Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation Systemprozesse der dezentralen TI

Version:	1.~~2~~3.0 CC
Revision:	~~231013~~266668
Stand:	~~28.06.2019~~05.08.2020
Status:	zur Abstimmung freigegeben
Klassifizierung:	Öffentlich_Enwurf
Referenzierung:	gemSpec_Systemprozesse_dezTI

Dokumentinformationen

Änderungen zur Vorversion

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

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	14.12.18		freigegeben	gematik
1.1.0	15.05.19		Einarbeitung Änderungsliste P18.1	gematik
1.~~1~~2.0	~~15.05~~28.06.19		~~freigegeben~~Einarbeitung P19.1	gematik
1.3.0 CC	05.08.20		Einarbeitung ~~P19.1~~Änderungsliste P22.3	gematik
~~1.2.0~~	~~28.06.19~~		~~freigegeben~~	~~gematik~~

Inhaltsverzeichnis

1 Einordnung des Dokuments

1.1 Zielsetzung

In der Spezifikation Systemprozesse der dezentralen TI werden Leistungen der TI-Plattform beschrieben und als Systemprozesse deklariert. Diese Systemprozesse beschreiben wie mit Produkttypen der TI zu verfahren ist, um eine Plattformleistung für Fachanwendungen der TI zu erbringen. Diese Lösung hat das Ziel, Basisleistungen der TI-Plattform einheitlich und produkttypunabhängig zu definieren.

1.2 Zielgruppe

Das Dokument richtet sich an Hersteller von Produkten der TI, welche fachanwendungsspezifische Funktionalitäten implementieren dafür auf dezentrale Komponenten der TI-Plattform bzw. Dienste der TI-Plattform zugreifen und zu deren Umsetzung die Systemprozesse dezentrale TI nutzen.

1.3 Geltungsbereich

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

Wichtiger Schutzrechts-/Patentrechtshinweis

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

1.4 Abgrenzungen

Die Nutzung der Systemprozesse dezentrale TI wird produkttypspezifisch festgelegt. Bspw. haben die Produkttypen Konnektor, eHealth-KT und Mob-KT individuelle Spezifikationen und nutzen die Systemprozesse dezentrale TI nicht.

1.5 Methodik

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

Anforderungen werden im Dokument wie folgt dargestellt:
<AFO-ID> - <Titel der Afo>
Text / Beschreibung
[<=]

Dabei umfasst die Anforderung sämtliche zwischen Afo-ID und Textmarke [<=] angeführten Inhalte.

2 Leistungen

Produkttypen der dezentralen TI, welche Anwendungsfälle der Fachanwendungen umsetzen, nutzen dafür Komponenten der TI-Plattform, bspw. beim Zugriff auf Sicherheitsmodule wie Smartcards (eGK, SMC-B, …) oder ein HSM, Verwendung von Zertifikaten der TI oder Nutzung von Signatur-, Verzeichnis-, Zeit- und Namensdienst im zentralen Netz. In dieser Spezifikation werden die Leistungen der TI-Plattform einheitlich und produkttypunabhängig beschrieben und als Systemprozesse der dezentralen TI deklariert.

Durch das Zusammenschalten von Operationen und Bausteinen der verschiedenen Fachdomänen der TI-Plattform (Kartenzugriff, PKI, Kryptografische Verfahren) entstehen höherwertige Plattformbausteine mit einer vereinheitlichten Syntax für den Zugriff auf produkttypübergreifende Plattformleistungen („**PL_TUC_***"). Den Zusammenhang der verschiedenen Domänen und den damit komponierten höherwertigen Systemprozessen verdeutlicht die folgende Abbildung als *technische Dokumentenlandkarte* (in der Darstellung grün markiert).

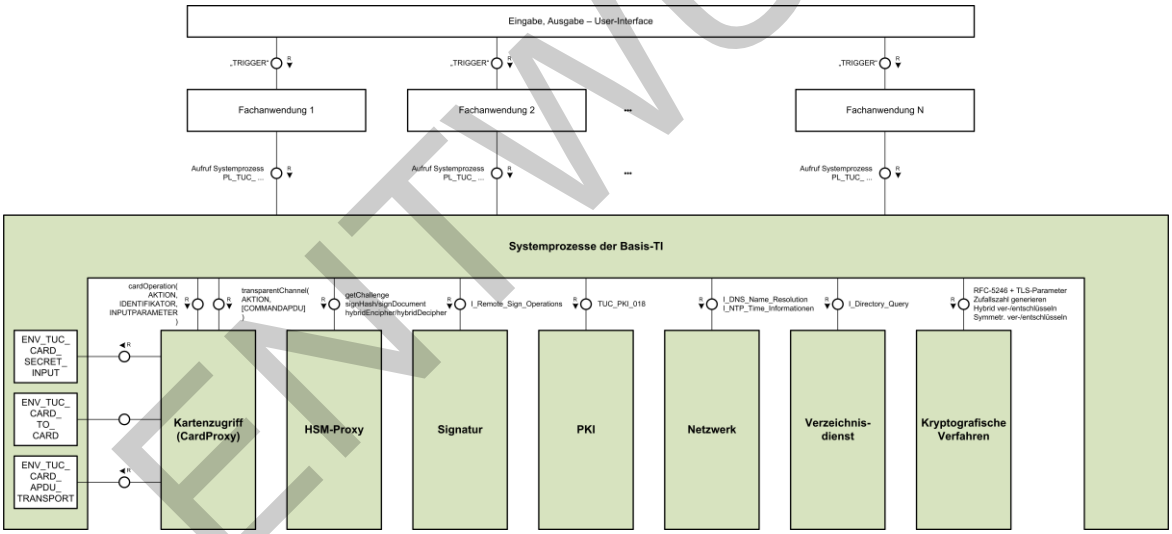

Abbildung 1: Systemprozesse der Basis-TI

Die Beschreibung dieser Systemprozesse der TI erfolgt normativ, es wird jedoch auf eine prozedurale Ablaufbeschreibung verzichtet. Es erfolgt eine Festlegung, was zu tun ist, um eine vorgegebene Plattformleistung zu erbringen. Die konkrete Realisierung dieser Leistung eines Systemprozesses ist abhängig von Umgebungsannahmen und muss unter bestimmten Bedingungen um umgebungsspezifische Operationen und Festlegungen ergänzt werden. Sie sorgen für einen umgebungsspezifischen Zuschnitt (tayloring) der Systemprozesse, um eine TI-übergreifend spezifizierte Leistung in einer konkreten Ablaufumgebung von einem konkreten Produkttypen oder Dienst einer Fachanwendung zu erbringen.

Die umgebungsspezifischen Operationen, Umgebungsannahmen oder -parameter müssen von der Realisierungsumgebung („**ENV_TUC**_*“) normativ festgelegt werden. Der Produkttyp, der die hier spezifizierten Plattformleistungen nutzt, muss Festlegungen treffen, wie diese umgebungsabhängigen Schnittstellen zu implementieren sind. Damit ergibt sich für die Realisierung der Systemprozesse in einer konkreten Fachanwendung für eine konkrete Realisierungsumgebung ein Spezifikationsanteil, der in der folgenden Abbildung orange gekennzeichnet ist.

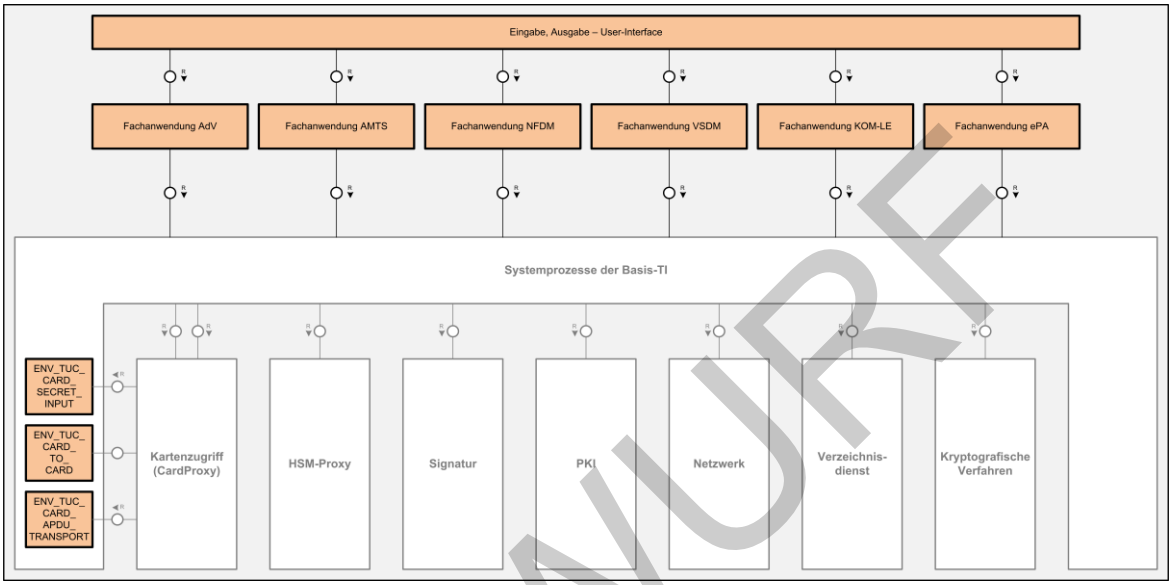

Abbildung 2: Umgebungsspezifische Operationen

2.1 Systemprozesse für den Zugriff auf Smartcards der TI

Der Zugriff auf Smartcards der TI wird in verschiedenen Produkttypen der TI durch einen CardProxy gemäß [gemSpec_CardProxy] gekapselt. Der CardProxy kommuniziert pro Instanz mit einer einzelnen eGK, SM-B, mit einem HBA oder einer anderen entsprechenden Karte. Der CardProxy stellt Anwendungen eine höherwertige Schnittstelle für den Zugriff auf eine Karte zur Verfügung und übersetzt die parametrisierbaren Operationen in kartenverständliche APDU-Sequenzen. Der CardProxy verwaltet intern den Freischaltstatus der Karte und organisiert bei technischer Notwendigkeit einer PIN-Eingabe oder Freischaltung durch eine weitere Karte auf Basis von Zugriffsregeln und dem aktuellen Freischaltzustand eines Artefakts auf der Karte.

Die Kommunikation mit dem CardProxy wird durch die hier beschriebenen Plattformbausteine gekapselt. Die Plattformbausteine leiten die Aufrufe an den CardProxy weiter der zum einen eine höherwertige Kartenoperationen als *cardOperation* bereitstellt und zum anderen eine direkte, bei Bedarf auf eine verschlüsselte, Kommunikation mit der Karte über *APDU*-Sequenzen erlaubt. In der Schnittstelle zur cardOperation sind sämtliche kartenspezifischen Aspekte gekapselt, jede Aktion auf und mit der Karte wird auf die jeweils angegebenen Rückgabewerte abgebildet. In der direkten Kommunikation über einen transparenten Kanal erfolgt keine Auswertung der zur und von der Karte übertragenen APDU-Kommandos.

2.1.1 Die Realisierungsumgebung des CardProxy

Der CardProxy benötigt einen Zugriff auf Umgebungsschnittstellen, die je nach Einsatzumgebung der Karten unterschiedlich ausgeprägt sind. Der CardProxy benötigt eine Transportschnittstelle der physischen Anbindung zur Karte, einen Kommunikationskanal zu einem Remote-CardProxy mit einer zweiten Karte für eine Freischaltung nach dem Zwei-Schlüssel-Prinzip (Card-2-Card) und eine Schnittstelle zur Eingabe eines PIN-Geheimnisses.

2.1.1.1 ENV_TUC_CARD_SECRET_INPUT – Realisierung Eingabe PIN-Geheimnis

Das System zur Umsetzung der Plattformleistungen zur Anbindung der Karte muss für seine konkrete Realisierungsumgebung festlegen, wie PIN- bzw. PUK-Geheimnisse von einem Benutzerinterface an die Karte gelangen.

TIP1-A_6889 - Eingabeschnittstelle für PIN
Das System zur Umsetzung der Plattformleistungen PL_TUC_CARD_* MUSS eine Eingabeschnittstelle definieren, mittels der die Eingabe eines PIN-Geheimnisses an der Schnittstelle CardProxy und Kartenterminal gemäß [gemSpec_CardProxy#Schnittstelle CardProxy und Kartenleser] übergeben wird.**[**<=**]**

TIP1-A_6890 - Eingabeschnittstelle für PUK
Das System zur Umsetzung der Plattformleistungen PL_TUC_CARD_* MUSS eine Eingabeschnittstelle definieren, mittels der die Eingabe eines PUK-Geheimnisses an der Schnittstelle CardProxy und Kartenterminal gemäß [gemSpec_CardProxy#Schnittstelle CardProxy und Kartenleser] übergeben wird.**[**<=**]**

TIP1-A_6891 - Eingabeschnittstelle für newPIN
Das System zur Umsetzung der Plattformleistungen PL_TUC_CARD_* MUSS eine Eingabeschnittstelle definieren, mittels der die Eingabe eines neuen PIN-Geheimnisses an der Schnittstelle CardProxy und Kartenterminal gemäß [gemSpec_CardProxy#Schnittstelle CardProxy und Kartenleser] übergeben wird.**[**<=**]**

TIP1-A_7017 - Statusinformationen im Rahmen der PIN-Verifikation
Produkttypen und Dienste der TI die eine PIN/PUK-Eingabe mittels ENV_TUC_CARD_SECRET_INPUT umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy#Sicherheitszustand] zurückmelden:

1. ErrorUserVerification „Fehler im Authentisierungsprotokoll"
2. OK „Sicherheitszustand passend gesetzt"

[<=**]**

2.1.1.2 ENV_TUC_CARD_TO_CARD – Realisierung Card-2-Card

Das System zur Umsetzung der Plattformleistungen zur Anbindung der Karte muss für seine konkrete Realisierungsumgebung festlegen, wie der Datentransport innerhalb einer Card-2-Card-Freischaltung zwischen zwei beteiligten Karten erfolgt.

TIP1-A_6892 - Umgebungsschnittstelle für Card-2-Card
Das System zur Umsetzung der Plattformleistungen PL_TUC_CARD_* MUSS eine Transportschnittstelle „Umgebung" definieren, mittels welcher der Datenaustausch von Challenge und Response im Card-2-Card-Verfahren zwischen zwei CardProxy-Instanzen

von CardProxy_A an CardProxy_B gemäß [gemSpec_CardProxy#Sicherheitszustand#Card-2-Card] realisiert wird.**[**<=**]**

TIP1-A_7018 - Statusinformationen im Rahmen von Card-2-Card
Produkttypen und Dienste der TI die eine Card-2-Card-Freischaltung mittels ENV_TUC_CARD_TO_CARD umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy#Sicherheitszustand] zurückmelden:

1. ErrorAuthentication „Fehler im Authentisierungsprotokoll"
2. ErrorImportCVC „Fehler im CV-Zertifikatimport"
3. OK „Sicherheitszustand passend gesetzt"
4. WrongEndEntityCVC „Das End-Entity-CV-Zertifikat enthält nicht die Rechte, die nötig sind um die Aktion freizuschalten"

[<=**]**

2.1.1.3 ENV_TUC_CARD_APDU_TRANSPORT – Realisierung APDU-Transport

Das System zur Umsetzung der Plattformleistungen zur Anbindung der Karte muss für seine konkrete Realisierungsumgebung festlegen, wie die elektrische Schnittstelle zwischen CardProxy und Karte als Kartenkontaktiereinheit IFD realisiert wird.

TIP1-A_6893 - Umgebungsschnittstelle für Kartenkommandos
Das System zur Umsetzung der Plattformleistungen PL_TUC_CARD_* MUSS eine Schnittstelle realisieren, mittels welcher der Transport der APDU-Kommandos zwischen CardProxy und Karte gemäß [gemSpec_CardProxy Konzept der Komponente Kartenterminal Proxy] über eine Kartenkontaktiereinheit gemäß [ISO7816-3] realisiert wird.**[**<=**]**

2.1.2 Konfiguration und Statusinformationen

Um die korrekte Funktionsweise einer CardProxy-Instanz in einer konkreten Realisierungsumgebung sicherzustellen, ist eine Konfiguration und Initialisierung des CardProxies erforderlich. Es muss festgelegt werden, welchen Kartentyp eine jeweilige CardProxy-Instanz unterstützen soll und welche Operation auf welchen Objekten der jeweiligen Karte in einer Anwendung zulässig sind.

2.1.2.1 Konfiguration des CardProxy

TIP1-A_6894 - Konfiguration des Kartenzugriffs
Das System zur Umsetzung der Kartenzugriffe mittels CardProxy MUSS für seine Realisierungsumgebung eine Konfigurationstabelle gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] für jeden unterstützten Kartentyp einer SmartCard der TI definieren.**[**<=**]**

Die Konfigurationstabelle legt die zulässigen Operationen für jeden unterstützten Kartentyp in einer konkreten Realisierungsumgebung fest. Um eine Eindeutigkeit in der Auswahl einer passenden Zugriffsregel eines Objektes auf der Karte zu erhalten, muss der Nutzer der Plattformleistung festlegen, in welchen Rollen ein Akteur in Anwendungsfällen mit Bezug zu einer Karte der TI interagieren kann.

TIP1-A_7019 - Konfiguration der Rollen des Benutzers einer Karte
Das System zur Umsetzung der Plattformleistungen für Systemprozesse der TI-Plattform MUSS für seine Realisierungsumgebung festlegen, welche Rollen ein Benutzer in Anwendungsfällen mit Bezug auf eine Karte der TI einnehmen darf.**[<=]**

TIP1-A_6895 - Festlegung des Zugriffsprofils zur Rollenauthentisierung gegenüber einer eGK
Das System zur Umsetzung der Plattformleistungen für Systemprozesse der TI-Plattform MUSS festlegen, welche Zugriffe eine SmartCard (HBA, SM-B) bei der Rollenauthentisierung gegenüber einer eGK in seiner konkreten Realisierungsumgebung freischalten darf.**[<=]**

Mit dieser Anforderung wird sichergestellt, dass die zum Einsatz kommenden Karten über die entsprechenden Zertifikate zur Rollenauthentisierung gegenüber einer eGK verfügen. Diese Rollen bilden Zugriffsrechte auf der eGK ab, die in der Konfigurationstabelle für den CardProxy verzeichnet sind.

2.1.2.2 Initialisierung CardProxy für eGK

Bei der Initialisierung des CardProxy in dessen Zugriff sich eine eGK befindet, soll die komplette CV-Zertifikatskette einer in der Realisierungsumgebung vorgehaltenen SM-B, die für die Freischaltung der eGK vorgesehen ist, übergeben werden. Daraus ergibt sich, dass die in der Realisierungsumgebung eingesetzte SM-B bereits über einen initialisierten CardProxy adressiert werden kann. Die Instanz des CardProxy mit eGK muss mit der Referenz der SM-B der übergebenen SM-B-CV-Zertifikatskette und dem bei der Initialisierung des SM-B-CardProxy ausgelesenen X.509-AUT-Zertifikats assoziiert werden.

TIP1-A_6896 - Initialisierung CardProxy für eGK
Das System zur Umsetzung der Plattformleistungen für Systemprozesse der TI-Plattform MUSS bei der Initialisierung des CardProxies mit Zugriff auf eine eGK

- die gesamte CV-Zertifikatskette einer für die Freischaltung vorgesehenen SM-B-Identität der Realisierungsumgebung an den eGK-CardProxy übergeben
- die vorgesehene SM-B mit diesem eGK-CardProxy für die Dauer des Zugriffs auf die eGK assoziieren und zu dieser Verbindung das C.HCI.AUT-Zertifikat der SM-B temporär speichern.
- die in PL_TUC_CARD_INFORMATION gelisteten Informationen zu dieser Karte mittels CardProxy aus der Karte auslesen.

[<=]

2.1.2.3 Initialisierung CardProxy für SM-B

Bei der Initialisierung des CardProxy in dessen Zugriff sich eine SM-B befindet, soll die SM-B mittels PIN-Eingabe freigeschaltet werden sowie das CV- und das X.509-AUT-Zertifikat ausgelesen werden.

TIP1-A_6897 - Initialisierung CardProxy für SM-B
Das System zur Umsetzung der Plattformleistungen für Systemprozesse der TI-Plattform MUSS bei der Initialisierung des CardProxies mit Zugriff auf eine SM-B

- eine Benutzerverifikation durchführen mittels PL_TUC_CARD_VERIFY_PIN und dem IDENTIFIKATOR *PIN.SMC* gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy für SMC-B]

- die in PL_TUC_CARD_INFORMATION gelisteten Informationen zu Karte mittels CardProxy aus der Karte auslesen
- das CV-CA-Zertifikat zum C.SMC.AUTR_CVC-Zertifikat der im Zugriff befindlichen SM-B und sofern vorhanden alle dazugehörigen Cross-Zertifikate der CVC-Root aus der TSL auslesen.

[<=]

2.1.2.4 PL_TUC_CARD_INFORMATION – Gesammelte Statusinformationen zu einer Karte

Der Systemprozess PL_TUC_CARD_INFORMATION sammelt Statusinformationen zu einer SmartCard, die über eine umgebungsspezifische Schnittstelle an das System angebunden wird und stellt diese zum Abruf durch andere Systemprozesse bereit. Die Informationen umfassen zum einen Auskünfte über Kartentyp und Kartengeneration bzw. -version und zum anderen Statusinformationen über auf der Karte vorhandene Anwendungen und PINs.

TIP1-A_6898 - Leistung zu Statusinformationen zu einer Karte
Produkttypen und Dienste der TI mit Zugriff auf Smartcards der TI MÜSSEN eine Plattformleistung PL_TUC_CARD_INFORMATION realisieren und mit Statusinformationen einer SmartCard befüllen, die über die Umgebungsschnittstelle ENV_TUC_CARD_APDU_TRANSPORT mit dem System verbunden wird.**[<=]**

TIP1-A_6899 - Liste verfügbarer Informationen zu einer SmartCard
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_INFORMATION umsetzen, MÜSSEN die folgenden Informationen zum Status einer angebundenen SmartCard sammeln, bei Änderung aktualisieren und für die Dauer der Verbindung zu dieser SmartCard zum Abruf bereitstellen.

Statusdatum	
• Kartentyp • ICCSN • Produkttypversion des COS • Produkttypversion des Objektsystems • Echtheit der Karte	Diese Informationen werden vom CardProxy bei der Initialisierung der Karte selbstständig erfasst
Informationen bei Kartentyp = eGK	

Status der Anwendungen auf der eGK: • DF.HCA • DF.AMTS • DF.NFD • DF.DPE	Aufruf der Cardproxy-*cardOperation* mit dem *Identifikator der Fachanwendung (siehe links)* gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy eGK G2] und dem Aktionsparameter *SELECT* Abbildung der Rückgabewerte von *cardOperation* je Fachanwendung wie folgt: OK → AVAILABLE FileDeactivated → HIDDEN ObjectNotFound → ABSENT ObjectTerminated → TERMINATED
Status der PINs der eGK: • PIN.CH • MRPIN.AMTS • PIN.AMTS_REP • MRPIN.NFD • MRPIN.DPE	Aufruf der Cardproxy-*cardOperation* mit dem *Identifikator der PIN (siehe links)* gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy eGK G2] und dem Aktionsparameter *GETSTATUS* Abbildung der Rückgabewerte von *cardOperation* je Fachanwendung wie folgt: PasswordProtected → TransportProtected PasswordDisabled → PasswordDisabled RetryCounter.0 → PasswordBlocked Wenn X > 0 RetryCounter.X → PasswordEnabledNotVerified.X OK → PasswordEnabledVerified
Authentisierungszertifikat der eGK C.CH.AUT	Auslesen des Zertifikats mittels PL_TUC_CARD_READ_FILE und dem IDENTIFIKATOR = EF.C.CH.AUT.R2048 oder IDENTIFIKATOR = EF.C.CH.AUT.E256, in Abhängigkeit des zum aktuellen Zeitpunkt PL_TUC_NET_SYNC_TIME zulässigen kryptografischen Verfahren gemäß [gemSpec_Krypt#2.1 Identitäten] gemäß [gemSpec_CardProxy] Transformation der ausgelesenen Daten in die X.509-Zertifikatstruktur
Authentisierungszertifikat der eGK (pseudonymisiert) C.CH.AUTN	Auslesen des Zertifikats mittels PL_TUC_CARD_READ_FILE und dem IDENTIFIKATOR = EF.C.CH.AUTN.R2048 oder IDENTIFIKATOR = EF.C.CH.AUTN.E256, in Abhängigkeit des zum aktuellen Zeitpunkt PL_TUC_NET_SYNC_TIME zulässigen kryptografischen Verfahren gemäß [gemSpec_Krypt#2.1 Identitäten] gemäß [gemSpec_CardProxy] Transformation der ausgelesenen Daten in die X.509-Zertifikatstruktur

Verschlüsselungszertifikat der eGK für elektronische Dokumente C.CH.ENC	Auslesen des Zertifikats mittels PL_TUC_CARD_READ_FILE und dem IDENTIFIKATOR = EF.C.CH.ENC.R2048 oder IDENTIFIKATOR = EF.C.CH.ENC.E256, in Abhängigkeit des zum aktuellen Zeitpunkt PL_TUC_NET_SYNC_TIME zulässigen kryptografischen Verfahren gemäß [gemSpec_Krypt#2.1 Identitäten] gemäß [gemSpec_CardProxy] Transformation der ausgelesenen Daten in die X.509-Zertifikatstruktur
Verschlüsselungszertifikat der eGK für elektronische Verordnungen C.CH.ENCV	Auslesen des Zertifikats mittels PL_TUC_CARD_READ_FILE und dem IDENTIFIKATOR = EF.C.CH.ENCV.R2048 oder IDENTIFIKATOR = EF.C.CH.ENCV.E256, in Abhängigkeit des zum aktuellen Zeitpunkt PL_TUC_NET_SYNC_TIME zulässigen kryptografischen Verfahren gemäß [gemSpec_Krypt#2.1 Identitäten] gemäß [gemSpec_CardProxy] Transformation der ausgelesenen Daten in die X.509-Zertifikatstruktur
Informationen bei Kartentyp = SM-B	
Status der PINs der SM-B PIN.SMC	Aufruf der Cardproxy-*cardOperation* mit dem *Identifikator der PIN* gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy SMC-B] und dem Aktionsparameter *GETSTATUS* Abbildung der Rückgabewerte von *cardOperation* je Fachanwendung wie folgt: PasswordProtected → TransportProtected PasswordDisabled → PasswordDisabled RetryCounter.0 → PasswordBlocked Wenn X > 0 RetryCounter.X → PasswordEnabledNotVerified.X OK → PasswordEnabledVerified
Authentisierungszertifikat der SM-B gegenüber der eGK C.SMC.AUTR_CVC	Auslesen des Zertifikats EF.C.SMC.AUTR_CVC.R2048 oder EF.C.SMC.AUTR_CVC.E256 , in Abhängigkeit des zum aktuellen Zeitpunkt PL_TUC_NET_SYNC_TIME zulässigen kryptografischen Verfahren gemäß [gemSpec_Krypt#2.1 Identitäten] aus dem CV-CertificateStore des CardProxy gemäß [gemSpec_CardProxy#Bausteine innerhalb von CardProxy]

Authentisierungszertifikat der SM-B gegenüber der eGK für die PIN Status Prüfung C.SMC.NULL_CVC	Einlesen des Zertifikates vom Speicherort.
Authentisierungszertifikat der SM-B gegenüber Fachdiensten mit TLS C.HCI.AUT	Auslesen des Zertifikats mittels PL_TUC_CARD_READ_FILE und dem IDENTIFIKATOR = EF.C.HCI.AUT.R2048 oder IDENTIFIKATOR = EF.C.HCI.AUT.E256, in Abhängigkeit des zum aktuellen Zeitpunkt PL_TUC_NET_SYNC_TIME zulässigen kryptografischen Verfahren gemäß [gemSpec_Krypt#2.1 Identitäten] gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy SMC-B] Transformation der ausgelesenen Daten in die X.509-Zertifikatstruktur
Zertifikat für einen lesbaren eGK-Protokolleintrag <optional vorhanden>	Auslesen des Zertifikats mittels PL_TUC_CARD_READ_FILE und dem IDENTIFIKATOR gemäß der Festlegung in [gemSpec_CardProxy#Konfigurationstabelle CardProxy SMC-B] und den Vorgaben zur Erzeugung eines Protokolleintrags auf der eGK Transformation der ausgelesenen Daten in die X.509-Zertifikatstruktur

[<=]

2.1.2.5 PL_TUC_EGK_STATUS – Gültigkeit der eGK prüfen

Der Systemprozess PL_TUC_EGK_STATUS fasst Leistungen verschiedener Domänen unter Einbeziehung einer elektronischen Gesundheitskarte zu einer höherwertigen Plattformleistung zusammen. Mit dieser wird eine Gültigkeitsprüfung der eGK durchgeführt, die zum einen Prüfschritte direkt auf der Karte durchführt und andererseits die Legitimität der Karte mittels Onlineabfrage beim Kartenherausgeber prüft.

TIP1-A_6901 - Prüfkriterien der Gültigkeit der eGK
Produkttypen und Dienste der TI mit Zugriff auf eine elektronische Gesundheitskarte mittels CardProxy MÜSSEN eine Plattformleistung PL_TUC_EGK_STATUS zur Prüfung des Status einer eGK umsetzen, die die eGK den folgenden Prüfkriterien unterzieht:

Prüfkriterium	**Prüfergebnis**
Abbildung des Werts zur Echtheit der Karte in *PL_TUC_CARD_INFORMATION.Echtheit* auf den Wahrheitswert **ja** wenn die Karte für echt befunden wurde, sonst **nein**.	Echtheit: **ja / nein**

Abbildung des Status der Gesundheitsanwendung auf der eGK in *PL_TUC_CARD_INFORMATION.DF.HCA* auf den Wert „**aktiv**", wenn Status = AVAILABLE „**nicht aktiv**", wenn Status ungleich AVAILABLE	Gesundheits-anwendung: **aktiv / nicht aktiv / Prüffehler**
Prüfung der Gültigkeit des Zertifikats der Karteninhaberidentität C.CH.AUTN der eGK aus PL_TUC_CARD_INFORMATION mittels PL_TUC_PKI_VERIFY_CERTIFICATE unter Verwendung der folgenden Parameter: • Zu prüfendes Zertifikat: C.CH.AUTN • Referenzzeitpunkt: „jetzt" (aktuelle gesetzliche Zeit) • PolicyList: <oid_egk_autn> • KeyUsage: „digitalSignature" • ExtendedKeyUsage: „id-kp-clientAuth" • OCSP-Graceperiod: NULL oder default • Offline-Modus: „nein" • OCSP-Response: NULL • Timeout: default • TOLERATE_OCSP_FAILURE: „ja"	Gültigkeit zu Referenzzeitpunkt: **„zeitlich gültig / ungültig" / Prüffehler** Mathematische Gültigkeit: **„mathematisch gültig / ungültig" / Prüffehler** OCSP-Prüfung: **„Online gültig / Online gesperrt / nicht geprüft / Prüffehler"**

[<=]

TIP1-A_6902 - Prüfergebnis der Echtheit_und_Gültigkeit der eGK
Produkttypen und Dienste der TI MÜSSEN zur Realisierung von PL_EGK_STATUS über das Ergebnis jedes Prüfkriteriums der Echtheit- und Gültigkeitsprüfung der eGK informieren und mit einem Status die erfolgreiche Prüfung aller Kriterien mitteilen.

a. Echtheit => „**ja / nein / Prüffehler**"
b. Gesundheitsanwendung => „**aktiv** / **nicht aktiv / Prüffehler**"
c. Karteninhaberzertifikat => „**zeitlich gültig / ungültig / Prüffehler**"
„**mathematisch gültig / ungültig / Prüffehler**"
„**Online gültig / Online gesperrt / Onlinestatus unbekannt / Prüffehler**"

[<=]

2.1.2.6 PL_TUC_CARD_RESET – Rücksetzen einer Karte

Mit dem Systemprozess PL_TUC_CARD_RESET soll der logische Kanal einer im Zugriff eines CardProxy befindlichen SmartCard der TI auf den Initialisierungsstand zurückgesetzt werden.

TIP1-A_7020 - Leistung zum Rücksetzen einer Karte
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Rücksetzen des logischen Kanals einer SmartCard als Plattformleistung PL_TUC_CARD_RESET gemäß [gemSpec_CardProxy] *cardOperation* mit dem

Aktionsparameter *RESETCHANNEL* und dem IDENTIFIKATOR „*“ (Wildcard) umsetzen und das Abschließen dieser Aktion mit dem Rückgabewert OK bestätigen.**[<=]**

2.1.3 Zugriff auf Smartcards der TI

Der folgende Abschnitt definiert Systemprozesse für den Zugriff auf Smartcards der TI als funktionale Abläufe. Voraussetzung für die korrekte Funktionsweise sind zum einen umgebungsspezifische Abläufe an den Außenschnittstellen, die von der jeweiligen Realisierungsumgebung festgelegt werden müssen. Zum anderen muss für die jeweils durch einen CardProxy adressierbaren Karten eine Konfigurationstabelle der zulässigen Kartenoperationen definiert werden.

2.1.3.1 PL_TUC_CARD_CHANGE_PIN – PIN Ändern

TIP1-A_6903 - Leistung zur Änderung einer PIN
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Ändern einer PIN auf einer SmartCard als Plattformleistung PL_TUC_CARD_CHANGE_PIN gemäß [gemSpec_CardProxy] *cardOperation für Passwortobjekte* mit dem Aktionsparameter *CHANGE* umsetzen.**[<=]**

TIP1-A_6904 - Aufrufparameter zum Ändern einer PIN
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_CHANGE_PIN umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* des Passwortobjektes gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6905-01~~TIP1-A_6905~~ - Ergebnis der Änderung einer PIN
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_CHANGE_PIN umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „PIN erfolgreich geändert“
2. CardTerminated „Karte nicht mehr verwendbar“
3. MemoryFailure „Karte defekt“
4. ObjectNotFound „IDENTIFIKATOR ungültig“
5. PasswordBlocked „PIN gesperrt”
1. ~~PasswordProtected „PIN mit Transportschutz”~~
6. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
7. WrongSecretWarning.X „PIN falsch, noch X Versuche“
8. WrongLength „neue PIN hat die falsche Länge“

[<=]

Durch den Systemprozess PL_TUC_CARD_CHANGE_PIN wird das PIN-Geheimnis einer referenzierten PIN auf einer SmartCard der TI geändert.

2.1.3.2 PL_TUC_CARD_ENABLE_PIN – PIN-Schutz einschalten

Mit dem Systemprozess PL_TUC_CARD_ENABLE_PIN wird die PIN-Verifikation der referenzierten PIN eingeschaltet.

TIP1-A_6906 - Leistung zum Einschalten einer PIN
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Einschalten einer PIN auf einer SmartCard als Plattformleistung PL_TUC_CARD_ENABLE_PIN gemäß [gemSpec_CardProxy] *cardOperation für Passwortobjekte* mit dem Aktionsparameter *ENABLE* umsetzen.**[<=]**

TIP1-A_6907 - Aufrufparameter zum Einschalten einer PIN
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_ENABLE_PIN umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* des Passwortobjektes gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6908 - Ergebnis des Einschaltens einer PIN
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_ENABLE_PIN umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „PIN erfolgreich eingeschaltet"
2. CardTerminated „Karte nicht mehr verwendbar"
3. MemoryFailure „Karte defekt"
4. ObjectNotFound „IDENTIFIKATOR ungültig"
5. PasswordBlocked „PIN gesperrt"
6. PasswordProtected „PIN mit Transportschutz"
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
8. WrongSecretWarning.X „PIN falsch, noch X Versuche"

[<=]

2.1.3.3 PL_TUC_CARD_DISABLE_PIN – PIN-Schutz abschalten

TIP1-A_6909 - Leistung zum Abschalten einer PIN
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Abschalten einer PIN auf einer SmartCard als Plattformleistung PL_TUC_CARD_DISABLE_PIN gemäß [gemSpec_CardProxy] *cardOperation für Passwortobjekte* mit dem Aktionsparameter *DISABLE* umsetzen.**[<=]**

TIP1-A_6910 - Aufrufparameter zum Abschalten einer PIN
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_DISABLE_PIN umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* des Passwortobjektes gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6911 - Ergebnis des Abschaltens einer PIN
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_DISABLE_PIN umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „PIN erfolgreich abgeschaltet"
2. CardTerminated „Karte nicht mehr verwendbar"
3. MemoryFailure „Karte defekt"
4. ObjectNotFound „IDENTIFIKATOR ungültig"
5. PasswordBlocked „PIN gesperrt"

6. PasswordProtected „PIN mit Transportschutz”
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
8. WrongSecretWarning.X „PIN falsch, noch X Versuche“

[<=]

Mit dem Systemprozess PL_TUC_CARD_DISABLE_PIN wird die PIN-Verifikation einer referenzierten PIN abgeschaltet. Objekte auf einer SmartCard mit Zugriffsbedingungen, die die referenzierte PIN enthalten, sind bei abgeschalteter PIN weniger geschützt.

2.1.3.4 PL_TUC_CARD_UNBLOCK_PIN – PIN mit PUK entsperren

TIP1-A_6912 - Leistung zum Entsperren einer PIN mittels PUK
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Entsperren einer PIN auf einer SmartCard als Plattformleistung PL_TUC_CARD_UNBLOCK_PIN gemäß [gemSpec_CardProxy] *cardOperation für Passwortobjekte* mit dem Aktionsparameter *UNBLOCK* umsetzen.**[<=]**

TIP1-A_6913 - Aufrufparameter zum Entsperren einer PIN mittels PUK
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_UNBLOCK_PIN umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* des Passwortobjektes gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6914 - Ergebnis der Entsperrung einer PIN mittels PUK
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_UNBLOCK_PIN umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „PIN erfolgreich entsperrt“
2. CardTerminated „Karte nicht mehr verwendbar“
3. MemoryFailure „Karte defekt“
4. ObjectNotFound „IDENTIFIKATOR ungültig“
5. PasswordBlocked „PUK gesperrt”
6. PasswordProtected „PIN mit Transportschutz”
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
8. WrongSecretWarning.X „PUK falsch, noch X Versuche“
9. WrongLength „neue PIN hat die falsche Länge“

[<=]

Mit dem Systemprozess PL_TUC_CARD_UNBLOCK_PIN wird eine gesperrte PIN entsperrt. Das Entsperren kann mit gleichzeitigem Setzen einer neuen PIN oder ohne das setzen einer neuen PIN erfolgen. Der Modus der Entsperrung erfolgt auf Grundlage der Festlegungen in der Konfiguration des CardProxies für einen bestimmten Kartentypen.

2.1.3.5 PL_TUC_CARD_VERIFY_PIN – Benutzer verifizieren

TIP1-A_6915 - Leistung zur Benutzerverifikation
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Benutzerverifikation mittels PIN als Plattformleistung PL_TUC_CARD_VERIFY_PIN gemäß [gemSpec_CardProxy] *cardOperation für Passwortobjekte* mit dem Aktionsparameter *VERIFY* umsetzen.**[<=]**

TIP1-A_6916 - Aufrufparameter der Benutzerverifikation
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_VERIFY_PIN umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* des Passwortobjektes gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6917 - Ergebnis der Leistung zur Eingabe einer PIN
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_VERIFY_PIN umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „PIN erfolgreich verifiziert"
2. PasswordBlocked „PIN gesperrt"
3. PasswordProtected „PIN mit Transportschutz"
4. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
5. WrongSecretWarning.X „PIN falsch, noch X Versuche"
6. ObjectNotFound „IDENTIFIKATOR ungültig"
7. CardTerminated „Karte nicht mehr verwendbar"
8. MemoryFailure „Karte defekt"

[<=]

Der Systemprozess PL_TUC_CARD_VERIFY_PIN führt eine kartenbasierte Benutzerverifikation durch. Dazu wird auf einer SmartCard der TI eine PIN-Eingabe angestoßen, über die sich ein Benutzer als Besitzer des Kartengeheimnisses authentifiziert.

2.1.3.6 PL_TUC_CARD_ACTIVATE_APPLICATION – Anwendung aktivieren

TIP1-A_6918 - Leistung zum Aktivieren einer Anwendung
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Sichtbarmachen einer Anwendung auf einer SmartCard als Plattformleistung PL_TUC_CARD_ACTIVATE_APPLICATION gemäß [gemSpec_CardProxy] *cardOperation für Ordner* mit dem Aktionsparameter *ACTIVATE* umsetzen.**[<=]**

TIP1-A_6919 - Aufrufparameter zum Aktivieren einer Anwendung
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_ACTIVATE_APPLICATION umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der Anwendung gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6920 - Ergebnis der Leistung zur Aktivieren einer Anwendung
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_ACTIVATE_APPLICATION umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Anwendung erfolgreich aktviert"
2. ObjectNotFound „IDENTIFIKATOR ungültig"
3. CardTerminated „Karte nicht mehr verwendbar"
4. MemoryFailure „Karte defekt"
5. ObjectTerminated „Objekt nicht mehr verwendbar"

6. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
7. UpdateRetryWarning „Aktion erfolgreich, Speicher mglw. defekt“

[<=]

Der Systemprozess PL_TUC_CARD_ACTIVATE_APPLICATION schaltet eine verborgene Anwendung auf einer SmartCard sichtbar.

2.1.3.7 PL_TUC_CARD_DEACTIVATE_APPLICATION – Anwendung deaktivieren

TIP1-A_6921 - Leistung zum Deaktivieren einer Anwendung
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Verbergen einer Anwendung auf einer SmartCard als Plattformleistung PL_TUC_CARD_DEACTIVATE_APPLICATION gemäß [gemSpec_CardProxy] *cardOperation für Ordner* mit dem Aktionsparameter DE*ACTIVATE* umsetzen.**[<=]**

TIP1-A_6922 - Aufrufparameter zum Deaktivieren einer Anwendung
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_DEACTIVATE_APPLICATION umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der Anwendung gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6923 - Ergebnis der Leistung zur Deaktivieren einer Anwendung
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_DEACTIVATE_APPLICATION umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Anwendung erfolgreich deaktviert“
2. ObjectNotFound „IDENTIFIKATOR ungültig“
3. CardTerminated „Karte nicht mehr verwendbar“
4. MemoryFailure „Karte defekt“
5. ObjectTerminated „Objekt nicht mehr verwendbar“
6. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
7. UpdateRetryWarning „Aktion erfolgreich, Speicher mglw. defekt“

[<=]

Mit dem Systemprozess PL_CAR_DEACTIVATE_APPLICATION wird eine Anwendung auf einer SmartCard verborgen.

2.1.3.8 PL_TUC_CARD_GET_CHALLENGE – Auslesen einer Zufallszahl

TIP1-A_6924 - Leistung zum Auslesen einer Zufallszahl
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Auslesen einer Zufallszahl gemäß [gemSpec_Krypt#2.2 Zufallszahlengeneratoren] als Plattformleistung PL_TUC_GET_CHALLENGE umsetzen. Bei Verwendung einer SmartCard MUSS dies gemäß [gemSpec_CardProxy] mittels *cardOperation für Ordner* mit dem Aktionsparameter *GETRANDOM* und dem IDENTIFIKATOR „*****“ (Wildcard) erfolgen. Bei Verwendung eines HSM MUSS dies unter Verwendung der durch das HSM bereitgestellten Zufallszahlengenerierung erfolgen.
[<=]

TIP1-A_6925 - Aufrufparameter für das Auslesen einer Zufallszahl
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_GET_CHALLENGE unter Verwendung einer SmartCard umsetzen, MÜSSEN vom Nutzer die Längenangabe *LENGTH* der auszulesenden Zufallszahl gemäß [gemSpec_CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[**<=**]**

TIP1-A_6926 - Ergebnis des Auslesens einer Zufallszahl
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_GET_CHALLENGE umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK + Daten „Aktion erfolgreich ausgeführt"

[<=**]**

Mit dem Systemprozess PL_TUC_GET_CHALLENGE kann eine Zufallszahl ausgelesen werden. Bei Verwendung einer elektronischen Gesundheitskarte genügt die Qualität der Zufallszahl zur Ableitung ephemerer Schlüsselparameter.

2.1.3.9 PL_TUC_CARD_READ_FILE – Lesen von Daten aus einer SmartCard

TIP1-A_6927 - Leistung zum Lesen einer Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Lesen des Inhalts einer Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_READ_FILE gemäß [gemSpec_CardProxy] *cardOperation für transparente Elementary Files* mit dem Aktionsparameter *READ* umsetzen.**[**<=**]**

TIP1-A_6928 - Aufrufparameter für das Lesen einer Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_READ_FILE umsetzen, MÜSSEN vom Nutzer den *IDENTIFIKATOR* der zu lesenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[**<=**]**

TIP1-A_6929 - Optionale Parameter für das Lesen einer Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_READ_FILE umsetzen, MÜSSEN die vom Nutzer optional bereitgestellten Parameter *OFFSET* und *LENGTH* bei Vorhandensein entgegennehmen und diese in der Umsetzung von [gemSpec_CardProxy] *cardOperation* verwenden, um die zu lesende Datenmenge zu beschränken.**[**<=**]**

TIP1-A_6930 - Ergebnis des Lesens des Inhalts einer Datei
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_READ_FILE umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK + Dateiinhalt „Daten wurden erfolgreich gelesen"
2. OffsetTooBig „OFFSET ungültig"
3. CorruptDataWarning + Dateiinhalt „Daten gelesen, Speicher mglw. defekt "
4. ObjectNotFound „IDENTIFIKATOR ungültig"
5. CardTerminated „Karte nicht mehr verwendbar"
6. SecurityStatusNotSatisfied „Aktion nicht erlaubt"

[<=**]**

Mit dem Systemprozess PL_TUC_CARD_READ_FILE werden Daten aus einer transparenten Datei einer SmartCard gelesen. Über die Parameter Offset und Length kann gesteuert werden, ab welcher Position in der Datei eine festgelegte Anzahl Bytes gelesen werden. Fehlen diese Parameter, wird der komplette Dateiinhalt ausgelesen.

2.1.3.10 PL_TUC_CARD_WRITE_FILE – Schreiben von Daten auf eine SmartCard

TIP1-A_6931 - Leistung zum Schreiben von Daten in eine transparente Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Schreiben von Daten in eine transparente Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_WRITE_FILE gemäß [gemSpec_CardProxy] *cardOperation für transparente Elementary Files* mit dem Aktionsparameter *UPDATE* und dem *OFFSET = 0* umsetzen.**[<=]**

TIP1-A_6932 - Aufrufparameter für das Schreiben einer transparenten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_WRITE_FILE umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zu schreibenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] sowie *NEWDATA* entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6933 - Ergebnis des Schreibens von Datei in eine transparente Datei
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_WRITE_FILE umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Daten erfolgreich geschrieben“
2. DataTooBig „Länge von NEWDATA ungültig“
3. MemoryFailure „Karte defekt”
4. ObjectNotFound „IDENTIFIKATOR ungültig”
5. CardTerminated „Karte nicht mehr verwendbar”
6. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
7. UpdateRetryWarning „Daten geschrieben, Speicher mglw. defekt“

[<=]

Mit dem Systemprozess PL_TUC_CARD_WRITE_FILE werden Binärdaten in eine transparente Datei einer SmartCard geschrieben. Die Schreiboperation fügt die neuen Daten an eventuell vorhandene Daten an.

2.1.3.11 PL_TUC_CARD_UPDATE_FILE – Aktualisieren von Daten in einer transparenten Datei einer SmartCard

TIP1-A_6934 - Leistung zum Aktualisieren von Daten in einer transparenten Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Aktualisieren von Daten in einer transparenten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_UPDATE_FILE gemäß [gemSpec_CardProxy] *cardOperation für transparente Elementary Files* mit dem Aktionsparameter *UPDATE* umsetzen.**[<=]**

TIP1-A_6935 - Aufrufparameter zum Aktualisieren von Daten in einer transparenten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_UPDATE_FILE umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zu aktualisierenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] sowie *NEWDATA* entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6936 - Optionaler Parameter für das Aktualisieren von Datei in einer transparenten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_READ_FILE umsetzen, MÜSSEN den vom Nutzenden optional bereitgestellten Parameter *OFFSET* bei Vorhandensein entgegennehmen und diesen in der Umsetzung von [gemSpec_CardProxy] *cardOperation* verwenden, um die Startposition der Schreiboperation innerhalb der Datei festzulegen.**[<=]**

TIP1-A_6937 - Ergebnis der Aktualisierung von Daten einer transparenten Datei
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_UDPATE_FILE umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Daten erfolgreich geschrieben"
2. DataTooBig „Länge von NEWDATA ungültig"
3. OffsetTooBig „OFFSET ungültig"
4. MemoryFailure „Karte defekt"
5. ObjectNotFound „IDENTIFIKATOR ungültig"
6. CardTerminated „Karte nicht mehr verwendbar"
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
8. UpdateRetryWarning „Daten geschrieben, Speicher mglw. defekt"

[<=]

Mit dem Systemprozess PL_TUC_CARD_UPDATE_FILE werden Binärdaten in eine transparente Datei einer SmartCard geschrieben, so dass vorhandene Daten überschrieben werden. Über den Parameter Offset kann gesteuert werden, ab welcher Position in der Datei die neuen Daten geschrieben werden. Fehlt dieser Parameter, beginnt die Schreiboperation am Anfang der Datei.

2.1.3.12 PL_TUC_CARD_DELETE_FILE – Löschen von Daten auf einer SmartCard

TIP1-A_6938 - Leistung des Löschens einer transparenten Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Löschen einer transparenten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_DELETE_FILE gemäß [gemSpec_CardProxy] *cardOperation für transparente Elementary Files* mit dem Aktionsparameter *DELETE* umsetzen.**[<=]**

TIP1-A_6939 - Aufrufparameter zum Löschen einer transparenten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_DELETE_FILE umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zu löschenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6940 - Ergebnis der Löschoperation einer transparenten Datei
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_DELETE_FILE umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Datei erfolgreich gelöscht"
2. MemoryFailure „Karte defekt"
3. ObjectNotFound „IDENTIFIKATOR ungültig"
4. CardTerminated „Karte nicht mehr verwendbar"
5. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
6. UpdateRetryWarning „Aktion erfolgreich, Speicher mglw. defekt"

[<=]

Der Systemprozess PL_TUC_CARD_DELETE_FILE entfernt eine transparente Datei auf einer SmartCard samt Dateiinhalt. Die gelöschte Datei ist im Anschluss nicht mehr adressierbar.

2.1.3.13 PL_TUC_CARD_ERASE_FILE – Rücksetzen des Inhalts einer transparenten Datei

Der Systemprozess PL_TUC_CARD_ERASE_FILE entfernt den Inhalt einer transparenten Datei. Die adressierte Datei ist weiterhin verwendbar.

TIP1-A_6941 - Leistung zum Rücksetzen einer transparenten Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Rücksetzen des Dateiinhalts einer transparenten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_ERASE_FILE gemäß [gemSpec_CardProxy] *cardOperation für transparente Elementary Files* mit dem Aktionsparameter *ERASE* umsetzen.**[<=]**

TIP1-A_6942 - Aufrufparameter zum Rücksetzen einer transparenten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_ERASE_FILE umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zurückzusetzenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6943 - Optionaler Parameter für das Rücksetzen des Inhalts einer transparenten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_ERASE_FILE umsetzen, MÜSSEN den vom Nutzenden optional bereitgestellten Parameter *OFFSET* bei Vorhandensein entgegennehmen und dieses in der Umsetzung von [gemSpec_CardProxy] *cardOperation* verwenden, um die Startposition der Operation innerhalb der Datei festzulegen.**[<=]**

TIP1-A_6944 - Ergebnis des Rücksetzens einer transparenten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_ERASE_FILE umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Daten erfolgreich gelöscht"
2. OffsetTooBig „OFFSET ungültig"
3. MemoryFailure „Karte defekt"
4. ObjectNotFound „IDENTIFIKATOR ungültig"

5. CardTerminated „Karte nicht mehr verwendbar”
6. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
7. UpdateRetryWarning „Daten gelöscht, Speicher mglw. defekt“

[<=]

2.1.3.14 PL_TUC_CARD_READ_RECORD – Lesen von Daten aus einer strukturierten Datei

Mit dem Systemprozess PL_TUC_CARD_READ_RECORD werden Daten aus einer strukturierten Datei auf einer SmartCard ausgelesen. Über die optionale Angabe der recordNumber wird gesteuert, ob nur ein einzelner Record oder alle Records der strukturierten Datei gelesen werden sollen.

TIP1-A_6945 - Leistung zum Lesen einer strukturierten Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Lesen einer strukturierten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_READ_RECORD gemäß [gemSpec_CardProxy] *cardOperation für strukturierte Elementary Files* mit dem Aktionsparameter *READ* umsetzen.**[<=]**

TIP1-A_6946 - Aufrufparameter für das Lesen einer strukturierten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_READ_RECORD umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zu lesenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6947 - Optionale Parameter für das Lesen einer strukturierten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_READ_RECORD umsetzen, MÜSSEN den vom Nutzenden optional bereitgestellten Parameter *RECORDNUMBER* bei Vorhandensein entgegennehmen und diese in der Umsetzung von [gemSpec_CardProxy] *cardOperation* verwenden, um die zu lesende Datenmenge zu beschränken.**[<=]**

TIP1-A_6948 - Ergebnis des Lesens einer strukturierten Datei
Produkttypen und Dienste der TI die die Plattformleistung PL_TUC_CARD_READ_RECORD umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK +Recordliste „Daten wurden erfolgreich gelesen“
2. CorruptDataWarning +Recordliste „Daten gelesen, Speicher mglw. defekt ”
3. ObjectNotFound „IDENTIFIKATOR ungültig”
4. RecordNotFound „RECORDNUMBER ungültig“
5. RecordDeactivated „Datensatz[RECORDNUMBER] deaktiviert“
6. CardTerminated „Karte nicht mehr verwendbar”
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt“

[<=]

2.1.3.15 PL_TUC_EGK_READ_PROTOCOL – Auslesen des Zugriffprotokolls der eGK

Mit dem Systemprozess PL_TUC_EGK_READ_PROTOCOL wird das gesamte Zugriffsprotokoll auf der elektronischen Gesundheitskarte ausgelesen. Im Gegensatz zur

generischen Leseoperation eines strukturierten Elementary Files wird in diesem Baustein der Zugriff auf die Karte durch die Kartenzugriffsschicht CardProxy optimiert und es werden alle Log-Einträge (maximal 50) in einer Liste zurückgegeben.

TIP1-A_6949 - Leistung zum Lesen des Zugriffprotokolls auf der eGK
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Auslesen des Zugriffprotokolls auf der eGK als Plattformleistung PL_TUC_EGK_READ_PROTOCOL gemäß [gemSpec_CardProxy] *cardOperation für strukturiertes Elementary File* mit dem Aktionsparameter *READ* und dem IDENTIFIKATOR *EF.Logging* gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy eGK G2] umsetzen.**[**<=**]**

TIP1-A_6996 - Aufbereitung Zugriffsprotokolleinträge
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_EGK_READ_PROTOCOL umsetzen, MÜSSEN alle aus der Karte gelesenen, binär-codierten Zugriffsprotokolleinträge gemäß [gemSpec_Karten_Fach_TIP#Tab_Karten_Fach_TIP_010_StrukturEF.Logging] in ein strukturiertes Format überführen und die Werte entsprechend des angegeben Datentyps decodieren.**[**<=**]**

TIP1-A_6950 - Ergebnis des Auslesens des Zugriffprotokolls der eGK
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_EGK_READ_PROTOCOL umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK + Liste/Zugriffsprotokoll „Daten wurden erfolgreich gelesen"
2. CorruptDataWarning + Liste „Daten gelesen, Speicher mglw. defekt "
3. ObjectNotFound „IDENTIFIKATOR ungültig"
4. CardTerminated „Karte nicht mehr verwendbar"
5. SecurityStatusNotSatisfied „Aktion nicht erlaubt"

[<=**]**

2.1.3.16 PL_TUC_CARD_WRITE_RECORD – Schreiben von Daten in eine strukturierte Datei

Der Systemprozess PL_TUC_CARD_WRITE_RECORD schreibt einen Datensatz in einen Record einer strukturierten Datei auf einer SmartCard. Enthält der zu schreibende Record bereits Daten, wird der alte Datensatz mit dem neuen Wert überschrieben.

TIP1-A_6951 - Leistung zum Schreiben von Daten in eine strukturierte Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Schreiben eines Records einer strukturierten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_WRITE_RECORD gemäß [gemSpec_CardProxy] *cardOperation für strukturierte Elementary Files* mit dem Aktionsparameter *UPDATE* umsetzen.**[**<=**]**

TIP1-A_6952 - Aufrufparameter zum Schreiben von Daten in eine strukturierte Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_WRITE_RECORD umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zu aktualisierenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy], die *RECORDNUMBER* sowie NEWDATA entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[**<=**]**

TIP1-A_6953 - Ergebnis der Schreiboperation in einer strukturierten Datei
Produkttypen und Dienste der TI die die Plattformleistung PL_TUC_CARD_WRITE_RECORD umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Datensatz erfolgreich geschrieben"
2. UpdateRetryWarning „Daten geschrieben, Speicher mglw. defekt "
3. WrongRecordLength „Länge von NEWDATA ungültig"
4. ObjectNotFound „IDENTIFIKATOR ungültig"
5. RecordNotFound „RECORDNUMBER ungültig"
6. RecordDeactivated „Datensatz[RECORDNUMBER] deaktiviert"
7. CardTerminated „Karte nicht mehr verwendbar"
8. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
9. OutOfMemoryError „Speicherplatz in Zieldatei zu klein"
10. MemoryFailure „Karte defekt"
11. BufferTooSmall „Kartenkommando zu lang"

[<=]

2.1.3.17 PL_TUC_CARD_APPEND_RECORD – Anfügen von Daten an eine strukturierte Datei

Mit dem Systemprozess PL_TUC_CARD_APPEND_RECORD wird ein Datensatz als neuer Record in einer strukturierten Datei an das Ende angefügt.

TIP1-A_6954 - Leistung zum Anfügen von Daten in einer strukturierten Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Anfügen eines Records in einer strukturierten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_APPEND_RECORD gemäß [gemSpec_CardProxy] *cardOperation für strukturierte Elementary Files* mit dem Aktionsparameter *APPEND* umsetzen.**[<=]**

TIP1-A_6955 - Aufrufparameter zum Anfügen von Daten in einer strukturierten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_APPEND_RECORD umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zu aktualisierenden Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] sowie *RECORDDATA* entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6956 - Ergebnis der Anfügeoperation in einer strukturierten Datei
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_CARD_APPEND_RECORD umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Datensatz erfolgreich angefügt"
2. UpdateRetryWarning „Daten angefügt, Speicher mglw. defekt "
3. WrongRecordLength „Länge von RECORDDATA ungültig"
4. ObjectNotFound „IDENTIFIKATOR ungültig"
5. FullRecordList „Kein zusätzlicher Record in Zieldatei zulässig"

6. CardTerminated „Karte nicht mehr verwendbar”
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
8. OutOfMemoryError „Speicherplatz in Zieldatei zu klein“
9. MemoryFailure „Karte defekt“
10. BufferTooSmall „Kartenkommando zu lang“

[<=]

2.1.3.18 PL_TUC_EGK_APPEND_PROTOCOL – Zugriff auf der eGK protokollieren

Mit dem Systemprozess PL_TUC_EGK_APPEND_PROTOCOL wird ein höherwertiger Baustein für das Schreiben eines Zugriffsprotokoll-Eintrags auf die eGK definiert. Nutzern dieser Plattformleistung genügt es, beim Aufruf den Identifikator der zu protokollierenden Fachanwendung mit der Art des durch die Fachanwendung erfolgten Zugriffs mitzuteilen. Der Systemprozess erzeugt aus diesen Daten zusammen mit den Angaben des Karteninhabers der SM-B-AUT-Identität, der diese eGK in einem Card-2-Card-Verfahren mit einem CV-Zertifikat freigeschaltet hat, einen Protokolldatensatz. Für das Protokollieren auf der eGK nutzt der Systemprozess die Schreiboperation des CardProxy der eGK.

TIP1-A_6957 - Leistung zum Protokollieren des eGK-Zugriffs
Produkttypen und Dienste, welche Systemprozesse der TI mit Zugriff auf die eGK realisieren, MÜSSEN das Hinzufügen eines Protokolleintrags auf der eGK als Plattformleistung PL_TUC_EGK_APPEND_PROTOCOL umsetzen.**[<=]**

TIP1-A_6958 - Aufrufparameter der Zugriffsprotokollierung
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_EGK_APPEND_PROTOCOL umsetzen, MÜSSEN vom Nutzenden die Protokollparameter

1. DATATYPE [1 Byte] „Identifikator der Fachanwendung“
2. ACCESSTYPE [1 Byte] „Identifikator der Zugriffsart“

entgegennehmen.**[<=]**

TIP1-A_6959 - Hinzufügen eines Protokolleintrags auf die eGK
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_EGK_APPEND_PROTOCOL umsetzen, MÜSSEN die Schritte zum Hinzufügen eines Protokolleintrags auf der eGK in der angegebenen Reihenfolge durchführen:

	Teilschritt Hinzufügen eines Protokolleintrags	Teilergebnis

1	Auslesen des `commonName`, `surName` und `givenName` aus dem zur Erzeugung eines Protokolleintrags auf der eGK vorgesehenen Zertifikats in PL_TUC_CARD_INFORMATION, sofern vorhanden; alternativ: Auslesen des `commonName`, `surName` und `givenName` des C.HCI.AUT-Zertifikats in PL_TUC_CARD_INFORMATION der zur Initialisierung des eGK-CardProxy verwendeten SM-B-Identität gemäß [CommonPKI] und [gemSpec_PKI# Tab_PKI_229]	`commonName`, `surName` und `givenName`
2	Auslesen der `ICCSN` aus den Kartenstammdaten PL_TUC_CARD_INFORMATION der zur Initialisierung des eGK-CardProxy verwendeten SM-B-Identität	`ICCSN`
3	Zusammenfügen der folgenden Informationen zu einem Protokolldatensatz gemäß [gem_Spec_Karten_Fach_TIP#4.1 – Tabelle 11: Tab_Karten_Fach_TIP_010_StrukturEF.Logging – Struktur der Rekords der Datei EF.Logging] *RECORDDATA* := Timestamp ("jetzt" aktuelle gesetzliche Zeit) + DATATYPE + ACCESSTYPE + ICCSN + ActorName als [`commonName` | (`surname`, `givenname`)]	Protokolldatensatz (erstellt, noch nicht geschrieben)
4	Schreiben des Protokolleintrags auf die eGK mittels PL_TUC_CARD_APPEND_RECORD mit IDENTIFIKATOR = *EF.Logging* gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy eGK G2] *RECORDDATA* = Protokolldatensatz aus Schritt 3	OK **=> OK** UpdateRetryWarning WrongRecordLength ObjectNotFound FullRecordList CardTerminated SecurityStatusNotSatisfied OutOfMemoryError MemoryFailure BufferTooSmall NotEnoughtMemorySpace **=> Fehler**
5	Rückmeldung an den Nutzenden **OK** „Datensatz erfolgreich geschrieben" **Fehler** „Keine passende Freischaltkarte oder eGK-Fehler"	

[<=]

2.1.3.19 PL_TUC_CARD_DELETE_RECORD – Löschen von Daten in einer strukturierten Datei

Mit dem Systemprozess PL_TUC_CARD_DELETE_RECORD wird ein einzelner Record einer strukturierten Datei oder werden alle Records einer strukturierten Datei auf einer SmartCard gelöscht. Beim Löschen eines einzelnen Records reduziert sich die Anzahl der Records in der strukturierten Datei um eins. Werden alle Records gelöscht, ist die Anzahl der Records nach erfolgreichem Abschluss der Operation null. Die strukturierte Datei ist weiterhin adressierbar.

TIP1-A_6960 - Leistung zum Löschen von Daten in einer strukturierten Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Löschen von Daten einer strukturierten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_DELETE_RECORD gemäß [gemSpec_CardProxy] *cardOperation für strukturierte Elementary Files* mit dem Aktionsparameter *DELETERECORD* umsetzen.**[<=]**

TIP1-A_6961 - Aufrufparameter zum Löschen von Daten in einer strukturierten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_DELETE_RECORD umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der betroffenen Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[<=]**

TIP1-A_6962 - Optionale Parameter für das Löschen von Daten einer strukturierten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_DELETE_RECORD umsetzen, MÜSSEN den vom Nutzenden optional bereitgestellten Parameter *RECORDNUMBER* bei Vorhandensein entgegennehmen und diese in der Umsetzung von [gemSpec_CardProxy] *cardOperation* verwenden, um die zu löschende Datenmenge auf einen einzelnen Record zu beschränken.**[<=]**

TIP1-A_6963 - Ergebnis der Löschoperation in einer strukturierten Datei
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_CARD_DELETE_RECORD umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Datensatz erfolgreich gelöscht“
2. UpdateRetryWarning „Daten gelöscht, Speicher mglw. defekt ”
3. ObjectNotFound „IDENTIFIKATOR ungültig”
4. RecordNotFound „RECORDNUMBER ungültig”
5. RecordDeactivated „Datensatz[RECORDNUMBER] deaktiviert“
6. CardTerminated „Karte nicht mehr verwendbar”
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt“
8. MemoryFailure „Karte defekt“

[<=]

2.1.3.20 PL_TUC_CARD_ERASE_RECORD – Rücksetzen eines Datensatzes in einer strukturierten Datei

Der Systemprozess PL_TUC_CARD_ERASE_RECORD löscht den Inhalt eines einzelnen der strukturierten Datei auf einer SmartCard. Der Record sowie die gesamte strukturierte

Datei bleiben dabei erhalten. Der zurückgesetzte Record sowie die strukturierte Datei sind weiterhin adressierbar.

TIP1-A_6964 - Leistung zum Rücksetzen eines Datensatzes in einer strukturierten Datei
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Rücksetzen eines Records einer strukturierten Datei auf einer SmartCard als Plattformleistung PL_TUC_CARD_ERASE_RECORD gemäß [gemSpec_CardProxy] *cardOperation für strukturierte Elementary Files* mit dem Aktionsparameter *ERASE* umsetzen.**[**<=**]**

TIP1-A_6965 - Aufrufparameter zum Rücksetzen eines Datensatzes in einer strukturierten Datei
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_ERASE_RECORD umsetzen, MÜSSEN vom Nutzenden den *IDENTIFIKATOR* der zu betroffenen strukturierten Datei gemäß [gemSpec_CardProxy#Konfigurationstabelle CardProxy] sowie die *RECORDNUMBER* des zurückzusetzenden Datensatzes entgegennehmen und in der Umsetzung von *cardOperation* verwenden.**[**<=**]**

TIP1-A_6966 - Ergebnis des Rücksetzens eines Datensatzes in einer strukturierten Datei
Produkttypen und Dienste der TI die die Plattformleistung PL_TUC_CARD_DELETE_RECORD umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *cardOperation* zurückmelden:

1. OK „Datensatz erfolgreich zurückgesetzt"
2. UpdateRetryWarning „Daten zurückgesetzt, Speicher mglw. defekt "
3. ObjectNotFound „IDENTIFIKATOR ungültig"
4. RecordNotFound „RECORDNUMBER ungültig"
5. RecordDeactivated „Datensatz[RECORDNUMBER] deaktiviert"
6. CardTerminated „Karte nicht mehr verwendbar"
7. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
8. MemoryFailure „Karte defekt"

[<=**]**

2.1.4 Transparenter Zugriff auf eine SmartCard

Mit dem Zugriff auf eine SmartCard über einen transparenten Kanal ist es möglich, von entfernter Stelle mit der Karte zu interagieren. Über den CardProxy werden Kartenkommandos direkt an die Karte weitergeleitet und deren Antwort-APDU zurückgegeben. Weder die kapselnden Systemprozesse noch CardProxy werten den Inhalt der an die Karte gesendeten und von dort empfangenen APDUs aus. Im speziellen Fall einer verschlüsselten Kommunikation (trusted channel) zwischen der Karte und einem Server in Card-to-Server-Kommunikation ist dies ohnehin nicht möglich.

2.1.4.1 PL_TUC_CARD_TC_OPEN

Der Systemprozess PL_TUC_CARD_TC_OPEN öffnet einen transparenten Kommunikationskanal zu einer SmartCard. Mit der Nutzung dieses Plattformbausteins findet kein direkter Zugriff auf die Karte statt, es aktiviert in der Kartenzugriffsschicht

eine exklusive Nutzung der Karte für diesen transparenten Kanal. Während dieser geöffnet ist, sind ausschließlich Aktionen mit den Systemprozessen PL_TUC_CARD_TC_SEND und _CLOSE möglich.

TIP1-A_6967 - Leistung zum Öffnen eines transparenten Kanals
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Öffnen eines transparenten Kanals zu einer SmartCard als Plattformleistung PL_TUC_CARD_TC_OPEN gemäß [gemSpec_CardProxy] *Funktion transparentChannel* mit dem Aktionsparameter *OPEN* umsetzen.**[**<=**]**

TIP1-A_6968 - Ergebnis des Öffnens eines transparenten Kanals
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_TC_OPEN umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *Funktion transparentChannel* zurückmelden**:**

1. OK „Aktion erfolgreich ausgeführt"
2. TransparentChannelAlreadyOpen „Transparenter Kanal bereits offen"

[<=**]**

2.1.4.2 PL_TUC_CARD_TC_SEND

Mittels des Systemprozesses PL_TUC_CARD_TC_SEND wird ein Kartenkommando zu einer Karte weitergeleitet, ohne den Inhalt auszuwerten. Gelangt das Kartenkommando erfolgreich zur Karte wird immer das Response-Kommando der Karte zurückgegeben.

TIP1-A_6969 - Leistung der transparenten Kommunikation zur Karte
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Senden von transparenten Kartenkommandos an eine SmartCard als Plattformleistung PL_TUC_CARD_TC_SEND gemäß [gemSpec_CardProxy] *Funktion transparentChannel* mit dem Aktionsparameter *SENDAPDU* umsetzen.**[**<=**]**

TIP1-A_6970 - Aufrufparameter zur transparenten Kommunikation zur Karte
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_CARD_TC_SEND umsetzen, MÜSSEN vom Nutzenden die *COMMANDAPDU*, welche an die Karte weitergeleitet werden soll, entgegennehmen und in der Umsetzung der *Funktion transparentChannel* verwenden.**[**<=**]**

TIP1-A_6971 - Ergebnis der transparenten Kommunikation zur Karte
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_TC_SEND umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *Funktion transparentChannel* zurückmelden**:**

1. OK plus responseAPDU „Aktion erfolgreich ausgeführt"
2. MissingAPDU „Fehlendes Kartenkommando"
3. TransparentChannelNotOpen „Transparenter Kanal nicht offen"

[<=**]**

2.1.4.3 PL_TUC_CARD_TC_CLOSE

Der Systemprozess PL_TUC_CARD_TC_CLOSE schließt einen transparenten Kommunikationskanal zu einer SmartCard und gibt diese als Ressource für andere Plattformleistungen wieder frei. Mit der Nutzung dieses Plattformbausteins findet kein direkter Zugriff auf die Karte statt, es deaktiviert in der Kartenzugriffsschicht die exklusive Nutzung der Karte für diesen transparenten Kanal.

TIP1-A_6972 - Leistung zum Schließen eines transparenten Kanals
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Schließen eines transparenten Kanals zu einer SmartCard als Plattformleistung PL_TUC_CARD_TC_CLOSE gemäß [gemSpec_CardProxy] *Funktion transparentChannel* mit dem Aktionsparameter *CLOSE* umsetzen.**[**<=**]**

TIP1-A_6973 - Ergebnis des Schließens eines transparenten Kanals
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_CARD_TC_CLOSE umsetzen, MÜSSEN das Ergebnis gemäß [gemSpec_CardProxy] *Funktion transparentChannel* zurückmelden**:**

1. OK „Aktion erfolgreich, Karte zurückgesetzt"
2. TransparentChannelNotOpen „Transparenter Kanal nicht offen"

[<=**]**

2.2 Kommunikation und Vernetzung

2.2.1 PL_TUC_TLS_SECURE_CHANNEL – TLS-Verbindung mit gegenseitiger Authentisierung

Der Systemprozess PL_TUC_TLS_SECURE_CHANNEL baut eine verschlüsselte Verbindung von einem Clientsystem auf Basis einer in einem Sicherheitsmodul (z.B. HSM, SmartCard) gespeicherten Identität der TI zu einem Zielsystem her. Dazu erfolgt eine **gegenseitige Authentisierung** zwischen dem Zielsystem und dem verwendeten Sicherheitsmodul und es werden symmetrische Sitzungsschlüssel, für die verschlüsselte Kommunikation zwischen Client- und Zielsystem, ausgehandelt.

TIP1-A_6974 - Leistung zur TLS-Verbindung mit gegenseitiger Authentisierung

Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_TLS_SECURE_CHANNEL für den Aufbau einer TLS-Verbindung auf Basis einer in einem Sicherheitsmodul gespeicherten Identität umsetzen.**[**<=**]**

TIP1-A_6975 - Aufrufparameter zur TLS-Verbindung mit gegenseitiger Authentisierung

Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_TLS_SECURE_CHANNEL umsetzen, MÜSSEN vom Nutzer den *URI* des Zielsystems und den *ROLLENBEZEICHNER* der erwarteten Rolle des Zielsystems als Parameter entgegennehmen und diese im Verbindungsaufbau verwenden.**[**<=**]**

TIP1-A_6976 - Aufbau der TLS-Verbindung mit gegenseitiger Authentisierung

Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_TLS_SECURE_CHANNEL umsetzen, MÜSSEN die Schritte zum Aufbau einer TLS-

Verbindung auf Basis einer in einem Sicherheitsmodul gespeicherten Identität in der angegebenen Reihenfolge durchführen:

	Teilschritt TLS-Verbindungsaufbau	**Teilergebnis**
1	Auflösung des FQDN in der URI der Adresse des Zielsystems über PL_TUC_NET_NAME_RESOLUTION	IP-Adresse Sonst: Der FQDN kann nicht aufgelöst werden **=> Fehler**

2	Aufbau einer TLS-Verbindung vom Client- zum Zielsystem gemäß der Festlegungen des TLS-Protokolls in [RFC-5246] und den gematik-spezifischen Ergänzungen in [gemSpec_Krypt#3.3.2 TLS-Verbindungen] und [gemSpec_Krypt#5.5 ECC-Unterstützung bei TLS] Folgende zusätzliche Festlegungen gelten für den Verbindungsaufbau gemäß TLS-Protokoll in [RFC-5246]:
	• Falls erforderlich, Auslesen einer Zufallszahl aus einem im Zugriff befindlichen Sicherheitsmodul gemäß Systemprozess PL_TUC_ GET_CHALLENGE
	• Prüfung des vom Zielsystem bereitgestellten Serverzertifikat C.FD.TLS-S auf Gültigkeit mittels PL_TUC_PKI_VERIFY_CERTIFICATE mit folgenden Parametern:
	  • Zu prüfendes Zertifikat: C.FD.TLS-S
	  • Referenzzeitpunkt: „jetzt“ (aktuelle gesetzliche Zeit)
	  • PolicyList: oid_fd_tls_s
	  • KeyUsage: mindestens digitalSignature
	  • ExtendedKeyUsage: id-kp-serverAuth
	  • OCSP-Graceperiod: NULL oder default
	  • Offline-Modus: „nein“
	  • OCSP-Response NULL
	  • Timeout: default
	  • TOLERATE_OCSP_FAILURE: default
	• Wird vom Nutzer ein ROLLENBEZEICHNER gemäß [gemSpec_OID] übergeben, Abgleich zwischen diesem und der von PL_TUC_PKI_VERIFY_CERTIFICATE zurückgegebenen Rolle des C.FD.TLS-S-Zertifikats des Zielsystems
	• Clientauthentisierung gegenüber dem Zielsystem mit der Inhaberidentität C.HCI.AUT der SM-B-Identität. Bei Verwendung einer SmartCard ist diese aus PL_TUC_CARD_INFORMATION zu entnehmen. Bei Verwendung eines HSMs ist die Identität aus dem HSM zu entnehmen.
	• Signatur der ephemeren Schlüssel im TLS-Protokoll (Kontext: Diffie-Hellman Schlüssel signieren) mittels PL_TUC_SIGN_HASH_nonQES, dem IDENTIFIKATOR des privaten Schlüssels PrK.HCI.AUT.R2048 bzw. PrK.HCI.AUT.E256 des zuvor ermittelten C.HCI.AUT-Zertifikats (bei SmartCard gemäß [gemSpec_CardProxy] in PL_TUC_CARD_INFORMATION gespeichert) und
		aufgebaute TLS-Verbindung
		
		Sonst:
		Ist das Zielsystem nicht erreichbar, schlägt der Verbindungsaufbau fehl
		=> Fehler
		Ist das Serverzertifikat gemäß TUC_PKI_018 **mathematisch** oder **zeitlich ungültig** oder meldet die erfolgreiche **Online**prüfung die **Sperrung** des Zertifikats („revoked“)**,** wird der Verbindungsaufbau wird abgelehnt
		=> Fehler
		ROLLENBEZEICHNER und **Rolle des Serverzertifikats** passen nicht zueinander, der Verbindungsaufbau wird abgelehnt
		=> Fehler
		Gegenseitige Authentisierung fehlgeschlagen, Verbindungsaufbau abgebrochen
		=> Fehler

	dem gewählten kryptografischen Verfahren R2048 bzw. E256 sowie dem entsprechenden SIGNATURVERFAHREN	

3	Rückmeldung an den Nutzer **OK** „Verbindungsaufbau erfolgreich“ **Fehler** „Verbindungaufbau nicht erfolgreich“	

[<=]

2.2.2 PL_TUC_NET_NAME_RESOLUTION

Mit dem Systemprozess PL_TUC_NET_NAME_RESOLUTION wird ein URI einer Netzwerkkomponente der TI mittels des Namensdienstes der zentralen TI-Plattform in eine IP-Adresse aufgelöst.

TIP1-A_6977 - Auflösen von URI in IP-Adresse
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, SOLLEN eine Auflösung von Netzwerk-URI in IP-Adresse als Plattformleistung PL_TUC_NET_NAME_RESOLUTION über die Schnittstelle I_DNS_Name_Resolution zum TI-Namensdienst gemäß [gemSpec_Net#Namensdienst] anbieten.**[<=]**

2.2.3 PL_TUC_NET_SYNC_TIME

Über den Systemprozess PL_TUC_NET_SYNC_TIME können sich Dienste und Komponenten der Telematikinfrastruktur mit dem Zeitserver der zentralen TI-Plattform synchronisieren.

TIP1-A_6978 - Synchronisierung mit Zeitdienst
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Zeitsynchronisation als Plattformleistung PL_TUC_NET_SYNC_TIME über die Schnittstelle I_NTP_Time_Informationen zum Zeitdienst der Telematikinfrastruktur gemäß [gemSpec_Net#Zeitdienst] umsetzen und diese TI-Zeit als gültige, gesetzliche Zeit betrachten.**[<=]**

2.3 Zugriffe auf den Verzeichnisdienst

Über die im Folgenden beschriebenen Systemprozesse können Zugriffe auf den Verzeichnisdienst der zentralen TI-Plattform durchgeführt werden.

2.3.1 PL_TUC_VZD_BIND - Verbindung aufbauen

A_17431 - Leistung zum Verbindungsaufbau zum VZD
Produkttypen und Dienste der TI, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_VZD_BIND für den Aufbau einer Verbindung zum Verzeichnisdienst der zentralen TI-Plattform umsetzen.
[<=]

A_17445 - Aufbau der Verbindung zum VZD
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_VZD_BIND umsetzen, MÜSSEN die Schritte zum Aufbau einer Verbindung zum Verzeichnisdienst der

zentralen TI-Plattform in der angegebenen Reihenfolge durchführen:

	Teilschritt des Verbindungsaufbaus	Teilergebnis
1	Ermittlung des FQDN und Port des VZD durch eine DNS-SD Namensauflösung gemäß [RFC6763] mit dem Bezeichner "_ldap._tcp.vzd.<DNS_TOP_LEVEL_DOMAIN_TI>."	FQDN und Port des VZD
2	Aufbau einer LDAPS-Verbindung zum VZD. Dabei wird das Serverzertifikat des Verzeichnisdiensts C.ZD.TLS-S nach TUC_PKI_018 geprüft (PolicyList: oid_vzd_ti (gemäß gemSpec_OID), intendedKeyUsage: intendedKeyUsage(C.ZD.TLS-S), ExtendedKeyUsages: serverAuth (1.3.6.1.5.5.7.3.1), Offlinemodus: nein, TOLERATE_OCSP_FAILURE: false , Prüfmodus: OCSP	erfolgreich aufgebaute LDAPS-Verbindung zum VZD. Sonst: Fehler sind ggf. gemäß [RFC-4511#Appendix A] zu behandeln und als ERROR an den Nutzer zu übergeben.

[<=]

2.3.2 PL_TUC_VZD_SEARCH - Verzeichnis abfragen

A_17432 - Leistung zur Abfrage des VZD
Produkttypen und Dienste der TI, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_VZD_SEARCH für die Abfrage des Verzeichnisdienst der zentralen TI-Plattform gemäß [gemSpec_VZD#Schnittstelle I_Directory_Query] umsetzen.
[<=]

A_17448 - Aufrufparameter der Verzeichnisabfrage
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_VZD_SEARCH umsetzen, MÜSSEN vom Nutzer den *SEARCH_REQUEST* als Aufrufparameter entgegennehmen und in der Umsetzung als LDAPv3 Search Request gemäß [RFC-4511#4.5.1] über die Schnittstelle `I_Directory_Query` an den Verzeichnisdienst der zentralen TI-Plattform senden.
[<=]

A_17449 - Ergebnis der Verzeichnisabfrage
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_VZD_SEARCH umsetzen, MÜSSEN die LDAPv3 Search Response gemäß [RFC-4511#4.5.2] vom Verzeichnisdienst empfangen und als *SEARCH_RESPONSE* an den Nutzer übergeben. Fehler MÜSSEN ggf. gemäß [RFC-4511#Appendix A] als *ERROR* an den Nutzer übergeben und behandelt werden.

[<=]

2.3.3 PL_TUC_VZD_UNBIND - Verbindung trennen

A_17446 - Leistung zur Verbindungstrennung zum VZD
Produkttypen und Dienste der TI, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_VZD_UNBIND für die Trennung einer Verbindung zum Verzeichnisdienst der zentralen TI-Plattform umsetzen.**[<=]**

A_17465 - Trennen der Verbindung zum VZD
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_VZD_UNBIND umsetzen, MÜSSEN in der Umsetzung einen LDAPv3 Unbind Request gemäß [RFC-4511#4.3] an den Verzeichnisdienst der zentralen TI-Plattform senden. Fehler MÜSSEN ggf. gemäß [RFC-4511# Appendix A] als *ERROR* an den Nutzer übergeben und behandelt werden.
[<=]

2.3.4 PL_TUC_VZD_ABANDON - Verzeichnisabfrage abbrechen

A_17447 - Leistung zum Abbrechen einer Verzeichnisabfrage
Produkttypen und Dienste der TI, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_VZD_ABANDON für den Abbruch einer Abfrage des Verzeichnisdienstes der zentralen TI-Plattform umsetzen.**[<=]**

A_17468 - Abbrechen einer Verzeichnisabfrage
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_VZD_ABANDON umsetzen, MÜSSEN in der Umsetzung einen LDAPv3 Abandon Request gemäß [RFC-4511#4.11] an den Verzeichnisdienst der zentralen TI-Plattform senden. Fehler MÜSSEN ggf. gemäß [RFC-4511# Appendix A] als *ERROR* an den Nutzer übergeben und behandelt werden.
[<=]

2.4 Vertraulichkeit, Authentizität, Integrität

2.4.1 PL_TUC_SIGN_HASH_nonQES – mit TI-Identität nonQES signieren

Der Systemprozess PL_TUC_SIGN_HASH_nonQES versieht einen übergebenen Hash-Wert mit einer auf einer TI-Identität basierenden digitalen nonQES Signatur. Dazu wird unter Verwendung eines Sicherheitsmoduls (SmartCard, HSM) oder Signaturdienstes und einer auf dem Sicherheitsmodul bzw. dem Signaturdienst gespeicherten Identität der TI ein Binärwert signiert.

TIP1-A_6979 - Leistung der nonQES-Signatur

Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Signieren eines Hashwertes mit einer TI-Identität als Plattformleistung PL_TUC_SIGN_HASH_nonQES umsetzen. Bei Verwendung einer SmartCard MUSS dies gemäß [gemSpec_CardProxy] *cardOperation für private Schlüsselobjekte* erfolgen. Bei Verwendung eines HSMs oder eines Signaturdienstes MUSS dies gemäß [gemSpec_HSMProxy#logische Operation *sign* für Signaturen] bzw. [gemSpec_SigD#Operationsdefinition I_Remote_Sign_Operations::sign_Data] erfolgen.
[<=]

TIP1-A_6980 - Aufrufparameter der nonQES-Signatur

Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_SIGN_HASH_nonQES umsetzen, MÜSSEN vom Nutzer den *IDENTIFIKATOR* des privaten Schlüssels der TI-Identität, das *SIGNATURVERFAHREN gemäß* [gemSpec_Krypt#3.7 Signatur binärer Inhaltsdaten (Dokumente)/5.7.2 ECDSA-Signaturen im CMS-Format] sowie den zu signierenden *HASHWERT* als Aufrufparameter entgegennehmen und in der Umsetzung verwenden.
Bei Nutzung einer SmartCard MUSS der *IDENTIFIKATOR* gemäß [gemSpecCardProxy#Konfigurationstabelle CardProxy] verwendet werden, *SIGNATURVERFAHREN* und *HASHWERT* MÜSSEN als Aktionsparameter bzw. Eingangsparameter von *cardOperation* gemäß [gemSpec_CardProxy] verwendet werden.
Bei Nutzung eines Signaturdienstes oder eines HSM MÜSSEN *IDENTIFIKATOR* und der *HASHWERT* als Aufrufparameter Identifier bzw. Data gemäß [gemSpec_SigD#Operationsdefinition I_Remote_Sign_Operations::sign_Data] bzw. Identity und Data gemäß [gemSpec_HSMProxy#logische Operation *sign* für Signaturen] übergeben werden.

[<=]

TIP1-A_6981 - Ergebnis der nonQES-Signatur

Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_SIGN_HASH_nonQES umsetzen, MÜSSEN das Ergebnis und ggf. Fehler an die nutzende Komponente zurückmelden:

- bei Verwendung einer Karte gemäß [gemSpec_CardProxy] *cardOperation:*
  1. OK + Hashsignatur „Signatur erfolgreich erstellt"
  2. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
  3. ObjectNotFound „IDENTIFIKATOR ungültig"
  4. KeyInvalid „Schlüsselobjekt nicht verwendbar"
  5. CardTerminated „Karte nicht mehr verwendbar"
  6. WrongToken „Übergabeparameter fehlerhaft"
- bei Verwendung eines Signaturdienstes gemäß [gemSpec_SigD#Operationsdefinition I_Remote_Sign_Operations::sign_Data]:
  1. SignedData die Hashsignatur
  2. Certificate das dem verwendeten privaten Schlüssel entsprechende Zertifikat
  3. Bei Fehlern Weitergabe des Fehlers an die nutzende Komponente und Behandlung
- bei Verwendung eines HSM gemäß [gemSpec_HSMProxy]:
  - im Erfolgsfall: signatur
  - im Fehlerfall: error.

[<=]

2.4.2 PL_TUC_HYBRID_ENCIPHER – Hybrid verschlüsseln

Der Systemprozess PL_TUC_HYBRID_ENCIPHER führt eine hybride Verschlüsselung eines Dokuments für ein oder mehrere Empfängerzertifikate durch. Dazu muss zunächst ein symmetrischer Schlüssel erzeugt werden, mit dem das Eingabedokument verschlüsselt wird. Dieser symmetrische Schlüssel wird anschließend mit dem öffentlichen Schlüsselmaterial der Dokumentenempfänger (bereitgestellt über X.509v3-Zertifikate) verschlüsselt und an das Dokument angefügt.

TIP1-A_6982 - Leistung zum hybriden Verschlüsseln
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_HYBRID_ENCIPHER zum hybriden Verschlüsseln eines Dokuments umsetzen.**[<=]**

TIP1-A_6983-01~~TIP1-A_6983~~ - Aufrufparameter zum hybriden Verschlüsseln
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_HYBRID_ENCIPHER umsetzen, MÜSSEN vom Nutzer die Aufrufparameter

1. Doc „"das zu verschlüsselnde Dokument"
2. {Cert(i)} „"Menge der Empfänger-/Ziel-Zertifikate"
3. Attribute "zusätzliche Attribute (optional)"

entgegennehmen.
[<=]

TIP1-A_6984-02~~TIP1-A_6984~~ - Ablauf der hybriden Verschlüsselung eines Dokuments
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_HYBRID_ENCYPHER umsetzen, MÜSSEN die Schritte zum Verschlüsseln eines gegebenen Dokuments in der angegebenen Reihenfolge durchführen:

	Teilschritt der hybriden Verschlüsselung	Teilergebnis
1	Erzeugung eines symmetrischen Schlüssels gemäß BSI-TR-03116-1#3.5 Schlüsselerzeugung] und den Festlegungen in [gemSpec_Krypt#3.5.1 Hybride Verschlüsselung] ->S_{symm}	Symmetrischer Schlüssel Falls Symmetrischer Schlüssel nicht erzeugt werden kann **=> Fehler**
2	Dokument Doc mit symmetrischem Schlüssel S_{symm} verschlüsseln -> Doc_{enc} Falls Doc ein XML-Dokument/Fragment ist: XMLEnc: Es MÜSSEN die Vorgaben aus [gemSpec_Krypt#3.1.4] beachtet werden. Sonst: CMS Es MÜSSEN die Vorgaben aus [gemSpec_Krypt#3.5.1] beachtet werden.	symmetrisch verschlüsseltes Dokument

3	Für jedes Empfängerzertifikat Cert(i) Schlüssel S_{symm} mit öffentlichem Schlüssel S_{public} der Empfängeridentität (liegt in Zertifikat Cert(i)) gemäß Vorgaben aus gemSpec_Krypt#3.1.5] (XML, RSA) bzw. [gemSpec_Krypt#3.5.2] (CMS, RSA) oder gemäß Vorgaben aus [gemSpec_Krypt#5.~~8~~7] (XML/CMS, ECC) verschlüsseln -> $(S_{symm})_{enc(i)}$	pro Empfängerzertifikat: mit öffentlichem Schlüssel der Empfängeridentität verschlüsselter symmetrischer Schlüssel
4a	XMLEnc: Alle verschlüsselten Dokumentenschlüssel $\{(S_{symm})_{enc(i)}\}$ als EncryptedKey und mit dem verschlüsselten Dokument Doc_{enc} zu einem EncryptedData-Element gemäß [XML-Enc 1.1] zusammenfügen: Doc_{enc} + $\{(S_{symm})_{enc(i)}\}$ + Attribute -> D Pro Empfängerzertifikat wird ein Element EncryptedKey/KeyInfo/X509Data/X509Certificate base64-kodiert und darin DER-kodiert abgelegt.	zusammengefügtes XML-ENC-EncryptedData-Element
4b	CMS: Alle verschlüsselten Dokumentenschlüssel $\{(S_{symm})_{enc(i)}\}$ und das verschlüsselte Dokument sind in einem Authenticated-Enveloped-Data Content Type gemäß [RFC-5083] und [RFC-5084] ~~zu einem CMS-Dokument [RFC-5652#6.1 EnvelopedData] zusammenfügen~~zusammenzuführen: Doc_{enc} + $\{(S_{symm})_{enc(i)}\}$ + Attribute -> D Bei Verschlüsselung des „content-encryption key" wird „key transport" verwendet Pro Empfängerzertifikat wird eine KeyTransRecipientInfo erzeugt, für RecipientIdentifier wird die Option IssuerAndSerialNumber verwendet ContentType = OID {… authEnvelopedData} = 1.2.840.113549.1.9.16.1.23	zusammengefügtes CMS-Dokument
5	Rückmeldung an den Aufrufenden, entweder 1. OK + verschlüsseltes Dokument D oder 2. Fehler	

[<=]

2.4.3 PL_TUC_HYBRID_DECIPHER – Hybrid entschlüsseln

Der Systemprozess PL_TUC_HYBRID_DECIPHER entschlüsselt ein hybrid verschlüsseltes Dokument. Dazu wird zunächst der verschlüsselte Dokumentenschlüssel aus dem Eingabedokument extrahiert und mit einem privaten Schlüssel des Empfängers entschlüsselt. Die Speicherung und Nutzung des privaten Schlüssels ist dabei bevorzugt unter Verwendung eines Sicherheitsmoduls (SmartCard, HSM) durchzuführen. Mit dem

wiederhergestellten Dokumentenschlüssel wird anschließend das Dokument in einem symmetrischen Verfahren entschlüsselt.

TIP1-A_6985 - Leistung zum hybriden Entschlüsseln
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_HYBRID_DECIPHER zum hybriden Entschlüsseln eines Dokuments umsetzen.**[<=]**

TIP1-A_6986 - Aufrufparameter zum hybriden Entschlüsseln
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_HYBRID_DECIPHER umsetzen, MÜSSEN vom Nutzer die Aufrufparameter

1. D „das verschlüsselte Dokument"
2. Id "(Identität des) Empfänger" (sofern nicht implizit eindeutig)

entgegennehmen.
[<=]

TIP1-A_6987 - Ablauf der hybriden Entschlüsselung eines Dokuments
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_HYBRID_DECYPHER umsetzen, MÜSSEN die Schritte zum Entschlüsseln eines verschlüsselten Dokuments in der angegebenen Reihenfolge durchführen:

	Teilschritt der hybriden Entschlüsselung	**Teilergebnis**
1a	Das übergebene verschlüsselte Dokument D ist ein XML Fragment vom Typ EncryptedData: Einlesen der übergebenen Daten (Dokument D und ggf. Empfänger Id) und Identifikation der verschiedenen Komponenten und Parameter gemäß [XMLEnc-1.1]: D -> Doc_{enc} + $(S_{symm})_{enc}$ + Attribute Die Informationen zum Auffinden des privaten Empfängerschlüssels stehen in <EncryptedKey>, bspw. das Empfängerzertifikat in EncryptedKey/KeyInfo/X509Data/X509Certificate base64-kodiert und darin DER-kodiert	Verschlüsseltes Dokument, verschlüsselter Dokumentenschlüssel und Empfängeridentität aus übergebenem Dokument Sonst: Verschlüsseltes Dokument, verschlüsselter Dokumentenschlüssel oder Empfängeridentität kann nicht bestimmt werden **=> Fehler**
1b	Sonst: Einlesen der übergebenen Daten (Dokument D und ggf. Empfänger Id) und Identifikation der verschiedenen Komponenten und Parameter gemäß [RFC-5652#6.1 EnvelopedData]: D -> Doc_{enc} + $(S_{symm})_{enc}$ + Attribute, insbesondere werden die RecipientInfos als KeyTransRecipientInfo-Angaben benötigt. Die Angabe des Schlüsselverschlüsselungsalgorithmus ist in [RFC-5652#6.2.1	Verschlüsseltes Dokument, verschlüsselter Dokumentenschlüssel und Empfängeridentität aus übergebenem Dokument Sonst:

	KeyTransRecipientInfo::KeyEncryptionAlgorithmIdentifier] und im Cryptogram als verschlüsselter Dokumentenschlüssel gemäß [RFC-5652#6.2.1 KeyTransRecipientInfo::EncryptedKey] enthalten, welcher in der Liste der Dokumentenempfänger anhand der KeyTransRecipientIdentifier::IssuerAndSerialNumber das Empfänger-Zertifikat identifiziert wird.	Verschlüsseltes Dokument, verschlüsselter Dokumentenschlüssel oder Empfängeridentität kann nicht bestimmt werden **=> Fehler**
2a	Bei Verwendung einer SmartCard: Entschlüsselung des Dokumentenschlüssels mittels CardProxy [gemSpec_CardProxy] *cardOperation für private Schlüsselobjekte* mit dem Aktionsparameter entweder • RSA: rsaDecipherOaep oder • ECC: elcSharedSecretCalculation Das Empfängerzertifikat kann über IssuerAndSerialNumber gegen das ENC.Zertifikat in den Kartenstammdaten in PL_TUC_CARD_INFORMATION geprüft werden, der dazugehörige private Schlüssel muss gemäß [gemSpec_CardProxy#Konfigurationstabelle] als IDENTIFIKATOR übergeben werden $(S_{symm})_{enc}$ -> S_{symm}	Entschlüsselter Dokumentenschlüssel Sonst: Auf der Karte befindet sich kein ENC.Zertifikat des angegebenen Empfängers mit zugehörigem privaten Schlüssel **=> Fehler**
2b	Sonst: Entschlüsselung des Dokumentenschlüssels unter Verwendung des zum angegebenen Empfänger gehörenden privaten Schlüssels. Bei Verwendung eines HSM gemäß [gemSpec_HSMProxy#logische Operation decrypt für Entschlüsselung] durchzuführen mit • $(S_{symm})_{enc}$ als cipher. • Id oder EncryptedKey/KeyInfo bzw. KeyTransRecipientInfo als identity $(S_{symm})_{enc}$ -> S_{symm}	Entschlüsselter Dokumentenschlüssel Sonst: Es ist kein privater Schlüssel für den Empfänger verfügbar **=> Fehler**
3	Dokument mit entschlüsseltem symmetrischem Schlüssel S_{symm} entschlüsseln Doc_{enc} -> Doc	Entschlüsseltes Dokument

4	Rückmeldung an den Aufrufenden, entweder 1. OK + unverschlüsseltes Dokument oder 2. Fehler	

[<=]

2.4.4 PL_TUC_SYMM_ENCIPHER – Symmetrisch verschlüsseln

Der Systemprozess PL_TUC_SYMM_ENCIPHER führt eine symmetrische Verschlüsselung eines Dokuments durch. Dazu können zusammen mit dem Dokument ein Schlüssel und associatedData (beide optional) übergeben werden. Falls kein Schlüssel übergeben wird, wird ein symmetrischer Schlüssel erzeugt.

A_14970 - Leistung zum symmetrischen Verschlüsseln
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_SYMM_ENCIPHER zum symmetrischen Verschlüsseln eines Dokuments umsetzen.**[<=]**

A_14971 - Aufrufparameter zum symmetrischen Verschlüsseln
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_SYMM_ENCIPHER umsetzen, MÜSSEN vom Nutzer die Aufrufparameter

1. Doc „der zu verschlüsselnde XML-Text"
2. Cert <optional> Symmetrischer Schlüssel (AES,256 Bit, gemäß[gemSpec_Krypt#A_17872] und [gemSpec_Krypt#A_18004])
3. AD <optional> associatedData, für Authenticated Encryption with Associated Data (AEAD)

entgegennehmen.

[<=]

A_14972 - Ablauf des symmetrischen Verschlüsseln eines Dokuments
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_SYMM_ENCYPHER umsetzen, MÜSSEN die Schritte zum Verschlüsseln eines gegebenen Dokuments in der angegebenen Reihenfolge durchführen:

	Teilschritt des symmetrischen Verschlüsseln	**Teilergebnis**
1	<optional, wenn kein Schlüssel übergeben wurde> Erzeugung eines symmetrischen Schlüssels gemäß [gemSpec_Krypt#GS-A_4367] und den Festlegungen in [gemSpec_Krypt#A_17872] und [gemSpec_Krypt#A_18004] -> S_{symm}	Symmetrischer Schlüssel Sonst: Symmetrischer Schlüssel kann nicht erzeugt werden **=> Fehler**

2	Dokument mit symmetrischem Schlüssel S_{symm} verschlüsseln -> Doc_{enc} gemäß [gemSpec_Krypt#A_17872]und [gemSpec_Krypt#A_18004]	mit symmetrischem Schlüssel verschlüsseltes Dokument
3	Rückmeldung an den Aufrufenden mit 1. OK + verschlüsseltes Dokument Doc_{enc} oder 2. OK + verschlüsseltes Dokument Doc_{enc} + erzeugter symmetrischer Schlüssel S_{symm}, oder 3. Fehler	

[<=]

2.4.5 PL_TUC_SYMM_DECIPHER – Symmetrisch entschlüsseln

Der Systemprozess PL_TUC_SYMM_DECIPHER entschlüsselt ein symmetrisch verschlüsseltes Dokument.

A_14982 - Leistung zum symmetrischen Entschlüsseln
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Plattformleistung PL_TUC_SYMM_DECIPHER zum symmetrischen Entschlüsseln eines Dokuments umsetzen.**[<=]**

A_14983 - Aufrufparameter zum symmetrischen Entschlüsseln
Produkttypen und Dienste der TI, die die Plattformleistung PL_TUC_SYMM_DECIPHER umsetzen, MÜSSEN vom Nutzer die Aufrufparameter

1. Doc_{enc} „das zu entschlüsselnde Dokument“
2. S_{symm} Symmetrischer Schlüssel
3. AD <optional> associatedData, für Authenticated Encryption with Associated Data (AEAD)

entgegennehmen.

[<=]

A_14984 - Ablauf des symmetrischen Entschlüsselns eines Dokuments
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_SYMM_DECYPHER umsetzen, MÜSSEN die Schritte zum Entschlüsseln eines verschlüsselten Dokuments in der angegebenen Reihenfolge durchführen:

	Teilschritt des symmetrischen Entschlüsseln	**Teilergebnis**

1	Dokument mit einer Chiffrat-Struktur gemäß [gemSpec_Krypt#A_18004] (Punkt 2) gemäß des kryptographischen Verfahrens aus [gemSpec_Krypt#A_17872]mit entschlüsseltem symmetrischem Schlüssel S_{symm} entschlüsseln(ggf. unter Verwendung der associatedData AD) Doc_{enc} -> Doc	entschlüsseltes Dokument
2	Rückmeldung an den Aufrufenden, entweder 1. OK + unverschlüsseltes Dokument Doc, oder 2. Fehler	

[<=]

2.4.6 PL_TUC_SIGN_DOCUMENT_nonQES – Dokument nonQES signieren

Der Systemprozess PL_TUC_SIGN_DOCUMENT_nonQES versieht ein übergebenes Dokument mit einer auf einer TI-Identität basierenden digitalen nonQES Signatur. Dazu wird unter Verwendung eines Sicherheitsmoduls (SmartCard, HSM) und einer darauf gespeicherten Identität der TI ein Dokument signiert.

A_17376 - Leistung der nonQES Dokumenten-Signatur
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN das Signieren eines Dokuments mit einer TI-Identität als Plattformleistung PL_TUC_SIGN_DOCUMENT_nonQES umsetzen. Bei Verwendung einer SmartCard MUSS dies gemäß [gemSpec_CardProxy] cardOperation für private Schlüsselobjekte erfolgen. Bei Verwendung eines HSMs MUSS dies gemäß [gemSpec_HSMProxy#logische Operation sign für Signaturen] erfolgen.
[<=]

A_17377 - Aufrufparameter der nonQES Dokumenten-Signatur
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_SIGN_DOCUMENT_nonQES umsetzen, MÜSSEN vom Nutzer den *IDENTIFIKATOR* des privaten Schlüssels der TI-Identität, das *SIGNATURVERFAHREN* gemäß [gemSpec_Krypt#3.1.1 XML-Signaturen für nicht-qualifizierte Signaturen, 3.7 Signatur binärer Inhaltsdaten (Dokumente)] (RSA) bzw. [gemSpec_Krypt#5.7.1 ECDSA-Signaturen im XML-Format, 5.7.2 ECDSA-Signaturen im CMS-Format] (ECC) sowie das zu signierende *DOKUMENT* und den *DOKUMENTENTYP* (XML, CMS) als Aufrufparameter entgegennehmen und in der Umsetzung verwenden. Das DOKUMENT MUSS gemäß DOKUMENTENTYP für die Signatur vorbereitet werden, dabei ist der zu signierende HASHWERT zu ermitteln.
Bei Nutzung einer SmartCard MUSS der *IDENTIFIKATOR* gemäß [gemSpecCardProxy#Konfigurationstabelle CardProxy] verwendet werden, das *SIGNATURVERFAHREN sowie der HASHWERT* MÜSSEN als Aktionsparameter bzw. Eingangsparameter von *cardOperation* gemäß [gemSpec_CardProxy] verwendet werden. Bei Nutzung eines HSMs MUSS die Verwendung der genannten Aufrufparameter für Identity und Data gemäß [gemSpec_HSMProxy#logische Operation *sign* für Signaturen] erfolgen.
[<=]

A_17380 - Ergebnis der nonQES Dokumenten-Signatur
Produkttypen und Dienste der TI die eine Plattformleistung PL_TUC_SIGN_DOCUMENT_nonQES umsetzen, MÜSSEN das Ergebnis und ggf. Fehler an die nutzende Komponente zurückmelden:

- bei Verwendung einer Karte gemäß [gemSpec_CardProxy] *cardOperation*:

1. OK + Hashsignatur „Signatur erfolgreich erstellt"
2. SecurityStatusNotSatisfied „Aktion nicht erlaubt"
3. ObjectNotFound „IDENTIFIKATOR ungültig"
4. KeyInvalid „Schlüsselobjekt nicht verwendbar"
5. CardTerminated „Karte nicht mehr verwendbar"
6. WrongToken „Übergabeparameter fehlerhaft"

- bei Verwendung eines HSM gemäß [gemSpec_HSMProxy]

- im Erfolgsfall: signatur
- im Fehlerfall: error

[<=]

2.4.7 PL_TUC_VERIFY_DOCUMENT_nonQES - nonQES Dokumentensignatur verifizieren

Der Systemprozess PL_TUC_VERIFY_DOCUMENT_nonQES überprüft die nonQES Signatur eines gegebenen Dokuments im Format XML oder CMS, unter Verwendung eines zusammen mit dem Dokument gegebenen X.509-Zertifikates der PKI der Telematikinfrastruktur.

A_17559 - Leistung zur Prüfung der nonQES Dokumentensignatur
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine nonQES Signaturprüfung eines Dokumentes im Format XML oder CMS als Plattformleistung PL_TUC_PKI_VERIFY_DOCUMENT_nonQES umsetzen.
[<=]

A_17561 - Aufrufparameter zur Prüfung der nonQES Dokumentensignatur
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_PKI_VERIFY_DOCUMENT_nonQES umsetzen, MÜSSEN vom Nutzer die folgenden Parameter entgegennehmen und in der Umsetzung verwenden:

1. SIGNED_DOCUMENT das signierte Dokument im Format XML gemäß [XMLDSig] oder CMS gemäß [RFC5652]
2. CERTIFICATE X.509-Signaturzertifikat, eingebettet im Dokument
3. SIGNATURE die Signatur des Dokumentes, eingebettet im Dokument oder getrennt ("detached")
4. TIME_REFERENCE Referenzzeitpunkt für Gültigkeitsprüfung

[<=]

A_17562 - Ablauf der Prüfung der nonQES Dokumentensignatur
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_PKI_VERIFY_DOCUMENT_nonQES umsetzen, MÜSSEN die Schritte zur Prüfung der Signatur eines Dokuments in der angegebenen Reihenfolge durchführen:

	Teilschritt der Prüfung	Teilergebnis
1	**"CoreValidation":** Es erfolgt die mathematische Prüfung der SIGNATURE bestehend aus der Prüfung der Hash-Kette bis zum signierten Hashwert und der Prüfung der kryptographischen Signatur unter Verwendung des öffentlichen Schlüssels, des Signaturwertes und des signierten Hashwertes. XML-Signatur: Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation. CMS-Signatur: Die Core Validation erfolgt entsprechend Cryptographic Message Syntax (CMS) Kapitel 5.6 Signature Verification Process [RFC5652].	Prüfergebnis der mathematischen Signaturprüfung falls keine Signatur ermittelbar oder Signatur ungültig **=> Fehler**
2	**„CheckSignatureCertificate“:** a) Signaturzertifikat (CERTIFICATE) aus dem Dokument entnehmen: XML-Signatur: Das Signaturzertifikat (CERTIFICATE) ist im XMLDSig Element `ds:KeyInfo/ds:X509Data` gespeichert [XMLDSig] oder wird als Eingangsparameter übergeben. CMS-Signatur: Das Signaturzertifikat (CERTIFICATE) für CAdES ist im Feld `certificates` im `SignedData` Container gespeichert [CAdES] oder wird als Eingangsparameter übergeben. b) Signaturzeitpunkt bestimmen: XML-Signatur: Das XML element `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 7.2.1 XAdES [XAdES]. CMS-Signatur: Das Attribut `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 11.3 CMS [CMS]. c) Durchführung der Zertifikatsprüfung: Aufrufen des Systemprozesses	Prüfergebnis der Zertifikatsprüfung falls Zertifikat nicht ermittelt werden kann oder Zertifikat ungültig ist **=> Fehler**

	PL_TUC_PKI_VERIFY_CERTIFICATE mit diesen Parametern: - Zertifikat = CERTIFICATE - Referenzzeitpunkt = TIME_REFERENCE - PolicyList = oid_smc_b_osig - KeyUsage = nonRepudiation - ExtendedKeyUsage = (leer) - Offline-Modus = nein	
3	**Prüfergebnis zurückgeben:** Die Ergebnisse aus den Schritten "CoreValidation" und „CheckSignatureCertificate“ werden an die nutzende Komponente zurück gegeben. Dabei wird unterschieden: - Die Signatur wurde gemäß den Regeln für die nonQES geprüft und für gültig befunden - Die Signatur ist ungültig - Die Signatur konnte nicht geprüft werden Bei Fehlern ist ggf. zu unterscheiden: - keine Signatur vorhanden/ermittelbar - kein Zertifikat vorhanden/ermittelbar - Die Signatur ist ungültig	

[<=]

2.5 Leistungen der PKI

2.5.1 PL_TUC_PKI_VERIFY_CERTIFICATE – Prüfung eines Zertifikats der TI

Der Systemprozess PL_TUC_PKI_VERIFY_CERTIFICATE kapselt die Prüfung eines X.509-Zertifikats der PKI der Telematikinfrastruktur. Es wird die zeitliche Gültigkeit zu einem Referenzzeitpunktes sowie die mathematische Gültigkeit geprüft. Zusätzlich kann via Parameter eine Online-Prüfung des Sperrstatus des Zertifikats verlangt werden.

TIP1-A_6991 - Leistung zur Prüfung eines Zertifikats in der TI
Produkttypen und Dienste, welche Systemprozesse der TI realisieren, MÜSSEN eine Zertifikatsprüfung als Plattformleistung PL_TUC_PKI_VERIFY_CERTIFICATE umsetzen.**[<=]**

TIP1-A_6992 - Aufrufparameter der Zertifikatsprüfung in der TI
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_PKI_VERIFY_CERTIFICATE umsetzen, MÜSSEN vom Nutzer die folgenden Parameter entgegennehmen und in der Zertifikatsprüfung verwenden:

1. Zu prüfendes Zertifikat ein Zertifikat der PKI der TI
2. EECertificateContainedInTSL optional (default: false)
   - true: Prüfung, ob ein EE-Zertifikat, in der TSL vorhanden und zeitlich gültig ist
   - false: Prüfung eines X.509-Zertifikats gemäß [gemSpec_PKI#TUC_PKI_018]

nur relevant, wenn EECertificateContainedInTSL=false:

3. Referenzzeitpunkt Prüfung auf Gültigkeit zu Referenzzeitpunkt
4. PolicyList zulässige Zertifikatstyp-OIDs
5. KeyUsage Anwendungsfall für kryptografisches Material
6. ExtendedKeyUsage Anwendungsfall für kryptografisches Material
7. OCSP-Graceperiod default: 10 Min
8. Offline-Modus ja/nein (wenn nein, dann Prüfmodus: OCSP)
9. OCSP-Response optional
10. Timeout: default: 10 Sek.
11. TOLERATE_OCSP_FAILURE: ja/nein, default: nein

[<=]

A_18072 - Ablauf der Zertifikatsprüfung in der TI
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_PKI_VERIFY_CERTIFICATE umsetzen, MÜSSEN die Schritte zur Prüfung eines Zertifikats in der angegebenen Reihenfolge durchführen:

	Teilschritt der Prüfung
1	Falls EECertificateContainedInTSL=false: Durchführung der Zertifikatsprüfung gemäß [gemSpec_PKI#TUC_PKI_018]
2	Falls EECertificateContainedInTSL=true: a) Prüfen, ob das übergebene Zertifikat in der TSL unter dem TypeIdentifier "http://uri.etsi.org/TrstSvc/Svctype/unspecified" in einem "TSPService"-Eintrag identisch aufzufinden ist. b) Prüfung der zeitlichen Gültigkeit des Zertifikats mittels [gemSpec_PKI#TUC_PKI_002], auf Basis der aktuellen Systemzeit als Referenzzeit. c) Ermitteln der Rolle des Zertifikats mittels [gemSpec_PKI#TUC_PKI_009]
3	Rückgabe von Prüfergebnis und Rückgabewerten

[<=]

TIP1-A_6993 - Ergebnis der Zertifikatsprüfung in der TI
Produkttypen und Dienste der TI, die eine Plattformleistung PL_TUC_PKI_VERIFY_CERTIFICATE umsetzen, MÜSSEN das Ergebnis jedes Prüfkriteriums und Fehler in der Zertifikatsprüfung zurückmelden:

Gültigkeit zu Referenzzeitpunkt	**„zeitlich gültig / ungültig / Prüffehler“** gemäß [gemSpec_PKI#TUC_PKI_002] bzw. [gemSpec_PKI#TUC_PKI_018]
Nicht relevant bei EECertificateContainedInTSL = true:	**„mathematisch gültig / ungültig / Prüffehler“**

Mathematische Gültigkeit:	gemäß [gemSpec_PKI#TUC_PKI_018]
Nicht relevant bei EECertificateContainedInTSL = true: OCSP-Prüfung:	**„Online gültig / Online gesperrt / Onlinestatus unbekannt / Prüffehler“** gemäß [gemSpec_PKI#TUC_PKI_018]
Rolle	**Rolle des Zertifikats** gemäß [gemSpec_PKI#TUC_PKI_009]

Fehler in der Verarbeitung beeinflussen die Prüfergebnisse wie folgt:

1. CERT_READ_ERROR, das Zertifikat kann nicht geprüft werden

Nur relevant bei EECertificateContainedInTSL = false:

2. CA_CERT_MISSING oder AUTHORITYKEYID_DIFFERENT, das Zertifikat darf nicht als gültig betrachtet werden, da kein gültiges Ausstellerzertifikat gefunden wurde.
3. OCSP_CERT_MISSING oder OCSP_SIGNATURE_ERROR, die Legitimität einer OCSP-Response kann nicht verifiziert werden, die OCSP-Prüfung muss abgebrochen werden und das Zertifikat ist nicht online-gültig
4. CERTHASH_EXTENSION_MISSING, CERTHASH_MISSMATCH, WARNING_CERT_UNKNOWN, die OCSP-Prüfung ist nicht erfolgreich und das Zertifikat ist nicht online-gültig

Nur relevant bei EECertificateContainedInTSL = true:

5. EE_CERT_NOT_FOUND, das Zertifikat konnte nicht in der TSL gefunden werden

[<=]

1585 # 3 Anhang A – Verzeichnisse

1586 ## 3.1 Abkürzungen

Kürzel	Erläuterung
APDU	Application Protocol Data Unit
eGK	elektronische Gesundheitskarte
HBA	Heilberufsausweis
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PL	Plattformleistung
PUK	Personal Unblocking Key
QES	qualifizierte elektronische Signatur
SM-B	Security Module B, Sammelbegriff für SMC-B und HSM-B
TI	Telematikinfrastruktur
TUC	Technical Use Case

1587 ## 3.2 Glossar

Begriff	Erläuterung
Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.
Sicherheitsmodul (Engl. Security Module)	Physikalischer Träger kryptographischer Geheimnisse, insbesondere zu Identitäten (z.B. zugehörige Schlüssel).

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

3.3 Abbildungsverzeichnis

3.4 Tabellenverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

3.5 Referenzierte Dokumente

3.5.1 Dokumente der gematik

Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument referenzierten Dokumente der gematik zur Telematikinfrastruktur. Der mit der vorliegenden Version korrelierende Entwicklungsstand dieser Konzepte und Spezifikationen wird pro Release in einer Dokumentenlandkarte definiert; Version und Stand der referenzierten Dokumente sind daher in der nachfolgenden Tabelle nicht aufgeführt. Deren zu diesem Dokument jeweils gültige Versionsnummern sind in der aktuellen, von der gematik veröffentlichten Dokumentenlandkarte enthalten, in der die vorliegende Version aufgeführt wird.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Glossar der Telematikinfrastruktur
[gemSpec_CardProxy]	gematik: Übergreifende Spezifikation Card Proxy
[gemSpec_SigD]	gematik: Spezifikation Signaturdienst
[gemSpec_HSMProxy]	gematik: Übergreifende Spezifikation HSM-Proxy
[gemSpec_Karten_Fach_TIP]	gematik: Befüllvorschriften für die Plattformanteile der Karten der TI

[gemSpec_Krypt]	gematik: Übergreifende Spezifikation Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur
[gemSpec_Net]	gematik: Übergreifende Spezifikation Netzwerk
[gemSpec_OID]	gematik: Spezifikation Festlegung von OIDs
[gemSpec_PKI]	gematik: Spezifikation PKI
[gemSpec_VZD]	gematik: Spezifikation Verzeichnisdienst

1609

1610 ### 3.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[CAdES]	ETSI: Electronic Signature Formats, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V2.2.1, 2008-07, via http://www.etsi.org
[RFC-4511]	Network Working Group (Juni 2006): Lightweight Directory Access Protocol (LDAP): The Protocol, https://tools.ietf.org/html/rfc4511
[RFC-5083]	Network Working Group (November 2007): Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type, https://tools.ietf.org/html/rfc5083
[RFC-5084]	Network Working Group (November 2007): Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS), https://tools.ietf.org/html/rfc5084
[RFC-5246]	Network Working Group (August 2008): The Transport Layer Security (TLS) Protocol Version 1.2, https://tools.ietf.org/html/rfc5246
[RFC-5652]	Network Working Group (September 2009): Cryptographic Message Syntax (CMS), https://tools.ietf.org/html/rfc5652

[RFC-6763]	Internet Engineering Task Force (Februar 2013): DNS-Based Service Discovery, https://tools.ietf.org/html/rfc6763
[XAdES]	European Telecommunications Standards Institute (ETSI): Technical Specication XML Advanced Electronic Signatures (XAdES). ETSI Technical Specication TS 101 903, Version 1.4.2, 2010
[XMLDSig]	W3C Recommendation (06.2008): XML-Signature Syntax and Processing http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/
[XMLEnc-1.1]	XML Encryption Syntax and Processing, W3C Recommendation 11 April 2013, http://www.w3.org/TR/xmlenc-core1/

1611