gematik
Gesellschaft für Telematikanwendungen der Gesundheitskarte mbH

Einführung der Gesundheitskarte

Implementierungsleitfaden Primärsysteme - Online-Rollout (Stufe 1)

Version:	1.7.0
Revision:	\main\rel_online\rel_ors1\36
Stand:	24.07.2015
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemILF_PS_ORS1]

Dokumentinformationen

Änderungen zur Vorversion

Merge zwischen ORS1 Release 1.4 und KOM-LE, Anpassungen aufgrund der Änderungen im BMV-Ä/EKV (KVK nur noch für den Bereich der Sonstigen Kostenträger ein gültiger Nachweis des Leistungsanspruches)

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
0.5.0	21.08.12		zur Abstimmung freigegeben	gematik
1.0.0	15.10.12		Einarbeitung Gesellschafterkommentare	gematik
			Einarbeitung Kommentare aus der übergreifenden Konsistenzprüfung	gematik
1.1.0	12.11.12		als Arbeitsgrundlage freigegeben	gematik
1.2.0	05.02.13		Anpassung der Kardinalität des Feldes „Vorname“ auf Pflichtfeld gemäß Beschluss des 16. LA	gematik
	27.02.13		Einarbeitung von Änderungen an der Außenschnittstelle des Konnektors	gematik
	18.03.13		Einarbeitung von Änderungen an der Signatur- und Verschlüsselungsschnittstelle des Konnektors	gematik
1.3.0	06.06.13		Einarbeitung Kommentare LA	gematik
1.4.0	15.08.13		Einarbeitung Änderungsliste vom 08.08.2013	gematik
1.5.0	21.02.14		Arbeitsstand zur fachlichen Prüfung durch den LA in Bezug auf die offenen Punkte der Konnektor-Spezifikation	gematik
1.6.0	26.08.14		Einarbeitung Beschluß der Gesellschafterversammlung vom 31.03.2014 zu Befüllungsregeln VSD-Schema	gematik
			Änderungen im BMV-Ä/EKV (KVK nur noch für den Bereich der Sonstigen Kostenträger ein gültiger Nachweis des Leistungsanspruches) eingearbeitet	gematik
	09.09.14		Übernahme des ORS1-Standes zur Einarbeitung der KOM-LE-Änderungen	gematik
1.7.0	24.07.15		freigegeben	gematik

© gematik - öffentlich

Inhaltsverzeichnis

Dokumentinformationen 2

Inhaltsverzeichnis 3

1 Einordnung des Dokuments 6

1.1 Zielsetzung 6

1.2 Zielgruppe 6

1.3 Geltungsbereich 6

1.4 Abgrenzung des Dokuments 7

1.5 Methodik 7

2 Systemüberblick 9

3 Konfiguration 11

3.1 Umgebung des Leistungserbringers 11

3.1.1 Begriffe der Konfigurationseinheiten 11

3.1.2 Beziehungen der Konfigurationseinheiten 12

3.2 Arbeitsplätze in der Leistungserbringerumgebung 13

3.2.1 Online-Szenario 14

3.2.2 Standalone-Szenario mit Online-Konnektor und Offline-Konnektor 14

3.2.3 Szenario mit logischer Trennung 15

3.3 Arbeitsplätze, Mandanten und Kartenterminals konfigurieren 15

3.3.1 Aufrufkontext 16

3.3.2 LE-Umgebungen 17

4 Funktionsmerkmale 18

4.1 Inbetriebnahme 18

4.1.1 Verbindungsaufbau zum Konnektor 20

4.1.1.1 Client Authentisierung *21*

4.1.2 Konnektordienstverzeichnis lesen 21

4.1.3 Nutzung von Webservice-Schnittstellen 23

4.1.4 Ereignisdienst/Systeminformationsdienst 24

4.1.4.1 Ereignismeldungen mittels Protokoll CETP *24*

4.1.4.2 Abonnieren von Ereignissen *27*

4.1.4.3 Ereignisse für Konnektorinformationen *29*

4.1.4.4 Ereignisdienst-Szenario VSDM-Informationen *30*

4.1.5 Karten/PIN-Handling 30

4.1.5.1 PS-Dialoge *30*

4.1.5.2 PIN-Änderung *31*

4.1.5.3 Freischaltung von Karten *32*

4.2 Kartensitzungen 33

4.2.1 Aufbau von Kartensitzungen 33
4.2.1.1 GetCards 33
4.2.1.2 GetCardTerminals 37
4.2.1.3 RequestCard 37
4.2.1.4 Exkurs: Auswurf von Karten mittels EjectCard 38
4.2.2 Kartensitzung eGK 39
4.2.3 Kartensitzung SM-B 39
4.2.4 Kartensitzung HBAx 40

4.3 Fachanwendung VSDM 40
4.3.1 Übersicht 41
4.3.2 Schnittstelle I_VSDService 41
4.3.3 Anwendungsfall „VSD lesen mit/ohne Online-Prüfung“ 43
4.3.4 Abläufe im Primärsystem 47
4.3.4.1 Patientendatensatz anzeigen 47
4.3.4.2 eGK einlesen 48
4.3.4.2.1 Online-Szenario 50
4.3.4.2.2 Standalone-Szenario (Primärsystem mit Offline-Konnektor verbunden) 51
4.3.4.3 Benutzerinteraktionen/Anforderungen 51
4.3.4.3.1 Manuelle Online-Prüfung und -Aktualisierung 53
4.3.4.4 Nutzung der VSDM-Ereignisse des Systeminformationsdienstes 53
4.3.4.5 Beispiele ReadVSD 54
4.3.5 Informationsmodell VSD 57
4.3.5.1 Versichertenstammdaten 57
4.3.5.2 Prüfungsnachweis 58
4.3.5.3 Zeichenkodierung von Daten 59
4.3.5.4 Dekodierung und Schemavalidierung 60
4.3.6 Schnittstelle I_KVKService 60
4.3.7 Datenaustausch mit mobilen Einsatzgeräten 60

4.4 Signaturerstellung und Verschlüsselung 61
4.4.1 Erstellen digitaler Signaturen 62
4.4.1.1 XML - Signatur 70
4.4.1.2 CMS - Signatur 75
4.4.1.3 S/MIME - Signatur 76
4.4.1.4 PDF - Signatur 76
4.4.1.5 PKCS#1 – Signatur (External Authenticate) 77
4.4.1.6 Nicht-qualifizierte elektronische Signatur 79
4.4.1.7 Qualifizierte elektronische Signatur 81
4.4.2 Verifizieren digitaler Signaturen 84
4.4.3 Zertifikatsdienst 88
4.4.3.1 Ablaufdatum von Zertifikaten prüfen 88
4.4.3.2 Kartenzertifikat lesen 89
4.4.3.3 Zertifikate verifizieren 91
4.4.4 Verschlüsselung 93
4.4.4.1 Verschlüsseln 93
4.4.4.2 Entschlüsseln 98

4.5 E-Mail-Kommunikation mittels KOM-LE 101
4.5.1 Übersicht 101
4.5.2 Schnittstellen 102
4.5.3 Abläufe im Primärsystem 103

4.5.3.1 Nachrichten generieren und übernehmen ... 104
4.5.3.2 Empfänger ermitteln ... 104
4.5.3.3 Nachrichten versenden ... 106
4.5.3.4 Nachrichten empfangen ... 108

5 Status und Logging ... 110
5.1 Erfolgreiche Verarbeitung VSDM ... 110
5.2 Statusinformationen ... 110
5.3 Meldungen/Logging ... 111

6 Fehlerbehandlung ... 112
6.1 Übersicht ... 112
6.2 Empfehlungen zur Fehlerbehandlung ... 112
6.3 SOAP-Fault ... 113
6.3.1 Sonderfall „VSD inkonsistent“ ... 114
6.3.2 Sonderfall „HBA/SM-B nicht freigeschaltet“ ... 114
6.3.3 Sonderfall „Prüfungsnachweis nicht entschlüsselbar“ ... 114
6.4 Warnungen ... 115
6.5 Sonderfall „Maximale Offline-Zeit der TI überschritten“ ... 117
6.6 Fehlercodes ... 118

7 Komfortfunktionen ... 124
7.1 Hintergrundverarbeitung bei Online-Prüfung ... 124
7.2 Auswertung von Karteninformationen (HBA/SM-B) ... 124

Anhang A - Verzeichnisse ... 125
A1 – Abkürzungen ... 125
A2 – Glossar ... 126
A3 – Abbildungsverzeichnis ... 126
A4 – Tabellenverzeichnis ... 127
A5 - Beispiele ... 129
A6 - Referenzierte Dokumente ... 130
A6.1 – Dokumente der gematik ... 130
A6.2 – Weitere Dokumente ... 130

Anhang B ... 135
B1 – Konfigurationsparameter ... 135
B1.1 – Konnektorkommunikation ... 135
B1.2 – Beziehungen zwischen den Konfigurationseinheiten ... 135
B2 - Abweichungen zur Schemaversion 5.1.0 ... 137
B2.1 – Beschreibung der Änderungen der Attribute in den Klassen ... 139
B2.2 – Beschreibung der Änderungen der Befüllungsvorschriften von Attributen ... 143
B2.3 – Verarbeitung von Datenfeldern durch das Primärsystem ... 143

1 Einordnung des Dokuments

1.1 Zielsetzung

Das Dokument beschreibt die für die Implementierung des Versichertenstammdatenmanagements und der Basisdienste QES, Signatur und Verschlüsselung in Primärsysteme erforderlichen Vorgaben.

Der Implementierungsleitfaden beschreibt darüber hinaus die praktische Anwendung folgender Konzepte und Spezifikationen:

- Systemspezifisches Konzept VSD [gemSysL_VSDM]
- Spezifikation Fachmodul VSDM [gemSpec_FM_VSDM]
- Spezifikation Schnittstelle Primärsystem [gemSpec_SST_PS_VSDM]
- Spezifikation Mobiles Kartenterminal [gemSpec_MobKT]
- Spezifikation Konnektor [gemSpec_Kon]

Die Kenntnis dieser Dokumente bzw. der entsprechend relevanten Teile wird als Arbeitsgrundlage für die Nutzung des vorliegenden Dokuments angenommen. Sie enthalten die normativen Vorgaben an die entsprechenden Komponenten.

1.2 Zielgruppe

Das Dokument richtet sich maßgeblich an Hersteller von Primärsystemen (Praxisverwaltungssysteme und Krankenhausinformationssysteme) von Leistungserbringern im Online-Rollout (Stufe 1).

1.3 Geltungsbereich

Die in diesem Dokument formulierten Anforderungen sind normativ für die an der Erprobung teilnehmenden Primärsysteme des Online-Rollouts (Stufe 1). Der Gültigkeitszeitraum der vorliegenden Version und deren Anwendung in Zulassungsverfahren wird durch die gematik GmbH in gesonderten Dokumenten (z. B. Dokumentenlandkarte, Produkttypsteckbrief, Leistungsbeschreibung) festgelegt und bekannt gegeben.

Alle Anforderungen zur Durchführung von Online-Prüfungen und -aktualisierungen sowie zur Übernahme von Prüfungsnachweisen gelten nur für Primärsysteme in Umgebungen vertragsärztlicher Leistungserbringer (nicht im stationären Bereich).

Die Anforderungen können im Anschluss an die Erprobung für Implementierungsleitfäden bzw. Konformitätsprofile der Sektoren verwendet werden.

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

Innerhalb dieses Dokuments wird auf die fachliche und technische Umsetzung des Online-Rollout (Stufe 1) in den Primärsystemen der Leistungserbringer eingegangen. Für nicht an der vertragsärztlichen Versorgung teilnehmende Leistungserbringer (z. B. Krankenhaus, Apotheke) sind die Anforderungen zur VSDM-Online-Prüfung und -aktualisierung sowie zum Prüfungsnachweis informativ.

Festlegungen für interne Geschäftsprozesse der Leistungserbringer sind nicht Bestandteil dieses Dokuments.

Weiterhin werden keine Festlegungen zur Zuordnung von HBA zu Primärsystem und Mandant getroffen, d.h. Identitätsmanagement sowie Rollen- und Rechteverwaltung liegen in der Hoheit des Primärsystems.

Die Aufrüstung von BCS-Kartenterminals auf den Standard eHealth-KT ist nicht Gegenstand dieses Dokuments. Der Zugriff auf BCS-Terminals vom Primärsystem aus ist ebenfalls nicht Bestandteil dieses Dokument. Entsprechende Beschreibungen finden sich im Leitfaden aus dem Basis-Rollout [gemLF_Impl_eGK] in der Version 1.4.

Die Außenschnittstelle des Konnektors wird durch [gemSpec_Kon] abschließend spezifiziert.

1.5 Methodik

Anforderungen als Ausdruck normativer Festlegungen werden durch eine eindeutige ID in eckigen Klammern 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:

⌦ **XXX-A_0000 <Titel der Afo>**

Text/Beschreibung⌫

Dabei umfasst die Anforderung sämtliche innerhalb der Textmarken angeführten Inhalte.

Die Darstellung der Anwendungsprozesse erfolgt prinzipiell auf der Grundlage der BPMN-Modellierung.

Die Darstellung der Versichertenstammdaten mittels Klassendiagramm erfolgt in UML.

`Listing, Bezeichner, Variablen` oder `XML-Elemente` werden in Courier dargestellt.

```
Beispiele werden hervorgehoben dargestellt. Bei der Auswertung der
(informativen) Beispiele ist zu beachten, dass die zugrundeliegenden
XML-Schemadateien und WSDLs versioniert sind und einem Releasemanagement
unterliegen. Eine Orientierung über die an der Konnektorschnittstelle zu
verwendenden Schemaversionen und Namensräumen bietet [gemSpec_Kon#AnhD].
```

In diesem Dokument werden die Begriffe Clientsystem und Primärsystem synonym verwendet. Der Begriff Clientsystem umfasst streng genommen zusätzlich Systeme in Geschäftsstellen der Kostenträger, welche aber nicht behandelt werden.

2 Systemüberblick

Auf der Grundlage der Spezifikationen der Fachanwendung VSDM und der Basis-TI beschreibt der Implementierungsleitfaden (ILF) die Nutzung von Komponenten und Schnittstellen der Telematikinfrastruktur durch Primärsysteme von Leistungserbringern im Rahmen des Online-Rollouts (Stufe 1) der elektronischen Gesundheitskarte. Die zentralen Funktionen im Online-Rollout (Stufe 1) sind die Fachanwendung des Versichertenstammdatenmanagements und der Basisdienste QES, Signatur und Verschlüsselung.

Das Primärsystem arbeitet als dezentrales System in der Umgebung des Leistungserbringers und kommuniziert über dezentrale Komponenten der TI (Konnektor) mit der Telematikinfrastruktur.

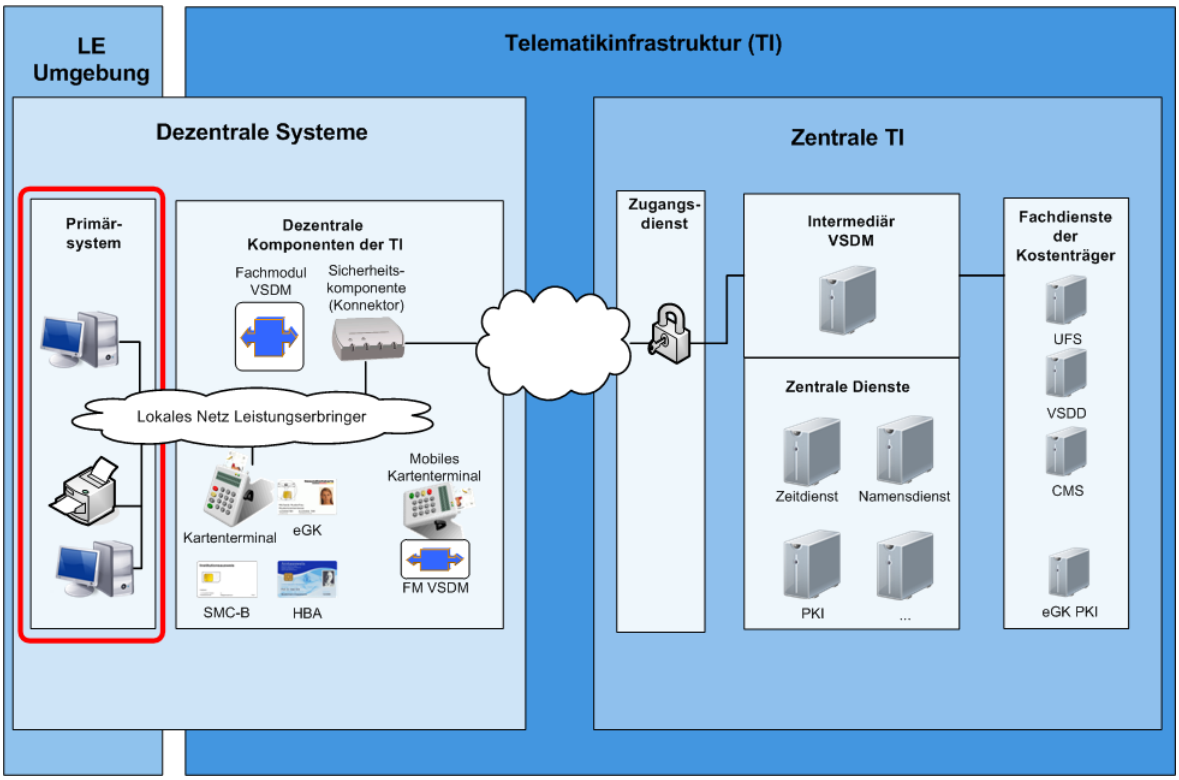


Abbildung 1: Primärsystem im Systemkontext

Mit Beginn des Online-Rollouts werden die Kartenterminals nicht mehr direkt durch das Primärsystem kontrolliert. Der Konnektor übernimmt die Kommunikation mit den Kartenterminals und den darin befindlichen Karten. Alle Sicherheitsleistungen werden vom Konnektor erbracht, so dass das Primärsystem nicht mehr direkt auf die Karten zugreift, sondern diese Aufgaben an den Konnektor delegiert.

Die Kommunikation zum Konnektor geschieht mittels SOAP an die vom Konnektor bereitgestellten Webservice-Schnittstellen. Ausnahmen hiervon bilden

- das Auslesen der verfügbaren Dienste am Dienstverzeichnisdienst des Konnektors (http),
- das Auslesen der Versichertenstammdaten aus mobilen Kartenterminals (CT-API),

- und das Übermitteln von Ereignissen vom Ereignisdienst des Konnektors an das Primärsystem (cetp).

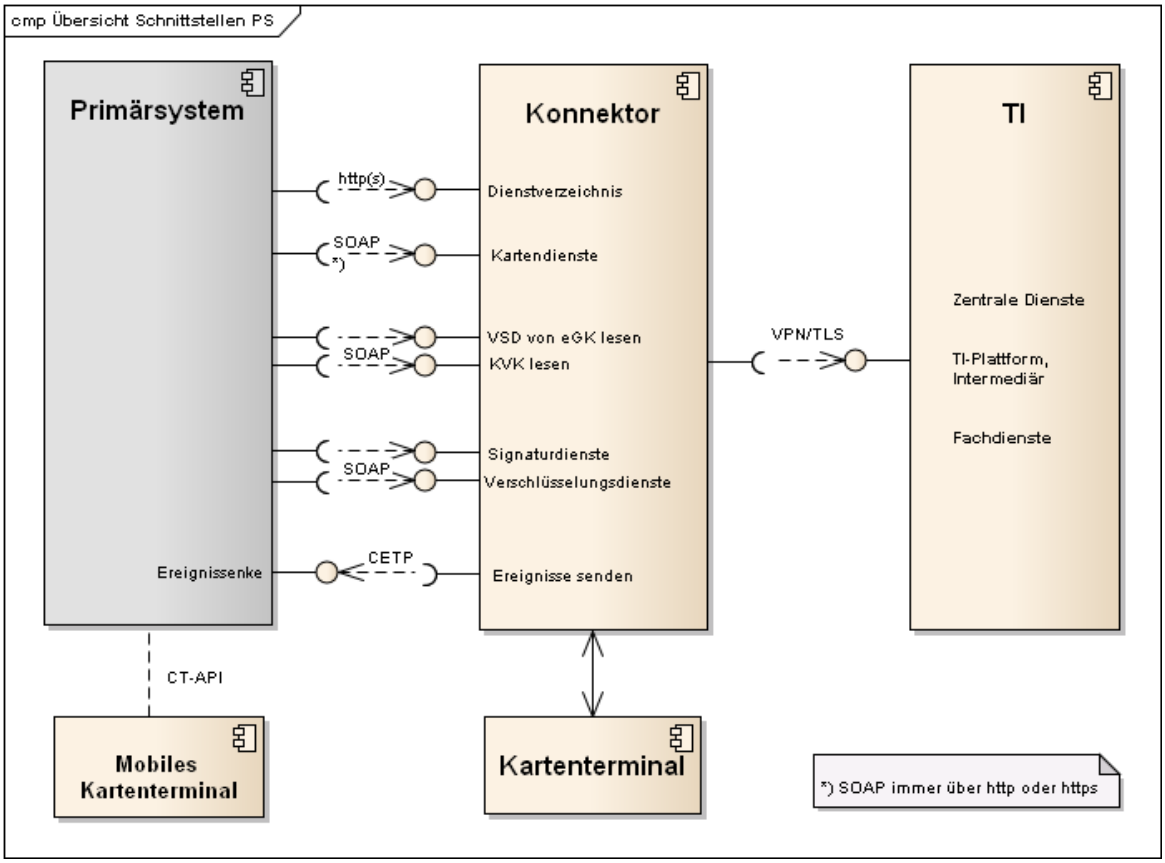


Abbildung 2: Komponenten und Schnittstellen am Primärsystem

Abbildung 2: Komponenten und Schnittstellen am Primärsystem stellt die Komponenten und Schnittstellen abstrakt dar und verwendet keine formalen Namen von Schnittstellen. Die Verbindung in die TI ist stark vereinfacht und dient nur der Übersicht.

Das mobile Kartenterminal (mobKT) wird über eine seitens des Primärsystems bereits existierende Schnittstelle angesprochen (CT-API), was in der entsprechenden Spezifikation normativ beschrieben ist [gemSpec_MobKT]. Gegenstand dieses Dokument sind die “neuen“ Schnittstellen des PS zum Konnektor. Die Schnittstelle zum mobilen Kartenterminal (mobKT) ist daher nicht Bestandteil dieses Dokuments und ist nur der Vollständigkeit halber dargestellt.

3 Konfiguration

3.1 Umgebung des Leistungserbringers

3.1.1 Begriffe der Konfigurationseinheiten

- Mandant (M): Ein Mandant ist innerhalb des Primärsystems eine eigenständige Organisationseinheit (z.B. ein Vertragsarzt). Der Datenhaushalt eines Mandanten ist in sich abgeschlossen. Werden innerhalb des Primärsystems mehrere Mandanten verwaltet, werden die Datenhaushalte voneinander abgegrenzt.
- Primärsystem (PS): Unter dem Begriff Primärsystem werden die Praxisverwaltungssysteme (PVS) in Arzt-/Zahnarztpraxen, die Krankenhausinformationssysteme (KIS) und die Apothekerverwaltungssysteme (AVS) zusammengefasst. Apothekenverwaltungssysteme werden im Online-Rollout (Stufe 1) nicht betrachtet.
- Arbeitsplatz (AP): Ein Arbeitsplatz ist eine fest installierte Einheit bestehend aus Bildschirm, Tastatur, Arbeitsplatzrechner und Kartenterminal und kann von mehreren Personen benutzt werden.
- Kartenterminal (KT): Mit der Einführung des Online-Rollouts (Stufe 1) kommt ein durch die gematik GmbH zugelassenes, netzwerkgestütztes eHealth-Kartenterminal zur Anwendung. Das Kartenterminal kann entweder am Online- oder am Offline-Konnektor angeschlossen sein.
- Online-Konnektor: Logische oder physische Einheit des Konnektors, die online mit der TI verbunden ist
- Offline-Konnektor: Logische oder physische Einheit des Konnektors, die keinen Online-Zugang zur TI besitzt.
- Der Extended Trusted Viewer (xTV) ist eine Software-Sicherheitskomponente, die auf bestimmten Arbeitsplätzen eingerichtet werden kann, wenn auf diesen Arbeitsplätzen Signatur- oder Verschlüsselungsfunktionen genutzt werden sollen. Die Konfigurationsinstanzen des xTV werden am Konnektor verwaltet und sind durch IDs am Primärsystem referenzierbar.
- Das mobile Kartenterminal (mobKT) ist ein durch die gematik GmbH zugelassenes, offline arbeitendes Kartenterminal für mobile Einsatzszenarien (z.B. Hausbesuch), welches zur Datenübernahme direkt an das Primärsystem angeschlossen und über Standardprotokolle von Kartenterminals (CT-API) angesprochen wird. Das mobKT wird nicht über den Konnektor verwaltet und nicht über dessen Schnittstellen angesprochen. Es ist nicht Bestandteil der Konnektorkonfiguration

3.1.2 Beziehungen der Konfigurationseinheiten

Im folgenden Diagramm und den nachfolgenden Tabellen werden die möglichen Konfigurationen in medizinischen Einrichtungen dargestellt.

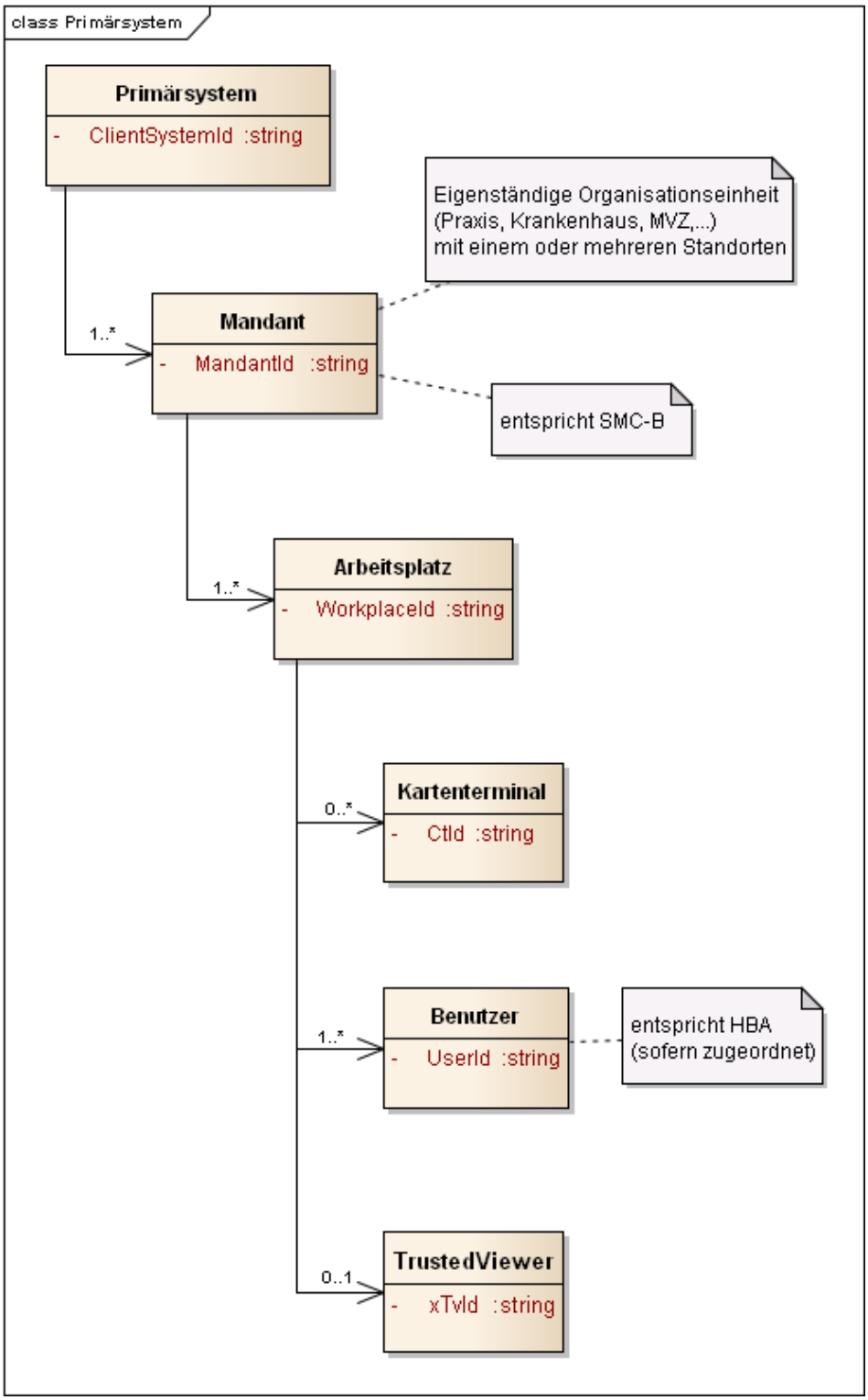


Abbildung 3: Konfigurationseinheiten

Eine tabellarische Aufstellung der Beziehungen zwischen den Konfigurationseinheiten befindet sich im Anhang B1.2.

Für die Zuordnung zwischen Karten und Akteuren gelten folgenden Annahmen/Festlegungen

- Eine SMC-B ist immer einem Mandanten zugeordnet, entspricht also in der Regel einer eigenständigen Organisationseinheit.
- Ein HBA ist immer einem Heilberufler (z. B. Arzt) zugeordnet, entspricht also genau einer natürlichen Person.
- Es gibt keine feste Zuordnung von HBA zu Mandant. Ein Arzt kann im konkreten Umfeld einer Leistungserbringerorganisation mehreren Mandanten (Organisationen) zugeordnet sein.

Wenn ein HSM-B anstelle einer SMC-B zum Einsatz kommt, verhält sich dieses aus Sicht des Primärsystems funktional wie eine SMC-B. Der Konnektor kapselt die funktionale Verwendung des HSM-B. Daher wird im Folgenden immer nur die SM-B angesprochen.

Außenstellen einer Praxis werden in diesem Dokument nicht gesondert betrachtet, da davon ausgegangen wird, dass die Außenstellen Bestandteile der Praxis sind (zusätzlicher Arbeitsplatz mit KT und z. B. VPN-Verbindung).

3.2 Arbeitsplätze in der Leistungserbringerumgebung

Um in der Umgebung des Leistungserbringers die Online-Prüfung und -Aktualisierung durchzuführen, können grundsätzlich drei verschiedene Szenarien verwendet werden, die sich in der Konfiguration der Arbeitsplätze widerspiegeln.

- Online-Szenario am Arbeitsplatz eines Primärsystems mit TI-Anbindung (3.2.1) oder im
- Standalone-Szenario mit Arbeitsplatz/Kartenterminal am Online-Konnektor und Lesen der VSD am Offline-Konnektor (physische Trennung, 3.2.2)
- Szenario mit logischer Trennung (keine Internetverbindung und keine Dienste am Konnektor verfügbar außer VSDM, 3.2.3)

Leistungserbringer, die ihr Primärsystem bzw. das lokale Netz nicht direkt über den Konnektor an die TI oder an das Internet anbinden wollen, können das Standalone-Szenario nutzen (siehe 3.2.2).

Nachfolgend werden die verschiedenen Szenarien dargestellt, wobei die Dienste nur schematisch und nicht streng zugeordnet zur TI dargestellt sind (beim Sicherheitsgateway KV-SafeNet ist nur der Zugangspunkt Teil der TI).

3.2.1 Online-Szenario

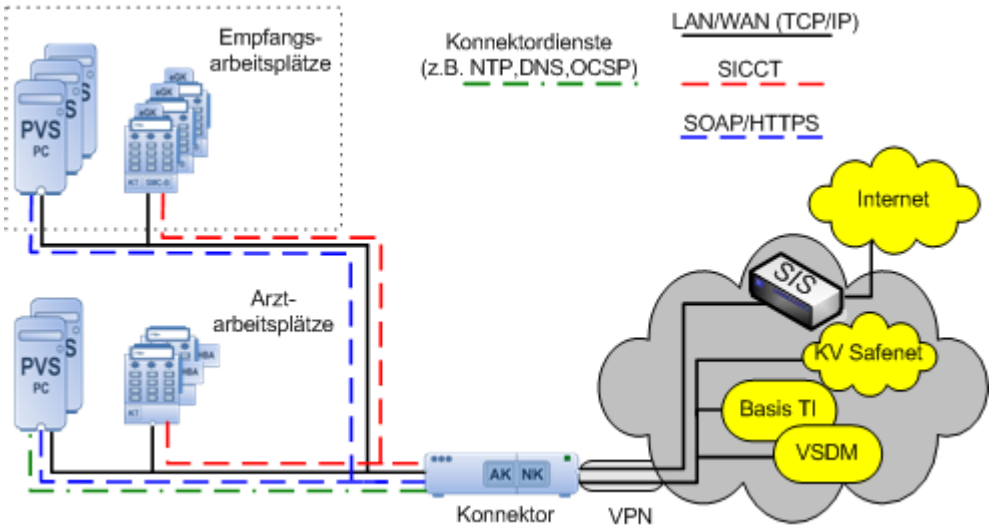


Abbildung 4: Online-Szenario

Im Online-Szenario gemäß Abbildung 4 ist der Konnektor sowohl mit dem Praxisnetz als auch mit der TI inklusive KV-SafeNet sowie dem Secure Internet Service (SIS) verbunden (je nach Konfiguration). Alle Dienste stehen über sichere Verbindungen dem Clientsystem zur Verfügung. In der Minimalausprägung kommt nur ein Terminal am Empfang zum Einsatz, wobei der Arztarbeitsplatz ohne KT arbeiten kann, sofern entsprechende Funktionen nicht genutzt werden sollen (z.B. QES).

3.2.2 Standalone-Szenario mit Online-Konnektor und Offline-Konnektor

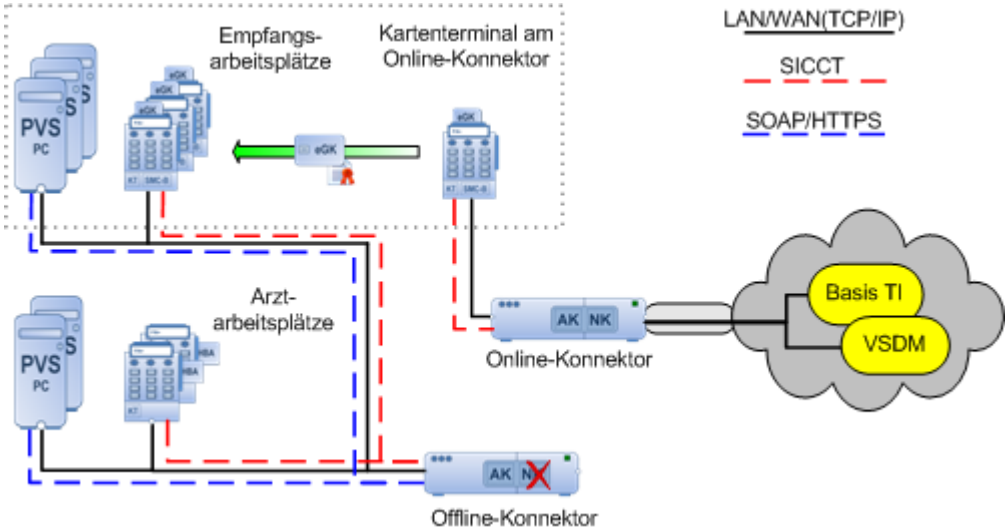


Abbildung 5: Standalone-Szenario mit physischer Trennung

Im Standalone-Szenario besteht keine Netzanbindung des Primärsystems an die Telematikinfrastruktur (TI). Es kommen ein zusätzlicher Konnektor und ein zusätzliches Kartenterminal zum Einsatz. Das Praxisnetz ist nicht mit dem Online-Konnektor resp. dem Internet oder KV-SafeNet verbunden. Um die Online-Prüfung und -Aktualisierung der eGK durchzuführen, wird die eGK in das Kartenterminal am Online-Konnektor gesteckt. Die Online-Prüfung und -Aktualisierung wird daraufhin automatisch gestartet. Während der

Durchführung werden dem Benutzer auf dem Display Hinweise zum Status und/oder Fehlermeldungen angezeigt (z.B. eGK gesperrt). Nach der Online-Prüfung und -Aktualisierung wird die eGK in ein am Offline-Konnektor angeschlossenes Kartenterminal gesteckt, welches standardmäßig einem Arbeitsplatz des Primärsystems zugeordnet ist, und die VSD inkl. Prüfungsnachweis werden übernommen. Der Ablauf erfolgt analog des in 4.3.4.2 beschriebenen Ablaufs.

Am Online-Konnektor ist der Betrieb eines „Kommunikations-PC“ (einzelner, nicht mit dem Praxisnetz verbundener PC) möglich, an dem – je nach Konnektorkonfiguration – alle Online-Funktionen genutzt werden können.

3.2.3 Szenario mit logischer Trennung

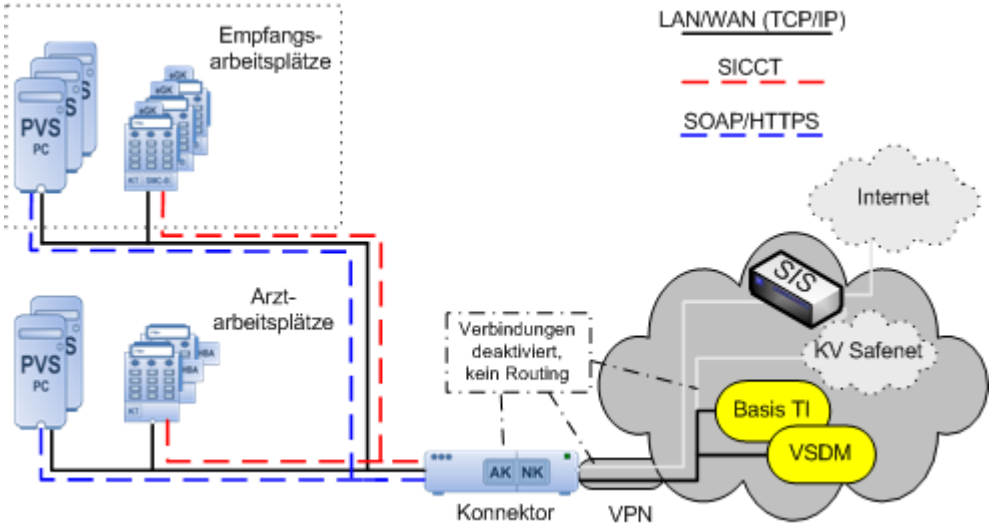


Abbildung 6: Szenario mit logischer Trennung im Konnektor

In diesem Szenario kommt nur ein Konnektor zum Einsatz, dessen Online-Funktionen eingeschränkt sind. Verbindungen in die TI sind nur durch den Konnektor intern oder durch Fachmodule möglich. Die Verbindung in das sichere Internet (SIS) oder andere Dienste außerhalb der TI sind über eine entsprechende Konnektorkonfiguration deaktiviert. Es stehen keinerlei Internet- oder Basis-TI-Dienste für das Clientsystem zur Verfügung. Einzig verfügbare Online-Operation für das Primärsystem ist `ReadVSD`. Details zur Konfiguration und zum Verhalten des Konnektors in diesem Szenario finden sich in [gemSpec_Kon#4.3.6].

Dieses Szenario entspricht für die Fachanwendung VSDM funktional dem Online-Szenario und wird daher für VSDM nicht separat betrachtet. Einziger Unterschied ist die netzwerktechnische Trennung im Konnektor zwischen Praxis und Internet.

3.3 Arbeitsplätze, Mandanten und Kartenterminals konfigurieren

Der Konnektor hat keine eigene Benutzerverwaltung und vertraut der Benutzerverwaltung (Konfigurationsverwaltung) des Primärsystems.[1]

[1] Vgl. [gemKPT_Arch_TIP#4.2]

In der Konfiguration des Primärsystems wird die Zuordnung zwischen Mandanten, Karten, Arbeitsplätzen und Kartenterminals verwaltet sowie die eindeutige Zuordnung zwischen Heilberuflern und ihren UserIDs.

Die Konfigurationsverwaltung des Primärsystems ermöglicht es einem Konnektor-Administrator, diese Parameter so in der Konnektorkonfiguration zu verwenden, dass sie der Konfiguration im Primärsystem entsprechen.

3.3.1 Aufrufkontext

Der Konnektor benötigt von seinen Clientsystemen die Angabe des Kontextes, aus dem heraus die Aufrufe erfolgen, um Aufrufberechtigungen überprüfen zu können. Im Aufrufkontext von Funktionsaufrufen sind Angaben zu Mandant, Arbeitsplatz und Primärsystem verpflichtend, Identifikation des Benutzers ist optional (für bestimmte Aufrufe notwendig).

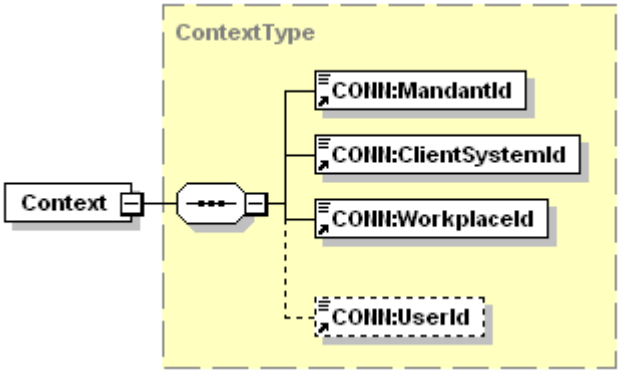


Abbildung 7: Abb_LFPS_01_Element Context gemäß ConnectorContext.xsd

⌦ **TIP1-A_4959 Konfigurierbarkeit von Kontext-Parametern**

Innerhalb des Primärsystems MUSS eine Konfigurationsverwaltung vorhanden sein, welche die Parameter `MandantId`, `ClientSystemId`, `WorkplaceId` und `UserId` entsprechend „Abb_LFPS_01_Element Context gemäß ConnectorContext.xsd" abbildet. Die Parameter sind alphanumerisch und haben eine Maximallänge von 64 Zeichen. ⌫

Die Parameter `MandantId`, `ClientSystemId` und `WorkplaceId` bilden das Datenelement `Context`, gemeinsam mit der optionalen und nur für den Zugriff auf den HBA in einigen Aufrufkontexten erforderlichen `UserId`.

Mandantenfähige Primärsysteme sollen Identifikatoren als `MandantId` verwenden, die ihrer internen Mandantenverwaltung entsprechen, falls vorhanden. Jedem Mandant muss eine SM-B zugeordnet werden. Nicht mandantenfähige Primärsysteme oder solche, in denen immer nur ein Mandant vorhanden ist, müssen die `MandantId` durchgängig auf einen festgelegten Wert setzen, welcher dem Wert in der Konnektorkonfiguration entspricht.

Das Primärsystem einer LE-Umgebung muss einen Identifikator besitzen, der für Konnektoraufrufe als Primärsystem-Identifier (`ClientSystemId`) genutzt werden kann.

Jeder Arbeitsplatz innerhalb einer LE-Umgebung muss einen lokal eindeutigen Identifikator besitzen, der als `WorkplaceId` genutzt werden kann. Erfolgen Aufrufe des Primärsystems nicht direkt vom Arbeitsplatzsystem (im Sinne eines Rich Clients), sondern werden über eine Server-Komponente des Primärsystems geleitet (Thin Client, z.B. Web-

Applikationen) muss der Server trotzdem eine Arbeitsplatz-ID des Aufrufers an den Konnektor übermitteln.

Die `UserId` ist eine eindeutige vom Primärsystem vergebene interne ID, die nur bei Zugriffen auf einen HBA erforderlich ist. Sie wird temporär im Konnektor gespeichert und einem HBA zugeordnet, wenn eine HBA-Kartensitzung in einen erhöhten Sicherheitszustand versetzt wird (PIN-Eingabe). Sie bleibt gespeichert und zugeordnet, solange die Kartensitzung gültig ist (i. d. R. solange der HBA gesteckt bleibt). Bei Zugriffen auf den HBA im weiteren Verlauf muss die bei der Eröffnung verwendete `UserId` im Kontext korrekt angegeben sein (z. B. Signatur oder Entschlüsselung). Das PS kann als `UserID` eine persistente interne Referenz eines Benutzers oder eine temporär generierte ID verwenden. Es muss sicherstellen, dass sie eindeutig ist und nicht mehrfach für verschiedene Benutzer verwendet wird. Ein Login-Name oder ein Klartextname sollten nicht verwendet werden.

⌦ **TIP1-A_4960 Nutzung von Kontextparametern**

Alle Arbeitsplätze eines Primärsystems, von denen aus der Konnektor genutzt wird, MÜSSEN den Konnektor mit einem für sie individuell eindeutigen Kontext aufrufen und dazu administrierbare Kontextinformationen verwenden. ⌫

3.3.2 LE-Umgebungen

⌦ **TIP1-A_4961 Zuordnung von Kartenzugriffen zu Arbeitsplätzen**

Wenn mehrere Kartenterminals und Karten in der Netzwerkumgebung des Primärsystems vorliegen, MÜSSEN Kartenterminals und Karten für Zugriffe durch einzelne Clientsystem-Arbeitsplätze selektiert werden. ⌫

Mehrere Selektionsstrategien sind möglich:

- Setzen von selektierenden Parametern in den Funktionsaufrufen von `GetCards` und `GetCardTerminals` aufgrund von konfigurativen Zuordnungen zwischen Arbeitsplatz und Kartenterminal
- Nutzung des Ereignisdienstes durch zielgerichtetes Abonnieren von Kartensteckereignissen (s. 4.1.4)
- Dialogsteuerung zur Auswahl unter verfügbaren Karten. Ein Auswahldialog kann notwendig sein, wenn an einem Arbeitsplatz mehrere Karten verfügbar sind, mit denen gleichartige Aktionen möglich sind. Ein Beispiel wäre die Auswahl unter mehreren am selben Arbeitsplatz verfügbaren SM-B oder HBAx ihm Rahmen des Signierens von Dokumenten. Auswahldialoge sollen vermieden werden, wenn sie nicht durch Anwendungsfälle motiviert sind.

4 Funktionsmerkmale

4.1 Inbetriebnahme

Primärsystem und Konnektor sind gemeinsam betriebsbereit, wenn

- die Konfiguration des Gesamtsystems (inklusive mindestens 1 Kartenterminal) erfolgt ist und die Konfiguration von Primärsystem und Konnektor an einander angeglichen sind,
- zwischen beiden Systemen eine Verbindung (HTTP oder HTTPS) besteht,
- das Primärsystem aktuelle Informationen über verfügbare Dienste hat,
- Ereignisse über den Ereignisdienst des Konnektors abonniert sind (sofern vorgesehen) und
- mindestens eine freigeschaltete SM-B verfügbar ist.

Um den Leistungsumfang des Online-Rollouts (Stufe 1) nutzen zu können, muss vom Primärsystem eine freigeschaltete SM-B verwendet werden. Dabei muss die Person, die den Konnektor in Betrieb nimmt, die PIN der SM-B eingeben und ggf. initialisieren.

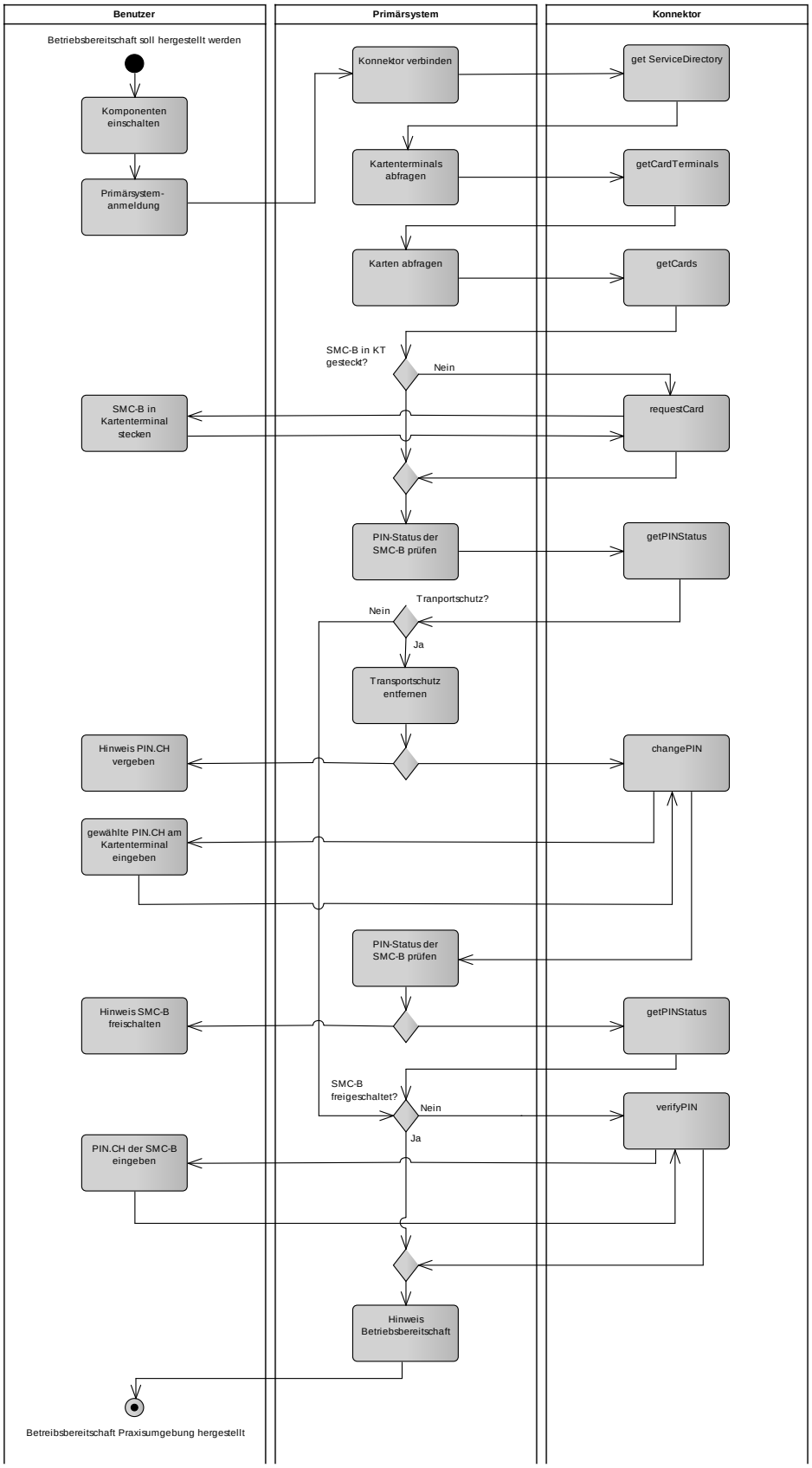


Abbildung 8: Betriebbereitschaft herstellen

4.1.1 Verbindungsaufbau zum Konnektor

Die Kommunikation zwischen Primärsystem und Konnektor basiert auf HTTP. Der Konnektor bietet drei Varianten der HTTP(S)-Verbindung an (vgl [gemSpec_Kon#3.4]):

1. TLS deaktiviert. Verwendung von HTTP ohne Absicherung auf Transportebene (nicht empfohlen)
2. TLS ohne Client-Authentisierung. Der Konnektor nimmt jede Clientsystem-Anfrage an.
3. HTTP mit Basic Authentication, kombiniert mit TLS ohne Client-Authentisierung. Das Primärsystem muss Username und Passwort für die Basic Authentication statisch konfigurieren, so dass eine Übereinstimmung mit der Konfiguration am Konnektor besteht.
4. TLS mit Client Authentication. Die Client Authentication muss mit den Zertifikaten erfolgen, die am Konnektor erzeugt wurden und vom Administrator in das Primärsystem importiert wurden oder mit konnektorfremden X.509-Zertifikaten der Clientsysteme, die über das Managementinterface in den Konnektor eingespielt wurden.

Der Einsatz von TLS auf der Kommunikationsstrecke zwischen Primärsystem und Konnektor ist Gegenstand der Erprobung in den Testregionen. Welche Variante der HTTP(S)-Verbindungen vom Konnektor faktisch angeboten wird, hängt von einer Reihe von Faktoren und Konfigurationen am Konnektor ab.[2]

⌦ **TIP1-A_4962 Nutzung von HTTP(S)-Authentisierungsmethoden**

Im Rahmen der Erprobung SOLL das Primärsystem HTTP Basic Authentication kombiniert mit TLS ohne Client-Authentisierung (2.) oder (4.) TLS mit zertifikatsbasierter Client Authentication gemäß TLS Version 1.1 verwenden. Das PS KANN auch TLS Version 1.2. unterstützen.⌫

Wenn der Konnektor so konfiguriert wird, dass TLS nicht erzwungen wird, bietet der Konnektor ggf. einen HTTP-Port an, sowie einen HTTPS-Port. Das Primärsystem kann den Konnektor in diesem Fall unter beiden Ports erreichen.

In seinem Dienstverzeichnisdienst stellt der Konnektor unter einer definierten URL in einem XML-Dokument („connector.sds“) die Liste aller Dienste, sowie deren Versionen und Endpunkte bereit, die vom Konnektor angeboten werden.

Es ist am Konnektor möglich, die Transportsicherung zum Dienstverzeichnisdienst des Konnektors anders zu konfigurieren als die Transportsicherung zu den restlichen Diensten.

⌦ **TIP1-A_4963 Authentifizierung gegenüber Dienstverzeichnisdienst**

Das Primärsystem SOLL in der Lage sein, den Service-Endpunkt des Konnektordienstverzeichnisdienstes mit einer Transportsicherungsmethode (TLS deaktiviert, HTTPS Basic Authentication oder HTTPS mit Client Authentication) anzusprechen,

[2] In [gemSpec_Kon#AnhK] werden eine Reihe dieser Faktoren im Kontext der Netzwerktopologie der Leistungserbringerumgebung geschildert, etwa die Unterscheidung zwischen einem Internet Access Gateway, das hinter einem Konnektor in Reihe geschaltet wird, und einem parallel zum Konnektor angebundenem Internet Access Gateway.

die sich ggf. von der Transportsicherungsmethode der weiteren Dienste unterscheidet. ⌫

4.1.1.1 Client Authentisierung

Wie in 4.1.1 beschrieben soll das Primärsystem mindestens eine von drei verfügbaren Authentisierungsmethoden beim Verbindungsaufbau zum Konnektor unterstützen.

a.) Für die Basic Authentication (auch “Basic Access Authentication”, ein Standard der HTTP-Authentifizierung) soll dabei das Primärsystem die notwendigen Parameter „Benutzername“ und „Passwort“ verwalten. Das Primärsystem muss über zwei entsprechende Konfigurationsparameter verfügen, die sich über die Systemkonfiguration des PS eingeben bzw. verändern lassen. Wird als Authentisierungsmethode Basic Authentication vereinbart, müssen hier die gleichen Werte für Benutzername und Passwort eingegeben sein, wie in der Managementschnittstelle des Konnektors.

Zwei weitere Alternativen können dazu genutzt werden, den TLS-Kanal zwischen Konnektor und Clientsystem durch X.509 - Clientauthentisierung abzusichern:

b.) Für die zertifikatsbasierte Client Authentication (mittels konnektoreigenen Zertifikaten) wird im Konnektor ein Zertifikat sowie ein privater Schlüssel erzeugt und exportiert. Es liegt als standardisiertes Format (p12) [PKCS#12] vor, wobei der Schlüsselspeicher durch eine PIN geschützt ist.

Am Konnektor-Managementinterface erzeugte und von dort exportierte[3] Clientzertifikate werden in die Clientsysteme importiert. Das PS importiert und verwaltet das Client-Zertifikat aus der p12-Datei. Dazu muss während des Import-Vorgangs die PIN des Zertifikats eingegeben werden (Transportsicherung). Anschließend hat das Primärsystem Zugriff auf den für den TLS-Verbindungsaufbau benötigten privaten Schlüssel.

c.) Für die zertifikatsbasierte Client Authentication (mittels konnektorfremden Zertifikaten) werden konnektorfremde X.509-Zertifikaten der Clientsysteme über das Managementinterface in den Konnektor eingespielt.

Das Primärsystem nutzt einen Systemschlüsselspeicher, z.B. den Zertifikatsspeicher von Windows oder den des Java JRE. Auch hier ist für den Import-Vorgang ein Passwort des Schlüsselspeichers einzugeben. Anschließend stehen das Zertifikat und der Schlüssel über entsprechende Systemfunktionen/Bibliotheken zur Verfügung. Idealerweise kann der Administrator des PS in diesem Zertifikatsspeicher „browsen“ und das gewünschte Zertifikat für die Verwendung auswählen. Alternativ kann in der PS-Konfiguration eine eindeutige Referenz des Zertifikats (Name oder Index) eingegeben werden.

4.1.2 Konnektordienstverzeichnis lesen

Aus der Konnektordokumentation des Herstellers ist der FQDN zu entnehmen, unter dem der Konnektor sein Dienstverzeichnis anbietet. Innerhalb des FQDN können Hostname und Domain-Name je nach Konfiguration der LE-Umgebung individuell konfiguriert sein. In diesem Falle muss der FQDN entsprechend in der Primärsystemkonfiguration angepasst werden.

---

[3] [gemSpec_Kon]#3.4, TIP1-A_4517 Schlüssel und X.509-Zertifikate für Client-Authentisierung erzeugen und exportieren sowie X.509-Zertifikate importieren

Beispiel 1: URL des Konnektordienstverzeichnisses

```
http://KON_HOSTNAME/connector.sds
```

Dieser Parameter muss in der Primärsystemkonfiguration erfasst werden.

Durch das Auslesen des Dienstverzeichnisdienstes erhält das Primärsystem Webservice-Endpunkte von versionierten Diensten des Konnektors.

⌦ **TIP1-A_4967 Cachen von Service-Endpunkten**

Das Primärsystem MUSS die Endpunkte der Services, die der Konnektor anbietet, aus dem Dienstverzeichnisdienst initial unter einem FQDN ermitteln, der im Primärsystem konfiguriert ist, und die Endpunktinformationen der Dienste lokal cachen. Wenn ein Verbindungsproblem auftritt (Dienst nicht erreichbar), muss das Primärsystem einen Refresh auf alle Endpunktinformationen des Dienstverzeichnisdienstes durchführen.⌫

⌦ **TIP1-A_4968 Fehlermeldung zu nicht unterstützbaren Dienstversionen bei der Inbetriebnahme des Konnektors**

Zum Aufbau eines lokalen Dienstverzeichnis-Cache MUSS das Primärsystem das Dienstverzeichnis des Konnektors mittels `http(s)` vom Konnektor unter der konfigurierten URL auslesen. Werden die benötigten Dienste nicht in den Versionen gefunden, die das Primärsystem erwartet, muss dies mit einer aussagekräftigen Fehlermeldung dem Benutzer bei der Anmeldung angezeigt werden. ⌫

Beispiel 2: Dienstkonfiguration

```
<?xml version="1.0" encoding="UTF-8" ?>
-<CONN:ConnectorServices
xsi:schemaLocation="http://ws.gematik.de/conn/ServiceDirectory/v3.0
../conn/ServiceDirectory.xsd"
xmlns:VERS="http://ws.gematik.de/int/version/ProductInformation/v1.0"
xmlns:CONN="http://ws.gematik.de/conn/ServiceDirectory/v3.0"
xmlns:SI="http://ws.gematik.de/conn/ServiceInformation/v2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 + <PI:ProductInformation>
  <CONN:TLSMandatory>true</CONN:TLSMandatory>
  <CONN: ClientAutMandatory>true</CONN:ClientAutMandatory>
 - <SI:ServiceInformation>
  - <SI:Service Name="ReadVSD">
      <SI:Abstract>VSD von eGK lesen</SI:Abstract>
      <SI:Versions>
        <SI:Version TargetNamespace="http://ws.gematik.de/conn/vsds/
                                     VSDService/v6.0" Version="6.0">
            <SI:Abstract>VSD von eGK lesen Version 6.0</SI:Abstract>
            <SI:Endpoint Location="https://KON_HOSTNAME/services/readVSD"/>
            <SI:WSDL Location="https://KON_HOSTNAME/services/
                        wsdl/VSDService.wsdl"/>
        </SI:Version>
      </SI:Versions>
  + <SI:Service Name="ReadKVK">
  + <SI:Service Name="EventService">
  + <SI:Service Name="CardService">
  + <SI:Service Name="SignatureService">
 </SI:ServiceInformation>
</CONN:ConnectorServices>
```

Das Listing zeigt eine beispielhafte Dienstkonfiguration, wobei nur für den ersten Dienst die oberste Ebene dargestellt (aufgeklappt) ist. Für den Dienst `ReadVSD` sind neben einer Kurzbeschreibung eine versionsabhängige Beschreibung und die Endpunkte für die

Schnittstellenbeschreibung (WSDL) und die Kommunikation zu entnehmen. Je nach Sicherheitskonfiguration des Konnektors kann dabei ein Protokoll für verschlüsselte (https) oder unverschlüsselte Kommunikation vorgegeben werden. Ebenso kann der Port von den http-/https-Standardports abweichen.

Die vollständigen Schemadefinitionen des XML-Dokuments „connector.sds“ finden sich gemäß [gemSpec_Kon#4.1.3.1] in den Dateien `ServiceDirectory.xsd`, `ProductInformation.xsd` und `ServiceInformation.xsd`.

Da nicht davon ausgegangen werden kann, dass die Inhalte des Dienstverzeichnisdienstes statisch sind, sollte das Lesen des Verzeichnisses beim Programmstart, in Fehlersituationen (Verbindungsprobleme, Dienst nicht erreichbar) und nach Bootup des Konnektors erfolgen, um den Dienstverzeichnis-Cache zu erneuern. Die weitere Kommunikation mit den Diensten des Konnektors erfolgt dann über die im Dienstverzeichnisdienst propagierten Dienstendpunkte.

4.1.3 Nutzung von Webservice-Schnittstellen

⌦ TIP1-A_4964 Nutzung von SOAP

Das Primärsystem MUSS die Schnittstellen des Konnektors über eine Webservice-Schnittstelle auf Basis von SOAP nutzen ([WSDL1.1] und [BasicProfile1.2]). Das Primärsystem MUSS ausschließlich das Character Encoding UTF-8 verwenden. ⌫

Das Primärsystem MUSS 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. Falls in der SOAP-Nachricht base64-encodierte (verschlüsselte) XML-Elemente vorhanden sind, können diese XML-Elemente andere Zeichenkodierungen als UTF-8 aufweisen.

⌦ TIP1-A_4965 Nutzung des Dienstverzeichnisdienstes des Konnektors

Zu den Diensten, die der Konnektor laut Dienstverzeichnisdienst anbietet, MUSS das Primärsystem die Operationen und Parameter des Dienstes verwenden, wie sie in den zugehörigen Schemadateien (WSDLs, XSDs sowie den Schnittstellenbeschreibungen der Konnektorspezifikation) festgelegt sind. ⌫

Die Dienste des Konnektors sind versioniert. Es ist möglich, dass ein Konnektor mehrere Versionen eines Dienstes gleichzeitig anbietet. Die Versionierung der Dienste hilft dem Primärsystem dabei, genau die Dienstversionen zu nutzen, die es client-seitig implementiert hat.

⌦ TIP1-A_4966 Fähigkeit, unter Dienstversionen auszuwählen

Das Primärsystem MUSS in der Lage sein, die von ihm unterstützte Dienstversion unter mehreren vom Konnektor angebotenen Dienstschnittstellen auszuwählen. ⌫

Gemäß den Schnittstellenvorgaben erfolgt die SOAP-Kommunikation über http oder https.

Beispiel 3¸HTTP-SOAP-Header

```
POST /services/readVSD HTTP/1.1
Host: KON_HOSTNAME
Content-Type: text/xml; charset="utf-8"
Content-Length: XXXX
SOAPAction: "Some-URI"

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/
...
```

4.1.4 Ereignisdienst/Systeminformationsdienst

Das Primärsystem kann den Ereignisdienst als Basisanwendung des Systeminformationsdienstes (`EventService`) des Konnektors nutzen, um über konnektorspezifische Ereignisse zeitnah in einem Push-Mechanismus informiert zu werden. Die dabei an das Primärsystem zurückgegebenen Informationen können vom Primärsystem zu folgenden Zwecken genutzt werden:

- Anzeige von Statusinformationen zu TI-Komponenten, z.B. Verbindungsstatus des Konnektors
- Verwaltung von Informationen zu gesteckten Karten
- Kontrolle der Kartenverfügbarkeit
- Einlesen von Karten zum Zeitpunkt des Steckens der Karte in das Arbeitsplatzterminal
- Ablaufoptimierung und Performance-Verbesserung durch Push-Kommunikation

Neben den eigentlichen Operationen für das Verarbeiten von Ereignissen (siehe 4.1.4.1) stellt der `EventService` auch Operationen zum Zugriff auf Ressourcen und Abfragen verfügbarer Karten und Kartenterminals bereit (siehe 4.2.1). Details finden sich in den WSDL- und XSD-Dateien zur entsprechenden Service-Schnittstelle `EventService.wsdl` und `EventService.xsd`.

4.1.4.1 Ereignismeldungen mittels Protokoll CETP

Der Ereignisdienst des Systeminformationsdienstes nutzt das leichtgewichtige proprietäre Protokoll CETP (Connector Event Transport Protocol), das eine Abonnierung bestimmter Ereignistypen (Topics) durch das Primärsystem erfordert, siehe [gemSpec_Kon#4.1.6].

⌦ **TIP1-A_4969 Nutzung des Ereignisdienstes nach Vorgabe von [gemSpec_Kon]**

Die Nutzung des Ereignisdienstes durch das Primärsystem MUSS nach Vorgaben von [gemSpec_Kon#4.1.6] und den dort referenzierten Schemadateien erfolgen. ⌫

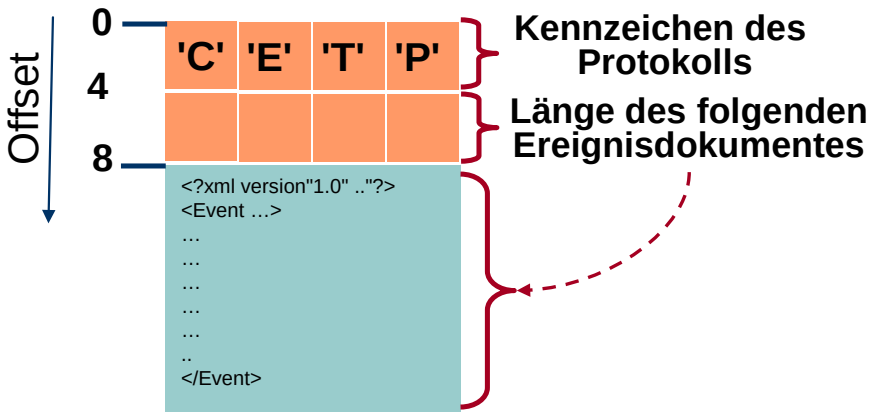


Abbildung 9: PIC_KON_022 Grundsätzlicher Aufbau der Ereignisnachricht

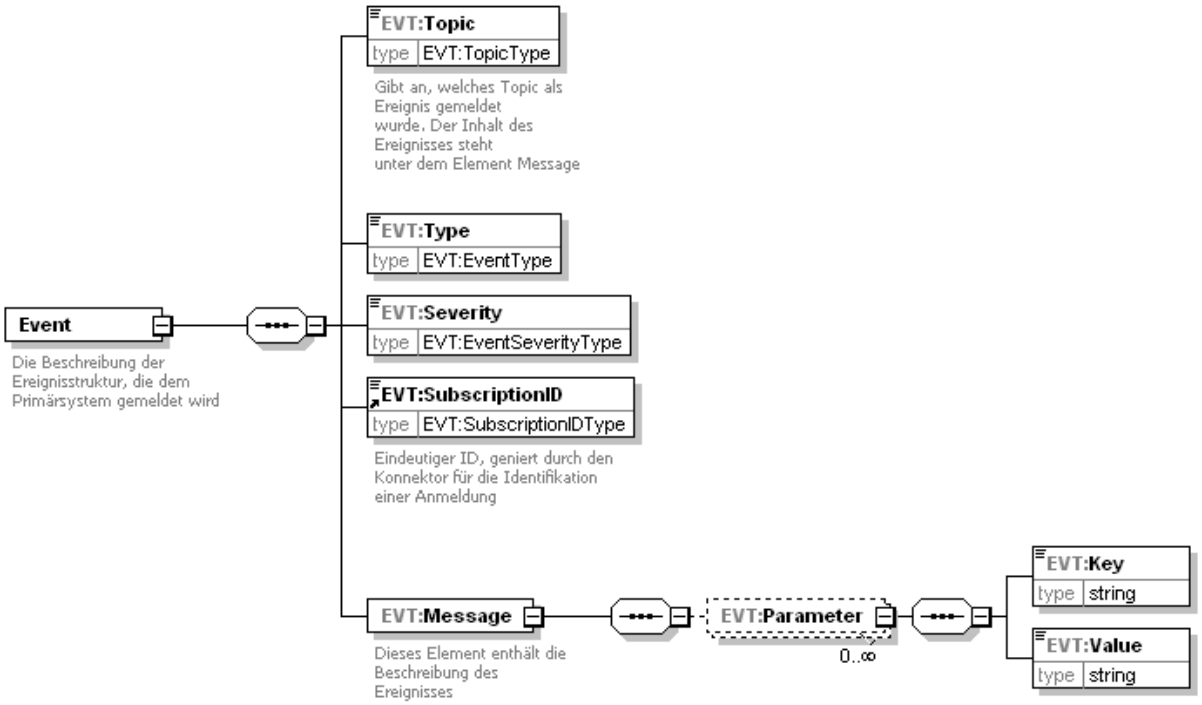


Abbildung 10: XML-Element Event

Beispiel 4: Vollständigen Ereignisstruktur einer CETP-Event-Nachricht

```
<?xml version="1.0" encoding="UTF-8"?>
<EVT:Event
 xsi:schemaLocation="http://ws.gematik.de/conn/EventService/v7.0
 ../conn/EventService.xsd"
 xmlns:EVT="http://ws.gematik.de/conn/EventService/v7.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <EVT:Topic>Card/Inserted</EVT:Topic>
  <EVT:Type>Operation</EVT:Type>
  <EVT:Severity>Info</EVT:Severity>
  <EVT:SubscriptionID>subwpid007.01</EVT:SubscriptionID>
  <EVT:Message>
    <EVT:Parameter>
      <EVT:Key>CardHandle</EVT:Key>
      <EVT:Value>c123456789123456789</EVT:Value>
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>CardType</EVT:Key>
      <EVT:Value>EGK</EVT:Value>
      <!--z.B. EGK|HBA-qSIG|HBA|SMC-B|HSM-B|SMC-KT|KVK|ZOD_2.0|UNKNOWN-->
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>CardVersion</EVT:Key>
      <EVT:Value>2.2.2_2.2.1</EVT:Value>
      <!--Version bei eGK,HBAx,SMC-KT,SM-B aus gemProdT_eGK]-->
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>ICCSN</EVT:Key>
      <EVT:Value>8027612345123456781</EVT:Value>
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>CtID</EVT:Key>
      <EVT:Value>101</EVT:Value>
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>SlotID</EVT:Key>
      <EVT:Value>101</EVT:Value>
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>InsertTime</EVT:Key>
      <EVT:Value>20121201103117</EVT:Value>
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>CardHolderName</EVT:Key>
      <EVT:Value>Muster</EVT:Value>
    </EVT:Parameter>
    <EVT:Parameter>
      <EVT:Key>KVNR</EVT:Key>
      <EVT:Value>A123456789</EVT:Value>
      <!--10-stellige unveränderliche Versichertennummer / Versicherten_ID-->
    </EVT:Parameter>
  </EVT:Message>
</EVT:Event>
```

Das Attribut Filter des Elements Topic ist nicht angegeben, da es optional und nur beim Abonnieren von Ereignissen zu verwenden ist (siehe folgender Abschnitt).

Für die Umsetzung des Ereignisdienstes auf Primärsystemseite ist - abhängig von Architektur und eingesetzter Technologie - zu entscheiden, ob ein solcher Dienst im Primärsystem (server-seitig) einmalig oder auf jedem Arbeitsplatz (client-seitig) bereitgestellt wird.

Sonderfall `CardType=UNKNOWN`

Wird durch den Benutzer eine Karte gesteckt, die durch den Konnektor nicht korrekt identifiziert und gelesen werden kann (falsche Karte, Karte falsch gesteckt, Karte defekt), meldet der Konnektor dies durch das Ereignis `CARD/INSERTED` mit dem speziellen Kartentyp `UNKNOWN`. Das Primärsystem sollte eine entsprechende Meldung ausgeben und den Benutzer ggf. zur Korrektur auffordern.

4.1.4.2 Abonnieren von Ereignissen

Zum Abonnieren von Topics stellt der Konnektor die Funktionen `Subscribe`, `Unsubscribe` und `GetSubscription` zur Verfügung. Beim Abonnieren von Topics lassen sich Filter auf Ereignisse setzen, wobei sich mittels XPath-Ausdrücken Ereignisse über `Typ` und `Severity` filtern lassen. Alternativ können auch alle Ereignisse abonniert werden. In diesem Fall muss das Primärsystem bei jedem Empfang einer Ereignisnachricht entscheiden, ob und wie diese zu verarbeiten ist.

Wenn es eine Vielzahl von Kartenterminals gibt, die im Netzwerk registriert sind, kann der Fall eintreten, dass mehrere Karten gleichzeitig gesteckt sind. Mit Hilfe selektierender Informationen lassen sich Kartenzugriffe auf die Karten einschränken, die genutzt werden sollen. Die selektierenden Informationen können aus dem Ereignisdienst bezogen werden und helfen dabei, CardHandles zu erlangen, mit denen Kartenzugriffe realisiert bzw. Kartensitzungen aufgebaut werden können.

Ereignisse können gezielt abonniert werden, so dass einzelne Arbeitsplätze nur Ereignisinformationen erhalten, welche die Steckung von Karten in Kartenterminals betreffen, die ihnen zugeordnet sind**.**

Eine Reihe von Informationen über den Status von Karten können unmittelbar zum Zeitpunkt des Steckens einer Karte zur Verfügung gestellt werden, insbesondere die Kartenterminal-ID, an dem aktuell eine Karte gesteckt ist.

⌦ **TIP1-A_4970 Karteninformationen mittels Ereignisdienst verarbeiten**

Das Primärsystem SOLL den Ereignisdienst dazu nutzen, zum Ereigniszeitpunkt Karteninformationen weiterzuverarbeiten und den Nutzern anwenderfreundlich zur Verfügung zu stellen. ⌫

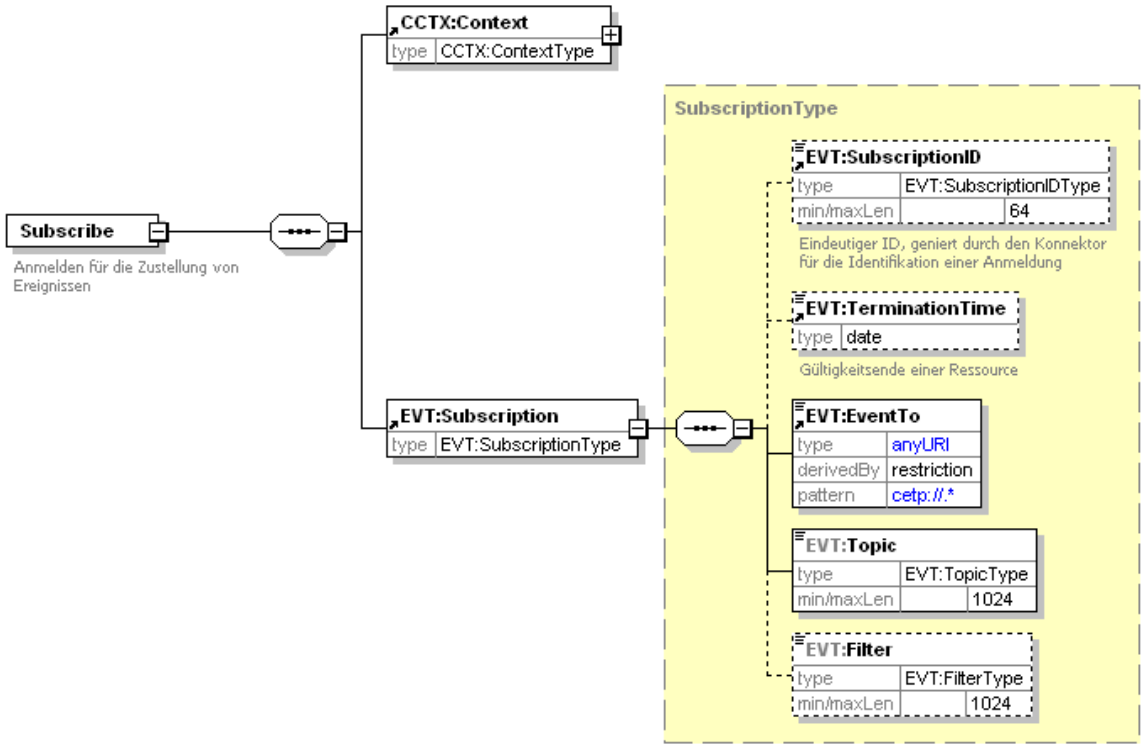


Abbildung 11: Struktur des Elements Subscribe

Tabelle 1: Wichtige Topics für Kartenereignisse

Name	Key/Value im Element Message	Auslöser
CARD/INSERTED	CardHandle =$CARD.CARDHANDLE; CardType =$CARD.TYP; CardVersion =$CARD.VER; ICCSN =$CARD.ICCSN CtID =$CARD.CTID SlotID =$CARD.SLOTID InsertTime =$CARD.INSERTTIME CardHolderName=$CARD.CARDHOLDERNAME KVNR =$CARD.KVNR“	Ereignis des Steckens einer Karte
CARD/REMOVED		Entfernen einer Karte aus dem KT

Eine vollständige Übersicht der vom Konnektor erzeugten Ereignisse mit den dazugehörigen Key/Value-Parametern findet sich in [gemSpec_Kon#AnhF].

Die Ereignisse, die durch Fachmodul VSDM erzeugt und über den Konnektor übermittelt werden, finden sich in 4.3.4.4.

Beispiel 5: SOAP-Request einer Subscription

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <m:Subscribe
      xmlns:m="http://ws.gematik.de/conn/EventService/v7.0"
      xmlns:m0="http://ws.gematik.de/conn/ConnectorContext/v2.0"
      xmlns:m1="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
```

```
        xsi:schemaLocation="http://ws.gematik.de/conn/EventService/v7.0
                              ../conn/EventService.xsd
                            http://ws.gematik.de/conn/ConnectorContext/v2.0
                              ../conn/ConnectorContext.xsd
                            http://ws.gematik.de/conn/ConnectorCommon/v5.0
                              ../conn/ConnectorCommmon.xsd">
        <m0:Context>
               <m1:MandantId>m0001</m1:MandantId>
               <m1:ClientSystemId>csid0001</m1:ClientSystemId>
               <m1:WorkplaceId>wpid007</m1:WorkplaceId>
        </m0:Context>
        <m:Subscription>
          <m:EventTo>cetp://ap007.local:20000</m:EventTo>
           <m:Topic>CARD/INSERTED/</m:Topic>
           <m:Filter>/EVT:Event/EVT:Message/EVT:Parameter[EVT:Key="CtID" and
EVT:Value="101" and ../EVT:Parameter[EVT:Key="CardType" and EVT:Value="EGK"] and
../../EVT:Severity="Info"]</m:Filter>
        </m:Subscription>
     </m:Subscribe>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Im obigen Beispiel werden Ereignisse des Typs CARD/INSERTED abonniert. Es findet dabei zusätzlich ein XPath-Ausdruck als Filter Anwendung, der nur Ereignisse liefert, die sich auf das Kartenterminal mit der Nummer 101 (`CtID=101`), auf den Kartentyp EGK beziehen (`CardType=EGK`) sowie `Severity=Info` (normale Verarbeitung). Das Beispielereignis CARD/INSERTED aus 4.1.4.1 würde damit an cetp://ap007.local:20000 zugestellt werden.

Die Gültigkeit einer Subscription ist auf einen Zeitraum von 25 Stunden begrenzt und muss daher im Tagesrhythmus erneuert werden. Beim Aufruf von `RenewSubscriptions` muss neben dem Aufrufkontext die `SubscriptionID` mitgeliefert werden, die bei der erstmaligen Anmeldung erzeugt wurde und das Ereignisabonnement identifiziert, das erneuert werden soll.

Die Response des Aufrufes von `RenewSubscriptions` gibt Auskunft über den `Status` der Erneuerung und die `TerminationTime` zur `SubscriptionID`.

4.1.4.3 Ereignisse für Konnektorinformationen

Informationen über den Status bzw. Statusänderungen des Konnektors können durch den Ereignisdienst aktuell zur Verfügung gestellt werden, insbesondere zur Online-Verbindung des Konnektors.

⌦ **TIP1-A_4971 Konnektorstatus mittels Ereignisdienst anzeigen**

Das Primärsystem SOLL den Ereignisdienst dazu nutzen, Informationen zum Status des Konnektors zum Ereigniszeitpunkt weiterzuverarbeiten und den Nutzern zur Verfügung zu stellen. ⌫

Tabelle 2: Topics für Konnektorinformationsereignisse

Name	Key/Value im Element Message	Auslöser
`NETWORK/VPN_TI/UP`	keine	Erfolgreicher Aufbau des VPN-Tunnel zur TI

Name	Key/Value im Element Message	Auslöser
NETWORK/VPN_TI/DOWN		Abbau des VPN-Tunnels zur TI
OPERATIONAL_STATE/..	value=true/false	Diverse, siehe [gemSpec_Kon]

Beispiel 6: Subscription-Ausschnitt für kritische Konnektorereignisse

```
...
    <Topic>
       OPERATIONAL_STATE
    </Topic>
...
```

In diesem Beispiel werden alle Konnektorereignisse mit dem Topic „OPERATIONAL_STATE“ auf Topic-Ebene 1 mit dem Schweregrad „Critical“ abonniert. Dies könnte genutzt werden, um den Anwender auf diesen Zustand des Konnektors hinzuweisen, um ggf. weitere Maßnahmen durchzuführen (Fehleranalyse am Konnektor durch Administrator). Werden – wie in diesem Beispiel – keine Topics der Ebene 2 oder 3 angegeben, werden alle entsprechenden Ereignisse zugestellt.

4.1.4.4 Ereignisdienst-Szenario VSDM-Informationen

Durch den Ereignisdienst können Statusinformationen zum Prozess eines angestoßenen VSDM-Updates sowie das Auslesen der VSD für eine Fortschrittsanzeige sofort zur Verfügung gestellt werden. Die entsprechenden Ereignisse `VSDM/PROGRESS/UPDATE` und `VSDM/PROGRESS/READVSD` sind im Abschnitt 4.3.4.4 beschrieben.

Das Primärsystem soll den Ereignisdienst dazu nutzen, den Nutzern eine Forschrittsanzeige zum Prozess eines VSDM-Updates zur Verfügung zu stellen.

4.1.5 Karten/PIN-Handling

4.1.5.1 PS-Dialoge

Das Primärsystem soll für den Benutzer Dialoge zur Verfügung stellen, um die PIN einer SMC-B, eines HSM-B oder eines HBA zu ändern sowie um diese Karten freizuschalten (PIN-Eingabe zur Erhöhung des Sicherheitszustands).

Eine Pin-Änderung ist notwendig, wenn die entsprechende Karte mit einer Transport-PIN ausgeliefert wurde. Diese PIN muss geändert werden, damit die Karte bezüglich entsprechender Sicherheitsfunktionen verwendet werden kann. Ferner kann der LE die PIN zyklisch ändern, um ein höheres Sicherheitsniveau zu gewährleisten. Zur PIN-Änderung muss das Primärsystem die Liste der verfügbaren Karten abfragen und der Benutzer anschließend die gewünschte Karte auswählen. Durch Aufruf der Operation `changePIN` (siehe 4.1.5.2) und anschließender Eingabe der alte PIN (ggf. Transport-PIN) sowie einer neue PIN am Kartenterminal erfolgt die Änderung auf der Karte.

Die Freischaltung einer Karte erfolgt in ähnlicher Weise, indem nach Auswahl einer verfügbaren Karte (Dialog im PS) die Operation `verifyPIN` für diese Karte am Konnektor aufgerufen wird. Die Freischaltung einer Karte zur Erhöhung des Sicherheitszustands ist in 4.1.5.3 beschrieben.

Das Primärsystem soll immer einen Hinweisdialog anzeigen, wenn der Zugriff auf eine Karte wegen eines nicht erhöhten Sicherheitszustands fehlschlägt oder das PS anderweitig eine PIN-Eingabe für eine Karte initiiert. In diesem Fall soll der Benutzer zur weiteren Eingabe an das entsprechende Kartenterminal verwiesen werden.

4.1.5.2 PIN-Änderung

⌦ **TIP1-A_4972 PIN-Initialisierung auslösen**

Das Primärsystem MUSS Dialoge bereitstellen, mit denen die PIN.SMC der SMC-B oder des HSM-B bzw. PIN.CH oder PIN.QES eines HBA initialisiert wird. Zur (erstmaligen) Vergabe einer PIN muss `CardService.changePin` verwendet werden. ⌫

Die Initialisierung der PIN.SMC der SM-B erfolgt im Rahmen der erstmaligen Nutzung des Konnektors bzw. der SM-B durch das Primärsystem. Ein zyklische Änderung der PIN erfolgt mit Hilfe der gleichen Funktion.

Beispiel 7: Webservice-Call CardService.ChangePin

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SOAP-ENV:Body>
      <m:ChangePin
        xmlns:m="http://ws.gematik.de/conn/CardService/v8.0"
        xmlns:m0="http://ws.gematik.de/conn/ConnectorContext/v2.0"
        xmlns:m1="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
        xmlns:m2="http://ws.gematik.de/conn/CardServiceCommon/v2.0"
        xsi:schemaLocation="http://ws.gematik.de/conn/CardServiceCommon/v2.0
                             ../conn/CardServiceCommon.xsd
                            http://ws.gematik.de/conn/CardService/v8.0
                             ../conn/CardService.xsd
                            http://ws.gematik.de/conn/ConnectorContext/v2.0
                             ../conn/ConnectorContext.xsd
                            http://ws.gematik.de/conn/ConnectorCommon/v5.0
                             ../conn/ConnectorCommon.xsd">
        <m0:Context>
            <m1:MandantId>m0001</m1:MandantId>
            <m1:ClientSystemId>csid0001</m1:ClientSystemId>
            <m1:WorkplaceId>wpid007</m1:WorkplaceId>
            <m1:UserId>mmuster01</m1:UserId>
        </m0:Context>
        <m1:CardHandle>c123456789123456789</m1:CardHandle>
        <m2:PinTyp>PIN.CH</m2:PinTyp>
      </m:ChangePin>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Alle PIN-Eingaben erfolgen über eine sichere PIN-Eingabe am Kartenterminal.

Bei mehrfacher Falscheingabe einer PIN kann die daraus resultierende Sperrung durch `CardService.unblockPIN` aufgehoben werden.

Hinweis: Die eGK muss für die Anwendungen des Online-Rollout (Stufe 1) nicht mit einer PIN freigeschaltet werden.

4.1.5.3 Freischaltung von Karten

Bestimmte Operationen erfordern einen erhöhten Sicherheitszustand eines HBA bzw. SM-B (SMC-B oder HSM-B). Die entsprechende Karte muss im Rahmen einer Inbetriebnahme freigeschaltet werden, d. h. der Benutzer muss während definierter Prozesse (z. B. tägliche Inbetriebnahme des Konnektors und/oder des Primärsystems) durch Aufruf der Operation `verifyPIN` angestoßen die PIN eingeben und so den Sicherheitszustand der SM-B erhöht haben.

Das Primärsystem kann den aktuellen Status einer Karte mittels der Operation `GetPinStatus` abfragen um zu prüfen, ob eine Freischaltung notwendig ist. Unter den verpflichtenden Rückgabewerten gilt: `VERIFIED` zeigt den erhöhten Sicherheitszustand an, der Wert `PinStatus.VERIFIABLE` zeigt an, dass eine Freischaltung noch nicht durchgeführt wurde. Die Rückgabewerte TRANSPORT_PIN und EMPTY_PIN bedeuten, dass die PIN noch mit einer Transport- bzw. Leer-PIN ausgestattet ist und noch initialisiert werden muss. Zur Initialisierung sind noch die in `LeftTries` angegebene Anzahl von PIN-Eingabeversuchen möglich. Das Primärsystem kann den Nutzer auf die Anzahl noch möglicher PIN-Eingaben aufmerksam machen, was insbesondere dann vorteilhaft ist, wenn nur noch ein einziger, letzter Versuch möglich ist. Der Rückgabewert BLOCKED weist darauf hin, dass die PIN dreimal falsch eingegeben wurde.

Konkret ist die Eingabe einer PIN in den folgenden Szenarien erforderlich:

- Hochsetzen des Sicherheitszustandes einer SM-B pro Kartensitzung SM-B durch Eingabe der PIN.SMC.
  Anwendungsfälle: Aufbau der TLS-Verbindung zum Intermediär mit gegenseitiger Authentifizierung, Nutzung der SM-B im Rahmen der Card-to-Card-Authentisierung, einfache Signatur (siehe 4.4.1.1).

- Hochsetzen des Sicherheitszustandes des HBA pro Kartensitzung HBA durch Eingabe der PIN.CH.
  Anwendungsfall: Nutzung des HBA zur Card-to-Card-Authentisierung.

- Die Eingabe der PIN.QES des HBA im Zuge der Erstellung der QES. (s. 4.4.1.7)

Der Online-Rollout (Stufe 1) erfordert für den Zugriff auf die geschützten Daten der eGK die Benutzung einer durch Eingabe der PIN.SMC freigeschalteten SM-B oder eines HBA. Durch die Freischaltung wird der Sicherheitszustand der Karten auf das erforderliche Niveau gebracht. Auf diesem Sicherheitsniveau bleiben sie solange, bis sie den Sicherheitszustand verlieren, etwa durch Ziehen der Karte aus ihrem Kartenslot oder durch Neustart des Konnektors.

Die Freischaltung der SM-B kann alternativ zur PIN auch mittels einer bereits freigeschalteten anderen SM-B oder eines bereits freigeschalteten HBA erfolgen.

Die freigeschaltete Kartensitzung der SM-B kann von einem Clientsystem des freischaltenden Mandanten nachgenutzt werden. Zur Nachnutzung des freigeschalteten HBA muss nicht nur der Mandant, sondern auch die User-ID identisch sein und die personenbezogene Verwendung des HBA belegen.

Der Aufbau des SOAP-Request entspricht dem in Beispiel 7: Webservice-Call CardService.ChangePin

4.2 Kartensitzungen

4.2.1 Aufbau von Kartensitzungen

Die Fachanwendung VSDM sowie der Basisdienste QES Signatur und Verschlüsselung erfordern Zugriffe auf eGK, HBA (im Folgenden analog zu verwenden: HBA-qSig, ZOD 2.0) und SM-B. Zu diesen Karten müssen vom Primärsystem aus Kartensitzungen aufgebaut werden, was dem Besitz eines gültigen Karten-Handles einer gesteckten Karte entspricht.

Der Aufbau einer Kartensitzung erfolgt entweder über den Ereignisdienst (siehe 4.1.4.2), was die komfortable und schnellste Möglichkeit aus Sicht des Primärsystems ist, ein `CardHandle` zu erlangen, oder das Primärsystem muss unter den vorhandenen Karten je nach Anwendungsfall vorhandene Karten abfragen und die gewünschte Karte selektieren. Der Zugriff auf die Karten wird dabei auf ihren Nutzungskontext eingeschränkt. Bei der Angabe des Nutzungskontextes (`Context`, vgl. 3.3.1) sind `MandantID`, `PrimärsystemID` und `ArbeitsplatzID` generell verpflichtend.

Kartenoperationen zum Abruf von Karten durch das Primärsystem werden durch den Konnektor über den Systeminformationsdienst `EventService` mit den Operationen `GetCardTerminals`, `GetCards` (siehe [gemSpec_Kon#4.1.6]) sowie dem Kartendienst `CardService` [gemSpec_Kon#4.1.5] angeboten.

4.2.1.1 GetCards

Mittels Systeminformationsdienst `EventService.getCards` kann das Primärsystem direkt ein `CardHandle` anfordern. Dazu ist der entsprechende `Context` (insbesondere die Identifikation des Arbeitsplatzes) korrekt zusammenzustellen. Im Ergebnis der Operation erhält das Clientsystem eine Liste der verfügbaren zugeordneten Karten (siehe normative Vorgaben in [gemSpec_Kon#4.1.6.5.2]). Falls gewünscht, kann unter den zurückgegebenen Karten anhand des Typs `CARDCMN:CardType` die eGK ausgewählt werden (Wertetabelle Kartentypen: [gemSpec_Kon#TAB_KON_500]).

Im Normalfall sollte jedem Arbeitsplatz ein Kartenterminal zugeordnet sein. Falls in einer Umgebung mit mehreren Kartenterminals (größere Praxis, Aufnahme im Krankenhaus) einem Arbeitsplatz mehrere Terminals zugeordnet sind, sollte der Benutzer im Primärsystem auswählen können, welches für den aktuellen Zugriff zu verwenden ist. Gleiches gilt für den Terminal-Slot, sofern mehrere Slots im KT zur Verfügung stehen.

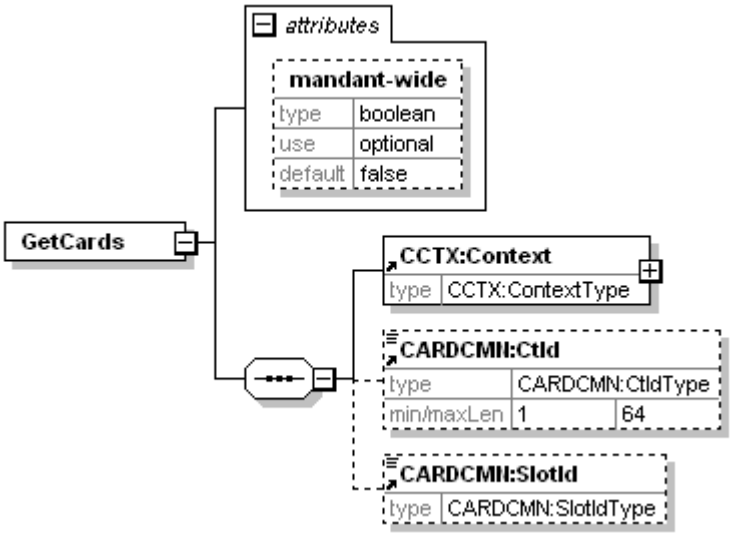


Abbildung 12: Aufrufparameter von GetCards

Beispiel 8: SOAP-Aufruf GetCards

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:m0="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:m1="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:m2="http://ws.gematik.de/conn/CardServiceCommon/v2.0">
  <SOAP-ENV:Body>
    <m:GetCards xmlns:m="http://ws.gematik.de/conn/EventService/v7.0"
                mandant-wide="false">
      <m0:Context>
        <m1:MandantId>m0001</m1:MandantId>
        <m1:ClientSystemId>csid0001</m1:ClientSystemId>
        <m1:WorkplaceId>wpid007</m1:WorkplaceId>
      </m0:Context>
      <m2:CtId>101</m2:CtId>
      <m2:SlotId>01</m2:SlotId>
    </m:GetCards>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Im Beispiel oben werden durch das Primärsystem (bzw. einen konkreten Arbeitsplatz) beim Konnektor alle verfügbaren Karten angefordert, die im Kartenterminal mit der ID 101 im Slot 01 stecken. Durch die genaue Angabe eines konkreten Slots kann maximal eine Karte zurückgeliefert werden.

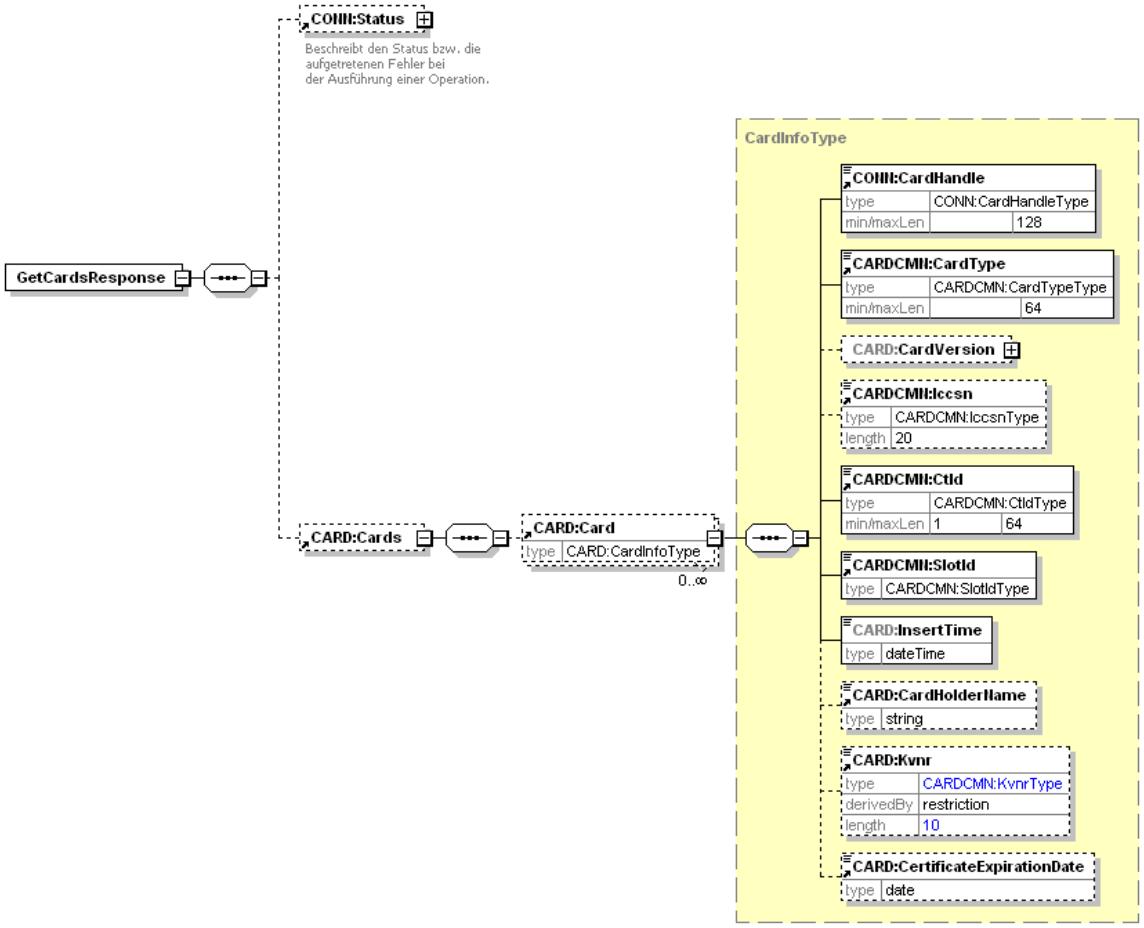


Abbildung 13: GetCardsResponse

Die Abbildung 10 zeigt die Schemadefinition des Wrapper-Elements `GetCardsResponse` mit dem wiederholbaren Element `Card`. Diese entspricht einem Kartenobjekt im Konnektor, welches detailliert in [gemSpec_Kon#4.1.6.5.2]) beschrieben wird. Eine entsprechende SOAP-Antwort könnte folgendermaßen aussehen (nur ein Kartenobjekt gemäß dem obigen Request).

Beispiel 9: GetCardsResponse mit einem Kartenobjekt als Rückgabe

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:CARD="http://ws.gematik.de/conn/CardService/v8.0"
 xmlns:CARDCMN="http://ws.gematik.de/conn/CardServiceCommon/v2.0"
 xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
 xmlns:EVT="http://ws.gematik.de/conn/EventService/v7.0"
 <SOAP-ENV:Body>
  <EVT:GetCardsResponse>
       <CONN:Status>
         <CONN:Result>OK</CONN:Result>
       </CONN:Status>
       <CARD:Cards>
         <CARD:Card>
              <CONN:CardHandle>c123456789123456789</CONN:CardHandle>
              <CARDCMN:CardType>EGK</CARDCMN:CardType>
              <CARD:CardVersion>
                <CARD:SpecPart1>
                     <CARD:Major>2</CARD:Major>
```

```
                    <CARD:Minor>2</CARD:Minor>
                    <CARD:Revision>2</CARD:Revision>
               </CARD:SpecPart1>
               <CARD:SpecPart2>
                    <CARD:Major>2</CARD:Major>
                    <CARD:Minor>2</CARD:Minor>
                    <CARD:Revision>1</CARD:Revision>
               </CARD:SpecPart2>
             </CARD:CardVersion>
             <CARDCMN:Iccsn>8027612345123456781</CARDCMN:Iccsn>
             <CARDCMN:CtId>101</CARDCMN:CtId>
             <CARDCMN:SlotId>01</CARDCMN:SlotId>
             <CARD:InsertTime>2012-12-17T09:30:47</CARD:InsertTime>
             <CARD:CardHolderName>Muster</CARD:CardHolderName>
             <CARD:Kvnr>A123456789</CARD:Kvnr>
             <CARD:CertificateExpirationDate>2016-08-01
             </CARD:CertificateExpirationDate>
          </CARD:Card>
        </CARD:Cards>
  </EVT:GetCardsResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Beim Aufruf von `GetCards` ist die Angabe von Slot und Kartenterminal optional. Wird diese weggelassen, prüft der Konnektor die Verfügbarkeit von Karten in allen Slots aller dem Arbeitsplatz zugeordneten Kartenterminals. Sind dem Arbeitsplatz am Empfang eines MVZ, z.B. 3 Kartenterminals mit je 2 Slots zugeordnet, könnten maximal 6 Kartenobjekte vom Konnektor zurückgeliefert werden. Darüber hinausgehend kann mittels des Attributs `mandant-wide="true"` eine Abfrage initiiert werden, die die Kartenobjekte für sämtliche gesteckte Karten zurückliefert, die sich in allen dem Mandanten zugeordneten Kartenterminals befinden. Die Einschränkung auf die Zuordnung zum angegeben Arbeitsplatz entfällt damit, d.h. die entsprechenden Werte **`csid0001`** und **`wpid007`** im folgenden Beispiel werden ignoriert. [4]

Beispiel 10: Context mit „mandantwide=true“

```
...
    <m:GetCards xmlns:m="http://ws.gematik.de/conn/EventService/v7.0"
                mandant-wide="true">
      <m0:Context>
        <m1:MandantId>m0001</m1:MandantId>
        <m1:ClientSystemId>csid0001</m1:ClientSystemId>
        <m1:WorkplaceId>wpid007</m1:WorkplaceId>
      </m0:Context>
    </m:GetCards>
...
```

Die Operation `getCards` liefert bei Verwendung eines oder mehrerer HSM in der Leistungserbringerumgebung als Kartentyp HSM-B zusammen mit einem `CardHandle` zurück, das eine virtuelle Karte repräsentiert. Aus Sicht der Schnittstelle sind SMC-B und HSM-B gleichwertig, die entsprechenden Karten-Handles gleichartig zu verwenden. Falls der Sonderfall auftritt, dass in der Liste der zurück gelieferten Karten sowohl solche des Typs SMC-B als auch des Typs HSM-B enthalten sind, obliegt dem aufrufenden System

[4] Das Primärsystem kann dazu über einen Schalter „alle Kartenterminals abfragen“ verfügen, den der Benutzer bei Bedarf aktiviert, wenn z. B. das eigene bzw. Standard-Kartenterminal momentan nicht verfügbar ist.

die Entscheidung, welche zu verwenden ist (z.B. anhand von Priorisierung bezüglich Performance der verschiedenen „Karten“).

4.2.1.2 GetCardTerminals

Mit der Operation `GetCardTerminals` des Systeminformationsdienstes kann das PS alle zugeordneten KTs bzw. Slots abfragen und dem Benutzer eine Liste zur Auswahl anbieten.

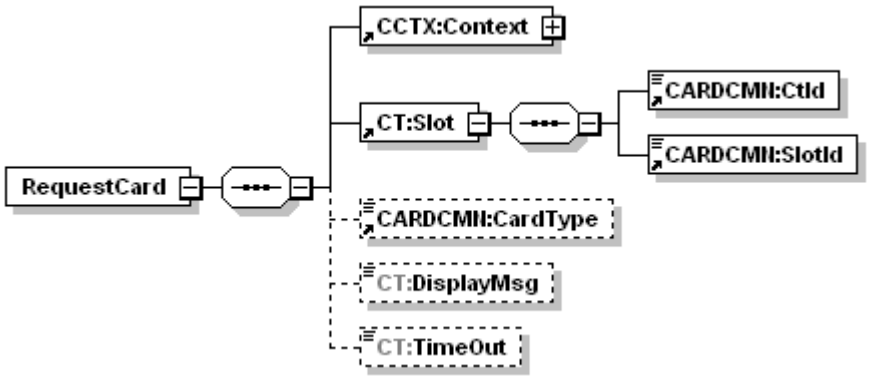
Dieser Fall kann sinnvoll sein, falls die Verfügbarkeit von Kartenterminals im Betrieb geprüft werden soll oder ein Abgleich der Konfiguration damit angestoßen wird.

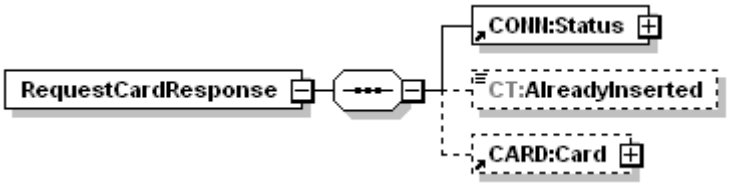
Der Aufruf und die Operation ist ähnlich dem Aufruf von `GetCards` und detailliert in [gemSpec_Kon#4.1.6.5.1] beschrieben.

4.2.1.3 RequestCard

Als Alternative zum Kartenzugriff mittels Informationen des Systeminformationsdienstes - die im Push-Verfahren vom Konnektor bereit gestellt werden - gibt es für das Primärsytem die Möglichkeit, Informationen für den Kartenzugriff im Pull-Verfahren direkt vom Kartenterminal zu beziehen. Dazu dient die Konnektorschnittstelle `CardTerminalService.RequestCard`.

Tabelle 3: Operation RequestCard

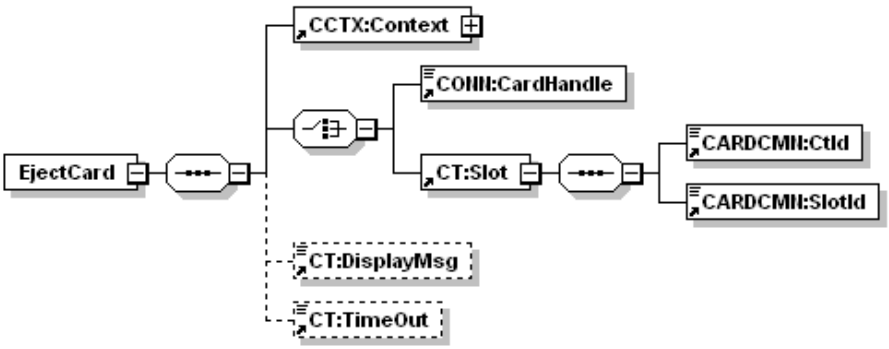
Name	`RequestCard`	
Beschreibung	Liefert die Information zu einer Karte, die in dem Slot eines Kartenterminals steckt oder innerhalb einer bestimmten Zeit (Timeout) gesteckt wird.	
Aufrufparameter		
	Name	Beschreibung
	`CCTX:Context`	MandantId, CsId, WorkplaceId verpflichtend
	`CT:Slot`	Adressiert den 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

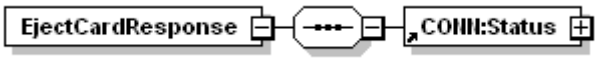
		aufzufordern.
	`CT:TimeOut`	Die Zeit in msec, die maximal gewartet wird bis zum Stecken einer Karte. Wird dieser Parameter nicht übergeben, SOLL der Konnektor den Wert 5000 msec verwenden. Optional KANN dieser Default-Wert im Konnektor konfigurierbar sein.
Rückgabe		
	Name	Beschreibung
	`CONN:Status`	Enthält den Ausführungsstatus der Operation (`OK` oder `Warning` mit Fehlermeldung)
	`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 Informationen zur Karte zurückgegeben: `GetCardsResponse`, wie als Response von `GetCards` beschrieben (4.2.1.1).

4.2.1.4 Exkurs: Auswurf von Karten mittels EjectCard

Einige Kartenterminals besitzen mechanische Vorrichtungen zum Auswurf von Karten aus dem Kartenleser. Diese Funktion kann mittels `CardTerminalService.EjectCard` genutzt werden, um Karten auszuwerfen. Eine geeignete Anzeige auf dem Display des Kartenterminals informiert den Benutzer darüber, die Karte zu entnehmen. Diese Anzeige fordert auch im Falle von Kartenlesern, die nicht über eine Auswurf-Funktion verfügen, dazu auf, die Karten zu entnehmen.

Tabelle 4: Operation EjectCard

Name	**`EjectCard`**
Beschreibung	Beendet die Kommunikation mit der Karte und wirft sie aus, falls das Kartenterminal eine solche mechanische Funktion hat.
Aufrufparameter	
	Name	Beschreibung
	Context	MandantId, CsId, WorkplaceId verpflichtend

	`CONN: CardHandle`	Adressiert die Karte, die ausgeworfen soll. Unterstützt werden die Kartentypen EGK, KVK, HBAx, SM-B und UNKNOWN.
	`CT:Slot`	Adressiert alternativ den Slot eines Kartenterminals, aus dem die Karte ausgeworfen werden soll. Die Adressierung erfolgt über die Identifikation des Kartenterminals `CARDCMN:CtId` und die Nummer des Slots `CARDCMN:SlotId`.
	`CT: DisplayMsg`	Das optionale Feld kann genutzt werden, um den Nutzer über eine Display-Message zu anzuzeigen, die von der Standard-Display-Message abweicht.
	`CT:TimeOut`	Die Zeit in msec, die maximal gewartet wird bis eine Karte gezogen ist. Wird dieser optionale Parameter nicht übergeben, verwendet der Konnektor den Wert 5000 msec, falls kein anderer Wert im Konnektor konfiguriert wurde.
Rückgabe		
	Name	Beschreibung
	`Status`	Enthält den Ausführungsstatus der Operation (`OK` oder `Warning` mit Fehlermeldung)

4.2.2 Kartensitzung eGK

Die Kartensitzung einer eGK wird durch das Primärsystem dadurch aufgebaut, dass es ein `CardHandle` für diese eGK erlangt und nutzt. Dies erfolgt nach dem Stecken der eGK in ein Kartenterminal über eine Ereignismeldung vom Konnektor oder durch eine Benutzerinteraktion am PS (erzeugt `EventService.getCards`).

Sobald ein `CardHandle` für eine gesteckte eGK im Primärsystem vorliegt, bleibt diese gültig, solange die Karte im Kartenterminal gesteckt bleibt. Der Konnektor speichert entsprechende Informationen für die Dauer des Vorhandenseins der eGK – ebenso wie etwaige Veränderungen des Sicherheitszustands der eGK, z.B. durch eine C2C-Authentisierung mittels SMC/HBA.

4.2.3 Kartensitzung SM-B

Die Kartensitzung einer SM-B wird durch das Primärsystem dadurch aufgebaut, dass es ein `CardHandle` für diese SM-B erlangt und nutzt.

Mittels Systeminformationsdienst `EventService.getCards` kann das Primärsystem direkt ein `CardHandle` anfordern. Dazu ist der entsprechende `Context` (insbesondere die Identifikation des Mandanten) korrekt zusammenzustellen. Sofern ein bestimmtes Kartenterminal für die SM-B vorgesehen ist, sollte die entsprechende Kartenterminal-ID im Aufruf enthalten sein.

Im Ergebnis der Operation erhält das Clientsystem eine Liste der verfügbaren zugeordneten Karten (s. [gemSpec_Kon#4.1.6.5.2]). Gegebenenfalls muss unter den zurückgegebenen Karten anhand des Typs die SM-B (bzw. eine der verfügbaren SM-Bs) ausgewählt werden.

Darüber hinaus kann der Ereignisdienst dazu verwendet werden, das `CardHandle` zu erhalten (siehe Kap. 4.1.4). Dazu muss das Primärsystem ein passendes Topic am Ereignisdienst abonniert haben und ggf. eine Interaktion an dem korrespondierenden Arbeitsplatz auslösen.

Zur Nutzung einer SM-B muss eine Kartensitzung, bestehend aus `CardHandle` und `Context` in den Schnittstellenaufrufen verwendet werden. Das Primärsystem kann das `CardHandle` von SM-B für eine geeignete Zeit zwischenspeichern (Caching) und muss bei Bedarf (z. B. Handle ungültig geworden) ein entsprechendes Handle beim Konnektor neu abfragen.

4.2.4 Kartensitzung HBAx

Im Folgenden bezeichnet „HBAx“ den HBA sowie die HBA-Vorläuferkarten wie HBA-qSig und ZOD-2.0.

Die Anwendungsfälle Signieren und Verschlüsseln sind auf eine zuverlässige Identifikation des HBA bzw. seiner Vorläuferkarten angewiesen. Dabei muss die Nutzung der Signaturkarte durch die Person erfolgen, auf welche die Signaturkarte ausgestellt ist. Die HBAx-Kartensitzung, mit der eine Anwendungsschnittstelle (Signieren oder Verschlüsseln, siehe 4.4) aufgerufen wird, muss aus `Context` inklusive `UserId`, sowie dem `CardHandle` bestehen. Die Angabe der `UserId` stellt den Bezug zu einem konkreten Benutzer her und ist ausschließlich bei Signaturerstellung und Verschlüsselung verpflichtend. In einigen wenigen speziellen Anwendungsfällen, etwa beim Auslesen des AUT-Zertifikates des HBAx, ist es möglich, eine HBA-Kartensitzung ohne `UserId` zu verwenden.

Mittels Systeminformationsdienst `EventService.getCards` kann das Primärsystem direkt ein `CardHandle` anfordern. Dazu ist der entsprechende `Context` (insbesondere die Identifikation des Arbeitsplatzes) korrekt zusammenzustellen. Sofern ein bestimmtes Kartenterminal für den HBA vorgesehen ist, sollte die entsprechende `KartenterminalID` im Aufruf enthalten sein.

Im Ergebnis der Operation erhält das Clientsystem eine Liste der verfügbaren zugeordneten Karten (s. [gemSpec_Kon#4.1.6.5.2]). Gegebenenfalls muss unter den zurückgegebenen Karten anhand des Typs der HBAx (bzw. einer der verfügbaren HBAs) ausgewählt werden.

Darüber hinaus kann der Ereignisdienst dazu verwendet werden, das `CardHandle` zu erhalten (siehe 4.1.4).

Zur Nutzung eines HBAxs muss eine Kartensitzung, bestehend aus `CardHandle` und `Context` inklusive `UserId` in den Schnittstellenaufrufen verwendet werden.

4.3 Fachanwendung VSDM

Zur Umsetzung der Fachanwendung VSDM im Rahmen der Erprobung sollen die an das Primärsystem gestellten Anforderungen bezüglich Implementierung und Qualitätskriterien erfüllt werden.

VSDM-A_2925 Umsetzung organisatorischer Anforderungen

Der Hersteller des Primärsystems MUSS für die Erprobungsphase die an ihn gerichteten organisatorischen Anforderungen erfüllen.

4.3.1 Übersicht

In diesem Kapitel wird das Lesen der VSD von der eGK beschrieben. Die zugrunde liegenden Anwendungsfälle sind in der Systemlösung VSDM [gemSysL_VSDM] beschrieben.

Nach dem 1.1.2015 ist die KVK nur noch für den Bereich der Sonstigen Kostenträger ein gültiger Nachweis des Leistungsanspruches, jedoch nicht mehr für den Bereich der GKV-Kostenträger. Daher darf nach dem 1.1.2015 die KVK gemäß [KBV_ITA_VGEX_Mapping_KVK] nur noch im Bereich der Sonstigen Kostenträger verarbeitet werden[5].

Es wird angenommen, dass zu Beginn der Erprobung bereits der größte Teil der Patienten mit einer eGK vorstellig geworden ist und die Versichertenstammdaten der eGK (inklusive KVNR) dem Patientenstamm hinzugefügt wurden. Falls dies nicht der Fall ist, muss eine manuelle Zuordnung der KVNR über die Suchfunktion (z. B. Name) des Primärsystems erfolgen.

Eine Aufstellung der notwendigen Arbeitsplatzkonfigurationsparameter befindet sich im Anhang B1.

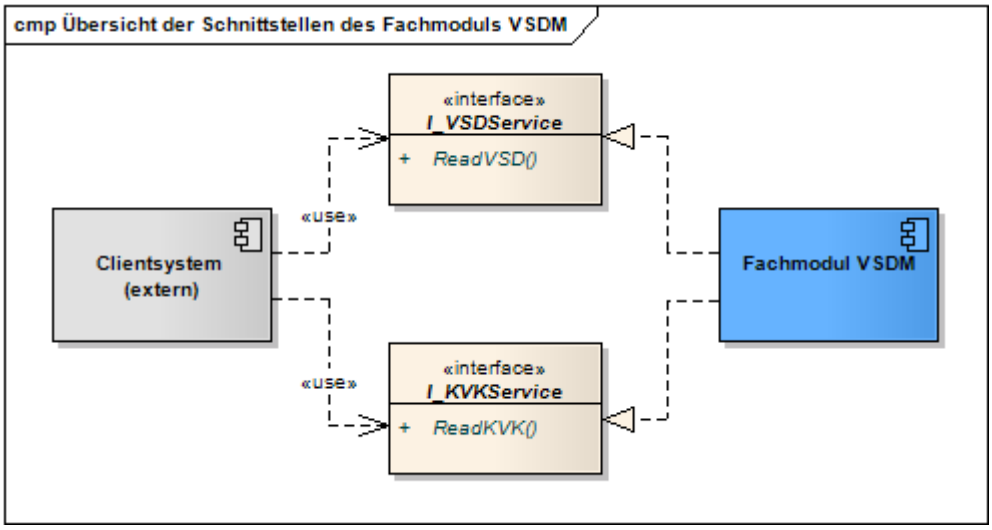


Abbildung 14 – Übersicht der Schnittstellen des Fachmoduls VSDM

4.3.2 Schnittstelle I_VSDService

Die normativen Festlegungen, Schemadarstellung und detaillierte Erläuterung der Parameter zur Schnittstelle befinden sich in [gemSpec_SST_PS_VSDM#4]. Die Schnittstelle stellt die Operation `ReadVSD` [gemSpec_SST_PS_VSDM#4.2] zur Verfügung, mit der sowohl die Online-Prüfung und –Aktualisierung als auch das Lesen der VSD und des Prüfungsnachweises erfolgt.

[5] [KBV_ITA_VGEX_Mapping_KVK], Kap. 2.2.2 mit Verweis auf die Regelungen gemäß Anlage 4a BMV-Ä/EKV

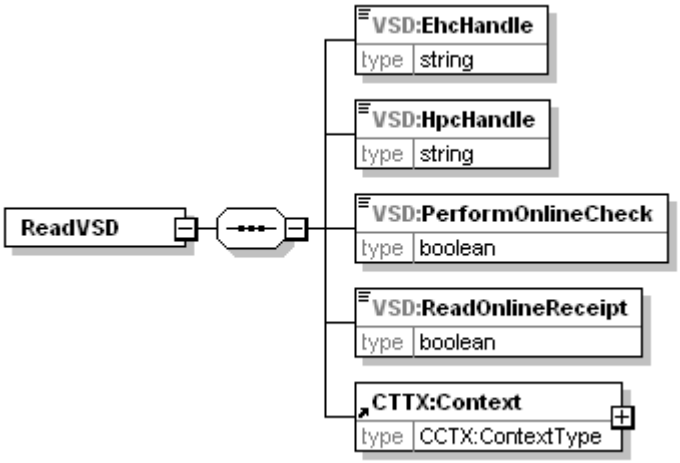


Abbildung 15: Eingangsparameter ReadVSD

Das folgende Schema zeigt die Antwortstruktur der Operation. Dabei sind zwei Elemente optional: Das Element `GeschützteVersichertendaten` wird nur geliefert, wenn der Zugriff durch eine Card-to-Card-Authentisierung mit entsprechender Rolle freigeschaltet wurde. Der `Pruefungsnachweis` wird nur zurückgeliefert, wenn er angefordert worden ist und entschlüsselt werden konnte. Näheres zum Fehlerhandling, wenn der Prüfungsnachweis nicht gelesen werden konnte, findet sich in 6.3.

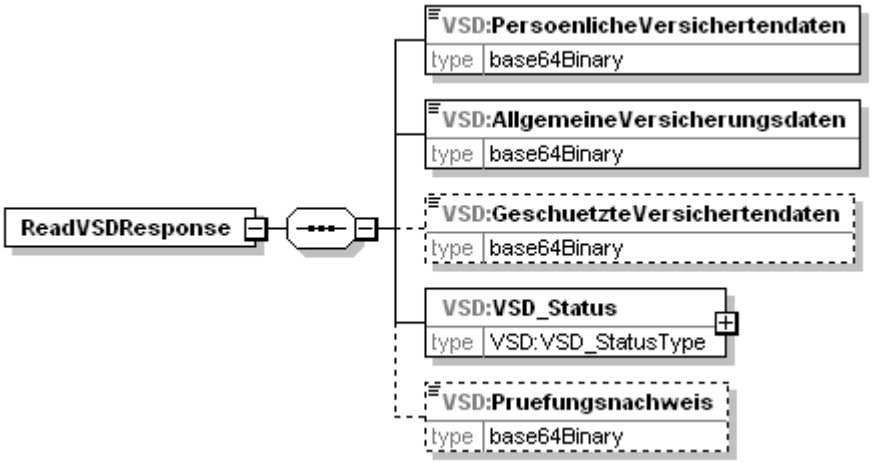


Abbildung 16: Abb_SST_PS_VSDM_05 - Schema der Ausgangsparameter `ReadVSD`

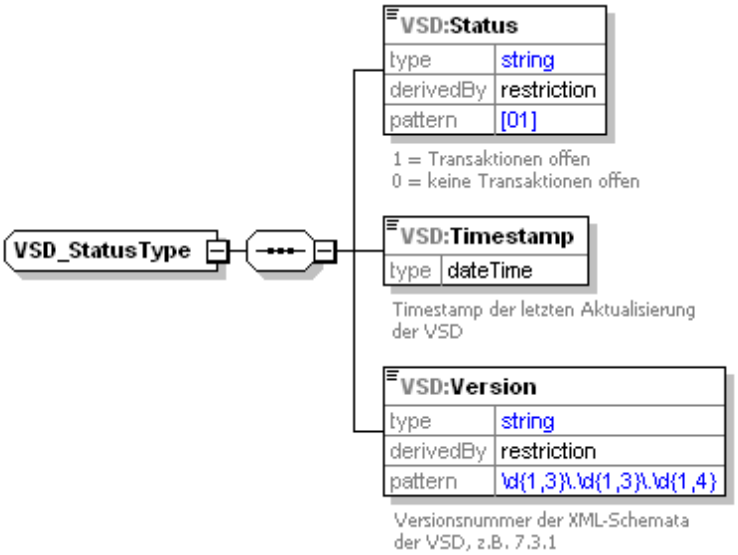


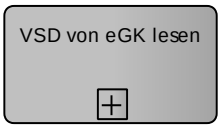
Abbildung 17: Abb_SST_PS_VSDM_06 - Schema von `VSD_Status`

Eine detaillierte Beschreibung zur Kodierung der Daten in den Containern befindet sich im Abschnitt 4.3.5.3 und zum Informationsmodell VSD (Inhalt der dekodierten Container) in Abschnitt 4.3.5.1 sowie im Anhang der Systemlösung VSDM [gemSysL_VSDM].

4.3.3 Anwendungsfall „VSD lesen mit/ohne Online-Prüfung“

Die nachfolgende Prozessmodellierung wurde zur Verbesserung der Lesbarkeit in Subprozesse aufgeteilt.

Subprozesse werden durch ein “+“ in der Aktivität dargestellt



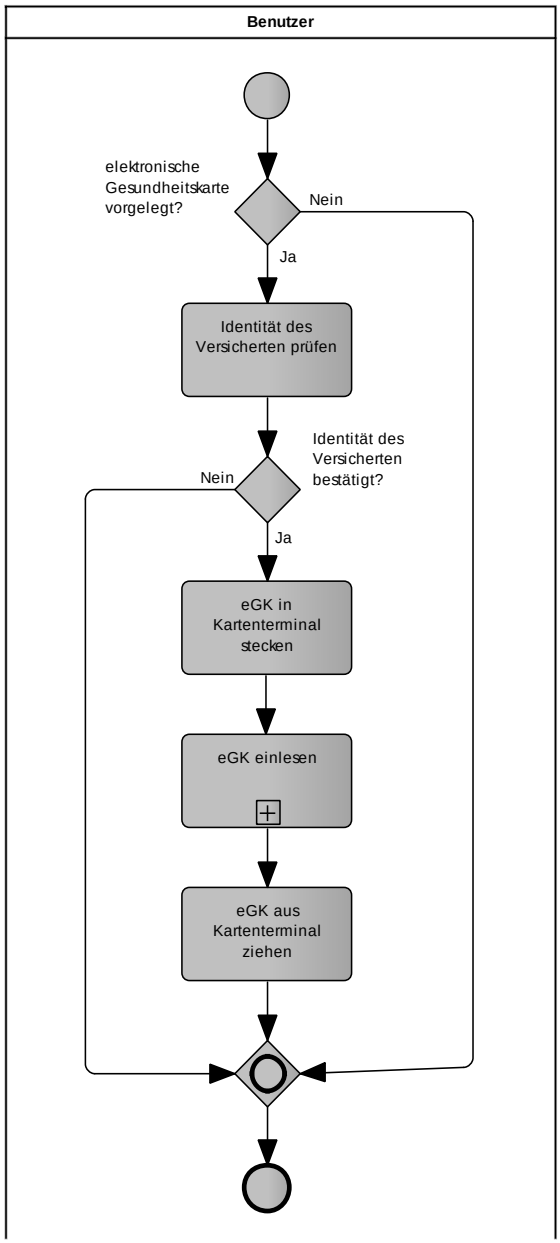


Abbildung 18 – Anwendungsfall „VSD lesen mit/ohne Online-Prüfung“

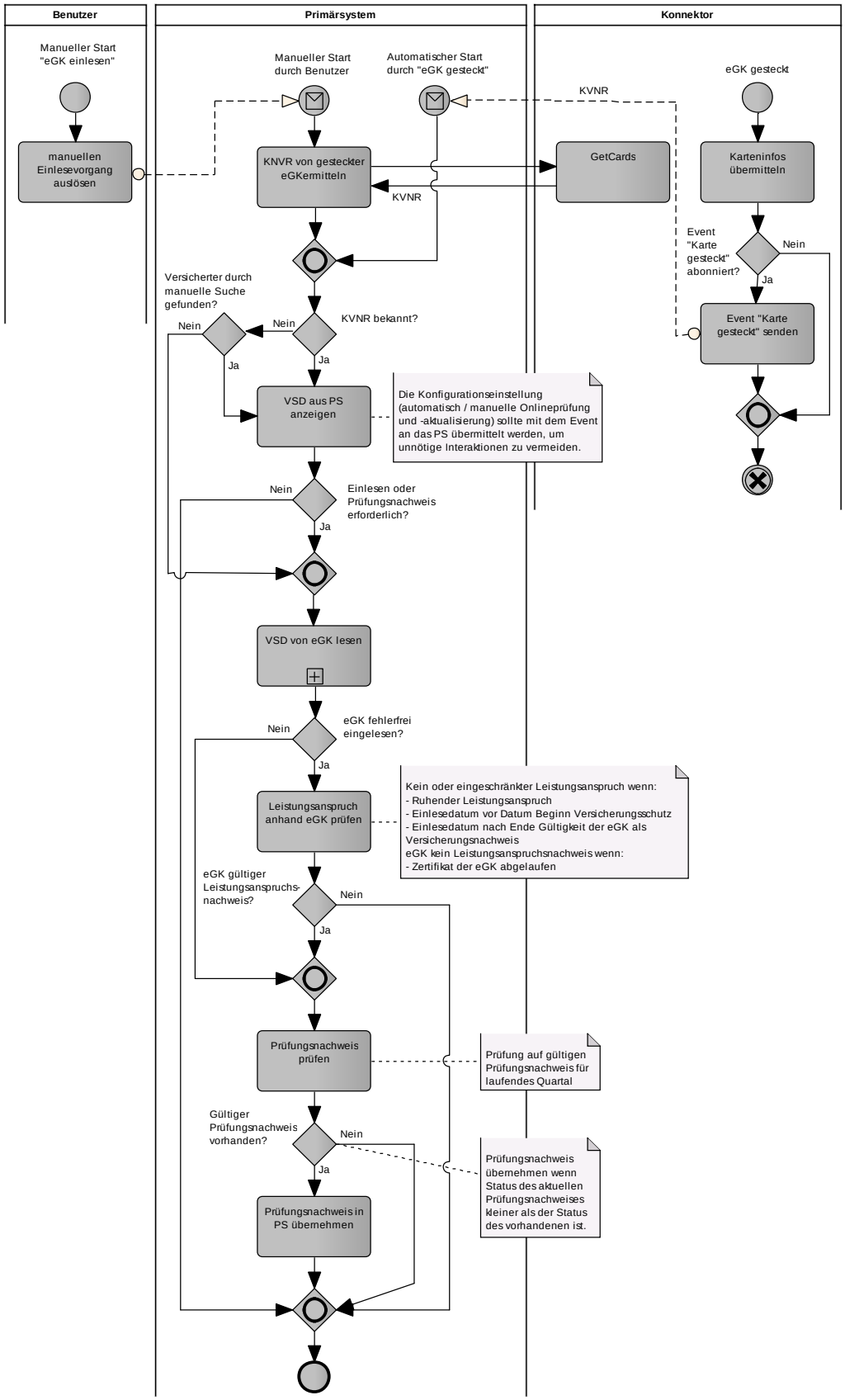


Abbildung 19 – Subprozess „eGK einlesen“

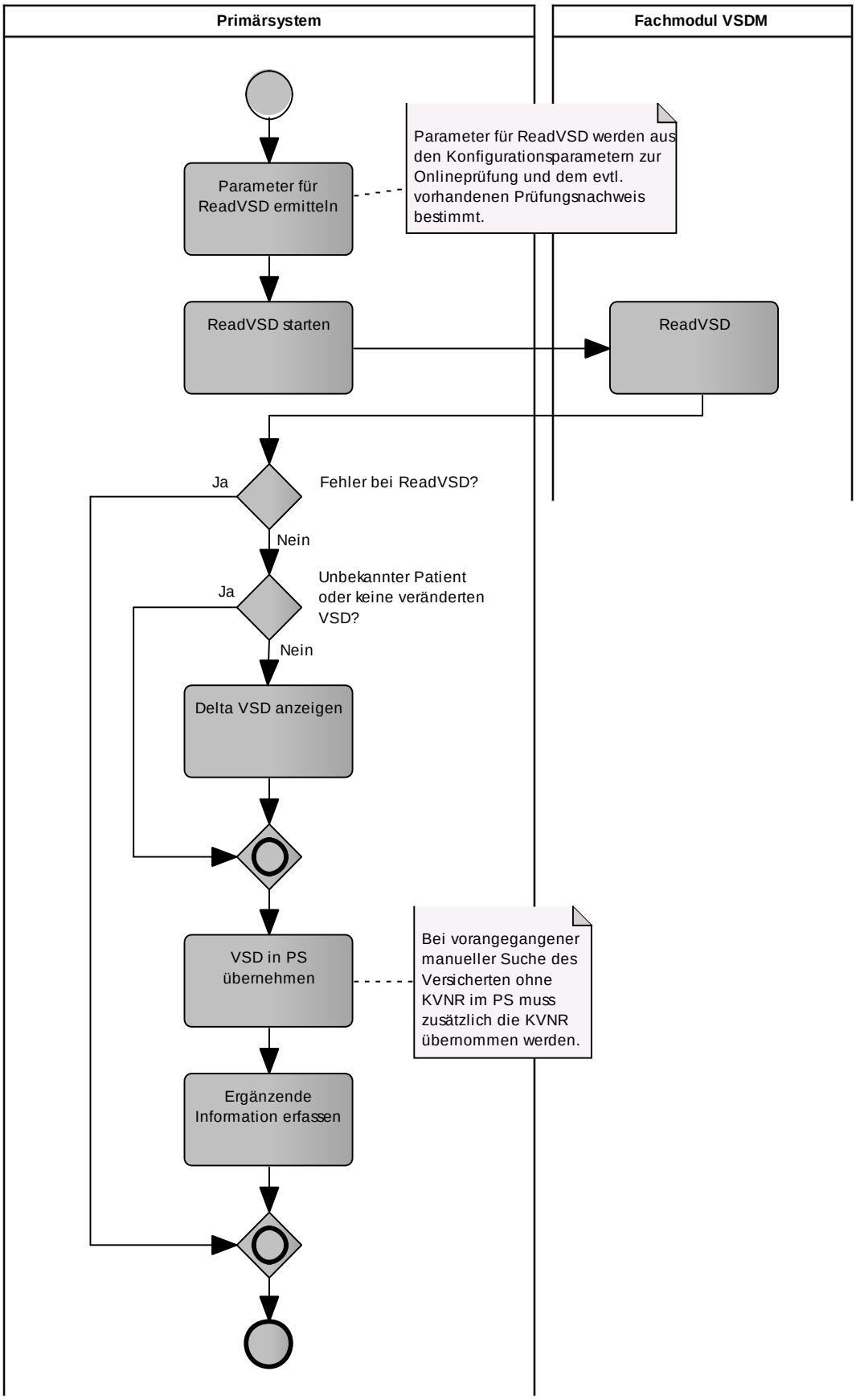


Abbildung 20 – Subprozess „VSD von eGK lesen“

Der Anwendungsfall „VSD lesen mit/ohne Online-Prüfung“ kann gemäß Abbildung 19 – Subprozess „eGK einlesen“ durch einen manuellen Aufruf aus dem Primärsystem oder durch den Ereignisdienst des Konnektors initiiert werden. Die entsprechenden Ereignisse und Parameter sind in 4.1.4.3 beschrieben.

4.3.4 Abläufe im Primärsystem

Im Primärsystem dient bei der Anmeldung die eGK zur Aufnahme bzw. Identifikation des Versicherten. Dabei werden die Versichertenstammdaten ausgelesen und im Primärsystem gespeichert.

Beim Erstkontakt eines Versicherten im Quartal muss zusätzlich eine Online-Prüfung und -Aktualisierung durchgeführt und die Gültigkeit der eGK überprüft werden.

Dies kann auch in einem begründeten Verdacht eines Leistungsmissbrauchs unabhängig von der quartalsweisen Online-Prüfung und -Aktualisierung notwendig werden. Vor dem Einlesen der Versichertenstammdaten muss die Identität des Versicherten anhand der vorgelegten eGK geprüft werden.

4.3.4.1 Patientendatensatz anzeigen

Die Versichertennummer der eGK (KVNR) ersetzt die bisherige Versichertennummer der KVK. Die KVNR ist, anders als bei der KVK, lebenslang gültig und eindeutig.

In diesem Dokument wird im Folgenden die Abkürzung KVNR immer für den 10stelligen unveränderlichen Teil verwendet.

⌦ **VSDM-A_2542 Versichertennummer überschreiben**

Das Primärsystem DARF die Versichertennummer NICHT durch die KVNR überschreiben, solange die KVK nicht vollständig als Leistungsanspruchsnachweis durch die eGK abgelöst wurde. ⌫

Im Gegensatz zur manuellen Suche des Versicherten (z.B. mittels Name, Vorname und Geburtsdatum) besteht durch den Einsatz der eGK die Möglichkeit, den Versicherten anhand seiner eindeutigen Krankenversicherungsnummer (KVNR) automatisch im Primärsystem zu identifizieren. Beim erstmaligen Einlesen einer eGK zu einem bekannten Patienten ist eine manuelle Zuordnung zum bereits vorhandenen Patientenstamm nötig.

Zur Aufnahme eines Versicherten wird die eGK in das Kartenterminal gesteckt. Grundsätzlich lässt sich der Aufnahmeprozess auf zwei unterschiedliche Arten durchführen:

1. Automatische Identifikation des Datensatzes des Versicherten im Primärsystem beim Stecken der eGK
2. Manuelle Identifikation des Datensatzes des Versicherten im PS vor dem Stecken der eGK oder bei nicht erfolgreicher Identifikation mittels KVNR der eGK

Auf welche Weise der Aufnahmeprozess gestartet wird, wird in der Konfiguration des Primärsystems festgelegt oder ist ein Leistungsmerkmal des PS. Empfohlen wird die Unterstützung der automatischen Suche im PS, die – falls dies nicht erfolgreich war – immer durch eine manuelle Suche ergänzt werden können muss.

Automatische Identifikation des Versicherten

Voraussetzung für die automatische Identifikation des Versicherten mittels KVNR ist deren Kenntnis. Dies kann, ohne Auslesen der VSD, durch ein Abonnement des Events „Karte gesteckt“ oder durch eine Statusabfrage der gesteckten Karte(n) beim Konnektor erfolgen.

⌦ **VSDM-A_2872 Identifikation des Versicherten mittels KVNR**

Das Primärsystem SOLL die Zuordnung von Versichertem und Datensatz im Primärsystem zur Identifikation des Versicherten mit der KVNR (unveränderlicher Teil) durchführen, da nur die KVNR einen eindeutigen Bezug zum Versicherten herstellt.⌫

Nach der Übermittlung der KVNR durch den Konnektor prüft das Primärsystem, ob sich der Versicherte bereits im Patientenstamm des Primärsystems befindet.

⌦ **VSDM-A_2529 Automatische Anzeige im Primärsystem nach Identifikation des Versicherten mittels KVNR**

Das Primärsystem SOLL nach der Identifikation des Versicherten mittels KVNR die Patientenstammdaten anzeigen.⌫

Befindet sich der Versicherte noch nicht im Patientenstamm, wird der Benutzer darüber informiert. Im Falle einer Neuanlage werden die Versichertenstammdaten von der eGK gelesen und zur Neuaufnahme angezeigt.

Manuelle Identifikation des Versicherten

Bei dieser Konfiguration muss der Benutzer vor dem Stecken der eGK die Patientenstammdaten anhand von Suchparametern (z.B. Name, Vorname und Geburtsdatum) im Bestand des Primärsystems suchen. Anschließend steckt er die eGK des Versicherten in das Kartenterminal, um die Daten des Versicherten einzulesen. Dieser Ablauf sollte nur in Ausnahmefällen angewendet werden, wenn die Identifikation anhand einer manuell oder automatisch ermittelten KVNR fehlschlägt.

Bei einer manuellen Identifizierung des Versicherten im PS sollte der Benutzer beim Öffnen des Patientendatensatzes einen speziellen Hinweis erhalten, wenn die eGK des Patienten im laufenden Quartal bereits eingelesen worden ist, aber noch keine erfolgreiche Online-Prüfung durchgeführt werden konnte (Prüfungsnachweis aus laufendem Quartal ist zwar vorhanden, das Ergebnis ist aber 3-6).

4.3.4.2 eGK einlesen

Ist der Versicherte nicht im Patientenstamm vorhanden, kein gültiger Prüfungsnachweis aus dem laufenden Quartal vorhanden oder liegen andere Gründe für eine Aktualisierung vor, muss das Primärsystem das Lesen der eGK initiieren und dabei ggf. eine Online-Prüfung und -Aktualisierung anstoßen.

⌦ **VSDM-A_2535 PS: Automatische Online-Prüfung und -Aktualisierung**

Das Primärsystem MUSS beim Stecken/Einlesen der eGK eine Online-Prüfung und -Aktualisierung gemäß Konfiguration in Tabelle 3 initiieren, wenn der Parameter auf ALWAYS gesetzt ist oder wenn der Parameter auf FIRST gesetzt ist und für das laufende Quartal noch kein Prüfungsnachweis über eine erfolgreiche Online-Prüfung vorliegt.⌫

⌦ **VSDM-A_2532 Hinweis zur Durchführung Online-Prüfung und -Aktualisierung aufgrund Datum der letzten Aktualisierung**

Das Primärsystem SOLL dem Benutzer einen Hinweis zur Durchführung einer Online-Prüfung und -Aktualisierung geben, wenn das in den Patientenstammdaten hinterlegte Datum der letzten Aktualisierungsprüfung nicht gesetzt ist oder vor dem aktuellen Quartal liegt.⌫

Ein Online-Prüfung und -Aktualisierung muss dabei in folgenden Fällen durchgeführt werden:

- erster Besuch des Versicherten im laufenden Quartal
- vorhandener aktueller Prüfungsnachweis aus im Quartal vorangegangener Online-Prüfung mit den Ergebnissen
  - 3 = Aktualisierung VSD auf eGK technisch nicht möglich,
  - 4 = Authentifizierungszertifikat eGK ungültig,
  - 5 = Online-Prüfung des Authentifizierungszertifikats technisch nicht möglich,
  - 6 = Aktualisierung VSD auf eGK technisch nicht möglich, da maximaler Offline-Zeitraum überschritten
- wenn der Benutzer dies anfordert
- falls im Primärsystem hinterlegt ist, dass die Online-Prüfung immer durchgeführt werden soll, um bestmögliche Aktualität der Daten zu erreichen

Tabelle 5: Konfigurationsparameter zur Online-Prüfung und -Aktualisierung

Empfohlene Konfigurationsparameter zur Online-Prüfung und -Aktualisierung im PS		
`MODE_ ONLINE_ CHECK`	`ALWAYS` (Immer)	Eine Online-Prüfung wird ungeachtet einer vorangegangen Prüfung oder Aktualisierung immer angefordert
	`FIRST` (Quartal)	Eine Online-Prüfung wird nur beim ersten Kontakt im Quartal angefordert. Die Prüfung wird wiederholt wenn die vorangegangene Prüfung wegen technischer Probleme abgebrochen wurde. (Gesetzlichen Minimalanforderung im Rahmen der vertrags(zahn-)ärztlichen Versorgung)
	`NEVER` (niemals)	Nur Standalone-Szenario (PS am Offline-Konnektor): Eine Online-Prüfung wird niemals vom PS angefordert.
	`USER` (Benutzer-interaktion)	Der Benutzer entscheidet individuell über die Durchführung einer Online-Prüfung und -Aktualisierung. Falls das PS die Notwendigkeit einer Online-Prüfung festgestellt hat, sollte dies in Form einer Bestätigung erfolgen.

⌦ **VSDM-A_2988 PS: Konfigurationsparameter für PerformOnlineCheck**

Das Primärsystem MUSS über einen Konfigurationsparameter zur Steuerung des Verhaltens der Operation `ReadVSD` bezüglich Online-Prüfung und -Aktualisierung gemäß Tabelle 5: Konfigurationsparameter zur Online-Prüfung und -Aktualisierung verfügen.⌫

Folgende Tabelle zeigt die notwendigen Werte der Parameter `ReadOnlineReceipt` und `PerformOnlineCheck` in Abhängigkeit von der Systemkonfiguration (des gewünschten Verhaltens) und des Vorhandensein eines gültigen Prüfungsnachweises für das aktuelle Quartal.

Tabelle 6: Entscheidungstabelle Parametrisierung ReadVSD

Konfiguration der Online-Prüfung	Status des gespeicherten Prüfungs-nachweises im PS (lfd. Quartal) *)	ReadVSD Parameter	
		`ReadOnlineReceipt`	`PerformOnlineCheck`
`MODE_ONLINE_CHECK = USER` (Online-Szenario und Bestätigung durch Nutzer)	Nicht vorhanden	`true`	`true`
	1,2	`false`	`true`
	3-6	`true`	`true`
`MODE_ONLINE_CHECK = ALWAYS` (Online-Szenario)	Nicht vorhanden	`true`	`true`
	1,2	`false`	`true`
	3-6	`true`	`true`
`MODE_ONLINE_CHECK = FIRST` (Online-Szenario)	Nicht vorhanden	`true`	`true`
	1,2	`false`	`false`
	3-6	`true`	`true`
`MODE_ONLINE_CHECK = NEVER` (PS am Offline-Konnektor des Standalone-Szenario)	Nicht vorhanden	`true`	`false`
	1,2	`false`	`false`
	3-6	`true`	`false`

*) Diese Spalte entspricht dem Element `Pruefungsnachweis.Ergebnis` und bedeutet für die Werte 1 und 2 einen im PS vorlegenden Prüfungsnachweis nach fehlerfreier Online-Prüfung (1=Aktualisierung erfolgreich durchgeführt, 2=keine Aktualisierung notwendig). Die Werte 3-6 deuten auf einen Fehler bei der Online-Prüfung oder –Aktualisierung und damit die Notwendigkeit einer erneuten Prüfung hin.

4.3.4.2.1 Online-Szenario

Damit das Clientsystem steuern kann, ob eine Online-Prüfung durchgeführt werden soll, bietet die Operation den Parameter `PerformOnlineCheck`. Ist der Parameter auf `true` gesetzt, führt das Fachmodul eine Aktualisierungsanfrage durch. Es wird davon ausgegangen, dass das Primärsystem die durchgeführten Online-Prüfungen aufzeichnet.

Ist der Parameter auf `false` gesetzt, führt das Fachmodul nur aus fachlichen Gründen gemäß [gemSysL_VSDM#VSDM-UC_01] eine Aktualisierungsanfrage durch, z. B. wenn die Gesundheitsanwendung der eGK bereits gesperrt ist.

Ebenfalls legt das Clientsystem mittels des Parameters `ReadOnlineReceipt` fest, ob ein Prüfungsnachweis zurückgegeben wird. Ist der Parameter `ReadOnlineReceipt=true` gesetzt, wird ein Prüfungsnachweis zurückgegeben, andernfalls enthält die Antwort (Response) keinen Prüfungsnachweis.

Im Online-Szenario ist die Parametrisierung `PerformOnlineCheck=false` und `ReadOnlineReceipt=true` grundsätzlich zulässig, aber im normalen Ablauf nicht sinnvoll. Entweder ist das Fachmodul dabei nicht in der Lage, einen vorhanden Prüfungsnachweis zu entschlüsseln (Prüfungsnachweis erzeugt bei einem anderen LE) oder es handelt sich um einen älteren Prüfungsnachweis, der beim gleichen Leistungserbringer erzeugt worden war. Ein solcher älterer Prüfungsnachweis sollte im Primärsystem bereits gespeichert sein.

4.3.4.2.2 Standalone-Szenario (Primärsystem mit Offline-Konnektor verbunden)

Im Standalone-Szenario ist die Parametrisierung `PerformOnlineCheck=true` beim Aufruf `ReadVSD` **nicht** zulässig („Offline-Konnektor“), da in diesem Fall die Aktualisierung immer scheitert und dadurch ein entsprechend negativer Prüfungsnachweis erzeugt würde. Im Standalone-Szenario ist der Parameter über die Konfiguration des Primärsystems auf `false` zu setzen.

Im Standalone-Szenario ist die Parametrisierung `PerformOnlineCheck=false` und `ReadOnlineReceipt=true` der Standardfall und im normalen Ablauf zu setzen. Es ist davon auszugehen, dass am Online-Konnektor zuvor immer eine Prüfung und ggf. Aktualisierung der Karte stattgefunden hat sowie dabei ein entsprechender Prüfungsnachweis erzeugt und auf die Karte geschrieben worden ist. Dieser wird durch diese Parameterkombination von der Karte gelesen.

4.3.4.3 Benutzerinteraktionen/Anforderungen

⌦ **VSDM-A_2536 Hinweis bei Start Online-Prüfung und -Aktualisierung**

Das Primärsystem MUSS dem Benutzer einen Hinweis geben, wenn die Online-Prüfung und -Aktualisierung gestartet wird.⌫

Ist eine Online-Prüfung und -Aktualisierung nicht notwendig, soll dem Benutzer ein entsprechender Hinweis angezeigt werden. Er kann nun entscheiden, ob die VSD von der eGK gelesen werden sollen. Dies kann der Fall sein, wenn die eGK im Quartal bereits eingelesen wurde, aber eine Aktualisierung der VSD in einer anderen Praxis stattgefunden hat. So können die Daten im Primärsystem an den aktuellen Stand angepasst werden.

Der Benutzer muss die Möglichkeit haben, eine Online-Prüfung auch manuell durchzuführen.

⌦ **VSDM-A_2540 PS: Fortschrittsanzeige bei Online-Prüfung und -Aktualisierung**

Das Primärsystem SOLL dem Benutzer den Fortschritt der Online-Prüfung und -Aktualisierung visuell anzeigen.⌫

Kann die Online-Prüfung und -Aktualisierung nicht durchgeführt werden, z.B. weil der Konnektor zum Zeitpunkt der Anfrage offline ist, darf ein für das aktuelle Quartal im Primärsystem existierender Prüfungsnachweis nicht überschrieben werden.

⌦ **VSDM-A_2537 PS: Hinweis bei fehlgeschlagener Online-Prüfung und –Aktualisierung**

Das Primärsystem MUSS dem Benutzer einen Hinweis geben, wenn die Online-Prüfung und -Aktualisierung aufgrund Nichterreichbarkeit der TI (offline) nicht durchgeführt werden konnte.⌫

⌦ **VSDM-A_2957 PS: Prüfungsnachweise speichern**

Das Primärsystem MUSS alle übernommenen Prüfungsnachweise pro Quartal speichern.⌫

⌦ **VSDM-A_2788 PS: Bereitstellung Ausführungszeiten Online-Prüfung und –Aktualisierung**

Das Primärsystem MUSS Informationen zu Ausführungszeiten der Online-Prüfung und -Aktualisierung für den Support, z. B. in Form von Protokolldateien mit Zeitstempeln, bereitstellen.⌫

Unabhängig von einer Protokollierung der Ausführungszeiten im Primärsystem stehen im Fachmodul des Konnektors Performance- und Fehlerprotokolle zur Auswertung zur Verfügung.

Nach Beendigung wird das Ergebnis der Prüfung durch das Primärsystem angezeigt.

Im Fehlerfall muss dem Benutzer eine aussagekräftige Meldung mit der Fehlerursache angezeigt werden, damit das Ersatzverfahren eingeleitet werden kann.

Bei einer fehlerfreien Durchführung werden die Stammdaten des Versicherten am Primärsystem angezeigt.

Liegen Unterschiede zwischen den im Primärsystem gespeicherten und den von eGK gelesenen VSD vor, soll das PS dem Benutzer die Unterschiede in geeigneter Form darstellen, z. B. Vergleich Alt/Neu mit Hervorhebung der Veränderungen.

⌦ **VSDM-A_2538 PS: Anzeige Delta VSD**

Das Primärsystem SOLL dem Benutzer nach dem Lesen der VSD von der eGK und vor der Übernahme/Speicherung geänderte VSD im Vergleich zu bereits vorhandenen Patientenstammdaten anzeigen.⌫

Der Prüfungsnachweis muss in das Praxisverwaltungssystem übernommen werden, da er Bestandteil der Abrechnung ist.

⌦ **VSDM-A_2873 PS: Standardmäßige Übernahme des Prüfungsnachweises in PVS**

Das PS MUSS, falls es sich um das System eines vertragsärztlichen Leistungserbringer handelt, über die Funktion oder eine Konfiguration verfügen, um bei der Operation `ReadVSD` den Prüfungsnachweis standardmäßig zu übernehmen.⌫

Zur Prüfung des Leistungsanspruchs des Versicherten prüft das Primärsystem das aktuelle Tagesdatum gegen die Angaben zum Versicherungsschutz. Die eGK ist kein gültiger Leistungsanspruchsnachweis, wenn das Tagesdatum vor Beginn des Versicherungsschutzes oder nach dessen Ende liegt.

⌦ **VSDM-A_2543 PS: Hinweis: eGK ist ungültiger Leistungsanspruchsnachweis**

Das Primärsystem MUSS dem Benutzer einen Hinweis anzeigen, wenn die eGK keinen gültigen Leistungsanspruchsnachweis aufgrund der Prüfung des Zeitraums zwischen "Beginn Versicherungsschutz" und "Ende" darstellt.⌫

Dies ist auch der Fall, wenn ein ruhender Leistungsanspruch vorliegt.

⌦ **VSDM-A_2544 Hinweis bei ruhendem Leistungsanspruch**

Das Primärsystem MUSS dem Benutzer einen Hinweis anzeigen, wenn die eGK aufgrund eines ruhenden Leistungsanspruchs keinen gültigen Leistungsanspruchsnachweis darstellt oder der Leistungsanspruch eingeschränkt ist.⌫

4.3.4.3.1 Manuelle Online-Prüfung und -Aktualisierung

⌦ **VSDM-A_2545 PS: Manuelle Initiierung Online-Prüfung und -Aktualisierung**

Das Primärsystem MUSS dem Benutzer die Möglichkeit bieten, die Online-Prüfung und -Aktualisierung manuell zu starten.⌫

Bei dieser Konfiguration entscheidet der Benutzer, ob eine Online-Prüfung und -Aktualisierung durchgeführt wird. Dazu erhält er vom Primärsystem die Information, ob es sich um den Erstbesuch des Versicherten im Quartal handelt (siehe auch [VSDM-A_2532]), oder ob eine erneute Online-Prüfung und -Aktualisierung (z. B. offline) erforderlich ist.

⌦ **VSDM-A_2533 PS: Hinweis zur erneuten Online-Prüfung und -Aktualisierung**

Das Primärsystem SOLL einen Hinweis zur erneuten Online-Prüfung und -Aktualisierung geben, wenn der Status des Prüfungsnachweises für das aktuelle Quartal ungleich 1 oder 2 ist.⌫

Der weitere Ablauf entspricht dem der oben genannten Online-Prüfung und -Aktualisierung.

4.3.4.4 Nutzung der VSDM-Ereignisse des Systeminformationsdienstes

Folgende Tabelle beschreibt die über den Systeminformationsdienst (EventService) des Konnektors durch das Fachmodul bereitgestellten Ereignisse. Sofern das Primärsystem entsprechende Ereignisse abonniert hat (bezogene auf bestimmte Kartenterminals oder alle), werden diese Ereignisse entsprechend zugestellt (siehe Lane „Konnektor“ in Abbildung 19).

Tabelle 7: VSDM-Ereignisse

Name	Key/Value im Element Message	Auslöser

`VSDM/PROGRESS/UPDATE`	`CardHandle =$CARD.CARDHANDLE;`
	`ICCSN =$CARD.ICCSN`
	`CtID =$CARD.CTID`
	`SlotID =$CARD.SLOTID`
	`CardHolderName=$CARD.CARDHOLDERNAME`
	`KVNR =$CARD.KVNR`
		Start einer Aktualisierung der eGK (Update CMS oder Update VSD)

`VSDM/PROGRESS/READVSD`	`CardHandle =$CARD.CARDHANDLE;`
	`ICCSN =$CARD.ICCSN`
	`CtID =$CARD.CTID`
	`SlotID =$CARD.SLOTID`
	`CardHolderName=$CARD.CARDHOLDERNAME`
	`KVNR =$CARD.KVNR`
		Start des Lesens der VSD

Die Nutzung des Systeminformationsdienstes soll sowohl zum Auswerten von Kartenereignissen (Karte gesteckt, Karte entfernt) als auch der VSDM-Ereignisse für eine Fortschrittsanzeige vom Primärsystem umgesetzt werden.

4.3.4.5 Beispiele ReadVSD

Das in der WSDL angegebene SOAP-Encoding „document/literal“, sorgt in Kombination mit dem definierten Schema `VSDService.xsd` und dem darin enthaltenen Root-Element `ReadVSD` für die Kodierung im Beispiel unten (wrapped document/literal, keine Typangaben innerhalb der Elemente, das Element `ReadVSD` entspricht dem Namen der Methode). Damit lässt sich der Body der SOAP-Nachricht direkt gegen das Schema prüfen.

Beispiel 11: Ausschnitt aus VSDService.wsdl

```
...
<binding name="VSDServiceBinding" type="VSD:VSDServicePortType">
  <soap:binding style="document"
                transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="ReadVSD">
      <soap:operation

soapAction="http://ws.gematik.de/conn/vsds/VSDService/v5.2#ReadVSD"/>
      <input>
            <soap:body use="literal"/>
      </input>
...
```

Beispiel 12: Beispiel für einen SOAP-Call ReadVSD

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:m="http://ws.gematik.de/conn/vsds/VSDService/v5.2"
xmlns:m0="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:m1="http://ws.gematik.de/conn/ConnectorCommon/v5.0">
       <SOAP-ENV:Body>
              <m:ReadVSD>
                     <m:EhcHandle>ehc0123456789</m:EhcHandle>
                     <m:HpcHandle>hpc112233</m:HpcHandle>
                     <m:PerformOnlineCheck>true</m:PerformOnlineCheck>
                     <m:ReadOnlineReceipt>true</m:ReadOnlineReceipt>
                     <m0:Context>
                            <m1:MandantId>m0001</m1:MandantId>
                            <m1:ClientSystemId>cs0001</m1:ClientSystemId>
                            <m1:WorkplaceId>wp007</m1:WorkplaceId>
                     </m0:Context>
              </m:ReadVSD>
       </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In obigem SOAP-Aufruf wird die Operation ReadVSD mit folgenden Parametern aufgerufen:

Karten-Handle:

- eGK-Karten-Handle „ehc0123456789“, welches zuvor über eine Meldung des Ereignisdienstes des Konnektors oder über `EventService.getCards()` ermittelt wurde
- SM-B-Karten-Handle „hpc112233“, welches zuvor über eine Meldung des Ereignisdienstes des Konnektors oder über `EventService.getCard()` ermittelt wurde

Online-Prüfung und Prüfungsnachweis:

- mit dem Parameter `PerformOnlineCheck=true` wird eine Online-Prüfung und -Aktualisierung durch den Konnektor initiiert, bevor die VSD zurückgegeben werden
- mit dem Parameter `ReadOnlineReceipt=true` wird der Prüfungsnachweis als Bestandteil von `ReadVSDResponse` angefordert. Dieser wird im Online-Szenario direkt während der Verarbeitung von `ReadVSD` durch das Fachmodul erzeugt und je nach Status (erfolgreich, nicht notwendig, Warnung) mit entsprechendem Ergebnis zurückgeliefert

Context:

- `MandantId` mit Wert „m0001“, die sowohl im Primärsystem als auch im Konnektor so hinterlegt sein muss
- `ClientSystemId` mit Wert „cs0001“, die im Primärsystem fest hinterlegt und im Konnektor konfiguriert und dem Mandanten „m0001“ zugeordnet sein muss
- `WorkplaceId` „wp007“, die sowohl im Primärsystem als auch im Konnektor konfiguriert ist und im Konnektor dem Mandanten „m0001“ als auch dem Primärsystem „cs0001“ zugeordnet ist
- Die Angabe eines Benutzers (`UserID`) ist für `ReadVSD` nur notwendig, wenn ein Karten-Handle eines HBAx verwendet wird (anstelle SM-B).

Auf diese Anfrage zum Fachmodul VSDM des Konnektors sind verschiedene Antworten möglich. Dabei sollen drei Fälle unterschieden werden:

- Erfolg: Rückgabe der VSD inklusive erfolgreich durchgeführter Online-Prüfung und -Aktualisierung (bzw. nicht notwendiger Prüfung)
- Warnung: Rückgabe der VSD, aber mit nicht erfolgreicher Online-Prüfung (entsprechende Ergebnis-Codes im Prüfnachweis)
- Fehler: SOAP-Fault (siehe 6.3)

Beispiel 13: ReadVSDResponse bei Erfolg oder Warnung

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:VSD="http://ws.gematik.de/conn/vsds/VSDService/v5.2"
<SOAP-ENV:Body>
    <VSD:ReadVSDResponse>
      <VSD:PersoenlicheVersichertendaten>UjBsR09Eb...1GUXhEUzhi1GUXhEU
         </VSD:PersoenlicheVersichertendaten>
      <VSD:AllgemeineVersicherungsdaten>UjBsR09EbGhjZ0dT…1tQ1p0dU1GUXhEUzhi
         </VSD:AllgemeineVersicherungsdaten>
      <VSD:GeschuetzteVersichertendaten>UjBsR09EbGh...BRU1tQ1p0dU1GUXhEUzhi
         </VSD:GeschuetzteVersichertendaten>
      <VSD:VSD_Status>
            <VSD:Status>0</VSD:Status>
            <VSD:Timestamp>2001-12-17T09:30:47</VSD:Timestamp>
            <VSD:Version>5.2.0</VSD:Version>
      </VSD:VSD_Status>
      <VSD:Pruefungsnachweis>UjBsR09EbGhjZ...U1GUXhEUzhi</VSD:Pruefungsnachweis>
    </VSD:ReadVSDResponse>
  </SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

Die Inhalte der Elemente `PersoenlicheVersichertendaten`, `AllgemeineVersicherungsdaten`, `GeschuetzteVersichertendaten` und `Pruefungsnachweis` sind komprimiert sowie base64-kodiert (siehe 4.3.5.3) und müssen vor dem Parsen entsprechend dekodiert werden.

4.3.5 Informationsmodell VSD

4.3.5.1 Versichertenstammdaten

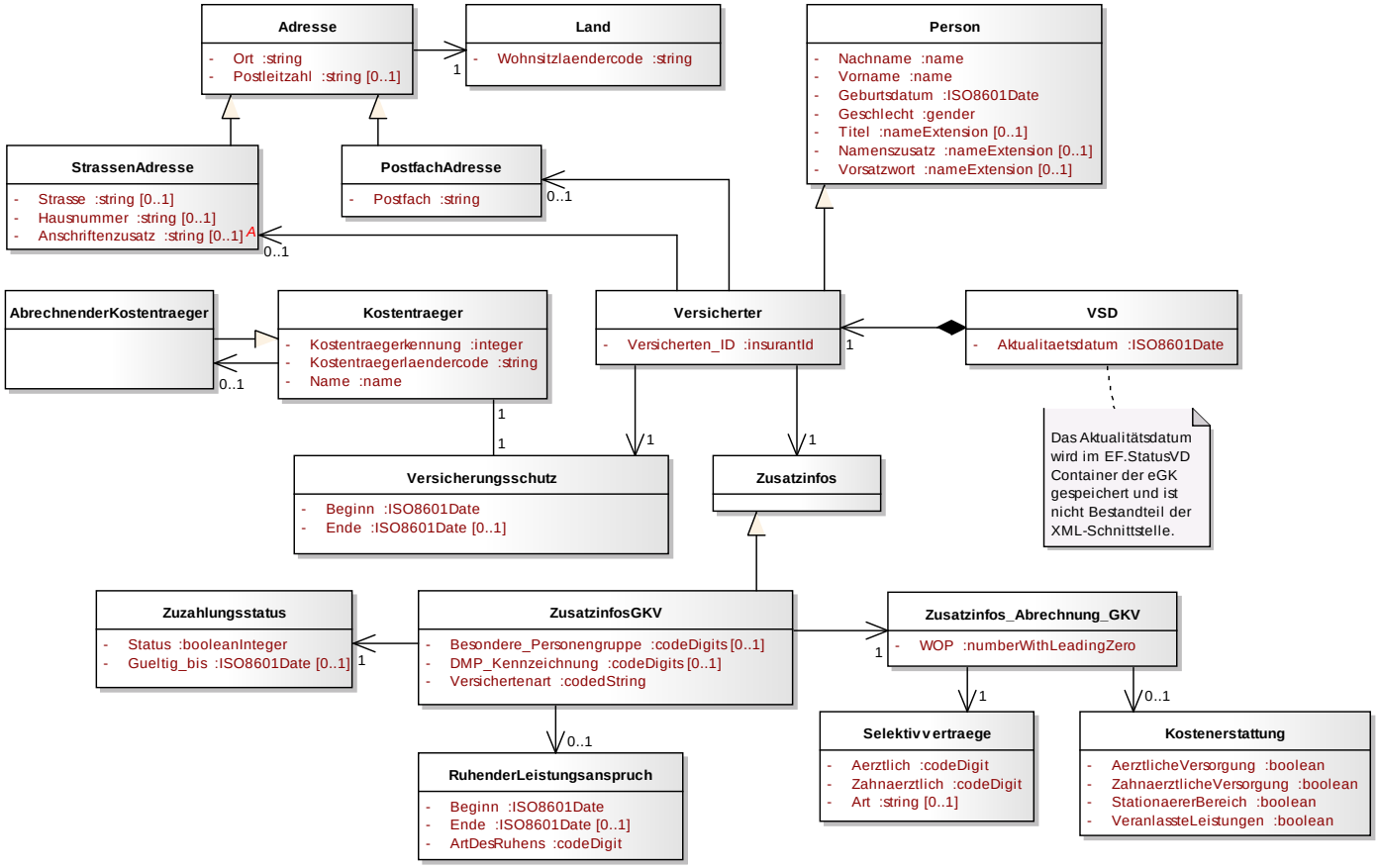


Abbildung 21: Informationsmodell Versichertenstammdaten

Folgende Tabelle zeigt einige für das Primärsystem relevante Änderungen in der VSD-Schemaversion 5.2 gegenüber Version 5.1. Die meisten Änderungen betreffen die Verarbeitungslogik und/oder Datenspeicherung im Primärsystem (z.B. Änderung der Kardinalität oder zusätzliche Daten).

Tabelle 8: Änderungen im VSD-Schema 5.2

Klasse	Änderung
Person	Änderung der minimalen Feldlänge des Feldes „Vorname“ von zwei auf ein Zeichen
Adresse	Änderung der Kardinalität des Feldes „Postleitzahl“, **jetzt optional**
Zusatzinfos GKV	Wegfall des Feldes Rechtskreis und Versichertenstatus RSA
Zusatzinfos_Abrechnung_GKV	Änderung der Kardinalität WOP, **jetzt verpflichtend**
Kostenerstattung	Umbenennung der Felder für ambulante und stationäre Kostenerstattung Änderung der Kardinalität der Klasse „Kostenerstattung“, **jetzt optional** Aufnahme der Felder für zahnärztliche Versorgung und veranlasste Leistungen
Zusatzinfos PKV	Wegfall aller Klassen zur PKV
Ruhender Leistungsanspruch	Aufnahme neue Klasse mit den Feldern Beginn, Ende und Art des Ruhens **Hierbei ist ein spezieller Hinweis im PS sinnvoll, da diese Information Einfluss auf den weiteren Prozess beim LE haben kann.**
Selektivverträge	Aufnahme neue Klasse mit den Feldern ärztliche, zahnärztliche und Art der Selektivverträge **Hierbei ist ein spezieller Hinweis im PS sinnvoll, da diese Information Einfluss auf den weiteren Prozess beim LE haben kann.**

Im Online-Rollout (Stufe 1) kann bei bereits im Feld befindlichen Karten der Generation 1plus auch ein Schema der Version 5.1 gespeichert sein und mittels ReadVSD geliefert werden. Dies geschieht, wenn die betreffende Karte nicht zuvor auf das Schema 5.2 aktualisiert wurde. Die Schemaversion 5.1 ist Bestandteil des Basis-Rollouts und die normativen Vorgaben entsprechend im Release 0.5.3 veröffentlicht.

4.3.5.2 Prüfungsnachweis

Mit Einführung des Online-Rollouts (Stufe1) wird mit den Versichertenstammdaten in der Regel auch der Prüfungsnachweis an das Primärsystem übergeben. Für jeden Patienten wird der für das jeweilige Quartal gültige Prüfungsnachweis im Primärsystem gespeichert. Der auf der eGK des Versicherten befindliche Prüfungsnachweis wird bei erneuter Online-Prüfung und -Aktualisierung überschrieben, so dass sich immer nur der Prüfungsnachweis der letzten Online-Prüfung und -Aktualisierung auf der eGK befindet.

Abbildung 22: Informationsmodell Prüfungsnachweis

4.3.5.3 Zeichenkodierung von Daten

Die von `ReadVSD` und `ReadKVK` zurück gelieferten Ausgangsparameter (Response der SOAP-Nachricht sind mehrheitlich base64-kodierte und gzip-komprimierte XML-Strukturen (VSD_Status).

Zur besseren Einordnung hier eine Übersicht der verschiedenen Datenformate und Konvertierungen für die Container PD, VD, GVD und Prüfungsnachweis.

Tabelle 4 : Übersicht Datenformate

Speicherort/Schnittstelle	Datenelement	Format
auf der eGK gespeichert	Container `EF.PD, EF.VD, EF.GVD`	XML-Elemente gemäß `Schema_VSD_5.2.xsd`, gzip-komprimiert, kodiert nach ISO8859-15 (GVD zugriffsgeschützt)
	Container `EF.Prüfungsnachweis`	XML-Element gemäß `Schema_VSD_5.2.xsd`, gzip-komprimiert, intern kodiert nach ISO8859-15 (symmetrisch verschlüsselt und integritätsgeschützt)
	Container `EF.StatusVD`	25 Byte Binärformat (Version, Status, Zeitstempel)
über die Schnittstelle `ReadVSD` geliefert	SOAP-Nachricht mit VSD Hauptelementen in `ReadVSDResponse`	SOAP-Nachricht selbst ist standardkonform nach UTF-8 kodiert XML Elemente (`Schema_VSD_5.2.xsd`) `PersoenlicheVersichertendaten`, `AllgemeineVersicherungsdaten`, `GeschuetzteVersichertendaten`, `Pruefungsnachweis` sind gzip-komprimiert und base64-kodiert, intern XML kodiert nach

Speicherort/Schnitt-stelle	Datenelement	Format
		ISO8859-15
	`ReadVSDResponse.VSD_Status`	XML-Element VSD_Status (`Schema_VSD_5.2.xsd`)

Bevor die eigentlichen Datenstrukturen verarbeitet werden können, muss eine Dekodierung des Base64-Formates und eine Dekomprimierung erfolgen. Anschließend kann das Parsen und Validieren der XML-Strukturen durchgeführt werden.

Bis zu einem durch die Vertragspartner festzulegenden Zeitpunkt werden GVD zusätzlich im ungeschützten Bereich der eGK gespeichert.

4.3.5.4 Dekodierung und Schemavalidierung

Die Elemente `PersoenlicheVersichertendaten`, `AllgemeineVersicherungsdaten`, `GeschuetzteVersichertendaten` und `Pruefungsnachweis` müssen vor dem Parsen/Auslesen zunächst mittels des Base64-Algorithmus dekodiert werden und anschließend mit Hilfe von gzip dekomprimiert werden.

Danach stehen mindestens 2 XML-Elemente (`PersoenlicheVersichertendaten`, `AllgemeineVersicherungsdaten`) sowie ggf. die optionalen Elemente (`GeschuetzteVersichertedaten`, `Pruefungsnachweis`) zur weiteren Verarbeitung im Primärsystem zur Verfügung.

4.3.6 Schnittstelle I_KVKService

Da die KVK bis auf weiteres noch für den Bereich der Sonstigen Kostenträger und die PKV einen gültigen Versicherungsnachweis darstellt, muss dieser Kartentyp auch weiterhin verarbeitbar sein. Hierzu bietet das Fachmodul VSDM den Aufruf `ReadKVK` an, dem lediglich der Parameter `KVKHandle` übergeben werden muss. Analog zu den bisherigen Abläufen muss das Kartenhandle `KVKHandle` mittels der Basisfunktionen des Konnektors (z. B. `GetCards`) ermittelt werden. In der Rückgabe des Aufrufes erhält man ein `base64Binary`-kodiertes ASN.1-Objekt, das Versichertendatentemplate der KVK. Dieses Objekt wurde vom Fachmodul entsprechend den Anforderungen aus [G04] geprüft, so dass es wie bisher direkt verarbeitet werden kann.

4.3.7 Datenaustausch mit mobilen Einsatzgeräten

Mobile Kartenterminals kommen im Normalfall immer dann zum Einsatz, wenn die Daten nicht direkt in dem Abrechnungssystem erfasst werden können. Diese Fälle treten ein bei

- Hausbesuch
- Leistungserbringung im Umfeld eines anderen Leistungserbringers
- Notfallbehandlung.

Das Einlesen und Speichern von Versichertendaten mit Hilfe eines mobilen Kartenterminals ist auch ein mögliches Szenario für Ausfälle der dezentralen Komponenten der Telematikinfrastruktur (Konnektor bzw. Kartenterminal) als Alternative zum aufwendigeren Ersatzverfahren.

Die Schnittstelle zum mobilen Kartenterminal stellt für eGK-Daten eine Leseoperation mit 4 Ausprägungen zur Verfügung, mit denen die PD, VD, GVD sowie Statusinformationen übernommen werden können. Ein Prüfungsnachweis wird durch das mobile Kartenterminal nicht erzeugt und ist damit nicht auslesbar. Anstelle dessen wird als Bestandteil der Statusinformationen eine Zulassungsnummer des mobilen Kartenterminals übermittelt. Die Verwendung dieser Nummer zu Abrechnungszwecken erfolgt nach Maßgabe der Vertragspartner.

Da in einem mobilen Kartenterminal mehrere Datensätze gespeichert werden können, soll die Übernahme in das Primärsystem derart gestaltet sein, dass die Zuordnung zu den Patientenstammdaten möglichst automatisch abläuft. Eine mehrfache Authentisierung am mobilen Kartenterminal soll vermieden werden.

Die Schnittstelle zum Datenaustausch mit mobilen Kartenterminals basiert auf der Simulation eines Kartenterminals (CT-API) und ist in [gemSpec_MobKT] beschrieben. Die komprimierten Container (gzip) können dabei über spezielle Kartenkommandos direkt gelesen werden. Die anschließende Weiterverarbeitung entspricht der nach der Base64-Dekodierung der XML-Elemente im Anschluss an `ReadVSD` der Webservice-Schnittstelle.

Um mehrere Datensätze auslesen zu können, muss das Primärsystem die Fortschaltsperre des mobilen Kartenterminals in seinem Leseprozess berücksichtigen. Die Fortschaltsperre am MobKT macht es erforderlich, Datensätze einzeln auszulesen und nach dem Auslesen zu löschen, um weitere Datensätze lesen zu können. Durch das Löschen des als übertragen markierten Datensatzes durch das Primärsystem wird sichergestellt, dass Datensätze nicht mehrfach ausgelesen werden können. Die Notwendigkeit des Löschens als ausgelesen markierte Datensätze (Fortschaltsperre) wird vom MobKT durchgesetzt (vgl. [gemSpec_MobKT]#6.5).

4.4 Signaturerstellung und Verschlüsselung

Der Konnektor stellt eine generische Schnittstelle zur Erstellung von Signaturen und Verschlüsselung von Dokumenten zur Verfügung (`SignatureService`). Diese Schnittstelle kann vom Primärsystem in einer Vielzahl von Szenarien genutzt werden.

Die Nutzung des `SignatureService` ist optional. Die Vielfalt der bereitgestellten Funktionalität eröffnet dem Primärsystem eine Reihe von Möglichkeiten, die Signaturserviceschnittstelle im Rahmen ihrer Workflows zu nutzen. Dem Primärsystem wird kein spezielles Nutzungsszenario zur Realisierung vorgeschlagen.

Der `SignatureService` bietet drei Funktionalitäten an, die mithilfe der Smartcards SM-B, HBA und HBA-Vorläuferkarten umgesetzt werden:

- Signieren von Dokumenten (XML, MIME, Text, Tiff, PDF/A, Binär)
- Verschlüsseln von Dokumenten (XML, MIME, Text, Tiff, PDF/A, Binär), eGK kann verwendet werden
- Zertifikatsabfragen auf Smartcards

Die drei Operationen des SignatureService können einzeln genutzt werden. In Kombination eröffnen sie die Möglichkeit, Dokumente mithilfe von Zertifikats- und Verschlüsselungsmaterial von Smartcards zu verschlüsseln und zu signieren, so dass die Schutzziele Integrität und Vertraulichkeit auf dem Niveau einer qualifizierten elektronischen Signatur (QES) umgesetzt werden.

Das Primärsystem kann den Leistungsumfang der Signaturanwendungskomponente (SAK) des Konnektors nur nutzen, wenn am Konnektor der entsprechende Parameter konfiguriert ist.

Der Konnektor kann den Revocation-Status von Zertifikaten im Rahmen des Signatur- und Verschlüsselungsdienstes nur dann überprüfen, wenn der Konnektor die volle Online-Funktionalität nutzt.

Formate von Dokumenten sind dem Clientsystem bekannt und müssen an den unten beschriebenen Schnittstellenaufrufen auch dem Konnektor bekannt gegeben werden, damit dieser die dokumententypspezifischen Verarbeitungsschritte durchführen kann.

Die nicht-XML-Formate werden dabei nach MIME-Typ-Klassen unterschieden:

- “PDF/A” für MIME-Typ „application/pdf-a”,
- “Text” für MIME-Typ “text/plain”,
- “TIFF“ für MIME-Typ “image/tiff”
- „Binär“ für alle übrigen MIME-Typen.

4.4.1 Erstellen digitaler Signaturen

Der Konnektor bietet seinen Clients im `SignatureService` eine Operation zum Signieren von Dokumenten mittels Smartcards an (`SignDocument`) sowie eine Operation zum Verifizieren von signierten Dokumenten (`VerifyDocument`). Die QES beinhaltet die Verwendung eines Trusted Viewers, der über die Schnittstellen `SetXTVView` und `GetXTVView` angesteuert wird.

Hinweis: Eine normative und noch detailliertere Beschreibung der Signaturschnittstelle erfolgt in [gemSpec_Kon#4.1.8.5]. Dort finden sich ggf. auch Erläuterungen zu den Parametern `OptionalInput` etc., die alle Signaturvarianten betreffen und hier nicht aufgeführt sind. Die im Folgenden beschriebenen Parameter dienen nur der Einführung in die Benutzung der Signaturschnittstelle, zu deren vollständigem Verständnis auch die Standards [OASIS-DSS], [CAdES], [XAdES] etc., sowie das Schema „SignatureService“ (z.B. bzgl. der Option OCSP-Antworten in die Signatur einzubetten) herangezogen werden müssen.

Wenn bei der Nutzung der Signatur- und Verschlüsselungsschnittstelle AdES-Profile gelten, so gelten ausschließlich die AdES-BES-Profile. Dabei gelten 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.

Tabelle 9: Konnektorschnittstelle Basisdienst Signaturdienst (nonQES und QES)

Name	`SignatureService`	
Version (KDV)	wird im Produktsteckbrief des Konnektors definiert (aktuell: 7.0.0)	
Namensraum	Aktuell: http://ws.gematik.de/conn/SignatureService/v7.0	
Namensraum-Kürzel	`SIG`	
Operationen	**Name**	**Kurzbeschreibung**
	`SignDocument`	Dokument signieren

	`VerifyDocument`	Signatur verifizieren
	`SetXTVView`	Darstellung von xTV festlegen
	`GetXTVView`	aktuelle Darstellung von xTV erfragen
WSDL	`SignatureService.wsdl`	
Schema	`SignatureService.xsd`	

Die Signaturabläufe unterscheiden sich geringfügig bei Anwendungsfällen, in denen eine QES erzeugt wird, und solchen Anwendungsfällen, in denen nicht qualifiziert signiert wird.

Entscheidend dafür, ob qualifiziert signiert wird oder nicht, sind die verwendeten Zertifikate sowie der Dokumententyp. Insbesondere unterstützt die Operation `SignDocument` den HBAx nur für QES, nicht für nonQES. Im Parameter `CCTX:Context` kann der HBAx nur für die QES, nicht jedoch für nonQES verwendet werden.

Der Unterschied im Ablauf ergibt sich unter anderem im Einsatz des xTV, der für die qualifizierte Signatur vorgesehen ist.

Die Operation `SignDocument` und ihre Parameter lehnen sich an [OASIS-DSS] an. Folgende Typen von Signaturen können am Konnektor erstellt werden:

- XML - Signatur (s. 4.4.1.1), QES oder nonQES
- CMS - Signatur (s. 4.4.1.2), QES oder nonQES
- S/MIME - Signatur (s. 4.4.1.3), QES oder nonQES
- PDF - Signatur (s. 4.4.1.4), QES oder nonQES
- PKCS#1 - Signatur (s.4.4.1.5), nonQES

Bei den Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur“ DARF der HBAx nur mit dem QES-Zertifikat für QES verwendet werden, die nonQES DARF nur mit dem OSIG-Zertifikat der SM-B verwendet werden.

Tabelle 10: Zuordnung zwischen HBAx oder SM-B, Dokumententypen und Signaturtypen

	XML	PDF/A	Text	TIFF	MIME	Binär
SM-B	XML-Signatur, nonQES	PDF-Signatur, nonQES	CMS-Signatur, nonQES	CMS-Signatur, nonQES	S/MIME-Signatur, nonQES	CMS-Signatur, PKCS#1-Signatur, nonQES
HBAx	XML-Signatur, QES	PDF-Signatur, QES	CMS-Signatur, QES	CMS-Signatur, QES	S/MIME-Signatur, QES	CMS-Signatur, PKCS#1-Signatur, nonQES

Das Primärsystem muss den `SignatureService` mit Parametern aufrufen, die jeweils auf einen einzelnen speziellen Daten- und Signaturtyp ausgelegt sind, und die Signatur mit einer einzelnen Signaturkarte durchführen. Eine Mischung von verschiedenen

Datentypen und Signaturtypen in einem einzelnen Aufruf von SignDocument ist nicht zulässig.

Das Primärsystem muss es dem Benutzer ermöglichen, `signDocument` und `VerifyDocument` mit Stapeln von Dokumenten der Dokumententypen XML, PDF/A, Text, TIFF, MIME aufzurufen, die jeweils insgesamt nicht größer sind als 250 MB. Der gesamte, zu signierende Dokumentenstapel eines Aufrufes von `signDocument` darf nicht größer als 250MB sein.

Für die Einzelsignatur wird die Schnittstelle der Stapelsignatur nachgenutzt: Bei der Signatur einzelner Dokumente besteht die Liste der zu signierenden bzw. zu verifizierenden Dokumente jeweils aus einem einzelnen Dokument.

Für die Einzelsignatur wird die Schnittstelle der Stapelsignatur nachgenutzt: Bei der Signatur einzelner Dokumente besteht die Liste der zu signierenden bzw. zu verifizierenden Dokumente jeweils aus einem einzelnen Dokument.

Eine Parallelsignatur wird durch mehrmaligen Aufruf von `signDocument` unter Angabe des entsprechenden Parameters erzeugt.

Dokumenteninkludierende sowie dokumentenexkludierende Gegensignaturen auf bereits im Dokument bestehende Signaturen werden durch Aufruf von `signDocument` unter Angabe eines entsprechenden Parameters erzeugt.

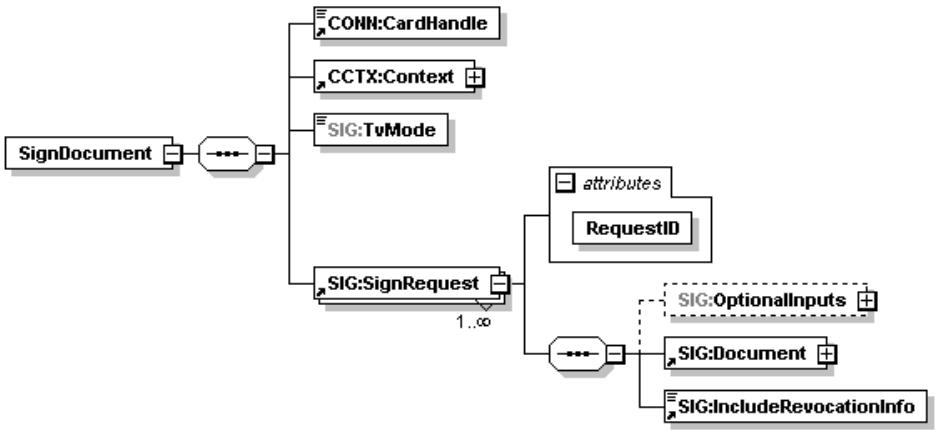


Abbildung 23: Eingangsparameter SignDocument

Anhand der Eingangsparameter steuert der Konnektor den weiteren Signaturvorgang.

- Einfache Signatur ohne Berücksichtigung womöglich bereits bestehender Signaturen, falls `dss:ReturnUpdatedSignature` fehlt.
- Parallelsignatur, falls `dss:ReturnUpdatedSignature` = `http://ws.gematik.de/conn/sig/sigupdate/parallel`
- Dokumentinkludierende Gegensignatur, falls `dss:ReturnUpdatedSignature` = `http://ws.gematik.de/conn/sig/sigupdate/counter/documentiincluding`

- Dokumentexkludierende Gegensignatur, falls `dss:ReturnUpdatedSignature` = `http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding`

Eine Parallelsignatur wird durch mehrmaligen Aufruf von `signDocument` unter Angabe des entsprechenden Parameters (`dss:ReturnUpdatedSignature`) erzeugt.

Gegensignaturen auf bereits im Dokument bestehende Signaturen werden durch Aufruf von `signDocument` unter Angabe des entsprechenden Parameters (`dss:ReturnUpdatedSignature`) erzeugt. Über die Eingangsparameter lässt sich steuern, ob eine dokumenteninkludierende oder eine dokumentenexkludierende Gegensignatur erzeugt wird.

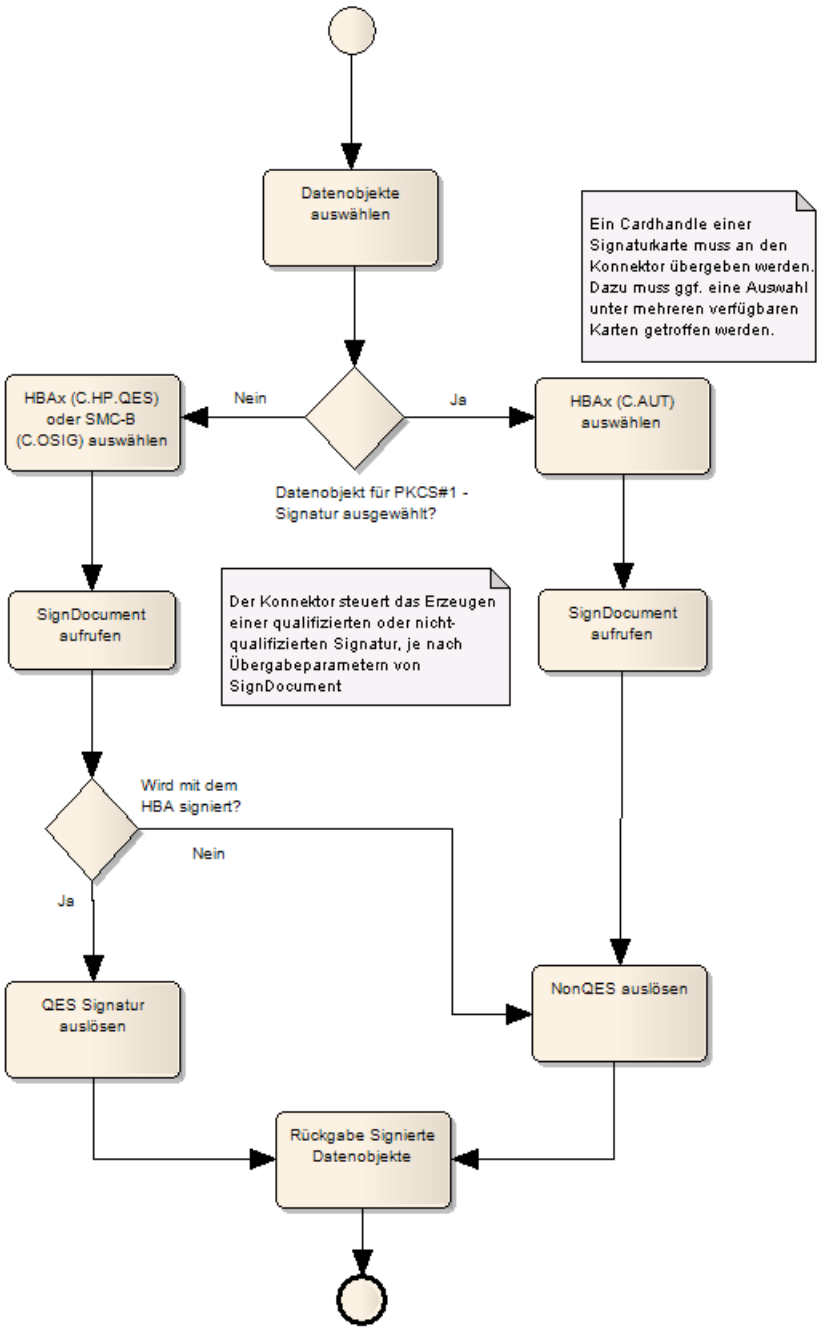


Abbildung 24: Anwendungsfall „Dokumente digital signieren“

Der Konnektor ermöglicht im Zusammenspiel mit einer geeigneten Signaturkarte eine Stapelsignatur gemäß [TR–03114]. Das PS stellt Dokumente zu einem Stapel zusammen, um sie gemeinsam über `SignDocument` zu signieren.

Die Übergabe des Dokumentenstapels an den Konnektor realisiert das Primärsystem als mehrfache Anlage des in [OASIS-DSS] Section 2.4.2 spezifizierten Elementes `dss:Document`. Das darin enthaltene Attribut `ShortText` muss mit einem Ausdruck gefüllt werden, der auf die Identität des Dokumentes schließen lässt, etwa ein Name oder eine Kurzbeschreibung des Dokumentes.

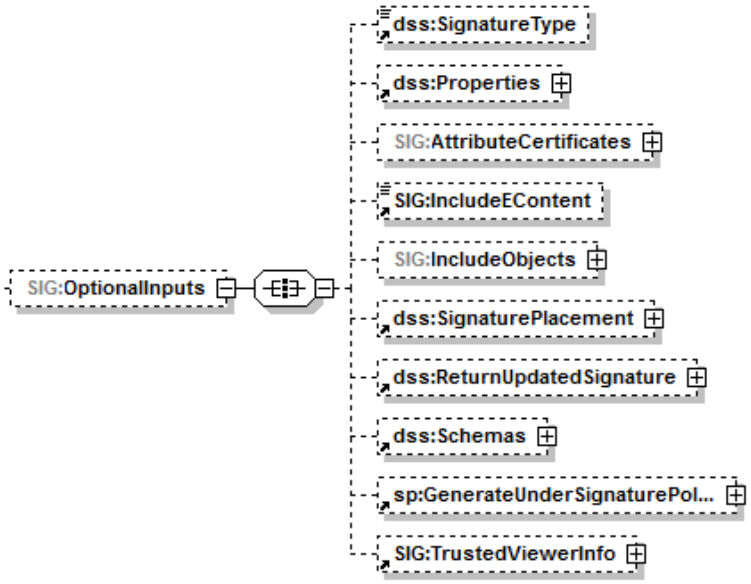
Das Signieren eines einzelnen Dokumentes stellt den Sonderfall eines Dokumentenstapels der Größe 1 dar.

In Bezug auf die QES-Stapelsignatur unterscheiden sich HBAs von HBA-Vorläuferkarten:

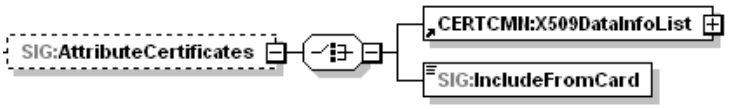
- Die HBA-Vorläuferkarten enthalten ggf. keine für die Stapelsignatur gemäß [TR–03114] geeigneten Signaturzertifikate. Der Konnektor arbeitet mit HBA-Vorläuferkarten die QES eines Dokumentenstapels durch wiederholte Auslösung von Einzelsignaturen inklusive wiederholter PIN-Eingabe am Kartenterminal ab.
- Für HBAs steuert der Konnektor anhand der Angaben im Signaturzertifikat zur maximalen Anzahl in einem Schritt signierbarer Dokumente ggf. eine wiederholte Eingabe der Signatur-PIN am Kartenterminal, falls der gesamte Dokumentenstapel so groß ist, dass er durch den Konnektor in mehreren Teilschritten signiert werden muss.

Listen mit Dokumenten, die nicht qualifiziert signiert werden, signiert der Konnektor ohne Abfragen einer PIN, solange die SM-B freigeschaltet ist.

Tabelle 11: Aufrufparameter zur Signaturerstellung, für mehrere Signaturtypen gültig

`CONN:CardHandle`	CardHandle der Kartensitzung der Signaturkarte
`CCTX:Context`	Aufrufkontext QES mit HBAx: `MandantId`, `ClientSystemId`, `WorkplaceId`, `UserId` verpflichtend Aufrufkontext nonQES mit SM-B: `MandantId`, `ClientSystemId`, `WorkplaceId` verpflichtend; `UserId` nicht ausgewertet
`SIG:OptionalInputs`	Enthält optionale Eingangsparameter (angelehnt an `dss:OptionalInputs` gemäß [OASIS-DSS] Section 2.7): 
`IncludeEContent`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.7), definierte Element kann das Einfügen des signierten Dokumentes in die Signatur angefordert werden.
`dss:ReturnUpdated Signature`	Durch dieses in [OASIS-DSS] (Abschnitt 4.5.8) definierte Element kann eine übergegebene 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/documentincluding Hierdurch wird eine dokumenteninkludierende Gegensignatur für das Dokument und alle vorhandenen parallelen Signaturen erzeugt. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt.
`TvMode`	Legt das Verhalten des Extended Trusted Viewers für den SignRequest-Stapel fest. Erlaubte Werte für bei QES und nonQES: `CONFIRMED` / `UNCONFIRMED`: Der Benutzer kann sich den Inhalt des signierten Dokumentes unter Nutzung des `ShortText`-Attributes aus den `SIG:Document`-Elementen anzeigen und bestätigen lassen (Confirmed, Bestätigungsmodus) bzw. ohne Bestätigung im xTV anzeigen lassen (Unconfirmed, Ansichtsmodus). Zusätzlich erlaubter Wert für TvMode bei nonQES: `NONE` Keine Anzeige im Extended Trusted Viewer.
SIG:SignRequest	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. Innerhalb eines SignRequests MUSS der Konnektor für die enthaltenen Dokumente in Summe eine Gesamtgröße von <= 250 MB unterstützen._
`SIG:IncludeRevocationInfo`	Dieses verpflichtende Element fordert die Einbettung der zum Zeitpunkt der Signaturerstellung vorliegenden Sperrinformationen an. Das Element ist verpflichtend, so dass Sperrinformationen, bzw. Informationen zum OCSP-Status bei der Erstellung einer Signatur immer im Signaturdokument eingefügt sind. Wenn schon Sperrinformationen vorliegen, werden zusätzliche Sperrinformationen ergänzt. Der Wertebereich ist boolean: Im Falle des Wertes `true` fügt der Konnektor den OCSP-Status ein, andernfalls nicht. Der Wert `true` ist der Default-Wert. Für nicht-qualifizierte elektronische Signaturen (nonQES) wird diese Funktionalität nur dann unterstützt, wenn eine Gegensignatur für bestehende qualifizierte elektronische Signaturen (QES) zu erstellen ist. In diesem Fall wird bei der Signaturprüfung der enthaltenen QES die vorliegende Sperrinformation eingebettet.

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.
`SIG:AttributeCertificates`	 Dieses Element ist nur im Fall einer QES-Signatur relevant. Ist dieses Element nicht vorhanden, werden keine Attributzertifikate eingebetttet. Ist dieses Element vorhanden, können über das Element CERTCMN:X509DataInfoList Attributzertifikate beim Aufruf in dem Format mitgegeben werden, wie es die Operation ReadCardCertificate des Zertifikatsdienstes liefert. Alternativ kann der Konnektor über das Element SIG:IncludeFromCard angewiesen werden, alle Attributzertifikate zum Basiszertifikat von der signierenden Karte zu lesen.

Das Primärsystem muss `SIG:IncludeRevocationInfo` durchgängig so setzen, dass OCSP-basierten Sperrinformationen in die Signatur eingesetzt werden. Diese PS-Konfiguration sorgt dafür, dass das Einbetten des Sperrstatus zum Zeitpunkt der Erzeugung der Signatur standardmäßig eingebettet wird, ohne dass der Signierende darüber in jedem Einzelfall entscheiden muss. Als Konsequenz dieser Konfiguration ist bei der Überprüfung einer Signatur keine OCSP-Anfrage mehr erforderlich.

Das Primärsystem kann Metainformationen zu Dokumenten, die qualifiziert signiert werden in Form eines Kurztextes bereitstellen, die Benutzern einen Überblick über den Inhalt dieser Dokumente verschaffen. Bei dem Kurztext bzw. der Metainformation kann es sich beispielsweise um einen Dateinamen handeln, falls das zu signierende Dokument eine Datei ist. Die Kurztexte werden am xTV angezeigt, um dem Benutzer transparent zu machen, welches Dokument signiert wird. Dies ist insbesondere bei größeren Dokumentenstapeln vorteilhaft, bei denen die Gefahr besteht, dass Dokumente unbeabsichtigt mitsigniert werden. Der Kurztext wird der Schnittstelle `SignDocument` vom Primärsystem dem zu signierenden Dokument im Attribut `ShortText` übergeben (nur bei QES).

Um die Verwendung des xTV an den Arbeitsplätzen des Primärsystems zu optimieren, kann die Anzeige konfiguriert werden. Dazu kann das Primärsystem mittels der Operationen `SetXTVView` und `GetXTVView` die Anzeige des Trusted Viewers steuern. Die Steuerung besteht in einer Auswahl passender, im Konnektor konfigurierter TV-Aliase, die für die Anzeige am Arbeitsplatz optimiert sind.

Um eine optimale Anzeige an einem bestimmten Arbeitsplatz zu gewährleisten, muss der Administrator im Vorfeld einmalig am Konnektor für das Primärsystem optimierte xTV-Konfigurationen unter einem Alias anlegen, der daraufhin für diesen Arbeitsplatz mit `SetXTVView` gesetzt wird.

Tabelle 12: Operationen des SignatureService zum Steuern des xTV

Operationsname	Beschreibung
`SetXTVView`	An den Arbeitsplätzen des Primärsystems können speziell in Hinsicht auf Position und Größe des xTV angepasste xTV-Ansichten ausgewählt werden. Dazu wird der passende und am Konnektor konfigurierte xTV-Alias ausgewählt und für den Arbeitsplatz gesetzt.[6]
`GetXTVView`	Um das Setzen des für einen speziellen Arbeitsplatz optimalen xTV-Alias zu unterstützen, kann der aktuell konfigurierte xTV-View abgefragt werden.[7]

4.4.1.1 XML - Signatur

Die XML-Signatur wird per Default als XMLDsig/ XAdES-X (extended) Enveloped Signature umgesetzt, wenn `SignDocument` nicht anderslautend parametrisiert wird.

Eine normative und vollständige Beschreibung der Signaturschnittstelle erfolgt in [gemSpec_Kon#4.1.8.5] und den dort referenzierten Standards.

Für XML-Dokumente, die qualifiziert signiert werden, oder non-QES-Dokumente, die im xTV angezeigt werden sollen, müssen passende XML-Schemata, sowie XSLT-Stylesheets mitgegeben bzw. referenziert werden, damit diese im xTV angezeigt werden können.

Unter den XML-Formaten können die folgenden Formate sicher angezeigt werden:

- VHiTG-Arztbrief [CDA] / eArztbrief [eAB][8],
- eGesamtaufstellung [eGES],

Um die sicher anzeigbaren XML-Formate signieren und signaturprüfen zu können, muss das Primärsystem die Signaturrichtlinien aus [gemSpec_Kon]#Anhang C umsetzen. Im Rahmen der Umsetzung der Signaturrichtlinien wählt das Primärsystem die anzuwendende Signaturrichtlinie aus. Bei signaturrichtlinienkonformem Aufruf von `SignDocument` und `VerifyDocument` werden die sicher anzeigbaren XML-Formate

- zuverlässig erkannt und zur sicheren Anzeige im xTV gebracht
- die Typkonformität der Dokumentenformate mit den jeweils kanonischen XML-Schemata sicher gestellt
- zu PDF verarbeitet, um im xTV angezeigt zu werden
- mit Kurztexten versehen

---

[6] [gemSpec_Kon#4.1.8.5.4]

[7] [gemSpec_Kon#4.1.8.5.5]

[8] Der eArztbrief [eAB] kann auch als PDF-Signatur umgesetzt werden. Dazu erzeugt das PS eigenständig ein PDF, das vom Konnektor signiert wird. Als Alternative dazu kann der Konnektor CDA-Release2-Dokumente gemäß VHitG-Arztbrief-Spezifikation wie angegeben als XML signieren.

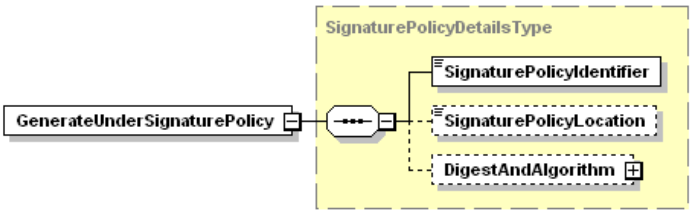
- die Aufrufe der Schnittstelle auf die erforderliche Parametrisierung eingeschränkt.

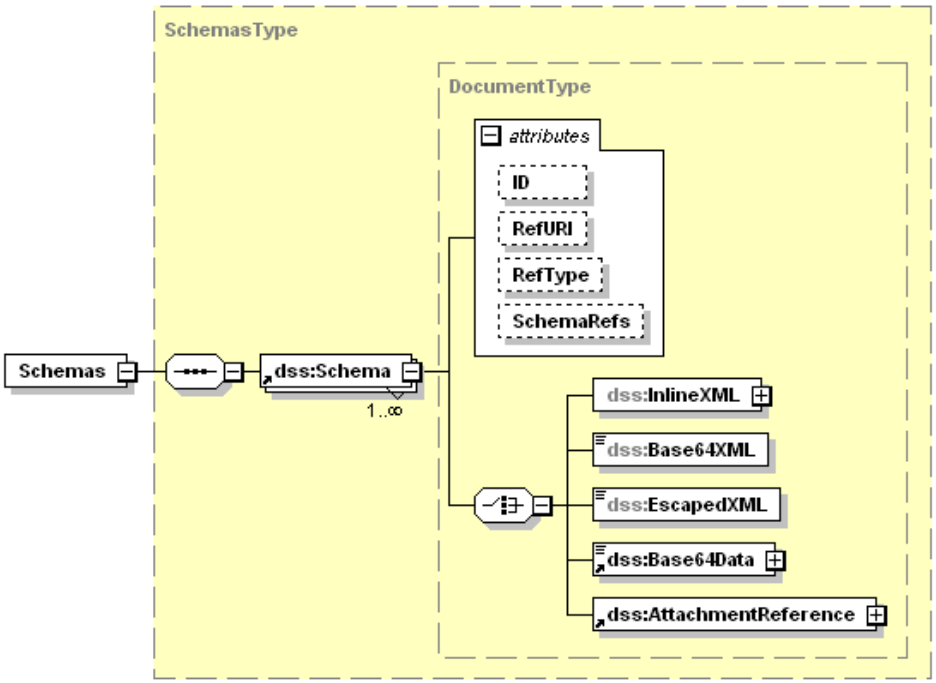
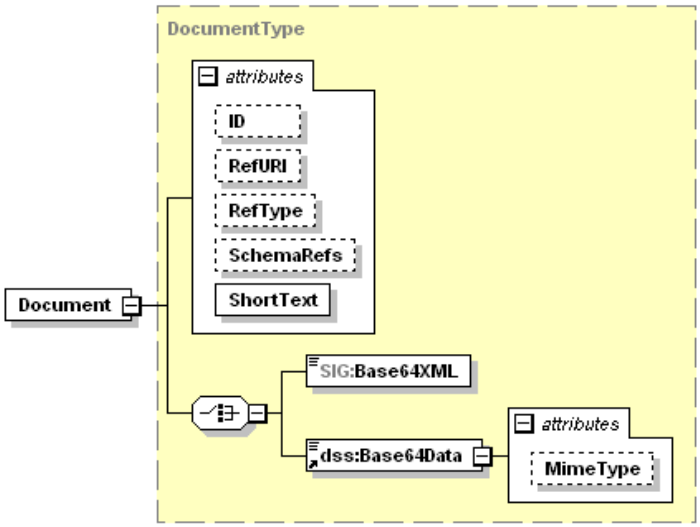
Sicher anzeigbare XML-Formate kann der Konnektor im xTV zur Ansicht bringen und dem Benutzer ermöglichen, eine qualifizierte elektronische Signatur auf diese XML-Formate aufzubringen.

Alle XML-Formate, die nicht sicher angezeigt werden können, gelten als „freie“ XML-Formate.

Tabelle 13: Aufrufparameter speziell für die XML-Signatur

Optionen zur Steuerung der XML-Signatur		
`dss:OptionalInputs`	`dss:SignatureType`	Der Parameterwert `urn:ietf:rfc:3275` wählt XML-Signaturen gemäß [RFC3275] und [XMLDSig]. Das zu verwendende Profil ist XAdES-BES ([XAdES]). Die Rückgabe einer solchen Signatur erfolgt als ds:Signature-Element. Die Rückgabe einer solchen Signatur erfolgt als ds:Signature-Element.
	`SIG:IncludeObject`	Dieses Element enthält eine Folge von `dss:IncludeObject`-Elementen gemäß [OASIS-DSS] (Abschnitt 3.5.6), die bei einer XML-Signatur gemäß [RFC3275] die Liste der Objekte, die in die Signatur einbezogen werden sollen, spezifiziert. Ist eine XML-Signatur angefordert und fehlt dieses Element, so wird jeweils das komplette Dokument signiert.
	`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. Bei anderen Signaturtypen wird das Element ignoriert und eine Warnung (Fehlercode 4197, Parameter SignaturePlacement wurde ignoriert) zurückgeliefert.
	`dss:ReturnUpdated Signature`	Durch dieses in [OASIS-DSS#Abs.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/documentincluding Hierdurch wird eine dokumenteninkludierende Gegensignatur für das Dokument und alle vorhandenen parallelen Signaturen erzeugt. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt.

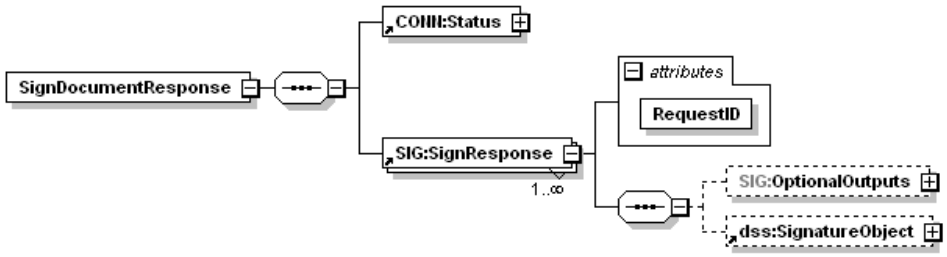
	`sp:GenerateUnderSignaturePolicy`	 Über dieses in [OASIS-SP], Kapitel 2.2.1.1.1.1 Optional Input <GenerateUnderSignaturePolicy>, definierte Element wird die für die sichere Anzeige von XML-Dokumenten (DF_SV_XML) erforderliche Singnaturrichtlinie ausgewählt. Die im Element `sp:SignaturePolicyIdentifier` übergebene URI identifiziert die Signaturrichtlinie, welche die sichere Anzeige für das zu signierende XML-Dokument steuert. Mögliche Werte: • `urn:gematik:fa:sak:cda:r1:v1` `Signaturrichtlinie für CDA Release 2 – Dokumente als VHitG Arztbrief.` • `urn:gematik:fa:sak:eges:r1:v1` Signaturrichtlinie für eine eGesamtaufstellung der KVNO gemäß `gesamtaufstellung_arzt.xsd` • `urn:gematik:fa:sak:eges:r2:v1` Signaturrichtlinie für eine eGesamtaufstellung der KVNO gemäß `gesamtaufstellung_ermaechtigt.xsd` • `urn:gematik:fa:sak:eges:r3:v1` Signaturrichtlinie für eine eGesamtaufstellung der KVNO gemäß `gesamtaufstellung_ krankenhaus.xsd`
	`dss:Schemas`	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schematas übergeben werden, die zur Validierung der übergebenen XML-Dokumente verwendet werden können.

<table>
<tr><td></td><td colspan="2"> 
</td></tr>
<tr><td></td><td>dss:Schema</td><td>Dieses Element enthält ein XML-Schema zur Validierung des übergebenen XML-Dokuments. Das Attribut RefURI kennzeichnet dabei den Namensraum des XML-Schemas entsprechend [OASIS-DSS] (Abschnitt 2.8.5) und bestimmt damit das Format des XML-Dokuments.</td></tr>
<tr><td rowspan="4">SIG:TrustedViewerInfo</td><td>SIG:XsltStylesheets</td><td>Liste von XSLT-Stylesheets zur Aufbereitung des XML-Dokuments für die Anzeige. Hierbei MUSS das Haupt-Stylesheet als erstes Element übergeben werden.</td></tr>
<tr><td>CONN:XsltStylesheet</td><td>Dieses Element enthält ein base64-codiertes Stylesheet im CONN:Data-Element und eine das Stylesheet identifizierende URI im CONN:RefURI-Element.</td></tr>
<tr><td colspan="2">
</td></tr>
<tr><td></td><td></td></tr>
<tr><td>SIG:Document</td><td></td><td> 

Dieses an das dss:Document Element aus [OASIS-DSS] Section 2.4.2 angelehnte Element kann mehrmals auftreten</td></tr>
</table>

		und enthält die zu signierenden Dokumente, wobei die Kindelemente Base64XML und dss:Base64Data auftreten können. Das Attribut ShortText kann Metainformationen zum Dokument enthalten, die den Inhalt des Dokumentes beschreiben. ShortText dient dem Benutzer zur Identifikation des zu signierenden Dokumentes und muss gefüllt sein, wenn der xTV verwendet wird.

Tabelle 14: Rückgabe XML-Signatur

Rückgabe		
	`CONN:Status`	Enthält den Status der ausgeführten Operation.
	`SIG:SignResponse`	Eine `SignResponse` kapselt den ausgeführten Signaturauftrag pro Dokument. Die Zuordnung zwischen `SignRequest` und `SignResponse` erfolgt über die RequestID. Für Dokumente, die vom Benutzer durch Deselektion von der Signaturerzeugung ausgeschlossen wurden, wird ebenfalls ein Element SignResponse zurückgegeben.
	`SIG:OptionalOutputs`	Enthält (angelehnt an `dss:OptionalOutputs`) optionale Ausgangsparameter: 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.
	`vr:VerificationReport`	Dieses in [OASIS-VR] spezifizierte Element `ReturnVerificationReport` wird zurückgeliefert, falls bei einer Gegensignatur (QES oder nonQES) ein Problem mit der bereits vorhandenen Signatur des Dokumentes aufgetreten ist. Das Primärsystem muss den Nutzer über Probleme informieren, die bei der Prüfung von eingebetten Signaturen aufgetreten sind, die mit dem Dokument zusammen signiert wurden (Gegensignatur-Probleme).

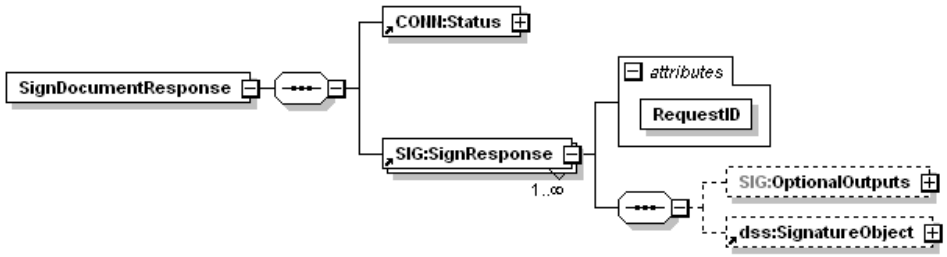
4.4.1.2 CMS - Signatur

Beim Erzeugen einer CMS-Signatur gemäß [RFC5652] wird als Default-Signaturverfahren eine Detached Signature erzeugt, wenn `SignDocument` nicht anderslautend parametrisiert wird.

Tabelle 15: Aufrufparameter speziell für die CMS-Signatur

Optionen zur Steuerung der CMS-Signatur		
`dss:OptionalInputs`	`dss:SignatureType`	Der Parameterwert `urn:ietf:rfc:5652` wählt eine CMS-Signatur gemäß [RFC5652] Das zu verwendende Profil ist CAdES-BES ([CAdES]). Die Signatur wird als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert.
	`dss:ReturnUpdated Signature`	Durch dieses in [OASIS-DSS#Abs.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/documentincluding Hierdurch wird eine dokumenteninkludierende Gegensignatur für das Dokument und alle vorhandenen parallelen Signaturen erzeugt. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt.
	`SIG:IncludeEContent`	Durch dieses in [OASIS-DSS#Abs.3.5.7] definierte Element kann bei einer CMS-basierten Signatur das Einfügen des signierten Dokumentes in die Signatur angefordert werden.
`SIG:Document`		Dieses in [OASIS-DSS] Section 2.4.2 spezifizierte Element kann mehrmals auftreten und enthält die zu signierenden Dokumente. Das Kindelement `dss:Base64Data` kann folgende (Klassen von) MIME-Types enthalten: • text/plain – 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 muss gesetzt werden. Das Attribut ShortText kann Metainformationen zum Dokument enthalten, die den Inhalt des Dokumentes beschreiben.

Tabelle 16: Rückgabe CMS-Signatur

Rückgabe		
	`CONN:Status`	Enthält den Status der ausgeführten Operation.
	`dss:SignatureObject`	Enthält im Erfolgsfall die erzeugten CMS-Signaturen in Form eines oder mehrerer `dss:SignatureObject`-Elemente gemäß [OASIS-DSS] (Abschnitt 3.2), wobei die Signatur als `dss:Base64Signature` mit der URI `urn:ietf:rfc:5652` als Type zurückgeliefert wird.

4.4.1.3 S/MIME - Signatur

Das Erzeugen einer S/MIME-Signatur gemäß [RFC5751] erfolgt entsprechend den Vorgaben der CMS-Signatur, wenn nicht wie im Folgenden beschrieben, Besonderheiten der S/MIME-Signatur zu verwenden sind.

Das Rückgabedokument ist eine MIME-Nachricht vom Typ „application/pkcs7-mime“ mit einer CMS-Struktur vom Typ_SignedData.

Tabelle 17: Aufrufparameter speziell für die S/MIME-Signatur

Optionen zur Steuerung der S/MIME-Signatur		
`dss:OptionalInputs`	dss:SignatureType	Der Parameterwert `urn:ietf:rfc:5751` wählt eine S/MIME-Signatur gemäß [RFC5751] für die übergebene MIME-Nachricht.
`SIG:Document`		Dieses in [OASIS-DSS] Section 2.4.2 spezifizierte Element kann mehrmals auftreten und enthält die zu signierenden Dokumente. Das Kindelement `dss:Base64Data` kann MIME-Types beliebiger MIME-Type für nicht näher unterschiedene Binärdaten enthalten. Der MIME-Type „text/plain“ wird interpretiert als „text/plain; charset=iso-8859-15”. Das Attribut ShortText soll Metainformationen zum Dokument enthalten, die den Inhalt des Dokumentes beschreiben und muss gefüllt sein, wenn eine Anzeige im xTV stattfinden soll.

4.4.1.4 PDF - Signatur

Die Signatur eines PDF erfordert keine zusätzlichen steuernden Parameter, sie wird ausschließlich gemäß [PAdES-2] in der Variante einer CMS-basierten Enveloped Signature (eingebetteten Signatur) umgesetzt (vgl.4.4.1.2).

Tabelle 18: Aufrufparameter speziell für die PDF-Signatur

Optionen zur Steuerung der PDF-Signatur		
`dss:OptionalInputs`	dss:SignatureType	Der Parameterwert http://uri.etsi.org/02778/3 wählt eine PadES-Basic Signatur gemäß [PadES-3], wobei das Dokument mit der integrierten Signatur als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert wird.

`SIG:Document`		Dieses in [OASIS-DSS] Section 2.4.2 spezifizierte Element kann mehrmals auftreten und enthält die zu signierenden Dokumente.
		Das Kindelement `dss:Base64Data` hat den MIME-Typ:
		`application/pdf-a`
		`dss:Document`
		Das Attribut ShortText soll Metainformationen zum Dokument enthalten, die den Inhalt des Dokumentes beschreiben und muss gefüllt sein, wenn eine Anzeige im xTV stattfinden soll.

Tabelle 19: Rückgabe PDF-Signatur

	`dss:SignatureObject`	Enthält im Erfolgsfall die erzeugten PDF-Signaturen in Form eines oder mehrerer `dss:SignatureObject`-Elemente gemäß [OASIS-DSS] (Abschnitt 3.2), wobei die Signatur als `dss:Base64Signature` mit der URI http://uri.etsi.org/02778/3 als `Type` zurückgeliefert wird.

4.4.1.5 PKCS#1 – Signatur (External Authenticate)

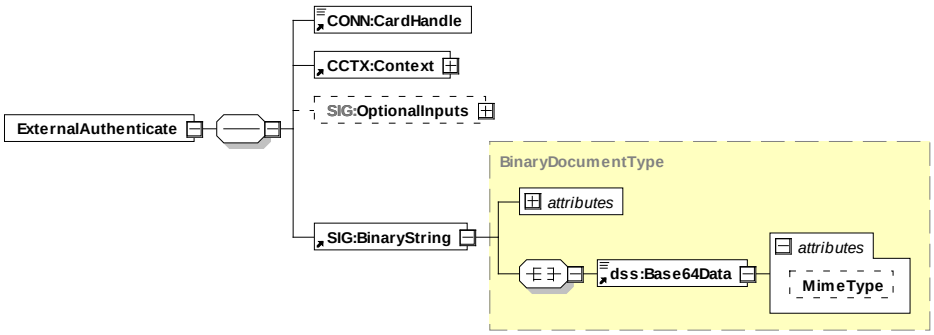
Eine PKCS#1 - Signatur gemäß [RFC2313] wird im Rahmen der Verschlüsselung von RSA Public-Key-Kryptosystemen verwendet. Die PKCS#1-Signatur darf nur in Verbindung mit dem Authentisierungszertifikat des HBAx genutzt werden (vgl. 4.4.1.2).

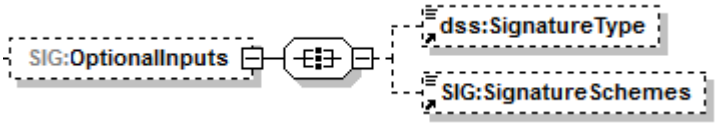
Zur Erstellung einer PKCS#1-Signatur bietet der Konnektor an seiner Außenschnittstelle die Operation `ExternalAuthenticate` an.

Tabelle 20: Aufrufparameter für die PKCS#1-Signatur, `ExternalAuthenticate`

Name	`ExternalAuthenticate`
Beschreibung	Diese Operation versieht einen Binärstring der maximalen Länge 512 Bit mit einer nicht-qualifizierten elektronischen Signatur (nonQES). Dazu wird das Signaturverfahren PKCS#1 verwendet. Das AUT-Zertifikat der SM-B und das AUT-Zertifikat des HBAx werden unterstützt.

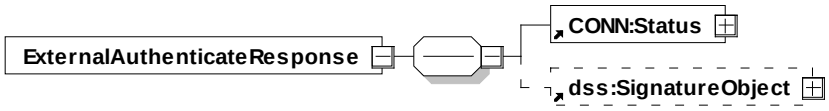
Aufrufparameter



Name	Beschreibung
`CONN:CardHandle`	Identifiziert die zu verwendende Signaturkarte. Die Operation unterstützt HBAx und SM-B.
`CCTX:Context`	Aufrufkontext für HBAx: MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend Aufrufkontext für SM-B: MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
`SIG:OptionalInputs`	Enthält optionale Eingangsparameter: 
`SIG:BinaryString`	Dieses Element enthält im Kindelemente `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.
`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element wird der Typ der zu erzeugenden Signaturen bestimmt. Als Signaturtyp wird die PKCS#1-Signatur unterstützt: 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. 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:SignatureSchemes`	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.

Tabelle 21: Rückgabe PKCS#1 - Signatur

Rückgabe		
	`CONN:Status`	Enthält den Status der ausgeführten Operation.

	`dss:SignatureObject`	Enthält im Erfolgsfall die erzeugte Signatur in Form eines `dss:SignatureObject`-Elemente gemäß [OASIS-DSS].
		Der Signaturwert wird im XML-Element `dss:SignatureObject/dss:Base64Signature` übergeben. Das XML-Attribut `dss:SignatureObject/dss:Base64Signature[@Type]` kennzeichnet durch den Wert urn:ietf:rfc:3447 den Signatur-Typ.
		Die XML-Elemente
		`dss:SignatureObject/ds:Signature`
		`dss:SignatureObject/dss:Timestamp`
		`dss:SignatureObject/dss:SignaturePtr`
		`dss:SignatureObject/dss:Other`
		werden nicht verwendet.

4.4.1.6 Nicht-qualifizierte elektronische Signatur

Das Primärsystem löst eine Signatur durch Übergabe der Kartensitzung, des Dokumentes bzw. des Dokumentenstapels, sowie einiger formatabhängiger Detailfestlegungen aus.

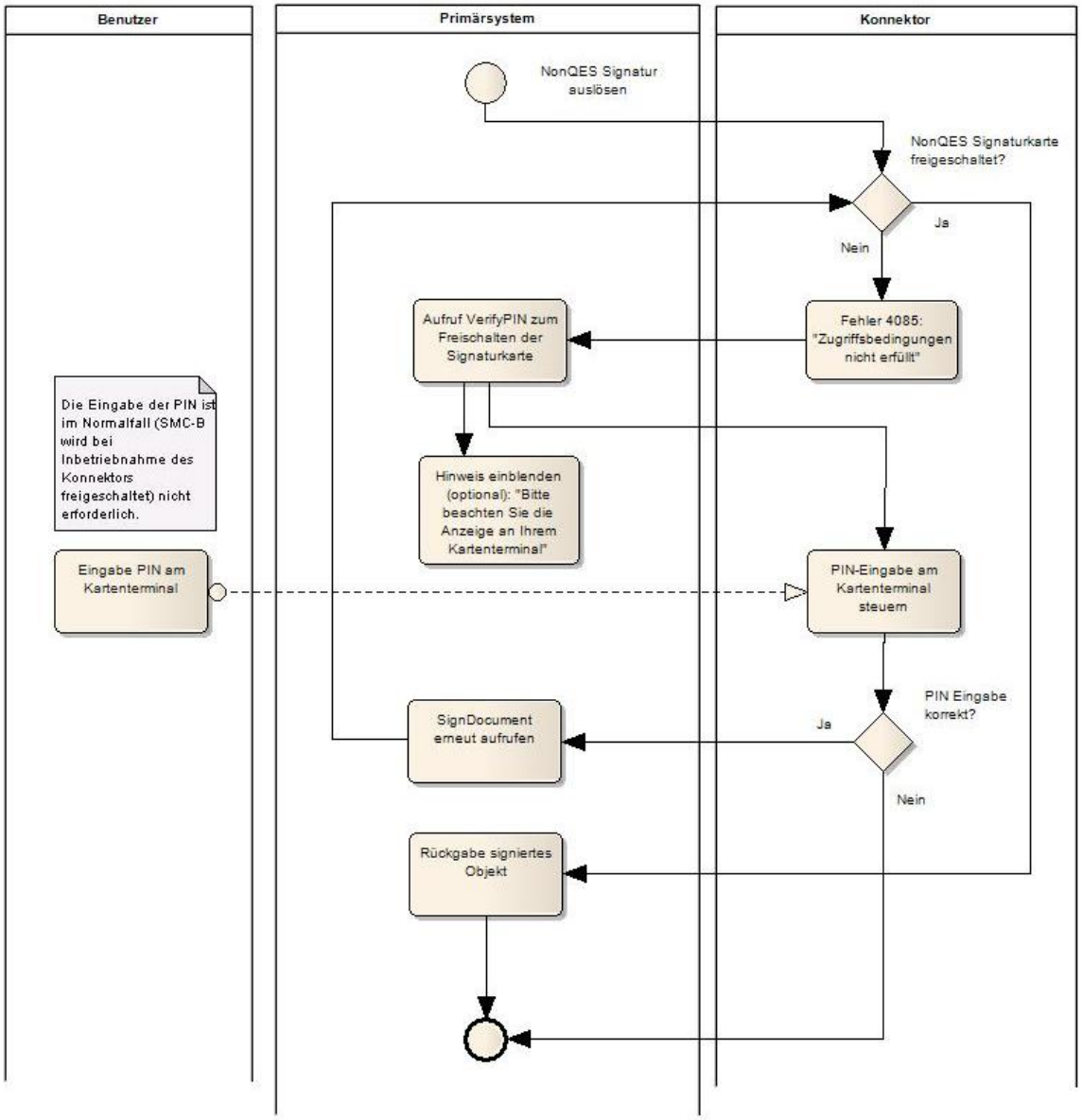


Abbildung 25: Subprozess nonQES Signatur auslösen[9]

Tabelle 22: Ablauf Signaturerzeugung nonQES - Signatur

Nr.	Operation	Beschreibung
1.	Dokumentenstapel bilden	Auswahl von einem oder mehreren zu signierenden Dokumenten der Dokumententypen XML, PDF/A, Text, TIFF, MIME oder Binär inklusive der zum jeweiligen Dokument gehörigen Kurztexte (`ShortText`), z.B. Dokumentennamen.
2.	HBAx oder SM-B auswählen	Zur Nutzung des SignatureService ist der Aufbau einer Kartensitzung zu einer Signaturkarte erforderlich. Mit `getCardSession` kann die Signaturkarte ausgewählt

[9] Der abgebildete Ablauf setzt voraus, dass der Konfigurationsparameter `TvMode` auf none gesetzt wurde.

Nr.	Operation	Beschreibung
		werden.
3.	Operation SignDocument aufrufen	Funktionsaufruf unter Angabe der Parameter Zertifikatsreferenz, Signature-Type, Kurztext (`ShortText`) usw. laut Schnittstellenspezifikation[10]
4.	Einsatz des xTV	Interaktion mit dem xTV je nach Konfiguration von TvMode: `Confirmed`: Der xTV liefert ausführliche Informationen zu den signierten Dokumenten sowie zur Signatur. Eine Bestätigung durch den Benutzer ist nicht erforderlich, die Anzeige ist rein informativ. `Unconfirmed`: Der xTV liefert Basisinformationen zum Signaturvorgang `None`: Der xTV kommt nicht zum Einsatz (Szenario wie in Abbildung 25: Subprozess nonQES Signatur auslösen)
5.	PIN-Eingabe	Eine PIN-Eingabe ist nicht erforderlich, wenn die SM-B sich bereits in einem geeigneten Sicherheitszustand vorliegt. Andernfalls tritt der Fehler 4085 auf, den das Primärsystem abfangen muss, um das OSIG-Zertifikat der SM-B mit der PIN.SMC unter Verwendung von `VerifyPIN` freizuschalten. Wenn die PIN.SMC freigeschaltet ist, lässt sich der erhöhte Sicherheitszustand in weiteren Kartensitzungen nachnutzen. Der Sicherheitszustand bleibt solange bestehen, bis die Karte gezogen wird oder ein andersartiger Verbindungsabbruch eintritt.
6.	Ergebnisvalidierung	Rückgabewerte und `Status` prüfen. Prüfen, ob in der Rückgabe der `SignedDocumentList` alle Dokumente enthalten sind, die zur Signatur vorgesehen waren.

4.4.1.7 Qualifizierte elektronische Signatur

Zur Auslösung der QES kann die SM-B nicht verwendet werden und es können die Dokumententypen Binär und MIME nicht qualifiziert signiert werden.

Das `Context`–Element muss dabei im Falle einer QES-Signatur eine `userID` enthalten, die einen eindeutigen Bezug auf den Nutzer enthält, der die Signatur auslöst.

Beispiel 14: Beispiel qualifizierte CMS – Signatur auf einem Text-Dokument

```
...
<SIG:SignDocument
xsi:schemaLocation="http://ws.gematik.de/conn/SignatureService/v7.0
SignatureService.xsd"
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:SIG="http://ws.gematik.de/conn/SignatureService/v7.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
      <CONN:CardHandle>c123456789123456789</CONN:CardHandle>
      <CCTX:Context>
            <CONN:MandantId>m0001</CONN:MandantId>
```

[10] [gemSpec_Kon#4.1.8.5.1]

```
              <CONN:ClientSystemId>cs0001</CONN:ClientSystemId>
              <CONN:WorkplaceId>wp007</CONN:WorkplaceId>
              <CONN:UserId>u0001</CONN:UserId>
       </CCTX:Context>
       <SIG:OptionalInputs>
              <dss:SignatureType>urn:ietf:rfc:5652</dss:SignatureType>
       <SIG:SignatureForm>urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:BES</SIG
:SignatureForm>
              <SIG:TrustedViewerInfo>
                     <SIG:TvMode>CONFIRMED</SIG:TvMode>
              </SIG:TrustedViewerInfo>
       </SIG:OptionalInputs>
       <dss:Document>
       <dss:Base64Data
MimeType="text/plain">VHJpbmtlIGRpY2ggc2F0dCBpbiBkZWluZXIgVmFzZSwgcnVoZSBzYW5mdC
wga2xlaW5lIEFzdGVyIQ==</dss:Base64Data>
       </dss:Document>
</SIG:SignDocument>
...
```

Das PS kann Dokumente über den SignatureService des Konnektors qualifiziert signieren, unabhängig vom Szenario (Online-Szenario, Standalone-Szenario mit Online- und Offline-Konnektor, Szenario mit logischer Trennung). Wenn eine OCSP-Anfrage Online durchgeführt werden kann, kann das Ergebnis in die Signatur eingebettet werden, so dass beim Verifizieren bekannt ist, dass das benutzte Zertifikat zum Zeitpunkt der Erstellung gültig war. Das Erstellen einer QES ist ansonsten auch ohne OCSP-Anfrage möglich.

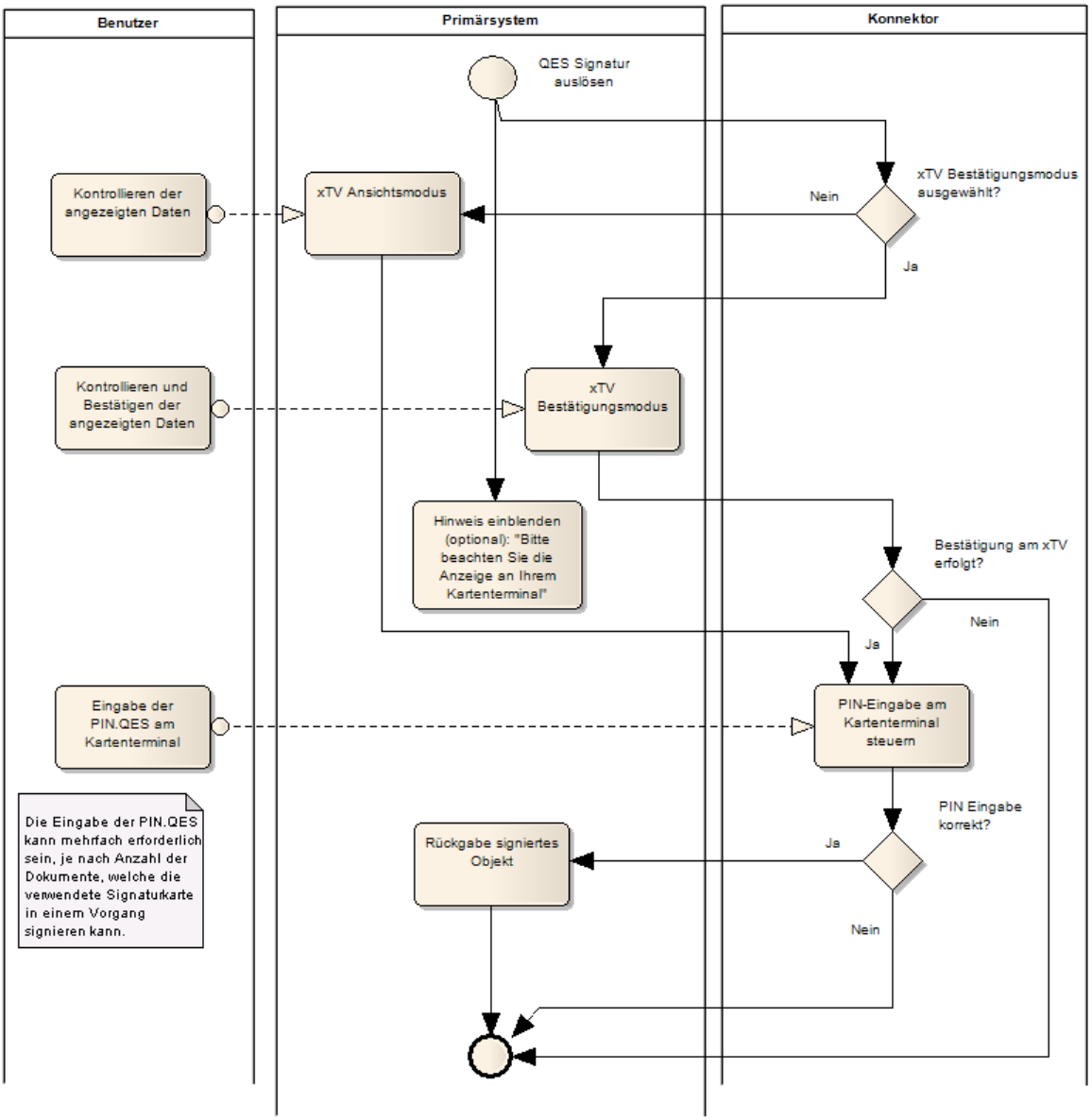


Abbildung 26: Subprozess QES Signatur auslösen

Tabelle 23: Ablauf Signaturerzeugung

Nr.	Operation	Beschreibung
1.	Dokumentenstapel bilden	Auswahl von einem oder mehreren zu signierenden Dokumenten der Dokumententypen XML, PDF/A, Text oder TIFF inklusive der zum jeweiligen Dokument gehörigen Kurztexte (`ShortText`), z.B. Dokumentennamen.
2.	HBAx auswählen	Kartensitzung zum HBAx aufbauen. `getCardSession` wählt die Signaturkarte aus.
3.	Operation SignDocument aufrufen	Funktionsaufruf unter Angabe der Parameter Zertifikatsreferenz, Signature-Type, usw. laut Schnittstellenspezifikation[11]

[11] [gemSpec_Kon#4.1.8.5.1]

Nr.	Operation	Beschreibung
4.	Benutzer-Interaktion am xTV	Die Anzeige des xTV kann vom Primärsystem je nach Übergabewert TvMode konfiguriert werden `Confirmed`: Der xTV liefert ausführliche Informationen zu den signierten Dokumenten, sowie zur Signatur. Eine Bestätigung des Vorgangs durch den Benutzer ist erforderlich. Die Benutzer können Dokumente deselektieren, um sie von der Signatur auszuschließen. `Unconfirmed`: Der xTV liefert Basisinformationen zum Signaturvorgang. Eine Bestätigung des Vorgangs ist nicht möglich.
5.	PIN-Eingabe	Der Benutzer muss einmal oder ggf. mehrfach seine Signatur-PIN.QES eingeben.
6.	Ergebnisvalidierung	Rückgabewerte und `Status` prüfen. Prüfen, ob in der Rückgabe der `SignedDocumentList` alle Dokumente enthalten sind, die zur Signatur vorgesehen waren.

Mit dem (optionalen) Einblenden eines Hinweises der Form "Bitte beachten Sie die Anzeige an Ihrem Kartenterminal" kann das Primärsystem dafür sorgen, dass die Abfrage einer PIN-Eingabe am Kartenterminal vom Benutzer nicht übersehen wird.

4.4.2 Verifizieren digitaler Signaturen

Das Primärsystem muss es dem Benutzer ermöglichen, `VerifyDocument` mit Stapeln von Dokumenten der Dokumententypen XML, PDF/A, Text, TIFF, MIME aufzurufen, die jeweils nicht größer sind als 25 MB.

Zusätzlich kann `VerifyDocument` aufgerufen werden, um Signaturen im Format PKCS#1 (V2.1) gemäß [RFC3447] zu prüfen.

Die Verifikation qualifizierter und nicht-qualifizierter Signaturen unterscheidet sich aus Sicht der Primärsysteme nicht.

Wenn über den Konnektor im Verifikationsprozess keine OCSP-Abfrage durchgeführt werden kann, wird dies im Ergebnis der Verifikation vermerkt.[12]

Die vollständige und kanonische Darstellung der Schnittstelle zum Verifizieren digitaler Signaturen findet sich in [gemSpec_Kon#4.1.8.5.2].

Tabelle 24: Ablauf Verifizieren digitaler Signaturen

Nr.	Operation	Beschreibung
1.	Dokumente auswählen	Auswahl signierter Dokumente vom Typ XML, PDF/A, Text TIFF, S/MIME inklusive der zum jeweiligen Dokument gehörigen Kurztexte (`ShortText`), z.B. Dokumentennamen.
2.	Operation `VerifyDocument` aufrufen	Funktionsaufruf `VerifyDocument` laut Schnittstellenspezifikation[13] unter Angabe des Dokumententyps (s.u.)

[12] Eine scheiternde OCSP-Anfrage, etwa bei Verwendung eines Offline-Konnektors, ist kein Fehlerfall.

Nr.	Operation	Beschreibung
3.	Prüf-Ergebnis weiterverarbeiten	Entgegennehmen und Weiterverarbeiten des standardisierten Prüfberichts in einer `VerificationReport`-Struktur gemäß [OASIS-VR] und ggf. Anzeigen des Verifikationsergebnisses am xTV.

Das PS ruft die Verifikationsschnittstelle unter Angabe des signierten Dokumentes, des Dokumententyps, sowie einiger formatabhängiger Detailfestlegungen auf. Je nach Dokumententyp müssen ggf. Schemadateien oder XSLT-Dateien oder entsprechende Referenzen übergeben werden, um über den xTV anzeigen zu können, was signiert wurde:

- Für Dokumententypen, die dem Konnektor bekannt sind ([CDA], [eGES], [eAB]): Angabe der passenden RefURI
- Für andere Dokumententypen müssen ggf. XML-Schemata und ggf. XSLT-Dateien mitgeliefert werden.

Tabelle 25: Aufrufparameter für `VerifyDocument`

Optionen zur Steuerung von `VerifyDocument`		
	`CCTX:Context`	MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId wird nicht ausgewertet
	`TvMode`	Der optionale Parameter legt das Verhalten des Extended Trusted Viewers fest. Erlaubter Wert für TvMode bei QES und nonQES: UNCONFIRMED (Ansichtsmodus): Dem Benutzer wird das Ergebnis der Signaturprüfung angezeigt. Der Benutzer kann sich den Inhalt des signierten Dokuments und des Zertifikats im Extended Trusted Viewer anzeigen lassen. Zusätzlich erlaubter Wert für TvMode bei nonQES: NONE Keine Anzeige im Extended Trusted Viewer. (siehe weitere Anzeige gemäß TIP1-A_4673 xTV: Anzeige verpflichtender Parameter bei Signaturprüfung und Beschreibung Parameter TvMode bei Operation SignDocument). default: UNCONFIRMED
	`SIG:Document`	Enthält im Fall der Prüfung von detached oder enveloped Signaturen die zur Signatur gehörenden bzw. diese umschließenden Dokumente (siehe [OASIS-DSS] Section 2.4.2) inklusive der zum jeweiligen Dokument gehörigen Kurztexte (`ShortText`), z.B. Dokumentennamen.
	`dss:SignatureObject`	Enthält die zu prüfenden Signaturen bei nicht eingebetteten Signaturen. Hierbei werden XML-Signaturen als `ds:Signature` Element und alle anderen Signaturen als `dss:Base64Signature` mit entsprechend gesetztem Type-Attribut (`SignatureType`) übergeben. (Zu Details siehe

[13] [gemSpec_Kon#4.1.8.5.2]

Optionen zur Steuerung von `VerifyDocument`		
		[OASIS-DSS], Kapitel 4.1)
	`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. Im Falle der boolschen Belegung `true` bettet der Konnektor die Sperrinformationen ein, andernfalls nicht.
`dss:OptionalInputs`	`SIG:VerifyManifests`	Durch das in [OASIS-DSS] (Abschnitt 4.5.1) definierte Element kann die Prüfung eines ggf. vorhandenen Manifests angefordert werden.
	`SIG:UseVerificationTime`	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:AdditionalKeyInfo`	Durch das in [OASIS-DSS] (Abschnitt 4.5.4) spezifizierte Element kann zusätzliches, für die Prüfung benötigtes, Schlüsselmaterial übergeben werden.
	`vr:ReturnVerificationReport`	Durch dieses in [OASIS-VR] spezifizierte Element kann die Erstellung eines ausführlichen Prüfberichtes angefordert werden, der mindestens der Konformitätsstufe 2 („Comprehensive“) entspricht
	`dss:Schemas`	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element (vgl. auch 4.4.1.1) können eine Menge von XML-Schematas übergeben werden, die zur Validierung der übergebenen XML-Dokumente verwendet werden können.
	`SIG:TrustedViewerInfo`	Enthält Informationen zur Ansteuerung des Trusted Viewers und legt das Verhalten des Extended Trusted Viewers fest. <u>Erlaubte Werte für TvMode bei QES und nonQES</u>: `CONFIRMED` (Bestätigungsmodus): Der Benutzer kann sich den Inhalt des signierten Dokumentes anzeigen lassen. `UNCONFIRMED` (Ansichtsmodus) <u>Zusätzlich erlaubte Werte für TvMode bei nonQES</u>: `NONE` Keine Anzeige im Extended Trusted Viewer. (siehe auch Beschreibung Parameter TvMode bei Operation SignedDocument).

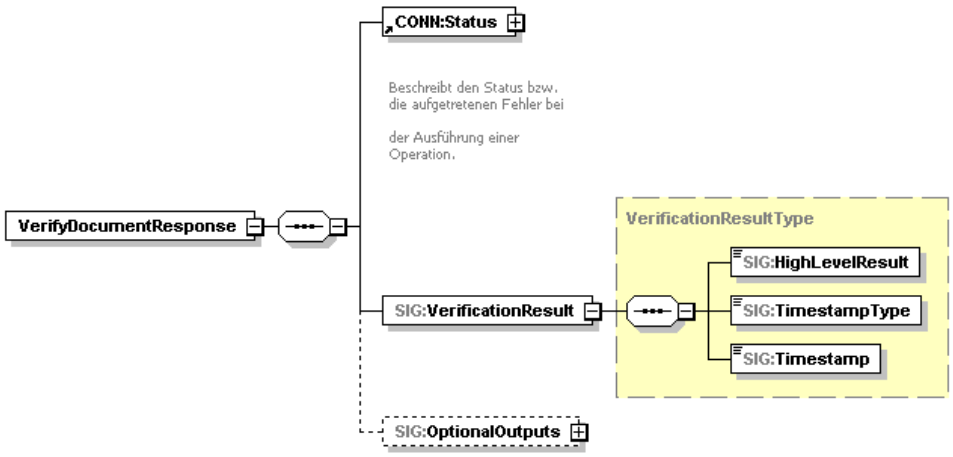
Das Feld `SIG:IncludeRevocationInfo` soll durch eine Konfigurationseinstellung im Primärsystem standardmäßig mit dem Wert `true` oder `false` belegt werden, so dass nicht der Nutzer in jedem Einzelfall über die Belegung des Wertes entscheiden muss. Da schon bei der Signaturerzeugung der Sperrstatus eingebettet wurde, und so die Gültigkeit zum Zeitpunkt der Erstellung bekannt sein sollte, kann eine erneute Überprüfung des Sperrstatus zum Zeitpunkt der Verifikation entfallen.

Bei der Signaturprüfung von PKCS#1 – Signaturen müssen abweichend von den oben genannten Parameterstrecken der anderen Dokumententypen folgende Werte clientseitig gefüllt werden:

Tabelle 26: Parameter `VerifyDocument im Spezialfall PKCS#1-Signatur`

Optionen zur Steuerung von `VerifyDocument im Spezialfall PKCS#1`		
Signaturverfahren	`VerifyDokument/dss:SignatureObject/dss:Base64Signature/@Type`	„urn:ietf:rfc:3447“ (PKCS#1-Signatur)
Signaturwert	`VerifyDokument/dss:SignatureObject/dss:Base64Signature`	Übergabe der PKCS#1-Signatur
Message	`VerifyDokument/SIG:Document/dss:Base64Data`	Übergabe der signierten Daten
Zertifikat	`VerifyDokument/SIG:OptionalInputs/dss:AdditinalKeyInfo/dss:KeyInfo/ds:X509Data/dss:X509Certificate`	Übergabe des Zertifikates

Tabelle 27: Rückgabe `VerifyDocument`

Rückgabe		
	`Status`	Enthält den Ausführungsstatus der Operation.
	`HighLevelResult`	Wertebereich: - „VALID" - "INCONCLUSIVE" - "INVALID" „INCONCLUSIVE “ bedeutet, dass bei unvollständiger Prüfung keine eindeutige Schlussfolgerung über die Gültigkeit der Signatur gezogen werden kann.
`dss:OptionalOutputs`	`dss:VerifyManifestResults`	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:DocumentWithSignature`	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:UpdatedSignature`	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:VerificationReport`	Dieses in [OASIS-VR] spezifizierte Element wird zurückgeliefert, falls das `ReturnVerificationReport`-Element als Eingabeparameter verwendet wurde.

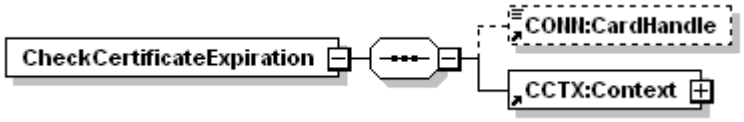
4.4.3 Zertifikatsdienst

Der `CertificateService` des Konnektors bietet Operationen zum Abfragen von Kartenzertifikaten und ihrer Gültigkeit an.

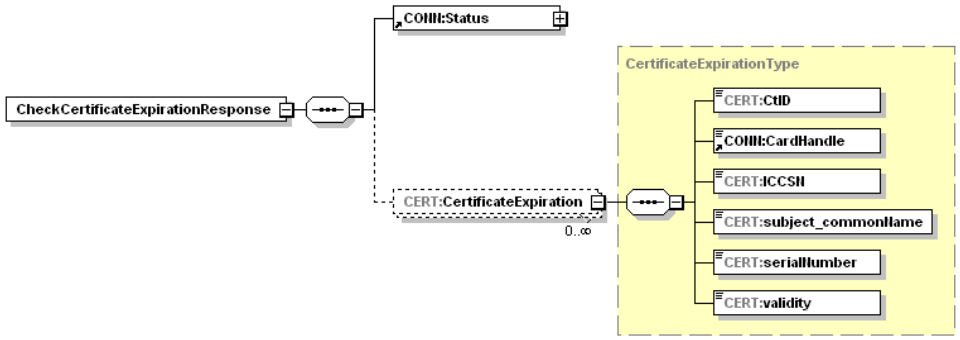
4.4.3.1 Ablaufdatum von Zertifikaten prüfen

Die Operation `CheckCertificateExpiration` kann dazu verwendet werden, die Gültigkeitsdauer von Zertifikaten zu überprüfen, um ablaufende Zertifikate zu identifizieren. Damit kann der Nutzer auf ein Zertifikat aufmerksam gemacht werden, dessen Gültigkeit abgelaufen ist.

Tabelle 28: Operation CheckCertificateExpiration[14]

Name	`CheckCertificateExpiration`	
Beschreibung	Gibt das Datum des Ablaufs eines bestimmten Zertifikats oder gesammelt die Ablaufdaten der Zertifikate aller zu einem Mandanten im Konnektor registrierten Karten zurück.	
Aufrufparameter		
	Name	**Beschreibung**
	`CardHandle`	Optional. Identifiziert eine einzelne Karte. Wenn keine bestimmte Karte identifiziert wird, werden alle Karten ausgewertet, die für einen Mandanten zum Zeitpunkt des Funktionsaufrufes im Konnektor registriert sind.
	`Context`	MandantId, CsId, WorkplaceId verpflichtend; UserId optional

[14] [gemSpec_Kon#4.1.9.5.1]

Rückgabe		
	`Status`	Enthält den Ausführungsstatus der Operation.
	`CertificateExpiration`	Eine Liste von Tupeln aus (CtID, CardHandle, ICCSN, subject.CommonName, serialNumber, validity) der Zertifikate der Karten.

Beispiel 15: Ablaufdatum von Zertifikaten auslesen

```
...
<CERT:CheckCertificateExpiration
xsi:schemaLocation="http://ws.gematik.de/conn/CertificateService/v6.0
CertificateService.xsd"
xmlns:CERT="http://ws.gematik.de/conn/CertificateService/v6.0"
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <CONN:CardHandle>c123456789123456789</CONN:CardHandle>
       <CCTX:Context>
              <CONN:MandantId>m0001</CONN:MandantId>
              <CONN:ClientSystemId>cs0001</CONN:ClientSystemId>
              <CONN:WorkplaceId>wp007</CONN:WorkplaceId>
              <CONN:UserId>u0001</CONN:UserId>
       </CCTX:Context>
</CERT:CheckCertificateExpiration>
...
```

4.4.3.2 Kartenzertifikat lesen

Das Auslesen von Kartenzertifikaten ermöglicht Clientsystemen eine Reihe von Optionen, darunter:

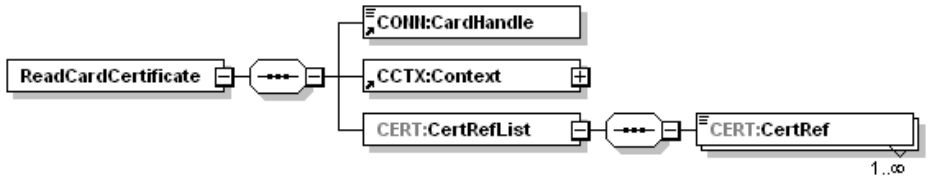
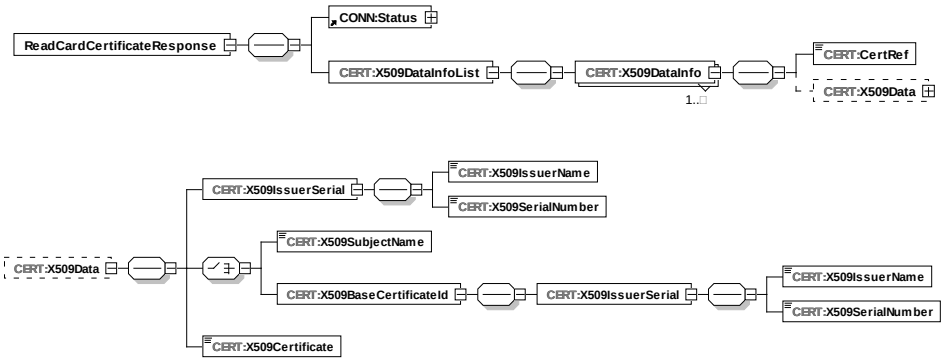
- Auslesen von Attributzertifikaten, um eine anwendungsfallbezogene Auswahl unter den Attributzertifikaten vorzunehmen, die im `SignDocument` – Request mitgegeben werden, je nachdem, welche Attributzertifikate für das signierte Dokument passend sind.
- Auslesen des öffentlichen Verschlüsselungsschlüssels, um beim Aufruf von `EncryptDocument` das ENC-Zertifikat mitzuliefern.

Die Operation `ReadCardCertificate` liest folgende Zertifikate aus:

- `C.AUT` (Authentisierungszertifikat, HBAx, SM-B)

- C.ENC (Verschlüsselungszertifikat, HBAx, SM-B)
- C.SIG (nicht-qualifiziertes Signaturzertifikat, SM-B)
- C.QES (qualifiziertes Signaturzertifikat HBAx, zusätzlich: die Attributzertifikate C.QES-AC1, C.QES-AC2, C.QES-AC3)

Tabelle 29: Operation ReadCardCertificate[15]

Name	ReadCardCertificate		
Beschreibung	Liest ein X.509-Zertifikat von einer Karte.		
Aufrufparameter			
	Name	**Beschreibung**	
	CardHandle	Gibt die Karte an, von der das Zertifikat gelesen werden soll. Es können Zertifikate von HBAx, SM-B ausgelesen werden.	
	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, C.QES-AC1, C.QES-AC2, C.QES-AC3	
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

[15] [gemSpec_Kon#4.1.9.5.2]

			etc.)
		`X509SerialNumber`	Enthält die serialNumber des Zertifikats.
		`X509SubjectName`	Enthält das Feld subject.CommonName des Zertifikats. Bezüglich des Encodings sind die in [XML DSIG#4.4.4.4.1] angegebenen Regeln zu beachten (Escaping von Sonderzeichen etc.).
		`X509BaseCertificateId`	Enthält im Falle eines Attributzertifikats Issuer-Name und Seriennummer des Basiszertifikates, falls vorhanden.
		`X509Certificate`	Enthält das base64-codierte Zertifikat, dessen Binärstruktur wiederum ASN.1-codiert (gemäß [COMMON_PKI]) vorliegt.

Beispiel 16: Beispiel Lesen des C.AUT Zertifikates

```
...
<CERT:ReadCardCertificate
xsi:schemaLocation="http://ws.gematik.de/conn/CertificateService/v6.0
CertificateService.xsd"
xmlns:CERT="http://ws.gematik.de/conn/CertificateService/v6.0"
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <CONN:CardHandle>c123456789123456789</CONN:CardHandle>
       <CCTX:Context>
              <CONN:MandantId>m0001</CONN:MandantId>
              <CONN:ClientSystemId>cs0001</CONN:ClientSystemId>
              <CONN:WorkplaceId>wp007</CONN:WorkplaceId>
              <CONN:UserId>u0001</CONN:UserId>
       </CCTX:Context>
       <CERT:CertRefList>
              <CERT:CertRef>C.QES</CERT:CertRef>
       </CERT:CertRefList>
</CERT:ReadCardCertificate>
...
```

4.4.3.3 Zertifikate verifizieren

Das Primärsystem muss es Nutzern ermöglichen, X.509-Zertifikate über die Konnektorschnittstelle `VerifyCertificate` zu verifizieren. Unterstützt werden X.509-Zertifikate von SM-B und HBAx.

Die vollständige und kanonische Darstellung der Schnittstelle zum Verifizieren von Zertifikaten findet sich in [gemSpec_Kon#4.1.9.5.3].

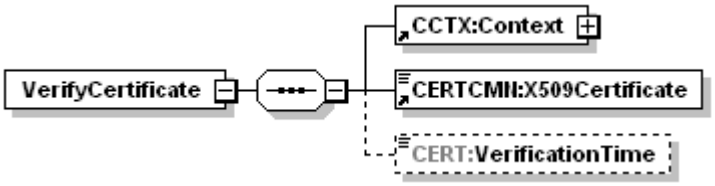
Tabelle 30: Ablauf Verifizieren von Zertifikaten

Nr.	Operation	Beschreibung
1.	Zertifikate auswählen	Auswahl von Zertifikaten, insbesondere von Zertifikaten, die zuvor über den Konnektor von der SM-B oder dem HBAx gelesen wurde (4.4.3.2).

Nr.	Operation	Beschreibung
2.	Operation `VerifyCertificate` aufrufen	Funktionsaufruf `VerifyCertificate` laut Schnittstellenspezifikation unter Angabe des Zertifikates (s.u.)
3.	Prüf-Ergebnis entgegennehmen	Entgegennehmen und Weiterverarbeiten der Prüfungsergebnisse • valid • valid with qualifications • invalid

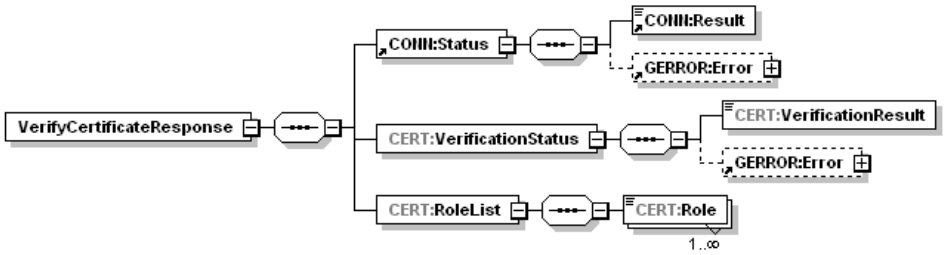
Das PS ruft die Verifikationsschnittstelle des Zertifikates unter Angabe des Mandanten, des Zertifikates, sowie optional einer Referenzzeit. Falls keine Referenzzeit übergeben wird, verwendet der Konnektor seine Systemzeit.

Tabelle 31: Aufrufparameter für `VerifyCertifikate`

Optionen zur Steuerung von `VerifySignature`		
Aufrufparameter		
	`CCTX:Context`	Aufrufkontext (Mandant)
	`CERTCMN: X509Certificate`	Zu prüfendes Zertifikat (base64 kodiert), wie in Response zur Operation ReadCardCertificate enthalten.
	`CERT: VerificationTime`	Der für die Prüfung zu verwendende Referenzzeitpunkt. Falls der Parameter nicht angegeben ist, wird als Referenzzeitpunkt die Systemzeit verwendet.

Der Konnektor verifiziert die X.509 – Zertifikate u.a. auch gegen den Vertrauensraum der TSL und liefert als Ergebnis Statusinformationen und Identifier der in den Zertifikaten enthaltenen Rollen.

Tabelle 32: Rückgabeparameter von `VerifyCertifikate`

Rückgabe		
	`CONN:Status`	Enthält den Ausführungsstatus der Operation.
	`CERT: VerificationStatus`	Enthält eines der drei möglichen Prüfungsergebnisse in CERT:VerificationResult • valid

		• valid with qualifications • invalid sowie weiter Details zu den Zuständen „valid with qualifications“ und „invalid“ in GERROR:Error.
	`CERT:RoleList`	OIDs der im Zertifikat gespeicherten Rollen.

4.4.4 Verschlüsselung

Der `EncryptionService` des Konnektors stellt Operationen zur kartenbasierten Hybridverschlüsselung sowie zur Entschlüsselung hybrid verschlüsselter Daten bereit.

Die Dokumentenformate XML, PDF/A, TIFF, MIME Text oder Binär können vom `EncryptionService` verarbeitet werden. Der Konnektor bietet die hybride und symmetrische Ver- und Entschlüsselung nach dem Cryptographic Message Syntax (CMS) Standard an [RFC5652].

Hybride Verschlüsselung wird nur für X.509-Zertifikate angeboten.

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]

Wenn XML-Dokumente ver- und entschlüsselt werden, können mit einer XPath-Angabe gezielt XML-Nodes angesteuert werden, die ver- bzw. entschlüsselt werden.

CMS wird gemäß [gemSpec_Kon#4.1.7] profiliert.

Zur Nutzung des Verschlüsselungsdienstes ist eine Kartensitzung mit der verwendeten Karte erforderlich. Der Konnektor unterstützt zur Verschlüsselung die Kartentypen HBAx und SM-B, nicht aber die eGK.

Tabelle 33: KeyReference im `EncryptionService`

Karte	KeyReference
HBAx	C.ENC
SM-B	C.ENC

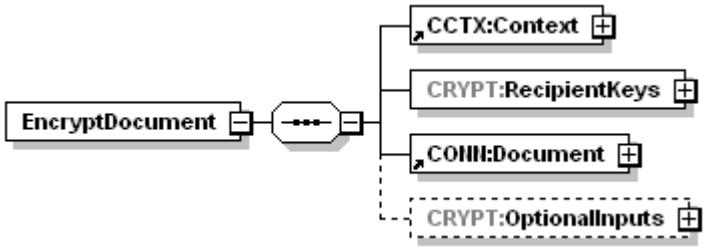
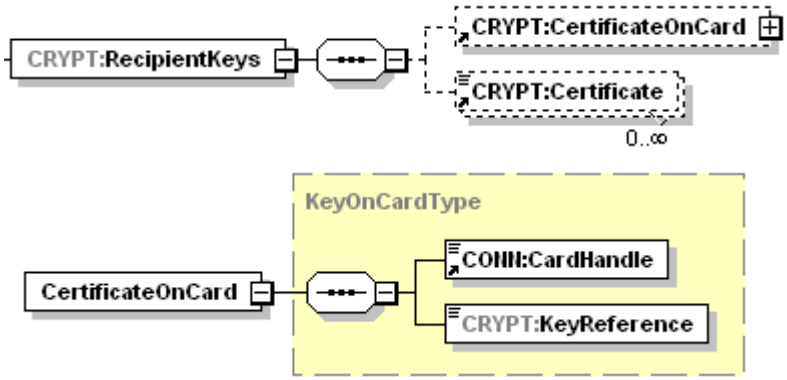
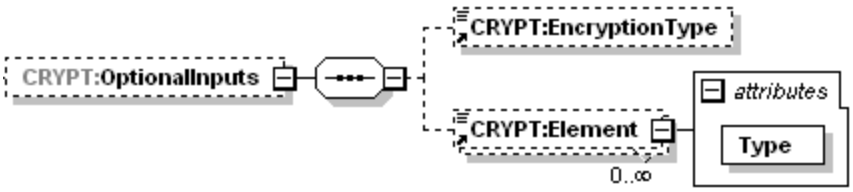
4.4.4.1 Verschlüsseln

Durch `EncryptDocument` wird ein Dokument hybrid für öffentliche Verschlüsselungs-schlüssel verschlüsselt. Die Verschlüsselungsschnittstelle des Konnektors ist für die Nutzung von Schlüsselmaterial konzipiert, das aus dem Vertrauensraum der TI stammt. Für die Nutzung der Verschlüsselungsfunktion des Konnektors, etwa für Szenarien, in denen Dokumente für Kommunikationspartner verschlüsselt werden, wäre es nützlich, wenn das Primärsystem einen Zertifikatsspeicher nutzt, der die öffentlichen Verschlüs-

selungsschlüssel zur Übergabe an den Konnektor enthalten kann. Daneben kann das Primärsystem, geeignete Zertifikate aus öffentlichen Verzeichnisdiensten entnehmen, falls solche zur Verfügung stehen.

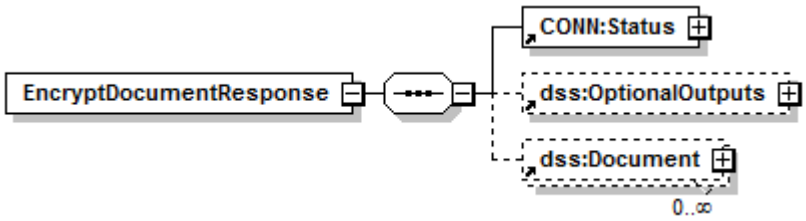
Die vollständige Beschreibung der Verschlüsselungsschnittstelle ist in [gemSpec_Kon#4.1.7.5] zu finden.

Tabelle 34: Operation EncryptDocument[16]

Name	`EncryptDocument`	
Beschreibung	Diese Operation verschlüsselt übergebene Dokumente hybrid. 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, indem mehrere Zertifikate übergeben oder referenziert werden. Es werden die folgenden Karten unterstützt: HBAx und SM-B. Bei XML-Dokumenten werden ein oder mehrere XML-Elemente des Dokumentes verschlüsselt. Für alle übrigen Dokumenttypen wird immer das gesamte Dokument verschlüsselt.	
Aufrufparameter		
		
		
	`CardHandle`	Identifiziert die zu verwendende Karte mit dem (öffentlichen) Schlüssel. Ist das Element nicht vorhanden, so werden nur Zertifikate

[16] [gemSpec_Kon#4.1.7.5.1]

		per Element `Certificate` übergeben.
	`KeyReference`	Wert `C.ENC` referenziert auf die KeyReference der HBAx und SM-B. Ist der Parameter nicht angegeben oder leer, gilt der Default-Wert `C.ENC`.
	`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.
	`EncryptionType`	Zu wählendes Verschlüsselungsverfahren, wobei folgende URI vorgesehen sind: • XMLEnc: „http://www.w3.org/TR/xmlenc-core1/" • 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 oder ein XML-Dokument im Element `CONN:Document/CONN:Base64XML`. 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. Im Fall S/MIME ist das in dss:Document:dssBase64Data übergebene Dokument eine MIME-Nachricht.
	`Document`	Dieses entsprechend [OASIS-DSS] Section 2.4.2 spezifizierte Element enthält die zu verschlüsselnden Dokumente, wobei die Kindelemente dss:`Base64XML` und `dss:Base64Data` verwendet werden. Im Fall `dss:Base64Data` wird ein etwaig übergebenes MIME-Type-Attribut nicht ausgewertet.
	`Element`	Dieses möglicherweise mehrfach auftretende Element ist nur relevant für XML-Dokumente. XPath Ausdruck, der das Element ermittelt, welches verschlüsselt werden soll. Der Ausdruck darf nur ein Element-Node des XML-Dokumentes als Ergebnis liefern. Dieses Element wird verschlüsselt. Das XML-Attribut Type kann einen der Werte `http://www.w3.org/2001/04/xmlenc#Element`

		`http://www.w3.org/2001/04/xmlenc#Content` annehmen. Gemäß XMLEnc steuert der Parameter, ob das gesamte Element oder nur sein Content verschlüsselt wird. Wird der Parameter weggelassen, so wird das Root-Element, d. h. das gesamte Dokument verschlüsselt. In diesem Fall ist Type `http://www.w3.org/2001/04/xmlenc#Element` anzusetzen. Sind mehrere Elemente angegeben, so darf keines der Elemente unter den angegebenen Elementen Vorfahren haben, was sicherstellt, dass keine zu signierenden Dokumententeile überlappen.
Rückgabe		
	`Status`	Enthält den Ausführungsstatus der Operation.
	`dss:OptionalOutputs`	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/dssBase64Data` zurückgegeben. Im Fall S/MIME wird die Base64-codierte S/MIME-Nachricht im Element `CONN:Document/dssBase64Data` zurückgegeben. Das Attribut `CONN:Document/dssBase64Data[@MimeType]` wird auf „application/pkcs7-mime“ gesetzt. Die S/MIME-Nachricht hat Content-Transfer-Encoding: base64.

Beispiel 17: Beispiel Verschlüsseln eines Textes mit einem C.ENC Schlüssel

```
...
<CRYPT:EncryptDocument
xsi:schemaLocation="http://ws.gematik.de/conn/EncryptionService/v6.0
EncryptionService.xsd"
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:CRYPT="http://ws.gematik.de/conn/EncryptionService/v6.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
       <CRYPT:Card>
              <CONN:CardHandle>c123456789123456789</CONN:CardHandle>
              <CCTX:Context>
                     <CONN:MandantId>m0001</CONN:MandantId>
                     <CONN:ClientSystemId>cs0001</CONN:ClientSystemId>
                     <CONN:WorkplaceId>wp007</CONN:WorkplaceId>
                     <CONN:UserId>u0001</CONN:UserId>
              </CCTX:Context>
              <CRYPT:KeyReference>C.ENC</CRYPT:KeyReference>
       </CRYPT:Card>
       <CRYPT:OptionalInputs>
              <CRYPT:EncryptionType>urn:ietf:rfc:5652</CRYPT:EncryptionType>
       </CRYPT:OptionalInputs>
       <dss:Document>
              <dss:Base64Data
MimeType="text/plain">RGllIEF1c3NlbnNjaG5pdHRzdGVsbGUgZGVzIEtvbm5la3RvcnMgd2lyZC
BkdXJjaCBbZ2VtU3BlY19Lb25dIGFic2NobGllw59lbmQgc3BlemlmaXppZXJ0LiA=</dss:Base64Da
ta>
       </dss:Document>
</CRYPT:EncryptDocument>
...
```

Für die Verschlüsselung ist es nicht erforderlich, Karten in einen (durch PIN-Eingabe) erhöhten Sicherheitszustand zu versetzen, da deren öffentliches Schlüsselmaterial verwendet wird.

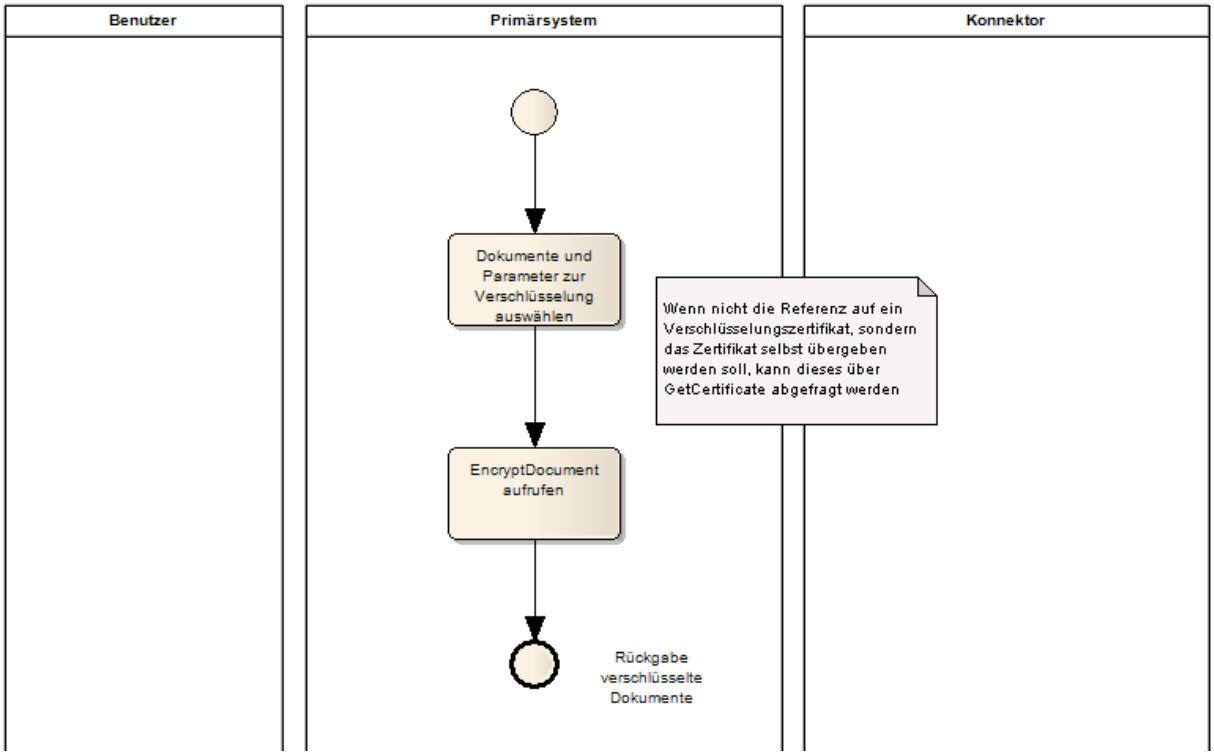


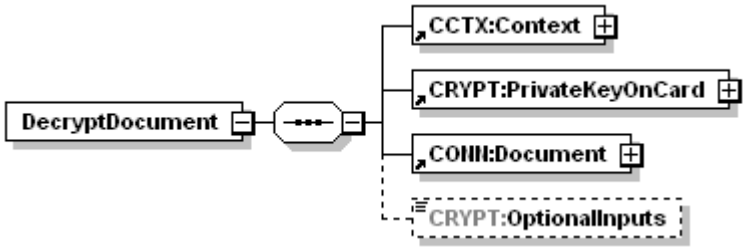
Abbildung 27: Ablauf Verschlüsseln

4.4.4.2 Entschlüsseln

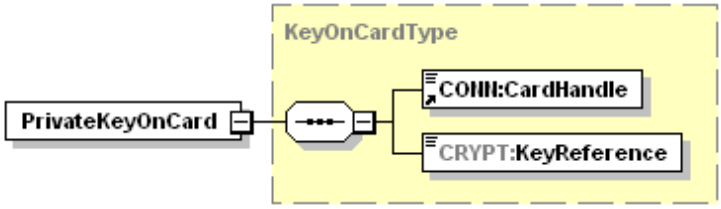
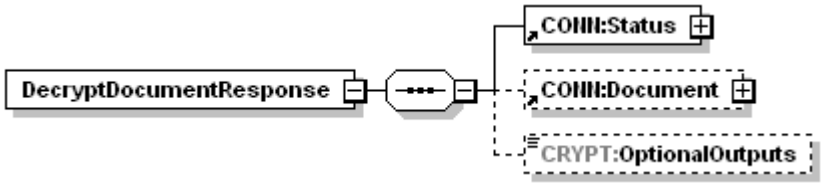
Die Operation `DecryptDocument` entschlüsselt ein hybrid verschlüsseltes Dokument. Konnektoren sind darauf ausgelegt, diejenigen Dokumente entschlüsseln zu können, die durch Konnektoren verschlüsselt wurden. Die Parameter der Entschlüsselung sind dementsprechend analog zu den Parametern der Verschlüsselung zu verwenden.

Die vollständige Beschreibung der Entschlüsselungsschnittstelle ist in [gemSpec_Kon#4.1.7.5.2] zu finden.

Tabelle 35: Operation DecryptDocument[17]

Name	`DecryptDocument`	
Beschreibung	Für die Entschlüsselung wird ein asymmetrischer Schlüssel zu einem X.509v3-Zertifikat genutzt. Dieses Zertifikat und der Schlüssel müssen von einer Karte kommen.	
Aufrufparameter		
	`PrivateKeyOnC`	Identifiziert die zu verwendende Karte und dessen (privaten)

[17] [gemSpec_Kon#4.1.7.5.2]

	`ard`	Schlüssel. Es werden die folgenden Karten unterstützt: HBAx und SM-B. 
	`CardHandle`	Identifiziert die gesteckte Karte.
	`KeyReference`	Wert `C.ENC` referenziert auf die KeyReference der HBAx und SM-B. Ist der Parameter nicht angegeben oder leer, gilt der Default-Wert `C.ENC`.
	`CONN:Document`	Enthält das base64-codierten Dokument, das entschlüsselt werden sollen.
	`CRYPT:OptionalInputs`	Kann – in zukünftigen Versionen der Spezifikation – optionale Ausgabeparameter enthalten.
Rückgabe		
	`Status`	Enthält den Ausführungsstatus der Operation.
	`CRYPT:OptionalOutputs`	Kann – in zukünftigen Versionen der Spezifikation – optionale Ausgabeparameter enthalten.
	`CONN:Document`	Enthält das entschlüsselte Dokument in base64-codierter Form

Beispiel 18: Beispiel Entschlüsseln eines Textes mit einem C.ENC Schlüssel

```
...
<CRYPT:DecryptDocument
xsi:schemaLocation="http://ws.gematik.de/conn/EncryptionService/v6.0
EncryptionService.xsd"
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:CRYPT="http://ws.gematik.de/conn/EncryptionService/v6.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
       <CRYPT:Card>
              <CONN:CardHandle>c123456789123456789</CONN:CardHandle>
              <CCTX:Context>
                     <CONN:MandantId>m0001</CONN:MandantId>
                     <CONN:ClientSystemId>cs0001</CONN:ClientSystemId>
                     <CONN:WorkplaceId>wp007</CONN:WorkplaceId>
                     <CONN:UserId>u0001</CONN:UserId>
              </CCTX:Context>
              <CRYPT:KeyReference>C.ENC</CRYPT:KeyReference>
       </CRYPT:Card>
       <CRYPT:OptionalInputs>text</CRYPT:OptionalInputs>
       <dss:Document>
              <dss:Base64Data
MimeType="text/plain">UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</dss:Base
64Data>
       </dss:Document>
</CRYPT:DecryptDocument>
...
```

Im Rahmen der Entschlüsselung wird auf privates Schlüsselmaterial zugegriffen. Die verwendeten Karten müssen sich daher in einem erhöhten Sicherheitszustand befinden, der ggf. erst durch eine PIN-Eingabe hergestellt werden muss. Da man sich insbesondere beim HBAx nicht darauf verlassen kann, dass dieser Zustand vorliegt, muss das Primärsystem den Kartenzustand abfragen und die Karte ggf. einmalig freischalten.

Mit dem (optionalen) Einblenden eines Hinweises der Form "Bitte beachten Sie die Anzeige an Ihrem Kartenterminal" muss das Primärsystem dafür sorgen, dass die Abfrage einer PIN-Eingabe am Kartenterminal vom Benutzer nicht übersehen wird.

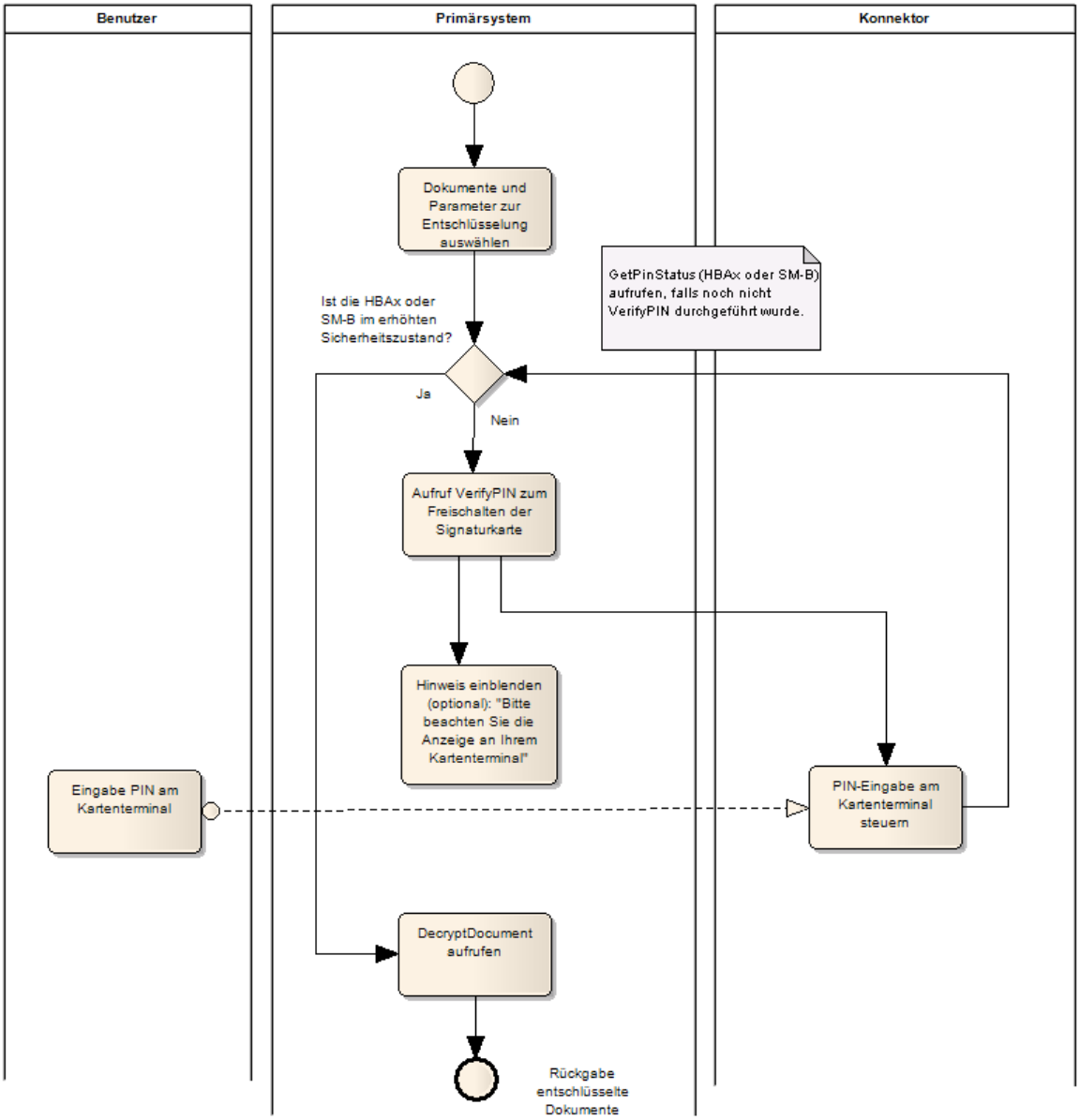


Abbildung 28: Ablauf Entschlüsseln

4.5 E-Mail-Kommunikation mittels KOM-LE

Dieses Kapitel beschreibt, wie das Primärsystem Schnittstellen einer E-Mail-Funktionalität im Rahmen des Leistungsumfanges des Projektes „Kommunikation Leistungserbringer (KOM-LE)“ nutzt.

4.5.1 Übersicht

KOM-LE stellt Primärsystemen die Möglichkeit zur Verfügung, mit anderen KOM-LE-Teilnehmern (Ärzten, Arztpraxen, Krankenhäusern, usw.) eine Ende-zu-Ende gesicherte E-Mail-Kommunikation zu führen, ohne dass sich das Primärsystem um die Sicherung der E-Mail kümmern muss. Die Verschlüsselung, Signatur, Entschlüssellung und Signaturprüfung der gesamten E-Mail unter Nutzung der Smartcards HBA und SM-B wird dabei vollständig vom KOM-LE-Clientmodul übernommen.

4.5.2 Schnittstellen

Das Primärsystem nutzt Schnittstellen zum Clientmodul gemäß der gängigen E-Mail–Standards POP3, SMTP sowie die Verzeichnisdienstschnittstelle (VZD, nicht zu verwechseln mit der Schnittstelle zum Dienstverzeichnisdienst des Konnektors) via LDAP am Konnektor in der im Folgenden beschriebenen Ausprägung.

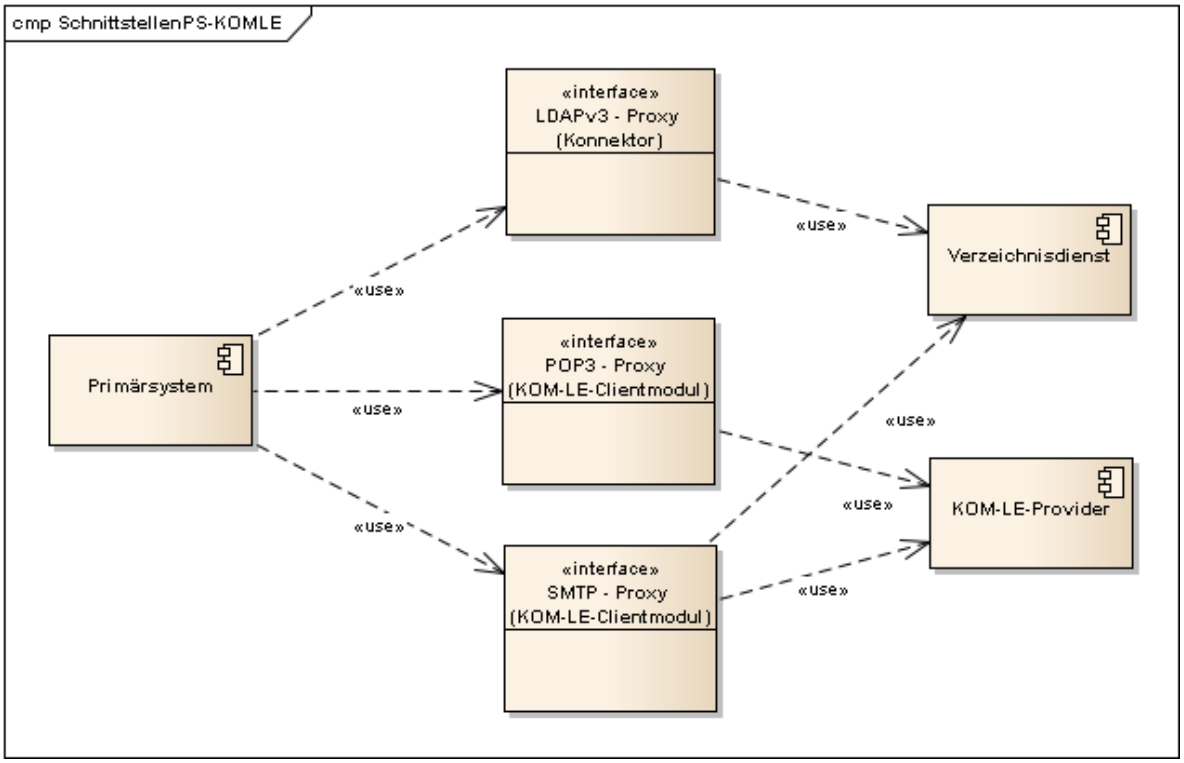


Abbildung 29: KOM-LE-Schnittstellen des PS

⌦ **KOM-LE-A_2197 Verwenden des KOM-LE-Clientmoduls**

Das PS MUSS E-Mails unter Nutzung eines KOM-LE–Clientmoduls und seiner Leistungsmerkmale versenden und empfangen können. ⌫

Zur Nutzung von KOM-LE wird vorausgesetzt, dass

- die Basisdaten des KOM-LE-Nutzers (vgl. Tabelle 36: Suchkriterien LDAP Search) in den Verzeichnisdienst (VZD) eingetragen sind,
- der Nutzer sich bei einem KOM-LE-Provider angemeldet hat, der ihm eine KOM-LE-E-Mail-Adresse eingerichtet und in dem zentralen Verzeichnisdienst (VZD) eingetragen hat,
- der Nutzer über eine freigeschaltete SM-B verfügt (bzw. einen freigeschalteten HBA), sowie
- seinen Konnektor für den Online-Modus konfiguriert hat.

Das PS kann eine E-Mail-Kommunikation mittels KOM-LE nur im Online-Modus des Konnektors durchführen (keine logische Trennung, kein Offline-Modus).

Der Konnektor hat im Online-Modus die Konfigurationseinstellungen:

- MGM_LU_ONLINE=Enabled
- MGM_LOGICAL_SEPARATION=Disabled

⌦ **KOM-LE-A_2198 Umkonfigurieren in den Online-Modus**

Das PS MUSS den Benutzer im Falle von Fehlern aufgrund eines sich im Offline-Modus befindlichen Konnektor auf die Notwendigkeit einer Umkonfiguration des Konnektors aufmerksam machen, damit der Benutzer KOM-LE nutzen kann. ⌫

4.5.3 Abläufe im Primärsystem

Eine Einbindung von KOM-LE in das Primärsystem eröffnet einen nutzerfreundlichen Nachrichtenaustausch zwischen den Kommunikationsteilnehmern. Die Leistungserbringer benutzen dabei KOM-LE in den für E-Mail-Kommunikation bekannten Anwendungsfälllen.

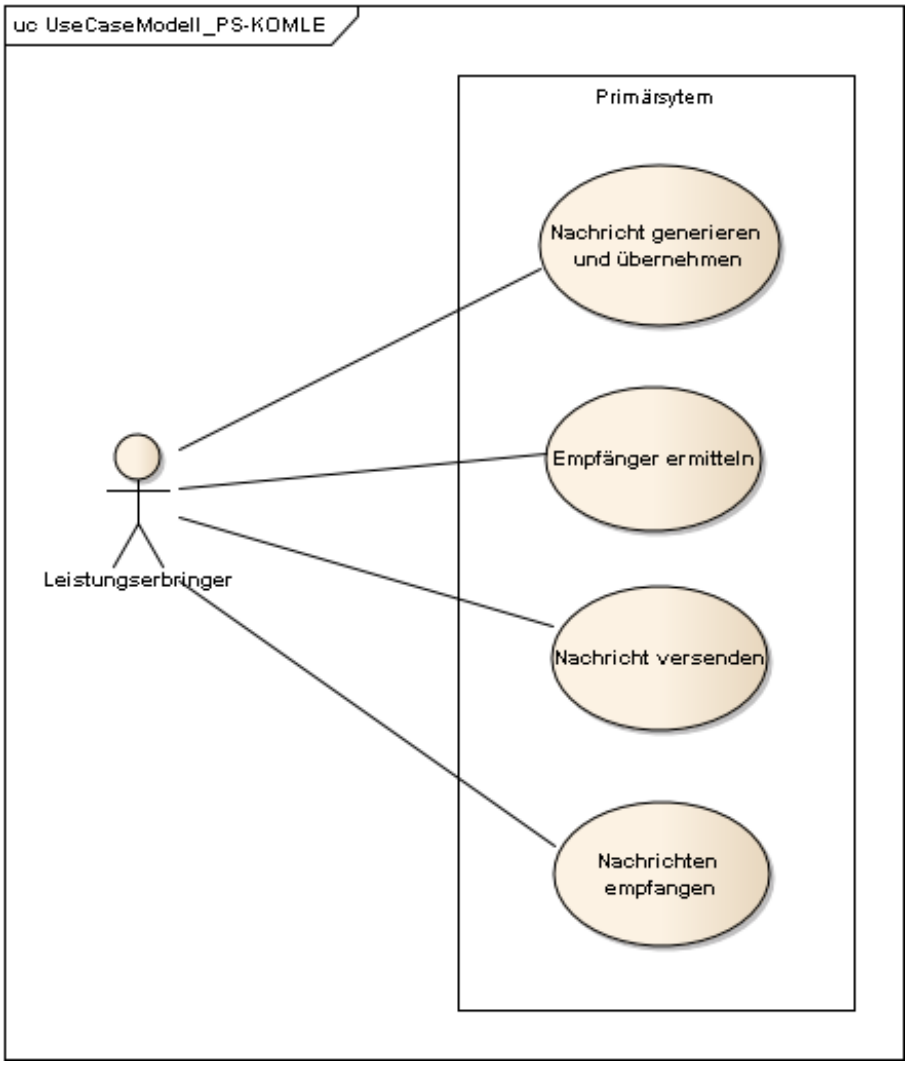


Abbildung 30: KOM-LE-Anwendungsfälle

4.5.3.1 Nachrichten generieren und übernehmen

Die Eingabe des Nachrichtentextes und das Anfordern von Lese- und/oder Zustellbestätigungen wird direkt vom PS heraus gesteuert. Als Anlage der KOM-LE–Nachricht kommen neben unsignierten Dokumenten auch (qualifiziert) signierte Dokumente in Frage. Alle Anhänge können jeweils auch separat für LE oder LE-Institutionen verschlüsselt sein.

⌦ **KOM-LE-A_2199 Nachrichtengenerierung aus dem PS heraus**

Das PS MUSS es dem Benutzer ermöglichen, Nachrichten und ggf. Anhänge zum Versand mittels KOM-LE direkt aus dem PS heraus zu erzeugen. Insbesondere müssen zu versendende Arztbriefe, wie der VhitG-Arztbrief, direkt aus dem PS bzw. der Behandlungsdokumentation heraus erzeugt werden und Nachrichtentexte des Benutzers im Primärsystem editierbar sein. ⌫

4.5.3.2 Empfänger ermitteln

Es können nur E-Mails an Empfänger versendet werden, die als Teilnehmer von KOM-LE im Verzeichnisdienst (VZD) mit ihren Verschlüsselungszertifikaten und KOM-LE-E-Mail-Adressen eingetragen sind, da die KOM-LE-Nachricht ausschließlich für bekannte KOM-LE-Teilnehmer verschlüsselt werden kann.

⌦ **KOM-LE-A_2200 Verwendung von KOM-LE-E-Mail-Adressen**

Zum Versand einer E-MAIL MUSS das PS die Header-Felder to, cc, bcc gemäß [RFC822] mit KOM-LE-E-Mail-Adresse aus dem VZD der TI füllen. Die Empfänger von KOM-LE–Nachrichten müssen über KOM-LE–E-Mail-Adressen verfügen, die aus dem VZD abgefragt werden können. ⌫

Um KOM-LE–E-Mail-Adressen von Empfängern aus dem Verzeichnisdienst (VZD) abfragen zu können, agiert das PS als LDAP-Client gegenüber dem LDAP-Proxy des Konnektors.

⌦ **KOM-LE-A_2201 VZD-Suchanfragen mittels LDAP**

Das PS MUSS als LDAP-Client gemäß LDAPv3 Standards [RFC4510] die LDAP-Operationen Bind, Unbind, Search, Abandon, StartTLS nutzen können, um eine LDAP search Operation durchzuführen. ⌫

Der VZD ist für LDAP-Suchoperationen des Primärsytems über den Konnektor erreichbar, der als LDAP-Proxy agiert. Die LDAP-Adresse ist im Dienstverzeichnisdienst des Konnektors hinterlegt.

⌦ **KOM-LE-A_2202 Nutzung des LDAP-Proxys des Konnektors**

Das PS MUSS die LDAP search Operation gemäß [RFC4511]#4.5.1 an den VZD über den LDAP-Proxy des Konnektors absetzen, dessen Adresse im Dienstverzeichnisdienst des Konnektors verzeichnet ist. ⌫

Die Suche nach der KOM-LE–E-Mail-Adresse des Nachrichtenempfängers erfolgt primär über den Namen des Empfängers – dem Namen der Person für Personen als Empfänger, oder Institutionennamen bei Institutionen als Empfänger - aber auch über zusätzliche Informationen wie Adressen, Fachgebiet oder Institutionstyp.

⌦ **KOM-LE-A_2203 Search Operation mittels LDAP-Directory Basisdatensatz Attribut**

Das PS MUSS die Mail-Adressen der Empfänger über Suchkriterien des Namens, der Postadresse der LE-Institution oder des Fachgebiets in einer LDAP search Operation gemäß [RFC4511#4.5.1] nach einem entsprechenden LDAP-Directory Basisdatensatz Attribut nach Tabelle 35 suchen können. ⌫

Tabelle 36: Suchkriterien LDAP Search[18]

Suchkriterium	Beschreibung für die Suche nach Heilberuflern	Beschreibung für die Suche nach Institutionen	LDAP-Directory Basisdatensatz Attribut
Vollständiger Name	Der commonName enthält den vollständigen Namen des Inhabers, ohne akademische Titel (auch wenn sie im Personalausweis des Antragstellers eingetragen sind).	Name der Institution (Erste zwei Zeilen des Anschriftenfeldes)	cn
Vorname	Vorname Heilberufler		gn
Nachname/Organisationsname	Nachname Heilberufler	Name der Organisation/Einrichtung des Gesundheitswesens	sn
Anzeigename	Nachname, Vorname des Heilberuflers	Name der Organisation/Einrichtung des Gesundheitswesens	displayName
Titel	Der Titel des LE (z. B. Dr. med)		title
Organisationsname	Die Bezeichnung der Organisation des Gesundheitswesens (z. B. Arztpraxis Dr. Mustermann)	Name der Organisation/Einrichtung des Gesundheitswesens	organization
Strasse, Hausnummer	Strasse, Hausnummer	Strasse, Hausnummer	streetAddress
Postleitzahl	Postleitzahl	Postleitzahl	postalCode
Stadt	Stadt	Stadt	localityName
Bundesland	Bundesland	Bundesland	stateOrProvinceName
Fachgebiet/Typ	Das Fachgebiet des Arztes	Institutionstyp	subject
Langname	Für die Verwendung von überlangen Namen von Heilberuflern	Für die Verwendung von überlangen Namen von Institutionen, z.B. Praxisgemeinschaften unter Aufzählung aller beteiligten Ärzte	otherName

Es ist möglich, dass eine Suchoperation mehrere Ergebnisse erbringt.

⌦ **KOM-LE-A_2204 Auswahl der E-Mail-Adresse des gewünschten Empfängers**

Aus den Resultaten der LDAP-Suche MUSS das PS die E-Mail–Adresse des gewünschten Empfängers übernehmen. Falls es mehrere Suchergebnisse gibt, müssen die Ergebnisinformationen dem Nutzer vollständig zur Anzeige gebracht werden, damit dieser die gewünschte E-Mail-Adresse auswählt. ⌫

18 Datenfelder gemäß [gemSpec_Kon] Tabelle 253, [gemSpec_PKI] S. 56, [BÄK_cert]

4.5.3.3 Nachrichten versenden

Der Versand von KOM-LE – Nachrichten erfolgt über das KOM-LE – Clientmodul, das die Nachricht für jeden Empfänger verschlüsselt und die gesamte Nachricht signiert.

⌦ **KOM-LE-A_2205 E-Mail-Versand als Funktion des Primärsytems**

Das PS MUSS die zu versendende Nachricht aus seinem E-Mail-Modul heraus versenden, so dass Bestandsdaten des PVS Gegenstand der KOM-LE werden können. ⌫

Die zu versendenden Dokumente können vor dem Versand vom PS über einen Aufruf der Signaturschnittstelle des Konnektors signiert und/oder verschlüsselt werden.

Das PS erstellt die Nachricht im „message/rfc822“ MIME – Format. Den Schutz der Nachricht über S/MIME übernimmt das KOM-LE–Clientmodul.

⌦ **KOM-LE-A_2206 Erstellung von MIME-Nachrichten**

Das PS MUSS eine E-Mail-Nachricht als „message/rfc822“ MIME Einheit erzeugen und über das KOM-LE-Clientmodul versenden. ⌫

Das PS muss das Schützen der Nachricht nicht übernehmen, da das Clientmodul die Nachricht automatische mit der SM-B der Organisation des Absenders signiert und für alle Empfänger verschlüsselt. Dabei wird der S-MIME-Standard verwendet.

⌦ **KOM-LE-A_2207 SMTP-Kommunikation über das KOM-LE-Clientmodul**

Das PS DARF NICHT direkt mit dem KOM-LE-Dienst (MTA) kommunizieren und MUSS stattdessen mit dem KOM-LE–Client mittels SMTP-Kommandos kommunizieren. ⌫

⌦ **KOM-LE-A_2208 SMTP-Authentifizierung über KOM-LE–Clientmodul**

Für die SMTP-Authentifizierung über das KOM-LE–Clientmodul MUSS das PS die SASL Mechanismen PLAIN und LOGIN verwenden. ⌫

Beim Aufbau der SMTP-Verbindung ist es erforderlich, Kartenverwaltungsinformationen zur SM-B mitzuliefern, die zum Integritätsschutz der Nachricht verwendet werden soll. Dazu müssen `MandantId`, `ClientsystemId` und `WorkplaceId` der Kartensitzung der erforderlichen SM-B über den Benutzernamen dem Clientmodul mitgeteilt werden.

⌦ **KOM-LE-A_2209 Nutzerkreis der KOM-LE–E-Mail-Adresse beim Nachrichtenversand**

Die Nutzerverwaltung des PS MUSS sicherstellen, dass der Nachrichtenversand über eine KOM-LE-E-Mail-Adresse nur von Personen initiiert werden kann, die vom Antragsteller der KOM-LE-E-Mail-Adresse dafür autorisiert wurden. ⌫

⌦ **KOM-LE-A_2210 Angaben zum Aufbau der SMTP-Verbindung zum KOM-LE-Clientmodul**

Bei Anwendung der SASL-Mechanismen PLAIN und LOGIN für die SMTP-Authentifizierung MUSS das PS einen persistent gespeicherten SMTP-Benutzernamen gemäß Tabelle `Bildungsregel_SMTP-POP3_Benutzername` verwenden, sowie das Passwort verwenden, das zur Authentifizierung gegenüber dem KOM-LE-Dienst (MTA) verwendet wird. Die Attribute der Tabelle

Bildungsregel_SMTP-POP3_Benutzername werden durch das „#“ – Zeichen getrennt. ⌫

Tabelle 37: Bildungsregel_SMTP-POP3_Benutzername mit Beispiel

Attribut	Beispiel
Benutzername des Absenders am KOM-LE-Dienst (E-Mail-Adresse)	erik.mustermann@komle.de
Domain Adresse des KOM-LE-Dienstes (des MTAs) inkl. Portnummer	mail.komle.de:465
MandantId	1
ClientsystemId	KOM_LE
WorkplaceId	7

Für das aufgeführte Beispiel ergibt sich der SMTP-Benutzername:

Beispiel 19: Beispiel eines SMTP-Benutzernames

```
erik.mustermann@komle.de#mail.komle.de:465#1#KOM_LE#7
```

Als Resultat der Authentisierung erhält das PS SMTP-Antwortcodes vom KOM-LE-Client, der die Verbindung zum KOM-LE-Dienst (MTA) als Proxy offen hält, etwa „250“ im Falle einer erfolgreich versendeten Nachricht, oder aber eine spezifische Fehlermeldung.

⌦ **KOM-LE-A_2211 Nutzung des SMTP–DATA-Kommandos**

Das PS MUSS das DATA-Kommando zum Versenden einer KOM-LE-Nachricht über die zuvor geöffnete SMTP-Verbindung absetzen, mit der „<CRLF>.<CRLF>“ Zeichensequenz das Ende der Nachricht markieren und schließlich den Antwortcode weiterverarbeiten. ⌫

⌦ **KOM-LE-A_2212 Beendung der SMTP-Verbindung mit QUIT**

Das PS MUSS die SMTP-Verbindung mit dem QUIT-Kommando beenden. ⌫

⌦ **KOM-LE-A_2213 Verwendung von Zustellbestätigungen**

Das PS MUSS so konfiguriert werden können, dass es beim Versenden einer Nachricht eine Zustellbestätigung gemäß [RFC3461] anfordern kann. ⌫

⌦ **KOM-LE-A_2214 Verwendung von Lesebestätigungen**

Das PS SOLL so konfiguriert werden können, dass es beim Versenden einer Nachricht beim Empfänger eine Lesebestätigung anfordern kann. Es SOLL möglich sein, die Lesebestätigung zu verweigern. ⌫

⌦ **KOM-LE-A_2215 Informieren über gescheiterten Nachrichtenversand**

Wenn das KOM-LE-Clientmodul für alle Empfänger der zu versendenden Nachricht keine Verschlüsselungszertifikate ermitteln kann, bricht es den Versand ab und liefert dem PS den Antwortcode „451“ zurück. Das PS MUSS beim Erhalt

dieses Antwortcodes den Nutzer über das Scheitern des Nachrichtenversandes mit folgendem Fehlertext informieren: „Die Nachricht konnte nicht gesendet werden, weil für keinen Empfänger gültige Verschlüsselungszertifikate ermittelt werden konnten.“ Wenn nur ein Teil des gewünschten Empfängerkreises adressiert werden konnte, MUSS der Nutzer mit der entsprechenden Meldung darüber informiert werden: „Die Nachricht wurde nur an einen Teil der gewünschten Adressaten versendet, denn es konnten nicht für alle Empfänger gültige Verschlüsselungszertifikate ermittelt werden.“ ⌫

4.5.3.4 Nachrichten empfangen

Der Empfang von KOM-LE–Nachrichten erfolgt über das KOM-LE–Clientmodul, das die Nachricht für den Empfänger entschlüsselt, sofern die dafür erforderliche Smartcard/HSM im System registriert und freigeschaltet ist.

⌦ **KOM-LE-A_2216 Nutzerkreis der KOM-LE–E-Mail-Adresse beim Nachrichtenempfang**

Die Nutzerverwaltung des PS MUSS sicherstellen, dass der Zugriff auf empfangene KOM-LE-Nachrichten Personen vorbehalten ist, die vom Antragsteller der KOM-LE-E-Mail-Adresse dafür autorisiert wurden. ⌫

⌦ **KOM-LE-A_2217 Freischaltung der für KOM-LE erforderlichen Smartcards**

Für den Empfang entschlüsselter Nachrichten erforderliche Smartcards/HSMs MÜSSEN freigeschaltet vorliegen. Ohne diese Freischaltung können Nachrichten nicht entschlüsselt entgegen genommen werden. Sind die Smartcards nicht freigeschaltet, MUSS das PS Informationen über den Status der Freischaltung von Smartcards sichtbar machen. Der Benutzer MUSS darauf aufmerksam gemacht werden, dass er zum Empfang entschlüsselter Nachrichten diese Smartcards freischalten muss.⌫

Das PS übergibt dem KOM-LE–Clientmodul in der POP3–Kommunikation alle zum Nachrichtenempfang erforderlichen Informationen. Auch für die Abholung von Nachrichten ist es dabei erforderlich, Angaben über die Ansteuerung von Smartcards des Empfängers innerhalb der POP3-Authentifizierung zu übergeben.

⌦ **KOM-LE-A_2218 Angaben zum Aufbau der POP3-Verbindung zum KOM-LE-Clientmodul**

Zur POP3–Authentifizierung gegenüber dem KOM-LE-Dienst (MTA als POP3-Server) MUSS das PS einen persistent gespeicherten POP3 Benutzernamen gemäß Tabelle `Bildungsregel_SMTP-POP3_Benutzername` verwenden, sowie das Passwort verwenden, das zur Authentifizierung gegenüber dem KOM-LE-Dienst (MTA) verwendet wird. Die Attribute der Tabelle `Bildungsregel_SMTP-POP3_Benutzername` werden durch das „#“ – Zeichen getrennt. Ist der KOM-LE–E-Mail-Adresse des Empfängers nicht eine SM-B, sondern ein HBA zugeordnet, MUSS an das Ende des POP3-Benutzernamens zusätzlich ein „#“ sowie die `UserId` für den Zugriff auf den HBA angehängt werden. ⌫

Beispiel 20: Beispiel eines POP3-Benutzernames

```
erik.mustermann@komle.de#mail.komle.de:465#1#KOM_LE#7#4
```

Die folgende POP3-Kommunikation erfolgt gemäß POP3-Protokoll über den KOM-LE–Client.

Das KOM-LE–Clientmodul leitet die POP3-Anfragen des Primärsystems an den KOM-LE-Fachdienst (MTA) weiter und entschlüsselt abgeholte Nachrichten, um sie in entschlüsselter und verifizierter Form an das Primärsystem weiterzugeben.

⌦ **KOM-LE-A_2219 Nachrichten mittels POP3 abholen**

Das PS MUSS gemäß [RFC2449] dem KOM-LE-Clientmodul POP3-Anfragen zusenden und POP3-Antwortcodes von ihm empfangen können. ⌫

Das PS schließt die POP3-Verbindung nach Bedarf, falls nicht das Clientmodul die Verbindung schließt.

⌦ **KOM-LE-A_2220 Anzeige entgegengenommener Nachrichten**

Das PS MUSS die empfangene Nachricht entgegen nehmen können und eine Anzeige der Nachricht ermöglichen. ⌫

⌦ **KOM-LE-A_2221 E-Mail-Anhänge darstellen**

Das PS MUSS E-Mail-Anhänge in Standardformaten PDF, JPEG, GIF, TXT, DOC auf der GUI anzeigen können. ⌫

⌦ **KOM-LE-A_2222 E-Mail-Anhänge verarbeiten**

Das PS MUSS E-Mail-Anhänge, die Arztbriefe wie den VhitG-Arztbrief enthalten, weiter verarbeiten können und dabei Methoden der Patientenidentifikation benutzen, die es auch beim Versand von Arztbriefen verwendet hat.⌫

Das Clientmodul erzeugt bei der Prüfung der Nachrichtensignatur einen Signaturprüfungsbericht im PDF-Format. Der Bericht wird durch das Clientmodul als Anhang mit dem Namen `Signaturpruefungsbericht.pdf` der Orginalnachricht beigefügt.

Falls ein in der empfangenen Nachricht enthaltenes Dokument mit Mitteln der TI elektronisch signiert wurde (Nutzung der Konnektorschnittstelle `SignDocument`, s. Kapitel 4.4.1), kann das PS dem Benutzer anbieten, die Signatur des Dokumentes über die in Kapitel 4.4.2 beschriebene Konnektorschnittstelle `VerifyDocument` überprüfen zu lassen.

5 Status und Logging

5.1 Erfolgreiche Verarbeitung VSDM

Eine vollständig erfolgreiche Verarbeitung umfasst immer das erfolgreiche Lesen der angeforderten Daten von der eGK sowie eine erfolgreiche Online-Prüfung, falls angefordert. Letzteres kann entweder bedeuten, dass keine Aktualisierungsaufträge für die eGK vorlagen (erfolgreiche Anfrage an Update Flag Service) oder ein oder mehrere Aufträge vorlagen und die Aktualisierung(en) erfolgreich war(en). Aus Sicht des PS sind 3 Szenarien erfolgreich (ohne Warnung, ohne Fehler):

- Lesen der VSD mit dem Parameter `PerformeOnlineCheck=false`. In diesem Fall erfolgt online lediglich eine Überprüfung des Zertifikats der eGK, welches erfolgreich war (Zertifikat nicht gesperrt). In diesem Fall ist davon auszugehen, dass aus dem laufenden Quartal bereits ein Nachweis über ein erfolgreiches Online-Update vorliegt.
- Lesen der VSD mit den Parametern `PerformeOnlineCheck=true`, `ReadOnlineReceipt=true` und `Pruefungsnachweis.Ergebnis=1` (keine Online-Prüfung notwendig, Prüfziffer vom UFS ist Bestandteil des Prüfungsnachweises)
- Lesen der VSD mit den Parametern `PerformeOnlineCheck=true`, `ReadOnlineReceipt=true` und erfolgreicher Online-Prüfung und -Aktualisierung (`Pruefungsnachweis.Ergebnis=2`, Prüfziffer vom CCS ist Bestandteil des Prüfungsnachweises)

Grundsätzlich ist die Prüfziffer nur Bestandteil des Prüfungsnachweises, wenn das Elementergebnis den Wert 1 oder 2 enthält.

5.2 Statusinformationen

⌦ VSDM-A_2933 Anzeige Verfügbarkeit lokale Komponenten

Das Primärsystem SOLL dem Benutzer die Verfügbarkeit der lokalen Komponenten und der Telematikinfrastruktur beim Start anzeigen. ⌫

Änderungen des Verfügbarkeitsstatus und Fortschrittsanzeigen bei länger dauernden Aktivitäten sollen dem Benutzer derart angezeigt werden, dass sie den Arbeitsablauf nicht behindern.

Der Verfügbarkeitsstatus meint hier konkret den Status der VPN-Verbindung des Konnektors zur TI, die VPN-Verbindung des Konnektors zum SIS sowie ggf. Fehlerzustände des Konnektors. Das PS kann zur Abfrage die Operation `GetResourceInformation` des Systeminformationsdienstes (`EventService.xsd`) des Konnektors verwenden. Diese Operation liefert als Bestandteil von `GetResourceInformationResponse` das Element `Connector` (siehe `EventService.xsd` und `ConnectorCommon.xsd`). Das PS soll beim Start oder erstmaligem Verbindungsaufbau zum Konnektor mindestens den VPN-Status zur TI ermitteln und eine Meldung anzeigen, falls der Konnektor offline ist. Sofern im

konkreten Anwendungsfall beim LE auch der Zugang zum SIS über den Konnektor verwendet wird, sollte auch diese Verbindung abgefragt und im Fehlerfall eine entsprechende Meldung angezeigt werden. Fall der SIS nicht verwendet wird, ist keine Statusabfrage diesbezüglich notwendig. Es obliegt dem Primärsystem, weitere spezifische Fehlerzustände des Konnektors abzufragen und dem Benutzer anzuzeigen. (wiederholbares Element `Connector/ OperatingState/ErrorState`).

5.3 Meldungen/Logging

VSDM-A_2934 PS: Schreiben eines Fehlerprotokolls

Das Primärsystem SOLL alle in der Kommunikation mit dem Konnektor auftretenden Fehler und Warnungen in ein Fehlerprotokoll schreiben und diese Protokollinformationen für Supportmaßnahmen über eine Zeitraum von mindestens 14 Tagen zur Verfügung halten.

VSDM-A_2935 PS: Anzeige von Meldungen

Das Primärsystem SOLL alle in der Kommunikation mit dem Konnektor auftretenden Probleme für den Benutzer verständlich anzeigen und dabei erkennen lassen, ob durch den Anwender oder den verantwortlichen Leistungserbringer Maßnahmen zur Behebung eingeleitet werden müssen.

6 Fehlerbehandlung

6.1 Übersicht

Die Primärsystemschnittstellen des Konnektors bzw. des Fachmoduls VSDM antworten bei nicht erwartungsgemäßer Verarbeitung mit einer Warnung oder einer Fehlermeldung.

Fehlermeldungen treten bei Abbruch der Verarbeitung auf (keine VSD) und werden über einen SOAP-Fault an das Primärsystem gemeldet (6.3).

Warnungen sind als Meldungen im Prüfungsnachweis zu verstehen, dass ein Problem bei der Online-Prüfung oder -Aktualisierung aufgetreten ist. Letzteres konnte nicht erfolgreich durchgeführt werden, die VSD werden aber trotzdem von der Karte gelesen und zurückgeliefert. Normative Festlegungen zur Fehlerbehandlung sind in [gemSpec_OM] zu finden.

6.2 Empfehlungen zur Fehlerbehandlung

Das Primärsystem sollte eine fehlertolerante Verarbeitung aufweisen. Dazu gehört:

- Eine planmäßige Verarbeitung von Fehlern und Warnungen der Konnektor-schnittstellen, ohne abzubrechen oder die Arbeit des Benutzers zu blockieren.
- Verständliche Anzeige von Fehlerzuständen und ggf. Erzeugen von Log-Informationen.
- Wiederholung von Anfragen, sofern bei bestimmten Fehlercodes eine Wieder-holung sinnvoll ist (z.B. Netzwerk- /VPN-Fehler, die möglicherweise nur temporär sind), Wiederholungen ggf. nach Bestätigung durch den Benutzer.
- Einhaltung von Wartezeiten und maximaler Anzahl bei Wiederholungen zur Vermeidung von Performance-Problemen.

Idealerweise lassen sich das Verhalten bei Fehlern oder Warnungen über Konfigurations-parameter einstellen (Timeout für SOAP-Requests, Retries etc.)

Wenn am PS ein Timeout für SOAP-Requests vorgesehen ist, muss dieser Timeout mindestens doppelt so lang eingestellt sein wie der der Timeout beim VSD-Update, der an der Managementkonsole des Konnektors eingestellt wurde. Wenn aufgrund dieses am Fachmodul VSD eingestellten Timeouts eine VSD-Aktualisierung abgebrochen wird, tritt kein Fehlerfall ein, sondern das PS erhält die Versichertenstammdaten der eGK sowie ein Prüfnachweis mit der entsprechenden Kennziffer. Die Festlegung eines maximalen Zeitraumes, nach dem der Versuch einer VSD-Aktualisierung abgebrochen wird, muss an der Managementoberfläche des Konnektors eingestellt werden, und darf nicht über eine Einstellung von Timeout-Parametern am Primärsystem im Widerspruch zu den genannten Einstellungen am Konnektor herbeigeführt werden.

6.3 SOAP-Fault

Bei Abbruch der Verarbeitung antwortet die Operation `ReadVSD` mit einem Standard-SOAP-Fault, der neben den Standardelementen `faultcode` und `faultstring` auch das optionale Element `detail` mit der gematik-Fehlerstruktur enthält. Das standardmäßig optionale Element `actor` wird nicht verwendet.

Die Fehlerstruktur ist gemäß [gemSpec_OM#3.2.1] folgendermaßen definiert:

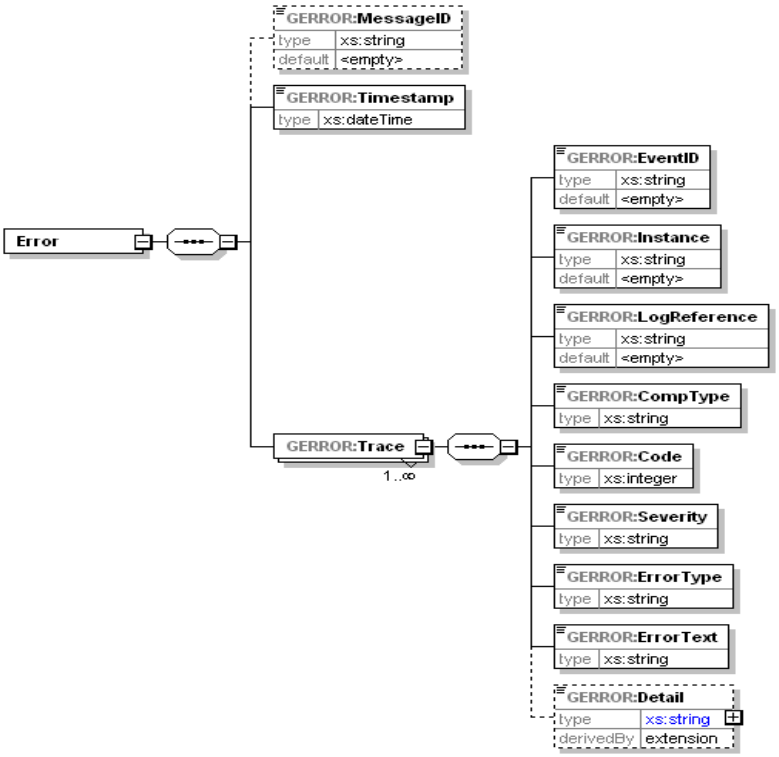


Abbildung 31: XML-Struktur der gematik Fehlermeldung [TelematikError.xsd], Version 2.0

Gemäß Festlegung der Fehlerstruktur in [gemSpec_OM] können die Elemente `MessageID`, `EventID`, `Instance` und `LogReference` leer bleiben. Weitere Beschreibungen und normative Festlegungen finden sich in [gemSpec_OM#3.2.1].

Beispiel 21: ReadVSD SOAP-Fault

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Fehlerbeschreibung allgmein</faultstring>
      <detail>
             <GERROR:Error
                   xsi:schemaLocation="http://ws.gematik.de/tel/error/v3.0
                                 ../tel/error/TelematikError.xsd"
                   xmlns:GERROR="http://ws.gematik.de/tel/error/v3.0">
               <GERROR:MessageID>m02234054321</GERROR:MessageID>
               <GERROR:Timestamp>2001-12-17T09:30:47</GERROR:Timestamp>
```

```
                <GERROR:Trace>
                    <GERROR:EventID>20120101002</GERROR:EventID>
                    <GERROR:Instance>01</GERROR:Instance>
                    <GERROR:LogReference>r34213456</GERROR:LogReference>
                    <GERROR:CompType>KONN</GERROR:CompType>
                    <GERROR:Code>3001</GERROR:Code>
                    <GERROR:Severity>FATAL</GERROR:Severity>
                    <GERROR:ErrorType>Technical</GERROR:ErrorType>
                    <GERROR:ErrorText>VSD nicht konsistent</GERROR:ErrorText>
                    <GERROR:Detail Encoding="String">
                           Ungültiger Status der eGK
                    </GERROR:Detail>
                </GERROR:Trace>
              </GERROR:Error>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

6.3.1 Sonderfall „VSD inkonsistent“

Beispiel 21: ReadVSD SOAP-Fault weist auf einen schweren Fehler beim Auslesen der Karte hin, der zum Abbruch der Operation `ReadVSD` geführt hat. In diesem Beispiel ist der Fehlercode 3001 ein Hinweis auf ein zuvor fehlgeschlagenes Update oder eine beschädigte Karte, wodurch die gespeicherten Daten inkonsistent geworden sind (Update nicht korrekt beendet). In diesem Fall ist eine Wiederholung der Operation inklusive eines Online-Updates notwendig, um den Fehler zu beseitigen, indem jetzt bei Vorliegen eines Aktualisierungsauftrags gültige Daten auf die eGK geschrieben und der Vorgang korrekt abgeschlossen werden kann. Im Online Szenario muss demnach die Operation `ReadVSD` mit `PerformOnlineCheck=true` aufgerufen werden, im Standalone-Szenario muss das Auto-Update am Online-Konnektor durchgeführt werden, bevor die Karte am Offline-Konnektor durch das PS korrekt eingelesen werden kann.

Tritt der Fehler wiederholt auf, ist die Karte als nicht nutzbar zu betrachten und muss ausgetauscht werden.

6.3.2 Sonderfall „HBA/SM-B nicht freigeschaltet“

Bestimmte Operationen erfordern einen erhöhten Sicherheitszustand eines HBA bzw. SM-B (SMC-B oder HSM). Ist dieser Zustand nicht gegeben, antwortet das Fachmodul bei entsprechenden Aufrufen mit den Fehlercodes 3041 oder 3042.

In diesem Fall soll das Primärsystem den Status der entsprechenden Karten prüfen und eine Freischaltung initiieren, sofern anzunehmen ist, dass der Benutzer die Freischaltung selbst vornehmen kann (Arztpraxis, siehe 4.1.5.3). In größeren Organisationen, z. B. Krankenhaus, ist anzunehmen, dass der Benutzer die Freischaltung nicht selbst vornimmt, sondern dies durch besonders berechtigtes Personal erfolgt, z. B. Administratoren. Daher ist in diesem Fall eine Warnmeldung sinnvoll mit dem Hinweis, sich an den Support zu wenden. Der Administrator muss in diesem Fall selbst die Freischaltung initiieren, die betroffene Karte identifizieren und die PIN am entsprechenden Terminal eingeben.

6.3.3 Sonderfall „Prüfungsnachweis nicht entschlüsselbar“

Das Element `Pruefungsnachweis` wird nur bei der Operation `ReadVSD` zurückgeliefert, wenn er angefordert worden ist und – im Falle des Standalone-Szenarios – durch das

Fachmodul im Offline-Konnektor entschlüsselt werden konnte. Falls der Prüfungsnachweis noch nicht vorhanden ist (neue Karte) oder zuvor bei der Online-Prüfung eines anderen Leistungserbringers verschlüsselt worden ist, kann er nicht gelesen bzw. entschlüsselt werden. Daraufhin wird die Operation `ReadVSD` mit speziellen Fehlermeldungen abgebrochen (Codes 3039, 3040). Das PS soll den Benutzer in diesem Fall darauf hinweisen und zur erneuten Online-Prüfung auffordern. Nach durchgeführter Online-Prüfung ist ein lesbarer und entschlüsselbarer Prüfungsnachweise auf der eGK vorhanden. In darauffolgend wiederholter Operation `ReadVSD` durch das PS am Offline-Konnektor können VSD und Prüfungsnachweis gelesen werden.

6.4 Warnungen

Um Warnungen verarbeiten zu können, die Bestandteil des Prüfungsnachweises sind, muss dieser vom Primärsystem bei `ReadVSD` durch den Parameter `ReadOnline-Receipt=true` angefordert werden. Nach entsprechender Dekodierung (base64, gzip, siehe 4.3.5.3) kann der Prüfungsnachweis als XML-Struktur geparst werden.

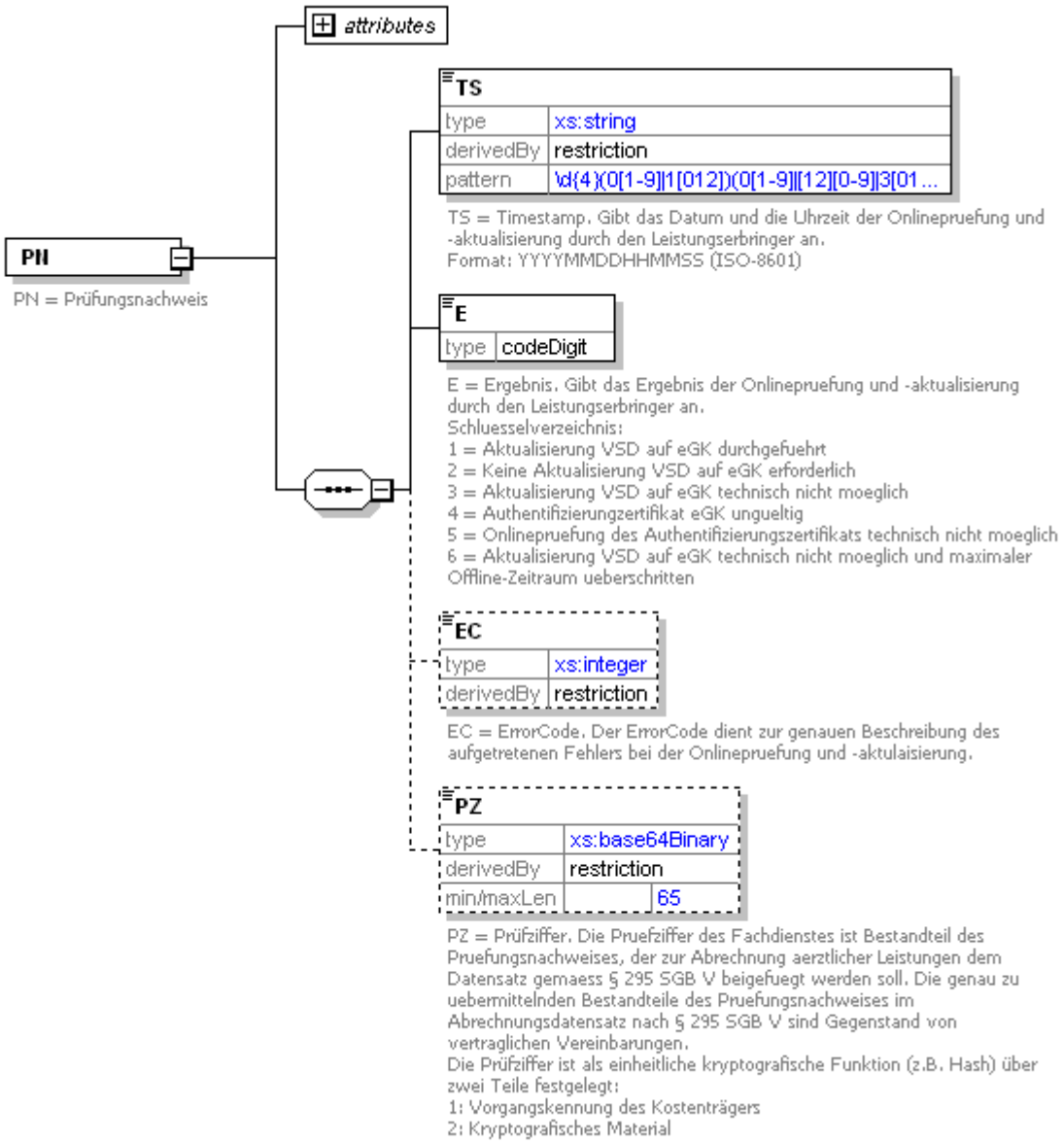


Abbildung 32: Prüfungsnachweis

Beispiel 22: Prüfungsnachweis mit ErrorCode

```
<?xml version="1.0" encoding="UTF-8"?>
<PN CDM_VERSION="0.0.0"
  xsi:schemaLocation="http://ws.gematik.de/fa/vsdm/pnw/v1.0
  ../fa/vsds/Pruefungsnachweis.xsd"
  xmlns="http://ws.gematik.de/fa/vsdm/pnw/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <TS>20130115160533</TS>
      <E>3</E>
      <EC>12101</EC>
</PN>
```

In obigem Beispiel weist das Element `PN.E=3` darauf hin, dass die Aktualisierung der eGK aus technischen Gründen nicht möglich war, die VSD aber trotzdem von der eGK gelesen worden sind. Im Errorcode `PN.EC` ist eine genauere Fehlerschreibung in Form

des Codes 12101 enthalten. („Für die angegebene Kombination aus ICCSN und Update-Identifier liegt kein Update vor“). Daher enthält das Element `PZ` in diesem Fall keine kodierte Prüfziffer.

Beispiel 23: Prüfungsnachweis ohne ErrorCode

```
<?xml version="1.0" encoding="UTF-8"?>
<PN CDM_VERSION="0.0.0"
  xsi:schemaLocation="http://ws.gematik.de/fa/vsdm/pnw/v1.0
  ../fa/vsds/Pruefungsnachweis.xsd"
  xmlns="http://ws.gematik.de/fa/vsdm/pnw/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <TS>20130115160533</TS>
      <E>5</E>
</PN>
```

In den Fällen, in denen die TI nicht erreichbar ist (offline) oder die Prüfung der Karte bereits vorher scheitert (Zertifikat der eGK ungültig oder dessen Online-Prüfung nicht möglich), enthält der Prüfungsnachweis im Ergebnis die Werte `PN.E=[4-6]`.

6.5 Sonderfall „Maximale Offline-Zeit der TI überschritten“

Im besonderen Fall `PN.E=6` ist die Aktualisierung nicht möglich und ein im Fachmodul konfigurierter Zeitraum wurde überschritten. Dieser Zustand (TI ist lange offline) soll dem Benutzer durch das Primärsystem deutlich hervorgehoben angezeigt werden. Der LE soll Maßnahmen ergreifen, um den Fehler zu analysieren und zu beseitigen, sofern die Ursache in der Verantwortung des LE liegt.

Die Festlegung der zu konfigurierenden maximalen Offline-Zeit[19] erfolgt durch die Vertragspartner. Im Auslieferungszustand des Konnektors ist der Zeitraum auf 0 eingestellt. Dadurch erfolgt keine Überprüfung auf Überschreiten eines maximalen Offline-Zeitraums und die Warnung mit `PN.E=6` würde nicht auftreten.

Ziel des besonderen Umgangs mit dieser Fehlersituation ist die Vermeidung von Missbrauch durch z. B. nicht hergestellte Netzwerkverbindungen, wodurch die Online-Prüfung immer fehlschlagen würde, trotzdem aber ein Prüfungsnachweis erzeugt wird. Der Zeitraum sollte so gewählt werden, dass in diesem Intervall üblicherweise selbst über ein Wochenende ein Fehler behoben werden kann. Bevor diese Warnung auftritt, ist am PS des LE bereits für die entsprechende Zeit zuvor bei jeder Online-Prüfung eine Warnung angezeigt worden: Prüfungsnachweis gleich 3 (Aktualisierung VSD auf eGK technisch nicht möglich) oder gleich 5 (Online-Prüfung des Authentifizierungszertifikats technisch nicht möglich). Sofern beim Auftreten dieser ersten Warnungen eine Fehlerbehebung in üblichen Reaktionszeiten erfolgt, tritt der Sonderfall der Warnung über die lange Offline-Zeit nicht auf.

Die Fehleranalyse bzw. -behebung seitens des LE sollte in zwei Schritten erfolgen:

- Visuelle Überprüfung der lokalen Komponenten (Primärsystem, Konnektor, Kartenterminal) auf grundsätzliche Funktionsfähigkeit sowie Prüfung von

[19] Der Parameter TIME-OUT_TI_OFFLINE kann wie andere Konfigurationsparameter an der Administrationsoberfläche des Fachmoduls bzw. Konnektors konfiguriert werden.

physischen Netzwerkverbindungen, ggf. Neustart einzelner Komponenten und Wiederherstellung von fehlerhaften Netzwerkverbindungen

- Bei Fortbestehen des Fehlers ist der für den Support zuständige Service-provider zu informieren, damit dieser den Fehler analysiert und abstellt.

6.6 Fehlercodes

Fehlercodes sind in Kombination mit auslösender Komponente auszuwerten. Eine Liste der mögliche Bezeichner für Komponenten der TI befindet sich in [gemSpec_OM].

Die nachfolgenden Tabellen der Fehlercodes sollen als Auszug einen Überblick über mögliche Fehlersituationen vermitteln. Da deren Definition nicht in diesem Dokument erfolgt, müssen jeweils die gültigen Werte aus den entsprechenden Dokumenten verwendet werden. Die Fehlertexte in den Tabellen enthalten Kurzbeschreibungen der Fehler und sind keine Vorgaben für Fehlermeldungen des Primärsystems. Hier soll der Hersteller darauf achten, für die Zielgruppe verständliche Formulierungen zu verwenden.

Einige Fehlercodes sind übergreifend und werden von verschiedenen Komponenten gleichartig verwendet, daher sind Komponenten nicht angegeben.

Tabelle 38: Generische Fehlercodes [gemSpec_OM]

Code	ErrorText	Auslöser
1	Verbindung abgelaufen	Die Zeit einer Verbindung hat das vorgegebene Limit überschritten.
2	Verbindung zurückgewiesen	Die Verbindung wurde vom angefragten System zurückgewiesen.
3	Nachrichtenschema fehlerhaft	Das Nachrichtenschema war inkorrekt.
4	Version Nachrichtenschema fehlerhaft	Die Version d. Nachrichtenschemas stimmt nicht mit der geforderten Version überein.
6	Protokollfehler	Genauere Aufschlüsselung des Protokollfehlers wird in den Details erfasst
101	Kartenfehler	Karte reagiert nicht oder nicht wie vorgesehen, ohne dass eine der generischen Fehlerfälle dieses Verhalten erfassen
102	Gerätefehler	Karte reagiert nicht oder nicht wie vorgesehen, ohne dass eine der generischen Fehlerfälle dieses Verhalten erfassen
103	Softwarefehler	Software (ohne Fachmodul) reagiert nicht oder nicht wie vorgesehen, ohne dass eine der generischen Fehlerfälle dieses Verhalten erfassen
104	Fachmodul reagiert nicht	Fachmodul reagiert nicht oder nicht wie vorgesehen, ohne dass eine der generischen Fehlerfälle dieses Verhalten erfassen
105	eGK nicht lesbar	Die eGK ist defekt
106	Zertifikat auf eGK ungültig	Das Zertifikat des Versicherten auf der eGK ist nach Online-Prüfung gesperrt.

Code	ErrorText	Auslöser
107	Zertifikat auf eGK ungültig	Das Zertifikat des Versicherten der eGK ist nach Offline-Prüfung ungültig.
108	Protokollierung auf eGK nicht möglich.	Protokollierung auf der eGK gescheitert.
109	Fehler beim Lesen von Daten der SM-B/HBA	Daten von der SMC/HBA konnten nicht gelesen werden.
110	Fehler beim verarbeiten von Befehlen auf der eGK	Die eGK konnte Kartenkommandos vom Fachdienst nicht erfolgreich verarbeiten.
111	Fehler beim Lesen von Daten der eGK	Daten von der eGK konnte nicht gelesen werden.
112	Fehler beim Schreiben von Daten der eGK	Daten, z.B. Prüfungsnachweis, konnte nicht auf die eGK geschrieben werden.
113	Leseversuch von veralteter eGK	Daten sollen von einer eGK älter als Generation 1 plus gelesen werden.
114	Gesundheitsanwendung auf eGK gesperrt	Die Gesundheitsanwendung der eGK ist gesperrt.

Folgende Beispiele von Fehlercodes werden vom Konnektor erzeugt. Eine vollständige Übersicht ist in [gemSpec_Kon#AnhG] zu finden.

In der folgenden Tabelle sind die verursachenden Komponenten nicht explizit für jeden Fehlercode angegeben, da es sich immer um die Komponente „Konnektor“ handelt.

Tabelle 39: Basis-Fehlercodes des Konnektors

Code	ErrorText	Auslöser
4000	Syntaxfehler/Parameterfehler	Der Fehler tritt auf, wenn ein Aufrufparameter syntaktisch nicht korrekt ist. Dieser Fehlercode deutet auf einen Programmfehler hin. Parameter, die direkt durch die Endbenutzer eingegeben werden, dürfen nicht als Syntaxfehler gemeldet werden. Für diese Fehler werden dienstspezifische Fehlercodes definiert, damit das Primärsystem entsprechende Fehlermeldungen für den Anwender des Primärsystems erzeugen kann.
4001	Interner Fehler	Ein unerwarteter Fehler ist während der Verarbeitung aufgetreten, der nicht auf die Standardfehlercodes bzw. auf die dienstspezifischen Fehlercodes abgebildet werden kann.
4094	Timeout bei Kartenzugriff	Die Operation wurde wegen Zeitüberschreitung beim Zugriff auf eine Karte abgebrochen.
4002	Der Konnektor befindet sich in einem kritischen Betriebszustand	Kritischer Betriebszustand des Konnektors

Code	ErrorText	Auslöser
4003	Keine User-Id angegeben, die zur Identifikation der Kartensitzung_HBA benötigt wird.	Fehlende oder ungültige ID im Aufrufkontext der Operation
4004	Ungültige Mandanten-ID	Fehlende oder ungültige ID im Aufrufkontext der Operation
4005	Ungültige Clientsystem-ID	Fehlende oder ungültige ID im Aufrufkontext der Operation
4006	Ungültige Arbeitsplatz-ID	Fehlende oder ungültige ID im Aufrufkontext der Operation
4007	Ungültige Kartenterminal-ID	Fehlende oder ungültige ID im Aufrufkontext der Operation
4008	Karte nicht als gesteckt identifiziert	Karten-Handle nicht gültig, Karte nicht gesteckt
4009	SM-B ist dem Konnektor nicht als SM-B_Verwaltet bekannt	Karten-Handle (SM-B) nicht gültig, Karte nicht bekannt
4010	Clientsystem ist dem Mandanten nicht zugeordnet	Ungültige Konfiguration
4011	Arbeitsplatz ist dem Mandanten nicht zugeordnet	Ungültige Konfiguration
4012	Kartenterminal ist dem Mandanten nicht zugeordnet	Ungültige Konfiguration
4021	Es sind nicht alle Pflichtparameter MandantId, Client-SystemId, workplaceId gefüllt.	Unzureichende Parameter
4032	Verbindung zu HSM konnte nicht aufgebaut werden	Fehler in der Kommunikation zum HSM
4040	Fehler beim Versuch eines Verbindungsaufbau zu KT	Fehler in der Kommunikation zum KT
4045	Fehler beim Zugriff auf die Karte	Kartenfehler
4047	Karten-Handle ungültig	TUC_KON_011 „Karten-Handle prüfen“ TUC_KON_019 „PIN ändern“ Operation GetPinStatus
4048	Fehler bei der C2C-Authentisierung	TUC_KON_005 „Card-to-Card authentisieren“
4050	Öffnen eines weiteren Kanals zur Karte nicht möglich	TUC_KON_200 „SendeAPDU“ TUC_KON_011 „Karten-Handle prüfen“ TUC_KON_200 „SendeAPDU“
4051	Falscher Kartentyp	TUC_KON_011 „Karten-Handle prüfen“ GetPinStatus, Operation AuthorizeSMC

Code	ErrorText	Auslöser
4052	Kartenzugriff verweigert	TUC_KON_019 „PIN ändern“ TUC_KON_006 „Datenzugriffsaudit eGK schreiben“ TUC_KON_219 „Entschlüssele“ TUC_KON_200 „SendeAPDU“
4174	TI VPN-Tunnel: Verbindung konnte nicht aufgebaut werden	Verbindungsfehler

Folgende VSDM-spezifische Fehler werden durch das Fachmodul oder die Fachdienste erzeugt. Die verursachenden Komponenten sind dazu explizit aufgeführt.

Tabelle 40: Fehlercodes VSDM

Comp Type	Code	ErrorText	Auslöser
FM_VSDM	3001	VSD ungültig/nicht konsistent	Status-Flag ungültig
FM_VSDM	3011	Verarbeiten der Versichertendaten gescheitert	Lesen oder Dekomprimieren des VSD-Inhalts von der Karte gescheitert
FM_VSDM	3020	Lesen KVK gescheitert	KVK-Satz konnte nicht gelesen werden
FM_VSDM	3021	KVK Prüfsumme falsch, Daten korrupt	Die Überprüfung der Prüfsumme des KVK-Satzes ergab einen Fehler.
FM_VSDM	3039	Prüfungsnachweis nicht entschlüsselbar	Daten von der SMC/HBA konnte nicht gelesen werden.
FM_VSDM	3040	Es ist kein Prüfungsnachweis auf der eGK vorhanden	Es ist kein Prüfungsnachweis auf der eGK vorhanden.
FM_VSDM	3041	SM-B nicht freigeschaltet	SMC-B oder HSM-B-Sicherheitszustand ist nicht ausreichend, z. B. für C2C oder für TLS-Verbindungsaufbau zum Intermediär
FM_VSDM	3042	HBA nicht freigeschaltet	HBA-Sicherheitszustand ist nicht ausreichend, z. B. für C2C
UFS CCS	500	Internal Server Error	Der Server ist in einen unerwarteten Zustand geraten, der die weitere Verarbeitung der Nachricht verhindert.
UFS CCS	1011	Die aufgerufene Komponente ist temporär nicht verfügbar.	Bei der Verarbeitung einer Nachricht wurde festgestellt, dass für die Verarbeitung dieser Nachricht eine benötigte Komponente nicht verfügbar ist. Unter Komponenten werden in diesem Zusammenhang interne Systeme z.B. Datenbanken, HSM, usw. verstanden.

Comp Type	Code	ErrorText	Auslöser
UFS CCS	1006	Nachricht zurückgewiesen. Die Nachricht wurde an einen für diese Anfrage nicht zuständigen Fachdienst weitergeleitet.	Die Überprüfung der Lokalisierungsinformationen innerhalb eines Fachdienstes führt zu dem Ergebnis, dass die Nachricht an den falschen Empfänger (Fachdienst) gesendet wurde.
CCS	1014	Die zu dieser ConversationID zugehörige Fachdienst-Session ist abgelaufen.	Für die in der Nachricht angegebene ConversationID konnte keine zugehörige Session ermittelt werden bzw. die Session ist abgelaufen. Dieser Fehlercode soll verwendet werden, wenn der Fehlerfall bei der Überprüfung auf Nachrichtenebene auffällt. Alternativ kann der Fehlercode 00005 verwendet werden.
CCS	5	Die zu dieser ConversationID zugehörige Fachdienst-Session ist abgelaufen.	Für die in der Nachricht angegebene ConversationID konnte keine zugehörige Session ermittelt werden bzw. die Session ist abgelaufen. Dieser Fehlercode soll verwendet werden, wenn der Fehlerfall in der fachlichen Verarbeitung auf Anwendungsebene auffällt. Alternativ kann der Fehlercode 1014 verwendet werden.
UFS	11101	Für die eGK mit der angegebenen ICCSN ist der aufgerufene Dienst nicht zuständig.	Für die eGK mit der angegebenen ICCSN ist dieser UFS nicht zuständig. Es muss die, in der ICCSN enthaltene, Issuer Identification Number (IIN) geprüft werden. Eine IIN ist dann falsch, wenn sie nicht den/die Issuer (Kartenherausgeber) bezeichnet, für den/die dieser UFS betrieben wird. Eine darüber hinausgehende Überprüfung der ICCSN ist optional, um auch (einfache) UFS-Implementierungen zu ermöglichen, bei denen der UFS nur genau diejenigen ICCSN kennt, für die Update Flags existieren.
UFS	11999	Ein nicht spezifizierter Fehler ist aufgetreten, zu dem weitere Details im Dienst protokolliert worden sind.	Der aufgetretene Fehler ist keinem spezifizierten Fehlercode zuzuordnen. Weitere Details zum Fehler sind vom Dienst protokolliert worden.
UFS	11148	Die Payload ist nicht konform zum XML-Schema.	Im Payload ist kein zum XML-Schema konformer Request GetUpdateFlags angegeben.

Comp Type	Code	ErrorText	Auslöser
CCS	12101	Für die angegebene Kombination aus ICCSN und Update-Identifier liegt kein Update vor.	Die Kombination (ICCSN, Update-Identifier) ist dem Dienst nicht bekannt, d. h. der Dienst kann hierzu keinen Vorgang zuordnen, den er durchführen soll.
CCS	12102	Für das angefragte Update ist die Durchführung eines anderen Updates eine Vorbedingung.	Der zum Update-Identifier zugehörige Vorgang kann nicht durchgeführt werden, da die Durchführung eines anderen Updates eine Vorbedingung ist. Dieser Fehler kann zum Beispiel auftreten, wenn das Clientsystem eine vorgegebene Reihenfolge von Update-Identifier nicht einhält.
CCS	12103	Die Authentifizierung zwischen Fachdienst und eGK mittels des fachdienst-spezifischen, kartenindividuellen symmetrischen Schlüssels ist fehlgeschlagen.	Der zum Update-Identifier zugehörige Vorgang konnte nicht erfolgreich durchgeführt werden, da eine Authentifizierung zwischen Fachdienst und eGK mittels des fachdienst-spezifischen, kartenindividuellen symmetrischen Schlüssels nicht erfolgreich durchgeführt werden konnte.
CCS	12105	Die eGK ist defekt.	Der zum Update-Identifier zugehörige Vorgang konnte nicht erfolgreich durchgeführt werden, da die Chipkarte defekt ist. Dieser Fehler darf nur dann gemeldet werden, wenn der Fachdienst anhand der zurückgemeldeten Statuscodes der Chipkarte einen Defekt festgestellt hat, z. B. einen Speicherfehler. Dieser Fehler darf nicht zurückgemeldet werden, wenn lediglich die Kommunikation vom Clientsystem mit dem Element Abort abgebrochen wurde.
CCS	12999	Ein nicht spezifizierter Fehler ist aufgetreten, zu dem weitere Details im Dienst protokolliert worden sind.	Der aufgetretene Fehler ist keinem spezifizierten Fehlercode zuzuordnen. Weitere Details zum Fehler sind vom Dienst protokolliert worden.

7 Komfortfunktionen

Dieser Abschnitt beschreibt informativ einige optionale Komfortfunktionen, die das Primärsystem im Zusammenhang mit den neuen Funktionen des Online-Rollout (Stufe 1) anbieten kann. Diese sind nicht als Anforderungen formuliert, sondern sind Empfehlungen, die Leistungsmerkmale der verschiedenen Systeme sein können.

7.1 Hintergrundverarbeitung bei Online-Prüfung

Das Primärsystem sollte die Online-Prüfung und -Aktualisierung so durchführen, dass die Weiterarbeit des Benutzers am Primärsystem nicht blockiert wird. Sofern der Patient bereits bekannt ist und für das laufende Quartal noch kein Prüfungsnachweis vorliegt, kann die Online-Prüfung im Hintergrund angestoßen und die betreffende Akte parallel geöffnet werden. In der überwiegenden Anzahl der Fälle wird nur der Prüfungsnachweis in das Primärsystem übernommen, was durch eine Statusmeldung signalisiert werden kann. Dadurch werden Wartezeiten für den Benutzer beim Stecken der eGK vermieden. Lediglich bei geänderten Stammdaten des Patienten, z.B. Adressänderungen, muss das PS eine Benutzerinteraktion initiieren, indem die Änderungen visualisiert und übernommen werden können.

7.2 Auswertung von Karteninformationen (HBA/SM-B)

Beim Zugriff auf die vom Konnektor verwalteten Karten des Leistungserbingers (HBA, SM-B) kann das Primärsystem Ablaufinformationen der Kartenzertifikate prüfen und bei unterschreiten einer festen oder konfigurierbaren Frist (z.B. 3 oder 6 Monate) eine Warnung ausgeben. Dies kann nach verschiedenen Regeln geschehen (erstmalige Nutzung einer Karte pro Tag/Woche/Monat) und sollte den Benutzer nicht mit Warnungen überfrachten.

Diese Funktion kann ein wichtiges Komfortmerkmal sein, um den Arzt rechtzeitig vor Ablauf eines Kartenzertifikats zu warnen und Funktionseinschränkungen damit zu verhindern. Hintergrund ist, dass der HBA möglicherweise nicht in täglicher Routine angewendet wird (z.B. wenn der LE die Signaturfunktion nicht anwendet) und nur die SM-B zum Einsatz kommt, um den Zugriff auf die GVD der eGK freizuschalten. Die SM-B steckt aber außerhalb des Sichtbereichs in einer geschützten Umgebung in einem speziellen KT.

Anhang A - Verzeichnisse

A1 – Abkürzungen

Kürzel	Erläuterung
AP	Arbeitsplatz
BCS	Basic Command Set
C2C	Card to Card (Authentifizierung)
CETP	Connector Event Transport Protocol
CMS	Card Management System
DNS	Domain Name Service
DVD	Dienstverzeichnisdienst (des Konnektors)
eGK	Elektronische Gesundheitskarte
GVD	Geschützte Versichertendaten
HBA	Heilberufsausweis
HBAx	Sammelbegriff für HBA einschließlich HBA-Vorläuferkarten wie HBA-qSig und ZOD-2.0.
HSM	Hardware Security Module
HTTP(S)	Hypertext Transfer Protocol (secure)
ICCSN	Integrated Circuit Card Serial Number
KIS	Krankhausinformationssystem
KT	Kartenterminal
KV-SafeNet	Anbindungsmöglichkeit an das Sichere Netz der Kassenärztlichen Vereinigungen
LAN	Local Area Network
LE	Leistungserbringer
MVZ	Medizinisches Versorgungszentrum
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PD	Persönliche Versichertendaten
PS	Primärsystem
PVS	Praxisverwaltungssystem
QES	Qualifizierte elektronische Signatur
SAK	Signatur Anwendungskomponente
SGB	Sozialgesetzbuch
SICCT	Secure Interoperable ChipCard Terminal

Kürzel	Erläuterung
SIS	Sicherer Internet-Service
SM-B	Security Module Typ B, Sammelbegriff für SMC-B und HSM-B
SMC	Security Module Card
SOAP	Simple Object Access Protocoll
TI	Telematikinfrastruktur
UFS	Update Flag Service
VD	Allgemeine Versicherungsdaten
VPN	Virtual Private Network
VSDD	Versicherstammdatendienst
VSDM	Versicherstammdatenmanagement
WAN	Wide Area Network
WSDL	Web Services Description Language

A2 – Glossar

Das Glossar wird als eigenständiges Dokument, vgl. [gemGlossar] zur Verfügung gestellt.

A3 – Abbildungsverzeichnis

Abbildung 1: Primärsystem im Systemkontext 9
Abbildung 2: Komponenten und Schnittstellen am Primärsystem 10
Abbildung 3: Konfigurationseinheiten 12
Abbildung 4: Online-Szenario 14
Abbildung 5: Standalone-Szenario mit physischer Trennung 14
Abbildung 6: Szenario mit logischer Trennung im Konnektor 15
Abbildung 7: Abb_LFPS_01_Element Context gemäß ConnectorContext.xsd 16
Abbildung 8: Betriebbereitschaft herstellen 19
Abbildung 9: PIC_KON_022 Grundsätzlicher Aufbau der Ereignisnachricht 25
Abbildung 10: XML-Element Event 25
Abbildung 11: Struktur des Elements Subscribe 28
Abbildung 12: Aufrufparameter von GetCards 34
Abbildung 13: GetCardsResponse 35
Abbildung 14 – Übersicht der Schnittstellen des Fachmoduls VSDM 41
Abbildung 15: Eingangsparameter ReadVSD 42
Abbildung 16: Abb_SST_PS_VSDM_05 - Schema der Ausgangsparameter `ReadVSD` 42
Abbildung 17: Abb_SST_PS_VSDM_06 - Schema von `VSD_Status` 43

Abbildung 18 – Anwendungsfall „VSD lesen mit/ohne Online-Prüfung“ 44
Abbildung 19 – Subprozess „eGK einlesen“ 45
Abbildung 20 – Subprozess „VSD von eGK lesen“ 46
Abbildung 21: Informationsmodell Versichertenstammdaten 57
Abbildung 22: Informationsmodell Prüfungsnachweis 59
Abbildung 23: Eingangsparameter SignDocument 64
Abbildung 24: Anwendungsfall „Dokumente digital signieren“ 66
Abbildung 25: Subprozess nonQES Signatur auslösen 80
Abbildung 26: Subprozess QES Signatur auslösen 83
Abbildung 27: Ablauf Verschlüsseln 98
Abbildung 28: Ablauf Entschlüsseln 101
Abbildung 29: KOM-LE-Schnittstellen des PS 102
Abbildung 30: KOM-LE-Anwendungsfälle 103
Abbildung 31: XML-Struktur der gematik Fehlermeldung [TelematikError.xsd], Version 2.0 113
Abbildung 32: Prüfungsnachweis 116

A4 – Tabellenverzeichnis

Tabelle 1: Wichtige Topics für Kartenereignisse 28
Tabelle 2: Topics für Konnektorinformationsereignisse 29
Tabelle 3: Operation RequestCard 37
Tabelle 4: Operation EjectCard 38
Tabelle 5: Konfigurationsparameter zur Online-Prüfung und -Aktualisierung 49
Tabelle 6: Entscheidungstabelle Parametrisierung ReadVSD 50
Tabelle 7: VSDM-Ereignisse 53
Tabelle 8: Änderungen im VSD-Schema 5.2 58
Tabelle 9: Konnektorschnittstelle Basisdienst Signaturdienst (nonQES und QES) 62
Tabelle 10: Zuordnung zwischen HBAx oder SM-B, Dokumententypen und Signaturtypen 63
Tabelle 11: Aufrufparameter zur Signaturerstellung, für mehrere Signaturtypen gültig 67
Tabelle 12: Operationen des SignatureService zum Steuern des xTV 70
Tabelle 13: Aufrufparameter speziell für die XML-Signatur 71
Tabelle 14: Rückgabe XML-Signatur 74
Tabelle 15: Aufrufparameter speziell für die CMS-Signatur 75
Tabelle 16: Rückgabe CMS-Signatur 76
Tabelle 17: Aufrufparameter speziell für die S/MIME-Signatur 76

Tabelle 18: Aufrufparameter speziell für die PDF-Signatur 77
Tabelle 19: Rückgabe PDF-Signatur 77
Tabelle 20: Aufrufparameter für die PKCS#1-Signatur, `ExternalAuthenticate` 77
Tabelle 21: Rückgabe PKCS#1 - Signatur 79
Tabelle 22: Ablauf Signaturerzeugung nonQES - Signatur 80
Tabelle 23: Ablauf Signaturerzeugung 83
Tabelle 24: Ablauf Verifizieren digitaler Signaturen 84
Tabelle 25: Aufrufparameter für `VerifyDocument` 85
Tabelle 26: Parameter `VerifyDocument im Spezialfall PKCS#1-Signatur` ... 87
Tabelle 27: Rückgabe `VerifyDocument` 87
Tabelle 28: Operation CheckCertificateExpiration 88
Tabelle 29: Operation ReadCardCertificate 90
Tabelle 30: Ablauf Verifizieren von Zertifikaten 91
Tabelle 31: Aufrufparameter für `VerifyCertifikate` 92
Tabelle 32: Rückgabeparameter von `VerifyCertifikate` 92
Tabelle 33: KeyReference im `EncryptionService` 93
Tabelle 34: Operation EncryptDocument 94
Tabelle 35: Operation DecryptDocument 98
Tabelle 36: Suchkriterien LDAP Search 105
Tabelle 37: Bildungsregel_SMTP-POP3_Benutzername mit Beispiel 107
Tabelle 38: Generische Fehlercodes [gemSpec_OM] 118
Tabelle 39: Basis-Fehlercodes des Konnektors 119
Tabelle 40: Fehlercodes VSDM 121
Tabelle 41: Konfigurationsparameter des PS 135
Tabelle 42: Beziehung Mandat zu Primärsystem 135
Tabelle 43: Beziehung Mandat zu Arbeitsplatz 136
Tabelle 44: Beziehung Mandat zu Kartenterminals 136
Tabelle 45: Beziehung Primärsystem zu Arbeitsplatz 136
Tabelle 46: Beziehung Primärsystem zu Kartenterminal 137
Tabelle 47: Beziehung Arbeitsplatz zu Kartenterminal 137
Tabelle 48: Übersicht Änderungen der Attribute in den Klassen 138
Tabelle 49: Übersicht Änderungen Befüllungsvorschriften der Attribute 138
Tabelle 50: Klasse Person 139
Tabelle 51: Klasse Adresse 139
Tabelle 52: Klasse Zusatzinfos GKV 140
Tabelle 53: Klasse Zusatzinfos_Abrechnung_GKV 140

Tabelle 54: Klasse Kostenerstattung 141
Tabelle 55: Klassen zur PKV 141
Tabelle 56: Klasse Ruhender Leistungsanspruch 142
Tabelle 57: Selektivverträge 142
Tabelle 58: Postleitzahl 143
Tabelle 59: Geburtsdatum 143

A5 - Beispiele

Beispiel 1: URL des Konnektordienstverzeichnisses 22
Beispiel 2: Dienstkonfiguration 22
Beispiel 3¸HTTP-SOAP-Header 24
Beispiel 4: Vollständigen Ereignisstruktur einer CETP-Event-Nachricht 26
Beispiel 5: SOAP-Request einer Subscription 28
Beispiel 6: Subscription-Ausschnitt für kritische Konnektorereignisse 30
Beispiel 7: Webservice-Call CardService.ChangePin 31
Beispiel 8: SOAP-Aufruf GetCards 34
Beispiel 9: GetCardsResponse mit einem Kartenobjekt als Rückgabe 35
Beispiel 10: Context mit „mandantwide=true“ 36
Beispiel 11: Ausschnitt aus VSDService.wsdl 54
Beispiel 12: Beispiel für einen SOAP-Call ReadVSD 54
Beispiel 13: ReadVSDResponse bei Erfolg oder Warnung 55
Beispiel 14: Beispiel qualifizierte CMS – Signatur auf einem Text-Dokument 81
Beispiel 15: Ablaufdatum von Zertifikaten auslesen 89
Beispiel 16: Beispiel Lesen des C.AUT Zertifikates 91
Beispiel 17: Beispiel Verschlüsseln eines Textes mit einem C.ENC Schlüssel 97
Beispiel 18: Beispiel Entschlüsseln eines Textes mit einem C.ENC Schlüssel 100
Beispiel 19: Beispiel eines SMTP-Benutzernames 107
Beispiel 20: Beispiel eines POP3-Benutzernames 108
Beispiel 21: ReadVSD SOAP-Fault 113
Beispiel 22: Prüfungsnachweis mit ErrorCode 116
Beispiel 23: Prüfungsnachweis ohne ErrorCode 117

A6 - Referenzierte Dokumente

A6.1 – 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 entnehmen Sie bitte der aktuellsten, auf der Internetseite der gematik veröffentlichten Dokumentenlandkarte, in der die vorliegende Version aufgeführt wird.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Glossar der Telematikinfrastruktur
[gemLF_Impl_eGK]	gematik: Implementierungsleitfaden zur Einbindung der eGK in die Primärsysteme der Leistungserbringer (siehe www.gematik.de unter Spezifikation / Release 0.5.3)
[gemSpec_FM_VSDM]	gematik: Spezifikation Fachmodul VSDM
[gemSpec_Kon]	gematik: Spezifikation Konnektor
[gemSpec_MobKT]	gematik: Spezifikation Mobiles Kartenterminal
[gemSpec_OM]	gematik: Spezifikation Operations und Maintenance
[gemSpec_SST_PS_VSDM]	gematik: Schnittstellenspezifikation Primärsystem VSDM
[gemSysL_VSDM]	gematik: Systemspezifisches Konzept Versichertenstammdatenmanagement (VSDM)
[gemSpec_CM_KOMLE]	gematik: Spezifikation KOM-LE Clientmodul
[gemSpec_PKI]	gematik: Spezifikation PKI

A6.2 – Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[BasicProfile1.2]	Basic Profile Version 1.2 http://www.ws-i.org/Profiles/BasicProfile-1.2-2010-11-09.html
[CAdES]	ETSI: *Electronic Signature Formats*, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V1.7.4, 2008-07, via http://www.etsi.org
[CDA]	ISO/HL7 27932:2009 Data Exchange Standards -- HL7 Clinical Document Architecture, Release 2 http://www.hl7.org/documentcenter/private/standards/cda/r2/cda_r2_normativewebedition2010.zip
[CDA-Sig]	Erstellung von XML-Signaturen für Dokumente nach Clinical Documents

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
	Architecture – R2, Elektronische Signatur von Arztbriefen, Ärztekammern in NRW im Auftrag der Bundesärztekammer, Version 1.6 vom 19.04.2010
[COMMON_PKI]	T7 & TeleTrust (20.01.2009): Common PKI Spezifikation, Version 2.0 http://www.t7ev.org/themen/entwickler/common-pki-v20-spezifikation.html
[eAB]	http://www.d2d.de/uploads/media/PDF_Stylesheets_02.zip
[eGES]	Elektronische Gesamtaufstellung, Version 1.5 vom 08.02.2012
[G04]	GKV-SV, KBV und KZBV Technische Spezifikation – Lesegeräte Version 2.00 vom 1.1.2003
[KBV_ITA_VGEX_Mapping_KVK]	KBV, Anwendung der eGK. Technische Anlage zu Anlage 4a (BMV-Ä/EKV), Verarbeitung KVK/eGK im Rahmen der vertragsärztlichen Abrechnung im Basis-Rollout vom 27.05.2014
[MIME]	RFC 2045, RFC 2046, RFC 2047, RFC 2048, RFC 2049
[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-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
[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)

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[PDF]	PDF Reference and Adobe Extensions to the PDF Specification http://www.adobe.com/devnet/pdf/pdf_reference.html
[PKCS#12]	"Public-Key Cryptography Standards (PKCS) #12: Personal Information Exchange Syntax", June 1999 http://www.rsa.com/rsalabs/node.asp?id=2138
[RFC822]	RFC 822: Standard for ARPA Internet Text Messages, David H. Crocker, August 1982 http://www.ietf.org/rfc/rfc822.txt
[RFC2119]	RFC 2119 (März 1997): Key words for use in RFCs to Indicate Requirement Levels S. Bradner, http://tools.ietf.org/html/rfc2119
[RFC2313]	B. Kaliski:_PKCS #1: RSA Encryption, Version 1.5, RFC 2313, http://www.ietf.org/rfc/rfc2313.txt
[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
[RFC4510]	RFC 4510 (June 2006): Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map, http://www.ietf.org/rfc/rfc4510.txt
[RFC4511]	RFC 4511 (June 2006): Lightweight Directory Access Protocol (LDAP): The Protocol, http://www.ietf.org/rfc/rfc4511.txt
[RFC5652]	R. Housley: Cryptographic Message Syntax (CMS), RFC 5652 (September 2009) http://tools.ietf.org/html/rfc5652
[RFC5751]	RFC 5751 (Januar 2010) Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification http://tools.ietf.org/html/rfc5751
[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
[RFC5322]	RFC 5322: Internet Message Format, P. Resnick, Ed., Oktober 2008
[RFC5321]	RFC 5321: Simple Mail Transfer Protocol, J.

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
	Klensin, Oktober 2008
[RFC822]	RFC 822: Standard for ARPA Internet Text Messages, David H. Crocker, August 1982
[RFC2045]	RFC 2045: Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet Message Bodies, N. Freed, N. Borenstein, November 1996
[RFC2046]	RFC 2046: Multipurpose Internet Mail Extension (MIME) Part Two: Media Types, N. Feed, N. Borenstein, November 1996
[RFC2449]	RFC 2449: POP3 Extension Mechanism, R. Gellens, C. Newman, L. Lundblade, November 1998
[RFC3463]	RFC 3463: Enhanced Mail System Status Codes, G. Vaudreuil, Januar 2003
[RFC3464]	RFC 3464: An Extensible Message Format for Delivery Status Notifications, K. Moore, G. Vaudreuil, Januar 2003
[TR–03114]	BSI TR-03114, Technische Richtlinie Stapelsignatur mit dem Heilberufsausweis, Version: 2.0, Datum: 22.10.2007, Status: veröffentlichte Version, Fassung: 2007
[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 http://www.etsi.org/deliver/etsi_ts%5C101900_101999%5C101903%5C01.04.02_60%5Cts_101903v010402p.pdf
[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 Candidate Recommendation 3 March 2012 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/

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[XSLT]	W3C Recommendation (23 January 2007) XSL Transformations (XSLT) Version 2.0 http://www.w3.org/TR/2007/REC-xslt20-20070123/
RFC3447	B. Kaliski:_PKCS #1: RSA Encryption, Version 2.1, RFC 3447, http://www.ietf.org/rfc/rfc3447.txt
[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 http://www.etsi.org/deliver/etsi_ts/103100_103199/103171/02.01.01_60/ts_103171v020101p.pdf
[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, 2012-03 http://www.etsi.org/deliver/etsi_ts/103100_103199/103173/02.01.01_60/ts_103173v020101p.pdf
[PAdES Baseline Profile]	European Telecommunications Standards Institute (ETSI): Electronic Signatures and Infrastructure (ESI); PAdES Baseline Profile; ETSI Technical Specification TS 103 172, Version 2.1.1, 2012-03 http://www.etsi.org/deliver/etsi_ts/103100_103199/103172/02.01.01_60/ts_103172v020101p.pdf

Anhang B

B1 – Konfigurationsparameter

B1.1 – Konnektorkommunikation

Tabelle 41 enthält eine Übersicht der im Kontext dieses Dokuments relevanten Konfigurationsparameter des Primärsystems. Es handelt sich um funktionale Parameter, es wird keine Aussage zur technischen Umsetzung getroffen.

Tabelle 41: Konfigurationsparameter des PS

Konfigurationsparameter für die Konnektorkommunikation	
Konnektoradresse	Netzwerkadresse und Port des Konnektorverzeichnisdienstes
Primärsystem-ID	Eine alphanumerische ID des Primärsystems, welche im Aufrufkontext der Konnektorkommunikation als `ClientSystemId` zu übergeben ist.
Mandanten-ID	Eine alphanumerische ID des Mandanten, welche im Aufrufkontext der Konnektorkommunikation als `MandantId` zu übergeben ist.
MODE_ONLINE_CHECK	Art der durchzuführenden Online-Prüfung und -Aktualisierung, siehe 4.3.4.2, am Offline-Konnektor im Standalone-Szenario immer NEVER
READ_PN	Default-Wert zur Steuerung der Übernahme des Prüfungsnachweises, sollte für PS in Umgebungen vertragsärztlicher LE immer TRUE sein, kann für andere FALSE sein
Parameter für Konfigurationseinheiten (Kontextparameter, mehrere Instanzen möglich)	
Arbeitsplatz-ID	Eine alphanumerische ID des Arbeitsplatzes, welche im Aufrufkontext der Konnektorkommunikation als `WorkplaceId` zu übergeben ist.
Benutzer-ID	Eine alphanumerische ID des Benutzers, welche im Aufrufkontext der Konnektorkommunikation als `UserId` zu übergeben ist.
TrustedViewer-ID	ID einer Instanz des Trusted Viewers
Kartenterminal-ID	Eine alphanumerische ID des Kartenterminals, welches bei der Konnektorkommunikation als `CtId` übergeben werden soll.

B1.2 – Beziehungen zwischen den Konfigurationseinheiten

Gemäß [gemSpec_Kon#4.1.1]

Tabelle 42: Beziehung Mandat zu Primärsystem

Primärsystem:	Mandant	Beschreibung/Beispiel
1	1	In einer Einzelpraxis verwendet ein Vertragsarzt genau ein Primärsystem.

Primärsystem: Mandant		Beschreibung/Beispiel
1	n	In einer Praxisgemeinschaft wird von 2 Vertragsärzten ein Primärsystem genutzt, welches die beiden Mandanten getrennt voneinander verwaltet.
n	1	Diese Konstellation ist aus Sicht *eines* Primärsystems nicht zu betrachten
n	m	In einer größeren Praxisgemeinschaft werden von 4 unabhängig voneinander eigenständigen Vertragsärzten 2 unterschiedliche Primärsysteme genutzt. Jeweils 2 Ärzte teilen sich dabei ein Primärsystem.

Tabelle 43: Beziehung Mandat zu Arbeitsplatz

Mandant: Arbeitsplatz		Beschreibung/Beispiel
1	1	In einer Einzelpraxis verwendet ein Vertragsarzt genau einen Arbeitsplatz (Aufnahme).
1	n	In größeren Einzelpraxen, Gemeinschaftspraxen und Krankenhäusern werden mehrere Arbeitsplätze genutzt.
n	1	In einer Praxisgemeinschaft teilen sich 2 Vertragsärzte einen Arbeitsplatz (Aufnahme).
n	m	In einer größeren Praxisgemeinschaft oder im Krankenhaus werden 2 oder mehr Arbeitsplätze genutzt.

Tabelle 44: Beziehung Mandat zu Kartenterminals

Mandant: Kartenterminals		Beschreibung/Beispiel
1	1	In einer Einzelpraxis verwendet ein Vertragsarzt genau 1 Kartenterminal an einem Arbeitsplatz.
1	n	In größeren Einzelpraxen, Gemeinschaftspraxen und Krankenhäusern werden mehrere Kartenterminals genutzt.
n	1	In einer Praxisgemeinschaft teilen sich 2 Vertragsärzte ein Kartenterminal, vorausgesetzt, dass ein KT mind. 2 Karten-Slots für SM-Bs hat (> 3 Slots/Mandanten nicht möglich nach aktuellem Stand).
n	m	In einer größeren Praxisgemeinschaft oder im Krankenhaus werden 2 oder mehr Kartenterminals genutzt.

Tabelle 45: Beziehung Primärsystem zu Arbeitsplatz

Primärsystem: Arbeitsplatz		Beschreibung/Beispiel
1	1	In einer Einzelpraxis wird ein Primärsystem an genau einem Arbeitsplatz verwendet.
1	n	In größeren Einzelpraxen, Gemeinschaftspraxen und Krankenhäusern wird 1 Primärsystem an mehreren Arbeitsplätzen genutzt.

Primärsystem: Arbeitsplatz		Beschreibung/Beispiel
n	1	In Praxisgemeinschaften und Notfallpraxen werden mehrere Primärsysteme (je Mandant) an genau 1 Arbeitsplatz genutzt.
n	m	In größeren Praxisgemeinschaften oder im Krankenhaus werden mehrere Primärsysteme an mehreren Arbeitsplätzen genutzt (auch hier können mehrere Primärsysteme an einem Arbeitsplatz genutzt werden).

Tabelle 46: Beziehung Primärsystem zu Kartenterminal

Primärsystem: Kartenterminal		Beschreibung/Beispiel
1	1	In einer Einzelpraxis ist 1 Primärsystem mit genau einem Kartenterminal verbunden.
1	n	In größeren Einzelpraxen, Gemeinschaftspraxen und im Krankenhaus ist genau 1 Primärsystem mit mehreren Kartenterminals verbunden.
n	1	In Praxisgemeinschaften und Notfallpraxen werden mehrere Primärsysteme (je Mandant) an genau 1 Kartenterminal genutzt.
n	m	In größeren Praxisgemeinschaften oder im Krankenhaus werden mehrere Primärsysteme an mehreren Kartenterminals genutzt (auch hier können mehrere Primärsysteme an einem Kartenterminal genutzt werden).

Tabelle 47: Beziehung Arbeitsplatz zu Kartenterminal

Arbeitsplatz: Kartenterminal		Beschreibung/Beispiel
1	1	In einer Einzelpraxis wird an einem Arbeitsplatz genau ein Kartenterminal verwendet.
1	n	Kein valides Szenario denkbar, wenn das Kartenterminal dem Arbeitsplatz zugeordnet ist (lokal).
n	1	In Praxisgemeinschaften und Notfallpraxen teilen sich mehrere Arbeitsplätze genau ein Kartenterminal.
n	m	In größeren Praxisgemeinschaften oder im Krankenhaus werden an mehreren Arbeitsplätzen mehrere Kartenterminals genutzt (auch hier können sich mehrere Arbeitsplätze genau ein Kartenterminal teilen).

B2 - Abweichungen zur Schemaversion 5.1.0

Das im Basis-Rollout verwendete VSD-Schema in der Version 5.1.0 inklusive SRQs[20] erfährt mit der Einführung des Online-Rollout (Stufe 1) inhaltliche Änderungen. Das auf Basis des Lastenheftes VSDM entwickelte VSD-Schema zum Online-Rollout wird im Dokument [gemSysL_VSDM] dargestellt und beschrieben. Die nachfolgende Tabelle

[20] Die aktuelle Übersicht der SRQs zum Basis-Rollout kann der Dokumentenlandkarte auf der gematik-Webseite entnommen werden.

stellt die Änderungen dieses Schemas im Vergleich zum im Basis-Rollout verwendeten Schema in der Version 5.1.0 dar.

Tabelle 48: Übersicht Änderungen der Attribute in den Klassen

Klasse	Änderung
Person	Änderung der minimalen Feldlänge des Feldes „Vorname“ von zwei auf ein Zeichen
Adresse	Änderung der Kardinalität des Feldes „Postleitzahl“
Zusatzinfos GKV	Wegfall des Feldes Rechtskreis und Versichertenstatus RSA
Zusatzinfos_Abrechnung_GKV	Änderung der Kardinalität WOP
Kostenerstattung	Umbenennung der Felder für ambulante und stationäre Kostenerstattung Aufnahme der Felder für zahnärztliche Versorgung und veranlasste Leistungen
Zusatzinfos PKV	Wegfall aller Klassen zur PKV
Ruhender Leistungsanspruch	Aufnahme neue Klasse mit den Feldern Beginn, Ende und Art des Ruhens
Selektivverträge	Aufnahme neue Klasse mit den Feldern ärztliche, zahnärztliche und Art der Selektivverträge

Tabelle 49: Übersicht Änderungen Befüllungsvorschriften der Attribute

Attribut/Feld	Änderung
Geburtsdatum	Anpassung an die Vorgaben der DEÜV

B2.1 – Beschreibung der Änderungen der Attribute in den Klassen

Die Änderungen im Schema Version 5.2 werden in den folgenden Tabellen im Vergleich zum Schema Version 5.1 gegenüber gestellt und beschrieben.

Tabelle 50: Klasse Person

5.1.0	5.2.0
	Person - Nachname :name - Vorname :name - Geburtsdatum :ISO8601Date - Geschlecht :gender - Titel :nameExtension [0..1] - Namenszusatz :nameExtension [0..1] - Vorsatzwort :nameExtension [0..1]
Änderung	
Die minimale Feldlänge wird von zwei auf ein Zeichen reduziert	
Grund der Änderung	
Regelung gem. Art. 47 EGBGB für die Namenüberführung in deutsches Namenrecht. Berücksichtigung des anwendbaren ausländischen Namensrechts für Personen, die nicht dem deutschen Namensrecht unterliegen.	

Tabelle 51: Klasse Adresse

5.1.0	5.2.0
	Adresse - Postleitzahl :string [0..1] - Ort :string
Änderung	
Änderung der Kardinalität des Feldes „Postleitzahl“ von Pflichtfeld auf optionales Feld.	
Grund der Änderung	
Bei Anschriften ohne Postleitzahl darf das Feld nicht genutzt werden. Ein leeres Pflichtfeld könnte als fehlerhafte Befüllung interpretiert werden.	

Tabelle 52: Klasse Zusatzinfos GKV

5.1.0	5.2.0
Änderung	
Wegfall der Informationen zu „Rechtskreis“ und „Versichertenstatus RSA“	
Grund der Änderung	
Wegfall der Statusergänzungsmerkmale aufgrund Beanstandung BMG. Gesetzliche Grundlage zur Speicherung nicht mehr gegeben.	

Tabelle 53: Klasse Zusatzinfos_Abrechnung_GKV

5.1.0	5.2.0
Änderung	
Änderung der Kardinalität von „optional“ auf „verpflichtend“ Auslagerung der Informationen zur Kostenerstattung in eine eigene Klasse „Kostenerstattung“	
Grund der Änderung	
Gemäß Anlage 21 des BMV-Ä und EKV ist das WOP-Kennzeichen verpflichtend umzusetzen.	

Tabelle 54: Klasse Kostenerstattung

5.1.0	5.2.0
nicht vorhanden	
Änderung	
Definition einer neuen Klasse „Kostenerstattung“ zur Aufnahme der Informationen für Ärztliche Versorgung (früher Kostenerstattung_ambulant) Stationärer Bereich (früher Kostenerstattung_stationär) Zahnärztliche Versorgung Veranlasste Leistungen	
Grund der Änderung	
Anpassung an geänderte gesetzliche Bestimmungen (§ 13 Abs. 2 SGB V).	

Tabelle 55: Klassen zur PKV

5.1.0	5.2.0
	entfällt
Änderung	
Wegfall aller Informationen zur privaten Krankenversicherung	
Grund der Änderung	
Wegfall der Informationen aufgrund Kündigung des Gesellschaftervertrags durch die PKV.	

Tabelle 56: Klasse Ruhender Leistungsanspruch

5.1.0	5.2.0
nicht vorhanden	**RuhenderLeistungsanspruch** - Beginn :ISO8601Date - Ende :ISO8601Date - ArtDesRuhens :codeDigit
Änderung	
Definition einer neuen Klasse zur Aufnahme der Informationen Beginn Ende Art des Ruhens zur präzisen Abbildung des ruhenden Leistungsanspruchs gem. Abs. 2a Satz 3 § 291 SGB V.	
Grund der Änderung	
Festlegung des BMG: Keine Nutzung des Feldes „Ende des Versicherungsschutzes“ zur Abbildung des ruhenden Leistungsanspruchs.	

Tabelle 57: Selektivverträge

5.1.0	5.2.0
nicht vorhanden	
Änderung	
Definition einer neuen Klasse zur Aufnahme von Informationen zur Kennzeichnung abgeschlossener Selektivverträge.	
Grund der Änderung	
Beschluss in der 33. GSV (Schlichterspruch) zur Aufnahme der Informationen zu abgeschlossenen Selektivverträgen des Versicherten.	

B2.2 – Beschreibung der Änderungen der Befüllungsvorschriften von Attributen

Tabelle 58: Postleitzahl

5.1.0	5.2.0
VSD:Postleitzahl type: VSD:TF_Textstring_10 minLen/maxLen: 10 Gibt die Postleitzahl der Strassen- oder Postfachadresse an.	VSD:Postleitzahl type: xs:string derivedBy: restriction min/maxLen: 1 10 Gibt die Postleitzahl der Strassen- oder Postfachadresse an. Die Befüllung des Feldes Postleitzahl erfolgt gemäß den Festlegungen der DEÜV. In Verbindung mit dem Wohnsitzländercode "D" für Deutschland MUSS die Postleitzahl 5-stellig numerisch sein. Soweit Angaben zur Adresse und zum Postfach gemacht werden, MUSS die Postleitzahl zu beiden Adressdaten vorhanden sein. Bei Anschriften ohne Postleitzahl wird das Feld nicht verwendet.
Änderung	
Das Feld ist optional und muss bei Anschriften ohne Postleitzahl nicht befüllt werden.	
Grund der Änderung	
Befüllung gemäß DEÜV. Abweichend davon sind jedoch keine Leerzeichen zulässig. Ausprägung als optionales Feld.	

Tabelle 59: Geburtsdatum

5.1.0	5.2.0
VSD:Geburtsdatum type: VSD:ZP_Zeitpunkt_8 minLen/maxLen: 8 Gibt das Geburtsdatum des Versicherten in dem Format "YYYYMMDD" (ISO-8601) an.	VSD:Geburtsdatum type: VSD:ISO8601Date pattern: \d{4}(0[0-9]|1[012])(0[0-9]|[12][0-9]|3[01]) Gibt das Geburtsdatum des Versicherten an. Hinweis: Das Geburtsjahr MUSS immer gefuellt werden. Bei Inlaendern ist immer ein logisch richtiges Geburtsdatum anzugeben. Bei Auslaendern gilt folgendes: Zumindest das Geburtsjahr ist immer anzugeben. Im Geburtstag oder im Geburtstag und im Geburtsmonat ist bei Ausländern „00“ bzw. „0000“ zulässig, wenn der Geburtstag und der Geburtsmonat nicht zu ermitteln sind.
Änderung	
Für ausländische Versicherte kann die Angabe „00“ für nicht bekannte Geburtstage oder „0000“ für nicht bekannte Geburtstage und Geburtsmonate verwendet werden.	
Grund der Änderung	
Anpassung an die Vorgaben der DEÜV	

B2.3 – Verarbeitung von Datenfeldern durch das Primärsystem

In den Versichertenstammdaten der eGK sind Datenfelder enthalten, welche erst ab Beginn des Online-Wirkbetriebs sinnvoll nutzbar sind.

Hierzu gehören die Felder

- zur Kostenerstattung,

- zum ruhenden Leistungsanspruch,
- zu abgeschlossenen Selektivverträgen
- und zum Zuzahlungsstatus der Versicherten.

Eine Zuzahlungsbefreiung wird in der Übergangszeit, wie bisher, durch ein zusätzliches Dokument nachgewiesen welches durch die Krankenkasse ausgestellt wird.

Für die Befüllung und Interpretation des VSD-Schemas Version 5.2.0 gilt folgende Vorgehensweise:

- Die optionalen Elemente/Felder „Ruhender Leistungsanspruch“ und "Kostenerstattung" werden von den Kassen nicht personalisiert, d. h. nicht in den Datensatz geschrieben.
- Das Pflichtfeld „Status“ aus dem Element „Zuzahlungsstatus“ wird mit dem Wert 0 (von Zuzahlungspflicht nicht befreit) gefüllt. Das optionale Feld „Gueltig_bis“ aus dem Element „Zuzahlungsstatus“ wird nicht in den Datensatz geschrieben.
- Die Pflichtfelder „Aerztlich“ und „Zahnaerztlich“ aus dem Element „Selektivvertraege“ werden einheitlich mit dem Wert „9“ (= Feld wird nicht genutzt) gefüllt. Das optionale Feld „Art“ wird nicht genutzt.
- Die Inhalte der Felder „Zuzahlungsstatus", „Ruhender Leistungsanspruch“, "Kostenerstattung" und „Selektivvertraege“ werden bis zu einer anderweitigen Regelung im Bundesmantelvertrag der Ärzte nicht ausgewertet.

Ab wann eine direkte Verarbeitung dieser Felder durch das Primärsystem erfolgen soll, wird durch die Vertragspartner rechtzeitig bekannt gegeben.