte sind zur Konfiguration des LAN-Adapter zu verwenden. 2) Rufe TUC_KON_304 „Netzwerk-Routen einrichten" 3) Wenn (ANLW_WAN_ADAPTER_MODUS = DISABLED) und MGM_LU_ONLINE = ENABLED: • Rufe TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen". • Rufe TUC_KON_322 „Verbindung zu dem VPN-Konzentrator des SIS aufbauen"

	4) Firewall-Regeln aktualisieren und aktivieren. Tritt der Fehler 4164 auf, geht der Konnektor in den Betriebszustand EC_Firewall_Not_Reliable über.
Varianten/ Alternativen	Keine
Fehlerfälle	(→ 1) Fehlerhafte LAN IP-Konfiguration; 4162 (→ 4) Beim Aktualisieren oder Aktivieren der Firewall-Regeln ist es zu einem Fehler gekommen; 4164
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 310: TAB_KON_615 Fehlercodes TUC_KON_305 „LAN-Adapter initialisieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4162	Technical	Error	Es liegt eine fehlerhafte LAN IP-Konfiguration vor.
4164	Technical	Fatal	Beim Aktualisieren oder Aktivieren der Firewall-Regeln ist es zu einem Fehler gekommen.

[<=]

4.2.1.3.2 TUC_KON_306 „WAN-Adapter initialisieren"

TIP1-A_4755 - TUC_KON_306 „WAN-Adapter initialisieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_306 „WAN-Adapter initialisieren" umsetzen.

Tabelle 311: TAB_KON_616 - TUC_KON_306 „WAN-Adapter initialisieren"

Element	Beschreibung
Name	TUC_KON_306 WAN-Adapter initialisieren

Beschreibung	Initialisieren der WAN-Netzwerkschnittstelle
Auslöser	• Event ANLW/WAN/IP_CHANGED • Event DHCP/WAN_CLIENT/RENEW; BOOTUP
Vorbedingungen	
Eingangsdaten	Keine
Komponenten	Konnektor
Ausgangsdaten	Keine
Standardablauf	1) Wenn ANLW_WAN_ADAPTER_MODUS = DISABLED oder MGM_LU_ONLINE = Disabled: a) Aktive VPN-Tunnel TI oder SIS (VPN_TI oder VPN_SIS) müssen gestoppt werden, 2) Wenn ANLW_WAN_ADAPTER_MODUS = ENABLED und MGM_LU_ONLINE = ENABLED: a) Der WAN-Adapter wird abhängig von DHCP_CLIENT_WAN_STATE statisch oder dynamisch über DHCP konfiguriert. Die in „Tabelle TAB_KON_685 WAN-Adapter IP-Konfiguration„ und „Tabelle TAB_KON_686 WAN-Adapter Erweiterte Parameter„ gesetzten Werte sind zur Konfiguration des WAN-Adapter zu verwenden. b) Rufe TUC_KON_304 „Netzwerk-Routen einrichten“ c) Rufe TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen“. d) Rufe TUC_KON_322 „Verbindung zu dem VPN-Konzentrator des SIS aufbauen“ e) Firewall-Regeln aktualisieren und aktivieren. Tritt der Fehler 4164 auf, geht der Konnektor in den Betriebszustand EC_Firewall_Not_Reliable über.
Varianten/ Alternativen	Keine
Fehlerfälle	(→ 1) Fehlerhafte WAN IP-Konfiguration; 4163 (→ 2) Beim Aktualisieren oder Aktivieren der Firewall-Regeln ist es zu einem Fehler gekommen; 4164
Nichtfunktionale Anforderungen	Keine

Tabelle 312: TAB_KON_617 Fehlercodes TUC_KON_306 „WAN-Adapter initialisieren“

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4163	Technical	Error	Es liegt eine fehlerhafte WAN-IP-Konfiguration vor.
4164	Technical	Fatal	Beim Aktualisieren oder Aktivieren der Firewall-Regeln ist es zu einem Fehler gekommen.

[<=]

4.2.1.3.3 TUC_KON_304 „Netzwerk-Routen einrichten"

TIP1-A_4758 - TUC_KON_304 „Netzwerk-Routen einrichten"
Der Konnektor MUSS den technischen Use Case TUC_KON_304 „Netzwerk-Routen einrichten" umsetzen.

Tabelle 313: TAB_KON_622 - TUC_KON_304 „Netzwerk-Routen einrichten"

Element	Beschreibung
Name	TUC_KON_304 Netzwerk-Routen einrichten
Beschreibung	Anpassen der Routing-Tabelle
Auslöser	• TUC_KON_305 „LAN-Adapter initialisieren" • TUC_KON_306 „WAN-Adapter initialisieren" • Event NETWORK/VPN_TI/UP • Event NETWORK/VPN_TI/DOWN • Event NETWORK/VPN_SIS/UP • Event NETWORK/VPN_SIS/DOWN • Event MGM/LU_CHANGED/LU_ONLINE
Vorbedingungen	keine
Eingangsdaten	• IP-Konfiguration des LAN-Interface (gemäß Tabelle TAB_KON_683 LAN-Adapter IP-Konfiguration) • IP-Konfiguration des WAN-Interface (gemäß Tabelle TAB_KON_685 WAN-Adapter IP-Konfiguration) • ANLW_IAG_ADDRESS (IP-Adresse des IAG der Einsatzumgebung) • DNS_SERVERS_INT
Komponenten	Konnektor
Ausgangsdaten	Keine
Nachbedingungen	• Die Routing-Einträge im Konnektor wurden gesetzt.
Standardablau	Alle bestehenden Routen MÜSSEN vollständig durch die in diesem TUC ermittelten Routen ersetzt werden.

	1) Wenn (MGM_LU_ONLINE=Enabled) Der Konnektor MUSS die nachfolgenden Routen bereitstellen a) i. Ziel: Lokale Netze der Einsatzumgebung gemäß ANLW_LEKTR_INTRANET_ROUTES Next Hop: gemäß ANLW_LEKTR_INTRANET_ROUTES b) Wenn die VPN-Tunnel zur TI und zum SIS nicht aufgebaut sind: i. Ziel: Default Route Next Hop: ANLW_IAG_ADDRESSc) Wenn der VPN-Tunnel zur TI aufgebaut und der VPN-Tunnel zum SIS nicht aufgebaut sind: i. Ziel: Default Route Next Hop: ANLW_IAG_ADDRESS ii. Ziel: TI (NET_TI_OFFENE_FD, NET_TI_GESICHERTE_FD und NET_TI_ZENTRAL) Next Hop: Innere Tunnel IP-Adresse des VPN-Konzentrators TI iii. Ziel: ANLW_AKTIVE_BESTANDSNETZE Next Hop: Innere Tunnel IP-Adresse des VPN-Konzentrators TI iv. Ziel: VPN-Konzentrator TI Next Hop: ANLW_IAG_ADDRESS d) Wenn die VPN-Tunnel zur TI und zum SIS aufgebaut sind: i. Ziel: Default Route Next Hop: Innere Tunnel IP-Adresse des VPN-Konzentrators SIS ii. Ziel: TI (NET_TI_OFFENE_FD, NET_TI_GESICHERTE_FD und NET_TI_ZENTRAL) Next Hop: Innere Tunnel IP-Adresse des VPN-Konzentrators TI iii. Ziel: ANLW_AKTIVE_BESTANDSNETZE Next Hop: Innere Tunnel IP-Adresse des VPN-Konzentrators TI iv. Ziel: VPN-Konzentrator TI Next Hop: ANLW_IAG_ADDRESS v. Ziel: VPN-Konzentrator SIS Next Hop: ANLW_IAG_ADDRESSHinweis: Wenn der VPN-Tunnel zur TI nicht existiert, kann auch kein VPN-Tunnel zum SIS existieren, da die Default Route zum IAG zeigen muss, um einen VPN-Tunnel zur TI aufbauen zu können. **2) Wenn (MGM_LU_ONLINE=Disabled)** 1. Der Konnektor MUSS die nachfolgenden Routen bereitstellen. i. Ziel: Lokale Netze der Einsatzumgebung gemäß ANLW_LEKTR_INTRANET_ROUTES Next Hop: gemäß ANLW_LEKTR_INTRANET_ROUTES

	3) Firewall aktualisieren: Die Firewall des Konnektors MUSS die neu eingerichteten Routen berücksichtigen und seine Regeln entsprechend aktualisieren und aktivieren. Tritt der Fehler 4164 auf, geht der Konnektor in den Betriebszustand EC_Firewall_Not_Reliable über.
Varianten/Alternativen	Keine
Fehlerfälle	(→ 1-2) Eine oder mehrere Variablen enthalten eine ungültige oder keine IP; 4167 (→ 3) Beim Aktualisieren oder Aktivieren der Firewall-Regeln ist es zu einem Fehler gekommen; 4164
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 314: TAB_KON_623 Fehlercodes TUC_KON_304 „Netzwerk-Routen einrichten"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4167	Technical	Fatal	CreateRoutes: Ein oder mehrere Adressen sind ungültig.
4164	Technical	Fatal	Beim Aktualisieren oder Aktivieren der Firewall-Regeln ist es zu einem Fehler gekommen.

[<=]

4.2.1.4 Interne TUCs, auch durch Fachmodule nutzbar

Keine.

4.2.1.5 Operationen an der Außenschnittstelle

Keine

4.2.1.6 Betriebsaspekte

TIP1-A_5414 - Initialisierung „Anbindung LAN/WAN"
Der Konnektor MUSS in der Bootup-Phase zur Initialisierung des Funktionsmerkmals „Anbindung LAN/WAN":

- den LAN-Adapter initialisieren (TUC_KON_305)
- den WAN-Adapter initialisieren (TUC_KON_306)
- die Infrastrukturdaten vom KSR einlesen (TUC_KON_283)

[<=]

TIP1-A_4759 - Konfiguration LAN-Interface
Der Konnektor MUSS gewährleisten, dass die Konfiguration nur dann gespeichert wird, wenn alle Parameter der nachfolgenden Tabellen den dazugehörigen Bedingungen

entsprechen, sowie grundsätzlich zulässige Werte darstellen (gemäß RFCs).
Wenn die Konfiguration per Managementschnittstelle geändert wurde, MUSS das folgende Systemereignis ausgelöst werden:

```
TUC_KON_256 {
    topic = "ANLW/LAN/IP_CHANGED";
    eventType = Op;
    severity = Info;
    parameters = („IP=$dieNeueIP”);
    doDisp = false}
```

Wenn (DHCP_CLIENT_LAN_STATE=Disabled) gesetzt ist, MUSS der Administrator des Konnektor die Werte der folgenden Tabelle über die Managementschnittstelle setzen können.
Wenn (DHCP_CLIENT_LAN_STATE=Enabled) gesetzt ist, MUSS der Administrator des Konnektor die Werte der folgenden Tabelle angezeigt bekommen, kann diese jedoch nicht ändern.

Tabelle 315: TAB_KON_683 LAN-Adapter IP-Konfiguration

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
ANLW_LAN_IP_ ADDRESS	IP-Adresse	Dies ist die IP-Adresse des LAN-Adapters. Nur wenn DHCP_CLIENT_LAN_STATE=Disabled MUSS der Administrator die LAN-seitige IP-Adresse des Konnektors setzen können. Diese IP-Adresse MUSS innerhalb des ANLW_LAN_NETWORK_SEGMENT liegen.
ANLW_LAN_ SUBNETMASK	Subnetzmaske	Dies ist die zu ANLW_LAN_IP_ADDRESS gehörende Subnetzmaske. Der Administrator MUSS die Subnetzmaske setzen können. Der Konnektor MUSS gewährleisten das nur eine gültige Subnetzmaske gespeichert werden kann.
ANLW_LAN_ NETWORK_ SEGMENT	IP-Adresse / Subnetzmaske	ANLW_LAN_NETWORK_SEGMENT ist ein Zustandswert, der sich aus den Werten von ANLW_LAN_IP_ADDRESS und ANLW_LAN_SUBNETMASK ergibt. Der Wert bezeichnet ein lokales Netzwerk in der Umgebung des Anwenders an das der LAN-Adapter des Konnektors angeschlossen ist. Der Konnektor MUSS gewährleisten, das das Netzwerksegment NICHT mit einem der folgenden Netzwerksegmente überlappt: 1. NET_TI_DEZENTRAL 2. NET_TI_ZENTRAL 3. NET_TI_OFFENE_FD 4. NET_TI_GESICHERTE_FD

		5. NET_SIS 6. ANLW_BESTANDSNETZE 7. ANLW_AKTIVE_BESTANDSNETZE 8. ANLW_WAN_NETWORK_SEGMENT 9. ANLW_LEKTR_INTRANET_ROUTES

Der Administrator des Konnektor MUSS die Werte der folgenden Tabelle über die Managementschnittstelle setzen können.

Tabelle 316: TAB_KON_684 LAN-Adapter Erweiterte Parameter

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
ANLW_LAN_MTU	Nummer	Der Administrator MUSS Maximum Transmission Unit (MTU) setzen können. Der Konnektor MUSS sicherstellen, das der konfigurierte Wert in den Grenzen von 576 bis 9000 liegt. Default-Wert: 1400
ANLW_LAN_PARAMETER	Liste von IP, UDP und/oder TCP Parametern	Der Administrator SOLL weitere Konfigurationsparameter gemäß [gemSpec_Net#2.2.2.1,2.5] konfigurieren können.

[<=]

TIP1-A_4760 - Konfiguration WAN-Interface
Der Konnektor MUSS gewährleisten, dass die Konfiguration nur dann gespeichert wird, wenn alle Parameter der nachfolgenden Tabellen den dazugehörigen Bedingungen entsprechen.
Wenn die Konfiguration per Managementschnittstelle geändert wurde, MUSS das folgende Systemereignis ausgelöst werden:
TUC_KON_256 {
topic = "ANLW/WAN/IP_CHANGED";
eventType = Op;
severity = Info;
parameters = („IP=$dieNeueIP“);
doDisp = false}
Wenn (DHCP_CLIENT_WAN_STATE=Disabled) gesetzt ist, MUSS der Administrator des Konnektors die Werte der folgenden Tabelle über die Managementschnittstelle setzen können.
Wenn (DHCP_CLIENT_WAN_STATE=Enabled) gesetzt ist, MUSS der Administrator des Konnektors die Werte der folgenden Tabelle angezeigt bekommen, kann diese jedoch nicht ändern.

Tabelle 317: TAB_KON_685 WAN-Adapter IP-Konfiguration

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
ANLW_WAN_ IP_ADDRESS	IP-Adresse	Dies ist die IP-Adresse des WAN-Adapters. Nur wenn DHCP_CLIENT_WAN_STATE=Disabled

		und ANLW_WAN_ADAPTER_MODUS=ENABLED MUSS der Administrator die WAN-seitige IP-Adresse des Konnektors setzen können. Diese IP-Adresse MUSS innerhalb des ANLW_WAN_NETWORK_SEGMENT liegen.
ANLW_WAN_ SUBNETMASK	Subnetzmaske	Dies ist die zu ANLW_WAN_IP_ADDRESS gehörende Subnetzmaske. Der Administrator MUSS die Subnetzmaske setzen können. Der Konnektor MUSS gewährleisten, dass nur eine gültige Subnetzmaske gespeichert werden kann.
ANLW_WAN_ NETWORK_ SEGMENT	IP-Adresse / Subnetzmaske	ANLW_WAN_NETWORK_SEGMENT ist ein Zustandswert, der sich aus den Werten von ANLW_WAN_IP_ADDRESS und ANLW_WAN_SUBNETMASK ergibt. Der Wert bezeichnet ein lokales Netzwerk in der Umgebung des Anwenders an das der WAN-Adapter des Konnektors angeschlossen ist. Der Konnektor MUSS gewährleisten, dass das Netzwerksegment nicht mit einem der folgenden Netzwerksegmente überlappt: 1. NET_TI_DEZENTRAL 2. NET_TI_ZENTRAL 3. NET_TI_OFFENE_FD 4. NET_TI_GESICHERTE_FD 5. NET_SIS 6. ANLW_ BESTANDSNETZE 7. ANLW_AKTIVE_BESTANDSNETZE 8. ANLW_LAN_NETWORK_SEGMENT 9. ANLW_LEKTR_INTRANET_ROUTES

Der Administrator des Konnektor MUSS die Werte der folgenden Tabelle über die Managementschnittstelle setzen können.

Tabelle 318: TAB_KON_686 WAN-Adapter Erweiterte Parameter

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
ANLW_WAN_MTU	Nummer	Der Administrator MUSS Maximum Transmission Unit (MTU) setzen können. Der Konnektor MUSS sicherstellen, das der konfigurierte Wert in den Grenzen von 576 bis 9000 liegt. Default-Wert: 1400
ANLW_WAN_PARAMETER	Liste von IP, UDP und/oder	Der Administrator SOLL weitere Konfigurationsparameter gemäß

	TCP Parametern	[gemSpec_Net#2.2.2.1,2.5] konfigurieren können.

[<=]

TIP1-A_4761 - Konfiguration Anbindung LAN/WAN
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_624 – „Konfigurationsparameter der Anbindung LAN/WAN vorzunehmen.
Wenn (ANLW_INTRANET_ROUTES_MODUS = REDIRECT) gesetzt ist, MUSS der Konnektor jedes Paket aus einem konfigurierten Intranet mit einem ICMP-Redirect mit dem hinterlegten Next Hop beantworten und der Konnektor MUSS gewährleisten, dass keine IP-Pakete in eines oder mehrere der konfigurierten Intranet geroutet werden.
Wenn (ANLW_INTRANET_ROUTES_MODUS = BLOCK) gesetzt ist, MUSS der Konnektor alle IP-Pakete für ein Intranet (gemäß ANLW_LEKTR_INTRANET_ROUTES) ablehnen.

Tabelle 319: TAB_KON_624 – „Konfigurationsparameter der Anbindung LAN/WAN"

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
ANLW_ ANBINDUNGS_ MODUS	InReihe	Der Konnektor ist in Reihe zu dem IAG der Einsatzumgebung geschaltet. Wenn ANLW_WAN_ADAPTER_MODUS= ENABLED befindet sich der Konnektor in diesem Anbindungsmodus. Der Administrator MUSS diesen Wert einsehen und DARF ihn NICHT ändern können.
	Parallel	Der Konnektor ist parallel (zu allen bestehenden Systemen) ins Netzwerk der Einsatzumgebung angebunden. Wenn ANLW_WAN_ADAPTER_MODUS= DISABLED befindet sich der Konnektor in diesem Anbindungsmodus. Der Administrator MUSS diesen Wert einsehen und DARF ihn NICHT ändern können.
ANLW_ INTERNET_ MODUS	SIS	Der (am Konnektor LAN-seitig ankommende) Internet-Traffic wird per VPN an den SIS geschickt.
	IAG	Bei Anfragen ins Internet wird der Aufrufer per ICMP-Redirect (Type 5) auf die Route zum IAG verwiesen. Wenn (ANLW_ANBINDUNGS_MODUS = InReihe) DARF dieser Wert NICHT auswählbar sein - statt dessen MUSS dann der Wert SIS verwendet werden.
	KEINER	Es wird kein Traffic ins Internet geroutet

ANLW_ INTRANET_ ROUTES_ MODUS	REDIRECT	Der Konnektor MUSS sicherstellen, dass dieser Wert nur gesetzt werden kann, wenn der Administrator zuvor ein oder mehrere Intranet (ANLW_LEKTR_INTRANET_ROUTES) definiert hat.
	BLOCK	Der Konnektor MUSS alle IP-Pakete für ein Intranet (gemäß ANLW_LEKTR_INTRANET_ROUTES) ablehnen.
ANLW_WAN_ ADAPTER_ MODUS	ENABLED	Dieser Parameter ändert den Interface-Status des WAN-Adapters. Der Administrator MUSS diesen Wert einsehen können. Der Administrator MUSS diesen Wert ändern können.
	DISABLED	Dieser Parameter ändert den Interface-Status des WAN-Adapters. Der Administrator MUSS diesen Wert einsehen können. Der Administrator MUSS diesen Wert ändern können.
ANLW_ LEKTR_ INTRANET_ ROUTES	Tupel aus Netzwerksegment und dazugehörigem Next-Hop	Der Administrator MUSS in diese Liste Einträge hinzufügen, editieren und löschen können. Liste von Routen zur Erreichung der Clientsysteme und Kartenterminals vom Konnektor; jeweils mit IP-Netzwerk dazugehörigem Next Hop. Die Netzwerksegmente DÜRFEN NICHT mit den Netzbereichen • NET_SIS • NET_TI_DEZENTRAL NET_TI_ZENTRAL • NET_TI_OFFENE_FD • NET_TI_GESICHERTE_FD • ANLW_BESTANDSNETZE kollidieren.

ANLW_IAG_ ADDRESS	IP Adresse	ANLW_IAG_ADDRESS ist die Adresse des Default Gateways. Diese IP-Adresse MUSS innerhalb des ANLW_WAN_NETWORK_SEGMENT liegen. Die Adresse wird entweder über DHCP automatisch (DHCP_CLIENT_WAN_STATE=ENABLED oder DHCP_CLIENT_LAN_STATE=ENABLED) oder anderenfalls manuell durch den Administrator konfiguriert. Bei automatischer Konfiguration per DHCP MUSS der Administrator den Wert von ANLW_IAG_ADDRESS ausschließlich einsehen können.
ANLW_ AKTIVE_ BESTANDS NETZE	Liste von IP-Address-Segmenten	Der Administrator MUSS manuell aus der empfangenen Liste der zur Verfügung stehenden angeschlossene Netze des Gesundheitswesens mit WANDA Basic (gemäß TUC_KON_283 „Infrastruktur Konfiguration aktualisieren“) einzelne deaktivieren, bzw. nach vorheriger Deaktivierung, freischalten können. Nur die freigegeben Netze werden in dieser Variablen erfasst und sind aus den Netzwerken der Einsatzumgebung erreichbar. Wird eine Änderung an der Liste der freigegebenen Netze vorgenommen, so MUSS der Konnektor für jedes dieser freigegebenen Netz in DNS_SERVERS_BESTANDSNETZE ein DNS-Referer-Eintrag für jede der dazugehörigen Domains mit allen zugehörigen DNS-Servern im Konnektor hinterlegen. Die Werte hierzu werden der via TUC_KON_283 aktualisierten Bestandsnetze.xml entnommen. Für hier „nicht freigegebene“ oder zwischenzeitlich gelöschte Netze DARF der Konnektor NICHT Referer-Einträge in DNS_SERVERS_BESTANDSNETZE enthalten. Die Einträge in DHCP_AKTIVE_BESTANDSNETZE_ROUTES sind entsprechend zu aktualisieren. Der Konnektor MUSS nach jeder Änderung dieser Variablen durch den Administrator den TUC_KON_304 „Netzwerk-Routen einrichten“ aufrufen.
ANLW_ IA_ BESTANDSNETZE	AN	Der Konnektor MUSS alle über TUC_KON_283 übermittelten angeschlossenen Netze des Gesundheitswesens mit WANDA Basic aktivieren. Eine spätere manuelle Deaktivierung über das Management-Interface durch den Administrator ist möglich. Dieses

		Verhalten ist als Standardverhalten zu konfigurieren.
	AUS	Der Konnektor MUSS alle über TUC_KON_283 übermittelten angeschlossenen Netze des Gesundheitswesens mit WANDA Basic anbieten, diese aber nicht aktivieren. Eine spätere manuelle Aktivierung erfolgt über das Mangement-Interface durch den Administrator.

[<=]

TIP1-A_5537 - Anzeige IP-Routinginformationen
Der Konnektor MUSS über die Managmentschnittstelle die konfigurierten IP-Routen und die aktuelle IP-Routingtabelle mit mindestens folgenden Informationen anzeigen:

- Forwarding Status
- Zieladresse/Prefix
- Gateway (Next-Hop)
- Routing Typ
- Routing Protocol
- Routing Preference.

[<=]

TIP1-A_4762 - Konfigurationsparameter Firewall-Schnittstelle
Im Anschluss an eine Anpassung der ANLW_FW_SIS_ADMIN_RULES MUSS der Konnektor die Firewall neu erstellen und laden.

Tabelle 320: TAB_KON_625 - Konfigurationsparameter Firewall-Schnittstelle

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
ANLW_FW_SIS_ADMIN_RULES	Firewall Regelset	Der Administrator MUSS Firewall-Regeln (für den einschränkenden Zugriff auf die SIS), auf Grundlage der Parameter Absender-IP-Adresse, Empfänger-IP-Adresse, Protokoll (ggf. mit Absender-Port und Empfänger-Port) und Verbindungsrichtung, einfügen, editieren und löschen können.

[<=]

4.2.2 DHCP-Server

Innerhalb des Kapitels DHCP-Servers werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „DHCP“
- Konfigurationsparameter: „DHCP_SERVER_“

4.2.2.1 Funktionsmerkmalweite Aspekte

TIP1-A_4763 - DHCP-Server des Konnektors
Der Konnektor MUSS an seiner LAN-Schnittstelle einen DHCP-Server gemäß [RFC2131] und [RFC2132] anbieten.
[<=]

4.2.2.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.2.2.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.2.2.4 Interne TUCs, auch durch Fachmodule nutzbar

Keine.

4.2.2.5 Operationen an der Außenschnittstelle

4.2.2.5.1 Liefere Netzwerkinformationen über DHCP

TIP1-A_4765 - Liefere Netzwerkinformationen über DHCP
Der DHCP-Server des Konnektors MUSS an der Client-Schnittstelle eine Operation zur Lieferung von Netzwerkinformationen über DHCP anbieten**.**

Tabelle 321: TAB_KON_626 „Liefere Netzwerkinformationen über DHCP“

Name	Liefere Netzwerkinformationen über DHCP
Beschreibung	Der Konnektor MUSS anfragenden Clients per DHCP die konfigurierten Netwerkinformationen liefern (siehe Tabelle TAB_KON_628 und Tabelle TAB_KON_629).
Aufrufparameter	gemäß [RFC2131], [RFC2132]
Rückgabe	gemäß [RFC2131], [RFC2132]
Standardablauf	Die an den aufrufenden Client zu übergebenden Parameter ergeben sich aus Tabelle TAB_KON_628 und Tabelle TAB_KON_629: Falls DHCP_SERVER_STATE = Enabled: • Anhand der MAC-Adresse des anfragenden Client wird die Clientgruppe aus DHCP_SERVER_CLIENTGROUPS bzw. DHCP_SERVER_DEFAULT_CLIENTGROUP ausgewählt. • DHCP_OWNDNS_ENABLED

	• Enabled: DNS-Server = <konnektoreigene Adresse> • Disabled: DNS-Server = DHCP_DNS_ADDR • DHCP_NTP • Enabled: NTP-Server = <konnektoreigene Adresse> • Disabled: Keine Wertübermittlung • DHCP_OWNDGW_ENABLED • Enabled: DGW = <konnektoreigene Adresse> • Disabled: DGW = DHCP_DGW_ADDR • Falls Client-MAC-Adresse in DHCP_STATIC_LEASE • IP_Address = die in der Static Lease konfigurierte Adresse. • Falls Client IP-Adresse = 0.0.0.0 oder innerhalb DHCP_SERVER_DYNAMIC_RANGE • IP_Address = IP_Address aus DHCP_SERVER_DYNAMIC_RANGE • Sonst: keine Zuweisung (Empfehlung: DHCPNAK an den Client) • Netzmaske = DHCP_IP_NETMASK • Domainname = DHCP_DOMAINNAME • Hostname = DHCP_HOSTNAME • Lease Dauer = DHCP_LEASE_TTL • Routen bestehend aus • DHCP_AKTIVE_BESTANDSNETZE_ROUTES • DHCP_INTRANET_ROUTES • DHCP_ROUTES • Weitere DHCP-Optionen = DHCP_OPTIONS • MTU = ANLW_LAN_MTU
Fehlercodes	Vgl. [RFC2131], [RFC2132]
Vorbedingungen	Der DHCP-Server des Konnektors MUSS aktiviert und konfiguriert sein.
Nachbedingungen	Der DHCP-Server MUSS die DHCP-Antwort geliefert haben. Die Statusinformationen (z.B. Client Lease) müssen gemäß [RFC2131] gespeichert werden.
Hinweise	Keine

[<=]

4.2.2.6 Betriebsaspekte

TIP1-A_4766 - Deaktivierbarkeit des DHCP-Servers
Der DHCP- Server des Konnektors MUSS durch den Administrator über die Managementschnittstelle aktivierbar und deaktivierbar sein (gemäß TAB_KON_627). Der DHCP-Server MUSS bei der Auslieferung deaktiviert sein.
Bei der Aktivierung MUSS der Konnektor den TUC_KON_343 "Initialisierung DHCP-Server" durchlaufen.
Sobald DHCP_SERVER_STATE geändert wurde, muss TUC_KON_256{"DHCP/SERVER/STATECHANGED"; Op; Info; "STATE=$DHCP_SERVER_STATE "} aufgerufen werden.

Tabelle 322: TAB_KON_627 „Aktivierung des DHCP-Servers"

Referenz ID	Belegung	Bedeutung
DHCP_SERVER_STATE	Enabled / Disabled	Der DHCP-Server MUSS durch den Administrator aktivierbar und deaktivierbar sein.

[<=]

TIP1-A_4767 - Konfiguration des DHCP-Servers
Der Konnektor MUSS die Möglichkeit bieten die in Tabelle TAB_KON_628 und Tabelle TAB_KON_629 beschriebenen Parameter des DHCP-Servers über die Managementschnittstelle zu konfigurieren.

Tabelle 323: TAB_KON_628 „Basiskonfiguration des DHCP-Servers"

Referenz ID	Belegung	Bedeutung
DHCP_SERVER_NETWORK	IP-Adresse	IP-Netzwerk der Einsatzumgebung.
DHCP_SERVER_BROADCAST	IP-Adresse	Die Broadcast-Adresse des Konnektors am LAN-Interface
DHCP_SERVER_ DYNAMIC_RANGE	von – bis IP-Adresse	Adressbereich für Adressen die dynamisch vergeben werden dürfen.
DHCP_SERVER_ CLIENTGROUPS	Name der Clientgruppe; Liste an MAC-Adressen	Der Konnektor MUSS dem Administrator über die Managementschnittstelle die Möglichkeit bieten mindestens zwei Client-Gruppen zu verwalten.
DHCP_SERVER_ DEFAULT_CLIENTGROUP	Client-Gruppe	Standardmäßig eingestellte Client-Gruppe. Wird verwendet falls DHCP-Anfrage keiner anderen Client-Gruppe zugeordnet werden kann.

Tabelle 324: TAB_KON_629 „Client-Gruppenspezifische Konfigurationsoptionen des Konnektor-DHCP-Servers"

ReferenzID	Belegung	Bedeutung
Die gesamte Parameterliste ist für jede Client-Gruppe getrennt konfigurierbar		

DHCP_ OWNDNS_ ENABLED	Enabled/Disabled	Der Administrator MUSS konfigurieren können, ob der konnektoreigene DNS-Server als Parameter übergeben wird. Default-Wert: Disabled
DHCP_DNS_ ADDR	IP-Adressen der DNS-Server	Falls der konnektoreigene DNS-Server nicht übergeben werden soll, müssen die Adressen externer aus dem Netz der Einsatzumgebung erreichbaren DNS-Server als Parameter übergeben werden. Der Administrator MUSS diese Adressen konfigurieren können.
DHCP_NTP	Enabled/Disabled	Der Administrator MUSS konfigurieren können, ob der Konnektor die Adresse des Konnektor internen NTP-Servers per DHCP an die Clients sendet. Default-Wert: Enabled
DHCP_ OWNDGW_ ENABLED	Enabled/Disabled	Der Administrator MUSS konfigurieren können, ob der Konnektor beim Client als Default-Gateway gesetzt werden soll. Default-Wert: Disabled
DHCP_DGW_ ADDR	IP-Adresse des DGW	Falls der Konnektor nicht als Default Gateway gesetzt werden soll, muss die Adresse des zu verwendenden DGW als Parameter übergeben werden. Der Administrator MUSS die Adresse des DGW konfigurieren können.
DHCP_IP_ NETMASK	Netzmaske	Der Administrator MUSS die Netmask des Clients konfigurieren können.
DHCP_ DOMAINNAME	Domainname	Der Administrator MUSS den Domainnamen des Clients konfigurieren können.
DHCP_ HOSTNAME	Liste von Tupel aus Hostname und Mac-Adresse	Der Administrator MUSS eine Liste von Hostname der Clients konfigurieren können (Einträge einfügen, ändern, löschen).

DHCP_ STATIC_LEASE	Liste von Tupel aus IP- und Mac-Adresse	Der Administrator MUSS für jede MAC-Adresse Static Lease konfigurieren können.
DHCP_ LEASE_TTL	X Minuten	Der Administrator MUSS Lease-Dauer der dynamischen Adressen konfigurieren können.
DHCP_ AKTIVE_ BESTANDS NETZE_ ROUTES	Liste von Tupel: Netzwerksegment je INTRANET und Adresse für Next Hop je freigegebenem angeschlossenen Netze des Gesundheitswesens mit WANDA Basic	Der Administrator MUSS je freigegebenem angeschlossenen Netze des Gesundheitswesens mit WANDA Basic (aus ANLW_AKTIVE_BESTANDSNETZE) den an den Client zu übermittelnden Routen-Eintrag aktivieren oder deaktivieren können.
DHCP_ INTRANET_ ROUTES	Liste von Tupel: Netzwerksegment je INTRANET und Adresse für Next Hop in die definierten Intranets	Der Administrator MUSS je Intranet-Tupel (aus ANLW_LEKTR_INTRANET_ROUTES) den an den Client zu übermittelnden Routen-Eintrag aktivieren oder deaktivieren können.
DHCP_ ROUTES	Tupel Netzwerksegment und Adresse für Next Hop	Der Administrator MUSS Routen zur Verteilung an die Clients frei konfigurieren können. Der Konnektor MUSS sicherstellen, diese Listeneinträge keine Überschneidungen mit folgenden Netzsegmenten haben: - dem Netzwerksegment ANLW_LAN_NETWORK_SEGMENT - dem Netzwerksegment ANLW_WAN_NETWORK_SEGMENT - jedes Netzsegmente in ANLW_BESTANDSNETZE ANLW_AKTIVE_BESTANDSNETZE ANLW_LEKTR_INTRANET_ROUTES Die Routen SOLLEN über DHCP Option 121 (Windows Vista oder höher) bzw. DHCP Option 249 (Windows XP und darunter) verteilt werden.
DHCP_ OPTIONS	Liste an weiteren DHCP-Optionen.	Vom Administrator konfigurierbare Liste an weiteren DHCP-Options gemäß [RFC2132].

		Die Umsetzung dieser Konfigurationsmöglichkeit KANN entfallen.

[<=]

4.2.2.6.1 TUC_KON_343 „Initialisierung DHCP-Server"

TIP1-A_4768 - TUC_KON_343 „Initialisierung DHCP-Server"
Der Konnektor MUSS in der Bootup-Phase TUC_KON_343 "Initialisierung DHCP-Server" durchlaufen.

Tabelle 325: TAB_KON_630 - TUC_KON_343 „Initialisierung DHCP-Server"

Element	Beschreibung
Name	TUC_KON_343 "Initialisierung DHCP-Server"
Beschreibung	Falls DHCP-Server Konfiguration aktiv ist, muss der Konnektor in der Bootup-Phase oder bei einer Aktivierung des Servers den DHCP-Server starten.
Anwendungsumfeld	Bereitstellen der Netzwerkkonfiguration für den Betrieb
Eingangsanforderung	Keine
Auslöser und Vorbedingungen	Bootup oder Ereignis DHCP/SERVER/STATECHANGED
Eingangsdaten	Keine
Komponenten	Konnektor
Ausgangsdaten	Keine
Standardablauf	Falls DHCP_SERVER_STATE = enabled - den DHCP-Server starten Falls DHCP_SERVER_STATE = disabled - den DHCP-Server stoppen
Varianten/Alternativen	Keine
Fehlerfälle	4168: DHCP-Server konnte nicht gestartet werden
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 326: TAB_KON_631 Fehlercodes TUC_KON_343 „Initialisierung DHCP-Server"

Fehlercode	ErrorType	Severity	Fehlertext
4168	Technical	Error	Der DHCP-Server des Konnektors konnte nicht gestartet werden.

[<=]

4.2.3 DHCP-Client

Innerhalb des Kapitels DHCP-Client werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „DHCP“
- Konfigurationsparameter: „DHCP_CLIENT_“

4.2.3.1 Funktionsmerkmalweite Aspekte

TIP1-A_4769 - DHCP Client Funktionalität des Konnektors
Der Konnektor MUSS an seiner LAN- und WAN-Schnittstelle die Möglichkeit bieten jeweils DHCP zu nutzen.
Der DHCP-Client des Konnektors MUSS die empfangenen Parameter wie folgt verwenden:

- Die IP-Adresse und Subnetzmaske müssen dem Interface zugewiesen und in den Variablen ANLW_LAN_IP_ADDRESS bzw. ANLW_WAN_IP_ADDRESS und ANLW_LAN_SUBNETMASK gespeichert werden.
- Der für das Interface, auf Anfrage, gelieferte Wert der MTU Size KANN übernommen werden.
- Das Default Gateway (DGW) muss in der Variable ANLW_IAG_ADDRESS gespeichert werden.
- DNS-Server muss in der Variable DNS_SERVERS_INT gespeichert werden.

Weitere DHCP-Parameter DÜRFEN nicht übernommen werden.
[<=]

4.2.3.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_4771 - Reagieren auf DHCP/LAN_CLIENT/ STATECHANGED- und DHCP/WAN_CLIENT/ STATECHANGED-Ereignisse
Wenn das Ereignis DHCP/LAN_CLIENT/STATECHANGED oder DHCP/WAN_CLIENT/STATECHANGED empfangen wird, MUSS TUC_KON_341 „DHCP-Informationen beziehen“ aufgerufen werden.
[<=]

4.2.3.3 Interne TUCs, nicht durch Fachmodule nutzbar

4.2.3.3.1 TUC_KON_341 „DHCP-Informationen beziehen“

TIP1-A_4772 - TUC_KON_341 „DHCP-Informationen beziehen“
Der Konnektor MUSS den technischen Use Case TUC_KON_341 „DHCP-Informationen beziehen“ umsetzen.

Tabelle 327: TAB_KON_632 – TUC_KON_341 „DHCP Informationen beziehen“

Element	Beschreibung
Name	TUC_KON_341 DHCP-Informationen beziehen

Beschreibung	Der Konnektor muss seine WAN- und/oder LAN-Schnittstelle individuell über einen DHCP-Server aus dem Netz der Einsatzumgebung beziehen können.
Anwendungsumfeld	Netzwerkkonfiguration für den Betrieb des Konnektors
Eingangsanforderung	Der Konnektor muss zur Netzwerk-Interface-Konfiguration DHCP nutzen sofern keine statischen Informationen vorhanden sind.
Auslöser	Bootup, Ablauf einer DHCP-Lease, manuell angestoßenes DHCP-Renew, Aktivierung der DHCP-Client-Funktionalität.
Vorbedingung	aktivierte DHCP-Client Funktion über die Variablen DHCP_CLIENT_LAN_STATE bzw. DHCP_CLIENT_WAN_STATE
Eingangsdaten	Netzwerk-Adapter (LAN oder WAN) für den DHCP-Informationen bezogen werden sollen
Komponenten	Konnektor
Ausgangsdaten	DHCP-Informationen vom DHCP-Server der Einsatzumgebung
Standardablauf	• Ermitteln von DHCP-Informationen (DHCPDISCOVER und DHCPREQUEST) gemäß [RFC2131], [RFC2132] • Übernahme der ermittelten Werte, ausschließlich für die in Tabelle TAB_KON_683 LAN-Adapter IP-Konfiguration bzw. Tabelle TAB_KON_685 WAN-Adapter IP-Konfiguration aufgeführten Variablen • Wenn DHCP Client LAN-Adapter, nur bei IP-Adressen-Wechsel: Erzeugen eines Events durch den Aufruf von TUC_KON_256{"DHCP/LAN_CLIENT/RENEW"; Op; Info; "IP_ADDRESS=$Belegung"} • Wenn DHCP Client WAN-Adapter, nur bei IP-Adressen-Wechsel: Erzeugen eines Events durch den Aufruf von TUC_KON_256{"DHCP/WAN_CLIENT/RENEW"; Op; Info; "IP_ADDRESS=$Belegung"}
Varianten/Alternativen	Keine
Fehlerfälle	4169: Konnektor erhält keine DHCP-Informationen 4170: Konnektor besitzt identische IP-Adressen am WAN- und LAN-Interface
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 328: TAB_KON_633 Fehlercodes TUC_KON_341 „DHCP-Informationen beziehen“

Fehlercode	ErrorType	Severity	Fehlertext
4169	Technical	Error	Konnektor erhält keine DHCP-Informationen.
4170	Technical	Error	Konnektor besitzt identische IP-Adressen am WAN und LAN

[<=]

4.2.3.4 Interne TUCs, auch durch Fachmodule nutzbar

Keine.

4.2.3.5 Operationen an der Außenschnittstelle

Keine.

4.2.3.6 Betriebsaspekte

TIP1-A_4773 - Konfiguration des DHCP-Clients
Die DHCP-Client Funktionalität MUSS für LAN- und WAN-Interface vom Administrator getrennt aktivierbar und deaktivierbar sein (gemäß TAB_KON_634). Falls der DHCP-Client nicht verwendet wird MUSS sichergestellt werden, dass eine statische Konfiguration, für den LAN-Adapter gemäß Tabelle TAB_KON_683 LAN-Adapter IP-Konfiguration bzw. für den WAN-Adapter gemäß Tabelle TAB_KON_685 WAN-Adapter IP-Konfiguration, existiert bevor die Netzwerkeinstellungen übernommen werden.
Sobald Parameter geändert wurden, MUSS TUC_KON_256 „Systemereignis absetzen“ je nachdem auf welchem Interface der Client aktiviert oder deaktiviert wurde mit folgenden Parameter aufgerufen werden:

TUC_KON_256{"DHCP/LAN_CLIENT/STATECHANGED"; Op; Info;
"STATE=$DHCP_CLIENT_LAN_STATE"; doDisp = false}
oder
TUC_KON_256{"DHCP/WAN_CLIENT/STATECHANGED "; Op; Info;
"STATE=$DHCP_CLIENT_WAN_STATE "; doDisp = false}

Tabelle 329: TAB_KON_634 „Konfiguration des DHCP-Clients“

ReferenzID	Belegung	Bedeutung
DHCP_CLIENT_ LAN_STATE	Enabled/Disabled	Der Administrator muss den DHCP-Client an der LAN-Schnittstelle aktivieren oder deaktivieren können.
DHCP_CLIENT_ WAN_STATE	Enabled/Disabled	Der Administrator muss den DHCP-Client an der WAN-Schnittstelle aktivieren oder deaktivieren können.

[<=]

TIP1-A_4774 - Manuelles anstoßen eines DHCP-Lease-Renew
Der Administrator MUSS die Möglichkeit haben die DHCP-Lease des Konnektors für jedes Interface getrennt zu erneuern.
[<=]

TIP1-A_4776 - Setzen der IP-Adresse nach Timeout
Falls der DHCP-Client auf der LAN-Seite nach einem Timeout von 30s keine IP-Adresse bezogen hat, MUSS gemäß [RFC3927] eine Default-Adresse aus 169.254/16 vergeben werden.
[<=]

4.2.4 VPN-Client

Der VPN-Client beschreibt die Absicherung der Anbindung des Konnektors an die TI und die Bestandsnetze. Während der technische Kern dieser Funktion, der Aufbau der VPN-Kanäle zu den Konzentratoren, in [gemSpec_VPN_ZugD#TUC_VPN-ZD_0001] und [gemSpec_VPN_ZugD#TUC_VPN-ZD_0002] beschrieben wird, regelt dieses Kapitel die Interaktion, sowie die Konfiguration des VPN-Clients innerhalb des Konnektors.

Innerhalb des Kapitels VPN-Client werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „NETWORK“
- Konfigurationsparameter: „VPN_“

4.2.4.1 Funktionsmerkmalweite Aspekte

TIP1-A_4778 - Anforderungen an den VPN-Client
Der Konnektor MUSS sich im Rahmen des IPsec-Verbindungsaufbaus gegenüber den VPN-Konzentratoren mit seiner Identität ID.NK.VPN ausweisen.
Der VPN-Client im Konnektor MUSS das folgende Event generieren, sobald der VPN-Tunnel zur TI nicht mehr zur Verfügung steht:
Rufe TUC_KON_256 {"NETWORK/VPN_TI/DOWN"; Op; Warning;}
Der VPN-Client im Konnektor MUSS das folgende Event generieren, sobald der VPN-Tunnel zum SIS nicht mehr zur Verfügung steht:
Rufe TUC_KON_256 {"NETWORKVPN_SIS/DOWN"; Op; Warning;}
Der Hersteller des Konnektor MUSS sicherstellen, dass eine Anbindung an einen Konzentrator ausschließlich dann möglich ist, wenn (MGM_LU_ONLINE = Enabled) gesetzt ist.
Der Administrator des Konnektor MUSS durch die Managementschnittstelle manuell einen Verbindungsaufbau und einen Verbindungsabbau eines VPN-Tunnel zur TI (VPN_TI) oder zu den SIS (VPN_SIS) initiieren können.
[<=]

TIP1-A_4779 - Wiederholte Fehler beim VPN-Verbindungsaufbau
Der Konnektor MUSS gewährleisten, dass nach einem Fehler beim VPN-Verbindungsaufbau nicht unmittelbar ein weiterer Versuch des Verbindungsaufbaus durchgeführt wird.
Hierzu MUSS der Hersteller ein inkrementelles (schrittweise anwachsend) Verfahren wählen, welcher den zeitlichen Abstand zwischen einzelnen Versuchen des VPN-Verbindungsaufbau definiert. Dieser Abstand MUSS maximal fünf Minuten betragen. (Diese Pause soll es dem Konnektor ermöglichen, noch ausreichend Ressourcen für die verbleibenden Services zur Verfügung zu stellen).
[<=]

4.2.4.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_4780 - TI VPN-Client Start Events
Beim Auftreten einer der nachfolgenden Events MUSS der Konnektor den TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen" starten, sofern auch MGM_LU_ONLINE = Enabled.

- Event NETWORKVPN_TI/DOWN
- Event MGM/LU_CHANGED/LU_ONLINE

[<=]

TIP1-A_4781 - SIS VPN-Client Start Events
Beim Auftreten einer der nachfolgenden Events MUSS der Konnektor den TUC_KON_322 „Verbindung zu dem VPN-Konzentrator des SIS aufbauen" starten, sofern ANLW_INTERNET_MODUS = SIS, MGM_LU_ONLINE = Enabled und die Verbindung VPN-Konzentrator TI aufgebaut ist:

- Event NETWORKVPN_SIS/DOWN

[<=]

TIP1-A_5417 - TI VPN-Client Stop Events
Beim Auftreten einer der nachfolgenden Events MUSS der Konnektor den VPN-Tunnel zur TI beenden:

- MGM/LU_CHANGED/LU_ONLINE mit (Active=Disabled)

[<=]

TIP1-A_4782 - SIS VPN-Client Stop Events
Beim Auftreten einer der nachfolgenden Events MUSS der Konnektor den VPN-Tunnel zum SIS beenden:

- MGM/LU_CHANGED/LU_ONLINE mit (Active=Disabled)

[<=]

Hinweis: Wenn der IPsec-Tunnel VPN_SIS aufgebaut ist, zeigt die Default Route im Konnektor auf die innere Tunnel-IP-Adresse des VPN-Konzentrators SIS. Dies ist bei einer Trennung und dem Wiederaufbau der Verbindung VPN_TI zu beachten.

4.2.4.3 Interne TUCs, nicht durch Fachmodule nutzbar

4.2.4.3.1 TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen"

TIP1-A_4783 - TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen"
Der Konnektor MUSS den technischen Use Case TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen" umsetzen.

Tabelle 330: TAB_KON_635 – TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen"

Element	Beschreibung
Name	TUC_KON_321 Verbindung zu dem VPN-Konzentrator der TI aufbauen

Beschreibung	Es wird ein IPsec-Tunnel zum VPN-Konzentrator der TI aufgebaut werden. Über den erfolgreichen Aufbau wird per Event informiert.
Auslöser	Bootup-Phase TUC_KON_305 „LAN-Adapter initialisieren“ TUC_KON_306 „WAN-Adapter initialisieren“ Event MGM/LU_CHANGED/LU_ONLINE Event NETWORK/VPN/CONFIG_CHANGED Optional: Änderungen ANLW_AKTIVE_BESTANDSNETZE Manueller Aufruf über Managementschnittstelle
Vorbedingungen	Der TUC_KON_304 „Netzwerk-Routen einrichten“ MUSS fehlerfrei durchgelaufen sein.
Eingangsdaten	
Komponenten	Konnektor
Ausgangsdaten	Der virtuelle Adapter VPN_TI mit der IP-Adresse VPN_TUNNEL_TI_INNER_IP des Konnektors wurde zur Verfügung gestellt. • Innere Tunnel IP-Adresse des VPN-Konzentrators TI • DNS_SERVERS_TI • VPN_KONZENTRATOR_TI_IP_ADDRESS • DOMAIN_SRVZONE_TI
Standardablauf	1) Wenn der Auslöser = Event NETWORK/VPN/CONFIG_CHANGED oder eine Änderung von ANLW_AKTIVE_BESTANDSNETZE ist, muss der VPN-Tunnel TI abgebaut werden. 2) Wenn der VPN-Tunnel TI noch aktiv ist, ist der Ablauf abgeschlossen. Anderenfalls ist mit Ablaufschritt 3) fortzufahren. 3) Prüfen, MGM_LU_ONLINE = Enabled, falls nicht ist der TUC ohne Ausgabe einer Fehlermeldung zu beenden. 4) Prüfen, ob die im Konnektor hinterlegte CRL noch gültig ist. falls nicht, muss der TUC_KON_040 „CRL aktualisieren“ aufgerufen werden, Anschließend erneut prüfen, ob die im Konnektor hinterlegte CRL nun gültig ist. Falls die CRL nicht gültig ist, ist der TUC mit Fehler zu beenden. 5) Aufrufen von TUC_VPN-ZD_0001 „IPsec Tunnel TI aufbauen” Die folgenden Rückgabewerte des TUC_VPN-ZD_0001

	„IPsec Tunnel TI aufbauen" sind in die laufende Konfiguration des Konnektors zu übernehmen: • VPN_TUNNEL_TI_INNER_IP • DNS_SERVERS_TI 6) Aufrufen von TUC_KON_304 „Netzwerk-Routen einrichten" Sobald der Tunnel erfolgreich aufgebaut wurde, ist der folgende Event zu generieren: TUC_KON_256 {"NETWORK/VPN_TI/UP"; Op; Info;IP= $VPN_TUNNEL_TI_INNER_IP}
Varianten/Alternativen	Keine
Fehlerfälle	(→4) CRL ist abgelaufen (outdated); Herstellerspezifisch kann entweder (4a) oder (4b) umgesetzt werden: (4a) Kritischer Fehlerzustand EC_CRL_Out_Of_Date wurde noch nicht festgestellt: 4173 (4b) kritischer Fehlerzustand EC_CRL_Out_Of_Date wurde bereits festgestellt: 4002 (->4) Wenn Fehler 4173 bzw. 4002 nicht zutreffen, ist ein herstellerspezifischer Fehler zu verwenden. (→5) VPN-Tunnel konnte nicht aufgebaut werden; Fehlercode: 4174
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 331: TAB_KON_636 Fehlercodes TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4002	Security	Fatal	Der Konnektor befindet sich in einem kritischen Betriebszustand
4172	Technical	Fatal	Es ist keine Online-Verbindung zulässig.

4173	Technical	Fatal	Die CRL ist nicht mehr gültig (outdated).
4174	Technical	Fatal	TI-VPN-Tunnel: Verbindung konnte nicht aufgebaut werden

[<=]

4.2.4.3.2 TUC_KON_322 „Verbindung zu dem VPN-Konzentrator des SIS aufbauen"

TIP1-A_4784 - TUC_KON_322 „Verbindung zu dem VPN-Konzentrator des SIS aufbauen"
Der Konnektor MUSS den technischen Use Case TUC_KON_322 „Verbindung zu dem VPN-Konzentrator der SIS aufbauen" umsetzen.

Tabelle 332: TAB_KON_637 – TUC_KON_322 „Verbindung zu dem VPN-Konzentrator der SIS aufbauen"

Element	Beschreibung
Name	TUC_KON_322 Verbindung zu dem VPN-Konzentrator der SIS aufbauen
Beschreibung	Es muss ein IPsec-Tunnel zum VPN-Konzentrator der SIS aufgebaut werden
Auslöser	Bootup-Phase TUC_KON_305 „LAN-Adapter initialisieren TUC_KON_306 „WAN-Adapter initialisieren Event NETWORK/VPN/CONFIG_CHANGED Optional: Event MGM/LU_CHANGED/LU_ONLINE Manueller Aufruf über Managementschnittstelle
Vorbedingungen	ANLW_INTERNET_MODUS = SIS Die Verbindung VPN-Konzentrator TI ist aufgebaut. Der TUC_KON_304 „Netzwerk-Routen einrichten" muss erfolgreich durchgeführt worden sein.
Eingangsdaten	Keine
Komponenten	Konnektor
Ausgangsdaten	Der virtuelle Adapter VPN_SIS mit der IP-Adresse VPN_TUNNEL_SIS_INNER_IP wurde zur Verfügung gestellt.

	• Innere Tunnel-IP-Adresse des VPN-Konzentrators SIS • VPN_KONZENTRATOR_SIS_IP_ADDRESS • DNS_SERVER_SIS
Standardablauf	1) Wenn der Auslöser Event NETWORK/VPN/CONFIG_CHANGED ist, muss der VPN-Tunnel SIS abgebaut werden. 2) Wenn der VPN-Tunnel SIS noch aktiv ist, ist der Ablauf abgeschlossen. Anderenfalls ist mit Ablaufschritt 3) fortzufahren. 3) Prüfen, ob (MGM_LU_ONLINE=Enabled). falls nicht ist der TUC ohne Ausgabe einer Fehlermeldung zu beenden. 4) entfällt 5) Prüfen, ob die im Konnektor hinterlegte CRL noch gültig ist. falls nicht, MUSS der TUC_KON_040 „CRL aktualisieren" aufgerufen werden, Anschließend erneut prüfen, ob die im Konnektor hinterlegte CRL nun gültig ist. Falls die CRL nicht gültig ist, ist der TUC mit Fehler zu beenden. 6) Aufrufen von TUC_VPN-ZD_0002 „IPsec Tunnel SIS aufbauen" 7) Aufrufen von TUC_KON_304 „Netzwerk-Routen einrichten" Sobald der Tunnel erfolgreich aufgebaut wurde, ist der folgende Event zu generieren: TUC_KON_256 {"NETWORK/VPN_SIS/UP"; Op; Info;IP= $VPN_TUNNEL_SIS_INNER_IP}
Varianten/Alternativen	Keine
Fehlerfälle	(→3) Keine Online-Verbindung zulässig; 4172 (→5) CRL ist abgelaufen (outdated); Herstellerspezifisch kann entweder (5a) oder (5b) umgesetzt werden: (5a) Kritischer Fehlerzustand EC_CRL_Out_Of_Date wurde noch nicht festgestellt: 4173 (5b) kritischer Fehlerzustand EC_CRL_Out_Of_Date wurde bereits festgestellt: 4002 (->5) Wenn Fehler 4173 bzw. 4002 nicht zutreffen, ist ein herstellerspezifischer Fehler zu verwenden. (→6) VPN Tunnel konnte nicht aufgebaut werden; Fehlercode: 4176
Nichtfunktionale Anforderungen	Keine

Zugehörige Diagramme	Keine

Tabelle 333: TAB_KON_638 Fehlercodes TUC_KON_322 „Verbindung zu dem VPN-Konzentrator der SIS aufbauen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4002	Security	Fatal	Der Konnektor befindet sich in einem kritischen Betriebszustand
4172	Technical	Fatal	Es ist keine Online-Verbindung zulässig.
4173	Technical	Fatal	Die CRL ist nicht mehr gültig (outdated).
4176	Technical	Fatal	SIS-VPN-Tunnel: Verbindung konnte nicht aufgebaut werden

[<=]

4.2.4.4 Interne TUCs, auch durch Fachmodule nutzbar

Keine

4.2.4.5 Operationen an der Außenschnittstelle

Keine

4.2.4.6 Betriebsaspekte

TIP1-A_5415 - Initialisierung „VPN-Client"
Der Konnektor MUSS in der Bootup-Phase zur Initialisierung des Funktionsmerkmals „VPN-Client":

- die Verbindung zum VPN-Konzentrator TI aufbauen (TUC_KON_321)
- die Verbindung zum VPN-Konzentrator SIS aufbauen (TUC_KON_322)

[<=]

TIP1-A_4785-03 - Konfigurationsparameter VPN-Client
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen am VPN-Client gemäß Tabelle TAB_KON_639 vorzunehmen. Der Konnektor MUSS bei einer Änderung der Konfigurationswerte den folgenden Event auslösen:
Rufe TUC_KON_256 {"NETWORK/VPN/CONFIG_CHANGED"; Op; Info;; doDisp = false}

Tabelle 334: TAB_KON_639 – Konfigurationsparameter VPN-Client

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
IKE_ KEEPALIVE_ MODUS	Enabled/Disabled	Der Administrator MUSS einstellen können, ob IKE Keep-Alive-Pakete gesendet werden. Ein Hinweis MUSS ausgegeben werden, dass dies bei Nutzung von Dial-Up-Verbindungen nicht zu empfehlen ist. Dies dient der Vermeidung von Kosten bei Nutzung eines Internetzugangs ohne Flatrate. Default-Wert: Enabled
IKE_ KEEPALIVE_ INTERVAL	X Sekunden	Der Administrator MUSS die Zeit in Sekunden angeben können, nach der ein neues IKE Keep-Alive-Paket gesendet wird. Default-Wert: 30
IKE_ KEEPALIVE_ RETRY	X	Der Administrator MUSS angeben können, nach wie vielen IKE Keep-Alive-Paketen ohne Acknowledge Message die Verbindung beendet wird. Default-Wert: 3
VPN_IDLE_ TIMEOUT_ MODUS	Enabled/Disabled	Der Administrator MUSS einstellen können, ob nach Inaktivität die VPN-Verbindung automatisch abgebaut werden soll. Ein Hinweis MUSS ausgegeben werden, dass dies insbesondere bei Nutzung von Dial-Up-Verbindungen Enabled werden sollte. Default-Wert: Disabled
VPN_IDLE_ TIMEOUT	X Sekunden	Der Administrator MUSS die Zeit in Sekunden angeben können, nach der eine inaktive VPN-Verbindung zu einem Abbau der Verbindung führt. Default-Wert: 600
NAT_ KEEPALIVE_ MODUS	Enabled/Disabled	Der Administrator MUSS einstellen können, ob NAT Keep-Alive-Pakete gesendet werden. Ein Hinweis MUSS ausgegeben werden, dass dies insbesondere bei Nutzung von Dial-Up-Verbindungen nicht zu empfehlen ist. Default-Wert: Enabled
NAT_ KEEPALIVE_ INTERVAL	X Sekunden	Der Administrator MUSS die Zeit in Sekunden angeben können, nach der ein neues NAT Keep-Alive-Paket gesendet wird. Default-Wert: 20
VPN_ KONZENTRATOR_ TI_IP_ADDRESS	IP-Adresse	IP-Adresse des VPN-Konzentrators TI im Transportnetz zu dem der IPsec-Tunnel VPN_TI aufgebaut wird. Der Wert kann vom Administrator nur eingesehen werden.
VPN_ KONZENTRATOR_	IP-Adresse	IP-Adresse des VPN-Konzentrators SIS im Transportnetz zu dem der IPsec-Tunnel VPN_SIS aufgebaut wird. Der Wert kann vom Administrator nur eingesehen werden.

SIS_IP_ADDRESS		
VPN_TI_MTU	Paketgröße in Byte	Der Administrator MUSS die MTU für ESP-Pakete zur TI (excl. ESP-Header-Size) in den Grenzen von 576 bis 8076 konfigurieren können. Default-Wert: 1318
VPN_SIS_MTU	Paketgröße in Byte	Der Administrator MUSS die MTU für ESP Pakete zum SIS (excl. ESP-Header-Size) in den Grenzen von 576 bis 8076 konfigurieren können. Default-Wert: 1318
HASH_AND_URL	Enabled/Disabled	Der Administrator MUSS die Nutzung des hash&URL-Verfahrens zum Zertifikatsaustausch konfigurieren können. Wenn HASH_AND_URL = Enabled gesetzt ist, wird die URL für das hash&URL-Verfahren automatisch durch DNS SRV- und TXT-Anfragen mit Owner „_hashandurl._tcp.<DNS_DOMAIN_VPN_ZUGD_INT>„ ermittelt. Default-Wert: Disabled

[<=]

4.2.5 Zeitdienst

Der Zeitdienst schafft die Grundlage einer gleichen Systemzeit für alle in der TI einzusetzenden Produkttypen. Grundsätzlich ist ein NTP-Server der Stratum-3-Ebene innerhalb des Konnektors erforderlich, welcher die Zeitangaben eines NTP-Servers Stratum-2-Ebene abfragt (GS-A_3942). Die in [gemSpec_Net#5.1] „NTP-Topologie" getroffenen Anforderungen werden durch dieses Kapitel erweitert.

Innerhalb des Zeitdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „NTP"
- Konfigurationsparameter: „NTP_"

4.2.5.1 Funktionsmerkmalweite Aspekte

TIP1-A_4786 - Maximale Zeitabweichung
Falls der Leistungsumfang Online nicht aktiviert ist (MGM_LU_ONLINE=Disabled), MUSS sichergestellt werden, dass der maximale zulässige Fehler von +/- 20ppm (part per million) gegenüber einer Referenzuhr nicht überschritten wird. Dies entspricht einer maximalen Abweichung im Freilauf von +/- 34,56 Sekunden über 20 Tage.
[<=]

TIP1-A_4787 - Konfigurationsabhängige Funktionsweise
Der NTP-Server des Konnektors MUSS deaktiviert sein, falls der Konnektor Leistungsumfang Online nicht aktiviert ist (MGM_LU_ONLINE=Disabled.
[<=]

Falls die Systemzeit des Konnektors zu stark von der Zeit der zentralen TI-Plattform abweicht, deutet dies auf ein schwerwiegendes Problem im Konnektor oder der Umgebung hin, da dies im ordnungsgemäßen Betrieb nicht auftreten sollte.

TIP1-A_4788 - Verhalten bei Abweichung zwischen lokaler Zeit und erhaltenen Zeit
Der Konnektor DARF die im Konnektor vorgehaltene Systemzeit im Rahmen einer automatisierten Synchronisation NICHT aktualisieren, wenn die lokale Zeit von der im Rahmen der Synchronisation erhaltenen Zeit um mehr als NTP_MAX_TIMEDIFFERENCE abweicht. Dies betrifft NICHT Änderungen in der Darstellung der Systemzeit, die zeitzonenbedingt sind (MEZ -> MESZ -> MEZ), da die Zeitsynchronisation grundsätzlich UTC berücksichtigt. Bei einer erstmaligen Synchronisierung nach dem Boot-Vorgang oder bei einer erstmaligen Synchronisierung bei der Inbetriebnahme des Konnektors darf eine Synchronisation trotz einer Zeitabweichung größer einer Stunde durchgeführt werden. Daher MUSS der Konnektor bei einer Abweichung von mehr als einer Stunde in den kritischen Betriebszustand EC_TIME_DIFFERENCE_INTOLERABLE übergehen, ein weiterer fachlicher Betrieb des Konnektors DARF NICHT mehr erfolgen.
[<=]

Der kritische Betriebszustand kann anschließend über einen manuellen Eingriff (z. B. Reboot) behoben werden (siehe 3.3 Betriebszustand).

TIP1-A_4789 - Zustandsvariablen des Konnektor Zeitdiensts
TAB_KON_640 listet die zu verwendenden Zustandsvariablen des Konnektor NTP-Servers. Diese Werte DÜRFEN NICHT durch den Administrator geändert werden.

Tabelle 335: TAB_KON_640 Zustandswerte für Konnektor NTP-Server

ReferenzID	Belegung	Zustandswerte
NTP_WARN_PERIOD	30	Anzahl an Tagen nach der ersten erfolglosen Zeitsynchronisierung nach der eine Warnung an den Betreiber erfolgen soll
NTP_GRACE_PERIOD	50	Anzahl an Tagen nach der ersten erfolglosen Zeitsynchronisierung nach welcher der Konnektor in einen kritischen Betriebszustand übergehen muss. Dieser Parameter wirkt nur bei MGM_LU_ONLINE = Enabled.
NTP_MAX_ TIMEDIFFERENCE	3600	Maximale Zeitabweichung in Sekunden zwischen Systemzeit und Zeit des Stratum-2-Zeitservers zum Zeitpunkt der Zeitsynchronisierung. Dieser Parameter wirkt nur bei MGM_LU_ONLINE = Enabled.

[<=]

4.2.5.2 Durch Ereignisse ausgelöste Reaktionen
Keine.

4.2.5.3 Interne TUCs, nicht durch Fachmodule nutzbar
Keine.

4.2.5.4 Interne TUCs, auch durch Fachmodule nutzbar

4.2.5.4.1 TUC_KON_351 "Liefere Systemzeit"

TIP1-A_4790 - TUC_KON_351 „Liefere Systemzeit"
Der Konnektor MUSS den technischen Use Case TUC_KON_351 „Liefere Systemzeit" umsetzen.

Tabelle 336: TAB_KON_776 TUC_KON_351 „Liefere Systemzeit"

Element	Beschreibung
Name	TUC_KON_351 „Liefere Systemzeit"
Beschreibung	Der Konnektor MUSS die Systemzeit auf Anforderung an Fachmodule liefern können.
Anwendungsumfeld	Den Fachanwendungen ist die Systemzeit zu liefern.
Eingangsanforderung	Die Echtzeituhr des Konnektors wurde gemäß den geforderten Synchronisationsintervallen aktualisiert (bei MGM_LU_ONLINE=Enabled) oder manuell gesetzt (bei MGM_LU_ONLINE=Disabled)
Auslöser und Vorbedingungen	Fachmodule benötigen die aktuelle Systemzeit des Konnektors.
Eingangsdaten	Echtzeituhr des Konnektors
Komponenten	Konnektor, Fachmodule
Ausgangsdaten	Systemzeit des Konnektors
Standardablauf	Siehe [gemSpec_Net]
Varianten/Alternativen	Keine
Fehlerfälle	4178: Konnektor retourniert keine Systemzeit
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 337: TAB_KON_641 Fehlercodes TUC_KON_351 „Liefere Systemzeit"

Fehlercode	ErrorType	Severity	Fehlertext
4178	Technical	Error	Das Fachmodul konnte die aktuelle Systemzeit des Konnektors nicht abrufen

[<=]

4.2.5.5 Operationen an der Außenschnittstelle

4.2.5.5.1 Sync_Time

TIP1-A_4791 - Operation sync_Time
Der NTP-Server des Konnektors MUSS an der Client-Schnittstelle eine Operation sync_Time anbieten**.**

Tabelle 338: TAB_KON_642 Operation sync_Time

Name	I_NTP_Time_Information:sync_Time
Beschreibung	Der Konnektor MUSS anfragenden Clients (z.B. Ärztearbeitsplatz) per NTP-Version 4 die Systemzeit liefern
Aufrufparameter	Vgl. [NTPv4]
Rückgabe	Vgl. [NTPv4]
Vorbedingungen	MGM_LU_ONLINE=Enabled
Nachbedingungen	Der anfragende Client hat die korrekte Zeit geliefert bekommen.
Hinweise	Keine
Fehler	Der Aufruf schlägt fehl (bleibt unbeantwortet), wenn MGM_LU_ONLINE=Disabled

[<=]

4.2.5.6 Betriebsaspekte

TIP1-A_4792 - Explizites Anstoßen der Zeitsynchronisierung
Der Konnektor MUSS dem Administrator die Möglichkeit bieten, eine Synchronisation mit dem zentralen Zeitdienst explizit anzustoßen.
[<=]

TIP1-A_4793 - Konfigurierbarkeit des Konnektor NTP-Servers
Der Administrator MUSS die in TAB_KON_643 aufgelisteten Parameter über die Managementschnittstelle konfigurieren und die in TAB_KON_730 aufgelisteten Parameter ausschließlich einsehen können.

Tabelle 339: TAB_KON_643 Konfiguration des Konnektor NTP-Servers

ReferenzID	Belegung	Bedeutung
NTP_TIMEZONE	Zeitzone	Der Administrator MUSS die Zeitzone des Konnektors einstellen können. Default-Wert: Central European Time/Mitteleuropäische Zeit (CET/MEZ)
NTP_TIME	Zeit	Der Administrator MUSS die Zeit des Konnektors (NTP_TIME) über die Managementschnittstelle manuell einstellen können.

Tabelle 340: TAB_KON_730 Einsehbare Konfigurationsparameter des Konnektor NTP-Servers

ReferenzID	Belegung	Bedeutung
NTP_SERVER_ ADDR	IP-Adressen	Die Adressen des primären und sekundären Stratum-2-Zeitserver der zentralen TI-Plattform für die Synchronisation mit dem NTP-Server des Konnektors.

[<=]

TIP1-A_4794 - Warnung und Übergang in kritischen Betriebszustand bei nichterfolgter Zeitsynchronisierung
Befindet sich der Konnektor im Zustand EC_TIME_SYNC_PENDING_CRITICAL oder EC_Time_Difference_Intolerable, MUSS der Administrator eine Korrektur oder Bestätigung der Systemzeit vornehmen können. Anschließend MUSS der Konnektor wie nach einer erfolgreichen Zeitsynchronisation verfahren, d. h. der Tagezähler wird auf 0 zurückgesetzt.
[<=]

4.2.5.6.1 TUC_KON_352 Initialisierung Zeitdienst

TIP1-A_4795 - TUC_KON_352 „Initialisierung Zeitdienst"
Der Konnektor MUSS in der Bootup-Phase TUC_KON_352 "Initialisierung Zeitdienst" durchlaufen.

Tabelle 341: TAB_KON_644 – TUC_KON_352 „Initialisierung Zeitdienst"

Element	Beschreibung
Name	TUC_KON_352 „Initialisierung Zeitdienst"
Beschreibung	Der Konnektor muss zum Bootup den konnektoreigenen NTP-Server mit einem NTP-Server der zentralen TI-Plattform synchronisieren falls MGM_LU_ONLINE=Enabled.
Anwendungsumfeld	Synchronisierung der Systemzeit zur Startzeit
Eingangsanforderung	Keine
Auslöser	• Bootup • Event NETWORK/VPN_TI/UP
Vorbedingungen	Verbindung zum VPN-Konzentrator TI muss aufgebaut sein
Eingangsdaten	NTP-Server der zentralen TI-Plattform
Komponenten	Konnektor

Ausgangsdaten	Keine
Standardablauf	Falls MGM_LU_ONLINE=Enabled: • Durch eine DNS-Anfrage an den DNS-Forwarder zur Auflösung des SRV-RR mit dem Bezeichner "_ntp._udp.<DOMAIN_SRVZONE_TI>„ erhält der Konnektor Adressen der NTP-Server der zentralen TI-Plattform. • gemäß [NTPv4] • Falls keine Antwort erfolgt ist oder falls der Zeitserver nicht erreichbar ist, wird Fehler 4177 ausgelöst. Zur Feststellung werden die NTPv4 eigenen Timeoutwerte berücksichtigt.
Varianten/Alternativen	Keine
Fehlerfälle	4177: Der NTP-Server des Konnektors empfängt keine Systemzeit
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 342: TAB_KON_645 Fehlercodes TUC_KON_352 „Initialisierung Zeitdienst"

Fehlercode	ErrorType	Severity	Fehlertext
4177	Technical	Warning	Der NTP-Server des Konnektors konnte nicht synchronisiert werden.

[<=]

4.2.6 Namensdienst und Dienstlokalisierung

Innerhalb des Namensdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): keine Events vorhanden
- Konfigurationsparameter: „DNS_"

4.2.6.1 Funktionsmerkmalweite Aspekte

TIP1-A_4796 - Grundlagen des Namensdienstes

Der Konnektor MUSS einen Recursive Caching Nameserver zur Auflösung von DNS-Anfragen sowie einen autoritativen Nameserver zur Verwaltung der Zone „konlan." bereitstellen.
Der Caching-Nameserver des Konnektors MUSS für Clientsysteme aus dem lokalen Netzwerk (ANLW_LAN_NETWORK_SEGMENT oder ANLW_LEKTR_INTRANET_ROUTES) erreichbar sein.

Der Caching-Nameserver des Konnektors MUSS einen Timeout für die Bearbeitung von DNS-Abfragen beachten. Konnte eine DNS-Abfrage nicht durchgeführt werden, MUSS die Bearbeitung abgebrochen werden.
[<=]

TIP1-A_6480 - Resource Records der Zone konlan.
Der Konnektor MUSS in der Zone „konlan." die folgenden Resource Records bereitstellen:

- label: „konnektor.konlan.", ttl: <Time To Live>, class: IN, type: A, rdata: <LAN-seitige IP-Adresse des Konnektors>

Die in spitzen Klammern angegebenen Werte müssen implementierungs- und konfigurationsabhängig vergeben werden.
[<=]

TIP1-A_4797 - DNS-Forwards des DNS-Servers
Der DNS-Server des Konnektors MUSS die folgenden DNS-Forwards durchführen:

Tabelle 343: TAB_KON_687 DNS-Forwards des DNS-Servers

Domain	Forwarders	Bemerkungen
Namensraum TI, *.DNS_TOP_LEVEL_DOMAIN_TI	DNS_SERVERS_TI	DNS Forward Rule zur Auflösung aller DNS-Namen innerhalb des Namensraums der TI mit der Top Level Domain telematik (für die PU) und telematik-test (für die RU und TU).
Namensraum TI, Top Level Domain ti-wa (PU) und ti-wa-test (RU und TU).	DNS_SERVERS_TI	DNS Forward Rule zur Auflösung aller DNS-Namen innerhalb des Namensraums der TI mit der Top Level Domain ti-wa (für die PU) und ti-wa-test (für die RU und TU).
Namensraum angeschlossene Netze des Gesundheitswesens mit WANDA Basic (Domainnamen von angeschlossenen Netzen des Gesundheitswesens mit WANDA Basic gemäß Bestandsnetze.xml)	DNS_SERVERS_BESTANDSNETZE (Je Domainnamen eines angeschlossenen Netzes des Gesundheitswesens mit WANDA Basic alle zugehörigen DNS-Server IP-Adressen gemäß Bestandsnetze.xml)	Je angeschlossenes Netz des Gesundheitswesens mit WANDA Basic in ANLW_AKTIVE_BESTANDSNETZE wird eine DNS Forward Rule zur Auflösung von DNS-Namen innerhalb dieses Netzes verwendet.
Namensraum lokale Einsatzumgebung (DNS_DOMAIN_LEKTR)	DNS_SERVERS_LEKTR	DNS Forward Rule zur Auflösung aller DNS-Namen innerhalb der DNS-Domain DNS_DOMAIN_LEKTR

Namensraum Internet	DNS_ SERVERS_ SIS	Wenn der VPN-Tunnel SIS aktiv ist, muss eine Forward Rule für den Namensraum Internet über die DNS_SERVERS_SIS existieren.
Namensraum Internet	DNS_ SERVERS_ INT	Wenn der VPN-Tunnel SIS nicht aktiv ist, muss eine Forward Rule für den Namensraum Internet über die DNS_SERVERS_INT existieren
Lokale Zone „konlan."	autoritativer Nameserver des Konnektors	DNS Forward Rule zur Auflösung aller DNS-Namen innerhalb der Zone „konlan."

[<=]

TIP1-A_4798 - DNS Stub-Resolver
Der Stub-Resolver im Konnektor MUSS von allen internen Diensten zur Namensauflösung genutzt werden.
Der Stub-Resolver im Konnektor MUSS immer den Caching-Nameserver im Konnektor anfragen.
[<=]

TIP1-A_4799 - Aktualität der DNS-Vertrauensanker sicherstellen
Der Konnektor, der einen Caching Nameserver als Validating Resolver umsetzt, MUSS den DNSSEC-Vertrauensanker der TI aus dem Zertifikatspeicher in den Caching-Nameserver übernehmen, wenn ein Fehler bei der Validierung der Namensauflösung der TI aufgetreten ist.**[<=]**

4.2.6.2 Durch Ereignisse ausgelöste Reaktionen

Keine.

4.2.6.3 Interne TUCs, nicht durch Fachmodule nutzbar

Keine.

4.2.6.4 Interne TUCs, auch durch Fachmodule nutzbar

4.2.6.4.1 TUC_KON_361 „DNS-Namen auflösen"

TIP1-A_4801 - TUC_KON_361 „DNS-Namen auflösen"
Der Konnektor MUSS den technischen Use Case TUC_KON_361 „DNS-Namen auflösen" umsetzen.

Tabelle 344: TAB_KON_646 – TUC_KON_361 „DNS-Namen auflösen"

Element	Beschreibung

Name	TUC_KON_361 „DNS-Namen auflösen“
Beschreibung	Ein FQDN wird in ein oder mehrere IPs aufgelöst
Auslöser	interne Anfrage (Basisdienst oder Fachmodul)
Vorbedingungen	Die vom Konnektor zu verwendenden DNS-Server (DNS_SERVERS_INT, DNS_SERVERS_TI, DNS_SERVERS_SIS, DNS_SERVERS_BESTANDSNETZE) müssen konfiguriert sein.
Eingangsdaten	FQDN (Name, für den die IP-Adressen ermittelt werden sollen)
Komponenten	Konnektor
Ausgangsdaten	LIST_OF_IP_ADDRESSES
Standardablauf	1) Mit dem FQDN wird eine Anfrage an den Stub-Resolver des Konnektors (Typ A und AAAA) durchgeführt. Für alle ermittelten IPv4-Adressen und IPv6-Adressen werden als LIST_OF_IP_ADDRESSES zurückgeliefert. Da IPv6 nicht produktiv eingesetzt wird muss die aufrufende Instanz die IPv6-Adressen ignorieren. Falls keine IP-Adressen ermittelt werden konnten, wird eine leere Liste zurückgeliefert.
Varianten/Alternativen	Keine
Fehlerfälle	(→ 1) Timeout der Anfrage; Fehlercode 4179 (→ 1) DNS-Fehler; Fehlercode 4180
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 345: TAB_KON_647 Fehlercodes TUC_KON_361 „DNS Namen auflösen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4179	Technical	Error	„DNS: Anfrage wurde wegen Timeout abgebrochen.”

4180	Technical	Fatal	„DNS: Es ist ein Fehler bei der Namensauflösung aufgetreten“ Die Fehlerdetails sind gemäß DNS-Protokoll zu ergänzen.

[<=]

TIP1-A_4801-02 - ab PTV4: TUC_KON_361 „DNS-Namen auflösen“
Der Konnektor MUSS den technischen Use Case TUC_KON_361 „DNS-Namen auflösen“ umsetzen.

Tabelle 346: TAB_KON_646 – TUC_KON_361 „DNS-Namen auflösen“

Element	Beschreibung
Name	TUC_KON_361 „DNS-Namen auflösen“
Beschreibung	Ein FQDN wird in ein oder mehrere IPs aufgelöst
Auslöser	interne Anfrage (Basisdienst oder Fachmodul)
Vorbedingungen	Die vom Konnektor zu verwendenden DNS-Server (DNS_SERVERS_INT, DNS_SERVERS_TI, DNS_SERVERS_SIS, DNS_SERVERS_BESTANDSNETZE) müssen konfiguriert sein.
Eingangsdaten	FQDN (Name, für den die IP-Adressen ermittelt werden sollen)
Komponenten	Konnektor
Ausgangsdaten	LIST_OF_IP_ADDRESSES
Standardablauf	1) Mit dem FQDN wird eine Anfrage an den Stub-Resolver des Konnektors (Typ A und AAAA) durchgeführt. Für alle ermittelten IPv4-Adressen und IPv6-Adressen werden als LIST_OF_IP_ADDRESSES zurückgeliefert. Wird IPv6 nicht produktiv eingesetzt, muss die aufrufende Instanz die IPv6-Adressen ignorieren. Falls keine IP-Adressen ermittelt werden konnten, wird eine leere Liste zurückgeliefert.
Varianten/Alternativen	Keine
Fehlerfälle	(→ 1) Timeout der Anfrage; Fehlercode 4179 (→ 1) DNS-Fehler; Fehlercode 4180
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 347: TAB_KON_647 Fehlercodes TUC_KON_361 „DNS Namen auflösen“

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4179	Technical	Error	„DNS: Anfrage wurde wegen Timeout abgebrochen.”
4180	Technical	Fatal	„DNS: Es ist ein Fehler bei der Namensauflösung aufgetreten“ Die Fehlerdetails sind gemäß DNS-Protokoll zu ergänzen.

[<=]

4.2.6.4.2 TUC_KON_362 „Liste der Dienste abrufen“

TIP1-A_4802 - TUC_KON_362 „Liste der Dienste abrufen“
Der Konnektor MUSS den technischen Use Case TUC_KON_362 „Liste der Dienste abrufen“ umsetzen.

Tabelle 348: TAB_KON_648 – TUC_KON_362 „Liste der Dienste abrufen“

Element	Beschreibung
Name	TUC_KON_362 „Liste der Dienste abrufen“
Beschreibung	Ermittlung aller zu einer DNS-SD-Gruppe gehörenden DNS-Namen.
Auslöser	interne Anfrage (Basisdienst oder Fachmodul)
Vorbedingungen	Die vom Konnektor zu verwendenden DNS-Server müssen konfiguriert sein.
Eingangsdaten	FQDN des PTR Resource Records
Komponenten	Konnektor
Ausgangsdaten	LIST_OF_SRV_ENTITIES
Standardablauf	Mit dem FQDN wird eine Typ „PTR“ Anfrage an den Stub-Resolver des Konnektor gestellt.
Varianten/Alternativen	Keine
Fehlerfälle	(→ 1) Timeout der Anfrage; Fehlercode 4179 (→ 1) DNS-Fehler; Fehlercode 4180
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 349: TAB_KON_649 Fehlercodes TUC_KON_362 „Liste der Dienste abrufen“

Fehlercode	ErrorType	Severity	Fehlertext

Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4179	Technical	Error	„DNS: Anfrage wurde wegen Timeout abgebrochen.“
4180	Technical	Fatal	„DNS: Es ist ein Fehler bei der Namensauflösung aufgetreten“ Die Fehlerdetails sind gemäß [gemSpec_Net] zu ergänzen.

[<=]

4.2.6.4.3 TUC_KON_363 „Dienstdetails abrufen“

TIP1-A_4803 - TUC_KON_363 „Dienstdetails abrufen“
Der Konnektor MUSS den technischen Use Case TUC_KON_363 „Dienstdetails abrufen“ umsetzen.

Tabelle 350: TAB_KON_650 - TUC_KON_363 „Dienstdetails abrufen“

Element	Beschreibung
Name	TUC_KON_363 Dienstdetails abrufen
Beschreibung	Ermitteln aller DNS-SD-Details zu einem vollqualifizierten DNS-Namen.
Auslöser	interne Anfrage (Basisdienst oder Fachmodul)
Vorbedingungen	Die vom Konnektor zu verwendenden DNS-Server müssen konfiguriert sein.
Eingangsdaten	FQDN (der Name eines DNS-SD-Elements)
Komponenten	Konnektor
Ausgangsdaten	LIST_OF_SRV_ENTRIES LIST_OF_SRV_DETAILS
Standardablauf	1) Mit dem FQDN wird eine Typ-„SRV“-Anfrage an den Stub-Resolver des Konnektors gestellt. Die vom DNS-Server zurück gelieferten SRV-Einträge werden als LIST_OF_SRV_ENTRIES (bestehend aus TTL, Priority, Weight, Port, Target) zurückgeliefert. Wenn kein Eintrag gefunden werden konnte, wird eine leere Liste LIST_OF_SRV_ENTRIES zurückgeliefert. 2) Mit dem FQDN wird zusätzlich eine Typ-„TXT“-Anfrage an den Stub-Resolver des Konnektors gestellt. Wenn ein oder mehrere entsprechende Einträge gefunden werden konnten, werden diese in einer gemeinsamen Liste LIST_OF_SRV_DETAILS (bestehend aus TTL und TXT) zusammengefasst. Wenn kein Eintrag gefunden werden konnte, wird eine leere Liste LIST_OF_SRV_DETAILS zurückgeliefert. Falls keine FQDN ermittelt werden konnten, wird je

	eine leere Liste LIST_OF_SRV_ENTRIES und LIST_OF_SRV_DETAILS zurückgeliefert.
Varianten/Alternativen	Keine
Fehlerfälle	(→ 1-2) Timeout der Anfrage; Fehlercode 4179 (→ 1-2) DNS Fehler; Fehlercode 4180
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 351: TAB_KON_651 Fehlercodes TUC_KON_363 „Dienstdetails abrufen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4179	Technical	Error	„DNS: Anfrage wurde wegen Timeout abgebrochen."
4180	Technical	Fatal	„DNS: Es ist ein Fehler bei der Namensauflösung aufgetreten" Die Fehlerdetails sind gemäß [gemSpec_Net] zu ergänzen.

[<=]

4.2.6.5 Operationen an der Außenschnittstelle

TIP1-A_4804 - Basisanwendung Namensdienst

Der Konnektor MUSS für Clients eine Basisanwendung Namensdienst anbieten.

Tabelle 352: TAB_KON_652 Basisanwendung Namensdienst

Name	Namendienst	
Version	wird im Produktsteckbrief des Konnektors definiert	
Namensraum	Keiner	
Namensraum-Kürzel	Keiner	
Operationen	Name	Kurzbeschreibung
	GetIPAddress	Diese Operation ermöglicht die Auflösung von FQDNs in IP-Adressen
WSDL	Keines	
Schema	Keines	

[<=]

4.2.6.5.1 GetIPAddress

TIP1-A_5035 - Operation GetIPAddress
Der Namensdienst des Konnektors MUSS an der Client-Schnittstelle eine Operation GetIPAddress anbieten.

Tabelle 353: TAB_KON_653 Operation GetIPAddress

Name	GetIPAddress
Beschreibung	Diese Operation ermöglicht die Auflösung von FQDN in IP-Adressen. (DNS-Forwarder Abfrage ohne Cache)
Aufrufparameter	Address (FQDN) DNSSECValidation (Boolean)
Rückgabe	IPAddr (IPAddress) DNSSECValidated (Boolean)
Vorbedingungen	Der DNS-Server im Konnektor muss aktiv sein. Die Forward Nameserver (DNS_SERVERS_TI, DNS_SERVERS_SIS, DNS_SERVERS_BESTANDSNETZE) müssen konfiguriert sein.
Nachbedingungen	Keine
Standardablauf	Für Details zu DNS Namensauflösung wird auf [gemSpec_Net] verweisen.

[<=]

4.2.6.6 Betriebsaspekte

TIP1-A_5416 - Initialisierung „Namensdienst und Dienstlokalisierung“
Der Konnektor MUSS in der Bootup-Phase zur Initialisierung des Funktionsmerkmals „Namensdienst und Dienstlokalisierung“:

- den autoritativen Nameserver starten
- den Caching-Nameserver starten.

[<=]

TIP1-A_4805 - Konfigurationsparameter Namensdienst und Dienstlokalisierung
Der Administrator MUSS die in TAB_KON_654 aufgelisteten Parameter über die Managementschnittstelle konfigurieren und die in TAB_KON_731 aufgelisteten Parameter ausschließlich einsehen können.
Nach jeder Änderung MUSS sichergestellt werden, dass die Änderungen sofort am autoritativen bzw. am Caching-Nameserver zur Verfügung stehen.

Tabelle 354: TAB_KON_654 - Konfigurationsparameter Namensdienst

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion

DNS_SERVERS_INT	Liste von IP-Adressen der DNS-Server	Liste von DNS-Servern für das Transportnetz. Die IP-Adressen KÖNNEN auf einen öffentlich zugänglichen Adressbereich eingeschränkt sein.
DNS_DOMAIN_ VPN_ZUGD_INT	DNS Domainname	DNS-Domainname für die Service Discovery der VPN-Konzentratoren des VPN-Zugangsdienstes
DNS_SERVERS_ LEKTR	Liste von IP-Adressen der DNS-Server	Liste von DNS-Servern, die zur Namensauflösung von Namensräumen in der Einsatzumgebung verwendet werden. Der Administrator MUSS die Liste von DNS-Servern, die die DNS_DOMAIN_LEKTR auflösen, bearbeiten können. Die IP-Adressen der DNS-Server KÖNNEN auf den Adressbereich der ANLW_LAN_IP_ADDRESS eingeschränkt sein.
DNS_DOMAIN_ LEKTR	DNS Domainname	DNS Domainname, der von einem DNS-Server der Einsatzumgebung aufgelöst wird. Der Name DARF NICHT mit einem „.“ beginnen und nicht mit einem „.“ enden.
DNS_TA_CONFIG	Ist abhängig von der gewählten Umsetzung	Wenn der Konnektor als Validating Resolver für den Namensraum Internet implementiert ist gilt: Der Administrator MUSS die aktuellen DNSSEC Trustanchor für den Namensraum Internet auf geeignetem Weg in den Konnektor übernehmen können.

Tabelle 355: TAB_KON_731 Einsehbare Konfigurationsparameter Namensdienst

ReferenzID	Belegung	Bedeutung
DNS_SERVERS_TI	Liste von IP-Adressen der DNS-Server	Liste von DNS-Servern, die zur Namensauflösung des Namensraums der TI verwendet werden

DNS_SERVERS_SIS	Liste von IP-Adressen der DNS-Server	Liste von DNS-Servern, die zur Namensauflösung des Namensraums Internet bei Nutzung des SIS verwendet werden
DNS_SERVERS_ BESTANDSNETZE	Liste von IP-Adressen der DNS-Servern je Domäne je freigegebenem angeschlossenen Netz des Gesundheitswesens mit WANDA Basic	Liste von DNS-Servern je Domain eines dieser freigegebenen Netze.
DNS_TOP_LEVEL_ DOMAIN_TI	DNS Domainname	Top Level Domain des Namensraumes TI

[<=]

4.2.7 Optionale Verwendung von IPv6

Der Konnektor kann zusätzlich eine IPv6-Adresse an den Netzwerkschnittstellen zum Transportnetz implementieren. Entscheidet sich der Hersteller für den parallelen Einsatz von IPv4 und IPv6 (Dual-Stack-Mode), sind die nachfolgenden Anforderungen dieses Kapitels umzusetzen. Einhergehend mit der Entscheidung, IPv6 an diesem Interface zu konfigurieren, ist der spätere VPN-Tunnelaufbau zur TI und SIS über das IPv6 Interface möglich. Die durch den jeweiligen IPv6-Tunnel zu transportierenden IP-Pakete sind IPv4 adressierte Pakete.

A_17199 - IPv6 - Adressierung der Schnittstelle zum Internet (Option IPv6)
Der Konnektor MUSS bei Verwendung von IPv6 für den VPN-Tunnelaufbau zur TI und SIS auf geeignete Weise (z.B. DHCP vom IAG) mit einer IPv6-Adresse auf dem physikalischen Interface in Richtung Internet konfiguriert werden (Dual-Stack-Mode).**[<=]**

A_17200 - IPv6 - Fragmentierung der IKEv2-Nachrichten (Option IPv6)
Der Konnektor MUSS bei Verwendung von IPv6 für den VPN-Tunnelaufbau zur TI und SIS die Fragmentierung von IKEv2 Nachrichten gemäß [RFC7383] unterstützen.**[<=]**

A_17201 - IPv6 - Verhalten als IPv6 Router (Option IPv6)
Der Konnektor MUSS bei Verwendung von IPv6 für den VPN-Tunnelaufbau zur TI und SIS die notwendige Route für das Erreichen des Internets bereitstellen.**[<=]**

4.3 Konnektormanagement

Das Konnektormanagement dient ausschließlich Betriebsaspekten des Konnektors. Daher wird in diesem Kapitel weitestgehend auf die übliche Strukturierung nach TUCs (intern/für Fachmodule), Außenoperationen und Betriebsaspekten verzichtet. Lediglich der KSR-Client verwendet diese Kapitelstruktur.

Innerhalb des Konnektormanagements werden vorrangig folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „MGM"
- Konfigurationsparameter: „MGM_"

Eine Ausnahme hiervon bildet der Anteil der Software-Aktualisierung (KSR-Client). Dieser verwendet folgende Präfixe für Bezeichner:

- Events (Topic Ebene 1): „KSR"
- Konfigurationsparameter: „MGM_"

TIP1-A_4806 - Verpflichtende Managementschnittstelle
Der Konnektor MUSS LAN-seitig über eine Managementschnittstelle für Konfiguration und Diagnose verfügen.
Die Ausführung der Schnittstelle ist herstellerspezifisch, MUSS aber entweder als Konfigurations-Frontend im Sinne einer eigenständigen Client-Applikation oder als Web-Oberfläche ausgeprägt sein.
Wenn die Schnittstelle als Web-Oberfläche ausgeprägt ist, MUSS im Handbuch beschrieben sein, wo angegeben ist, welche Browser-Versionen für welche Betriebssysteme unterstützt werden (bspw. im Handbuch selbst oder über einen Link auf eine Web-Seite des Herstellers), und wo diese als installierbares Softwarepaket oder direkt ausführbare Datei bezogen werden können.
Die Verbindung zur Managementschnittstelle MUSS zur Sicherung der Vertraulichkeit, Integrität und Authentizität durch Nutzung eines kryptographischen Verfahrens gemäß [gemSpec_Krypt] abgesichert werden, falls die Sicherheit der übertragenen Daten nicht auf andere Weise erreicht wird. Die Absicherung der Daten kann z. B. durch Nutzung von TLS unter Berücksichtigung der in [gemSpec_Krypt] angegebenen Algorithmen und Schlüssellängen geschehen.
Die Managementschnittstelle MUSS in thematisch gegliederte Konfigurationsbereiche unterteilt sein. Die konkrete Gliederung selbst ist herstellerspezifisch.
Die Managementschnittstelle KANN einen Managementbereich aufweisen, der nur für autorisierte Techniker des Herstellers zugänglich ist. Ein Zugriff auf diesen Bereich MUSS durch eine eigene Authentisierungsfunktion geschützt werden (z. B. durch Passwortschutz).
[<=]

Die über die Managementschnittstelle zu erreichenden und zu verändernden Inhalte werden erhoben in:

- diesem Kapitel
- in allen Betriebsaspektekapiteln der Funktionsmerkmale, sowie der Übergreifenden Festlegungen
- den Fachmodulspezifikationen der Fachanwendungen (siehe Kapitel 4.3.4).
- Den übergreifenden Spezifikationen [gemSpec_Net] und [gemSpec_PKI]

Eine Ergänzung um weitere, herstellerspezifische Konfigurationsinhalte ist möglich.

TIP1-A_5661 - Automatisierung Managementschnittstelle
Der Konnektor MUSS für die Automatisierung von Konnektor-Tests alle Funktionen, die über die Managementschnittstelle bereitgestellt werden, über eine LAN-seitige Schnittstelle ohne graphische Benutzerführung bereitstellen.
Der Konnektorhersteller MUSS eine Dokumentation der Schnittstelle bereitstellen, welche die Nutzung so beschreibt, dass die Schnittstelle von der gematik in vollem Umfang genutzt werden kann. Die Dokumentation MUSS der gematik im Regelfall zwei Wochen vor Einreichung des Zulassungsobjekts bereitgestellt werden. Von diesem Regelfall KANN

in Abstimmung mit der gematik abgewichen werden.
Die Schnittstelle SOLL mittels JSON [RFC7159] bereitgestellt werden. Wenn die Bereitstellung nicht mittels JSON erfolgt, MUSS sie über eine vergleichbare Technologie erfolgen.
Der Zugriff auf die Schnittstelle MUSS in RU/TU erlaubt sein. Falls der Zugriff in der PU erlaubt ist, MUSS er dort ebenso wie die Managementschnittstelle abgesichert sein:

- Die Verbindung zu dieser Schnittstelle MUSS zur Sicherung der Vertraulichkeit, Integrität und Authentizität durch Nutzung eines kryptographischen Verfahrens gemäß [gemSpec_Krypt] abgesichert werden, falls die Sicherheit der übertragenen Daten nicht auf andere Weise erreicht wird. Die Absicherung der Daten kann z. B. durch Nutzung von TLS unter Berücksichtigung der in [gemSpec_Krypt] angegebenen Algorithmen und Schlüssellängen geschehen.
- Der Konnektor MUSS die Schnittstelle mittels Benutzername und Passwort oder einem mindestens gleich starken Mechanismus vor unberechtigtem Zugang schützen.

Ansonsten DARF der Zugriff in der PU NICHT möglich sein.
[<=]

TIP1-A_4807 - Mandantenübergreifende Managementschnittstelle
Das Management des Konnektors MUSS über die Managementschnittstelle mandantenübergreifend erfolgen. Dies bedeutet insbesondere, dass ein Administrator (gemäß seiner Zugriffsberechtigungen) in einer Management-Session alle Einstellungen einsehen und verändern können MUSS, egal welchem Mandanten diese Werte zugeordnet sind.
[<=]

TIP1-A_5658 - Konnektor, rollenspezifische Endpunkte der Managementschnittstelle
Der Konnektor MUSS die Managementschnittstelle mit zwei getrennten Endpunkten implementieren. Der Konnektor MUSS sicherstellen, dass auf den einen Endpunkt nur Nutzer mit der Rolle Lokaler-Administrator oder Super-Administrator zugreifen können, und auf den anderen Endpunkt nur Nutzer mit der Rolle Remote-Administrator.
[<=]

TIP1-A_5005 - Protokollierung in der Managementschnittstelle
Jede Änderung, die ein Administrator vornimmt, MUSS protokolliert werden durch
TUC_KON_271 „Schreibe Protokolleintrag“ {
topic=„MGM/ADMINCHANGES“;
eventType=Op;
severity=Info;
parameters =(„User=$AdminUsername,
RefID=$ReferenzID,
NewVal=$NeuEingestellterWert“)}
Der hier geforderte Logging-Level gilt, wenn nicht an anderer Stelle eine abweichende Regelung spezifiziert ist.
Wenn die Änderung über ein Remote-Management-System durchgeführt wird, ohne dass ein Remote-Administrator im Konnektor konfiguriert ist, so MUSS als User eine Referenz auf das Remote-Management-System verwendet werden.
Passwörter DÜRFEN NICHT in den Protokolleinträgen geschrieben werden.

[<=]

4.3.1 Zugang und Benutzerverwaltung des Konnektormanagements

Der Konnektor verfügt über keine Verwaltung der fachlichen Nutzer, wohl aber über eine Verwaltung der Nutzer, die in der Rolle eines Administrators den Konnektor konfigurieren und die Protokolle einsehen dürfen. Dabei werden drei Administrator-Rollen unterschieden:

1. Lokaler-Administrator: zur Konfiguration des Konnektors über die lokale Managementschnittstelle
2. Remote-Administrator: zur Konfiguration des Konnektors über die remote Managementschnittstelle.
3. Super-Administrator: zur Verwaltung von Benutzerkonten und zur Konfiguration des Konnektors über die lokale Managementschnittstelle

TIP1-A_4808 - Zugangsschutz der Managementschnittstelle
Der Konnektor MUSS sicherstellen, dass die Managementschnittstelle vor unberechtigtem Zugang geschützt ist. Die Managementschnittstelle MUSS durch eine Kombination aus Benutzername und Passwort oder einen mindestens gleich starken Mechanismus vor unberechtigtem Zugang geschützt sein.
Für die Erstellung und Verarbeitung von Passwörtern der Managementschnittstelle MÜSSEN die Empfehlungen der Grundschutz-Kataloge des BSI beachtet werden (siehe Maßnahme „M 2.11 Regelung des Passwortgebrauchs“ in [BSI_GK]).

Für die Passworterstellung MUSS der Konnektor mindestens folgende Aspekte berücksichtigen:

- dem Benutzer muss es möglich sein, die Zeichen eines Passworts aus den Zeichenklassen Großbuchstaben, Kleinbuchstaben, Sonderzeichen und Ziffern zu wählen. Ein Passwort muss Zeichen aus mindestens drei dieser Zeichenklassen enthalten.
- ein Passwort muss mindestens 8 Zeichen lang sein
- ein Passwort darf nicht die zugehörige Benutzerkennung enthalten (weder vorwärts noch rückwärts, bei Vergleich unter Ignorierung der Groß- und Kleinschreibung)
- die Wiederholung alter Passwörter beim Passwortwechsel durch den Benutzer selbst muss vom Konnektor verhindert werden (Passworthistorie). Dazu muss der Konnektor mindestens die letzten drei Passwörter eines Benutzers bei der Passwortneuvergabe erkennen und als neues Passwort ablehnen.

Für die Passwortverarbeitung MUSS der Konnektor mindestens folgende Aspekte berücksichtigen:

- für die Erstanmeldung neuer Benutzer müssen Einmalpasswörter vergeben werden, also Passwörter, die nach einmaligem Gebrauch gewechselt werden müssen. Gleiches gilt, wenn ein Passwort eines Benutzers vom Super-Admin zurückgesetzt wird.
- jeder Benutzer muss sein eigenes Passwort jederzeit ändern können
- bei der Eingabe darf das Passwort nicht im Klartext auf dem Bildschirm angezeigt werden
- die Passwörter müssen im Konnektor zugriffssicher gespeichert werden

- der Konnektor muss nach einem durch den Super-Admin konfigurierbaren Zeitraum (Voreinstellung: 120 Tage) einen Passwortwechsel beim nächsten Login initiieren
- erfolglose Anmeldeversuche müssen mit einer kurzen Fehlermeldung ohne Angabe von näheren Einzelheiten abgelehnt werden. Insbesondere darf bei erfolglosen Anmeldeversuchen nicht erkennbar sein, ob der eingegebene Benutzername oder das eingegebene Passwort (oder beides) falsch ist.
- Nach einer Fehleingabe des Passworts muss eine Verzögerung bis zur nächsten Eingabemöglichkeit des Passworts für dieselbe Benutzerkennung erfolgen. Die Verzögerung soll 3 Sekunden betragen.

[<=]

Näheres hierzu regeln die Schutzprofile des Konnektors.

TIP1-A_4810 - Benutzerverwaltung der Managementschnittstelle
Der Konnektor MUSS eine Benutzerverwaltung für die Managementschnittstelle enthalten, in der anmeldeberechtigte Administratoren-Benutzer definiert werden können. Die Benutzerverwaltung MUSS die Administrator-Rollen Lokaler-Administrator, Remote-Administrator und Super-Administrator unterstützen.
Den Administrator-Rollen MÜSSEN folgende Rechte zugewiesen sein:

- Lokaler-Administrator:
  - ausschließlicher Zugriff über lokalen Endpunkt der Managementschnittstelle
  - Verwaltung aller Konfigurationsdaten und Durchführung aller Administratoraktionen mit Ausnahme von:
    - Benutzerverwaltung gemäß Tabelle TAB_KON_655
- Remote-Administrator:
  - ausschließlicher Zugriff über remote-Endpunkt der Managementschnittstelle
  - Verwaltung aller Konfigurationsdaten und Durchführung aller Administratoraktionen mit Ausnahme von:
    - Benutzerverwaltung gemäß Tabelle TAB_KON_655
    - Konfigurationseinstellungen und Administratoraktionen gemäß Tabelle TAB_KON_851
- Super-Administrator:
  - ausschließlicher Zugriff über lokalen Endpunkt der Managementschnittstelle
  - Benutzerverwaltung gemäß Tabelle TAB_KON_655
  - Verwaltung aller Konfigurationsdaten und Durchführung aller Administratoraktionen

Tabelle 356: TAB_KON_655 Konfigurationen der Benutzerverwaltung (Super-Administrator)

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
MGM_USER_LIST	Liste von Benutzernamen und deren	Liste von Benutzern und deren Kontaktdaten. Benutzerkonten MÜSSEN angelegt, geändert und gelöscht werden können.

	Kontaktdaten	Das Passwort eines Benutzerkontos MUSS neu gesetzt werden können.
MGM_ADMIN_RIGHTS	Liste von Zugriffsrechten eines Benutzers	i. Eindeutige Zuordnung eines Benutzerkontos zu einer Rolle. Die Benutzerverwaltung MUSS sicherstellen, dass zu jeder Zeit mindestens ein Benutzerkonto mit der Rolle Super-Administrator vorhanden ist. Gewähren/Entziehen von Rechten für Benutzerkonten: ii. Zugriffsrechte bezüglich der Konfigurationsbereiche. iii. Recht zum Aufbau einer Remote-Management-Session und/oder zur Konfiguration des Remote-Management gemäß TAB_KON_663 (USER_INIT_REMOTESESSION). iv. Recht für einen Werksreset (USER_RESET_PERMISSION)

Die Benutzerverwaltung MUSS es jedem Benutzer ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_656 vorzunehmen:

Tabelle 357: TAB_KON_656 Konfigurationen der Benutzerverwaltung

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
MGM_USER_INFO	Kontaktdaten	Der angemeldete Benutzer MUSS seine Kontaktdaten ändern können. Der Benutzername DARF NICHT änderbar sein.

[<=]

4.3.2 Konnektorname und Versionsinformationen

TIP1-A_4811 - Festlegung des Konnektornamens
Der Konnektor MUSS die Konfiguration und Nutzung eines sprechenden Konnektornamens unterstützen, der identisch zum Hostnamen des Konnektors ist. Der Konnektorname MUSS dauerhaft an der Managementschnittstelle angezeigt werden.
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_657 vorzunehmen:

Tabelle 358: TAB_KON_657 Konfigurationsparameter des Konnektornamens

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion

MGM_KONN_ HOSTNAME	12 Zeichen	Der Konnektorname MUSS folgende Anforderungen erfüllen (in Anlehnung an die Definition eines „Labels" in [RFC1034]): • Verwendung der Buchstaben „A bis Z" und „a bis z", • Verwendung der Ziffern „0 bis 9", • als Sonderzeichen „–" (Minus), sowie • eine maximale Länge von 12 Zeichen, Die Verwendung weiterer Sonderzeichen sowie des Leerzeichens DARF NICHT möglich sein.

Optional KANN ein Hersteller zusätzlich zum Konnektor- bzw. Hostnamen die Konfiguration eines DNS-Suffixes vorsehen. Der DNS-Suffix DARF NICHT Bestandteil des Konnektornamens sein.
[<=]

TIP1-A_4812 - Anzeige der Versionsinformationen (Selbstauskunft)
Der Administrator MUSS die Versionsinformationen des Konnektors einsehen können. Dabei MÜSSEN alle über ProductInformation.xsd definierten Elemente verständlich angezeigt werden.
Ferner MUSS der Administrator dabei die aktuelle Firmware-Gruppenversion des Konnektors einsehen können.
[<=]

A_18929 - Sichtbarkeit der ECC-Vorbereitung an der Managementschnittstelle
Der Hersteller MUSS die ECC-Vorbereitung der gSMC-K durch die Bezeichnung „ECC-Vorbereitet" zusammen mit den Versionsinformationen des Konnektors an der Managementschnittstelle sichtbar machen.
[<=]

TIP1-A_7255 - Anzeige von Fachmodulversionen
Der Administrator MUSS die Versionen der in der Firmware des Konnektors enthaltenen Fachmodule einsehen können.
[<=]

Fachmodulversionsinformationen sind nicht Bestandteil der Selbstauskunft gemäß ProductInformation.xsd.

4.3.3 Konfigurationsdaten: Persistieren sowie Export-Import

TIP1-A_4813 - Persistieren der Konfigurationsdaten
Der Konnektor MUSS die Konfigurationsdaten nach Änderung persistieren. Dabei MÜSSEN Integrität, Authentizität und Vertraulichkeit der Konfigurationsdaten gewährt sein. Der Mechanismus hierfür ist herstellerspezifisch.
Der Konnektor MUSS sicherstellen, dass immer ein integerer Konfigurationssatz persistiert ist.
Bei der Konnektorinitialisierung MÜSSEN die persistierten Konfigurationsdaten eingelesen werden.
Die Verpflichtung zur Persistierung gilt für alle innerhalb der Konnektor- und Fachmodul-Spezifikationen erhobenen Konfigurationsdaten.
[<=]

TIP1-A_4814 - Export- Import von Konfigurationsdaten
Der Administrator MUSS die gesamten Konfigurationsdaten des Konnektors ex- und importieren können. Dazu gehören die Konfigurationsparameter des Konnektors, die

persistenten Daten wie im Informationsmodell des Konnektors (Tabelle TAB_KON_507 Informationsmodell Entitäten) definiert und die Pairing Informationen der Kartenterminals.
Die Konfigurationsdaten des Anwendungs- und Netzkonnektors KÖNNEN gemeinsam oder getrennt exportiert bzw. importiert werden. Das Format der Konfigurationsdaten ist herstellerspezifisch.
Auf hardwareseitig baugleichen Geräten:

- MUSS der Import von Konfigurationsdateien möglich sein, die unter der gleichen oder einer früheren Firmwareversion exportiert wurden
- SOLL der Import von Konfigurationsdateien möglich sein, die unter einer neueren Firmwareversion exportiert wurd

Der Import von Konfigurationsdateien, die von einem Konnektor mit anderer Hardwareversion exportiert wurden, KANN ermöglicht werden.

(für Fachmodule siehe Kapitel 4.3.4)
Der Konnektor MUSS sicherstellen, dass der Exportvorgang nur von einem am Konnektor angemeldeten User mit mindestens der Rolle Administrator ausgelöst werden kann.
Der Konnektor MUSS sicherstellen, dass der Importvorgang nur von einem am Konnektor angemeldeten User mit der Rolle Super-Administrator ausgelöst werden kann.
Sowohl Ex- als auch Import MÜSSEN protokolliert werden durch TUC_KON_271 „Schreibe Protokolleintrag“ {
topic = „MGM/CONFIG_EXIMPORT“;
eventType = Op;
severity = Info;
parameters = („User=$AdminUsername,
Mode=[Export/Import]“)}.

[<=]

A_19738 - Optionaler Import von Konfigurationsdaten durch lokalen Administrator
Der Konnektor KANN einem am Konnektor angemeldeten User mit der Rolle Lokaler-Administrator erlauben, den Importvorgang von Konfigurationsdateien auszulösen, wenn in den Konfigurationsdaten keine Benutzerdaten gemäß Tabelle TAB_KON_655 enthalten sind.**[<=]**

Nähere Vorgaben zum Ablauf des Imports der Kartenterminalinformationen finden sich im Kapitel 4.1.4.6.3.

TIP1-A_4815 - Export: Schutz der Integrität, Authentizität und Nichtabstreitbarkeit
Die **Integrität, Authentizität und Nichtabstreitbarkeit** der exportierten Daten MUSS sichergestellt werden. Dies MUSS durch eine Signatur mit der OSIG-Identität der SM-B oder mit einem herstellerspezifischen Schlüsselpaar realisiert werden. In die zu signierenden Daten MUSS eine Zeitangabe zum Signaturzeitpunkt integriert werden. Beim Import MUSS die Signatur vor der Übernahme der Daten erfolgreich verifiziert worden sein. Im Laufe des Importvorgangs MUSS dem Administrator das zur Signatur zugehörige Zertifikat (oder der herstellerspezifische öffentliche Schlüssel) sowie die Zeitangabe zum Signaturzeitpunkt der exportierten Konfiguration angezeigt werden, und der Administrator MUSS explizit bestätigen, dass er die zu dem angezeigten Zeitpunkt gehörige Konfiguration importieren will.
Wird die SM-B zur Signatur eingesetzt, so MUSS die Prüfung des genutzten Signaturzertifikats anhand von TUC_KON_037 erfolgen. Das Zertifikat der OSIG-Identität, mit dem die Daten signiert wurden, MUSS zusammen mit den exportierten

Daten gespeichert werden, um eine Verifikation der Signatur auf neuen Konnektoren auch ohne Zugriff auf die entsprechende SM-B zu ermöglichen.
Da Konfigurationsdaten mit einem Schutzbedarf von mindestens „Hoch" für Authentizität und Nichtabstreitbarkeit exportiert werden (z. B. Pairing-Geheimnisse (ShS.KT.AUT) der Kartenterminals), MUSS durch geeignete Maßnahmen sichergestellt werden, dass der Zugriff auf die Daten auf eine natürliche Person rückführbar ist. Dies kann organisatorisch (durch Einträge des Administrators in ein Betriebsführungs-Handbuch beim Nutzer) technisch (durch eine personenbezogene Administratorenverwaltung) oder äquivalent herstellerspezifisch erreicht werden.
[<=]

TIP1-A_4816 - Export: Schutz der Vertraulichkeit
Zum Schutz der **Vertraulichkeit** der exportierten Daten MÜSSEN die Daten vor dem Export verschlüsselt werden. Dies kann durch asymmetrische oder symmetrische Verschlüsselungsverfahren nach [gemSpec_Krypt] realisiert werden.
Wird ein rein symmetrisches Verfahren eingesetzt, so MUSS als Mindestanforderung eine Passphrase einer Mindestlänge von 16 Zeichen (Groß- und Kleinbuchstaben, Ziffern und Sonderzeichen) zur Verschlüsselung der Daten eingesetzt werden. Diese Passphrase MUSS dabei vom Konnektor zufällig generiert werden und aus einer Kombination von Buchstaben und Ziffern bestehen. Diese Passphrase MUSS dem Administrator anschließend angezeigt werden.
[<=]

4.3.4 Administration von Fachmodulen

Die Konfiguration von Fachmodulen ist innerhalb der Managementschnittstelle des Konnektors von der Konfiguration der Plattformanteile des Konnektors logisch entkoppelt. Die Festlegungen, welche Konfigurationsparameter und welche zusätzlichen administrativen Funktionen für ein Fachmodul benötigt werden, werden in den jeweiligen Fachmodulspezifikationen getroffen. Der Konnektor muss aber für jedes Fachmodul hinsichtlich der Administrierbarkeit des Fachmoduls die folgende Basisfunktionalität zur Verfügung stellen:

TIP1-A_4818 - Konfigurieren von Fachmodulen
Neben den Konfigurationsbereichen der Plattformanteile des Konnektors, MUSS die Managementschnittstelle auch die Konfiguration der im Konnektor enthaltenen Fachmodule unterstützen.
Ein Administrator MUSS die in den Fachmodulspezifikationen enthaltenen Konfigurationsparameter ändern und die dort definierten Informationen einsehen können.
Der Konnektor MUSS die Konfigurationsdaten von Fachmodulen nach deren Änderung persistieren, sowie bei einem Neustart eines Fachmoduls die Fachmodul-Konfigurationsdaten vor der Initialisierung des Fachmoduls einlesen.
Die persistierten Fachmodulkonfigurationsdaten MÜSSEN ebenso wie die plattformeigenen Konfigurationsdaten hinsichtlich ihrer Integrität und Authentizität sowie ihrer Vertraulichkeit geschützt werden.
Der Ex- und Import von Fachmodulkonfigurationen MUSS äquivalent zum Ex- und Import der Plattformanteile für den Administrator möglich sein (siehe 4.3.3). Die Konfigurationsdaten der Fachmodule KÖNNEN dabei in der Gesamt Export-Datei des Konnektors enthalten sein oder separat exportiert und importiert werden.
[<=]

TIP1-A_5484 - Persistente Speicherung von Konfigurationsdaten der Fachmodule
Der Konnektor MUSS den Fachmodulen die Möglichkeit bereitstellen, die in den Fachmodulspezifikationen gekennzeichneten Konfigurationsdaten persistent zu speichern,

auszulesen und zu löschen. Je Fachmodul muss ein exklusiv durch das Fachmodul nutzbarer Speicherbereich verwendet werden.
Namenskonvention zur Kennzeichnung der Konfigurationsdaten der Fachmodule für persistent zu speichernde Daten:
FM_<fmName>_<fmDataName>

Tabelle 359: TAB_KON_833 Bezeichner für persistente Konfigurationsdaten für Fachmodule

Bezeichner	Bedeutung
FM	fester Namensbestandteil zur Kennzeichnung von persistenten fachmodulspezifischen Konfigurationsdaten
_	Trennzeichen
<fmName>	Name des Fachmoduls (innerhalb eines Fachmoduls konstanter Bezeichner)
_	Trennzeichen
<fmDataName>	Name der persistent zu speichernden fachmodulspezifischen Konfigurationsdaten

[<=]

4.3.5 Neustart und Werksreset

TIP1-A_4819 - Auslösen eines Konnektorneustarts
Der Administrator MUSS einen Neustart des Konnektors auslösen können.
[<=]

TIP1-A_4820 - Werksreset des Konnektors
Ein Administrator mit USER_RESET_PERMISSION MUSS einen Werksreset des Konnektors auslösen können.
Zur Durchführung des Werksreset MUSS der Administrator nach Funktionsaufruf per Sicherheitsabfrage zur Bestätigung des Werksresets aufgefordert werden. Nach bestätigter Sicherheitsabfrage MUSS der Konnektor die gesamte Konfiguration des Konnektors und alle internen Speicher, mit Ausnahme des aktuellen Vertrauensraums sowie der Sicherheitsprotokolle und der installierten Firmware, auf den Auslieferungszustand zurücksetzen. Die in CERT_IMPORTED_CA_LIST enthaltenen Zertifikate MÜSSEN aus dem aktuellen Vertrauensraum gelöscht werden.
Die Durchführung des Werksresets MUSS protokolliert werden durch TUC_KON_271 „Schreibe Protokolleintrag“ {
topic = „MGM/FACTORYSETTINGS“;
eventType = Op;
severity = Info;
parameters = „User=$AdminUsername“}.
Dieser Protokolleintrag DARF NICHT durch einen erfolgreichen Werksreset verloren gehen.
Der Hersteller MUSS ferner einen alternativen, herstellerspezifischen Weg zum Auslösen des Werksresets vorsehen, welcher die Arbeitsabläufe beim Nutzer nur minimal unterbricht. Auch für diesen zusätzlichen Weg MUSS zuvor eine Authentisierung durch eine Kombination aus Benutzername und Passwort oder einem mindestens gleich starken Mechanismus erfolgen. Der Mechanismus MUSS auch dann funktionieren, wenn sich keiner der in der Benutzerverwaltung definierten Administratoren mehr erfolgreichen

anmelden kann.

[<=]

A_21743 - Laufzeitverlängerung gSMC-K: Erneuerte Zertifikate nach Werksreset verwenden
Der Konnektor, dessen gSMC-K-Zertifikate erneuert wurden, MUSS auch nach einem Werksreset die erneuerten Zertifikate verwenden.**[<=]**

4.3.6 Leistungsumfänge und Standalone-Szenarios

Obgleich der Konnektor in seinem Auslieferungszustand alle Leistungsmerkmale aufweisen muss, die gemäß Produkttypssteckbrief gefordert werden, so soll es dem Administrator doch ermöglicht werden grundsätzliche Leistungsumfänge gezielt deaktivieren zu können, um den Konnektor so besser in die organisatorische/technische Struktur der Betriebsstätte eingliedern zu können.

TIP1-A_4821-02 - Aktivieren/Deaktivieren von Leistungsumfängen
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_658 vorzunehmen:

Tabelle 360: TAB_KON_658 Aktivieren/Deaktivieren von Leistungsumfängen

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion

MGM_LU_ONLINE	Enabled/ Disabled	Der Administrator MUSS den „Leistungsumfang Online" aktivieren und deaktivieren können.
		Bei Veränderung MUSS TUC_KON_256 gerufen werden {
		topic = „MGM/LU_CHANGED/LU_ONLINE";
		eventType = Op;
		severity = Info;
		parameters = „Active=$MGM_LU_ONLINE"}

MGM_LU_SAK	Enabled/ Disabled	Der Administrator MUSS den „Leistungsumfang Signaturanwendungskomponente" aktivieren und deaktivieren können.
		Default-Wert: Enabled
		Bei Veränderung MUSS TUC_KON_256 gerufen werden {
		topic = „MGM/LU_CHANGED/LU_SAK";
		eventType = Op;
		severity = Info;
		parameters = „Active=$MGM_LU_SAK"}

[<=]

Der Konfigurationsparameter MGM_LU_SAK wirkt hauptsächlich in dem Funktionsmerkmal „Signaturdienst" (siehe Kapitel 4.1.8).

Ist MGM_LU_ONLINE Disabled, so baut der Konnektor grundsätzlich keine Online-Verbindungen auf (weder zur TI, noch zum SIS). Der Parameter wirkt hauptsächlich in den Funktionsmerkmalen:

- „Zertifikatsdienst“ (Kapitel 4.1.9)
- „TLS-Dienst“ (Kapitel 4.1.11)
- „Anbindung LAN/WAN“ (Kapitel 4.2.1)
- „VPN-Client“ (Kapitel 4.2.4)
- „Zeitdienst“ (Kapitel 4.2.5)
- „Software-Aktualisierungsdienst (KSR-Client)“ (Kapitel 4.3.9)
- „LDAP-Proxy" (Kapitel 4.1.12)

Ob es sich bei einem Konnektor um den losgelöst (stand alone) vom Netz der Einsatzumgebung betriebenen handelt, also einen Konnektor, auf welchen kein Clientsystem zugreift, muss diesem mitgeteilt werden:

TIP1-A_4822 - Konnektor Standalone einsetzen
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_659 vorzunehmen:

Tabelle 361: TAB_KON_659 Konnektor Standalone einsetzen

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion

MGM_STANDALONE_KON	Enabled/ Disabled	Der Administrator MUSS den Konnektor als alleinstehend konfigurieren können.
		Default-Wert: Disabled
		Bei Veränderung MUSS TUC_KON_256 gerufen werden {
		topic = „MGM/STANDALONE_CHANGED“;
		eventType = Op;
		severity = Info;
		parameters = „Active=$MGM_STANDALONE_KON“}

[<=]

Das Setzen von MGM_STANDALONE_KON auf Enabled dient dem Konnektor als Anzeige, dass dieser ohne angeschlossenes Clientsystem (Primärsystem) betrieben wird. Diese Information kann seitens der Fachmodule verwendet werden, damit diese sich im Standalone-Fall anders als im Normalfall verhalten.

4.3.7 Online-Anbindung verwalten

Um Zugang zur TI erlangen zu können, muss der Betriebsstättenverantwortliche einen Vertrag mit einem Zugangsdienstprovider (ZGDP) abgeschlossen haben. Von diesem erhält er eine ContractID. Der Administrator muss den Konnektor (genauer das NK-Zertifikat C.NK.VPN) mit dieser Information unter Nutzung einer SM-B über den Registrierungsdienst des ZGDP bei diesem freischalten.

Die Berechtigung zur Einwahl in die TI ist von der Gültigkeit der **beiden** bei der Freischaltung übermittelten Zertifikate abhängig (C.NK.VPN und C.HCI.OSIG). Die Berechtigung zur Einwahl in die TI wird verweigert, bzw. eine bestehende Verbindung zur TI wird beendet, wenn ein Zertifikat abgelaufen oder gesperrt ist. Aus diesem Grund muss der Administrator vor Ablauf eines der beiden Zertifikate eine wiederholte Registrierung mit neuem Netzkonnektorzertifikat bzw. neuer SM-B beim ZGDP durchführen. (Hinweis: neue NK-Zertifikate werden erst mit Etablierung der Nachladefunktionalität für gSMC-K verfügbar sein.)

Soll ein Konnektor außer Betrieb genommen werden oder wird der Vertrag mit einem ZGDP gekündigt, muss der Administrator den Konnektor über den Registrierungsdienst des ZGDP abmelden.

TIP1-A_4824 - Freischaltdaten des Konnektors bearbeiten
Der Administrator MUSS die in TAB_KON_661 aufgelisteten Parameter über die Managementschnittstelle konfigurieren und die in TAB_KON_732 aufgelisteten Parameter ausschließlich einsehen können.

Tabelle 362: TAB_KON_661 Konfigurationsparameter der Konnektorfreischaltung

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
MGM_ZGDP_CONTRACTID	String	Der Administrator MUSS die vom Zugangsdienstprovider für die Freischaltung des Konnektors erhaltene ContractID eingeben können.
MGM_ZGDP_SMCB	ICCSN	Der Administrator MUSS die zur Freischaltung zu verwendende SM-B aus der Liste der verwalteten SM-Bs auswählen können.

Tabelle 363: TAB_KON_732 Einsehbare Konfigurationsparameter der Konnektorfreischaltung

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
MGM_ZGDP_REGSERVER	URI	URI des Registrierungsservers des Zugangsdienstproviders

Den Zustand der Freischaltung verwaltet der Konnektor gemäß Tabelle TAB_KON_662 Zustandswerte der Konnektorfreischaltung.
Im Auslieferungszustand MUSS MGM_TI_ACCESS_GRANTED=Disabled belegt sein.

Tabelle 364: TAB_KON_662 Zustandswerte der Konnektorfreischaltung

ReferenzID	Belegung	Zustandswerte
MGM_TI_ACCESS_GRANTED	Enabled/ Disabled	Status der Freischaltung des Konnektors: - Enabled: Konnektor wurde erfolgreich beim Zugangsdienstprovider freigeschaltet - Disabled: Freischaltung noch nicht erfolgt

[<=]

TIP1-A_4825 - Konnektor zur Nutzung (wiederholt) freischalten
Der Administrator MUSS den Konnektor über folgenden Mechanismus zur Nutzung freischalten bzw. eine vorhandene Freischaltung mit einer neuen SM-B aktualisieren können (Voraussetzung ist eine korrekte Konfiguration aller für einen Online-Zugang erforderlicher Parameter):

1. Der Konnektor MUSS eine registerKonnektorRequest-Struktur gemäß ProvisioningService.xsd [gemSpec_VPN_ZugD] erstellen und mit den entsprechenden Parametern befüllen (aktuelles Datum/Uhrzeit, C.NK.VPN, MGM_ZGDP_CONTRACTID). Der Konnektor MUSS die Request-Nachricht mittels der ausgewählten SM-B (ID.HCI.OSIG von `MGM_ZGDP_SMCB`) im Element registerKonnektorRequest/Signature signieren und das SM-B-Zertifikat im Element X509Data ablegen.
   Ist der nötige Sicherheitszustand für den privaten Schlüssel der SM-B nicht gesetzt, MUSS der Konnektor zur PIN-Verifikation an dem Kartenterminal auffordern, in dem die SM-B steckt.
2. Der Konnektor ermittelt die URI des Registrierungsservers (MGM_ZGDP_REGSERVER) durch eine DNS-Anfrage nach dem SRV und TXT Resource Record „_regserver._tcp.<DNS_DOMAIN_VPN_ZUGD_INT>„
3. Der Konnektor ruft unter Verwendung der erzeugten Request-Nachricht die in [gemSpec_VPN_ZugD#Tab_ZD_registerKonnektor] definierte Operation I_Registration_Service::registerKonnektor mit der Zieladresse MGM_ZGDP_REGSERVER auf.
4. Der Konnektor zeigt dem Administrator den Inhalt von registerKonnektorResponse/AdditionalInformation und /Status an
5. Der Response der Operation wird verarbeitet:
   a. Setze MGM_TI_ACCESS_GRANTED auf
      - Enabled, wenn /RegistrationStatus = „Registriert"
      - Disabled, wenn /RegistrationStatus = „Nicht registriert"
   b. Persistiere diese Zustandsinformation zusammen mit dem VPN:ContractStatus
   c. Verteile das folgende interne Ereignis über TUC_KON_256 {
      topic = "MGM/TI_ACCESS_GRANTED";
      eventType = Op;
      severity = Info;
      parameters = „Active=$MGM_TI_ACCESS_GRANTED";
      doDisp = false }

Tritt während der Verarbeitungskette ein Fehler auf, so bricht die weitere Verarbeitung ab und der Administrator MUSS darüber geeignet informiert werden (u.a. Klartextanzeige des vom Registrierungsdienst gemeldeten Fehlers).
Wenn eine Reregistrierung mit einer neuen SMC-B fehlschlägt (Request wird mit einem SOAP-Error beantwortet) dann ist der Konnektor nicht registriert (MGM_TI_ACCESS_GRANTED = Disabled).
[<=]

TIP1-A_4826 - Status Konnektorfreischaltung einsehen
Der Administrator MUSS über die Managementschnittstelle den aktuellen Freischaltstatus einsehen können (MGM_TI_ACCESS_GRANTED). Ist der Konnektor aktuell freigeschaltet, so MUSS ihm dies zusammen mit dem VPN:ContractStatus angezeigt werden.
[<=]

Möchte ein Konnektoreigentümer das Gerät weiterveräußern oder vollständig außer Betrieb nehmen, so sollte er eine vorhandene Freischaltung zuvor rückgängig machen.

TIP1-A_4827 - Konnektorfreischaltung zurücknehmen
Ist MGM_TI_ACCESS_GRANTED=Enabled, dann MUSS der Administrator über die Managementschnittstelle des Konnektors die Freischaltung über den folgenden Mechanismus zurücknehmen können:

1. Der Administrator MUSS eine Sicherheitsabfrage zur Zurücknahme der Freischaltung bestätigen
2. Der Konnektor MUSS eine deRegisterKonnektorRequest-Struktur gemäß [gemSpec_VPN_ZugD] erstellen und mit den entsprechenden Parametern befüllen (aktuelles Datum/Uhrzeit, C.NK.VPN, MGM_ ZGDP_CONTRACTID)
3. Der Konnektor MUSS die Request-Nachricht mittels einer verfügbaren SM-B (ID.HCI.OSIG) im Element deRegisterKonnektorRequest/Signature signieren. (MGM_ ZGDP_SMCB ist zu bevorzugen, es kann aber auch jede andere SM-B verwendet werden)
   Ist der nötige Sicherheitszustand für den privaten Schlüssel der SM-B nicht gesetzt, MUSS der Konnektor zur PIN-Verifikation an dem Kartenterminal auffordern, in dem die SM-B steckt.
4. Der Konnektor ermittelt die URI des Registrierungsservers (MGM_ZGDP_REGSERVER) durch eine DNS-Anfrage nach dem SRV und TXT Resource Record „_regserver._tcp.<DNS_DOMAIN_VPN_ZUGD_INT>„
5. Der Konnektor ruft unter Verwendung der erzeugten Request-Nachricht die in [gemSpec_VPN_ZugD#Tab_ZD_deregisterKonnektor] definierte Operation I_Registration_Service::deRegisterKonnektor mit der Zieladresse MGM_ZGDP_REGSERVER auf.
6. Der Konnektor zeigt dem Administrator den Inhalt von deregisterKonnektorResponse/AdditionalInformation /ContractStatus und /RegistrationStatus an
7. Der Response der Operation wird verarbeitet:
   a. Setze MGM_TI_ACCESS_GRANTED auf
      - Enabled, wenn /RegistrationStatus = „Registriert"
      - Disabled, wenn /RegistrationStatus = „Nicht registriert"
   b. Persistiere diese Zustandsinformation zusammen mit dem Zeitpunkt
   c. Verteile das folgende interne Ereignis über TUC_KON_256: {
      topic = "MGM/TI_ACCESS_GRANTED";
      eventType = Op;
      severity = Info;
      parameters = „Active=$MGM_TI_ACCESS_GRANTED";
      doDisp=false }

Tritt während der Verarbeitungskette ein Fehler auf, so bricht die weitere Verarbeitung ab und der Administrator MUSS darüber geeignet informiert werden (u.a. Klartextanzeige des vom Registrierungsdienst gemeldeten Fehlers).
Wenn eine Deregistrierung mit einer neuen SMC-B fehlschlägt (Request wird mit einem SOAP-Error beantwortet) dann ist der Konnektor weiterhin registriert (MGM_TI_ACCESS_GRANTED = Enabled).
[<=]

TIP1-A_5655 - Deregistrierung bei Außerbetriebnahme
Der Hersteller des Konnektors MUSS im Handbuch den Administrator darüber informieren, dass der Konnektor bei dauerhafter Außerbetriebnahme (z. B. Verkauf, Schenkung, Entsorgung) beim Zugangsdienstprovider deregistriert werden muss.
[<=]

4.3.8 Re-Registrierung des Konnektors mit neuem NK-Zertifikat

A_21745-01 - Re-Registrierung mit neuem NK-Zertifikat automatisch durchführen
Nach einer vollständigen erfolgreichen automatischen Zertifikatserneuerung über TUC_KON_410 MUSS der Konnektor eine Re-Registrierung mit dem neuen Zertifikat beim Registrierungsdienst des VPN-Zugangsdienstes durchführen. Solange nach Bezug eines neuen C.NK.VPN-Zertifikats noch keine erfolgreiche Re-Registrierung durchgeführt wurde, MUSS der Konnektor genau einmal täglich TUC_KON_411 aufrufen.
[<=]

A_21881 - Re-Registrierung mit neuem NK-Zertifikat manuell durchführen
Der Konnektor MUSS die manuelle Re-Registrierung mittels TUC_KON_411 durch den Administrator auch im kritischen Betriebszustand `EC_NK_Certificate_Expired` ermöglichen.**[<=]**

A_21758-01 - TUC_KON_411 „Konnektor mit neuem NK-Zertifikat registrieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_411 "Konnektor mit neuem NK-Zertifikat registrieren" umsetzen.

Tabelle 365: TAB_KON_932 – TUC_KON_411 „Konnektor mit neuem NK-Zertifikat registrieren"

Element	Beschreibung
Name	TUC_KON_411 "Konnektor mit neuem NK-Zertifikat registrieren"
Beschreibung	Dieser TUC führt eine Deregistrierung mit dem alten und eine Neuregistrierung mit dem neuen NK-Zertifikat durch.
Auslöser	A_21745, Administrator
Vorbedingungen	Keine
Eingangsdaten	Keine
Komponenten	Konnektor, VPN-ZugD
Ausgangsdaten	Keine

Standardablauf	1. Der Konnektor ermittelt die URI des Registrierungsservers (MGM_ZGDP_REGSERVER) durch eine DNS-Anfrage nach dem SRV und TXT Resource Record „_regserver._tcp.<DNS_DOMAIN_VPN_ZUGD_INT>".
	2. Der Konnektor MUSS eine deRegisterKonnektorRequest-Struktur gemäß [gemSpec_VPN_ZugD] erstellen und mit den entsprechenden Parametern befüllen (aktuelles Datum/Uhrzeit, bei der letzten erfolgreichen Registrierung verwendetes C.NK.VPN-Zertifikat, MGM_ ZGDP_CONTRACTID).
	Der Konnektor MUSS die Request-Nachricht mittels einer verfügbaren SM-B (ID.HCI.OSIG) im Element deRegisterKonnektorRequest/Signature signieren. (MGM_ ZGDP_SMCB ist zu bevorzugen, es kann aber auch eine andere SM-B verwendet werden).
	3. Der Konnektor ruft unter Verwendung der erzeugten Request-Nachricht die in [gemSpec_VPN_ZugD#Tab_ZD_deregisterKonnektor] definierte Operation I_Registration_Service::deRegisterKonnektor mit der Zieladresse MGM_ZGDP_REGSERVER auf.
	Der Response der Operation wird verarbeitet:
	a. Setze MGM_TI_ACCESS_GRANTED auf
	- Enabled, wenn /RegistrationStatus = „Registriert"
	- Disabled, wenn /RegistrationStatus = „Nicht registriert"
	b. Persistiere diese Zustandsinformation zusammen mit dem Zeitpunkt
	c. Verteile das folgende Ereignis über TUC_KON_256: {
	topic = "MGM/TI_ACCESS_GRANTED";
	eventType = Op;
	severity = Info;
	parameters = „Active=$MGM_TI_ACCESS_GRANTED";
	doLog = true;
	doDisp=true }
	4. Der Konnektor MUSS eine registerKonnektorRequest-Struktur gemäß ProvisioningService.xsd [gemSpec_VPN_ZugD] erstellen und mit den entsprechenden Parametern befüllen (aktuelles Datum/Uhrzeit, erneuertes C.NK.VPN-Zertifikat, MGM_ ZGDP_CONTRACTID). Der Konnektor MUSS die Request-Nachricht mittels der ausgewählten SM-B (ID.HCI.OSIG) im Element registerKonnektorRequest/Signature signieren und das SM-B-Zertifikat im Element X509Data ablegen.

	5. Der Konnektor ruft unter Verwendung der erzeugten Request-Nachricht die in [gemSpec_VPN_ZugD#Tab_ZD_registerKonnektor] definierte Operation I_Registration_Service::registerKonnektor mit der Zieladresse MGM_ZGDP_REGSERVER auf.
	Der Response der Operation wird verarbeitet:
	a. Setze MGM_TI_ACCESS_GRANTED auf
	- Enabled, wenn /RegistrationStatus = „Registriert"
	- Disabled, wenn /RegistrationStatus = „Nicht registriert"
	b. Persistiere diese Zustandsinformation zusammen mit dem VPN:ContractStatus
	c. Verteile das folgende Ereignis über TUC_KON_256
	{
	topic = "MGM/TI_ACCESS_GRANTED";
	eventType = Op;
	severity = Info;
	parameters = „Active=$MGM_TI_ACCESS_GRANTED";
	doLog = true;
	doDisp = true }

Varianten/Alternativen	Automatische Registrierung: (->5) Wenn der Konnektor nicht mit dem neuen C.NK.VPN-Zertifikat registriert werden konnte, dann muss sich der Konnektor, beginnend mit Schritt 4, erneut mit dem alten C.NK.VPN-Zertifikat registrieren. Manuelle Registrierung: (->2) Der Administrator soll die zu verwendende SM-B auswählen können.

Fehlerfälle	(→ 2,4) Es konnte keine freigeschaltete SM-B ausgewählt werden:
	Fail=No_Smcb
	(->4,5) Im Fehlerfall
	TUC_KON_256 {
	topic = „SMC_K/REGISTER/ERROR";
	eventType = Op;
	severity = Error;
	parameters = „$Parameters";
	doLog = true;
	doDisp = true }
	Die Registrierung soll herstellerspezifisch erneut mehrmals versucht werden.
	Bei allen Fehlerfällen, die zum Abbruch führen:
	TUC_KON_256 {
	topic = „SMC_K/REGISTER/ERROR";
	eventType = Op;
	severity = Error;
	parameters = „$Parameters";
	doLog = true;
	doDisp = true }

Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 366: Tab_Kon_933 Fehlercodes TUC_KON_411 „Zertifikate aktualisieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
herstellerspezifisch			

[<=]

A_21781 - Nutzerhinweis bezüglich Fehler bei Re-Registrierung
Der Hersteller des Konnektors MUSS in seinem Handbuch den Nutzer/Administrator darauf hinweisen, dass das Ereignis mit dem Topic=SMC_K/REGISTER/ERROR dringend durch das Primärsystem abonniert werden sollte. Bei Auftreten des Fehlers mit Fail=No_Smcb muss in der Leistungserbringerumgebung dafür gesorgt werden, dass eine freigeschaltete SMC-B verfügbar ist, die der Konnektor beim nächsten Re-Registrierungsversuch verwenden kann.**[<=]**

4.3.9 Remote Management (Optional)

Im Betreibermodell der TI wird unter Remote Management ein delegierter Betrieb dezentraler Produkte durch einen durch den Anwender beauftragten Servicepartner

verstanden. Der Servicepartner stellt als Vertragsbestandteil bevollmächtigte Personen zur Verfügung, die sich ständig um die Betriebs- und Datensicherheit der dezentralen Produkte im Rahmen eines Remote Managements kümmern.

Voraussetzung für die Etablierung dieses Bestandteils des Betreibermodells der TI ist, dass ein dezentrales Produkt ein Remote Management technisch unterstützt. Die nachfolgend aufgeführten Anforderungen bilden die Grundlage für die Nutzung von Remote Management am Konnektor.

Zum Remote Management gehören die Verwaltung von Konfigurationsdaten und die Durchführung weiterer Administratoraktionen wie z. B. die Aktualisierung der Software des Konnektors. Im Rahmen des Remote Managements kann der Konnektor Remote Monitoring unterstützen. Dazu übermittelt der Konnektor Betriebszustandsdaten an das Remote- Management-System.

TIP1-A_7276-01 - Remote Management Konnektor
Der Konnektor KANN Remote Management technisch unterstützen.
Falls der Konnektor das Remote Management technisch unterstützt, MUSS der Konnektor alle Anforderungen, die das Remote Management (z.B. auch Remote-Administrator) betreffen, umsetzen.
Andernfalls sind die Anforderungen, die das Remote Management (z.B. auch Remote-Administrator) betreffen, für den Konnektor nicht relevant.**[**<=**]**

TIP1-A_5647 - Remote Management Konnektor: Personenbezogene Daten
Der Konnektor DARF über die Remote-Managementschnittstelle KEINE personenbezogenen Daten übertragen oder darstellen.
[<=**]**

TIP1-A_5648 - Remote Management Konnektor: Offene Schnittstelle
Der Hersteller des Konnektors MUSS die zur Nutzung der Remote-Managemenschnittstelle notwendigen Informationen offenlegen. Der Hersteller des Konnektors MUSS die Remote-Managementschnittstelle so spezifizieren und implementieren, dass diese auch für Dritte (z.B. einen durch den Anwender beauftragten Servicepartner) nutzbar ist.
[<=**]**

TIP1-A_5649 - Remote Management Konnektor: Standardbasierte Protokolle
Der Hersteller des Konnektors SOLL für die Implementierung der Remote-Managementschnittstelle standardbasierte Verfahren und Protokolle einsetzen.
[<=**]**

TIP1-A_5650 - Remote Management Konnektor: Aufbau der Verbindung
Der Konnektor MUSS sicherstellen, dass die Initiierung einer Remote-Managementverbindung im Sinne des Verbindungsaufbaus immer vom Konnektor ausgeht.
[<=**]**

TIP1-A_5651 - Remote Management Konnektor: Absicherung der Verbindung
Der Konnektor MUSS die Remote-Management-Verbindung durch Nutzung eines kryptographischen Verfahrens gemäß [gemSpec_Krypt] hinsichtlich Vertraulichkeit, Integrität und Authentizität absichern.
[<=**]**

Das Remote-Management-System authentisiert sich auf Transportebene zertifikatsbasiert gegenüber dem Konnektor.

TIP1-A_7277 - Authentifizierung des Remote-Management-Systems
Der Konnektor MUSS eine zertifikatsbasierte Authentifizierung des Remote-Management-Systems auf Transportebene durchführen.**[<=]**

TIP1-A_7278 - Authentisierung des Konnektors gegenüber Remote-Management-System
Der Konnektor MUSS sich gegenüber dem Remote-Management-System zertifikatsbasiert oder mittels Username/Password authentisieren.**[<=]**

TIP1-A_7281 - Authentifizierung des Konnektors durch das Remote-Management-System
Das Remote-Management-System MUSS eine Authentifizierung des Konnektors durchführen.**[<=]**

Die Authentifizierung des Remote-Management-Systems durch den Konnektor auf Transportebene ist verpflichtend gefordert.

Darüber hinaus können optional Remote-Administratoren in der Benutzerverwaltung des Konnektors konfiguriert werden. Wenn Remote-Administratoren in der Benutzerverwaltung konfiguriert sind, muss der Konnektor diese verpflichtend auf Anwendungsebene authentifizieren.

Wenn kein Remote-Administrator konfiguriert ist, vertraut der Konnektor der Benutzerverwaltung des Remote-Management-Systems. Auch wenn die Verwaltung von Remote-Administratoren an das Remote-Management-System delegiert ist, werden alle Zugriffe über das Remote-Management-System auf den Konnektor mit der Rolle Remote-Administrator ausgeführt. Das Remote-Management-System muss die Authentisierung der Remote-Administratoren und die Nachvollziehbarkeit der Zugriffe sicherstellen.

TIP1-A_7279 - Authentifizierung des Remote-Administrators
Wenn in der Benutzerverwaltung des Konnektors Administratoren mit der Administrator-Rolle Remote-Administrator konfiguriert sind, MUSS der Konnektor diese gemäß TIP1-A_4808 authentifizieren.**[<=]**

TIP1-A_7280 - Einschränkung der Rechte des Remote-Administrators
Der Konnektor DARF Remote-Administratoren Rechte gemäß TAB_KON_851 und TAB_KON_655 NICHT gewähren.**[<=]**

Tabelle 367: TAB_KON_851 Einschränkung der Rechte des Remote-Administrators (Blacklist)

Fachliche Anbindung der Clientsysteme		
TIP1-A_4517	Schlüssel und X.509-Zertifikate für die Authentisierung des Clientsystems erzeu-gen und exportieren sowie X.509-Zertifikate importieren	
TIP1-A_4518	Konfiguration der Anbindung Clientsysteme	
Kartendienst		
TIP1-A_5110	Übersicht über alle verfügbaren Karten	Karten vom Typ eGK und HBA DÜRFEN dem Remote-

		Administrator NICHT angezeigt werden
TIP1-A_5111	PIN-Management der SM-Bs für den Administrator	
Zertifikatsdienst		
TIP1-A_4704	Zertifikatsablauf anzeigen	Zertifikate von Karten vom Typ eGK und HBA DÜRFEN dem Remote-Administrator NICHT angezeigt werden
Protokollierungsdienst		
TIP1-A_4716	Einsichtnahme und Veränderung der Protokolle	Personenbezogene Daten DARF der Remote-Administrator NICHT einsehen und exportieren. Fachmodulprotokolle müssen daher entweder gesperrt, oder die personenbezogenen Daten aus diesen für den Remote-Administrator gefiltert werden.
TIP1-A_4814	Export- Import von Konfigurationsdaten	
Neustart und Werksreset		
TIP1-A_4820	Werksreset des Konnektors	

TIP1-A_5652 - Remote Management Konnektor: Konfiguration Remote Management
Der Konnektor MUSS sicherstellen, dass es ausschließlich einem Administrator mit einer der Rollen {Lokaler Administrator; Super-Administrator} und dem Recht USER_INIT_REMOTESESSION möglich ist, Konfigurationsänderungen gemäß TAB_KON_663 vorzunehmen.

Tabelle 368: TAB_KON_663 Konfigurationen des Remote Managements

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
MGM_REMOTE_ ALLOWED	Enabled/ Disabled	Der Administrator MUSS einstellen können, ob der Konnektor eine Remote-Management-Verbindung aufbauen kann, über die Konfigurationen vorgenommen werden können. Enabled: Der Konnektor kann eine

		Remote-Management-Verbindung aufbauen und erlaubt Konfiguration über das Remote-Management System. Disabled: Der Konnektorerlaubt keine Konfiguration über das Remote Management-System Default-Wert: Disabled
MGM_REMOTE_MONITORING_ALLOWED	Enabled / Disabled	Der Konnektor KANN Remote-Monitoring unterstützen. In diesem Fall MUSS der Konnektor dem Administrator die Aktivierung und Deaktivierung des Remote-Monitoring ermöglichen. Enabled: Der Konnektor baut eine Remote-Managementverbindung auf. Der Konnektor übermittelt Betriebszustände gemäß TAB_KON_503 an das Remote-Management-System. Disabled: Remote-Monitoring ist deaktiviert. Der Konnektor übermittelt keine Betriebszustände an das Remote-Management-System. Default-Wert: Disabled
Der Konnektor SOLL die Konfiguration der URL des Remote-Management-Systems, der Zertifikatsinformationen zur Authentisierung des Remote-Management-Systems und der Credentials für die Authentisierung des Konnektors beim Remote-Management-System ermöglichen.		

[<=]

TIP1-A_5653 - Remote Management Konnektor: Protokollierung Remote Management
Der Konnektor MUSS im Rahmen des Remote-Managements folgende Aktionen protokollieren:

- Beginn einer Remote-Session durch
  TUC_KON_271 „Schreibe Protokolleintrag“ {
  topic = „MGM/REMOTE_SESSION“;
  eventType = Op;
  severity = Info;
  parameters = („InitUser=$AdminUsername,
  RemoteID=<Kennung der Gegenstelle>,
  Mode=[InitSuccess/InitFail]“)}
- Verbindungsabbau Remote-Session durch
  TUC_KON_271 „Schreibe Protokolleintrag“ {
  topic = „MGM/REMOTE_SESSION“;
  eventType = Op;
  severity = Info;

parameters = („InitUser=$AdminUsername,
RemoteID=<Kennung der Gegenstelle>,
Mode=Exit"}
Die Protokollierungspflicht gilt nicht für das Remote Monitoring.
Wenn ein remote-Zugriff erfolgt, ohne dass ein Remote-Administrator im Konnektor konfiguriert ist, so MUSS als InitUser eine Referenz auf das Remote-Management-System verwendet werden.

[<=]

Ein Softwareupdate gemäß TIP1-A_5657 kann auch über das Remote Management initiiert, aktiviert und freigeschaltet werden.

4.3.10 Software- und Konfigurationsaktualisierung (KSR-Client)

Die Umsetzung des KSR-Clients bezüglich des Mechanismus zur Durchführung der Aktualisierungen, sowie die Art der Darstellung an der Managementschnittstelle sind herstellerspezifisch.

Innerhalb der Software- und Konfigurationsaktualisierung (KSR-Client) werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): „KSR"
- Konfigurationsparameter: „MGM_"

4.3.10.1 Funktionsmerkmalweite Aspekte

Der Konnektor muss einen KSR-Client bereitstellen, über den der Administrator sowohl den Konnektor selbst als auch die vom Konnektor verwalteten Kartenterminals (CT-Objects in CTM_CT_LIST mit CT.CORRELATION>=„gepairt" und CT.VALID_VERSION=True und CT.IS_PHYSICAL = Ja) softwareseitig aktualisieren kann.

Weiterhin muss über den KSR-Client eine Aktualisierung von ausgewählten Konfigurationsdaten möglich sein.

TIP1-A_4829 - Vollständige Aktualisierbarkeit des Konnektors
Die Software-Aktualisierung des Konnektor SOLL sicherstellen, dass alle Software-Bestandteile des Konnektors aktualisiert werden können, damit eine ungehinderte Nachnutzung der Hardware-Basis im Feld mit neuen Funktionalitäten nicht durch nichtaktualisierbare Software-Bestandteile gefährdet wird. Weicht ein Hersteller für sein Konnektormodell von dieser Forderung in Teilen ab, so MUSS er im Rahmen der Zulassung nachweisen, dass dies auf Grund von Sicherheitsaspekten für sein eingereichtes Konnektormodell zwingend erforderlich ist.
[<=]

TIP1-A_5657-02 - Freischaltung von Softwareupdates
Der Konnektor MUSS die Möglichkeit bieten, dass Softwareupdates durch den Nutzer bzw. einen von ihm beauftragten Administrator einzeln freigeschaltet werden.
[<=]

A_18387 - Automatische Softwareupdates
Der Konnektor MUSS die Möglichkeit bieten, die automatische Installation von Softwareupdates pro Gerät (Konnektor und Kartenterminals) ein- und auszuschalten.**[<=]**

A_18389 - Nur Nutzung von zugelassenen Versionen
Der Hersteller des Konnektors MUSS in seinem Handbuch den Nutzer darauf hinweisen, dass er sich bei der Arbeit mit dem Konnektor vergewissern muss, dass er mit einer

zugelassenen Version arbeitet und beschreiben, wie der Nutzer diese Information mittels seines Primärsystems erhalten kann.
[<=]

TIP1-A_6476 - Lieferung von Softwareupdates
Der Hersteller des Konnektors MUSS jede zugelassene Firmware-Version umgehend als Update-Paket über die in [gemSpec_KSR] definierte Schnittstelle P_KSRS_Upload im Konfigurationsdienst (KSR) ablegen.
Der Hersteller des Konnektors MUSS in den jeweiligen UpdateInformation/Firmware/FirmwareReleaseNotes eine Internet-URL zum Download des FW-Updates bereitstellen.

[<=]

TIP1-A_6026 - Anzeige URL zum Download des FW-Updates an der Managementschnittstelle
Das Managementinterface des Konnektors MUSS einem authentisierten Administrator die Internet-URL zum Download des FW-Updates anzeigen.
[<=]

4.3.10.2 Durch Ereignisse ausgelöste Reaktionen

TIP1-A_4831 - KT-Update nach Wiedererreichbarkeit erneut anstoßen
Wenn aus (TIP1-A_4840 Auslösen der durchzuführenden Updates) heraus für ein Kartenterminal noch ein ausstehendes Updates vorhanden ist, dessen Ausführungszeitpunkt nicht gesetzt oder überschritten ist, und für dieses Kartenterminal das Ereignis „CT/CONNECTED“ eintritt, so MUSS TUC_KON_281 „Kartenterminalaktualisierung anstoßen“ für dieses KT gerufen werden.
[<=]

4.3.10.3 Interne TUCs, nicht durch Fachmodule nutzbar

4.3.10.3.1 TUC_KON_280 „Konnektoraktualisierung durchführen“

TIP1-A_4832-02 - TUC_KON_280 „Konnektoraktualisierung durchführen“
Der Konnektor MUSS den technischen Use Case TUC_KON_280 „Konnektoraktualisierung durchführen“ umsetzen.

Tabelle 369: TAB_KON_664 – TUC_KON_280 „Konnektoraktualisierung durchführen“

Element	Beschreibung
Name	TUC_KON_280 „Konnektoraktualisierung durchführen“
Beschreibung	Dieser TUC aktualisiert den Konnektor mit einem Update, dessen Update-Dateien entweder direkt übergeben oder per UpdateInformation (vom KSRS bezogen) referenziert werden
Auslöser	• Der Administrator hat UpdateInformation zur Anwendung ausgewählt und bestätigt bzw. ein lokales Updatepaket bezogen und zur Anwendung übergeben. • automatisches Softwareupdate [A_18387]
Vorbedingungen	

Eingangsdaten	• UpdateInformation (gemäß [gemSpec_KSR#5.2]) oder • Updatepaket (herstellerspezifisch, von lokaler Datenquelle, mit enthaltenen FirmwareFiles)
Komponenten	Konnektor, Konfigurationsdienst
Ausgangsdaten	Keine
Nachbedingungen	Der Konnektor arbeitet basierend auf der übergebenen, im Updatepaket enthaltenen neuen Version.
Standardablauf	Der Konnektor MUSS die zur Anwendung übergebene UpdateInformation wie folgt anwenden: 1. Integrität und Authentizität der UpdateInformation prüfen (Mechanismus ist herstellerspezifisch) 2. Download aller in UpdateInformation.FirmwareFiles gelisteten Dateien. Dabei wird die Komprimierung des File Transfers vom Konfigurationsdienst über http „Content Coding" [RFC2616] „gzip" genutzt. 3. Integrität und Authentizität jeder der via UpdateInformation/FirmwareFiles heruntergeladenen Dateien prüfen (Mechanismus ist herstellerspezifisch) 4. Prüfen auf Zulässigkeit des Updates basierend auf der Firmware-Gruppe (siehe [gemSpec_OM#2.5] 5. Anwenden der zur Verfügung stehenden FirmwareFiles a. TUC_KON_256{ topic = „KSR/UPDATE/START"; eventType = Sec; severity = Info; parameters = („Target=Konnektor, Name=$MGM_KONN_HOSTNAME")} (betroffene Fachmodule und Basisdienste reagieren und stoppen sich) b. Herstellerspezifischer Mechanismus zur Aktualisierung der internen Konnektorsoftware durch die FirmwareFiles inklusive anschließender Prüfung auf Erfolg. c. Bestehende Konfigurationsdaten des Konnektors MÜSSEN erhalten bleiben und sofern erforderlich und möglich automatisch auf die Definitionen der neuen Firmware angepasst werden. d. Ist ein händischer Anpassungs- oder Ergänzungsbedarf der Konfigurationsdaten erforderlich, so MUSS der Administrator hierüber geeignet informiert werden e. TUC_KON_256 { topic = „KSR/UPDATE/SUCCESS"; eventType = Sec;

	severity = Info;
	parameters = („Target=Konnektor,
	Name= $MGM_KONN_HOSTNAME,
	NewFirmwareversion =
	UpdateInformation.FirmwareVersio
	n,
	ConfigurationChanged=<Ja/Nein>,
	ManualInputNeeded=<Ja/Nein>„) }
	Der TUC endet in jedem Fall mit:
	TUC_KON_256 {
	topic = „KSR/UPDATE/END“;
	eventType = Sec;
	severity = Info;
	parameters = („Target=Konnektor,
	Name=$MGM_KONN_HOSTNAME“) }
	(betroffene Fachmodule und Basisdienste reagieren und starten sich)

Varianten/Alternativen	Sofern direkt ein Updatepaket (mit enthaltenen FirmwareFiles) übergeben wurde beginnt der Ablauf ab Nummer 4 mit der Integritätsprüfung des Updatepakets

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu:
	a) Aufruf von TUC_KON_256 {
	topic = „KSR/ERROR“;
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („Target=Konnektor,
	Name= $MGM_KONN_HOSTNAME,
	Error=$Fehlercode,
	Bedeutung=$Fehlertext“) }
	
	b) Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes
	
	(→1) Integritätsprüfung UpdateInformation fehlgeschlagen, Fehlercode: 4181
	(→2) Fehler bei der Downloaddurchführung, Fehlercode: 4182
	(→3) Integritätsprüfung eines FirmwareFiles fehlgeschlagen, Fehlercode: 4183
	(→ 4) Firmwaregruppenprüfung fehlgeschlagen, Fehlercode: 4185
	(→5b) Interne Aktualisierung fehlgeschlagen, dann:
	1. Rollback auf vorherige Version
	2. Abbruch mit Fehlercode: 4184

Nichtfunktionale Anforderungen	Der laufende Updatevorgang MUSS in der Managementschnittstelle ausgewiesen und der Fortschritt mindestens für die Schritte 1-5b dargestellt werden.
Zugehörige Diagramme	Abbildung PIC_KON_105 Aktivitätsdiagramm Konnektoraktualisierung durchführen

Tabelle 370: TAB_KON_665 Fehlercodes TUC_KON_280 „Konnektoraktualisierung durchführen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4181	Security	Error	Integritätsprüfung UpdateInformation fehlgeschlagen.
4182	Security	Error	Download nicht aller UpdateFiles möglich.
4183	Security	Error	Integritätsprüfung UpdateFiles fehlgeschlagen.
4184	Security	Error	Anwendung der UpdateFiles fehlgeschlagen (<Details>).
4185	Security	Error	Firmware-Version liegt außerhalb der gültigen Firmware-Gruppe

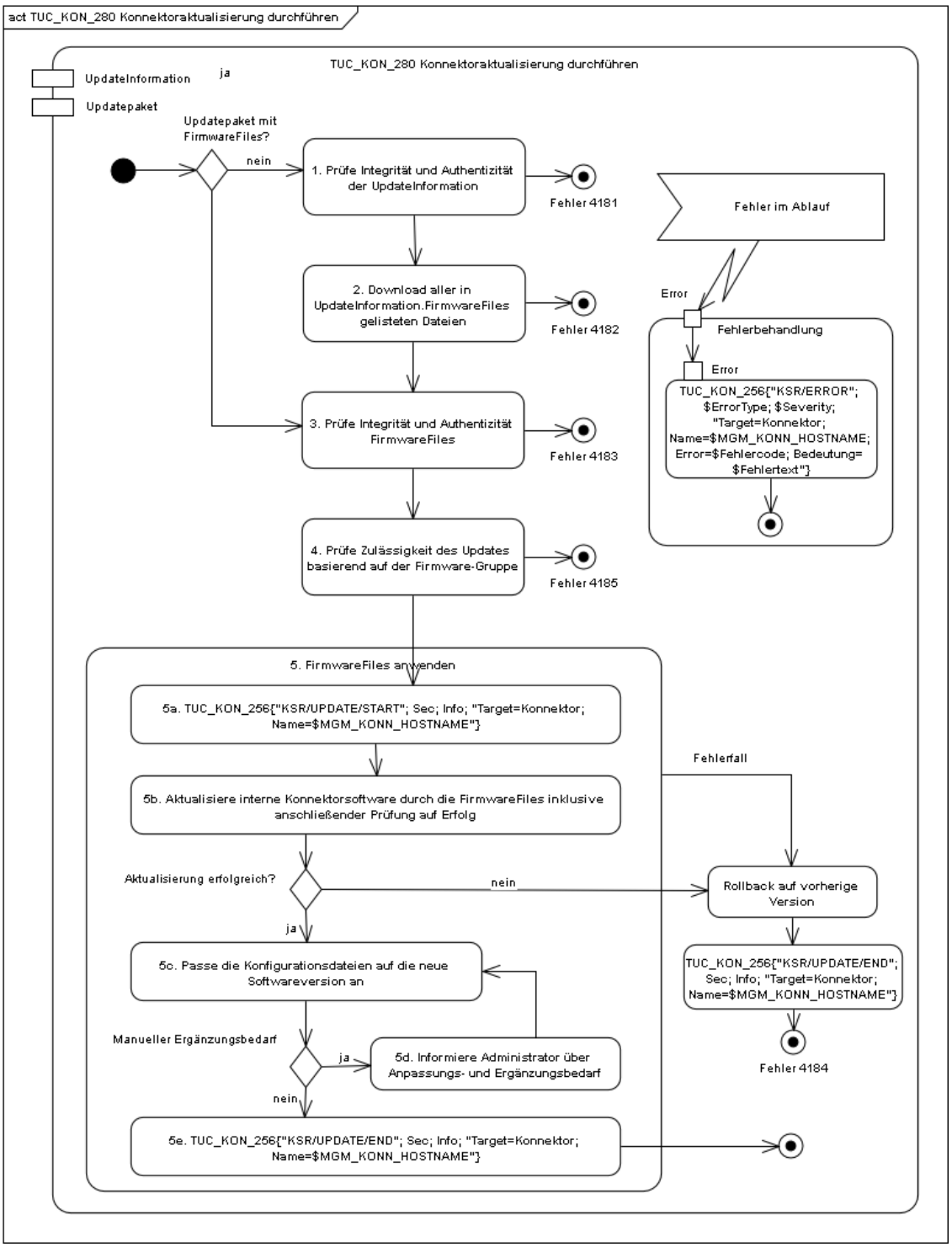


Abbildung 22: PIC_KON_105 Aktivitätsdiagramm Konnektoraktualisierung durchführen

[<=]

4.3.10.3.2 TUC_KON_281 „Kartenterminalaktualisierung anstoßen"

Im Vergleich zur Durchführung des Konnektor-Update (TUC_KON_280), werden die Updates der Kartenterminals nur durch den Konnektor initiiert. Der Konnektor liefert dem Kartenterminal das Updatefile, der eigentliche Updatevorgang (inklusive der Prüfung des Updatepakets auf Integrität und Authentizität) erfolgt ausschließlich und eigenverantwortlich auf Seiten des Kartenterminals.

TIP1-A_4833-02 - TUC_KON_281 „Kartenterminalaktualisierung anstoßen"
Der Konnektor MUSS den technischen Use Case TUC_KON_281 „Kartenterminalaktualisierung anstoßen" umsetzen.

Tabelle 371: TAB_KON_666 – TUC_KON_281 „Kartenterminalaktualisierung anstoßen"

Element	Beschreibung
Name	TUC_KON_281 „Kartenterminalaktualisierung anstoßen"
Beschreibung	Dieser TUC fordert ein Kartenterminal auf einen Update durchzuführen, dessen Update-Dateien entweder direkt übergeben oder per UpdateInformation (vom KSRS bezogen) referenziert werden
Auslöser	• Der Administrator hat UpdateInformation für ein Kartenterminal zur Anwendung ausgewählt und bestätigt bzw. ein lokales Updatepaket für ein Kartenterminal bezogen und zur Anwendung übergeben. • automatisches Softwareupdate [A_18387]
Vorbedingungen	• CT(ctId).IS_PHYSICAL=Ja • CT(ctId).CORRELATION>=”gepairt”
Eingangsdaten	• ctId (ID des Ziel-KTs) • UpdateInformation (gemäß [gemSpec_KSR]) oder • Updatepaket (herstellerspezifisch, von lokaler Datenquelle, mit enthaltenen FirmwareFiles)
Komponenten	Konnektor, Kartenterminal
Ausgangsdaten	Keine
Nachbedingungen	Das Kartenterminal arbeitet basierend auf der übergebenen, im Updatepaket enthaltenen neuen Version.

Standardablauf	Der Konnektor MUSS die zur Anwendung übergebene UpdateInformation wie folgt anwenden:
	1. Download der in UpdateInformation/FirmwareFiles gelisteten Datei (für KT-Updates darf nur genau ein FirmwareFile angegeben werden)
	2. TUC_KON_256{
	topic = „KSR/UPDATE/START”;
	eventType = Sec;
	severity = Info;
	parameters = („Target=KT, CtID=$ctId") }

	3. Durchführen des KT-Updates durch:
	a) Wechsel in eine Admin-Session durch TUC_KON_050
	„Beginne Kartenterminalsitzung"{role=„Admin"; ctId}
	b) Senden der SICCT Kommandos: SICCT CT Download INIT,
	SICCT CT Download DATA (Übermittlung des UpdateFiles) und
	SICCT CT Download FINISH an ctId
	c) TUC_KON_256{
	topic = „KSR/UPDATE/SUCCESS";
	eventType = Sec;
	severity = Info;
	parameters = („Target=KT,
	Name= $CT.HOSTNAME, CtID =$ctId,
	NewFirmwareversion =
	<UpdateInformation.FirmwareVersion>„}
	
	Der TUC endet in jedem Fall mit:
	• TUC_KON_256 {
	topic = „KSR/UPDATE/END";
	eventType = Sec;
	severity = Info;
	parameters = („Target=KT, CtID =$ctId") }

Varianten/Alternativen	Sofern direkt ein Updatepaket (mit enthaltenem FirmwareFile) übergeben wurde beginnt der Ablauf ab Nummer 2 mit Signalisierung des Beginns des KT-Updates

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu:
	a) Aufruf von TUC_KON_256 {
	topic = „KSR/ERROR";
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („Target=KT, Name=$CT.HOSTNAME,
	CtID =$ctId, Error=$Fehlercode,
	Bedeutung=$Fehlertext") }
	b) Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes
	(→1) Download fehlgeschlagen, Fehlercode: 4186
	(→3b) SICCT-Download fehlgeschlagen, Fehlercode: 4187

Nichtfunktionale Anforderungen	Die Durchführung eines KT-Updates DARF die weitere Operation des Konnektors NICHT behindern (weder auf Schnittstellenebene noch in der Managementschnittstelle). Der laufende Updatevorgang eines KT MUSS in der Managementschnittstelle ausgewiesen und der Fortschritt dargestellt werden. Der Konnektor MUSS mindestens 5 Kartenterminal-Updates parallel durchführen können.
Zugehörige Diagramme	keine

Tabelle 372: TAB_KON_667 Fehlercodes TUC_KON_281 „Kartenterminalaktualisierung anstoßen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4186	Security	Error	Download nicht aller UpdateFiles möglich.
4187	Security	Error	KT-Update fehlgeschlagen (<Fehlerinfo gemäß SICCT>)

[<=]

4.3.10.3.3 TUC_KON_282 „UpdateInformationen beziehen“

TIP1-A_4834 - TUC_KON_282 „UpdateInformationen beziehen“
Der Konnektor MUSS den technischen Use Case TUC_KON_282 „UpdateInformationen beziehen“ umsetzen.

Tabelle 373: TAB_KON_668 – TUC_KON_282 „UpdateInformationen beziehen“

Element	Beschreibung
Name	TUC_KON_282 „UpdateInformationen beziehen“
Beschreibung	Dieser TUC ermittelt vom zentralen Konfigurationsdienst sowohl für den Konnektor als auch für alle durch ihn verwalteten Kartenterminals die verfügbaren UpdateInformationen
Auslöser	• Manuell durch den Administrator • Automatisch
Vorbedingungen	Keine
Eingangsdaten	Keine
Komponenten	Konnektor, Konfigurationsdienst
Ausgangsdaten	Keine
Nachbedingungen	Der Konnektor verfügt über alle aktuellen UpdateInformationen

Standardablauf	Der Konnektor MUSS folgende Schritte durchlaufen:
	1. Der Konnektor MUSS die TLS-Verbindungen zum Konfigurationsdienst anhand des in MGM_KSR_FIRMWARE_URL angegebenen Wertes aufbauen. Dabei MUSS er das durch den Server präsentierte Zertifikat mittels TUC_KON_037 „Zertifikat prüfen“ {
	certificate = C.ZD.TSL-S;
	qualifiedCheck = not_required;
	offlineAllowNoCheck = true;
	policyList = oid_zd_tls_s;

	intendedKeyUsage= intendedKeyUsage(C.ZD.TSL-S);
	intendedExtendedKeyUsage = id-kp-serverAuth;
	validationMode = OCSP}
	auf Gültigkeit prüfen.
	Das Server-Zertifikat C.ZD.TLS-S MUSS für den Konfigurationsdienst ausgestellt sein.
	2. Der Konnektor MUSS sowohl für sich wie auch für jedes Kartenterminal (CT) aus CTM_CT_LIST mit CT.IS_PHYSICAL=Ja und CT.CORRELATION>=„gepairt" folgende Schritte durchlaufen:
	a. Belegen von listUpdatesRequest mit den korrekten Werten für ProductVendorID, ProductCode, HardwareVersion und FirmwareVersion
	b. Aufruf von I_KSRS_Download::list_Updates
	Liefert der Aufruf mindestens eine UpdateInformation mit einer UpdateInformation/Firmware/FWVersion > aktuelle Version der Konnektorsoftware, deren UpdateInformation/Firmware/FWPriority = „Kritisch", dann geht der Konnektor über in den Betriebszustand EC_Connector_Software_Out_Of_Date.
	Liefert der Aufruf mindestens eine UpdateInformation mit einer UpdateInformation/Firmware/FWVersion > aktuelle Version der Kartenterminalsoftware, deren UpdateInformation/Firmware/FWPriority = „Kritisch", dann geht der Konnektor über in den Betriebszustand EC_CardTerminal_Software_Out_Of_Date.
	3. Beenden der TLS-Verbindung

Varianten/Alternativen	Keine

Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu:
	a) Aufruf von TUC_KON_256 {
	topic = „KSR/ERROR";
	eventType = $ErrorType;
	severity = $Severity;
	parameters = („Error=$Fehlercode; Bedeutung=$Fehlertext")}
	
	b) Abbruch der Verarbeitung mit den ausgewiesenen Fehlercodes
	(→1) Konfigurationsdienst nicht erreichbar, Fehlercode: 4188
	(→1) Serverzertifikat ist nicht C.ZD.TLS_S, Fehlercode: 4189

	(→2b) Fehler beim Beziehen der Updatelisten, Fehlercode: 4190
Nichtfunktionale Anforderungen	Der Konnektor muss die Vorgaben aus [gemSpec_Krypt#3.3.2] für TLS-Verbindungen und hinsichtlich ECC-Migration die Vorgaben aus [gemSpec_Krypt#5] befolgen.
Zugehörige Diagramme	keine

Tabelle 374: TAB_KON_669 Fehlercodes TUC_KON_282 „UpdateInformationen beziehen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases, sowie der Fehlercodes von „I_KSRS_Download::listUpdates Response" können folgende weitere Fehlercodes auftreten:			
4188	Technical	Error	Konfigurationsdienst nicht erreichbar, konfigurierte Adresse kontrollieren.
4189	Security	Fatal	Konfigurationsdienst liefert falsches Zertifikat
4190	Technical	Error	Fehler beim Beziehen der Updatelisten

[<=]

4.3.10.3.4 TUC_KON_283 „Infrastruktur Konfiguration aktualisieren"

TIP1-A_5153 - TUC_Kon_283 „Infrastruktur Konfiguration aktualisieren"
Der Konnektor MUSS den technischen Use Case TUC_Kon_283 „Infrastruktur Konfiguration aktualisieren" umsetzen.

Tabelle 375: TAB_KON_799 – TUC_KON_283 „Infrastruktur Konfiguration aktualisieren"

Element	Beschreibung
Name	TUC_KON_283 Infrastruktur Konfiguration aktualisieren
Beschreibung	Dieser TUC liest die Infrastrukturdaten vom KSR ein.
Auslöser	Automatisch einmal täglich; BOOTUP, Administrator
Vorbedingungen	Der TUC_KON_304 „Netzwerk-Routen einrichten" MUSS fehlerfrei durchgelaufen sein. Der TUC_KON_321 „Verbindung zu dem VPN-Konzentrator der TI aufbauen" MUSS fehlerfrei durchgelaufen sein.
Eingangsdaten	Keine
Komponenten	Konnektor, Konfigurationsdienst

Ausgangsdaten	Keine

Standardablauf	Der Konnektor MUSS folgende Schritte durchlaufen:
	1. „Einlesen des Konfigurations-XML“:
	a. Der Konnektor MUSS eine TLS-Verbindung zum Konfigurationsdienst anhand des in MGM_KSR_KONFIG_URL angegebenen Parameters aufbauen. Dabei MUSS er das durch den Server präsentierte Zertifikat mittels TUC_KON_037 „Zertifikat prüfen“ {
	certificate = C.ZD.TSL-S;
	qualifiedCheck = not_required;
	offlineAllowNoCheck = true;
	policyList = oid_zd_tls_s;
	intendedKeyUsage = intendedKeyUsage(C.ZD.TSL-S);
	intendedExtendedKeyUsage = id-kp-serverAuth;
	validationMode = OCSP}
	auf Gültigkeit prüfen.
	Das Server-Zertifikat C.ZD.TLS-S MUSS für den Konfigurationsdienst ausgestellt sein.
	b. Herunterladen der Konfigurationsdaten mittels I_KSRS_Download::get_Ext_Net_Config (MGM_KSR_KONFIG_URL, „Bestandsnetze.xml“)
	2. Beenden der TLS-Verbindung
	„Prüfen der Versionskennung auf Änderungen“:
	Wenn das Element /Infrastructure/Version der heruntergeladenen Datei keine höhere Versionsnummer als die aktuell im Konnektor hinterlegte Version trägt, muss der TUC ohne Fehler beendet und ein Protokolleintrag geschrieben werden:
	TUC_KON_271 „Schreibe Protokolleintrag“ {
	topic = „KSR/UPDATE_KONFIG“;
	eventType = Op;
	severity = Info;
	parameters = („AlteVersion=$aktuelleVersion, NeueVersion=/Infrastructure/Version “)}
	3. Aktualisieren der Gesamtnetzliste
	Alle in der Datei enthaltenen Netzsegmente sind nach ANLW_BESTANDSNETZE zu übernehmen. In Abhängigkeit von ANLW_IA_BESTANDSNETZE sind neue angeschlossene Netze des Gesundheitswesens mit WANDA Basic nach ANLW_AKTIVE_BESTANDSNETZE zu übernehmen. Identifiziert wird ein Bestandsnetz hierbei an dessen ID in der Bestandsnetze.xml (<ID>). War der Aktivierungsstatus eines dieser Netze bereits durch den Administrator manuell konfiguriert, so muss dieser Status erhalten bleiben.
	4. „Aktualisieren von Konfigurationsinformationen“
	Haben sich Konfigurationsdaten zu einem in

ANLW_AKTIVE_BESTANDSNETZE gelisteten Netz verändert, so

a. sind die Änderungen entsprechend zu übernehmen und zu aktivieren (Anpassung ANLW_AKTIVE_BESTANDSNETZE, DHCP_AKTIVE_BESTANDSNETZE_ROUTES, DNS_SERVERS_BESTANDSNETZE).

b. alle Statusänderungen an ANLW_AKTIVE_BESTANDSNETZE sind zu protokollieren. Der Protokolleintrag je Änderung enthält den Status, <ID>, <Name> und <NetworkAddress/NetworkPrefix> als topic=KSR/UPDATE_KONFIG,protocolType=OP und protocolSeverity=INFO.

c. ist anschließend TUC_KON_304 „Netzwerk-Routen einrichten" aufzurufen

5. „Entfernen von nicht mehr gültigen angeschlossenen Netzen des Gesundheitswesens mit WANDA Basic"
Ist ein Netz in der neuen Datei gegenüber der alten Datei nicht mehr vorhanden, so:

   a. a) sind alle diesbezüglichen Daten zu entfernen und die Änderungen direkt aktiv zu schalten (Anpassung ANLW_AKTIVE_BESTANDSNETZE, DHCP_AKTIVE_BESTANDSNETZE_ROUTES, DNS_SERVERS_BESTANDSNETZE).

   b. b) ist anschließend TUC_KON_304 „Netzwerk-Routen einrichten" aufzurufen.

6. Protokollierung der heruntergeladenen Version von Bestandsnetze.xml durch Aufruf von
TUC_KON_271 „Schreibe Protokolleintrag" {
   topic = „KSR/UPDATE_KONFIG";
   eventType = Op;
   severity = Info;
   parameters = („AlteVersion=$aktuelleVersion,
                NeueVersion=/Infrastructure/Version
")}

Varianten/Alternativen	Keine
Fehlerfälle	(→ 1-5) Es ist ein unerwarteter Fehler aufgetreten; Fehlercode: 4198
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 376: Tab_Kon_726 Fehlercodes TUC_KON_283 „Infrastruktur Konfiguration aktualisieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4198	Technical	Error	Beim Übernehmen der angeschlossenen Netze des Gesundheitswesens mit WANDA Basic ist ein Fehler aufgetreten.

[<=]

4.3.10.4 Interne TUCs, auch durch Fachmodule nutzbar

4.3.10.4.1 TUC_KON_285 „UpdateInformationen für Fachmodul beziehen"

TIP1-A_6018 - TUC_KON_285 „UpdateInformationen für Fachmodul beziehen"
Der Konnektor MUSS den technischen Use Case TUC_KON_285 „UpdateInformationen für Fachmodul beziehen" umsetzen.

Tabelle 377: TAB_KON_833 – TUC_KON_285 „UpdateInformationen für Fachmodul beziehen"

Element	Beschreibung
Name	TUC_KON_285 „UpdateInformationen für Fachmodul beziehen"
Beschreibung	Dieser TUC ermittelt vom zentralen Konfigurationsdienst für ein Fachmodul die verfügbaren UpdateInformationen eines angegebenen SW-Pakets.

Auslöser	• Aufruf durch Fachmodul
Vorbedingungen	• Verbindung zum VPN-Konzentrator der TI wurde erfolgreich aufgebaut
Eingangsdaten	• productVendorID [String] - (Identifiziert den Hersteller des Produkts, für welches auf Updates geprüft werden soll.) • productCode [String] – (Identifiziert das Produkt zusammen mit ProductVendorID, für welches auf Updates geprüft werden soll.) • hwVersion [String] (Identifiziert die Hardware zusammen mit ProductCode und ProductVendorID, für welches auf Updates geprüft werden soll. [gemSpec_OM] beschreibt dieses Element ausführlich.) • fwVersion [String] aktuell im Produkt verwendete Firmwareversion Hinweis: Definition von productVendorID, productCode, hwVersion, fwVersion (entspricht FWVersion) siehe [gemSpec_KSR#TIP1-A_3331]
Komponenten	Konnektor, Konfigurationsdienst
Ausgangsdaten	• listOfUpdates [listUpdatesResponse] Liste von Update Informationen der verfügbaren Pakete für das angegebene Produkt; Datentyp listUpdatesResponse definiert in Konfigurationsdienst.xsd siehe [gemSpec_KSR]
Nachbedingungen	keine
Standardablauf	Der Konnektor MUSS folgende Schritte durchlaufen: 1. Der Konnektor MUSS die TLS-Verbindung zum Konfigurationsdienst anhand des in MGM_KSR_FIRMWARE_URL angegebenen Wertes aufbauen. Dabei MUSS er das durch den Server präsentierte Zertifikat mittels TUC_KON_037 „Zertifikat prüfen“ { certificate = C.ZD.TSL-S; qualifiedCheck = not_required; offlineAllowNoCheck = true; policyList = oid_zd_tls_s; intendedKeyUsage = intendedKeyUsage(C.ZD.TSL-S); intendedExtendedKeyUsage = id-kp-serverAuth; validationMode = OCSP} auf Gültigkeit prüfen.

	Das Server-Zertifikat C.ZD.TLS-S MUSS für den Konfigurationsdienst ausgestellt sein. 2. Belegen von listUpdatesRequest mit den korrekten Werten für ProductVendorID, ProductCode, HardwareVersion und FirmwareVersion = fwVersion 3. Aufruf von I_KSRS_Download::list_Updates gemäß [gemSpec_KSR#TIP1-A_3331] 4. Beenden der TLS-Verbindung
Varianten/Alternativen	Keine
Fehlerfälle	Fehler in den folgenden Schritten des Ablaufs führen zu: (→1) Konfigurationsdienst nicht erreichbar, Fehlercode: 4188 (→1) Serverzertifikat ist nicht C.ZD.TLS_S, Fehlercode: 4189 (→3) Fehler beim Beziehen der Updatelisten, Fehlercode: 4190
Nichtfunktionale Anforderungen	Der Konnektor muss die Vorgaben aus [gemSpec_Krypt#3.3.2] für TLS-Verbindungen und hinsichtlich ECC-Migration die Vorgaben aus [gemSpec_Krypt#5] befolgen.
Zugehörige Diagramme	keine

Tabelle 378: TAB_KON_834 Fehlercodes TUC_KON_285 „UpdateInformationen für Fachmodul beziehen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases, sowie der Fehlercodes von „I_KSRS_Download::listUpdates Response" können folgende weitere Fehlercodes auftreten:			
4188	Technical	Error	Konfigurationsdienst nicht erreichbar, konfigurierte Adresse kontrollieren.
4189	Security	Fatal	Konfigurationsdienst liefert falsches Zertifikat
4190	Technical	Error	Fehler beim Beziehen der Updatelisten

[<=]

4.3.10.4.2 TUC_KON_286 „Paket für Fachmodul laden"

TIP1-A_6019 - TUC_KON_286 „Paket für Fachmodul laden"
Der Konnektor MUSS den technischen Use Case TUC_KON_286 „Paket für Fachmodul laden" umsetzen.

Tabelle 379: TAB_KON_835 – TUC_KON_286 „Paket für Fachmodul laden“

Element	Beschreibung
Name	TUC_KON_286 „Paket für Fachmodul laden“
Beschreibung	Dieser TUC lädt ein bestimmtes SW-Paket für ein Fachmodul vom zentralen Konfigurationsdienst.
Auslöser	Aufruf durch Fachmodul
Vorbedingungen	• Verbindung zum VPN-Konzentrator der TI wurde erfolgreich aufgebaut
Eingangsdaten	• filename (Filename des SW-Pakets, welches vom KSR geladen werden soll)
Komponenten	Konnektor, Konfigurationsdienst
Ausgangsdaten	• swPackage (das durch filename am KSR identifizierte SW-Paket wurde heruntergeladen)
Nachbedingungen	keine
Standardablauf	1. Der Konnektor MUSS die TLS-Verbindung zum Konfigurationsdienst anhand des in MGM_KSR_FIRMWARE_URL angegebenen Wertes aufbauen. Dabei MUSS er das durch den Server präsentierte Zertifikat mittels TUC_KON_037 „Zertifikat prüfen“ { certificate = C.ZD.TSL-S; qualifiedCheck = not_required; offlineAllowNoCheck = true; policyList = oid_zd_tls_s; intendedKeyUsage = intendedKeyUsage(C.ZD.TSL-S); intendedExtendedKeyUsage = id-kp-serverAuth; validationMode = OCSP} auf Gültigkeit prüfen. Das Server-Zertifikat C.ZD.TLS-S MUSS für den Konfigurationsdienst ausgestellt sein. 2. Herunterladen der Softwarepakets swPackage mittels I_KSRS_Download::get_File (MGM_KSR_FIRMWARE_URL /$filename) 3. Beenden der TLS-Verbindung 4. swPackage an Aufrufer zurückgeben
Varianten/Alternativen	keine
Fehlerfälle	(→ 1) Verbindung zum KSR konnte nicht aufgebaut werden; Fehlercode: 4188 (→ 1) Serverzertifikat ist nicht C.ZD.TLS_S, Fehlercode: 4189 (→ 2) Wenn Größe des Pakets größer als 25MB, Fehlercode: 4242

	(→ 2) Sonstige Fehler beim Download: Das Paket konnte nicht geladen werden, Fehlercode: 4238
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 380: TAB_KON_836 Fehlercodes TUC_KON_286 „Paket für Fachmodul laden"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4188	Technical	Error	Konfigurationsdienst nicht erreichbar, konfigurierte Adresse kontrollieren.
4189	Security	Fatal	Konfigurationsdienst liefert falsches Zertifikat
4238	Technical	Error	Der Download des Pakets vom KSR ist fehlgeschlagen.
4242	Technical	Error	Der Download des Pakets vom KSR ist fehlgeschlagen. Das Paket ist größer als 25MB.

[<=]

4.3.10.5 Operationen an der Außenschnittstelle

Keine.

4.3.10.6 Betriebsaspekte

A_21899 - Kein Restart des Konnektors nach Aktualisierung der Bestandsnetze.xml

Der Konnektor DARF NICHT nach einer Aktualisierung der Datei Bestandsnetze.xml einen Restart durchführen.

[<=]

A_21900 - Minimale Anzahl von Reboots

Der Konnektor MUSS die Anzahl der Reboots minimieren.**[<=]**

4.3.10.6.1 TUC_KON_284 KSR-Client initialisieren

TIP1-A_5938 - TUC_KON_284 „KSR-Client initialisieren"

Der Konnektor MUSS in der Bootup-Phase TUC_KON_284 „KSR-Client initialisieren" durchlaufen.

Tabelle 381: TAB_KON_864 – TUC_KON_284 „KSR-Client initialisieren"

Element	Beschreibung
Name	TUC_KON_284 "KSR-Client initialisieren"

Beschreibung	Der Konnektor muss während des Bootups die Downloadpunkte für Konfigurationsdaten und Firmware ermitteln.
Eingangsanforderung	Keine
Auslöser und Vorbedingungen	Bootup Verbindung zum VPN-Konzentrator TI muss aufgebaut sein
Eingangsdaten	Keine
Komponenten	Konnektor
Ausgangsdaten	• MGM_KSR_KONFIG_URL • MGM_KSR_FIRMWARE_URL
Standardablauf	- Falls MGM_LU_ONLINE=Enabled: - Durch DNS-Anfragen an den DNS-Forwarder zur Auflösung der SRV-RR und TXT-RR mit den Bezeichnern „_ksrkonfig._tcp.ksr.<TOP_LEVEL_DOMAIN_TI>„ und „_ksrfirmware._tcp.ksr.<TOP_LEVEL_DOMAIN_TI>„ erhält der Konnektor URLs der Downloadpunkte des KSR für Konfigurationsdaten (MGM_KSR_KONFIG_URL) und für Firmware (MGM_KSR_FIRMWARE_URL).
Varianten/Alternativen	Keine
Fehlerfälle	Keine
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 382: TAB_KON_822 Fehlercodes TUC_KON_284 „KSR-Client initialisieren"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können keine weiteren Fehlercodes auftreten.			

[<=]

TIP1-A_4835-02 - Konfigurationswerte des KSR-Client
Der Administrator MUSS die in TAB_KON_670 aufgelisteten Parameter über die Managementschnittstelle konfigurieren und die in TAB_KON_820 aufgelisteten Parameter ausschließlich einsehen können.

Tabelle 383: TAB_KON_670 Konfigurationsparameter der Software-Aktualisierung

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
MGM_KSR_ AUTODOWNLOAD	Enabled/ Disabled	Der Administrator MUSS den automatischen Download verfügbarer Update-Pakete über den Konfigurationsparameter MGM_KSR_AUTODOWNLOAD an- und abschalten können. Default-Wert: Enabled
MGM_KSR_SHOW_ TRIAL_UPDATES	Enabled / Disabled	Der Administrator MUSS einschalten können, dass zusätzlich zur Anzeige von Update-Paketen für den Online-Produktivbetrieb auch die Anzeige von Erprobungs-Update-Paketen erfolgt. Wenn MGM_KSR_SHOW_TRIAL_UPDATES von Disabled auf Enabled gesetzt wird, muss ein Warnhinweis angezeigt werden, dass die Installation von Erprobungs-Update-Paketen nur für Teilnehmer der Erprobungen vorgesehen ist. Default-Wert: Disabled
MGM_KSR_AUTO_UPDATE	Enabled / Disabled	Der Administrator MUSS pro Gerät (Konnektor und Kartenterminals) das automatische Softwareupdate ein- und ausschalten können. Default-Wert: Enabled Falls MGM_KSR_AUTO_UPDATE=Enabled wird MGM_KSR_ AUTODOWNLOAD=Enabled gesetzt.
MGM_KSR_AUTO_ UPDATE_TIME	Wochentag / Uhrzeit Oder täglich / Uhrzeit	Der Administrator MUSS den Wochentag und die Uhrzeit einstellen können, wann automatische Softwareupdates durchgeführt werden. Als Wochentag MUSS es neben den einzelnen Wochentagen auch einen Wert für eine tägliche Prüfung auf Aktualität und gegebenenfalls Durchführung von Softwareupdates geben. Default-Wert: Montag / 1:00 Uhr

Tabelle 384: TAB_KON_820 Einsehbare Konfigurationsparameter der Software-Aktualisierung

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
MGM_KSR_ KONFIG_URL	URL	SOAP-Endpunkt des Konfigurationsdienstes zum Download von Konfigurationsdaten
MGM_KSR_ FIRMWARE_URL	URL	SOAP-Endpunkt des Konfigurationsdienstes zum Download der Firmware

[<=]

Hinweis: Die Adressen des Konfigurationsdienstes werden im Rahmen des VPN-Verbindungsaufbaus ermittelt (siehe [gemSpec_VPN_ZugD#5.1.1.2 TUC_VPN-ZD_0001])

TIP1-A_6025 - Zugang zur TI sperren, wenn Deadline für kritische FW-Updates erreicht
Der Konnektor MUSS täglich überprüfen, ob unter den auf die aktuelle Konnektor-Firmware anwendbaren Updates ein Update mit FWPriority = „Kritisch" ist, dessen Deadline (entspricht `UpdateInformation/DeploymentInformation/Deadline`) abgelaufen ist, d.h. Deadline <= Systemzeit. In diesem Fall MUSS der Konnektor den Verbindungsaufbau zur TI Plattform verhindern, bestehende Verbindungen in die TI abbauen und den kritischen Betriebszustand EC_FW_Not_Valid_Status_Blocked annehmen.
[<=]

TIP1-A_4836 - Automatische Prüfung und Download von Update-Paketen
Der Konnektor MUSS täglich die folgenden Schritte durchführen:

1. TUC_KON_282 „UpdateInformationen beziehen" aufrufen.
2. pro zurück geliefertem Listeneintrag prüfen, ob eine neuere Version enthalten ist, als auf dem zugehörigen Gerät (Konnektor selbst oder Kartenterminal) vorhanden
3. Ist für wenigstens ein Gerät eine neuere Version vorhanden, MUSS der Konnektor darüber via
   TUC_KON_256 „Systemereignis absetzen" {
   topic = „KSR/UPDATES_AVAILABLE";
   eventType = Op;
   severity = Info;
   parameters = (<Param>);
   doLog=false }
   informieren. Je gefundenem Update MUSS <Param> mit folgenden Werten belegt sein:
   <Param> = „ProductVendorID= $UpdateInformation/ProductVendorID;
   ProductCode= $UpdateInformation/ProductCode;
   ProductName=$UpdateInformation/ProductName;
   FirmwareVersion=$UpdateInformation/FirmwareVersion;
   Deadline=$UpdateInformation/DeploymentInformation/Deadline;
   FWPriority=$UpdateInformation/Firmware/FWPriority;
   FirmwareReleaseNotes=
   $UpdateInformation/Firmware/FirmwareReleaseNotes"
4. Die listUpdateResponse mit neueren Firmwareversionen MÜSSEN für eine spätere Einsichtnahme durch den Administrator bereitgehalten werden (via (TIP1-A_4837) „Übersichtsseite des KSR-Client). Ein neuerlicher Abruf dieser Informationen DARF NICHT erforderlich sein.
5. Sofern ein Update-Paket für den Konnektor vorliegt, MUSS der Konnektor die mit diesem Paket gelieferten Parameter `Priority` (entspricht `UpdateInformation/Firmware/FWPriority)` und `Deadline` (entspricht `UpdateInformation/DeploymentInformation/Deadline)` auswerten und bei `KSR:Priority=Kritisch` persistent ablegen.
6. Sofern `MGM_KSR_AUTODOWNLOAD` = Enabled, MUSS der Konnektor bei Update-Paketen, die den Konnektor selbst betreffen, das Update-Paket mit der höchsten FirmwareVersion über I_KSRS_Download::get_Updates herunterladen.
7. Ist der Download von Update-Paketen für den Konnektor abgeschlossen, MUSS der Konnektor darüber via

TUC_KON_256 „Systemereignis absetzen“ {
topic = „KSR/UPDATE/KONNEKTOR_DOWNLOAD_END“;
eventType = Op;
severity = Info;
parameters = (<Param>)}
informieren. Je heruntergeladenem FW-Paket MUSS <Param> mit folgenden Werten belegt sein:
<Param> = „ProductVendorID= $UpdateInformation/ProductVendorID;
ProductCode= $UpdateInformation/ProductCode;
ProductName=$UpdateInformation/ProductName;
FirmwareVersion=$UpdateInformation/Firmware/FWVersion;
Deadline=$UpdateInformation/DeploymentInformation/Deadline;
FWPriority=$UpdateInformation/Firmware/FWPriority;
FirmwareReleaseNotes
=$UpdateInformation/Firmware/FirmwareReleaseNotes“

8. Sofern `MGM_KSR_AUTODOWNLOAD` = Enabled, SOLL der Konnektor bei Update-Paketen, die Kartenterminals betreffen, pro KT-Modell das Update-Paket mit der höchsten FirmwareVersion über I_KSRS_Download::get_Updates herunterladen.

Der Konnektor MUSS immer nur die neusten Update-Pakete für eine Nutzung vorhalten. Eventuell vorhandene ältere, nicht genutzte Update-Pakete KÖNNEN überschrieben werden.**[**<=**]**

TIP1-A_4836-02 - ab PTV4: Automatische Prüfung und Download von Update-Paketen
Der Konnektor MUSS täglich die folgenden Schritte durchführen:

1. TUC_KON_282 „UpdateInformationen beziehen“ aufrufen.
2. pro zurück geliefertem Listeneintrag prüfen, ob eine neuere Version enthalten ist, als auf dem zugehörigen Gerät (Konnektor selbst oder Kartenterminal) vorhanden
3. Ist für wenigstens ein Gerät eine neuere Version vorhanden, MUSS der Konnektor darüber via
TUC_KON_256 „Systemereignis absetzen“ {
topic = „KSR/UPDATES_AVAILABLE“;
eventType = Op;
severity = Info;
parameters = (<Param>);
doLog=false }
informieren. Je gefundenem Update MUSS <Param> mit folgenden Werten belegt sein:
<Param> = „ProductVendorID= $UpdateInformation/ProductVendorID;
ProductCode= $UpdateInformation/ProductCode;
ProductName=$UpdateInformation/ProductName;
FirmwareVersion=$UpdateInformation/FirmwareVersion;
Deadline=$UpdateInformation/DeploymentInformation/Deadline;
FWPriority=$UpdateInformation/Firmware/FWPriority;
FirmwareReleaseNotes=
$UpdateInformation/Firmware/FirmwareReleaseNotes“
4. Ist für wenigstens ein Gerät eine neuere Version vorhanden, MUSS der Konnektor in den Betriebszustand EC_FW_Update_Available übergehen.

5. Die listUpdateResponse mit neueren Firmwareversionen MÜSSEN für eine spätere Einsichtnahme durch den Administrator bereitgehalten werden (via (TIP1-A_4837) „Übersichtsseite des KSR-Client). Ein neuerlicher Abruf dieser Informationen DARF NICHT erforderlich sein.
6. Sofern ein Update-Paket für den Konnektor selbst vorliegt, MUSS der Konnektor die mit diesem Paket gelieferten Parameter `Priority` (entspricht `UpdateInformation/Firmware/FWPriority)` und `Deadline` (entspricht `UpdateInformation/DeploymentInformation/Deadline)` auswerten und bei `KSR:Priority=Kritisch` persistent ablegen.
7. Sofern `MGM_KSR_AUTODOWNLOAD` = Enabled, MUSS der Konnektor bei Update-Paketen, die den Konnektor selbst betreffen, das Updatepaket mit der höchsten FirmwareVersion über I_KSRS_Download::get_Updates herunterladen, falls das Update-Paket nicht bereits von einem vorherigen Download auf dem Konnektor vorhanden ist.
8. Sofern I_KSRS_Download::get_Updates den http Status Code 503 Server Unavailable zurückgibt, MUSS der Konnektor die Informationen aus dem zurückgegebenen Retry-After Header nutzen, um den Zeitpunkt des Retry zu bestimmen.
9. Ist der Download von Update-Paketen für den Konnektor abgeschlossen, MUSS der Konnektor darüber via

   TUC_KON_256 „Systemereignis absetzen“ {
   topic = „KSR/UPDATE/KONNEKTOR_DOWNLOAD_END“;
   eventType = Op;
   severity = Info;
   parameters = (<Param>)}
   informieren. Je heruntergeladenem FW-Paket MUSS <Param> mit folgenden Werten belegt sein:
   <Param> = „ProductVendorID= $UpdateInformation/ProductVendorID;
   ProductCode= $UpdateInformation/ProductCode;
   ProductName=$UpdateInformation/ProductName;
   FirmwareVersion=$UpdateInformation/Firmware/FWVersion;
   Deadline=$UpdateInformation/DeploymentInformation/Deadline;
   FWPriority=$UpdateInformation/Firmware/FWPriority;
   FirmwareReleaseNotes
   =$UpdateInformation/Firmware/FirmwareReleaseNotes“
10. Sofern `MGM_KSR_AUTODOWNLOAD` = Enabled, SOLL der Konnektor bei Update-Paketen, die Kartenterminals betreffen, pro KT-Modell das Updatepaket mit der höchsten FirmwareVersion über I_KSRS_Download::get_Updates herunterladen, falls das Update-Paket nicht bereits von einem vorherigen Download auf dem Konnektor vorhanden ist.
11. Sofern I_KSRS_Download::get_Updates den http Status Code 503 Server Unavailable zurückgibt, MUSS der Konnektor die Informationen aus dem zurückgegebenen Retry-After Header nutzen, um den Zeitpunkt des Retry zu bestimmen.

Der Konnektor MUSS immer nur die neusten Update-Pakete für eine Nutzung vorhalten. Eventuell vorhandene ältere, nicht genutzte Update-Pakete KÖNNEN überschrieben werden.

Nach einem erfolgreichen Download DÜRFEN die Namen der Dateien eines Update-Paketes beim Abspeichern NICHT verändert werden.**[<=]**

TIP1-A_7220 - Konnektoraktualisierung File Transfer Ranges
Der Konnektor KANN für den Download von Update-Paketen über I_KSRS_Download::get_Updates die Option Range Requests [RFC7233#3.1] zur Fortsetzung von unterbrochenen Transfers nutzen.**[<=]**

TIP1-A_4837 - Übersichtsseite des KSR-Client
Die Administrationsoberfläche des KSR-Clients MUSS dem Administrator eine Übersichtseite anbieten, die einen Geräteeintrag für den Konnektor selbst, sowie eine Liste von Geräteeinträgen für jedes Kartenterminal (CT) aus CTM_CT_LIST mit CT.IS_PHYSICAL=Ja und CT.CORRELATION>=„gepairt" enthält.
Der Administrator MUSS die Liste der Kartenterminals nach Kartenterminalmodellen gruppieren können (gleiche Werte für ProductVendorID, ProductCode, HardwareVersion und FirmwareVersion).
Je Geräteeintrag MÜSSEN die über „Automatische Prüfung und Download von Update-Paketen" ermittelten listUpdatesResponse bereitstehen.
Je Geräteeintrag MUSS die Version der aktuell installierten Software dargestellt werden. Sind Bestandteile der installierten Software unabhängig aktualisierbar, so MUSS für jedes der Bestandteile die Version angezeigt werden.
Der Administrator MUSS eine Aktualisierung aller listUpdatesResponse über TUC_KON_282 „UpdateInformationen beziehen" auslösen können.
Geräteeinträge, die über listUpdatesResponse mit neuerer Firmwareversion als das zugehörige Gerät verfügen, MÜSSEN hervorgehoben werden.
Je Geräteeintrag MUSS die Zugehörigkeit der installierten Software und der Software-Updates zum Online-Produktivbetrieb oder zu einer Erprobung (inklusive Name der Erprobung) dargestellt werden.
[<=]

TIP1-A_4838 - Einsichtnahme in Update-Informationen
Für alle Geräteeinträge MUSS der Administrator zu den listUpdatesResponse sowohl die FirmwareGroupReleaseNotes als auch jedes enthaltene UpdateInformation-Element einsehen können. Dazu MUSS der Konnektor

- alle Felder der Struktur verständlich umsetzen und strukturiert anzeigen (inkl. der Notes für jedes Firmwarefiles- und Documentationsfiles-Element)
- jedes über das Documentationfiles-Element erreichbare Dokument auf Anforderung des Administrator herunterladen und anzeigen. Es MÜSSEN dabei mindestens die folgenden Dokumentenformate zur Anzeige gebracht werden können: Text, PDF, JPEG, TIFF

[<=]

TIP1-A_4839-01 - Festlegung der durchzuführenden Updates
Der Administrator MUSS in der Übersichtsliste einzelne Geräteeinträge bzw. Gruppen mit der jeweils anzuwendenden UpdateInformation für die Durchführung eines Updates markieren können.
Alternativ MUSS der Administrator neben der Markierung je Geräteeintrag bzw. Gruppe Update-Pakete lokal einspielen können (etwa durch ein Upload- bzw. Download-Interface in der Administrationsoberfläche).
Je Geräteeintrag MUSS der Administrator einen individuellen Ausführungszeitpunkt für die Durchführung des Updates einstellen können.
Der Administrator MUSS für den Geräteeintrag Konnektor festlegen können, ob dieses Update erst gestartet werden darf, wenn zuvor alle festgelegten KT-Updates erfolgreich durchlaufen wurden.
Der Administrator MUSS zu jeder Zeit die gerätebezogene Festlegung für ein Update

ändern bzw. löschen können, sofern dieses konkrete Update noch nicht begonnen wurde. Je Geräteeintrag MUSS der Administrator automatische Softwareupdates aktivieren und deaktivieren können.
[<=]

TIP1-A_4840-01 - Manuelles Auslösen der durchzuführenden Updates
Der Administrator MUSS für die Liste der markierten Geräteeinträge ein gesammeltes Update auslösen können. Dieses MUSS nach folgendem Muster ablaufen:

1. Alle Kartenterminaleinträge abarbeiten. Pro markiertem Kartenterminal:

- Wenn Ausführungszeitpunkt nicht gesetzt:
  Anwenden des definierten Updates mittels TUC_KON_281 „Kartenterminalaktualisierung anstoßen"
- Wenn Ausführungszeitpunkt gesetzt:
  Anwenden des definierten Updates mittels TUC_KON_281 sobald der Ausführungszeitpunkt erreicht ist oder, sofern der Konnektor zum Ausführungszeitpunkt nicht in Betrieb war, überschritten wurde. Konnte das Kartenterminal nicht erreicht werden, so MUSS das gesetzte Update im KSR-Client für eine spätere Anwendung erhalten bleiben (wird ereignisgesteuert neu ausgelöst).

2. Sofern die KonnektorUpdate-Abhängigkeit von KT-Updates nicht gesetzt wurde oder wenn alle vorgesehenen Kartenterminal-Updates durchgeführt wurden, MUSS das Konnektor-Updates mittels TUC_KON_280 „Konnektoraktualisierung durchführen" wie folgt begonnen werden:

- wenn Ausführungszeitpunkt nicht gesetzt: TUC-Aufruf direkt
- wenn Ausführungszeitpunkt gesetzt: TUC-Aufruf direkt sobald der Ausführungszeitpunkt erreicht ist oder, sofern der Konnektor zum Ausführungszeitpunkt nicht in Betrieb war, überschritten wurde

Wenn der Administrator ein Erprobungs-Update zur Installation auswählt, MUSS er über einen Warnhinweis darüber informiert werden,

- dass es sich um ein Erprobungs-Update handelt,
- für welche Erprobung es vorgesehen ist,
- dass das Update-Paket nur installiert werden sollte, wenn die Institution oder Organisation des Gesundheitswesens an der Erprobung teilnimmt,

dass, falls die Institution oder Organisation des Gesundheitswesens nicht an der Erprobung teilnimmt und dennoch das Update installiert wird, es zu funktionalen Einschränkungen des Konnektors kommen kann.**[<=]**

Wurde die ECC-Migration durchgeführt, so muss sichergestellt werden, dass der Konnektor auch wieder in den ursprünglichen Zustand, d.h. den Zustand vor der ECC-Migration (TI-Vertrauensanker für RSA und Firmware vor der ECC-Migration), zurückgesetzt werden kann.

A_18390 - Automatisches Auslösen der durchzuführenden Updates
Wenn für mindestens ein Gerät das automatische Softwareupdate aktiviert ist, MUSS der Konnektor zur MGM_KSR_AUTO_UPDATE_TIME die Updates nach folgendem Muster durchführen:

- Alle Geräte (Kartenterminals und Konnektor), für die MGM_KSR_AUTO_UPDATE=Enabled ist, werden markiert
- Alle Kartenterminaleinträge abarbeiten

- Pro markiertem Kartenterminal: Anwenden des automatischen Updates mittels TUC_KON_281 „Kartenterminalaktualisierung anstoßen"

- Sofern die Konnektorupdate-Abhängigkeit von KT-Updates nicht gesetzt wurde oder wenn alle vorgesehenen Kartenterminal-Updates durchgeführt wurden, MUSS für einen markierten Konnektor das Konnektor-Update mittels TUC_KON_280 „Konnektoraktualisierung durchführen" begonnen werden.

[<=]

A_18391 - Automatisches Updates nicht nachholen
Sofern der Konnektor zu MGM_KSR_AUTO_UPDATE_TIME nicht in Betrieb war, DÜRFEN die automatischen Updates später NICHT nachgeholt werden.**[<=]**

A_18779 - Hinweise in KSR Update Paket zu Auto-Update
Wenn mit einem Update erstmalig MGM_KSR_AUTO_UPDATE=Enabled aktiv wird, MUSS der Konnektorhersteller über das entsprechende KSR-Paket den Admin an der Konnektor Oberfläche darauf hinweisen, dass mit diesem Update der automatische Softwareupdate aktiv wird.
[<=]

A_20531 - Größe der Bestandsnetze.xml
Der Konnektor MUSS eine Bestandsnetze.xml mit einer Größe von mindestens 3 MByte und 2000 Netzen (XML Element <Network>) verarbeiten können.**[<=]**

4.3.11 Konnektorstatus

TIP1-A_5542 - Konnektor, Funktion zur Prüfung der Erreichbarkeit von Systemen
Der Konnektor MUSS an der Managementschnittstelle eine Funktion anbieten, die es ermöglicht die Erreichbarkeit von Systemen durch Eingabe der IP-Adresse oder des FQDN zu prüfen. Das Ergebnis des Tests MUSS angezeigt werden.
[<=]

4.4 Hardware-Merkmale des Konnektors

TIP1-A_4841 - Hardware für Dauerbetrieb
Der Konnektor MUSS sowohl in seiner Stromversorgung als auch in seinen restlichen Hardwarekomponenten auf einen 24x7-Dauerbetrieb ausgelegt sein.
Der Hersteller DARF NICHT davon ausgehen oder gar in seiner Guidance darauf verweisen, dass der Konnektor mehrere Stunden am Tag nicht betrieben wird.
[<=]

Diese Anforderung verlangt keinen Schutz gegen Stromausfall in den Betriebsräumen.

TIP1-A_4842 - Gehäuseversiegelung
Jeder Konnektor, der als Appliance (dezidierte, geschlossene Kombination aus spezifischer Hard- und Software) ausgeprägt ist, MUSS über eine fälschungssichere Gehäuseversiegelung verfügen. Die Versiegelung MUSS so angebracht werden, dass eine Öffnung des Gehäuses nicht ohne Beschädigung des Siegels erfolgen kann.
Der Konnektor MUSS die Umsetzung entsprechend der Festlegungen für das Kartenterminal nach der TR-03120 [BSI TR-03120], Kapitel bzgl. Gehäuseversiegelung 9 vornehmen.
Die optische Gestaltung der Siegel ist herstellerspezifisch.
[<=]

Die Prüfung auf Einhaltung der Versiegelungsvorgaben erfolgt nicht im Rahmen der CC-Evaluierung, sondern im Zuge der Prüfung auf funktionale Eignung.

TIP1-A_4843 - Zustandsanzeige
Im Betrieb MUSS der Zustand des Konnektors erkennbar sein. Zur Anzeige des Betriebszustandes des Konnektors SOLL es eine Signaleinrichtung (z. B. über Status-LEDs) am Konnektor geben. Falls keine Signalvorrichtung am Konnektorgehäuse verwendet wird MUSS es eine softwareseitige Lösung über das Managementinterface geben. Bei verbauter Hardware-Signalgebung KANN eine softwareseitige Lösung zusätzlich angeboten werden.
Es MÜSSEN mindestens folgende angezeigt werden:

- Power ON,
- Link Status pro physischer Netzwerkschnittstelle
- Fehler/Kritischer Betriebszustand gemäß Kapitel 3.3

Es SOLLEN folgende Zustände angezeigt werden:

- Status pro IPsec-Verbindung

[<=]

TIP1-A_4844-02 - Ethernet-Schnittstellen
Der Konnektor MUSS mindestens zwei Ethernetinterfaces nach [IEEE802.3] als physikalische Schnittstellen zur Verfügung stellen.
[<=]

TIP1-A_4845 - Verwendungsumgebung - Klima
Als normaler Einsatzort wird für den Konnektor ein Büroraum angenommen. Der Konnektor MUSS die in Tabelle TAB_KON_671 aufgeführten Anforderungen erfüllen, welche unter der Annahme des normalen Einsatzortes erhoben werden.

Tabelle 385: TAB_KON_671 Anforderungen Klima

Prüfung Klima
Trockene Wärme (Dry Heat) nach DIN EN 60068-2-2 Methode Bb wird für die Bedingungen als obere Lagertemperatur von 55°C und einer Beanspruchungsdauer von 16 h geprüft und die Funktionsfähigkeit MUSS bestätigt werden.
Kälte (Cold) nach DIN EN 60068-2-1 Methode Ab wird für die Bedingungen als untere Lagertemperatur von -10°C und einer Beanspruchungsdauer von 16 h geprüft und die Funktionsfähigkeit MUSS bestätigt werden.
Nach den beiden oben genannten Belastungen durch extreme Lagertemperaturen und der Nachbehandlungsdauer von 1 h MUSS die Funktionsfähigkeit des Konnektors gewährleistet sein, was durch Funktionsprüfungen nachzuweisen ist.
Die Funktionsfähigkeit im Betrieb MUSS bei einer oberen Temperatur von 40°C über eine Dauer von 24 h gewährleistet sein. Dies wird für den Konnektor durch Prüfung nach DIN EN 60068-2-2 Methode Bb bei gleichzeitigen Funktionsprüfungen nachgewiesen.

[<=]

TIP1-A_4846 - Verwendungsumgebung – Vibration
Die durch Vibrationen und mechanische Schockbelastungen auftretenden Belastungen MÜSSEN vom Konnektor schadensfrei gemäß IEC 68-2 Methode nach den Anforderungen aus TAB_KON_672 absolviert, geprüft und nachgewiesen werden.

Tabelle 386: TAB_KON_672 Anforderungen Vibration

Prüfung Vibration
Sinusförmige Schwingungstests (Vibration, sinusoidal) nach DIN EN 60068-2-6 Methode Fc in drei senkrecht stehenden Achsen in einem Frequenzbereich von 2 Hz bis 200 Hz üblicherweise 1 h je Achse MÜSSEN erfolgreich nachgewiesen werden. Bis zu einer Frequenz von 9 Hz wird dabei mit einer konstanten Amplitude von 1,5 mm belastet, darüber bis zur oberen Frequenz wird mit einer konstanten Beschleunigung von 5 m/s^2 (0,5 g) belastet.
Es MÜSSEN mechanische Schockprüfungen (Shock) nach DIN EN 60068-2-27 Methode Ea in drei senkrecht stehenden Achsen (sechs Richtungen) erfolgreich nachgewiesen werden. Es wird dabei für jede Achse mit 3 positiven und 3 negativen Schocks mit 150 m/s^2 (15 g) Amplitude und einer Dauer von 11 ms belastet.
Dauerschocktests (Bump) nach DIN EN 60068-2-29 Methode Eb in drei senkrecht stehenden Achsen mit halbsinusförmigen Schocks MÜSSEN erfolgreich nachgewiesen werden. Es wird dabei für jede Achse mit 1000 positiven und 1000 negativen Schocks mit 100 m/s^2 (10 g) Amplitude und einer Dauer von 16 ms belastet.

[<=]

TIP1-A_4846-02 - ab PTV4: Verwendungsumgebung – Vibration
Die durch Vibrationen und mechanische Schockbelastungen auftretenden Belastungen MÜSSEN vom Konnektor schadensfrei gemäß IEC 68-2 Methode nach den Anforderungen aus TAB_KON_672 absolviert, geprüft und nachgewiesen werden.

Tabelle 387: TAB_KON_672 Anforderungen Vibration

Prüfung Vibration
Sinusförmige Schwingungstests (Vibration, sinusoidal) nach DIN EN 60068-2-6 Methode Fc in drei senkrecht stehenden Achsen in einem Frequenzbereich von 2 Hz bis 200 Hz üblicherweise 1 h je Achse MÜSSEN erfolgreich nachgewiesen werden. Bis zu einer Frequenz von 9 Hz wird dabei mit einer konstanten Amplitude von 1,5 mm belastet, darüber bis zur oberen Frequenz wird mit einer konstanten Beschleunigung von 5 m/s^2 (0,5 g) belastet.
Es MÜSSEN mechanische Schockprüfungen (Shock) nach DIN EN 60068-2-27 Methode Ea in drei senkrecht stehenden Achsen (sechs Richtungen) erfolgreich nachgewiesen werden. Es wird dabei für jede Achse mit 3 positiven und 3 negativen Schocks mit 150 m/s^2 (15 g) Amplitude und einer Dauer von 11 ms belastet.
Dauerschocktests (Bump) nach DIN EN 60068-2-27Methode Eb in drei senkrecht stehenden Achsen mit halbsinusförmigen Schocks MÜSSEN erfolgreich nachgewiesen werden. Es wird dabei für jede Achse mit 1000 positiven und 1000 negativen Schocks mit 100 m/s^2 (10 g) Amplitude und einer Dauer von 16 ms belastet.

[<=]

5 Anhang A – Verzeichnisse

5.1 Abkürzungen

Kürzel	Erläuterung
AMTS	Arzneimitteltherapiesicherheit
APPL DO	Application Label Data Object
CC	Common Criteria
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DO	Datenobjekt
DSL	Digital Subscriber Line
ECC	Elliptic Curve Cryptography
EVG	Evaluierungsgegenstand
gSMC-K	Security Module Card Typ K (Konnektor)
gSMC-KT	Security Module Card Typ KT (Kartenterminal)
HBA	Heilberufsausweis
HSM-B	Hardware Security Module Typ B
IAG	Internet Access Gateway
ID	Identifier
ISP	Internet Service Provider
KT	Kartenterminal
KVK	Krankenversichertenkarte
LAN	Local Area Network
MTOM	Message Transmission Optimization Mechanism

NFDM	Notfalldatenmanagement
NK	Netzkonnektor
NTP	Netzwork Time Protokoll
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PIN	Personal
PKI	Public Key Infrastructure
PP	Protection Profile
PU	Produktivumgebung
PUK	Personal Unblocking Key
QES	Qualifizierte elektronische Signatur
RU	Referenzumgebung
SIS	Secure Internet Service
SMC-B	Security Module Card Typ B
SMTBD DO	SICCT Message-To-Be-Displayed Data Object
SOAP	Standard für die Kommunikation innerhalb der WEB-Services
TI	Telematikinfrastruktur
TLS	Transport Layer Security
TSF	TOE Security Functionality
TU	Testumgebung
TUC	Technischer Use Case
URL	Uniform Resource Locator
VPN	Virtual Private Network

VSDM	Versichertenstammdatenmanagement
VZD	Verzeichnisdienst
WAN	Wide Area Network
WANDA Basic	Weitere Anwendungen für den Datenaustausch ohne Nutzung der TI oder derer kryptografischen Identitäten
WANDA Smart	Weitere Anwendungen für den Datenaustausch mit Nutzung der TI oder derer kryptografischen Identitäten für eigene Anwendungszwecke
XML	Extensible Markup Language
ZD	Zertifizierungsdienst
ZOD 2.0	Zahnärzte Online Deutschland 2.0

5.2 Glossar

Begriff	Erläuterung
Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.

Das Glossar wird als eigenständiges Dokument, vgl. [gemGlossar] zur Verfügung gestellt.

5.3 Abbildungsverzeichnis

5.4 Tabellenverzeichnis

5.5 Referenzierte Dokumente

5.5.1 Dokumente der gematik

Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument referenzierten Dokumente der gematik zur Telematikinfrastruktur. Der mit der vorliegenden Version korrelierende Entwicklungsstand dieser Konzepte und Spezifikationen wird pro Release in einer Dokumentenlandkarte definiert, Version und Stand der referenzierten Dokumente sind daher in der nachfolgenden Tabelle nicht aufgeführt. Deren zu diesem Dokument passende jeweils gültige Versionsnummer sind in der aktuellsten, von der gematik veröffentlichten Dokumentenlandkarte enthalten, in der die vorliegende Version aufgeführt wird.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Glossar
[gemKPT_Arch_TIP]	gematik: Konzept Architektur der TI-Plattform
[gemKPT_Sich_Kon]	gematik: Sicherheitskonzept Konnektor
[gemKPT_Test]	gematik: Testkonzept
[gemSpec_COS]	gematik: Spezifikation des Card Operating System (COS) – Elektrische Schnittstelle
[gemSpec_Karten_Fach_TIP]	gematik: Befüllvorschriften für die Plattformanteile der Karten der TI
[gemSpec_Kon_SigProxy]	gematik: Spezifikation Konnektor Signaturproxy
[gemSpec_eGK_ObjSys]	gematik: Spezifikation der elektronischen Gesundheitskarte eGK-Objektsystem - für Karten der Generation 2
[gemSpec_eGK_ObjSys_G2.1]	gematik: Spezifikation der elektronischen Gesundheitskarte eGK-Objektsystem - für Karten der Generation 2.1
[gemSpec_eGK_P1]	gematik: Die Spezifikation der elektronische Gesundheitskarte; Teil 1 – Spezifikation der elektrischen Schnittstelle - für Karten der Generation 1+
[gemSpec_eGK_P2]	gematik: Die Spezifikation der elektronische Gesundheitskarte; Teil 2 – Grundlegende Applikationen - für Karten der Generation 1+
[gemSpec_gSMC-K_ObjSys]	gematik: Spezifikation der gSMC-K Objektsystem
[gemSpec_gSMC-KT_ObjSys]	gematik: Spezifikation gSMC-KT Objektsystem
[gemSpec_HBA_ObjSys]	gematik: Spezifikation HBA Objektsystem

[gemSpec_Krypt]	gematik: Übergreifende Spezifikation Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur
[gemSpec_KSR]	gematik: Spezifikation Konfigurationsdienst
[gemSpec_KT]	gematik: Spezifikation eHealth-Kartenterminal
[gemSpec_Net]	gematik: Spezifikation Netzwerk
[gemSpec_OID]	gematik: Spezifikation OID
[gemSpec_OM]	gematik: Übergreifende Spezifikation Operations und Maintenance
[gemSpec_Perf]	gematik: Performance und Mengengerüst TI-Plattform
[gemSpec_PKI]	gematik: Spezifikation PKI
[gemSpec_SMC-B_ObjSys]	gematik: Spezifikation SMC-B Objektsystem
[gemSpec_VPN_ZugD]	gematik: Spezifikation VPN-Zugangsdienst
[gemSpec_VZD]	gematik: Spezifikation Verzeichnisdienst
gemGitHub_tslSig]	https://github.com/gematik/examples-TelematikInterfaces/tree/master/tslService/detachedSignature

5.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[7816–4]	ISO/IEC 7816-4: 2005 (2nd edition) Identification cards — Integrated circuit cards - Part 4: Organization, security and commands for interchange
[ALGCAT]	Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen), Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen, vom 11.12.2015 (auch online verfügbar: https://www.bundesanzeiger.de mit dem Suchbegriff „BAnz AT 01.02.2016 B5“).
[Basic Profile1.2]	Basic Profile Version 1.2 http://www.ws-i.org/Profiles/BasicProfile-1.2-2010-11-09.html
[Basic Profile2.0]	Basic Profile Version 2.0 http://www.ws-i.org/Profiles/BasicProfile-2.0-2010-11-09.html
[BSI_GK]	BSI: IT-Grundschutz-Kataloge (15. Ergänzungslieferung 2016) https://download.gsb.bund.de/BSI/ITGSK/ IT-Grundschutz-Kataloge_2016_EL15_DE.pdf

[BSI-TR-03111]	Technical Guideline BSI TR-03111 Elliptic Curve Cryptography, Version 2.10, Date: 2018-06-01 https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03111/BSI-TR-03111_V-2-1_pdf.pdf?__blob=publicationFile&v=2
[BSI-TR03114]	BSI (22.10.2007): Technische Richtlinie – Stapelsignatur mit dem Heilberufsausweis; Version 2.0 https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03114/BSI-TR-03114.pdf?__blob=publicationFile&v=1
[BSI TR-03120]	BSI (23.10.2007): BSI - Technische Richtlinie – Sichere Kartenterminalidentität (Betriebskonzept); Version 1.0 https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03120/BSI-TR-03120.pdf?__blob=publicationFile&v=1
[CAdES]	ETSI: Electronic Signature Formats, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V2.2.1, via http://www.etsi.org
[Canon XML1.1]	Canonical XML Version 1.1 http://www.w3.org/TR/2008/REC-xml-c14n11-20080502/
[CDA]	ISO/HL7 27932:2009 Data Exchange Standards -- HL7 Clinical Document Architecture, Release 2
[CDA-Sig]	Erstellung von XML-Signaturen für Dokumente nach Clinical Documents Architecture – R2, Elektronische Signatur von Arztbriefen, Ärztekammern in NRW im Auftrag der Bundesärztekammer, Version 1.6 vom 19.04.2010
[COMMON_PKI]	Common PKI Specifications for Interoperable Applications Version 2.0, 20 January 2009 http://www.t7ev.org/themen/entwickler/common-pki-v20-spezifikation.html ISIS-MTT Core Specification, 2004, Version 1.1 https://www.teletrust.de/fileadmin/files/ ISIS-MTT_Profile_SigGOptions_v1.1.pdf
[CMS]	Cryptographic Message Syntax (CMS), September 2009 http://tools.ietf.org/html/rfc5652
[DIN 66003]	DIN 66003:1999 Informationsverarbeitung; 7-Bit-Code
[HPC-P1]	Spezifikation des elektronischen Heilberufsausweises Version 2.3.2, 05.08.2009, Teil I: Kommandos, Algorithmen und Funktionen der COS Plattform

[HPC-P2]	Spezifikation des elektronischen Heilberufsausweises Version 2.3.2, 05.08.2009, Teil II: HPC - Anwendungen und Funktionen
[HPC-P3]	Spezifikation des elektronischen Heilberufsausweises Version 2.3.2, 05.08.2009, Teil III: SMC - Anwendungen und Funktionen
[HüKo06]	BSI (2006): Hühnlein, Detlef/Korte, Ulrike:Grundlagen der elektronischen Signatur
[IEEE 802.3]	Technical Committee Computer Communications of the IEEE Computer Society, USA (1985): IEEE standards for local area networks: carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications ISBN: 0-7381-4253-0
[ISO 8601]	International Organization for Standardization (2006-09): Data elements and interchange formats -- Information interchange -- Representation of dates and times
[KVK]	Spitzenverbände der Krankenkassen, Kassenärztliche Bundesvereinigung und Kassenzahnärztlichen Bundesvereinigung (gültig ab 25. November 2009): Technische Spezifikation der Versichertenkarte Version: 2.08
[MIME]	RFC 2045, RFC 2046, RFC 2047, RFC 2048, RFC 2049
[NTPv4]	Internet Engineering Task Force (IETF) (06/2010): Network Time Protocol Version 4: Protocol and Algorithms Specification http://www.ietf.org/rfc/rfc5905.txt
[OASIS-AdES]	OASIS: Advanced Electronic Signature Profiles of the OASIS Digital Signature Service Version 1.0, OASIS Standard, http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-spec-v1.0-os.pdf
[OASIS-DSS]	OASIS: Digital Signature Service Core Protocols, Elements, and Bindings, Version 1.0, OASIS Standard, via http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf
[OASIS-SP]	OASIS: Signature Policy Profile of the OASIS Digital Signature Services Version 1.0, Committee Draft 01, 18 May 2009, http://docs.oasis-open.org/dss-x/profiles/sigpolicy/oasis-dssx-1.0-profiles-sigpolicy-cd01.pdf
[OASIS-VR]	OASIS: Profile for comprehensive multi-signature verification reports for OASIS Digital Signature Services Version 1.0, Committee Specification 01, 12 November 2010, http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cs01.pdf

[PAdES-1]	European Telecommunications Standards Institute (ETSI): Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 1: PAdES Overview – a framework document for PAdES, ETSI TS 102 778-1 V1.1.1, Technical Specification, 2009
[PAdES-3]	European Telecommunications Standards Institute (ETSI): Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 3: PAdES Enhanced – PAdES-BES and PAdES-EPES Profiles, ETSI TS 102 778-3 V1.1.2, Technical Specification, 2009
[PAdES-4]	European Telecommunications Standards Institute (ETSI): Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term – PAdES-LTV Profile, ETSI TS 102 778-4 V1.1.2, Technical Specification, 2009
[ISO 19005]	ISO 19005 – Document management – Electronic document file format for long-term preservation
[PDF/A-2]	ISO 19005-2:2011 – Document management – Electronic document file format for long-term preservation – Part 2: Use of ISO 32000-1 (PDF/A-2)
[PP_NK]	Common Criteria Schutzprofil (Protection Profile) Schutzprofil 1: Anforderungen an den Netzkonnektor BSI-CC-PP-0097
[PP_KON]	Common Criteria Schutzprofil (Protection Profile) Schutzprofil 2: Anforderungen an den Konnektor: BSI-CC-PP-0098
[RFC792]	IETF (September 1981) INTERNET CONTROL MESSAGE PROTOCOL http://tools.ietf.org/html/rfc792
[RFC1034]	RFC 1034 (November 1987): Domain Names – Concepts and Facilities http://tools.ietf.org/html/rfc1034
[RFC1122]	RFC 1122 (Oktober 1989): Requirements for Internet Hosts -- Communication Layers http://tools.ietf.org/html/rfc1122
[RFC1812]	F. Baker (ed.): Requirements for IP Version 4 Routers, IETF RFC 1812, http://www.ietf.org/rfc/rfc1812.txt
[RFC1918]	RFC1918 (Februar 1996): Address Allocation for Private Internets http://tools.ietf.org/html/rfc1918
[RFC2119]	RFC 2119 (März 1997): Key words for use in RFCs to Indicate Requirement Levels S. Bradner, http://tools.ietf.org/html/rfc2119

[RFC2131]	Network Working Group (03/1997): Dynamic Host Configuration Protocol http://www.ietf.org/rfc/rfc2131.txt
[RFC2132]	Network Working Group (03/1997): DHCP Options and BOOTP Vendor Extensions http://www.ietf.org/rfc/rfc2132.txt
[RFC2617]	Network Working Group (06/1999): HTTP Authentication: Basic and Digest Access Authentication http://www.ietf.org/rfc/rfc2617.txt
[RFC2818]	Network Working Group (05/2000): HTTP Over TLS http://www.ietf.org/rfc/rfc2818.txt
[RFC3447]	B. Kaliski:_PKCS #1: RSA Encryption, Version 2.1, RFC3447,
[RFC2616]	Network Working Group (06/1999): Hypertext Transfer Protocol -- HTTP/1.1 http://www.ietf.org/rfc/rfc2616.txt
[RFC2644]	D. Senie: *Changing the Default for Directed Broadcasts in Routers*, IETF RFC 2644, http://www.ietf.org/rfc/rfc2644.txt
[RFC2663]	P. Srisuresh, M. Holdrege: *IP Network Address Translator (NAT) Terminology and Considerations*, IETF RFC 2663, http://www.ietf.org/rfc/rfc2663.txt
[RFC3022]	RFC 3022 (Januar 2001): Traditional IP Network Address Translator (Traditional NAT) http://tools.ietf.org/html/rfc3022
[RFC3275]	D. Eastlage, J. Reagle, D. Solo: *(Extensible Markup Language) XMLSignature Syntax and Processing*, IETF RFC 3275, via http://www.ietf.org/rfc/rfc3275.txt
[RFC3279]	W, Polk, R. Hously, L. Bassham: *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IETF RFC 3279, http://www.ietf.org/rfc/rfc3279.txt
[RFC3629]	Network Working Group (11/2003): UTF-8, a transformation format of ISO 10646 http://www.ietf.org/rfc/rfc3629.txt
[RFC3927]	Network Working Group (05/2005): Dynamic Configuration of IPv4 Link-Local Addresses http://www.ietf.org/rfc/rfc3927.txt
[RFC3986]	Network Working Group (01/2005): Uniform Resource Identifier (URI): Generic Syntax

[RFC4122]	RFC 4122 (July 2005): A Universelly Unique Identifier UUID URN Namspace http://tools.ietf.org/html/rfc4122
[RFC4632]	Network Working Group (08/2006): Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan http://tools.ietf.org/html/rfc4632
[RFC5246]	RFC 5246 (August 2008): The Transport Layer Security (TLS) Protocol Version 1.2; http://tools.ietf.org/html/rfc5246
[RFC5652]	R. Housley: Cryptographic Message Syntax (CMS), RFC 5652 (September 2009) http://tools.ietf.org/html/rfc5652
[RFC 6598]	RFC 6598 (April 2012): IANA-Reserved IPv4 Prefix for Shared Address Space http://tools.ietf.org/html/rfc6598
[RFC6931]	RFC 6931 (April 2013): Additional XML Security Uniform Resource Identifiers (URIs) http://tools.ietf.org/html/rfc6931
[RFC7159]	RFC 7159 (March 2014): The JavaScript Object Notation (JSON) Data Interchange Format http://tools.ietf.org/html/rfc7159
[S/MIME]	RFC 5751 (Januar 2010): Secure/Multipurpose Internet Mail Extensions (S/MIME), Version 3.2, Message Specification http://www.ietf.org/rfc/rfc5751.txt
[SOAP1.1]	Simple Object Access Protocol (SOAP) 1.1 W3C Note (08 May 2000) https://www.w3.org/TR/2000/NOTE-SOAP-20000508/
[SOAP1.2]	SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) W3C Recommendation (27 April 2007) http://www.w3.org/TR/2007/REC-soap12-part1-20070427/
[SICCT]	TeleTrusT (17.12.2010): SICCT Secure Interoperable ChipCard Terminal, Version 1.21 https://www.teletrust.de/fileadmin/docs/projekte/sicct/SICCT-Spezifikation-1.21.pdf
[TIFF6]	TIFF Revision 6.0 (Final, June 3, 1992) https://www.adobe.io/open/standards/TIFF/_jcr_content/contentbody/download/file.res/TIFF6.pdf.html

[WSDL1.1]	W3C Note (15.03.2001): Web Services Description Language (WSDL) 1.1 http://www.w3.org/TR/wsdl
[XAdES]	European Telecommunications Standards Institute (ETSI): Technical Specication XML Advanced Electronic Signatures (XAdES). ETSI Technical Specication TS 101 903, Version 1.4.2, 2010
[XMLDSig]	W3C Recommendation (06.2008): XML-Signature Syntax and Processing http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/
[XMLEnc]	XML Encryption Syntax and Processing W3C Recommendation 11 April 2013 http://www.w3.org/TR/xmlenc-core1/
[XPATH]	W3C Recommendation (14 December 2010) XML Path Language (XPath) 2.0 (Second Edition) http://www.w3.org/TR/2010/REC-xpath20-20101214/
[XSLT]	W3C Recommendation (23 January 2007) XSL Transformations (XSLT) Version 2.0 http://www.w3.org/TR/2007/REC-xslt20-20070123/
[XAdES Baseline Profile]	European Telecommunications Standards Institute (ETSI): Electronic Signatures and Infrastructure (ESI); XAdES Baseline Profile; ETSI Technical Specification TS 103 171, Version 2.1.1, 2012-03
[CAdES Baseline Profile]	European Telecommunications Standards Institute (ETSI): Electronic Signatures and Infrastructure (ESI); CAdES Baseline Profile; ETSI Technical Specification TS 103 173, Version 2.1.1, 2013-04
[PAdES Baseline Profile]	European Telecommunications Standards Institute (ETSI): Electronic Signatures and Infrastructure (ESI); PAdES Baseline Profile; ETSI Technical Specification TS 103 172, Version 2.2.2, (2013-04)
[XSL]	W3C Recommendation (05.12.2006): Extensible Stylesheet language (XSL) Version1.1 http://www.w3.org/TR/2006/REC-xsl11-20061205/
[MTOM]	SOAP Message Transmission Optimization Mechanism W3C Recommendation 25 January 2005 http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/
[MTOM-SOAP1.1]	W3C Member Submission 05 April 2006 SOAP 1.1 Binding for MTOM 1.0 https://www.w3.org/Submission/soap11mtom10/
[WS-MTOM Policy]	W3C Member Submission 18 November 2007 MTOM Serialization Policy Assertion 1.1

[COS-G2]	Common Criteria Protection Profile, Card Operating System Generation 2, (PP COS G2), BSI-CC-PP-0082-V2

6 Anhang B – Profilierung der Signatur- und Verschlüsselungsformate (normativ)

6.1 Profilierung der Verschlüsselungsformate

6.2 Profilierung der Signaturformate

Tabelle 388: TAB_KON_779 „Profilierung der Signaturformate"

Aspekt (QES/nonQES)	Festlegung (XML-Signatur/CMS-Signatur/PDF-Signatur)
Zertifikatsreferenz (QES und nonQES)	XML-Signatur Bei der Signaturerstellung ist das XML-Element `SigningCertificate` gemäß den Vorgaben aus XAdES Kapitel 7.2.2 „The SigningCertificate element" anzulegen. Bei der Signaturprüfung ist es gemäß XAdES Kapitel G.2.2.5 „Verification technical rules" [XAdES] zu prüfen. Grundsätzlich sind auch Signaturen zu prüfen, die keine Zertifikatsreferenz enthalten. Das Prüfergebnis muss dann widerspiegeln, dass diese Sicherheitsfunktion nicht enthalten war. CMS-Signatur Bei der Signaturerstellung ist das Attribut `signing certificate reference` gemäß CAdES Kapitel 5.7.3 „Signing Certificate Reference Attributes" [CAdES] anzulegen. Bei der Signaturprüfung ist es gemäß CAdES Kapitel 5.6.3 „Message signature verification process" [CAdES] zu prüfen. Grundsätzlich sind auch Signaturen zu prüfen, die keine Zertifikatsreferenz enthalten. Das Prüfergebnis muss dann widerspiegeln, dass diese Sicherheitsfunktion nicht enthalten war. PDF-Signatur Bei der Signaturerstellung ist das Attribut `signing certificate reference` gemäß den Vorgaben aus [PAdES-3] Kapitel 4.4.3 „Signing Certificate Reference Attribute" anzulegen. Bei der Signaturprüfung ist es gemäß [PAdES-3]Kapitel 4.6.1 „Signing Certificate Reference Validation" zu prüfen. Grundsätzlich sind auch Signaturen zu prüfen, die keine Zertifikatsreferenz enthalten. Das Prüfergebnis muss dann widerspiegeln, dass diese Sicherheitsfunktion nicht enthalten war.

Signaturablage	PDF-Signatur Sie Signatur wird als Incremental Update gemäß [PDF/A-2] Kapitel 7.5.6 an das Dokument angefügt.
Parallelsignatur (QES und nonQES)	XML-Signatur Parallele Signaturen werden durch je ein `ds:signature`-Element pro Signatur abgebildet. Für die Signaturvariante „enveloping" werden parallele Signaturen nicht angeboten. CMS-Signatur: Parallele Signaturen werden durch je einen SignerInfo-Container pro Signatur realisiert. PDF-Signatur: Parallele Signaturen werden nicht angeboten.
Dokumentexkludierende Gegensignatur (QES und nonQES)	XML-Signatur Die Implementierung erfolgt mittels Countersignature gemäß [XAdES], Kapitel 7.2.4. Jede vorhandene Parallel-Signatur wird gegensigniert. CMS-Signatur: Die Implementierung erfolgt mittels der Countersignature gemäß CMS-Spezifikation [RFC5652]. Jede vorhandene Parallel-Signatur wird gegensigniert. PDF-Signatur: Dokumentexkludierende Gegensignaturen werden nicht angeboten.
Referenzierung (QES und nonQES)	XML-Signatur Bei der Signaturerzeugung verwendet der Konnektor in der Signatur nur ID-basierte Referenzen. Bei der Signaturprüfung reagiert der Konnektor bei Abweichungen hiervon mit Fehler 4208.
	XML-Signatur / CMS-Signatur Bei XML-Dokumenten und XML-Signaturen kann die Referenzierung von Objekten auf zwei Arten erfolgen: – Ist kein XML-Schema vorhanden, so werden die Werte des ID-Attributs des referenzierten Elements verwendet. – Wird ein XML-Schema übergeben, so muss dieses die ID-Attribute zur Referenzierung festlegen. Bei Abweichungen reagiert der Konnektor mit Fehlercode 4115.
Anzahl unterstützter Signaturen (QES und nonQES)	XML-Signatur / CMS-Signatur / PDF-Signatur Es müssen mindestens 20 Signaturen pro Dokument unterstützt werden. Sind mehr als die unterstützte Anzahl von Signaturen in einem Dokument enthalten, wird die Operation mit Fehler 4001 abgebrochen.

6.3 Profilierung VerificationReport

Anforderung eines ausführlichen Prüfberichts

Folgende Aufrufparameter müssen unterstützt werden:

```
<ReturnVerificationReport
xmlns="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#
                oasis-dssx-1.0-profiles-vr-cd1.xsd">
        <IncludeVerifier>false</IncludeVerifier>
        <IncludeCertificateValues>true</IncludeCertificateValues>
        <IncludeRevocationValues>true</IncludeRevocationValues>
        <ExpandBinaryValues>false</ExpandBinaryValues>
        <ReportDetailLevel>
                urn:oasis:names:tc:dss-
                x:1.0:profiles:verificationreport:reportdetail:allDetails
        </ReportDetailLevel>
</ReturnVerificationReport>
```

Verwendung des erzeugten VerificationReport

Für die folgenden Inhalte müssen die angegebenen Strukturen benutzt werden. Im Standard angegebene Pflichtfelder von erzeugten Strukturen müssen ggf. zusätzlich gefüllt werden:

1. Prüfzeitpunkt (Systemzeit des Konnektors zum Zeitpunkt der Prüfung)

   /VerificationReport/
   dss:VerificationTimeInfo/
   dss:VerificationTime

2. Signaturzeitpunkt(Ermittelter_Signaturzeitpunkt_Eingebettet)

   /VerificationReport/
   IndividualReport/
   SignedObjectIdentifier/
   SignedProperties/
   SignedSignatureProperties/
   XAdES:SigningTime

Die Signierzeit SigningTime ist nicht nur für XAdES-Signaturen, sondern allgemein für Signaturen gemäß AdES-Baseline-Profilierung, also auch für CAdES und PAdES zu füllen.

3. Angenommener Signaturzeitpunkt gemäß TIP1-A_5540 (QES) und TIP1-A_5545 (nonQES)

   /VerificationReport/
   IndividualReport/
   Details/
   dss:VerificationTimeInfo/
   dss:VerificationTime

4. der binäre Wert der Signatur

   /VerificationReport/

   IndividualReport/

   SignedObjectIdentifier/

   SignatureValue

5. Kurztext

   Der signierte Kurztext wird in folgendem XML-Element zurückgegeben:

   /VerificationReport/

   IndividualReport/

   SignedObjectIdentifier/

   SignedProperties/

   Other/
   SIG:ShortText

Die ungültigen Zeichen müssen aus dem ShortText entfern werden und der ShortText wird gekürzt, wenn er nicht den Längenvorgaben des Schemas entspricht.

6. Das folgende Element mit den Werten true/false gibt an, ob eine Zertifikatsreferenz gemäß Anhang B2 vorhanden ist (true) oder nicht (false):

/VerificationReport/

IndividualReport/

SignedObjectIdentifier/

SignedProperties/

Other/
SIG:ReferenceToSignerCertificate

7. Sämtliche signierte Attribute, deren Rückgabe nicht explizit über andere Elemente geregelt ist, werden als direkt anzeigbare Key/Value-Paare zurückgeben. Dabei sind sowohl Key und Value bereits für die Anzeige formatiert. Der Key wird in einer Zeile dargestellt. Der Value wird in mehreren Zeilen dargestellt, wobei ein Zeilenumbruch durch 'CARRIAGE RETURN (CR)' 'LINE FEED (LF)' erzeugt wird und keine weiteren Steuerzeichen erlaubt sind.

   /VerificationReport/

   IndividualReport/

   SignedObjectIdentifier/

   SignedProperties/

Other/
SIG:DisplayableAttributes

8. das Ergebnis der Signaturprüfung

   /VerificationReport/
   IndividualReport/
   Result

9. handelt es sich bei der Signatur um eine Gegensignatur wird diese als solche markiert

   /DetailedSignatureReport/
   Properties/
   UnsignedProperties/
   Other/
   SIG:CounterSignatureMarker

   und mit

   /DetailedSignatureReport/
   Properties/
   UnsignedProperties/
   Other/
   SIG:CounterSignatureMarker/
   SignatureValueReference/
   @IdRef

   auf jede (eine oder mehrere) gegensignierte Signaturen verwiesen. Dabei zeigt IdRef auf den jeweiligen gegensignierten Signaturwert

   /VerificationReport/
   IndividualReport/
   SignedObjectIdentifier/
   ds:SignatureValue/
   @Id

10. das Ergebnis der Zertifikatsprüfung,

    /VerificationReport/
    IndividualReport/
    Details/
    DetailedSignatureReport/
    CertificatePathValidity/
    PathValiditySummary/

ResultMajor

11. Inhalt des Zertifikates, auf dem beruhend signiert wurde

    /VerificationReport/

    IndividualReport/

    Details/

    DetailedSignatureReport/

    CertificatePathValidity/

    PathValidityDetail/

    CertificateValidity/

    CertificateValue

12. den Signaturalgorithmus der Dokumentensignatur (URI, angelehnt an den Wertebereich des Feldes ds:SignatureMethod),

    /VerificationReport/

    IndividualReport/

    Details/

    DetailedSignatureReport/

    SignatureOK/

    SignatureAlgorithm

13. aussagekräftiger Hinweis zum verminderten Beweiswert hinsichtlich Authentizität und Integrität der Signatur, wenn einer der bei der Signaturprüfung identifizierten und unterstützten Algorithmen zum Zeitpunkt der Signaturprüfung nicht mehr laut Algorithmenkatalog [ALGCAT] als geeignet eingestuft wird. Auszuwerten sind die Festlegungen des ALGCAT sowohl bezogen auf die Vergangenheit als auch auf die Zukunft.

    Für alle geprüften Zertifikate:

    ../

    vr:CertificateValidity/

    vr:SignatureOK/

    vr:SignatureAlgorithm/

    vr:Suitability/

    ./ResultMajor= urn:oasis:names:tc:dss:1.0:detail:invalid

    ./ResultMessage="Algorithmen seit <Jahr> als unsicher eingestuft"

14. PathValidity bis zur TrustAnchor-TSL

    //CertificateValidity/ChainingOK/ResultMajor (ab dem zweiten Zertifikat in der Kette)

    //CertificateValidity/CertificateStatus/CertStatusOK/ResultMajor

    //CertificateValidity/CertificateValue

    Für das Feld TrustAnchor ist

"urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:certDataBase"

zu verwenden.

15. Prüfergebnis des Gültigkeitszeitraums

    /VerificationReport/
    IndividualReport/
    Details/
    DetailedSignatureReport/
    CertificatePathValidity/
    PathValidityDetail/
    CertificateValidity/
    ValidityPeriodOK/
    ResultMajor

16. Prüfung der Extensions

    /VerificationReport/
    IndividualReport/
    Details/
    DetailedSignatureReport/
    CertificatePathValidity/
    PathValidityDetail/
    CertificateValidity/
    ExtensionsOK/
    ResultMajor

17. Zeitstempel und Herkunft der OCSP-Antwort für das Signaturzertifikat

    /VerificationReport/
    IndividualReport/
    Details/
    DetailedSignatureReport/
    CertificatePathValidity/
    PathValidityDetail/
    CertificateValidity/
    CertificateStatus/
    RevocationEvidence/
    OCSPValidity/
    OCSPIdentifier/
    ./XAdES:ResponderID/XAdES:ByName
    ./XAdES:ProducedAt

18. OCSP Antwort für das Signaturzertifikats
    /VerificationReport/
    IndividualReport/
    Details/
    /vr:DetailedSignatureReport/
    vr:CertificatePathValidity/
    vr:PathValidityDetail/
    vr:CertificateValidity/
    vr:CertificateStatus/
    vr:RevocationEvidence/
    vr:OCSPValidity/
    vr:OCSPValue

Sonderfälle:

Dokument mit parallelen Signaturen

Für jede Signatur wird ein IndividualReport erzeugt.

Dokument mit Signatur und Gegensignatur

Für jede Signatur wird ein IndividualReport erzeugt.

6.4 Profilierung der Dokumentenformate und Nachrichten

Tabelle 389: TAB_KON_775 „Profilierung der Dokumentformate und Nachrichten"

XML-Dokument	Es gelten folgende Mindestanforderungen, die der Konnektor bezüglich Dokumentenstruktur und Dokumenteninhalt unterstützen muss: - Hierarchietiefe des Dokumentenbaums: 30 Ebenen - Anzahl von XML-Elementen im Dokument: 30.000 - Anzahl von XML-Attributen je XML-Element: 20 - Anzahl von direkten Kindern eines XML-Elements: 50 - Länge von XML-Bezeichnern (z. B. Elementnamen, Attributnamen, Namespace-Prefixes, usw.): 200 - Anzahl von Transformationen: 64 - Element-Größe pro Einzelknoten im Base64-codierten Dokument: 30 MB
	Es dürfen keine ENTITY-Deklarationen im XML-Dokument vorkommen.
	Zu verifizierende XML-Dokumente dürfen im <Transforms>-Teil ihrer Referenzen weder XPath-Ausdrücke noch XSL-Transformationen enthalten.

	Bei Referenzen (ReferenceType) darf das Optionale URI-Attribut nicht vorhanden sein, oder es muss leer sein.
	XInclude darf nicht unterstützt werden.
	Die Attribute schemaLocation und noNamespaceSchemaLocation dürfen nicht unterstützt werden.

7 Anhang F – Übersicht Events

Tabelle 390: TAB_KON_777 Events Interne Mechanismen

Topic Ebene1 /Topic Ebene2 /Topic Ebene3	Typ	Schwere	Prot	AnClients	Parameter	Bedeutung	Auslöser (TUC/Op)
Interne Mechanismen							
BOOTUP /BOOTUP_COMPLETE	Op	Info	x	x		Änderung des Betriebszustandes	
OPERATIONAL_STATE /EC_CardTerminal_Software_Out_Of_Date ($ctId)	Op	Info	x	x	Value=true/false; CtID=$ctId; Bedeutung=$EC.description	Änderung des Betriebszustandes durch Änderung im Fehlerzustand (Änderung im Value).	
OPERATIONAL_STATE EC_CardTerminal_SMC-KT_Certificate_Expires_Soon ($ctId)	Op	Info	x	x	Value=true/false; CtID=$ctId; Bedeutung=$EC.description	Änderung des Betriebszustandes durch Änderung im Fehlerzustand (Änderung im Value).	TUC_KON_050
OPERATIONAL_STATE /EC_Connector_Software_Out_Of_Date	Op	Info	x	x	Value=true/false; Bedeutung=$EC.description	"	
OPERATIONAL_STATE /EC_FW_Not_Valid_Status_Blocked	Sec	Fatal	x	x	Value=true/false; Bedeutung=$EC.description	"	

OPERATIONAL_STATE /EC_Time_Sync_Not_Successful	Op	Info	x	x	Value=true/ false; LastSyncAttempt= $lastSyncAttempt Timestamp; LastSyncSuccess= $lastSyncSuccess Timestamp; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_TSL_Update_Not_Successful	Op	Info	x	x	Value=true/ false; Bedeutung= $EC.description; LastUpdateTSL= $lastUpdate TSLTimestamp	"	
OPERATIONAL_STATE /EC_TSL_Expiring	Sec	Info	x	x	Value=true/ false; NextUpdateTSL= $NextUpdate- Element der TSL; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_TSL_Trust_Anchor_Expiring	Sec	Info	x	x	Value=true/ false; ExpiringDate TrustAnchor= Ablaufdatum der Vertrauens ankergültigkeit; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_LOG_OVERFLOW	Op	Warning	x	x	Value=true/ false; Protokoll=$Protokoll; Bedeutung= $EC.description	"	TUC_KON_271

OPERATIONAL_STATE /EC_CRL_Expiring	Sec	Warning	x	x	Value=true/false; NextUpdateTSL=$NextUpdate-Element der TSL; Bedeutung=$EC.description	"	
OPERATIONAL_STATE /EC_Time_Sync_Pending_Warning	Sec	Warning	x	x	Value=true/false; LastSyncSuccess=$lastSyncSuccessTimestamp; Bedeutung=$EC.description	"	
OPERATIONAL_STATE /EC_TSL_Out_Of_Date_Within_Grace_Period	Sec	Warning	x	x	Value=true/false; NextUpdateTSL=$NextUpdate-Element der TSL; GracePeriodTSL=CERT_TSL_DEFAULT_GRACE_PERIOD_DAYS; Bedeutung=$EC.description	"	
OPERATIONAL_STATE /EC_CardTerminal_Not_Available($ctId)	Op	Error	x	x	Value=true/false; CtID=$ctId; Bedeutung=$EC.description	"	
OPERATIONAL_STATE /EC_No_VPN_TI_Connection	Op	Error	x	x	Value=true/false; Bedeutung=$EC.description	"	
OPERATIONAL_STATE /EC_No_VPN_SIS_Connection	Op	Error	x	x	Value=true/false; Bedeutung=$EC.description	"	

OPERATIONAL_STATE /EC_No_Online_Connection	Op	Error	x	x	Value=true/ false; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_IP_Adresses_Not_Available	Sec	Error	x	x	Value=true/ false; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_CRL_Out_Of_Date	Sec	Fatal	x	x	Value=true/ false; NextUpdateCRL= $NextUpdate der CRL; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_Firewall_Not_Reliable	Sec	Fatal	x	x	Value=true/ false; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_Random_Generator_Not_Reliable	Sec	Fatal	x	x	Value=true/ false; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_SecureKeyStore_Not_Available	Sec	Fatal	x	x	Value=true/ false; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_Security_Log_Not_Writable	Op	Fatal	x	x	Value=true/ false; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_Software_Integrity_Check_Failed	Sec	Fatal	x	x	Value=true/ false; Bedeutung= $EC.description	"	

OPERATIONAL_STATE /EC_Time_Difference_Intolerable	Sec	Fatal	x	x	Value=true/ false; Bedeutung= $EC.description; NtpTimedifference= Zeitabweichung; NtpMaxAllowedTime difference= NTP_MAX_TIMEDIFFERENCE; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_Time_Sync_Pending_Critical	Sec	Fatal	x	x	Value=true/ false; LastSyncSuccess= $lastSyncSuccess Timestamp; NtpGracePeriod= NTP_GRACE_PERIOD; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_TSL_Trust_Anchor_Out_Of_Date	Sec	Fatal	x	x	Value=true/ false; ExpiringDate TrustAnchor= Ablaufdatum der Vertrauensanker gültigkeit; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_TSL_Out_Of_Date_Beyond_Grace_Period	Sec	Fatal	x	x	Value=true/ false; Next UpdateTSL= $NextUpdate-Element der TSL; GracePeriodTSL= CERT_TSL_ DEFAULT_ GRACE_PERIOD_DAYS; Bedeutung= $EC.description	"	

OPERATIONAL_STATE /EC_CRYPTOPERATION_ALARM	Sec	Warning	x	x	Value=true/ false; Operation= $Operationsname; Count=$Summenwert; Arbeitsplatz =$<Liste operations-aufrufenden workplace IDs>; Meldung=' Auffällige Häufung von Operationsaufrufen in den letzten 10 Minuten'	"	
OPERATIONAL_STATE /EC_OTHER_ERROR_STATE($no)	$Type	$Severity	x	x	Value=true/ false; Bedeutung= $EC.description	"	
OPERATIONAL_STATE /EC_BNetzA_VL_Update_Not_Successful	Op	Info	x	x	Value=true/ false; LastUpdate BNetzAVL= $lastUpdateBNetzAVL Timestamp; Bedeutung= $EC.description;	"	
OPERATIONAL_STATE /EC_BNetzA_VL_not_valid	Sec	Warning	x	x	Value=true/ false; NextUpdate BNetzAVL= $NextUpdate-Element der BNetzA-VL; Bedeutung= $EC.description;	"	
Zugriffsberechtigungsdienst							
Dokumentvalidierungsdienst							
Dienstverzeichnisdienst							

Kartenterminaldienst							
CT /ERROR	$Error Type	$Severity	x	x	CtID=$CT.ID; Name=$CT.HOSTNAME; Error=$Fehlercode; Bedeutung= $Fehlertext	Bei der Kommunikation mitdem KT ist ein Fehler aufgetreten	TUC_KON_051 TUC_KON_053
CT /CONNECTED	Op	Info	x	x	CtID=$CT.CTID; Hostname= $CT.HOSTNAME	Die Verbindung zu einem Kartenterminal wurde hergestellt	
CT /DISCONNECTED	Op	Info	x	x	CtID=$CT.CTID; Hostname= $CT.HOSTNAME	Die Verbindung zu einem Kartenterminal wurde unterbrochen	
CT /TLS_ESTABLISHMENT_FAILURE	$Error Type	$Severity	x	x	CtID = $CT.ID; Name=$CT. HOSTNAME; Error=$Fehlercode; Bedeutung= $Fehlertext	Im Rahmen des Verbindungsaufbaus sind Fehler aufgetreten	TUC_KON_050
CT /CT_ADDING_ERROR	$Error Type	$Severity	x	x	IP=$IP-Adresse; Name=$Hostname; Error=$Fehlercode; Bedeutung= $Fehlertext	Bei dem Versuch ein KT der Verwaltung zuzufügen ist ein Fehler aufgetreten	TUC_KON_054
CT /SLOT_FREE	Op	Info	-	-	CtID=$CT.CTID; SlotNo= $CT.SLOTS_USED[X]	Internes Event von Kartenterminaldienst --> Kartendienst. Informiert, dass ein Slot frei wurde. Wird im Kartendienst ausgewertet und verursacht dort CARD/REMOVED	

CT /SLOT_IN_USE	Op	Info	-	-	CtID=$CT.CTID; SlotNo=<FU-Nummer aus Ereignisnachricht>	Internes Event von Kartenterminaldienst --> Kartendienst. Informiert, dass ein Slot belegt wurde. Wird im Kartendienst ausgewertet und verursacht dort CARD/INSERTED	
Kartendienst							
CARD /INSERTED	Op	Info	x	x	CardHandle= $CARD.CARDHANDLE; CardType=$CARD.TYP; CardVersion= $CARD.VER; ICCSN=$CARD.ICCSN; CtID=$CARD.CTID; SlotID= $CARD.SLOTID; InsertTime= $CARD.INSERTTIME; CardHolderName= $CARD.CARD HOLDERNAME; KVNR=$CARD.KVNR	Eine Karte wurde gesteckt	TUC_KON_001 (als Reaktion auf CTM /SLOT_IN_USE)
CARD /REMOVED	Op	Info	x	x	CardHandle= $CARD.CARDHANDLE; CardType=$CARD.TYP; CardVersion= $CARD.VER; ICCSN=$CARD.ICCSN; CtID=$CARD.CTID; SlotID= $CARD.SLOTID; InsertTime= $CARD.INSERTTIME; CardHolderName= $CARD.CARDHOLDER NAME; KVNR=$CARD.KVNR	Eine Karte wurde gezogen	Reaktion auf CTM/SLOT_FREE

CARD /PIN /VERIFY_STARTED	Op	Info	-	x			
CARD /PIN /VERIFY_FINISHED	Op	Info	-	x			
CARD /PIN /CHANGE_STARTED	Op	Info	-	x			
CARD /PIN /CHANGE_FINISHED	Op	Info	-	x			
CARD /PIN /ENABLE_STARTED	Op	Info	-	x	CardHandle=$; CardType=$; ICCSN=$; CtID=$; SlotID=$; PinRef=$PinRef; PinInputCtID =$PinInputKT	PIN-Schutz anschalten beginnt	TUC_KON_027
CARD /PIN /ENABLE_FINISHED	Op	Info	-	x	CardHandle=$; CardType=$; ICCSN=$; CtID=$; SlotID=$; PinRef=$PinRef; PinInputCtID =$PinInputKT	PIN-Schutz anschalten wurde beendet	TUC_KON_027
CARD /PIN /DISABLE_STARTED	Op	Info	-	x	CardHandle=$; CardType=$; ICCSN=$; CtID=$; SlotID=$; PinRef=$PinRef; PinInputCtID =$PinInputKT	PIN-Schutz ausschalten beginnt	TUC_KON_027

CARD /PIN /DISABLE_FINISHED	Op	Info	-	x	CardHandle=$; CardType=$; ICCSN=$; CtID=$; SlotID=$; PinRef=$PinRef; PinInputCtID =$PinInputKT	PIN-Schutz ausschalten wurde beendet	TUC_KON_027
Systeminformationsdienst							
Verschlüsselungsdienst							
Signaturdienst							
SIG /SIGNDOC /NEXT_SUCCESSFUL	Op	Info	-	X	$Jobnummer	Die nächste Signatur aus einem Signaturstapel wurde erfolgreich erstellt.	TUC_KON_166 „nonQES Signaturen erstellen“ TUC_KON_154 „QES Signaturen erstellen“
Zertifikatsdienst							
CERT /TSL /IMPORT	Op	Error	x	-	$Fehlerbeschreibung	Manueller Import der TSL fehlgeschlagen	TUC_KON_032 "TSL aktualisieren"
CERT /TSL /UPDATED	Op	Info	x	-		Eine neue TSL wurde erfolgreich in den TrustStore eingespielt	TUC_KON_032 "TSL aktualisieren"
CERT /CRL /INVALID	Op	Error	x	-		Prüfung der Signatur der CRL fehlgeschlagen	TUC_KON_040 "CRL aktualisieren"

CERT /CRL /IMPORT	Op	Error	x	-	$Fehlerbeschreibung	Manueller Import der CRL fehlgeschlagen	TUC_KON_040 "CRL aktualisieren"
CERT /CRL /UPDATED	Op	Info	x	-		Die CRL wurde erfolgreich aktualisiert	TUC_KON_040 "CRL aktualisieren"
CERT /CARD /EXPIRATION	Op	Warning	x	x	CARD_TYPE=gSMC-K; ICCSN=$ICCSN; Konnektor= $MGM_KONN_HOSTNAME; ZertName=<Name des Zertifikatsobjekts> ; Expiration Date=$validity	gSMC-K abgelaufen	TUC_KON_033 "Zertifikatsablauf prüfen"
CERT /CARD /EXPIRATION	Op	Warning	-	x	CARD_TYPE=$Type; ICCSN=$ICCSN; CARD_HANDLE= $CardHandle; CardHolderName= $CardHolderName; ZertName=<Name des Zertifikatsobjekts> ; ExpirationDate= $validity	Sonstige Karte abgelaufen	TUC_KON_033 "Zertifikatsablauf prüfen"
CERT /CARD /EXPIRATION	Op	Info	-	x	CARD_TYPE=gSMC-K; ICCSN=$ICCSN; Konnektor= $MGM_KONN_HOSTNAME; ZertName=<Name des Zertifikatsobjekts> ; ExpirationDate= $validity; DAYS_LEFT= $validity-$Today	gSMC-K läuft innerhalb von DAYS_LEFT Tagen ab	TUC_KON_033 "Zertifikatsablauf prüfen"
CERT /CARD /EXPIRATION	Op	Info	-	x	CARD_TYPE=$Type; ICCSN=$ICCSN; CARD_HANDLE= $CardHandle; CardHolderName= $CardHolderName; ZertName=<Name des Zertifikatsobjekts> ; ExpirationDate= $validity; DAYS_LEFT= $validity-$Today	Sonstige Karte läuft innerhalb von DAYS_LEFT Tagen ab	TUC_KON_033 "Zertifikatsablauf prüfen"

CERT /BNETZA_VL /UPDATED	Op	Info	x	-		Eine neue BNetzA-VL wurde erfolgreich in den TrustStore eingespielt	TUC_KON_031 " BNetzA-VL aktualisieren"
CERT /BNETZA_VL /IMPORT	Op	Error	x	-	$Fehlerbeschreibung	Manueller Import der BNetzA-VL fehlgeschlagen	TUC_KON_031 " BNetzA-VL aktualisieren"
Protokollierungsdienst							
LOG /ERROR	$ErrorType	$Severity	-	-	Error=$Fehlercode	Im Protokollierungsdienst auftretende Fehler werden verteilt	TUC_KON_271
LOG /CRYPTO_OP	Sec	Info	x	-	Operation= $Operationsname; <für alle betroffenen Schlüssel:> Karte=$ICCSN; Keyref=<Referenz auf den Schlüssel>; CARD_HANDLE= $CardHandle; CardHolderName= $CardHolderName		
TLS-Dienst							
Anbindung LAN/WAN							
ANLW /LAN /IP_CHANGED	Op	Warning	x	-	IP=$dieNeueIP	Wenn der LAN-Adapter eine neue IP oder Netzwerk bekommen hat	DHCP, Management schnittstelle

ANLW /WAN /IP_CHANGED	Op	Info	x	-	IP=$dieNeueIP	Wenn der WAN-Adapter eine neue IP oder Netzwerk bekommen hat	DHCP, Management schnittstelle
DHCP-Server							
DHCP /SERVER /STATECHANGED	Op	Info	x	x	STATE= $DHCP_SERVER_STATE		Administrator
DHCP Client							
DHCP /LAN_CLIENT /RENEW	Op	Info	x	x	IP_ADDRESS= <Belegung>		TUC_KON_341
DHCP /WAN_CLIENT /RENEW	Op	Info	x	x	IP_ADDRESS= <Belegung>		TUC_KON_341
DHCP /LAN_CLIENT /STATECHANGED	Op	Info	x	x	STATE= $DHCP_CLIENT_ LAN_STATE		
DHCP /WAN_CLIENT /STATECHANGED	Op	Info	x	x	STATE= $DHCP_CLIENT_ WAN_STATE		
VPN-Client							
NETWORK /VPN_TI /UP	Op	Info	x	x		Wenn der VPN-Tunnel zur TI erfolgreich aufgebaut worden ist.	
NETWORK /VPN_TI /DOWN	Op	Info	x	x		Wenn der VPN-Tunnel zur TI nicht mehr zur Verfügung steht.	AFO
NETWORK /VPN /CONFIG_CHANGED	Op	Info	x	-		Wenn die Konfiguration des VPN-Clients angepasst wurde.	Management schnittstelle

NETWORK /VPN_SIS /UP	Op	Info	x	x		Wenn der VPN-Tunnel zum SIS erfolgreich aufgebaut worden ist.	
NETWORK /VPN_SIS /DOWN	Op	Info	x	x		Wenn der VPN-Tunnel zum SIS nicht mehr zur Verfügung steht.	AFO
Zeitdienst							
NTP /ENTERCRITICALSTATE	Op	FATAL	x	-	MESSAGE= „CRITICALTIME DEVIATION”	Zeitabweichung von mehr als einer Stunde entdeckt	
Namensdienst und Dienstlokalisierung							
Leistungsumfänge und Standalone-Szenarios							
MGM /ADMINCHANGES	Op	Info	x	-	User= $AdminUsername; RefID=$ReferenzID; NewVal= $NeuEingestellter Wert“	Änderungen die der Admin vornimmt werden protokolliert	
MGM /CONFIG_EXIMPORT	Op	Info	x	-	User= $AdminUsername; Mode= [Export/Import]	Dokumentiert (via Mode), dass die Konnektor konfiguration exportiert oder importiert wurde.	
MGM /FACTORYSETTINGS	Op	Info	x	-	User= $AdminUsername	Ein ausgelöster Werksreset wird protokolliert	

MGM /REMOTE_SESSION	Op	Info	x	-	InitUser= $AdminUsername; RemoteID=<Kennung der Gegenstelle>; Mode= [InitSuccess/ InitFail/Exit]	Protokollierung des Versuchs, des Beginns und des Endes einer Remote-Management Session	
MGM /LU_CHANGED /LU_ONLINE	Op	Info	x	x	Active= $MGM_LU_ONLINE	Leistungsumfang Online wurde aktiviert/ deaktiviert	Administrator
MGM /LU_CHANGED /LU_SAK	Op	Info	x	x	Active= $MGM_LU_SAK	Leistungsumfang Signatur anwendungskomponente wurde aktiviert/deaktiviert	Administrator
MGM /STANDALONE_CHANGED	Op	Info	x	x	Active= $MGM_STANDALONE_KON	Festlegung des Konnektors als "Alleinstehend" wurde geändert	Administrator
In- und Außerbetriebnahme							
MGM /TI_ACCESS_GRANTED	Op	Info	x	-	Active= $MGM_TI_ACCESS_ GRANTED	Der Konnektor wurde erfolgreich freigeschaltet	Administrator
Software- Aktualisierungsdienst (KSR-Client)							
KSR /ERROR	$Error Type	$Severity	x	x	Target=Konnektor; Name= <MGM_KONN_HOSTNAME>; Error=$Fehlercode; Bedeutung= $Fehlertext	Während der Konnektor aktualisierung ist ein Fehler aufgetreten	TUC_KON_280
KSR /ERROR	$Error Type	$Severity	x	x	Target=KT; Name= <KT-Friendly Name>; CtID=$ctID; Error=$Fehlercode; Bedeutung= $Fehlertext	Während einer Kartenterminal aktualisierung ist ein Fehler aufgetreten	TUC_KON_281

KSR /ERROR	$Error Type	$Severity	x	x	Error=$Fehlercode; Bedeutung= $Fehlertext	Im KSR-Client ist ein Fehler aufgetreten	TUC_KON_282
KSR /UPDATE /START	Sec	Info	x	x	für TUC_KON_280 Target=Konnektor; Name= <MGM_KONN_HOSTNAME> für TUC_KON_281 Target=KT; CtID=$CtID	Ein Updateprozess im Konnektor wird gestartet, Ziel Konnektor oder Kartenterminal	TUC_KON_280 TUC_KON_281
KSR /UPDATE /SUCCESS	Sec	Info	x	x	für TUC_KON_280 Target=Konnektor; Name= <MGM_KONN_HOSTNAME>; NewFirmwareversion= <UpdateInformation.FirmwareVersion>; ConfigurationChanged =<Ja/Nein>; ManualInputNeeded= <Ja/Nein> für TUC_KON_281 Target=KT; Name= <KT-FriendlyName>; CtID=$ctID; NewFirmwareversion= <UpdateInformation.FirmwareVersion>	Die Firmware des Konnektors/ eines Kartenterminals wurde erfolgreich aktualisiert	TUC_KON_280 TUC_KON_281
KSR /UPDATE /END	Sec	Info	x	x	für TUC_KON_280 Target=Konnektor; Name= <MGM_KONN_HOSTNAME> für TUC_KON_281 Target=KT; CtID=$CtID	Ein Updateprozess im Konnektor wurde beendet	TUC_KON_280 TUC_KON_281

KSR /UPDATE /KONNEKTOR_DOWNLOAD_END	Op	Info	x	x	Je heruntergeladenem FW-Paket: ProductVendorID= $UpdateInformation/ ProductVendorID; ProductCode= $UpdateInformation/ ProductCode; ProductName= $UpdateInformation/ ProductName; FirmwareVersion= $UpdateInformation/ Firmware/FWVersion; Deadline= $UpdateInformation/ DeploymentInformation/ Deadline; FWPriority= $UpdateInformation/ Firmware/FWPriority; FirmwareReleaseNotes= $UpdateInformation/ Firmware/ FirmwareReleaseNotes	Download der Konnektor Firmware abgeschlossen	TIP1-A_6025

KSR /UPDATES_AVAILABLE	Op	Info	-	x	Je gefundenem FW-Paket: ProductVendorID= $UpdateInformation/ ProductVendorID; ProductCode= $UpdateInformation/ ProductCode; ProductName= $UpdateInformation/ ProductName; FirmwareVersion= $UpdateInformation/ FirmwareVersion; Deadline= $UpdateInformation/ DeploymentInformation/ Deadline; FWPriority= $UpdateInformation/ Firmware/FWPriority; FirmwareReleaseNotes= $UpdateInformation/ Firmware/ FirmwareReleaseNotes	Ein oder mehrere Updates auf neuere Versionen sind verfügbar	TIP1-A_4836
KSR /UPDATE_KONFIG	Op	Info	x	-	AlteVersion, NeueVersion	Aktualisierung Bestandsnetze	TUC_KON_283
Zertifikatserneuerung							
SMC_K/UPDATE/SUCCESS	Op	Info	x	x		Zertifikate erfolgreich erneuert	TUC_KON_410
SMC_K/DOWNLOAD/ERROR	Op	Error	x	x		Fehler beim Zertifikatsdownload	TUC_KON_410
SMC_K/UPDATE/ERROR	Op	Error	x	x	Iccsn=$Iccsn; Profile=$CertProfile; Serial=$Serialnumber;	Prüffehler bei Zertifikatsupdate	TUC_KON_410

					Fail=Iccsn | Date | Crypt | Serial | Ocsp		
SMC_K/REGISTER/ERROR	Op	Error	x	x	Iccsn=$Iccsn; SerialOld=$SerialnumberNKOld; SerialNew=$SerialnumberNKNew; Operation=Register | Deregister; Fail=No_Smcb | Other	Fehler bei Reregistrierung	TUC_KON_410

Die Abbildungsvorschrift von Fehler- auf Event-Type lautet:

- Security → Security,
- Technical → Operation,
- Infrastructure → Infrastructure,
- Business → Business,
- Other → Other

8 Anhang H – Mapping von „Architektur der TI-Plattform“ auf Konnektorspezifikation

Tabelle 391: TAB_KON_711 Architektur der TI-Plattform, Berechtigt Fachmodule

Interface	Operation	→	Funktionsmerkmal	Interface
I_Cert_Verification	verify_Certificate	→	Zertifikatsdienst	TUC_KON_037 "Zertifikat prüfen"
I_Crypt_Operations	decrypt_Document	→	Verschlüsselungsdienst	TUC_KON_071 "Daten hybrid entschlüsseln"
	encrypt_Document	→		TUC_KON_070 "Daten hybrid verschlüsseln"
I_DNS_Name_Information	get_FQDN	→	Namensdienst und Dienstlokalisierung	TUC_KON_364 „DNS Reverse Lookup durchführen“
	get_IP_Address	→		TUC_KON_361 „DNS Namen auflösen“
	get_Service_Information	→	Namensdienst und Dienstlokalisierung	TUC_KON_362 „Liste der Dienste abrufen“ TUC_KON_363 „Dienstdetails abrufen“
I_IP_Transport	send_Data_TI	→		
I_KT_Operations	interact_with_User	→	Kartenterminaldienst	TUC_KON_051 "Mit Anwender über Kartenterminal interagieren"
I_KV_Card_Handling	discard_Card_Usage_Reference	→	---	*--- keine Umsetzung notwendig. Erfolgt implizit*

	get_Card_Usage_Reference	→	---	*--- keine Umsetzung notwendig. Erfolgt implizit*
I_KV_Card_Operations	decrypt_Data	→	Kartendienst	TUC_KON_219 "Entschlüssele"
	do_Reset	→		TUC_KON_024 "Karte zurücksetzen"
	erase_Card_Data	→		TUC_KON_211 „LöscheRecordInhalt“ TUC_KON_204 „LöscheDateiInhalt“
	extract_card_data	→	Zertifikatsdienst	TUC_KON_034 "Zertifikatsinformationen extrahieren"
	read_Card_Data	→	Kartendienst	TUC_KON_202 "LeseDatei"
		→		TUC_KON_209 "LeseRecord"
		→		TUC_KON_215 "SucheRecord"
	read_KVK	→		TUC_KON_202 "Lese Datei"
	send_APDU	→		TUC_KON_200 "SendeAPDU"
	sign_Data	→		TUC_KON_218 "Signiere"
	verify_eGK	→		TUC_KON_018 "eGK-Sperrung prüfen"
	write_Card_Data	→		TUC_KON_203 "SchreibeDatei"

		→		TUC_KON_210 "SchreibeRecord"
		→		TUC_KON_214 "FügeHinzuRecord"
	write_eGK_Protocol	→		TUC_KON_006 "Datenzugriffsaudit eGK schreiben"
I_KV_Card_Reservation	handle_Session	→	Kartendienst	TUC_KON_023 "Karte reservieren"
I_KV_Card_Unlocking	authorize_Card	→	Kartendienst	TUC_KON_005 "Card-to-Card authentisieren"
	change_PIN	→		TUC_KON_019 "PIN ändern"
	enable_PIN disable_PIN	→		TUC_KON_027 „PIN-Schutz ein-/ausschalten”
	do_C2C	→		TUC_KON_005 "Card-to-Card authentisieren"
	get_PIN_Status	→		TUC_KON_022 "Liefere PIN-Status"
	initialize_PIN	→		TUC_KON_019 "PIN ändern"
	unblock_PIN	→		TUC_KON_021 "PIN entsperren"
	verify_PIN	→		TUC_KON_012 "PIN verifizieren"
I_Notification_From_FM	notify	→	Systeminformationsdienst	TUC_KON_256 "Systemereignis absetzen"

I_Local_Storage	write_Data read_Data erase_Data	→	Konnektormanagement	TIP1-A_5484
I_Poll_System_Information	get_Ressource_Information	→	Systeminformationsdienst	TUC_KON_254 "Liefere Ressourcendetails"
	get_Ressource_List	→		TUC_KON_252 "Liefere KT_Liste"
	get_Ressource_List	→		TUC_KON_253 "Liefere Karten_Liste"
I_Reg_Notification	register_for_Notifications	→	---	*--- keine Umsetzung notwendig. Erfolgt implizit*
I_Role_Information	get_Role	→	Kartendienst	TUC_KON_036 „LiefereFachliche Rolle“
I_SAK_Operations	sign_Document_QES	→	Signaturdienst	TUC_KON_150 „Dokumente QES signieren“
	verify_Document_QES	→		TUC_KON_151 "QES Dokumentensignatur prüfen"
I_Sign_Operations	sign_Document	→	Signaturdienst	TUC_KON_160 „Dokumente nonQES signieren“
	external_Authenticate	→		TUC_KON_160 „Dokumente nonQES signieren“
	verify_Document	→		TUC_KON_161 „nonQES Dokumentsignatur prüfen“

	get_Certificate	→	Kartendienst	TUC_KON_216 „LeseZertifikat“
I_Symm_Crypt_Operations	decrypt_Document_Symmetric	→	Verschlüsselungsdienst	TUC_KON_073 "Daten symmetrisch entschlüsseln"
	encrypt_Document_Symmetric	→		TUC_KON_072 "Daten symmetrisch verschlüsseln"
I_Synchronised_System_Time	get_Time	→	Zeitdienst	TUC_KON_351 "Liefere Systemzeit"
I_TLS_Client	send_Secure	→	TLS-Dienst	TUC_KON_110 "Kartenbasierte TLS-Verbindung aufbauen"
		→		TUC_KON_111 "Kartenbasierte TLS-Verbindung abbauen"
		→	Anbindung LAN/WAN	AFOs: Routing der IP-Pakete von Fachmodul (=Konnektor intern) --> VPN_TI
I_Directory_Query	search_Directory	→	LDAP-Proxy	TUC_KON_290 „LDAP-Verbindung aufbauen“
		→		TUC_KON_291 „Verzeichnis abfragen“
		→		TUC_KON_292 „LDAP-Verbindung trennen"

		→		TUC_KON_293 „Verzeichnisabfrage abbrechen"
I_KSRC_FM_Support	list_available_Packages	→	Software-Aktualisierung (KSR-Client)	TUC_KON_285 „UpdateInformationen für Fachmodul beziehen"
	load_Package	→		TUC_KON_286 „Paket für Fachmodul laden"

Tabelle 392: TAB_KON_712 Architektur der TI-Plattform, Berechtigt Clientsysteme

Interface	Operation	→	Funktionsmerkmal	Interface:Operation
I_Crypt_Operations	decrypt_Document	→	Verschlüsselungsdienst	EncryptionService:DecryptDocument
	encrypt_Document	→		EncryptionService:EncryptDocument
I_DNS_Name_Resolution	get_FQDN	→	Namensdienst und Dienstlokalisierung	GetFQDN
	get_IP_Address	→		GetIPAddress
I_IP_Transport	send_Data_External	→	Anbindung LAN/WAN	AFOs: Routing der IP-Pakete von Client --> VPN_SIS
I_KV_Card_Handling	discard_Card_Usage_Reference	→	---	*--- keine Umsetzung notwendig. Erfolgt implizit*
	get_Card_Usage_Reference	→	---	*--- keine Umsetzung notwendig. Erfolgt implizit*

I_KV_Card_Unlocking	change_PIN	→	Kartendienst	CardService :ChangePin
	disable_PIN	→		CardService :EnablePin
	enable_PIN	→		CardService :DisablePin
	get_PIN_Status	→		CardService :GetPinStatus
	initialize_PIN	→		CardService :ChangePin
	unblock_PIN	→		CardService :UnblockPin
	verify_PIN	→		CardService :VerifyPin
I_Poll_System_Information	get_Ressource_Information	→	Systeminformationsdienst	EventService :GetResourceInformation
	get_Ressource_List	→		EventService :GetCardTerminals
	get_Ressource_List	→		EventService :GetCards
I_Reg_Notification	register_for_Notifications	→	Systeminformationsdienst	EventService :Subscribe
		→		EventService :Unsubscribe
		→		EventService :GetSubscription
I_SAK_Operations	sign_Document_QES	→	Signaturdienst	SignatureService :SignDocument
	verify_Document_QES	→		SignatureService :VerifyDocument
I_Sign_Operations	sign_Document	→	Signaturdienst	SignatureService :SignDocument

	verify_Document	→		SignatureService :VerifyDocument
	external_Authenticate	→	Authentifizierungsdienst	AuthSignatureService :ExternalAuthenticate
	get_Certificate	→	Zertifikatsdienst	CertificateService :ReadCardCertificate
I_NTP_Time_Information	sync_Time	→	Zeitdienst	I_NTP_Time_Information :sync_Time
I_Directory_Query	search_Directory	→	LDAP-Proxy	LDAP-Operation (TIP1-A_5521)

Tabelle 393: TAB_KON_713 Architektur der TI-Plattform, Berechtigt eHealth-KT

Interface	Operation	→	Funktionsmerkmal	Interface:Operation
I_Notification	notify	→	SICCT	Ereignisdienst :SICCT-Ereignisnachrichten
		→	SICCT	Ereignisdienst :ServiceAnnouncement

Tabelle 394: TAB_KON_714 Architektur der TI-Plattform, Berechtigt Administrator

Interface	Operation	->	Funktionsmerkmal	Interface:Operation
I_Change_System_Time	set_System_Time	->	Zeitdienst	TIP1-A_4793 Konfigurierbarkeit des Konnektor NTP-Servers
I_Facade_Access_Configuration	add_Clientsystem	->	Fachliche Anbindung der Clientsysteme	TIP1-A_4518 Konfiguration der Anbindung Clientsysteme
	remove_Clientsystem			

	set_CS_Access_Mode			
I_KSRC_Local_Management	do_local_Update	->	Software-Aktualisierung (KSR-Client)	TUC_KON_280 "Konnektoraktualisierung durchführen"
I_KSRC_Management	do_Update	->	Software-Aktualisierung (KSR-Client)	TUC_KON_280 "Konnektoraktualisierung durchführen"
				TUC_KON_281 "Kartenterminalaktualisierung anstoßen"
	list_available_Updates			TUC_KON_282 "Update Informationen beziehen"
I_KTV_Management	configure_KTs	->	Kartenterminalverwaltung	Managementschnittstelle :TIP1-A_4555 Manuelles Hinzufügen eines Kartenterminals
				Managementschnittstelle :TIP1-A_4540 Reaktion auf KT Service Announcement
				Managementschnittstelle :TIP1-A_4556 Pairing mit Kartenterminal durchführen
				Managementschnittstelle :TIP1-A_4557 Ändern der Korrelationswerte eines Kartenterminals

9 Anhang I – Umsetzungshinweise (informativ)

In diesem Anhang finden sich Darstellungen und Informationen, die ein Konnektorhersteller zur Umsetzung der normativen Anforderungen in ein konkretes Produkt berücksichtigen kann. Sie wurden im Rahmen der Erhebung der normativen Anforderungen erarbeitet, um die Umsetzbarkeit der Anforderungen zu bestätigen.

Dieser Anhang soll als Unterstützung für eine Umsetzung verstanden werden und erhebt keinen Anspruch auf Korrektheit und Vollständigkeit.

9.1 Systemüberblick

9.1.1 – Hinweise zur Sicherheitsevaluierung nach Common Criteria

Gemäß dem Sicherheitskonzept des Konnektors [gemKPT_Sich_Kon] muss die Software des Konnektors nach Common Criteria (CC) evaluiert und geprüft werden.

Diese Software erbringt Sicherheitsleistungen in zwei wesentlichen Funktionsblöcken. Durch diese Aufteilung ist es möglich, dass die einzelnen Funktionsblöcke zeitlich voneinander unabhängig bzw. sogar von unterschiedlichen Herstellern implementiert, evaluiert und geprüft werden können. Es werden zwei Schutzprofile (Protection Profile) für die Funktionsblöcke des Konnektors erstellt. Es handelt sich dabei um die Schutzprofile des Netzkonnektors (KONN.NK) sowie des Anwendungskonnektors (KONN.AK) inklusive der Signaturanwendungskomponente. Das Schutzprofil des Sicherheitsmoduls für den Konnektor (SM-K) wird in diesem Kapitel nicht betrachtet.

Diese Schutzprofile definieren eine implementierungsunabhängige Menge von Sicherheitsanforderungen für die einzelnen Konnektorfunktionsblöcke bzw. Konnektorbestandteile. Anhand dieser Schutzprofile werden von den Herstellern der Konnektoren die Sicherheitsvorgaben (Security Targets) für die konkreten Umgebungen erstellt, welche als Eingangsdokumente für den Zertifizierungsprozess der jeweiligen konkreten Komponenten eingesetzt werden. Diese zu evaluierenden Komponenten werden als Evaluierungsgegenstand (EVG) bezeichnet.

9.1.1.1 Separationsmechanismen des Konnektors

Damit es nach einer erfolgreichen Evaluierung eines Konnektors auch weiterhin möglich bleibt, Software oder Daten, die keinen direkten Einfluss auf Sicherheitsfunktionen der EVGs aufweisen, ohne eine Re-Evaluierung definiert auszutauschen, hinzuzufügen oder zu erweitern, ist eine Separation der Komponenten des EVG dringend anzuraten.

Implementiert der Hersteller keine bzw. nicht ausreichende Separationsmechanismen, so ist bei bestimmten Update-Arten von einer aufwändigen Re-Evaluierung des entsprechenden EVGs auszugehen. Die Separation dient also der Trennung zwischen ausführbarem Code des EVG, welcher Sicherheitsfunktionen umsetzt, und zusätzlichem ausführbarem Code auf dem Konnektor, welcher keine Sicherheitsfunktionen umsetzt.

Die Wahl der Separationsmechanismen steht dem Hersteller frei und muss in den Sicherheitsvorgaben für den EVG beschrieben und als solcher evaluiert werden. Aus diesen Sicherheitsvorgaben ergibt sich auch, welche Update-Arten bei welchen Separationsmechanismen eine Re-Evaluierung des EVG erfordern und wie aufwendig diese Re-Evaluierung ausfällt.

Unter diese Update-Arten können beispielsweise – je nach Konnektorarchitektur, CC-Dokumentation oder Konnektorimplementierung – Bestandteile des unter dem Konnektor arbeitenden Betriebsystems, die Installation dezentraler Komponenten von Fachlogik oder Konfigurationsdaten des Konnektors fallen.

Als Beispiel für Separationsmechanismen sei auf die folgende informative Aufzählung verwiesen, welche jedoch keinen Anspruch auf Vollständigkeit besitzen kann und nur mögliche Alternativen aufzeigt:

- Java-Sandbox-Konzept,
- Interpreter mit restriktiver Laufzeitprüfung,
- vom Betriebssystem bereitgestellte Prozess- und Speichertrennung,
- virtuelle Maschinen,
- physische Trennung durch separierte Hardware.

Je nach gewähltem Architekturansatz des Herstellers sind nicht alle hier genannten Alternativen für die Separation des EVG auf dem Konnektor anwendbar.

Insbesondere sollte der Hersteller den eigentlichen Update-Prozess und die dafür verantwortliche Komponente mit besonderer Sorgfalt beschreiben, spezifizieren und implementieren. Bei einer fehlerhaften Implementierung dieser Komponente besteht die Gefahr einer Schwächung oder des Ausschaltens von Sicherheitsfunktionen des EVG. Die Update-Komponente muss eine sichere Zuweisung der Updates zu den separierten Bestandteilen des EVGs gewährleisten. Auch ist zu betonen, dass der EVG immer die Integrität der Daten des Updates und die Authentizität des Absenders prüfen muss, bevor ein Update akzeptiert wird. Der Update-Komponente muss somit besondere Beachtung geschenkt werden.

9.1.1.2 Granularität der TSF

Die TSF (TOE Security Functionality) eines EVG besteht aus Subsystemen und Modulen, wobei ein Modul die genaueste Beschreibung einer Funktionalität darstellt und unterhalb der Subsysteme angesiedelt ist. Subsysteme beschreiben das Design des EVG und können wiederum – je nach Komplexität eines EVGs – aus weiteren Subsystemen bestehen. Ein Entwickler sollte außer der Modulbeschreibung keine weiteren Informationen zur Implementierung der dort beschriebenen Funktionalität benötigen.

Die Subsysteme und Module der TSF gliedern sich in drei Klassen:

(a) SFR-Enforcing Subsysteme und Module. Hierunter fallen die Subsysteme und Module, welche eine funktionale Sicherheitsanforderung direkt durchsetzen.

(b) SFR-Supporting Subsysteme und Module. Hierunter fallen die Subsysteme und Module, welche bei der Durchsetzung einer funktionalen Sicherheitsanforderung unterstützend wirken.

(c) SFR-Non-Interfering Subsysteme und Module. Hierunter fallen die Subsysteme und Module, welche keine Leistung bei der Erfüllung einer funktionalen Sicherheitsanforderung erbringen.

Sollte nach einer erfolgreichen CC-Evaluierung eines Konnektors die Notwendigkeit zur Änderung der Software des Konnektors gegeben sein, so ist unter Umständen eine Re-Evaluierung des EVG erforderlich. Diese Notwendigkeit kann sich aus der Behebung von nachträglich erkannten Fehlern, aufgetretenen Sicherheitslücken, Schwächen eines Standardverfahrens oder einer erforderlichen Erweiterung der Funktionalität ergeben.

Im Rahmen der Aufzählung der Anforderungen an die Beschreibung des EVG-Design (ADV_TDS) wird bereits die Aufteilung der TSF auf Subsysteme und Module beschrieben. Trotzdem soll hiermit ausdrücklich geraten werden, die Aufteilung der TSF auf die Subsysteme und Module selbst und die Aufteilung der Subsysteme und Module auf die drei o. g. Klassen möglichst feingranular durchzuführen.

Denn so

1. können einfacher umfassende Tests durchgeführt und die Testabdeckung sichergestellt werden,
2. kann bei der Veränderung von Programmcode der Evaluator die Auswirkungen auf SFR-Enforcing, Supporting oder Non-Interfering SFRs einfacher herausfinden und damit die Kosten und den zeitlichen Aufwand einer Re-Evaluierung senken.
3. kann bei der Veränderung von Programmcode, welcher als SFR-Non-Interfering eingestuft wird, das Maintenance-Verfahren anstelle einer Re-Evaluierung angewandt werden, welches einen erheblichen zeitlichen und damit auch monetären Vorteil gegenüber dem Re-Evaluierungsverfahren darstellt.

9.2 Übergreifende Festlegungen

9.2.1 Interne Mechanismen

9.2.1.1 Zufallszahlen und Schlüssel

Der Konnektor kann zur Erzeugung von Zufallszahlen und Einmalschlüsseln einen Hardware- oder Software-Generator verwenden. Als Quelle für Zufallszahlen kann der Konnektor die gSMC-K verwenden.

9.3 Funktionsmerkmale

9.3.1 Anwendungskonnektor

9.3.1.1 Administration des Informationsmodells

Wie die Administration der persistenten Entitäten und Beziehungen des Informationsmodells im Detail über die bereitzustellende Administrationsoberfläche erfolgt, entscheidet der Hersteller.

Es wird folgende Reihenfolge für die Pflege des Informationsmodells empfohlen.

1. Mandantenübergreifende Administration:
   - Es werden die Entitäten Arbeitsplätze, Clientsysteme mit Authentifizierungsmerkmalen CS-AuthMerkmal und SMC-B_Verwaltet erfasst.
     Die Eingabe der Kartenterminals erfolgt über die Kartenterminalverwaltung.
   - Es wird die Beziehungen zwischen Arbeitsplatz und Kartenterminals „lokal“ und „entfernt (zentral)“ eingepflegt.
2. Mandantenbezogene Administration:
   - Die Definition bzw. Auswahl eines Mandanten bildet den Einstiegpunkt.

- Pro Mandanten werden aus den bereits eingepflegten Entitäten „Kartenterminal", „Arbeitsplatz", „Clientsystem", „SMC-B_Verwaltet" die für den Mandanten im Zugriff erlaubten zugeordnet.
- Pro Mandant erfolgt eine Zuordnung der Arbeitsplätze zu Clientsystemen.
- Pro Mandant erfolgt eine Zuordnung der lokalen Kartenterminals, über die jeweils pro Arbeitsplatz die Eingabe der Remote-PIN erfolgen darf.

9.3.1.2 Vorgehensvariante für das Handling von CardSessions

Das in der [TIP1-A_4560] „Rahmenbedingungen für Kartensitzungen" geforderte Verhalten, ließe sich über folgenden Mechanismus umsetzen:

Verschiedene Clientsystem (oder verschiedene Nutzer an einem Clientsystem) möchten auf Daten der über CardHandle adressierten Smartcard zugreifen.

Für die Zugriffe müssen, je nach Definition der Zugriffsbedingung in der Zielkarte, bestimmte Sicherheitszustände erreicht werden (durch Verifikation einer PIN oder durch C2C). Diese erreichten Sicherheitszustände werden innerhalb einer Karte jeweils an einen logischen Kanäle (bzw. den Basiskanal) gebunden, d. h., das Erhöhen oder Absenken eines Sicherheitszustands wirkt nicht außerhalb des logischen Kanals, in dem die Veränderung verursacht wurde.

Finden nun Clientsystemzugriffe in unterschiedlichen Kontexten (Mandant, Clientsystem, Arbeitsplatz und Nutzer verschieden) auf die gleiche Karte statt, so muss sichergestellt sein, dass PIN-Eingaben und durchgeführte C2C nur für den Kontext wirksam sind, in welchem sie durchgeführt wurden. Dies ließe sich erreichen, wenn jeder Kontext auf einen eigenen logischen Kanal der Karte abgebildet würde. Leider unterstützen der HBA und die SMC-B nur vier, die eGK nur einen logischen Kanal. Mehrere gleichzeitige unterschiedliche Kontexte wären somit nicht möglich.

Eine mögliche Lösung für beliebig viele gleichzeitige Kontexte:

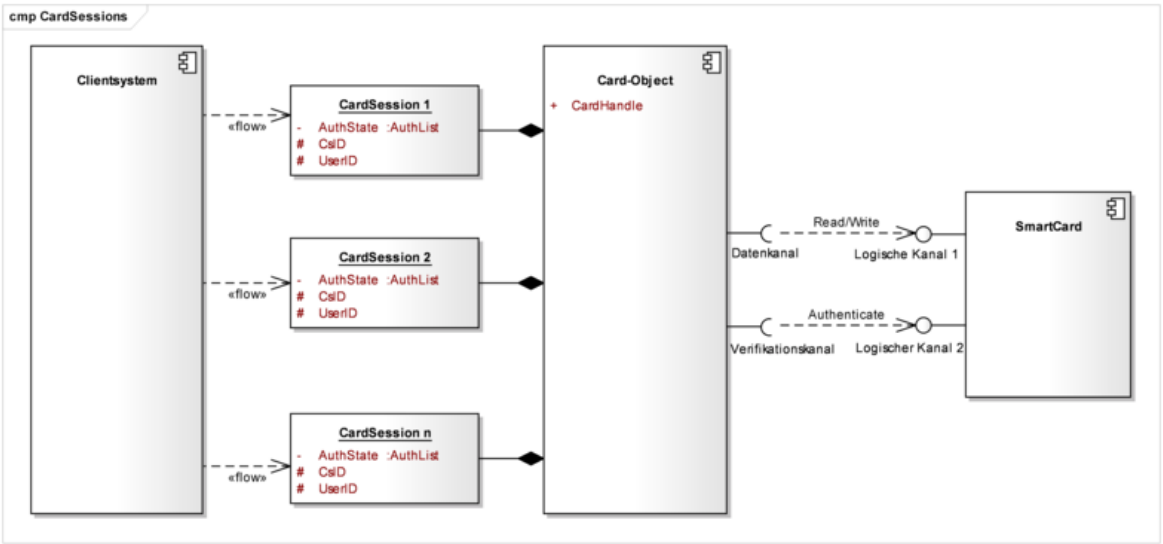


Abbildung 23: PIC_KON_120 Abbildung von CardSessions auf logische Kanäle

Der Kartendienst fungiert als Multiplexer. Er spiegelt die Zugriffsrechte der Karte und wendet deren Regeln selbständig gegen die unterschiedlichen Zugriffe durch Clientsysteme an.

Für jedes Card-Objekt wird „in Richtung Clientsystem" pro Kontext genau eine CardSession erzeugt und verwaltet. Zugriffe des Clientsystems erfolgen somit „im Kontext" einer CardSession.

In Richtung Karte verwendet das Card-Object genau zwei Kanäle (zwei logische oder einen logischen und einen Basiskanal):

- Einen Datenkanal, über den die Datenbewegungen und kryptographischen Operationen laufen und
- Einen Verifikationskanal, der ausschließlich für Authentisierungszwecke verwendet wird

In jeder CardSession werden die in ihrem Kontext erreichten Sicherheitszustände vermerkt. Das Vorgehen für die Durchführung der Verifikationen und des Vermerkens der erreichten Sicherheitszustände, sowie der Datenzugriffe folgt folgenden Regeln (hier für PIN-Verifikation, sinngleich auch für C2C):

- Soll über eine CardSession eine PIN-Verifikation für PinRef_A gegen eine Karte durchgeführt werden und der erhöhte Sicherheitszustand für PinRef_A ist noch nicht erreicht (bsp. direkt nach einem Karten-Reset), dann leite die Verifikation über den Datenkanal (initiale Freischaltung des Datenkanals für folgende Datenzugriffe).
- Soll über eine CardSession eine PIN-Verifikation für PinRef_A gegen eine Karte durchgeführt werden und der erhöhte Sicherheitszustand für PinRef_A ist auf dem Datenkanal bereits erreicht, dann leite die Verifikation über den Verifikationskanal.
- Wurde durch eine CardSession eine Verifikation für PinRef_A erfolgreich durchgeführt, wird dieser erreichte Sicherheitszustand für PinRef_A in der zugreifenden CardSession vermerkt
- Datenzugriffe auf oder Kryptooperationen mit Karten werden durch den Kartendienst nur zugelassen, wenn die zugreifende CardSession über einen für diese Zugriffe benötigten erhöhten Sicherheitszustand verfügt. Ist der benötigte Vermerk für die zugreifende CardSession nicht vorhanden, beantwortet der Kartendienst die Anfrage mit der passenden Kartenfehlermeldung. Es erfolgt keine Interaktion mit der Karte.

Diese Regeln führen dazu, dass eine durch CardSession Y fehlgeschlagene Verifikation für PinRef_A die zuvor erfolgreich durch CardSession X durchgeführt Verifikation nicht beeinflusst. Kartenzugriffe auf dem Datenkanal sind für CardSession X weiterhin möglich, da der Verlust des erhöhten Sicherheitszustands durch fehlerhafte Verifikation immer nur im Verifikationskanal erfolgt.

Dieser Mechanismus funktioniert mit zwei Kanälen zu einer Karte für beliebig viele CardSessions.

9.3.1.3 Darstellung von Terminal-Anzeigen auf einem Kartenterminal

Die folgenden Ausführungen dienen der Klarstellung für die korrekte Verwendung der zur Verfügung stehenden Datenobjekte (DO) zur Darstellung von Terminal-Anzeigen an einem Kartenterminal nach SICCT- und eHealth-Kartenterminal-Spezifikation.

Die SICCT-Spezifikation enthält eine Liste von Datenobjekten (DO), die von den Kartenterminals unterstützt werden müssen oder können. Dabei gibt es zwei Datenobjekte zur Anzeige von Terminal-Anzeigen: APPL DO und SMTBD DO.

Kartenterminals müssen APPL DO (steht für Application Label Data Object) unterstützen. APPL DOs müssen immer eine 7 Bit ISO646DE/DIN66003-Codierung enthalten [DIN66003].

Kartenterminals können SMTBD DO (steht für SICCT Message-To-Be-Displayed Data Object) unterstützen, müssen dieses aber nicht. Über SMTBD DOs können weitere Zeichensätze am Display angezeigt werden.

Der Konnektor soll APPL DOs verwenden. Er kann SMTBD DOs verwenden, wenn er sicherstellt, dass das angesteuerte Kartenterminal diese unterstützt und die dargestellte Meldung der Klartextmeldung entspricht, die mittels APPL DO erreicht worden wäre.

Um dem Kartenterminal den Umbruch längerer Texte über das Zeilenende hinaus zu erleichtern, enthalten die Terminal-Anzeigen das Zeichen 0x0B als „Soll-Zeilenumbrüche". Die „Soll-Zeilenumbrüche" werden nicht als Textzeichen gezählt. Sie zeigen einen potentiellen Zeilenumbruch an. Diese müssen vom Kartenterminal herausgefiltert werden und werden nicht durch andere Zeichen ersetzt.

Die Maximallänge für Terminal-Anzeigen beträgt ohne PIN-Eingabe (OUTPUT [O]) 48 Zeichen.

Besonderheit bei Terminal-Anzeigen, die zu einer PIN-Eingabe (INPUT [I]) auffordern:

Für die PIN-Eingabe wird eine strukturierte Terminal-Anzeige übergeben, aufgeteilt auf maximal 40 Zeichen für die Terminal-Anzeige plus maximal 10 Zeichen für den sog. PIN-Prompt (bei Platz für zusätzliche 6 Zeichen für die PIN-Eingabe). Ein gültiger String hat die Form: <Terminal-Anzeige>0x0F<PIN-Prompt>. Auch die Terminal-Anzeige für Eingaben soll mit „Soll-Zeilenumbrüchen" versehen werden.

Bei der Übertragung der Terminal-Anzeige ist auf die korrekte Codierung der Zeichenkette zu achten. Der einzige Zeichensatz, der von allen Kartenterminals unterstützt werden MUSS, ist (7 Bit) ISO646DE/DIN66003 [DIN66003]. Dadurch darf eine Terminal-Anzeige auch deutsche Sonderzeichen enthalten.

Hex Code	**...0**	**...1**	**...2**	**...3**	**...4**	**...5**	**...6**	**...7**	**...8**	**...9**	**...A**	**...B**	**...C**	**...D**	**...E**	**...F**
0...	*diverse Steuerzeichen - nicht verwendet -*															
1...	*diverse Steuerzeichen - nicht verwendet -*															
2...	*space*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	*del*

Abbildung 24: PIC_KON_007 Übersicht Zeichensatz ISO646DE/DIN66003

10 Anhang K – Szenarien im dezentralen Umfeld

Die folgenden Szenarien für den Einsatz der Produkte der Telematikinfrastruktur beschreiben informativ Varianten und Optionen, die durch die Spezifikationen abgedeckt werden.

Die vorliegenden Abbildungen in diesem Anhang fokussieren auf das dezentrale Umfeld und verzichten daher auf die Darstellung der zentralen Anteile, wie das zentrale Netzwerk der Telematikinfrastruktur, welches über den „VPN-Konzentrator TI" erreichbar ist.

Der Konnektor, sowie die Netzwerkkomponenten Switch und IAG (Internet Access Gateway) sind in den folgenden Szenarien zum Schutz vor unerlaubtem Zugriff gemäß den Annahmen des Sicherheitskonzeptes vor unbefugten physischen Zugriffen geschützt installiert.

Die folgenden Abschnitte stellen jeweils ein Szenario in der Übersicht als Diagramm, eine Beschreibung sowie eine kurze Auflistung der Voraussetzungen und Auswirkungen dar.

10.1 Szenario 1: Einfache Installation ohne spezielle Anforderungen und ohne bestehende Infrastruktur

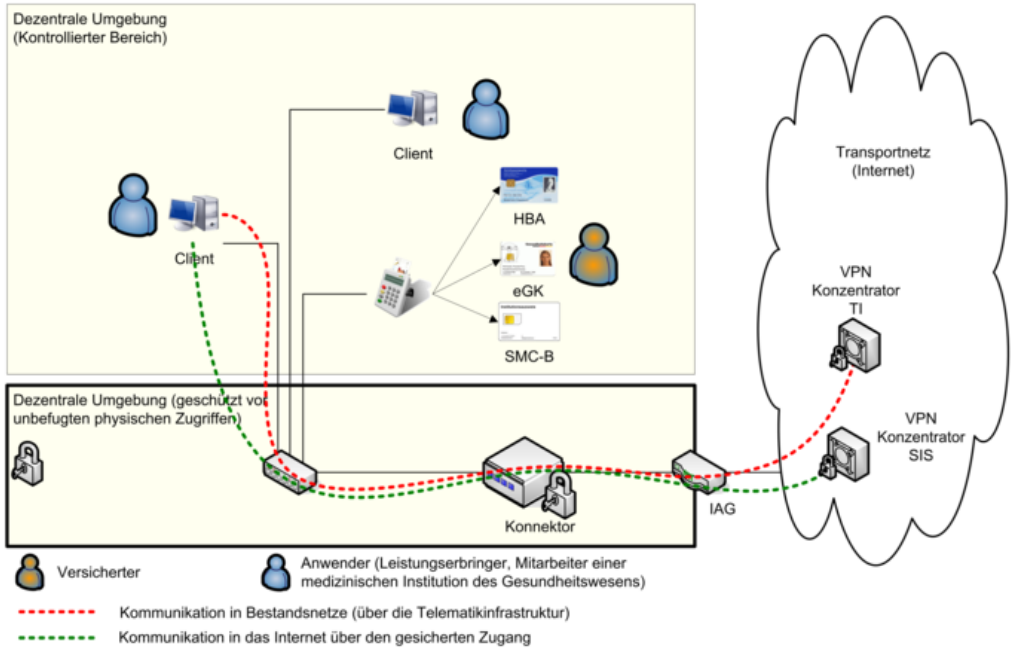


Abbildung 25: Szenario einer einfachen Installation

10.1.1 Beschreibung des Szenarios

Abbildung 25 zeigt ein einfaches Szenario für das dezentrale Umfeld. Es wird der Konnektor als Default-Gateway für jegliche IP-Kommunikation aus dem LAN in das WAN eingesetzt. Dabei übernimmt der Konnektor das Routing der Kommunikation in das Internet über den SIS (Secure Internet Service) und in die an die TI angeschlossenen Bestandsnetze. Die Bezeichnung IAG (Internet Access Gateway) steht für die Geräte, die

den Internetzugang ermöglichen und typischerweise vom Internet Service Provider (ISP) zur Verfügung gestellt werden (z.B. DSL Router und DSL Modem).

Ein oder mehrere Clientsysteme können über den Konnektor Anwendungsfälle der Telematikinfrastruktur initiieren und über den Konnektor und die zentrale TI-Plattform in Bestandsnetze kommunizieren (rote gestrichelte Linie). Dabei ist die Nutzung der Anwendungsfälle der Telematikinfrastruktur je nach Konfiguration des Konnektors entweder nur authentifizierten Clients möglich oder beliebigen Clients.

In diesem einfachen Szenario werden über ein einziges Kartenterminal die SMC-B, der HBA und auch die eGK des Versicherten gelesen, es können dazu alternativ auch mehrere Kartenterminals genutzt werden.

Darüber hinaus können die Clientsysteme über den SIS (Secure Internet Service) auf Dienste des Internets zugreifen.

10.1.2 Voraussetzungen

- Anbindung der bestehenden Clientsysteme an ein zum Konnektor kompatibles LAN muss möglich sein.
- Konfiguration des Konnektors als Default-Gateway in den Clientsystemen und Konfiguration der notwendigen VPN-Tunnel im Konnektor, um in die verschiedenen Netze zu routen.
- Verfügbarkeit einer SMC-B

10.1.3 Auswirkungen

- Die Clientsysteme können über den Konnektor Anwendungsfälle der TI initiieren
- Die Clientsysteme können über den Konnektor auf das Internet und Bestandsnetze zugreifen

10.2 Szenario 2: Installation mit mehreren Behandlungsräumen

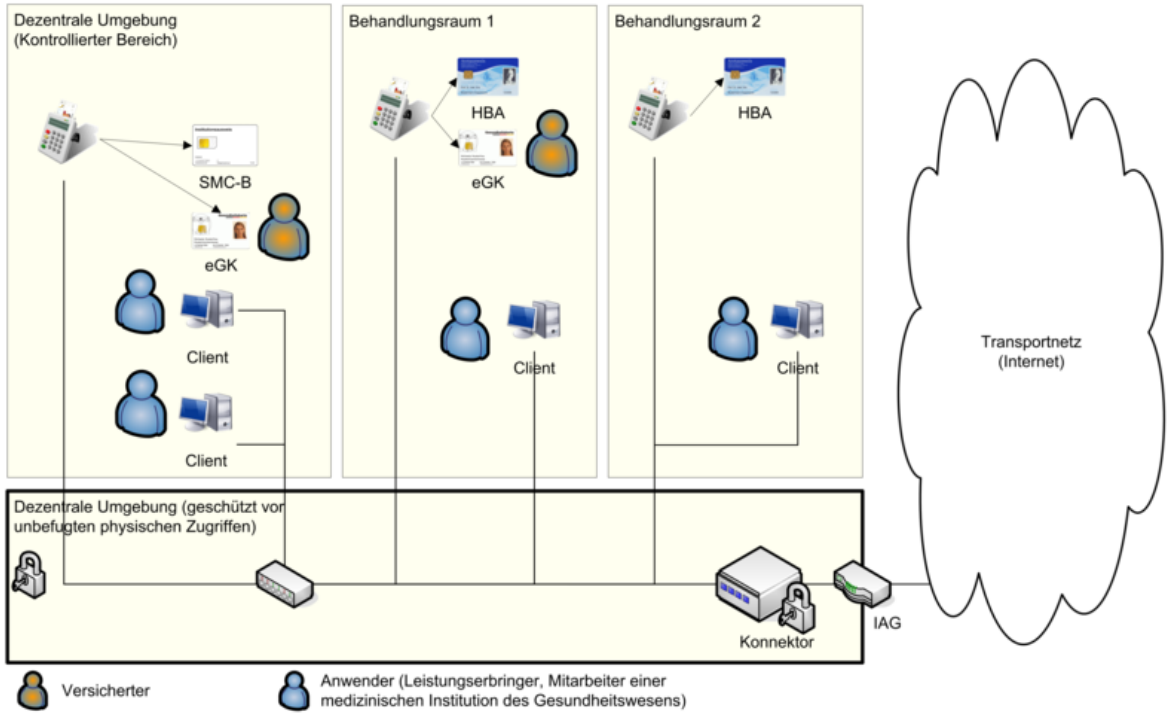


Abbildung 26: Szenario einer Installation mit mehreren Behandlungsräumen

10.2.1 Beschreibung des Szenarios

Mit der in Szenario 1 skizzierten Topologie kann auch ein Szenario bedient werden, bei dem mehrere Behandlungsräume unterstützt werden (siehe Abbildung 26). Dabei ist in jedem Behandlungsraum mindestens ein Kartenterminal vorzusehen, so dass die eGK gelesen werden kann.

Auf die Darstellung der Kommunikationswege in zentrale Netze wurde in Abbildung 26 verzichtet, da sich hier keine Änderung gegenüber Szenario 1 ergibt.

Durch die Ressourcenverwaltung des Konnektors wird sichergestellt, dass bei Anwendungsfällen diejenigen Kartenterminals angesprochen werden, welche dem Arbeitsplatz zugeordnet sind, von dem aus der Anwendungsfall initiiert wurde.

10.2.2 Voraussetzungen

- Anbindung der bestehenden Clientsysteme an ein zum Konnektor kompatibles LAN muss möglich sein.
- Konfiguration des Konnektors als Default-Gateway in den Clientsystemen und Einrichtung der notwendigen VPN-Tunnel im Konnektor, um in die verschiedenen Netze zu routen.
- Verfügbarkeit einer SMC-B und mehrerer Kartenterminals und Clientsysteme

- Die Clientsysteme und Kartenterminals und deren Relationen sind dem Konnektor über Konfiguration bekannt gemacht worden.

10.2.3 Auswirkungen

- Die Clientsysteme können über den Konnektor Anwendungsfälle der TI initiieren
- Die Clientsysteme können über den Konnektor auf das Internet (über den SIS) und Bestandsnetze zugreifen
- Der HBA-Inhaber muss seinen HBA mit sich führen und kann diesen in den einzelnen Kartenterminals der Behandlungsräume nutzen.

10.3 Szenario 3: Integration in bestehende Infrastruktur ohne Netzsegmentierung

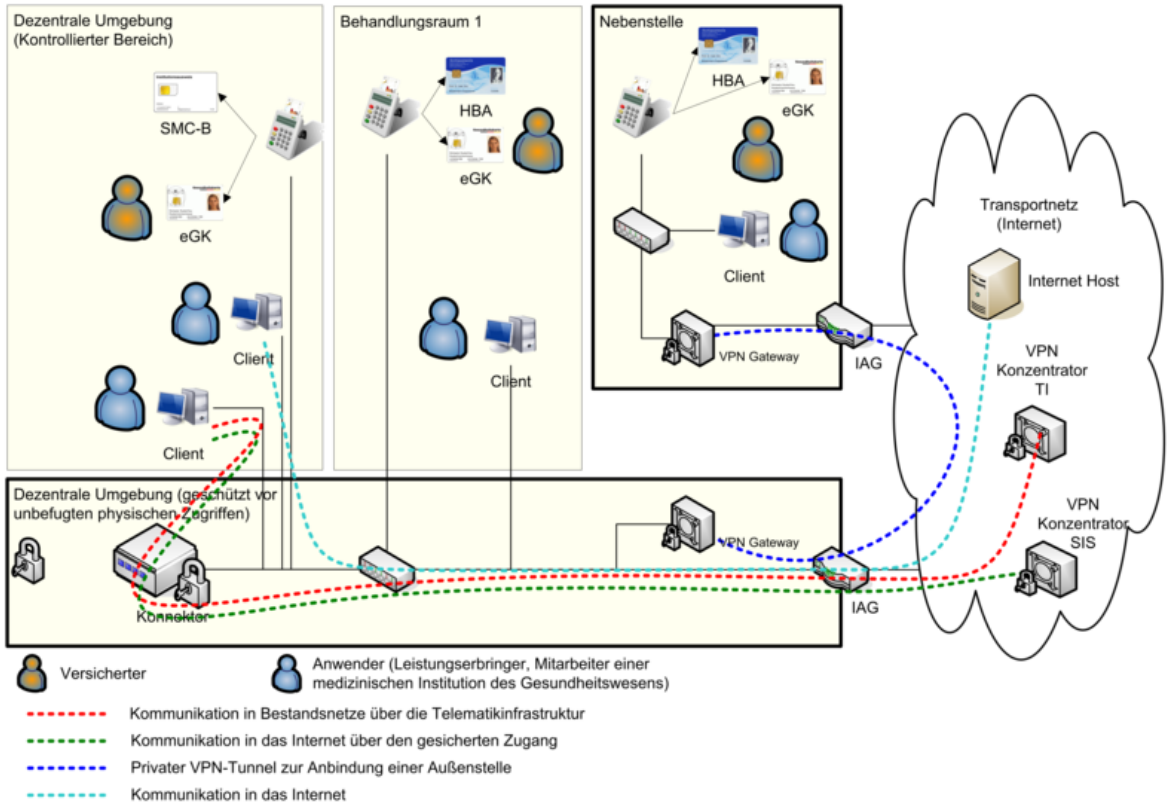


Abbildung 27: Szenario einer Integration der TI Produkte in eine bestehende Infrastruktur

10.3.1 Beschreibung des Szenarios

Im Falle einer bereits vorhandenen Infrastruktur im dezentralen Bereich können die Produkte der TI, insbesondere der Konnektor, so in die Infrastruktur integriert werden, dass Bestandsanwendungen bereits erprobte Kommunikationswege weiter nutzen können.

Wie in Abbildung 27 beispielhaft dargestellt, existiert bereits eine Infrastruktur, die sowohl einen Internetzugang für die Arbeitsplätze ermöglicht (gestrichelte Linie in türkis), als auch eine Nebenstelle über VPN anbindet (gestrichelte Linie in blau). In diesem Fall wird der Konnektor als zusätzliches Gerät an das bestehende Netzwerk angeschlossen und nutzt den bereits vorhandenen Internetanschluss zur Kommunikation in die TI.

Für die Clientsysteme muss in diesem Szenario je nach individuellem Anforderungsprofil entschieden werden, ob das jeweilige Clientsystem über die Telematikinfrastruktur kommunizieren können soll und den gesicherten Internetzugang (SIS) nutzen soll oder nicht.

Soll ein Clientsysteme nicht über die Telematikinfrastruktur kommunizieren, bleibt der IAG als Default-Gateway dieses Clientsystems konfiguriert. In diesem Fall routet der IAG die eingehenden IP-Pakete mit öffentlichen Zieladressen weiter in das Internet. Die gestrichelte Linie in türkis zeigt beispielhaft einen Zugriff in das Internet.

Soll ein Clientsystem über die Telematikinfrastruktur kommunizieren oder den gesicherten Internetzugang (SIS) nutzen, muss der Konnektor als Default-Gateway konfiguriert werden. In diesem Fall routet der Konnektor die eingehenden IP-Pakete, die nicht für ihn bestimmt sind, entweder durch den VPN-Tunnel der TI über die Telematikinfrastruktur in ein angeschlossenes Bestandsnetz, (gestrichelte Linie in rot) oder durch den VPN-Tunnel zum SIS (Secure Internet Service) in das Internet (gestrichelte Linie in grün). Sollte kein sicherer Internetzugang konfiguriert sein, so würde der Konnektor den Traffic verwerfen und ggf. per ICMP dem Client eine anderes Gateway (IAG) vorschlagen. Alternativ können die von den Clients benötigten Routing-Informationen manuell oder per DHCP konfiguriert werden.

10.3.2 Voraussetzungen

- Konnektor ist kompatibel zur bestehenden Infrastruktur (Vernetzung)
- Die bestehende Infrastruktur verfügt über einen Internetzugang
- Verfügbarkeit einer SMC-B und mehrerer Kartenterminals
- Die Clientsysteme und Kartenterminals und deren Relationen sind dem Konnektor über Konfiguration bekannt gemacht worden.

10.3.3 Auswirkungen

- Produkte der Telematik können „minimal-invasiv" in die bestehende Infrastruktur integriert werden. Bestehende Kommunikationswege können weiter genutzt werden.
- Für Clients kann je nach individuellen Anforderungsprofil der sichere Internetzugang über den Konnektor genutzt werden oder der direkte Internetzugang über den bestehenden IAG

10.4 Szenario 4: Integration in bestehende Infrastruktur mit Netzsegmentierung

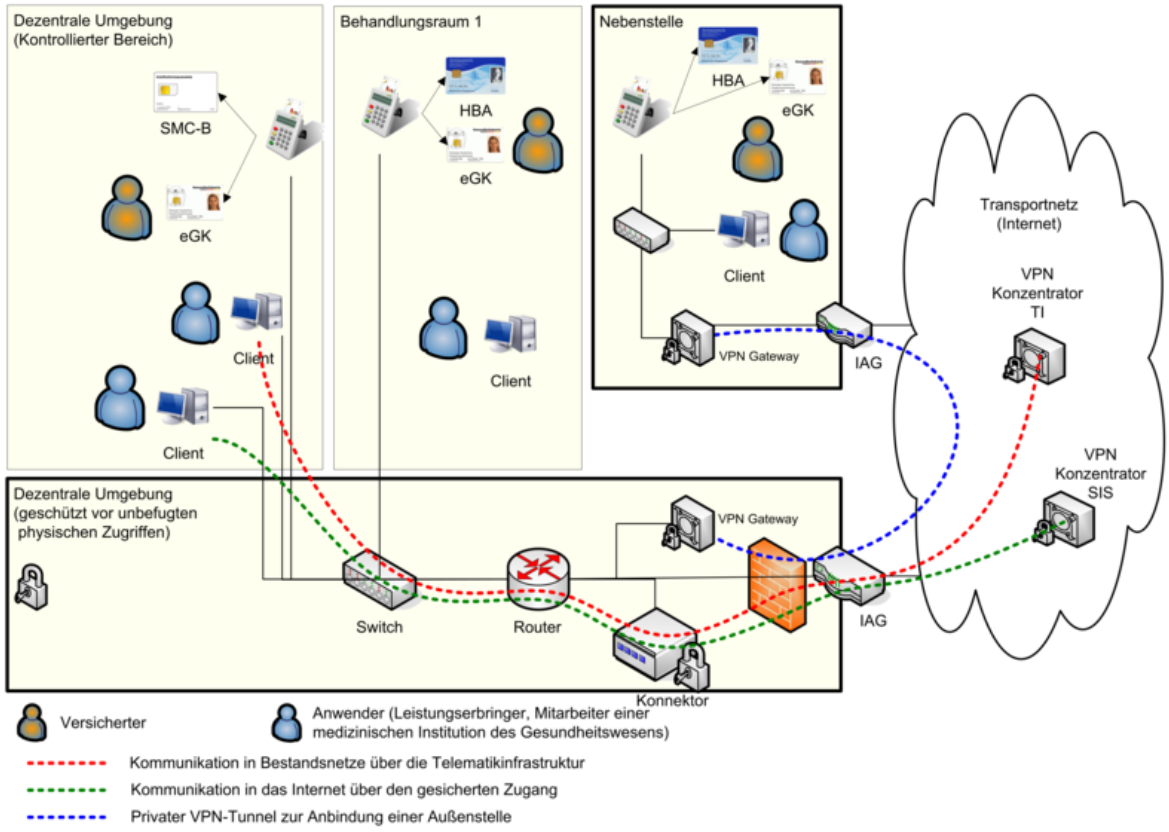


Abbildung 28: Szenario einer Integration der TI Produkte in eine bestehende Infrastruktur mit existierendem Router

10.4.1 Beschreibung des Szenarios

Das vorliegende Szenario skizziert eine etwas komplexere dezentrale Umgebung, in der das Netzwerk segmentiert ist und dedizierte Router als Default-Gateway für die Clientsysteme genutzt werden. In diesem Fall kann die Konfiguration der Clients unverändert bleiben und der Konnektor wird als zusätzliches Gerät in das Netzwerk integriert und dem Router bekanntgemacht als Gateway für den sicheren Internetzugang und den Zugang zu den an die Telematikinfrastruktur angeschlossenen Bestandsnetze.

10.4.2 Voraussetzungen

- Konnektor ist kompatibel zur bestehenden Infrastruktur (Vernetzung)
- Verfügbarkeit einer SMC-B und mehrerer Kartenterminals
- Der Konnektor ist dem bestehenden Router als Gateway bekannt gemacht.

10.4.3 Auswirkungen

- Produkte der Telematik können „minimal-invasiv" in die bestehende Infrastruktur integriert werden. Bestehende Kommunikationswege können weiter genutzt werden.
- Die Default-Gateway-Konfiguration der Clients muss nicht geändert werden.

10.5 Szenario 5: Zentral gesteckter HBA

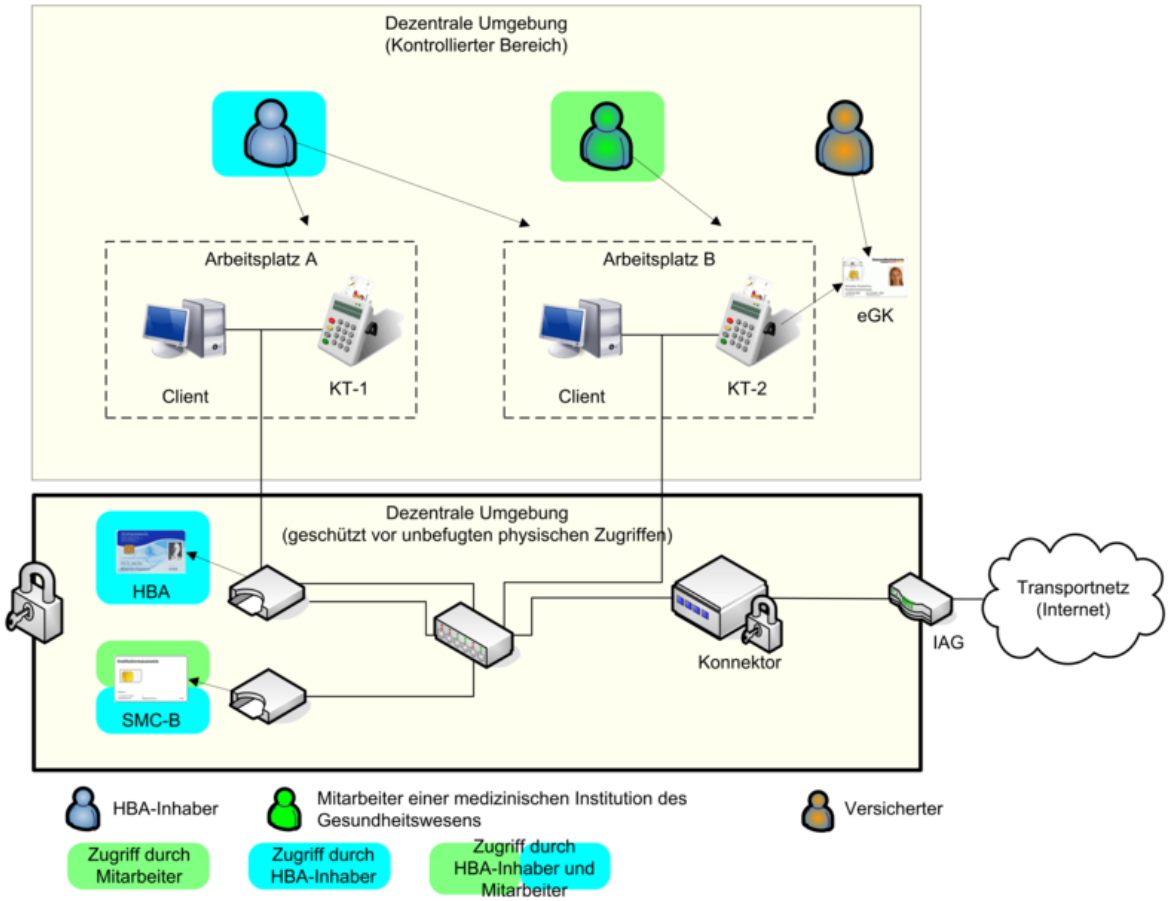


Abbildung 29: Szenario mit zentral gesteckten HBA und SMC-B

10.5.1 Beschreibung des Szenarios

Dieses Szenario zeichnet sich dadurch aus dass ein HBA nicht durch seinen Inhaber mitgeführt und am Arbeitsplatz gesteckt wird, sondern zentral und geschützt vor unbefugten physischen Zugriffen gesteckt bleibt.

Der HBA-Inhaber greift über jeden konfigurierten Arbeitsplatz auf seinen HBA zu. Die Remote-PIN-Eingabe erfolgt unter Verwendung des lokal am Arbeitsplatz vorhandenen eHealth-Kartenterminals.

Die Mechanismen zum Zugriff auf eine zentral gesteckte SMC-B funktionieren analog zum HBA.

10.5.2 Voraussetzungen

Folgende zusätzliche Punkte müssen erfüllt sein, um dieses Szenario umzusetzen:

- Stecken der zentral gesteckten Karten HBA und SMC-B (ohne direkte Aufsicht) und Sicherstellung des Schutzes vor unbefugtem physischen Zugriff
- Konfiguration im Konnektor: Lokales eHealth-Kartenterminals als lokales eHealth-Kartenterminal für eine Remote-PIN-Eingabe eines bestimmten Arbeitsplatzes.
  Im abgebildeten Beispiel KT-1 für Arbeitsplatz A und KT-2 für Arbeitsplatz B.
- Konfiguration im Konnektor: Assoziation der gewünschten Arbeitsplätze zum jeweiligen Kartenterminal mit zentral gesteckter Karte.
  Im abgebildeten Beispiel Arbeitsplatz A assoziiert mit dem eHealth-Kartenterminal des HBAs und Arbeitsplatz B mit eHealth-Kartenterminal des HBAs und dem eHealth-Kartenterminal der SMC-B.

10.5.3 Auswirkung

- HBA muss nicht mehr durch seinen Inhaber mitgeführt werden
- SMC-B muss nicht mehr unter ständiger Aufsicht eines Mitarbeiters einer Organisation des Gesundheitswesens sein.

10.6 Szenario 6: Installation mit zentralem PS

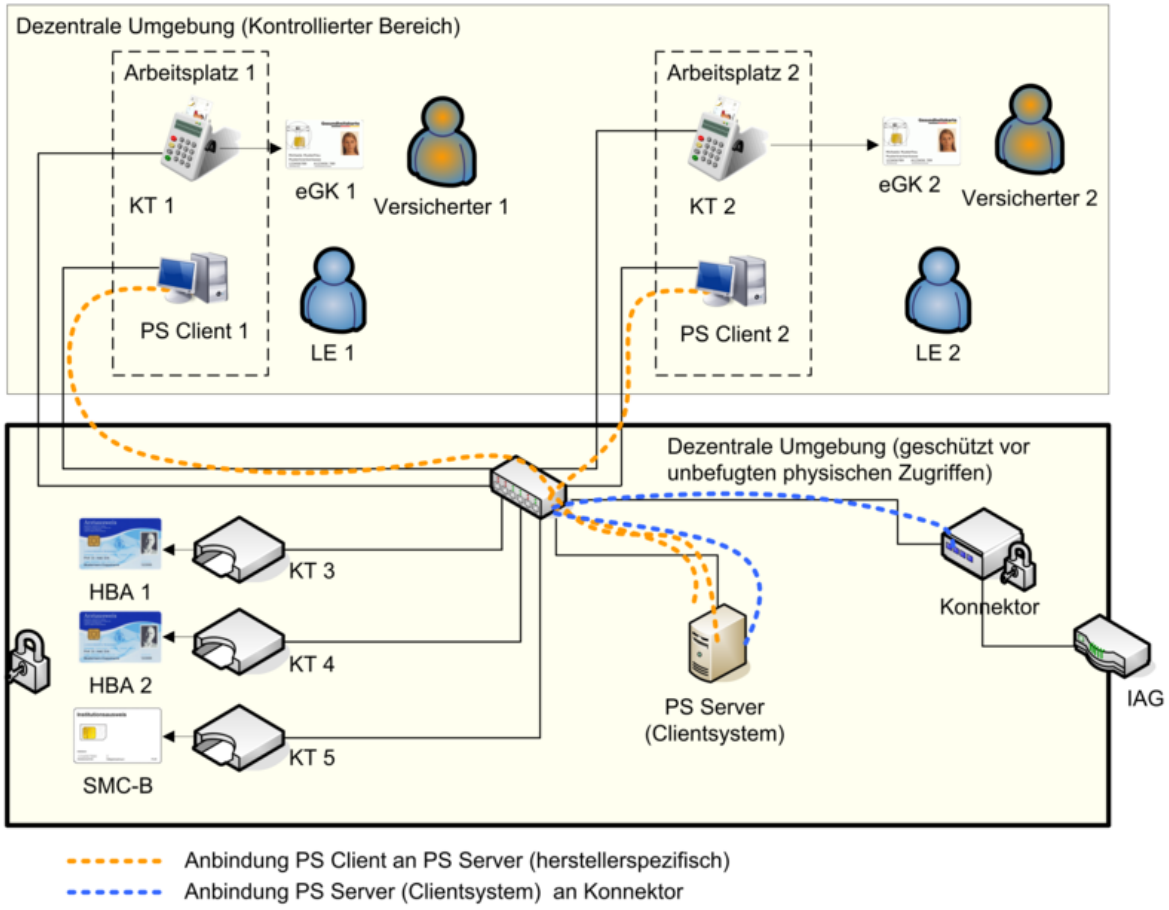


Abbildung 30: Szenario mit zentralem Primärsystem als Clientsystem

10.6.1 Beschreibung des Szenarios

Das Szenario skizziert eine dezentrale Konfiguration, bei der das Primärsystem aus einem Serveranteil „PS Server" und mehreren Clientanteilen „PS Client" besteht. Die Anbindung zwischen dem „PS Server" und den „PS Clients" ist herstellerspezifisch. Der „PS Server" fungiert als ein einziges Clientsystem gegenüber der TI bzw. dem Konnektor (z.B. als Terminalserver). Die Clientsystemschnittstelle des Konnektors wird ausschließlich vom „PS Server" genutzt. Der „PS Server" muss bei der Kommunikation mit dem Konnektor eine Übersetzung der zugreifenden „PS Clients" auf die entsprechende Entität „Arbeitsplatz" des Konnektors durchführen

Beispielhaft zeigt das Szenario zwei Arbeitsplätze mit jeweils einem Kartenterminal für die eGK sowie zentral gesteckte SMC-B und HBAs. Alternativ sind auch lokal am Arbeitsplatz gesteckte HBAs möglich.

10.6.2 Voraussetzungen

- Netzanbindung aller Komponenten (u. a. KT, PS Client, PS Server, Konnektor) in der dezentralen Umgebung bis einschließlich zur Netzwerkschicht (IP-Ebene)
- Konfiguration des Primärsystems mit seinen Anteilen „PS Server" und ggf. mehreren „PS Clients" passend zum Informationsmodell des Konnektors (herstellerspezifisch).
- Konfiguration des Konnektors. U. a.:
  - Informationsmodel:
    Beim Beispielszenario u.a Entitäten „Clientsystem" für „PS Server", „Arbeitsplatz" für „Arbeitsplatz 1" und Arbeitsplatz 2", „Kartenterminal" und „KT-Slot" für „KT 1" – „KT 5", „Mandat" für die vorgesehene Anzahl von Mandaten, „SM-B_Verwaltet" sowie entsprechende Entitätenbeziehungen.
  - Anbindung PS Server (ggf. über TLS)
  - Pairing der Kartenterminals
- Gesteckte Karten (SMC-B, HBA, eGK)
- Anmeldung Nutzer am „PS Client"

10.6.3 Auswirkungen

- An den verschiedenen Arbeitsplätzen können für die definierten Mandaten und Nutzer Anwendungsfälle der TI initiiert werden.
- HBA-Inhaber müssen entsprechen der gewählten HBA-Deployment-Varianten
  - ihren HBA zentral stecken und über das Remote-PIN-Verfahren zugreifen
  - ihren HBA mit sich führen und lokal in Kartenterminal der Arbeitplätze stecken

10.7 Szenario 7: Mehrere Mandanten

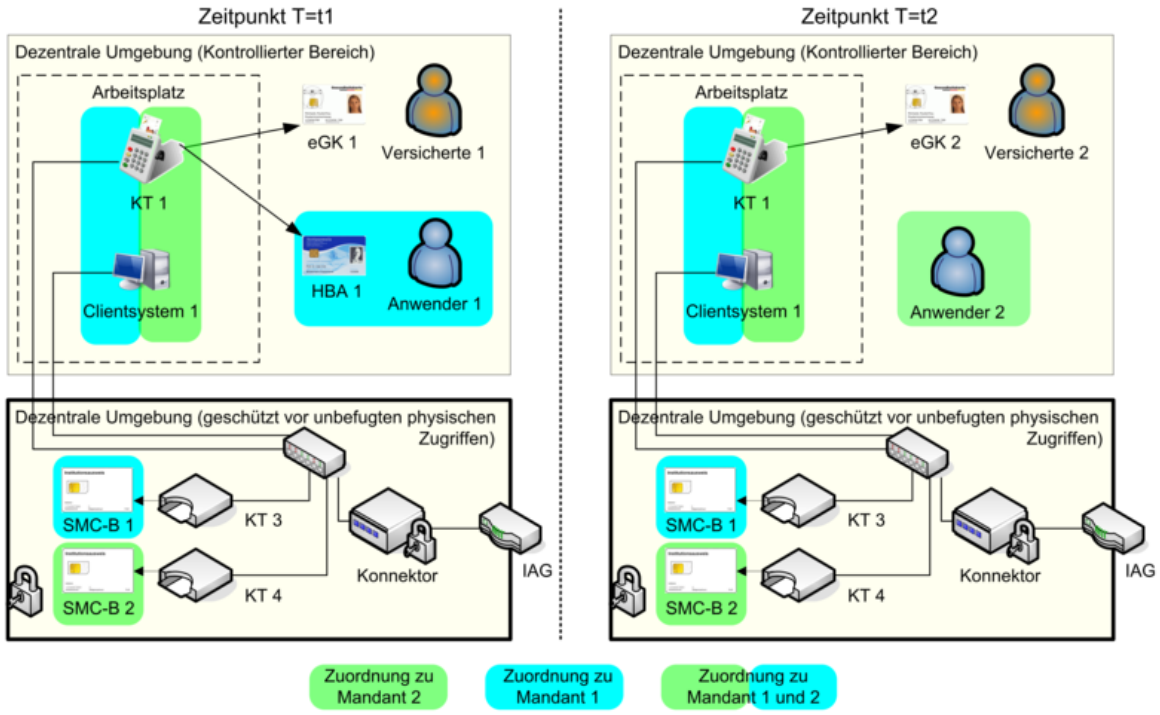


Abbildung 31: Szenario für den Zugriff

10.7.1 Beschreibung des Szenarios

Das Szenario skizziert eine dezentrale Konfiguration, bei der mehrere Mandanten vorhanden sind, wobei jedem Mandant eine eigene SMC-B zugeordnet ist. Die SMC-Bs sind zentral zusammen mit dem Konnektor geschützt vor unbefugten physischen Zugriffen installiert. Die Komponenten Arbeitplätze, Clientsysteme und Kartenterminals müssen eine Zuordnung zum Mandanten haben, wobei Zuordnungen zu mehreren Mandaten möglich sind. Das Beispiel zeigt einen Arbeitplatz mit „Clientsystem 1" und „KT 1", der zu unterschiedlichen Zeiten durch verschiedene Mandaten verwendet wird. Zum Zeitpunkt T=t1 greift ein Anwender 1 mit HBA 1 über einen Anwendungsfall im Kontext Mandat 1 auf die TI zu, wobei der Versicherte 1 mit eGK 1 am Anwendungsfall beteiligt ist. Zum Zeitpunkt T=t2 wird ein anderer Anwendungsfall im Kontext von Mandat 2 durch einen Anwender 2 ohne HBA initiiert, wobei der Versicherte 2 mit eGK 2 am Anwendungsfall beteiligt ist. Das Clientsystem stellt hierbei den Mandantenbezug sowie die Nutzer Authentisierung sicher. Als Variante können auch mehrere Mandanten eine Zuordnung zu einer einzelnen SMC-B haben. Weiterhin können auch in diesem Szenario HBAs zentral gesteckt werden.

10.7.2 Voraussetzungen

- Netzwerkanbindung aller Komponenten (u. a. KT, Clientsystem, Konnektor) in der dezentralen Umgebung bis einschließlich zur Netzwerkschicht (IP-Ebene)
- Konfiguration der Clientsysteme („Clientsystem 1"), passend zum Informationsmodell des Konnektors (herstellerspezifisch).

- Konfiguration des Konnektors. U. a.:
  - Konfiguration Konnektor:
    Beim Beispielszenario u.a Entitäten „Clientsystem" für „Clientsystem 1", „Arbeitsplatz" für „Arbeitsplatz 1", „Kartenterminal" und „KT-Slot" für „KT 1" – „KT 4", „Mandat" für „Mandant 1" und „Mandant 2", „SM-B_Verwaltet" für „SMC-B 1" und SMC-B 2" sowie entsprechende Entitätenbeziehungen
  - Anbindung „Clientsystem 1" (ggf. über TLS)
  - Pairing der Kartenterminals
- Gesteckte Karten (SMC-B 1, SMC-B 2, HBA 1, eGK 1, eGK 2)
- Anmeldung eines Anwenders mit Mandantenbezug am Clientsystem

10.7.3 Auswirkungen

- An den verschiedenen Arbeitsplätzen können für die definierten Mandaten und Anwender Anwendungsfälle der TI initiiert werden.
- HBA-Inhaber müssen entsprechen der gewählten HBA-Deployment-Varianten
  - ihren HBA zentral stecken und über das Remote-PIN-Verfahren zugreifen
  - ihren HBA mit sich führen und lokal in Kartenterminal der Arbeitplätze stecken

10.8 Szenario 9: Standalone Konnektor - Physische Trennung

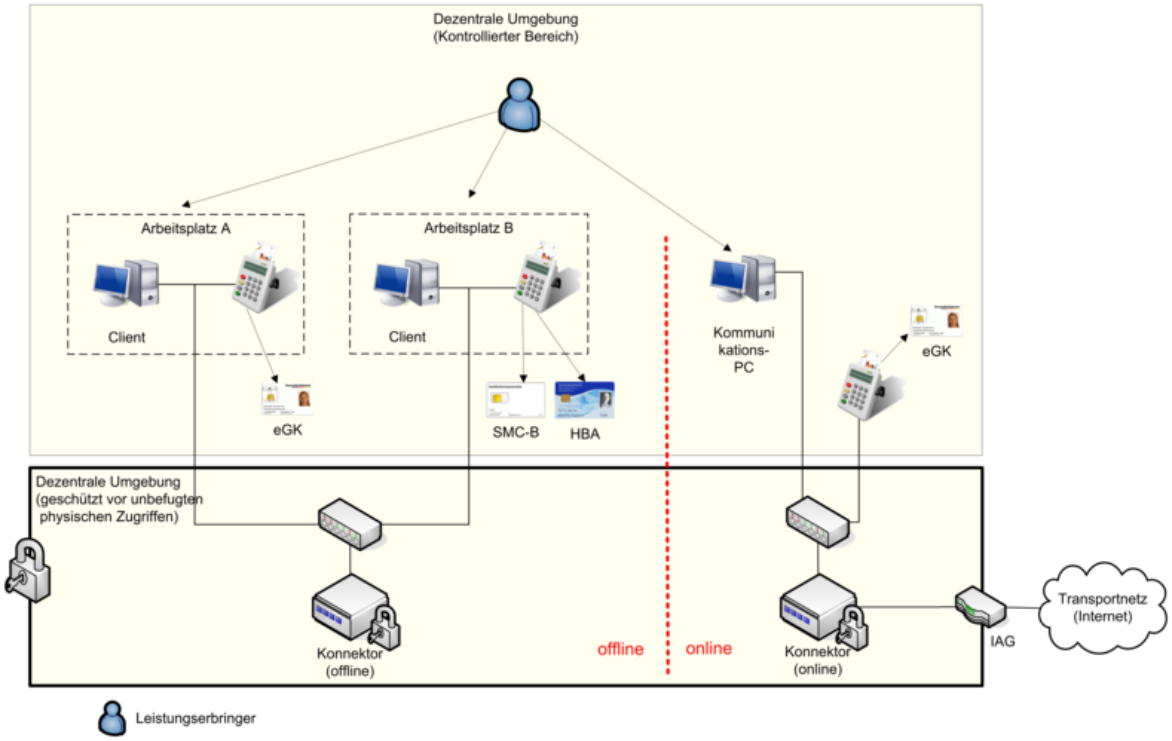


Abbildung 32: Standalone-Szenario mit physischer Trennung im Konnektor

10.8.1 Beschreibung des Szenarios

Dieses Szenario stellt eine Variante des Standalone-Szenarios dar, bei dem eine physische Trennung der Konnektoren eingesetzt wurde.

Im Standalone-Szenario besteht eine Trennung zwischen den Praxissystemen der dezentralen Umgebung, welche offline (also, ohne Anbindung an die zentrale TI-Plattform) betrieben werden und den für das Update der eGK durch die Fachanwendung VSDM notwendigen Komponenten, welche online (also, mit Verbindung in die zentrale TI-Plattform) betrieben werden.

Die physische Trennung im Standalone-Szenario zeichnet sich dadurch aus, dass getrennte Komponenten zum Einsatz kommen. Der Online-Konnektor ist mit der zentralen TI-Plattform verbunden und ermöglicht das VSDM Update der eGKs. Ein am Online-Konnektor angebundener Kommunikations-PC kann darüber hinaus über den sicheren Internetzugang der TI auf das Internet und über den VPN-Konzentrator TI auf Bestandsnetze zugreifen.

Sollten die Online-/Offline-Systeme nicht netztechnisch voneinander getrennt sein, so obliegt es dem Administrator der Praxissysteme sicherzustellen, dass die netztechnische Verbindung keine Gefährdung für die Praxissysteme zur Folge hat.

Im Offline-Konnektor sind einzelne Funktionen nicht verfügbar, andere haben einen eingeschränkten Funktionsumfang. So kann z.B. eine QES erzeugt oder geprüft aber dabei keine aktuelle Statusauskunft (OCSP-Response) für die eingesetzten Zertifikate eingeholt werden. Dies hat zur Folge, dass bei Erzeugung einer QES keine Statusauskunft für das Signaturzertifikat in die Signatur eingebettet werden kann und bei einer Prüfung der QES nur eine eventuell in die Signatur eingebettet Statusauskunft des Zertifikats berücksichtigt werden kann.

Der Nutzer muss in diesem Fall selber entscheiden ob der gebotene Funktionsumfang für seinen Anwendungsfall ausreichend ist.

10.8.2 Voraussetzungen

Folgende zusätzliche Punkte müssen erfüllt sein, um dieses Szenario umzusetzen:

- Konfiguration im Konnektor: Es muss konfiguriert werden, welche Komponenten von welchem Konnektor (online/offline) verwendet werden dürfen.
- Ein eHealth-Kartenterminal oder ein Arbeitsplatz darf immer nur mit einem der Konnektoren verbunden sein.
- Konfiguration im Konnektor: Im Offline-Konnektor wird kein VPN-Kanal konfiguriert.
- Clients bzw. Kommunikations-PC müssen sicherstellen, dass sie nur den jeweils richtigen Konnektor ansprechen.
- Es sollte eine netztechnische Trennung des Online- und Offline-Segmentes erfolgen. Wird dies nicht umgesetzt, dann obliegt es dem Administrator der Praxissysteme sicherzustellen, dass die netztechnische Verbindung keine Gefährdung für die Praxissysteme zur Folge hat.
  Sollte keine netztechnische Trennung erfolgen, so kann nur einer der Konnektoren als DHCP-Server agieren. Es wird empfohlen hier den Offline-Konnektor zu verwenden, da dort tendenziell mehr Systeme angeschlossen sind. Die am Online-Konnektor angeschlossenen Systeme müssen dann direkt konfiguriert werden.

10.8.3 Auswirkung

- Erhöhter Aufwand durch separate Konnektoren und separate eHealth-Kartenterminals.
- Trennung der Praxissysteme von der zentralen TI-Plattform ist für den Leistungserbringer nachweislich sichergestellt.
- Eingeschränkte Funktionalität der TI für Praxissysteme (nur Offline-Funktionalität)
- Notwendige Prüfung des Leistungserbringers, ob eingeschränkte Funktionalität (insbesondere bei Sicherheitsfunktionen) akzeptabel ist.
- Sicherer Internetzugang der TI nur über den Kommunikations-PC nutzbar.
- Zugang zu Bestandsnetzen über den VPN-Konzentrator TI nur über den Kommunikations-PC nutzbar

11 Anhang L – Datentypen von Eingangs- und Ausgangsdaten

Tabelle 395: Aufzählтypen

Typname	Werteliste
[Boolean]	{true | false}
[EncryptionType]	{CMS | XMLEnc | S/MIME}
[EventType]	{Op | Sec | Perf}
[EventSeverity]	{Debug | Info | Warn | Err | Fatal}
[KtOutputMode]	{Input | OutputWait | OutputConfirm | OutputKeep | OutputErase}
[PinStatus]	{VERIFIED | VERIFYABLE | BLOCKED | TRANSPORT_PIN | EMPTY_PIN | DISABLED}
[PinResult]	{OK | REJECTED | BLOCKED | ERROR}
[PukResult]	{OK | REJECTED | BLOCKED | ERROR}
[VerificationResult]	{VALID | INVALID | INCONCLUSIVE}