

Telematikinfrastruktur 2.0

Spezifikation Proof of Patient Presence (PoPP)-Service (Stufe 1)

Version:	1.0.0_CC3
Revision:	1430888
Stand:	21.11.2025
Status:	zur Abstimmung freigegeben
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemSpec_PoPP_Service

Dokumentinformationen

Änderungen zur Vorversion

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

Dokumentenhistorie

Version	Stand	Kap./Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	24.11.2025		Anpassung der Spezifikation für die erste Umsetzungsstufe	gematik

Inhaltsverzeichnis

1 Einordnung des Dokuments.....	7
1.1 Zielsetzung.....	7
1.2 Zielgruppe.....	7
1.3 Geltungsbereich.....	7
1.4 Abgrenzungen.....	8
1.5 Methodik.....	8
2 Systemüberblick.....	10
2.1 Überblick zum Ablauf eGK (Stufe 1) aus Versicherten Sicht.....	11
2.2 Überblick der Anwendungsfälle für die Ausstellung des PoPP-Token in der Stufe 1.....	11
2.3 Akteure und Rollen.....	12
2.3.1 Herstellung und Betrieb.....	12
2.3.1.1 Hersteller PoPP-Service.....	13
2.3.1.2 Anbieter PoPP-Service.....	13
2.3.1.3 gematik.....	13
2.3.1.4 Primärsystem-Hersteller.....	13
2.3.1.5 Kostenträger (KTR).....	14
2.3.2 Nutzer.....	14
2.3.2.1 Leistungserbringerinstitution (LEI).....	14
2.3.2.2 Versicherte.....	14
3 Systemkontext PoPP-Service.....	15
3.1 Produkt-Zerlegung und Außenschnittstellen.....	15
3.2 Systemkontext.....	16
3.3 Nachbarsysteme.....	18
4 Funktionale Anwendungsfälle des PoPP-Service.....	20
4.1 Übersicht der Systemanwendungsfälle für die Ausstellung des PoPP-Token.....	20
4.2 Leistungserbringerinstitution (LEI) am PoPP-Service registrieren und anmelden.....	21
4.3 Check-in in einer LEI mit eGK und Ausgabe des PoPP-Token.....	22
4.4 Import von Daten in eGK-Hash-Datenbank.....	27
5 Übergreifende Festlegungen.....	28
5.1 PoPP-Token: Nachweis des Versorgungskontexts.....	28
5.1.1 PoPP-Token-Erstellung.....	28
5.1.2 PoPP-Token Prüfung.....	32
5.2 Datenschutz und Sicherheit.....	33

5.2.1 Vertrauenswürdige Ausführungsumgebung (VAU).....	35
5.2.1.1 Allgemein.....	35
5.2.1.2 Einbinden des ZETA Guard der gematik.....	37
5.2.1.3 Informative Erläuterung zu den Zielen der VAU und den konkreten Umsetzungshinweisen.....	38
5.2.1.4 Lifecycle eines Verarbeitungskontextes.....	40
5.2.1.5 Anforderungen an das HSM.....	42
5.2.1.6 Schlüsselnutzung direkt im Verarbeitungskontext.....	43
5.2.1.7 Speicherung von Daten.....	45
5.2.1.8 Transport von Daten und Authentisierung/Authentifizierung bei Kommunikation.....	45
5.2.1.9 Protokollierung und Monitoring.....	46
5.2.1.10 Konfigurierbarkeit.....	47
5.2.1.11 Anforderungen an den Hersteller.....	47
5.2.1.12 Anforderungen an den Anbieter.....	49
5.2.1.13 Bereitstellung durch die gematik.....	52
5.3 Identitäten und Zertifikate PoPP-Service.....	52
5.3.1 Überblick.....	52
5.3.2 Algorithmus für Schlüsselpaare.....	53
5.3.3 Entity Statement.....	54
5.3.4 PoPP-Service Signaturen.....	54
5.3.5 TLS.....	54
5.3.6 CV-Zertifikat.....	55
5.3.7 TSL-Handling.....	56
5.4 ZETA Guard im PoPP-Service.....	57
5.4.1 Bereitstellung, Konfiguration und Verwendung vom ZETA Guard.....	58
5.5 Vertrauenswürdige Uhrzeit im PoPP-Service.....	58
5.6 Federation Entity Statement.....	59
6 Funktionsmerkmale.....	61
6.1 PoPP-Service - Resource Server.....	61
6.1.1 eGK-Handling.....	61
6.1.1.1 eGK-Handling, Einführung.....	61
6.1.1.2 Szenario.....	65
6.1.1.3 eGK öffnen.....	66
6.1.1.4 eGK G2 kontaktbehaftet.....	68
6.1.1.5 eGK G2 kontaktlos.....	72
6.1.1.6 eGK G3 kontaktbehaftet und kontaktlos.....	74
6.1.1.7 Prüfung des X.509-Zertifikates einer eGK.....	75
6.1.1.8 eGK-Handling Fehlercodes.....	76
6.1.1.9 eGK-Hash-Datenbank.....	77
6.1.1.9.1 Einleitung und Mengengerüst.....	77
6.1.1.9.2 Use Cases im laufenden Betrieb.....	78
6.1.1.9.3 Definition von Begriffen zur Wahrscheinlichkeit.....	83
6.1.1.9.4 Use Cases zur Befüllung durch Kostenträger.....	84
6.1.1.9.5 Weitere Anforderungen an die eGK-Hash-Datenbank.....	88
6.1.1.9.6 Anmerkungen zur Implementierung.....	93
6.1.2 Schnittstelle für Token Abrufe.....	94
6.2 PoPP-Client.....	95

6.3 Fehlerbehandlung.....	95
7 Testanforderungen.....	97
8 Betrieb.....	98
8.1 Schnittstellen und Anwendungsfälle.....	98
8.2 Leistungsanforderungen und Performance.....	98
9 Anhang A - Verzeichnisse.....	99
9.1 Abkürzungen.....	99
9.2 Glossar.....	101
9.3 Abbildungsverzeichnis.....	103
9.4 Tabellenverzeichnis.....	103
9.5 Referenzierte Dokumente.....	104
9.5.1 Dokumente der gematik.....	104
9.5.2 Weitere Dokumente.....	106
9.6 Allgemeine Erläuterungen.....	109
9.6.1 Entity Statement des PoPP-Service.....	109
9.7 Offene Punkte / Klärungsbedarf.....	110

1 Einordnung des Dokuments

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die Anforderungen an die Herstellung, den Test und den Betrieb des Produkttyps PoPP-Service in der Stufe 1 der PoPP-Lösung. Diese umfasst die Anwendungsfälle für PoPP mit der elektronischen Gesundheitskarte (eGK) in einer Leistungserbringerinstitution (LEI).

Weitere Anwendungsfälle wie PoPP mit der GesundheitsID oder mit der eGK-in-Fernversorgung bilden die Stufe 2 der PoPP-Lösung.

Der PoPP-Service ist der Serveranteil der PoPP-Lösung, wobei PoPP für "Proof of Patient Presence" steht. Der PoPP-Service erzeugt die Bestätigung eines Versorgungskontexts in Form eines kryptografisch gesicherten PoPP-Token. Dieses Token bestätigt, dass ein bestimmter Versicherter mit einer bestimmten LEI zusammengekommen ist. Dieser Versorgungskontext wird beim Zugriff von Leistungserbringern auf TI (Telematikinfrastruktur)-Fachdienste (FD), wie bspw. VSDM 2.0, "ePA für alle" oder E-Rezept, in Form eines PoPP-Token an diese übermittelt.

Neben dem PoPP-Service, der als Plattformdienst der TI 2.0 betrieben wird, tragen weitere Komponenten zur PoPP-Lösung bei:

- PoPP-Clients, die als Teil der Primärsysteme (PS) implementiert werden,
- PoPP-Module, die als Teil von Versicherten genutzte Anwendungen implementiert werden, welche bei der mobilen Nutzung der eGK oder bei Nutzung der GesundheitsID in die Kommunikation zur Erstellung von PoPP-Token eingebunden sind.
PoPP-Module werden erst für die Stufe 2 von PoPP relevant.

Die Spezifikation oder Beschreibung dieser weiteren Komponenten erfolgt in anderen Dokumenten.

1.2 Zielgruppe

Das Dokument richtet sich an Hersteller und Anbieter des PoPP-Service, der eine Spezifikation und Steckbriefe für die Stufe1 benötigt.

1.3 Geltungsbereich

Dieses Dokument enthält normative Festlegungen zur TI des deutschen Gesundheitswesens. Der Gültigkeitszeitraum der vorliegenden Version und deren Anwendung in Zulassungs- oder Abnahmeverfahren wird durch die gematik GmbH in gesonderten Dokumenten (bspw. gemPTV_ATV_Festlegungen, Produkt- oder Anbietertypsteckbrief) festgelegt und bekanntgegeben.

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

Spezifiziert werden in dem Dokument die von dem Produkttyp bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttyps beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert (siehe auch Anhang A – Verzeichnisse).

Die vollständige Anforderungslage für den Produkttyp ergibt sich aus weiteren Konzept- und Spezifikationsdokumenten, diese sind in dem Produkttypsteckbrief des Produkttyps PoPP-Service [gemProdT_PoPP_Service_PTV] verzeichnet. Für den Anbieter des PoPP-Service ist auch der Anbietertypsteckbrief für den PoPP-Service [gemAnbT_PoPP_Service_ATV] relevant.

Nicht Bestandteil des vorliegenden Dokuments sind die Festlegungen zum Themenbereich PoPP-Client, PoPP-Modul oder App-Anforderungen zur PoPP-Token Kommunikation.

1.5 Methodik

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

Da in dem Beispielsatz "Eine leere Liste DARF NICHT ein Element besitzen." die Phrase "DARF NICHT" semantisch irreführend wäre (wenn nicht ein, dann vielleicht zwei?), wird in diesem Dokument stattdessen "Eine leere Liste DARF KEIN Element besitzen." verwendet. Die Schlüsselworte werden außerdem um Pronomen in Großbuchstaben ergänzt, wenn dies den Sprachfluss verbessert oder die Semantik verdeutlicht.

Anwendungsfälle und Anforderungen werden im Dokument wie folgt dargestellt:

<AF-ID> - <Titel des Anwendungsfalles>

Text/Beschreibung

[<=]

bzw.

<AFO-ID> - <Titel der Afo>

Text/Beschreibung

[<=]

Dabei umfasst der Anwendungsfall bzw. die Anforderung sämtliche zwischen ID und Textmarke [<=] angeführten Inhalte.

Hinweis auf offene Punkte

Offener Punkt: Das Kapitel wird in einer späteren Version des Dokuments ergänzt.

2 Systemüberblick

In diesem Kapitel wird ein einführender Überblick über die PoPP-Lösung der Stufen 1 und 2 gegeben.

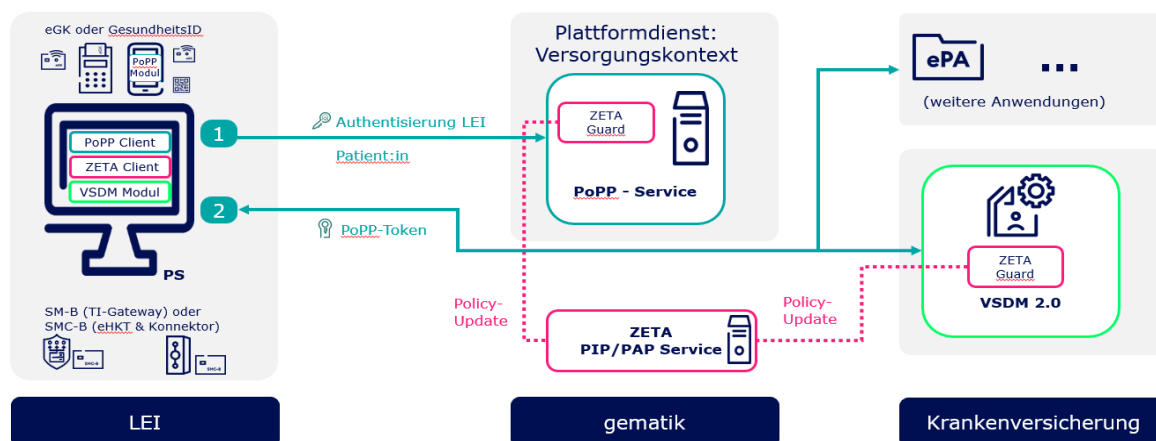


Abbildung 1: Einfacher Systemüberblick

Proof of Patient Presence (PoPP) ist ein Nachweis, der belegt, dass sich ein Versicherter zu einem bestimmten Zeitpunkt in einem Versorgungskontext mit einer bestimmten Leistungserbringerinstitution (LEI) befindet. Im kryptografisch gesicherten PoPP-Token sind somit Informationen über die LEI und über den Versicherten zusammengeführt. Dabei ist es die Aufgabe der PoPP-Lösung, die Authentifizierung der LEI durchzuführen und durch Authentifizierung des Versicherten per GesundheitsID (Stufe 2) oder Authentifizierung der elektronischen Gesundheitskarte (eGK) eines Versicherten den Versorgungskontext zu bestätigen. Das Ergebnis ist das PoPP-Token, welches der LEI zur Autorisierung für den Zugriff auf die Daten des Versicherten in Diensten der Telematikinfrastruktur (TI) dient.

Die Lösung besteht aus dem zentralen Server und verschiedenen Client Komponenten:

1. **PoPP-Service** - der als TI-Plattfordmdienst PoPP-Token für eine LEI erstellt. Die LEI authentisiert sich dafür mit der SM(C)-B. Die Identität des Versicherten wird kryptografisch gesichert ermittelt entweder über die Authentifizierung seiner eGK oder über die Authentifizierung des Versicherten via GesundheitsID (Stufe 2).
2. **Primärsystem (PS) der LEI** - als Host der Client-Seite. Das PS besteht aus den folgenden Komponenten:
 - a. Basis-PS - enthält die bisherige Funktionalität, inklusive der Verwaltung von Patientenstammdaten (PVS/KIS), sowie eine neue Businesslogik, die für die Steuerung der Praxisabläufe bei der Erstellung eines PoPP-Token benötigt wird.
 - b. PoPP-Client - enthält die Funktionalität für die Kommunikation des Basis-PS mit dem PoPP-Service. Dazu gehören die Anfrage zur Erzeugung von PoPP-Token für den eGK- und den GesundheitsID-Pfad (Stufe 2) und die Bereitstellung des PoPP-Token durch den PoPP-Service.
Die Beschreibung und Anforderungen an den PoPP-Client finden sich in [gemILF_PoPP_Client],

- c. ZETA Client - enthält die Funktionalität zur Erzeugung und Management der für den Zero Trust Ablauf notwendigen Identitäten und Abläufe gemäß [gemSpec_ZETA],
 - d. Anwendungs-Module zur Verwendung und Steuerung von TI-Anwendungen von (VSDM 2.0, ePA, E-Rezept).
3. **PoPP-Modul für Versicherte (PoPP-Modul)** - ist ein integriertes Anwendungsmodul, mittels dessen ein Versicherter den Check-in mit seiner GesundheitsID starten kann. Die Beschreibung und Anforderungen an das PoPP-Modul finden sich in [gemSpec_PoPP_Modul]. Das PoPP-Modul ist erst für Stufe 2 relevant.

2.1 Überblick zum Ablauf eGK (Stufe 1) aus Versicherten Sicht

Der Nachweis des Versorgungskontexts (PoPP-Token) erfordert die Authentifizierung von LEI und Versicherten. Die LEI authentifiziert sich mit ihrer SMC-B. Für den Versicherten wird seine eGK authentifiziert. Dieser Authentifizierungsprozess wird als Check-in bezeichnet.

Der Versicherte übergibt die eGK in der LEI an die MFA oder den LE. Alles Weitere führt der PoPP-Client durch.

2.2 Überblick der Anwendungsfälle für die Ausstellung des PoPP-Token in der Stufe 1

Die in diesem Kapitel aufgeführten Anwendungsfälle schildern die Absichten des Nutzers in Verbindung mit dem Primärsystem und dienen als Lesehilfe zu den fachlichen Anwendungsfällen. Die Anwendungsfälle erheben keinen Anspruch auf Vollständigkeit.

Für die Stufe 1 sind zwei Anwendungsfälle relevant: **UC_PoPP_1a** und **UC_PoPP_2a**.

Tabelle 1: PoPP-Use Cases (Business Sicht)

ID	Anwendungsfälle
UC_PoPP_1a	<p>PoPP-Token bei physischer Anwesenheit in der LEI - eGK</p> <p>Ein Versicherter möchte eine Versorgung in einer LEI in Anspruch nehmen. Die LEI benötigt für den Zugriff auf die Daten des physisch anwesenden Versicherten einen Nachweis des Versorgungskontexts. Dazu wird die eGK des Versicherten an einem geeigneten Lesegerät der LEI präsentiert. Nachdem der Versicherte den Check-in-Prozess mit der eGK abgeschlossen hat, erhält die LEI im PS den notwendigen Nachweis des Versorgungskontexts.</p>
UC_PoPP_2a	<p>PoPP-Token bei physischer Anwesenheit außerhalb der LEI - eGK</p> <p>Ein Versicherter möchte eine Versorgung außerhalb einer LEI in Anspruch nehmen. Dazu kommt das Personal der LEI zum Versicherten. Die LEI benötigt für den Zugriff auf die Daten des physisch anwesenden Versicherten einen Nachweis des Versorgungskontexts. Dazu wird die eGK des Versicherten an einem geeigneten mobilen Lesegerät der LEI präsentiert. Nachdem das LEI-Personal die eGK eingelesen hat, ist der Check-in-Vorgang abgeschlossen und die LEI erhält im PS den notwendigen Nachweis des Versorgungskontexts.</p>

Hinweis: Das vom PoPP-Service erstellte PoPP-Token enthält die Information, mit welcher Methode (siehe proofMethod im Kapitel[5.1.1- PoPP-Token-Erstellung]) der Versorgungskontext nachgewiesen wurde. Somit ist es den PoPP-Token nutzenden Anwendungen und Diensten möglich, in ihrem Anwendungs- oder Dienstkontext Autorisierungsentscheidungen auch aufgrund der Prüfmethode zu treffen.

2.3 Akteure und Rollen

Der PoPP-Service wird vom Anbieter PoPP-Service im Internet betrieben. Es müssen für die verschiedenen Betriebsumgebungen (Produktivumgebung (PU), Referenzumgebung(RU)) und nach Bedarf die gemäß [gemKPT_Test] geforderten Test- und Referenzumgebungen (TU-, RU- und DEV) jeweils voneinander unabhängige Instanzen betrieben werden. Für die Absicherung gegenüber dem Internet wird der von der gematik bereitgestellte ZETA Guard verwendet.

2.3.1 Herstellung und Betrieb

Folgende Rollen und Akteure kommen bei Herstellung und Betrieb des PoPP-Service vor.

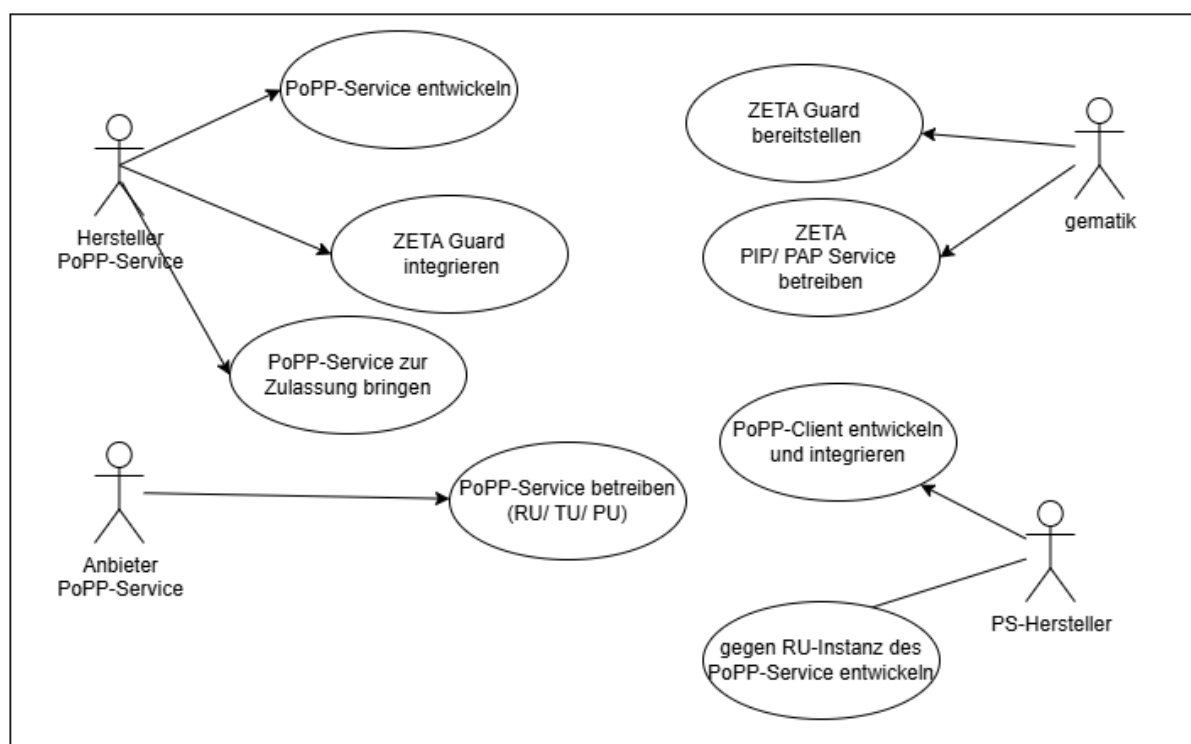


Abbildung 2: Rollen und Akteure bei Herstellung und Betrieb des PoPP-Service

2.3.1.1 Hersteller PoPP-Service

Der Hersteller des PoPP-Service implementiert und entwickelt den PoPP-Service gemäß den Vorgaben der gematik. Er erwirkt eine Produktzulassung für den PoPP-Service.

2.3.1.2 Anbieter PoPP-Service

Der Anbieter PoPP-Service verantwortet und betreibt den zugelassenen PoPP-Service gemäß den Vorgaben der gematik. Dabei muss er für die Verbindungen ins Internet, den von der gematik beigestellten ZETA Guard verwenden. Neben dem Betrieb von Instanzen für den Produktivbetrieb müssen weitere Instanzen für den Testbetrieb gemäß gemKPT_Test betrieben werden.

2.3.1.3 gematik

Die gematik spezifiziert den PoPP-Service und schreibt die Entwicklung sowie den Produktivbetrieb des PoPP-Service aus.

Die gematik unterstützt den Hersteller des PoPP-Service durch die Beistellung des ZETA Guard. Darüber hinaus betreibt die gematik die Zero Trust Komponenten Policy Information Point (PIP) und Policy Administration Point (PAP): ZETA PIP und PAP-Service.

2.3.1.4 Primärsystem-Hersteller

PS-Hersteller setzen die PoPP-Client- und ZETA Client-Funktionalität um. Sie nutzen den vom Anbieter in der RU bereitgestellten PoPP-Service, um ihre jeweilige Umsetzung zu testen.

2.3.1.5 Kostenträger (KTR)

Kostenträger übermitteln (oder beauftragen die Übermittlung von) Informationen zum Befüllen einer Datenbank an den PoPP-Service (siehe [6.1.1.9- eGK-Hash-Datenbank]). Diese Informationen werden für die Verwendung einer eGK im PoPP-Kontext benötigt.

2.3.2 Nutzer

Folgende Rollen und Akteure kommen im Betrieb der Anwendung des PoPP-Service vor.

2.3.2.1 Leistungserbringerinstitution (LEI)

Die LEI nutzen den PoPP-Service, um eine Zugriffsberechtigung (PoPP-Token) bspw. für den Abruf von Daten eines Versicherten aus seiner elektronische Patientenakte (ePA) erzeugen zu lassen. Dabei benutzt die LEI ein für die PoPP-Lösung angepasstes PS, in dem PoPP-Client-Funktionalität, ZETA Client-Funktionalität und Funktionalität für eine Fachanwendung, die das PoPP-Token verwendet, implementiert ist. Die LEI bindet bei Anwendungsfällen mit Nutzung der eGK in der LEI die eGK des Versicherten über den PoPP-Client an den PoPP-Service an. Dabei verwendet die LEI ein eH-KT oder einen Standardkartenleser. Bzgl. Kauf und Betrieb von Standardkartenlesern wird die gematik Sicherheitshinweise herausgeben, die sich an LEI richten.

2.3.2.2 Versicherte

Versicherte nutzen den PoPP-Service indirekt, wenn sie sich bei einer LEI einchecken oder auf andere Art und Weise bei der Erstellung des Versorgungskontexts zwischen LEI und Versicherten mitwirken. Die Versicherten steuern ihre Krankenversicherungsnummer (KVNR) bei, indem sie sich oder ein Vertreter mit ihrer eGK in der LEI vorstellen (Nutzung der Praxis Hardware für die Prüfung der eGK auf Authentizität),

3 Systemkontext PoPP-Service

In diesem Kapitel findet sich die logische Zerlegung des Produkttyps Proof of Patient Presence (PoPP)-Service anhand eines Komponenten-Diagramms, die Beschreibung des Systemkontexts mit allen bereitstellenden Außenschnittstellen des PoPP-Service sowie die Auflistung und kurze Beschreibung der Nachbarsysteme.

3.1 Produkt-Zerlegung und Außenschnittstellen

Der Produkttyp PoPP-Service besteht aus mehreren Komponenten. Die Komponenten für die Erstellung des PoPP-Token sind in PoPP-Service Resource Server zusammengefasst. Die Kommunikation zwischen LEI und PoPP-Service wird durch ein Zero Trust Access (ZETA) Guard abgesichert. Für die Kommunikationswege zwischen einem Versicherten und dem PoPP-Service wird ein weiterer Autorisierungsdienst, der PoPP-Service Authorization Server, benötigt, solange ein ZETA Guard diese Funktionalität noch nicht übernimmt.

Der PoPP-Service Resource Server Stufe 1 besteht aus mehreren Komponenten für die Bearbeitung von Token Request, bei denen die elektronische Gesundheitskarte (eGK) verwendet wird (eGK-Kommunikation, eGK-Hash-Datenbank, für den Support von eGK Kommunikation).

Die Komponenten für die abschließende Token-Erstellung und der sichere Schlüsselspeicher gehören ebenfalls zum Produkttyp. Ferner gehören betriebliche Komponenten für die Erfassung und Aufbereitung von Monitoring Daten für die gematik Betriebsdatenerfassung (BDE) und ein Security Monitoring dazu.

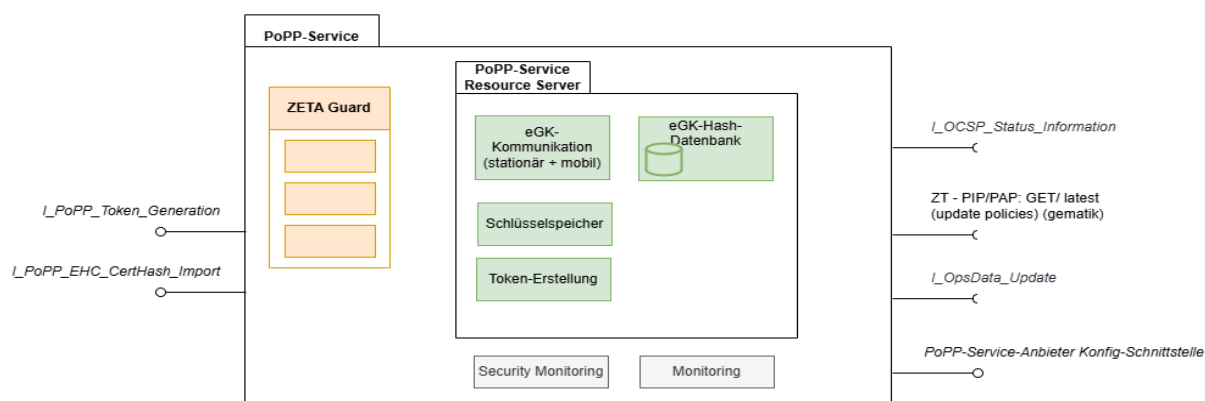


Abbildung 3: Produkttypzerlegung

Der PoPP-Service stellt folgende Außenschnittstellen für die Stufe 1 bereit:

- **I_PoPP_Token_Generation** wird vom Primärsystem einer LEI verwendet, um vom PoPP-Service einen PoPP-Token zu erlangen.
- **I_PoPP_EHC_CertHash_Import** wird von den TSPs der Kassen aufgerufen und dient der Befüllung der CertHash-Datenbank, mittels der das Mapping von CV-Zertifikaten zu X.509-Zertifikaten von eGKs erreicht wird. Dies dient der Unterstützung von Anwendungsfällen bei denen der Versicherte die eGK mit einem kontaktlosen Kartenlesegerät der LEI verwendet.

Der PoPP-Service benutzt die folgenden Schnittstellen in der Stufe 1:

- **ZETA PIP/PAP:** GET/latest (update policies) (gematik)
Wird vom ZETA Guard für die Versorgung mit den stets aktuellen Zugriffs-Policies und weiteren Konfigurationsdaten genutzt (Details siehe [gemSpec_ZETA]).
- **I_OCSF_Status_Information:**
Wird vom PoPP-Service für die Statusabfragen für eGK (PoPP-Service Resource), für SM(C)-B (ZETA Guard), für die eigenen TI 1.0 PoPP-Token-Signatur-Identität und das eigene Internet-TLS-Server-Zertifikat verwendet.
- **I_OpsData_Update:**
Wird vom PoPP-Service für die Lieferung von Betriebsdaten verwendet.

Für den PoPP-Service-Anbieter wird eine interne Schnittstelle angeboten:

- **PoPP-Service-Anbieter Konfig-Schnittstelle**
Interne Schnittstellen, die vom PoPP-Service-Anbieter genutzt werden, um den PoPP-Service zu konfigurieren.

3.2 Systemkontext

Ein Systemkontext beschreibt die Umgebung, in der ein System operiert, und die Interaktionen zwischen dem System und externen Entitäten. Für die Stufe1 gilt:

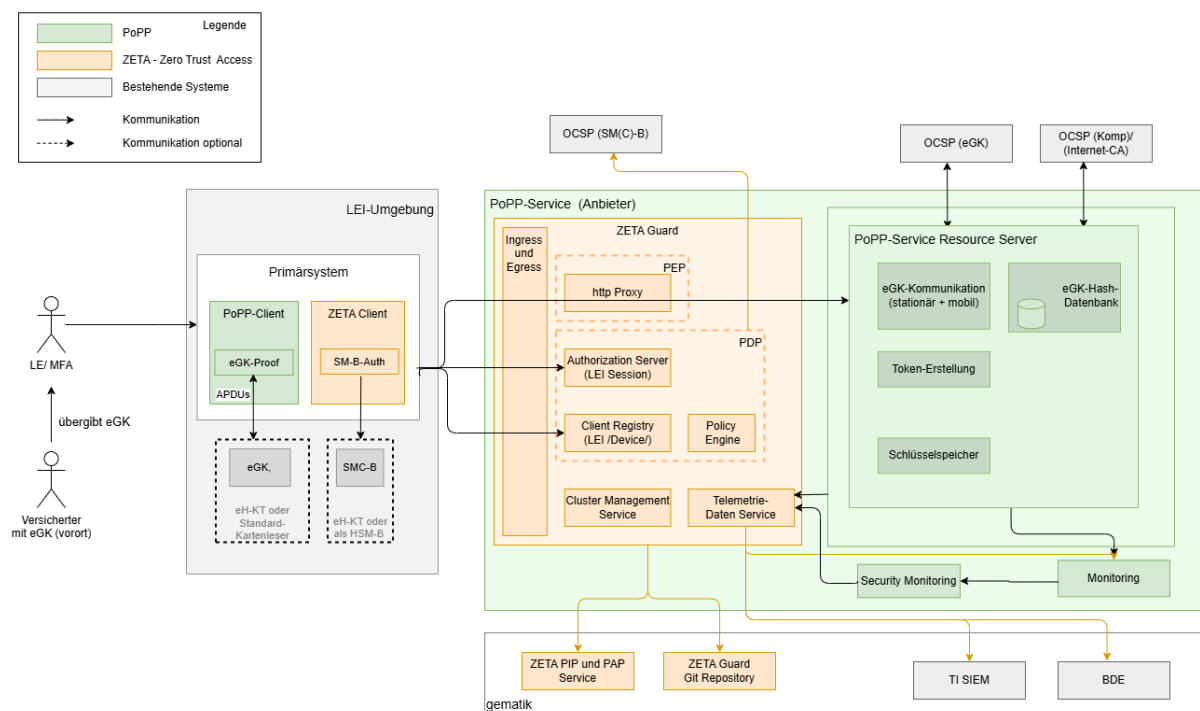


Abbildung 4: Systemkontext PoPP-Lösung (Stufe 1)

Die obige Abbildung zeigt die vom PoPP-Service-Anbieter verantworteten Komponenten und Interaktionen. Die zum ZETA Guard gehörenden Komponenten sind in orange dargestellt, die Komponenten, welche die PoPP-Businesslogik implementieren sind in grün dargestellt. Dargestellt sind zusätzlich heute bereits vorhandene und genutzte Komponenten und Dienste, die für die Nutzerauthentisierung bzw. die

Betriebsüberwachung (bspw. mittels Betriebsdatenerfassung - kurz BDE) in Anwendungsfällen der TI 2.0 weitergenutzt werden können (grau).

Das PS mit den beiden integrierten Modulen PoPP-Client und ZETA Client triggert über verschiedene Aufrufwege die PoPP-Token-Erstellung, nachdem bspw. eine LEI einen Check-in-Vorgang für einen Patienten initiiert hat. Die PoPP-Client Funktionalität verantwortet die fachlichen Abläufe zur PoPP-Token-Erstellung.

Die ZETA Client Funktionalität verantwortet die für Zero Trust relevante Kommunikation mit dem PoPP-Service. Dabei greift sie auf eine freigeschaltete SM(C)-B zu und benutzt diese zur Authentifizierung für die Registrierung und Anmeldung im Policy Decision Point (PDP) des ZETA Guard. Dabei wird ein PoPP-Client Access Token erzeugt, das für die weiteren Business-Logik Aufrufe (PoPP-Client Funktionalität) verwendet wird. Diese erfolgen dann über den Policy Enforcement Point (PEP) des ZETA Guard.

Tabelle 2: Kurzbeschreibung der Komponenten in der PoPP-Lösung für die Stufe 1

Komponente	Kurzbeschreibung	Anforderungen
PoPP-Service	Der PoPP-Service umfasst folgende Komponenten der PoPP-Lösung: <ul style="list-style-type: none"> • ZETA Guard • PoPP-Service Ressource Server. 	
ZETA Guard	Der Zero Trust Cluster ZETA Guard schützt die Kommunikation zwischen dem PoPP-Service Resource Server und den LEI.	[gemSpec_ZETA] [5.4- ZETA Guard im PoPP-Service]
PoPP-Service Resource Server	PoPP-Service Resource Server enthält die eigentliche Businesslogik des PoPP-Service für das Ausstellen von PoPP-Token an LEI-Systeme (PoPP-Clients im PS einer LEI).	
eGK-Kartenkommunikation	In der Komponente eGK-Kartenkommunikation sind die Funktionalitäten für die Token-Erstellung in den Fällen, wenn ein Versicherter mittels seiner eGK beteiligt ist.	[6.1.1- eGK-Handling]
eGK-Hash-DB	Die Komponente eGK-Hash-DB enthält die Datenbank und steuernde Funktionalität für den Abgleich von anonymisierten eGK-Metadaten für die Unterstützung der Nutzung von eGK.	[6.1.1.9- eGK-Hash-Datenbank]
Token-Erstellung	Die Komponente Token-Erstellung enthält die Funktionalität zum Erstellen des PoPP-Token; hier	[5.1- PoPP-Token: Nachweis des Versorgungskontexts]

	werden die Daten von der LEI und dem Versicherten im PoPP-Token zusammengefügt.	
Schlüsselspeicher	Die Komponente Schlüsselspeicher enthält die Funktionalität zur sicheren Ablage der verwendeten Schlüssel und zur Speicherung von Daten.	[5.2.1.7- Speicherung von Daten] [5.2.1.6- Schlüsselnutzung direkt im Verarbeitungskontext]
Monitoring	Die Komponente Monitoring sammelt Funktionalitäten für das Eigen-Monitoring der verwendeten Systeme zur Überwachung und Analyse des Betriebszustandes.	(bis zur Veröffentlichung in C_11939)
BDE-Lieferung	Die Komponente BDE-Lieferung fasst die Daten des PoPP-Service für die Betriebsdatenerfassung (BDE) der gematik zusammen und versendet sie.	(bis zur Veröffentlichung in C_11939)
Security Monitoring	Die Komponente Security Monitoring enthält Funktionen für das Sicherheits- und Event-Monitoring im PoPP-Service.	(bis zur Veröffentlichung in C_11939)
Primärsystem (PS)	Bestehende Primärsysteme werden für PoPP erweitert.	
ZETA Client	Der ZETA Client im PS einer LEI ist der direkte Kommunikationspartner von ZETA Guard, der Zero Trust-Komponente des PoPP-Service.	[gemSpec_ZETA]
PoPP-Client	Der PoPP-Client ist der direkte Kommunikationspartner des PoPP-Service im PS einer LEI.	[gemILF_PoPP_Client]

3.3 Nachbarsysteme

In der Abbildung "Systemkontext PoPP-Lösung" sind die Nachbarsysteme des PoPP-Service dargestellt:

- OCSP-Responder des TSP für SM(C)-B:
im Internet verfügbar für die OCSP-Prüfung von SM(C)-B durch den ZETA Guard,
- OCSP-Responder des TSP für eGK:
im Internet verfügbar für die OCSP-Prüfung von eGK durch den PoPP-Service,

- OCSP-Responder des TSP für Komponenten PKI:
im Internet verfügbar für die OCSP-Prüfung des TI 1.0 Komponenten Zertifikats der PoPP-Service Identität, verwendet bei der Signatur der Application Protocol Data Units (APDU) im eGK-Ablauf,
- OCSP-Responder Internet-CA:
im Internet verfügbar für die OCSP-Prüfung des TLS-Internet-Zertifikats des PoPP-Service (für die Client-Schnittstelle),
- die gematik stellt folgende Dienste bereit:
PIP- und PAP-Service für Zero Trust, git-Repository für die ZETA Guard Images, Überwachung und Betriebsdatenerfassung: BDE und Telematikinfrastruktur Security Information and Event Management (TI-SIEM).

In der Abbildung "Systemkontext PoPP-Lösung" sind folgende nutzende Systeme nicht dargestellt:

- Fachanwendungen und FD im Internet, sowie Dienste und Komponenten, die noch in der TI 1.0 verortet sind, wie E-Rezept-FD, ePA < 3.x:
 - VSDM 2.0,
 - ePA für alle,
 - E-Rezept.

Außerdem sind die App-Backend-Systeme nicht abgebildet.

4 Funktionale Anwendungsfälle des PoPP-Service

In diesem Kapitel werden die Anwendungsfälle beschrieben, die von der Proof of Patient Presence (PoPP)-Lösung Stufe 1 abgedeckt werden. Diese Anwendungsfälle resultieren jeweils in der Ausstellung eines PoPP-Token für die LEI.

Neben der Erstellung eines PoPP-Token wird auch die Registrierung und Anmeldung der LEI betrachtet. Ausgangspunkt sind die PoPP-Use Cases aus Tabelle "PoPP-Use Cases (Business Sicht)".

Darüber hinaus wird ein Use Case zur Befüllung der eGK-Hash-Datenbank durch die Kostenträger definiert.

4.1 Übersicht der Systemanwendungsfälle für die Ausstellung des PoPP-Token

Die Anwendungsfälle für die Ausstellung eines PoPP-Token im Gesamtsystem bestehen aus zwei Anteilen:

- Authentisierung einer LEI,
- Authentisierung einer eGK mit anschließender Erstellung eines PoPP-Token.

Die benannten Anwendungsfälle werden in den nachfolgenden Kapiteln beschrieben. Anwendungsfälle werden wie Anforderungen behandelt, das heißt, die beschriebenen Sequenzen und Abläufe sind normativ.

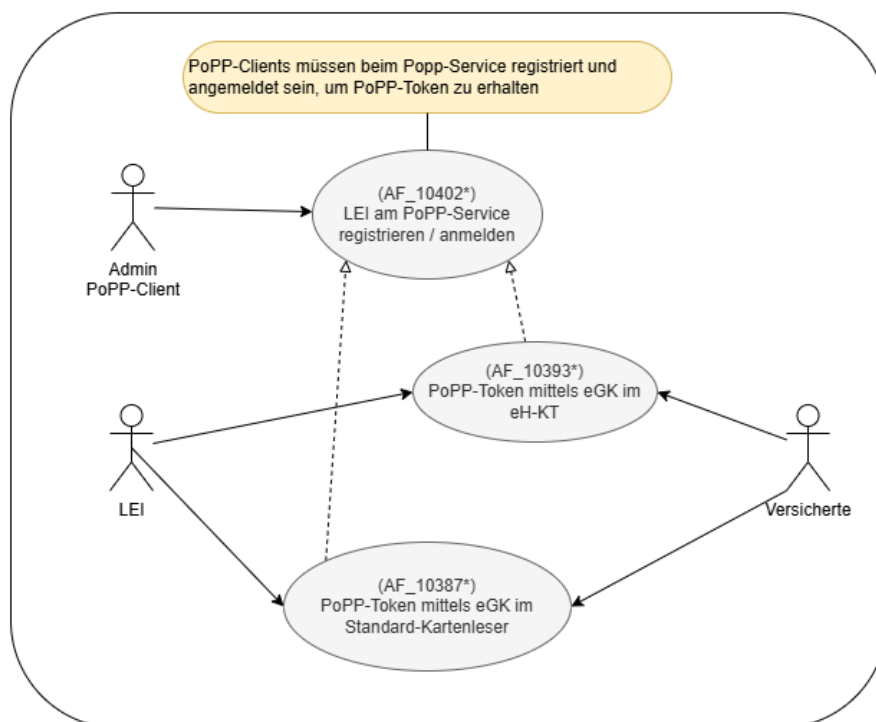


Abbildung 5: Anwendungsfälle zur Attestierung des Versorgungskontexts für Stufe 1

Zur Umsetzung der in der Tabelle "PoPP-Use Cases (Business Sicht)" dargestellten Use Cases ist der Ablauf folgender Anwendungsfall-Ketten erforderlich:

Tabelle 3: Zuordnung der Anwendungsfälle zu den Use Cases

Anwendungs-fall	Kurzbeschreibung	technische Anwendungsfälle
UC_PoPP_1a	PoPP-Token bei physischer Anwesenheit in der LEI - eGK	<ol style="list-style-type: none"> 1. zunächst AF_10402*- LEI am PoPP-Service registrieren/ anmelden und dann 2. entweder AF_10393* - PoPP-Token mittels eGK im eH-KT 3. oder AF_10387* - PoPP-Token mittels eGK im Standard-Kartenleser
UC_PoPP_2a	PoPP-Token bei physischer Anwesenheit außerhalb der LEI-eGK	Identisch zu UC_PoPP_1a

4.2 Leistungserbringerinstitution (LEI) am PoPP-Service registrieren und anmelden

Um mit dem PoPP-Service arbeiten zu können, muss sich das PS einer LEI am PoPP-Service registrieren und anmelden. Die Registrierung erfolgt einmalig mit dem ZETA Client im PS unter Verwendung des Zero Trust Access (ZETA) Guard des PoPP-Service. Die Anmeldung für registrierte PS erfolgt über eine Authentifizierung gemäß den Vorgaben von [gemSpec_ZETA].

Die Registrierung und die Anmeldung der LEI erfolgen implizit bei der Verwendung des PoPP-Service zur Erstellung eines PoPP-Token. Seitens der LEI ist eine freigeschaltete SM(C)-B erforderlich. Sind diese Voraussetzungen gegeben, kann beim Check-in von Versicherten in der LEI dann der Versorgungskontext mit einem PoPP-Token attestiert werden.

Die Anforderungen und Abläufe für die Registrierung der LEI sind in [gemSpec_ZETA#Kapitel Ablauf der SM(C)-B Authentifizierung mit DPop] beschrieben. Im Ergebnis der LEI-Registrierung und -Anmeldung liegt im PS ein Access Token mit Demonstrating Proof of Possession (DPoP) Bindung vor, mit welchem der funktionale Zugriff des PS auf den PoPP-Service erlaubt wird.

Sobald ein PS am PoPP-Service erfolgreich registriert wurde und angemeldet ist, kann es PoPP-Token erzeugen lassen bzw. abrufen.

AF_10402 -LEI am PoPP-Service registrieren / anmelden

Attribute	Bemerkung
Beschreibung	Die LEI registriert sich am PoPP-Service und meldet sich am PoPP-Service an, indem sie sich gegenüber diesem authentifiziert. Nach erfolgreicher Anmeldung kann die LEI über das PS PoPP-Token für

	einen Versicherten beim PoPP-Service abrufen.
Vorbedingung	<ul style="list-style-type: none"> • SM(C)-B ist freigeschaltet. • Einboxkonnektor mit eHealth-Kartenterminal (eH-KT) und SMC-Boder TI-Gateway mit Highspeed-Konnektor und SMC-B als Karte im eH-KT oder als SM(C)-B im Hardware-Sicherheitsmodul (HSM) des HSK, • PS implementiert einen PoPP-Client. • PS implementiert einen ZETA Client.
Ablauf	Technische Beschreibung siehe in [gemSpec_ZETA#Kapitel Ablauf der SM(C)-B Authentifizierung mit DPoP A_26091*]
Nachbedingung	<ul style="list-style-type: none"> • Die LEI ist am PoPP-Service registriert und angemeldet. • Im PS liegt ein Access Token mit DPoP Bindung vor. Es wird verwendet, um PoPP-Token Anforderungen der LEI im Resource Server des PoPP-Service zu bearbeiten.
Akzeptanzkriterien	<ul style="list-style-type: none"> • AK-03 - Bearbeitung von Anfragen nach Authentifizierung eines PoPP-Clients
Alternativen	keine

[<=]

ML-164206 -AK-03 - Bearbeitung von Anfragen nach Authentifizierung eines PoPP-Client

Nach erfolgreicher Registrierung und Authentifizierung mit DPoP gemäß [gemSpec_ZETA] ist der PoPP-Client dem PoPP-Service bekannt. Eingehende Anfragen des PoPP-Clients MUSS der PoPP-Service Resource Server entgegennehmen und verarbeiten.

[<=]

4.3 Check-in in einer LEI mit eGK und Ausgabe des PoPP-Token

Bevor der PoPP-Service für eine LEI ein PoPP-Token erstellt, ist es erforderlich, dass sich die LEI mit Hilfe der SM(C)-B beim PoPP-Service authentisiert. Des Weiteren ist es erforderlich die Anwesenheit einer eGK nachzuweisen. Der dazu verwendete PoPP-Client ist Bestandteil des PS.

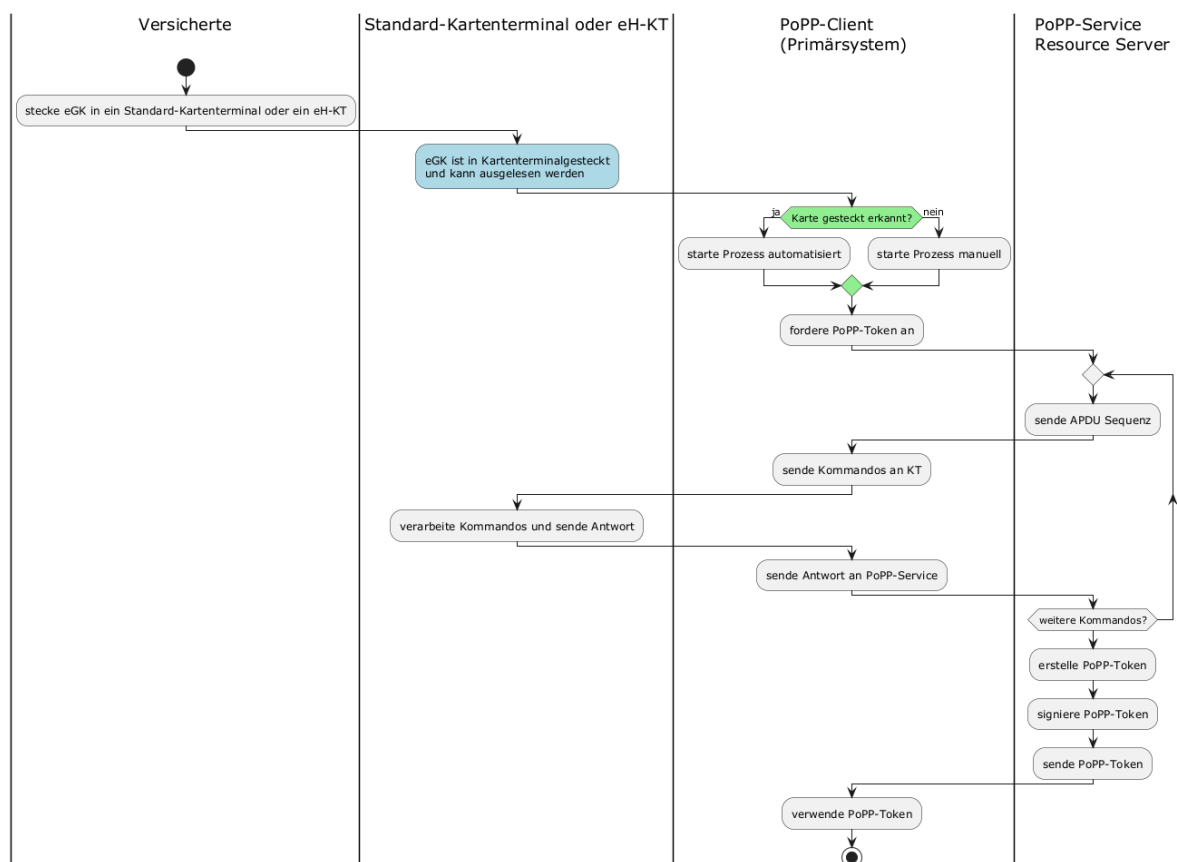


Abbildung 6: Ausstellung PoPP-Token nach Stecken der eGK in LEI

AF_10387 -PoPP-Token mittels eGK im Standard-Kartenleser

Attribute	Bemerkung
Beschreibung	<p>Erkennt der PoPP-Client als Bestandteil des PS das "Einstecken" (kontaktbehaftet oder kontaktlos) einer eGK in einen Kartenleser, so ist es möglich, dass der Prozess zum Ausstellen eines PoPP-Token automatisch startet. Alternativ wird der Prozess manuell im PS gestartet.</p> <p>Die Kommunikation des PoPP-Service mit der "gesteckten" Karte wird durch den PoPP-Client vermittelt. Nach erfolgreichem Abschluss der Kommunikation stellt der PoPP-Service ein PoPP-Token aus und sendet dieses an den PoPP-Client.</p>
Vorbedingung	<ol style="list-style-type: none"> 1. Das PS der LEI ist über einen Einboxkonnektor mit eH-KT und SMC-B oder über ein TI-Gateway mit Highspeed-Konnektor und SMC-B als Karte im eH-KT oder als SM(C)-B im HSM des HSK an die TI angeschlossen. 2. Der PoPP-Client ist am PoPP-Service registriert und angemeldet (siehe Anwendungsfall "Leistungserbringerinstitution (LEI) am PoPP-Service registrieren und anmelden"). 3. Zur Authentifizierung am PoPP-Service hat der PoPP-Client

	<p>ein gültiges PoPP-Client Access Token.</p> <p>4. Der PoPP-Client kann auf eine Anbindung der eGK über einen Standard-Kartenleser oder ein eH-KT zu reagieren ("Stecken der eGK", egal, ob kontaktbehaftete oder kontaktlose Kommunikation).</p>
Ablauf	<ol style="list-style-type: none"> 1. Der Versicherte "steckt" seine eGK in einen Standard-Kartenleser oder in ein eH-KT. 2. Der PoPP-Client reagiert auf das "Stecken" der eGK (bspw. über PC/SC oder WinCard API oder Ereignisnachrichten des Konnektors) entweder automatisch oder manuell im PS getriggert. 3. Der PoPP-Client öffnet eine Verbindung zum PoPP-Service. Die Verbindung wird über das PoPP-Client Access Token authentifiziert. 4. Der PoPP-Service sendet eine Reihe von Kommando-APDUs an den PoPP-Client. 5. Der PoPP-Client leitet die Kommando-APDUs an die eGK weiter. 6. Die eGK verarbeitet die Kommando-APDUs und sendet die korrespondierenden Antwort-APDUs an den PoPP-Client. 7. Der PoPP-Client sendet die Antwort-APDUs der eGK an den PoPP-Service. 8. Schritte 4-7 werden wiederholt, bis der PoPP-Service alle notwendigen Kommandos abgearbeitet hat: <ol style="list-style-type: none"> a. Der PoPP-Service überprüft die Echtheit der eGK mittels CV-Zertifikaten. b. Der PoPP-Service liest das X.509-Zertifikat CH.AUT der eGK aus. c. Prüfung des X.509-Zertifikats hinsichtlich Vertrauensraum der TSL, dass es sich um ein eGK-Zertifikat mit entsprechenden Werten handelt, zeitlicher Gültigkeit und Sperrstatus Online Certificate Status Protocol (OCSP) durch den PoPP-Service. 9. Der PoPP-Service erstellt und signiert das PoPP-Token. <ol style="list-style-type: none"> a. Die Informationen über den Versicherten werden aus dem X.509-Zertifikat der eGK entnommen. b. Die Informationen über die LEI werden aus der PoPP-Client Authentifizierungs-Session (PoPP-Client Access Token) entnommen. 10. Der PoPP-Service übermittelt das PoPP-Token an den PoPP-Client.
Nachbedingung	<p>Das PoPP-Token liegt im PS vor, ein Anwendungsmodul innerhalb des PS kann das PoPP-Token als Autorisierung bei einem FD verwenden, bspw. zum Abruf der</p>

	Versichertenstammdaten.
Akzeptanzkriterien	<ul style="list-style-type: none"> • AK-07 - Prüfung der eGK vor Ausstellung eines PoPP-Token • AK-08 - Verwendung der Versichertendaten der eGK
Alternativen	Alternativ zur eGK kann der Versicherte den Versorgungskontext mit seiner GesundheitsID attestieren.
Hinweis	Es sind grundsätzlich alle Kartenleser verwendbar und entsprechend weder im PoPP-Client noch im PoPP-Service technische Maßnahmen notwendig, um die Verwendung bestimmter Kartenleser durchzusetzen. Die gematik wird Sicherheitshinweise, die sich an die Nutzer (Leistungserbringerinstitutionen) richten, bzgl. Standard-Kartenlesern herausgeben.

[<=]

AF_10393 -PoPP-Token mittels eGK im eH-KT

Attribute	Bemerkung
Beschreibung	Erkennt der PoPP-Client als Bestandteil des PS das "Einstecken" (kontaktbehaftet oder kontaktlos) einer eGK in einen Kartenleser, so ist es möglich, dass der Prozess zum Ausstellen eines PoPP-Token automatisch startet. Alternativ wird der Prozess manuell im PS gestartet. Die Kommunikation des PoPP-Service mit der "gesteckten" Karte wird durch den PoPP-Client vermittelt. Nach erfolgreichem Abschluss der Kommunikation stellt der PoPP-Service einen PoPP-Token aus und sendet dieses an den PoPP-Client.
Vorbedingung	<ol style="list-style-type: none"> 1. Das PS der LEI ist über einen Einboxkonnektor mit eH-KT und SMC-B oder TI-Gateway mit einen Highspeed-Konnektor und SMC-B als Karte im eH-KT oder als SM(C)-B im HSM des HSK an die TI angeschlossen. 2. Der PoPP-Client ist am PoPP-Service registriert und angemeldet (siehe Anwendungsfall "Leistungserbringerinstitution (LEI) am PoPP-Service registrieren und anmelden"). 3. Zur Authentifizierung am PoPP-Service hat der PoPP-Client ein gültiges PoPP-Client Access Token. 4. Der PoPP-Client hat die Möglichkeit auf eine Anbindung der eGK über ein eH-KT an einem Konnektor ("Stecken der eGK" gleich kontaktbehaftete oder kontaktlose Kommunikation) zu reagieren.
Ablauf	<ol style="list-style-type: none"> 1. Der Versicherte "steckt" seine eGK in ein eH-KT. 2. Der PoPP-Client reagiert auf das "Stecken" der eGK (bspw. über die Auswertung von Konnektornachrichten) entweder automatisch oder manuell im PS getriggert. 3. Der PoPP-Client öffnet eine Verbindung zum PoPP-Service. Die Verbindung wird über das PoPP-Client Access Token

	<p>authentifiziert.</p> <ol style="list-style-type: none"> 4. Der PoPP-Service sendet eine Reihe von Kommando-APDUs an den PoPP-Client. 5. Der PoPP-Client leitet die Kommando-APDUs über den Konnektor an die eGK weiter. 6. Die eGK verarbeitet die Kommando-APDUs und sendet die korrespondierenden Antwort-APDUs über den Konnektor an den PoPP-Client. 7. Der PoPP-Client sendet die Antwort-APDUs der eGK an den PoPP-Service. 8. Schritte 4-7 werden wiederholt, bis der PoPP-Service alle notwendigen Kommandos abgearbeitet hat: <ol style="list-style-type: none"> a. Der PoPP-Service überprüft die Echtheit der eGK mittels CV-Zertifikaten. b. Der PoPP-Service liest das X.509-Zertifikat CH.AUT der eGK aus. c. Prüfung des X.509-Zertifikats hinsichtlich Vertrauensraum der TSL, dass es sich um ein eGK-Zertifikat mit entsprechenden Werten handelt, zeitlicher Gültigkeit und Sperrstatus Online Certificate Status Protocol (OCSP) durch den PoPP-Service. 9. Der PoPP-Service erstellt und signiert das PoPP-Token. <ol style="list-style-type: none"> a. Die Informationen über den Versicherten werden aus dem X.509-Zertifikat der eGK entnommen. b. Die Informationen über die LEI werden aus der PoPP-Client Authentifizierungs-Session (PoPP-Client Access Token) entnommen. 10. Der PoPP-Service übermittelt das PoPP-Token an den PoPP-Client.
Nachbedingung	Das PoPP-Token liegt im PS vor, ein Anwendungsmodul innerhalb des PS kann das PoPP-Token als Autorisierung bei einem FD verwenden, bspw. zum Abruf der Versichertenstammdaten.
Akzeptanzkriterien	<ul style="list-style-type: none"> • AK-07 - Prüfung der eGK vor Ausstellung eines PoPP-Token • AK-08 - Verwendung der Versichertendaten der eGK
Alternativen	Alternativ zur eGK kann der Versicherte den Versorgungskontext mit seiner GesundheitsID attestieren.

[<=]

ML-163355 -AK-07 - Prüfung der eGK vor Ausstellung eines PoPP-Token

Der PoPP-Service MUSS den Status der eGK prüfen. Ist die eGK nicht gültig, MUSS der PoPP-Service die Ausstellung eines PoPP-Token verweigern. **[<=]**

ML-163356 -AK-08 - Verwendung der Versichertendaten der eGK

Der PoPP-Service MUSS bei gültiger eGK und validiertem PoPP-Client ein PoPP-Token ausstellen, signieren und an den anfragenden PoPP-Client senden. Dabei MUSS der PoPP-Service sicherstellen, dass ausschließlich die Daten aus der eGK und die SMC-B Daten des anfragenden PoPP-Client bei der Ausstellung des PoPP-Token verwendet werden. [**<=**]

4.4 Import von Daten in eGK-Hash-Datenbank

Kostenträger (oder deren Dienstleister) übermitteln Informationen zu eGK-Zertifikaten für den Import in die eGK-Hash-Datenbank. Die Details dazu befinden sich im Kapitel [6.1.1.9.4- Use Cases zur Befüllung durch Kostenträger].

5 Übergreifende Festlegungen

5.1 PoPP-Token: Nachweis des Versorgungskontexts

Die Funktionalität "PoPP-Token: Nachweis des Versorgungskontexts" stellt auf Anforderung von PoPP-Clients ein PoPP-Token bereit. Das PoPP-Token ist ein kryptografisch gesicherter Nachweis eines Versorgungskontexts über zwei Identitäten des Gesundheitswesens (Versicherte, bzw. dessen eGK und LEI).

Damit ist die zentrale Businesslogik des PoPP-Service beschrieben. Die weiteren Funktionsmerkmale, die für die Erstellung eines PoPP-Token in den unterschiedlichen Konstellationen benötigt werden, sind logisch in den Kapiteln für den PoPP-Resource Server gefasst.

5.1.1 PoPP-Token-Erstellung

In diesem Kapitel sind die Anforderungen an den PoPP-Service zusammengefasst, die für das Erstellen eines PoPP-Token notwendig sind. Für die Bedingungen, unter denen der PoPP-Service einen PoPP-Token ausstellt, siehe [\[4.1- Übersicht der Systemanwendungsfälle für die Ausstellung des PoPP-Token\]](#).

A_26432 -PoPP-Service - PoPP-Token JWT

Der PoPP-Service MUSS PoPP-Token immer gemäß [RFC7519] als JWT ausstellen und im Compact Serialization Format kodieren.[<=]

A_26431 -PoPP-Service - PoPP-Token Claims

Der PoPP-Service MUSS im PoPP-Token die Claims gemäß [I_PoPP-Token_Generation.yaml], Schema "Token Claims" verwenden.[<=]

Tabelle 4: PoPP-Token Claims (informativ) berücksichtigt sind Stufe 1 und Stufe 2

Name	Wert
version	Version des PoPP-Token-Formats (Fester Wert "1.0.0")
iss	Aussteller des Token. Der Wert muss die URL des PoPP-Service ohne Pfad und ohne trailing Slash sein.
iat	Zeitpunkt der PoPP-Token-Erstellung. Alle time Werte in Sekunden seit 1970, [RFC 7519].
proofMethod	Methode, die verwendet wurde, um die Identität des Versicherten nachzuweisen. Alle zulässigen Werte sind in [I_PoPP-Token_Generation.yaml] spezifiziert. Auszug: "healthid" - Authentifizierung des Versicherten per GesundheitsID "ehc-practitioner-..." - Ermittlung der Versichertenidentität via eGK, die über eine LEI angebunden ist (kein Nachweis, dass die eGK dabei lokal vor Ort in der LEI verwendet wurde)

	"ehc-provider-..." - Ermittlung der Identität via eGK, die über ein PoPP-Modul angebunden ist
patientProofTime	<p>Der Zeitpunkt, zu dem der Nachweis des Patienten durchgeführt wurde.</p> <ul style="list-style-type: none"> Bei Nachweismethoden mit eGK ist dies der Zeitpunkt, zu dem die eGK verwendet wurde. Bei Nachweismethoden mit GesundheitsID ist dies der Zeitpunkt, zu dem die Identität mithilfe von OpenID Connect verifiziert wurde. <p>Alle time Werte in Sekunden seit 1970, [RFC 7519].</p>
patientId	KVNR - Versichertennummer des Patienten
insurerId	Institutionskennzeichen der Krankenversicherung (IK-Nummer)
actorId	Telematik-ID der am PoPP-Service authentifizierten LEI (bspw. über SMC-B), die den Behandlungskontext über die oben angegebene Methode nachgewiesen hat.
actorProfessionOid	OID der agierenden LEI passend zu actorId (Telematik-ID) gemäß [gemSpec_OID].
authorization_details	<p>Das Claimauthorization_details gemäß [RFC9396] ist optional.</p> <p>Es dient dazu, Autorisierungsanforderungen detailliert und strukturiert zu definieren. Es ermöglicht die Spezifikation komplexer Berechtigungen, wie etwa spezifische Operationen oder Bedingungen. Das Claim ist so gestaltet, dass es sich flexibel erweitern lässt und daher für zukünftige Anforderungen und Entwicklungen in der Autorisierungsverwaltung geeignet ist.</p> <p>Anmerkung: Claim-Bezeichnung ist in Snake-Case und entspricht den Vorgaben aus dem [RFC9396] bzw. von IANA.</p>

A_26961 -PoPP-Service - PoPP-Token Claims über Leistungserbringer (LE)

Der PoPP-Service MUSS als "actorId" und "actorProfessionOid" die Werte der Institution setzen, die sich gegenüber dem PoPP-Service authentifiziert hat und den Nachweis über den Versorgungskontext mit dem Versicherten erbracht hat.【<=】

A_26962 -PoPP-Service - PoPP-Token Claims über Versicherte

Der PoPP-Service MUSS Claims über den Versicherten ausschließlich nach der vorherigen Authentifizierung (GesundheitsID oder eGK) in das PoPP-Token aufnehmen und dabei genau die aus der Authentifizierung erhaltenen Daten verwenden.【<=】

A_26433 -PoPP-Service - PoPP-Token Header und Signatur

Der PoPP-Service MUSS die JWT PoPP-Token mit dem Algorithmus ES256 mit dem im HSM gehaltenen PoPP-Token-Signaturschlüssel signieren und dabei Header-Attribute gemäß [I_PoPP-Token_Generation.yaml], Schema "TokenHeaders", setzen und mitsignieren (protected headers).【<=】

Tabelle 5: PoPP-Token Header (informativ)

Name	Wert
typ	Typ des Token. Fester Wert "vnd.telematik.popp+jwt"
alg	Algorithmus mit welchem das PoPP-Token signiert wurde. Fester Wert 'ES256'.
kid	Key-ID des Schlüssels, der zur Signierung des Token verwendet wurde. Daten zum zugehörigen Signaturprüfchlüssel lassen sich vom JWK-Endpunkt des PoPP-Service abrufen.

Informatives Beispiel für einen PoPP-Token:

```
{
  "typ": "vnd.telematik.popp+jwt",
  "alg": "ES256",
  "kid": "x_vW4LVDipvu8iUQ5a1KsZLWtH6jh4eJ4c5offXtMV0"
}
.
{
  "iat": 1722593256,
  "iss": "https://popp.example.com",
  "proofMethod": "ehc-practitioner-trustedchannel",
  "patientProofTime": 1722593255,
  "patientId": "X123456789",
  "insurerId": "123456789",
  "actorId": "1-2012345678",
  "actorProfessionOid": "1.2.276.0.76.4.50"
}
```

(Signatur vernachlässigt)

A_26434 -PoPP-Service - Bereitstellung der öffentlichen Schlüssel zur Verifikation der PoPP-Token als JWKS

Der Anbieter eines PoPP-Service MUSS die öffentlichen Schlüssel zur Verifikation der PoPP-Token als JWK-Set nach [RFC7517] bereitstellen und dabei zu jedem öffentlichen Schlüssel die folgenden Attribute angeben:

Name	Wert
kid	In dem JWK-Set eindeutige Kennung des Schlüssels. Es wird empfohlen Thumbprint des öffentlichen Schlüssels gemäß [RFC7638] als 'kid' zu verwenden.
use	Verwendungszweck des Schlüssels. Fester Wert 'sig'.
ktu	Schlüsseltyp. Derzeit werden nur elliptische Kurven unterstützt. Fester Wert 'EC'
crv	Bezeichnung der elliptischen Kurve. Erlaubte Werte:

	<ul style="list-style-type: none"> P-256 für die NIST Curve
x	X-Koordinate des öffentlichen Schlüssels
y	Y-Koordinate des öffentlichen Schlüssels
alg	Gibt den kryptografischen Algorithmus an, der mit dem Schlüssel verwendet werden soll. Erlaubte Werte: <ul style="list-style-type: none"> ES256 für den P-256 Schlüssel (ECDSA-Signatur mit SHA-256).
x5c	Die X.509-Zertifikatskette, die diesen Schlüssel zertifiziert. Enthält als einziges Element das EE-TI-Zertifikat des PoPP-Service (C.ZD.SIG). Das Zertifikat ist als DER und anschließend als Base64 kodiert, siehe [RFC7517#Abschnitt 4.7].

[<=]

Informatives Beispiel für einen PoPP JWK-Set

```
{
  keys: [
    {
      "kid": "x_vw4LVDipvu8iUQ5aIksZLWtH6jh4eJ4c5offXtMV0",
      "use": "sig",
      "kty": "EC",
      "crv": "P-256",
      "x": "C3Q12wBw1K49LCeJBjDNhT_0TmWe6zZ_8pUNLF7IEfE",
      "y": "5CNecFczeOzRPhsuXeDXxJyFjG0vfIgcXqKkst6csto",
      "alg": "ES256",
      "x5c": [
        "MIICBzCCAA6gAwI..."
      ]
    }
  ]
}
```

Hinweis: Siehe auch Tabelle "Entity Statement des PoPP-Service als Protected Resource" im Anhang.

A_26533 -PoPP-Service - Veröffentlichung der öffentlichen PoPP-Token-Verifikations-Schlüssel als signiertes JWKS

Der Anbieter eines PoPP-Service MUSS das JWK-Set zur Verifikation der PoPP-Token als JWT gemäß [RFC7519] bereitstellen. Das JWT MUSS mit dem Entity Statement-Signing Key signiert sein, die Signatur muss spätestens nach 24 Stunden erneuert werden. Die URL zum Herunterladen des JWT mit signierten JWK-Set muss im Entity Statement unter `metadata.oauth_resource.signed_jwks_uri` angegeben werden. Als Content-Type HTTP-Header muss `application/jwk-set+jwt` verwendet werden.

[<=]

Hinweis: siehe auch [jwk-set+jwt].

A_28529 -PoPP-Service - Rechtzeitige Ankündigung neuer PoPP-Token-Signaturschlüssel (Key-Rollover)

Der Anbieter des PoPP-Service MUSS rechtzeitig vor Ende der zeitlichen Gültigkeit des aktuellen PoPP-Token-Signaturschlüsselpaars ein neues Schlüsselpaar im HSM erzeugen lassen und sowohl für den öffentlichen Schlüssel das Zertifikat entsprechend A_26495* beziehen, als auch den öffentlichen Schlüssel entsprechend A_26434* und A_26533*

bereitstellen. Veröffentlichung per JWK-Set und Erhalt des TI-Zertifikats MÜSSEN mindestens eine Woche vor Ablauf des aktuellen Schlüsselpaars geschehen. [≤]

5.1.2 PoPP-Token Prüfung

Das Kapitel enthält die Anforderungen an die Systeme, die PoPP-Token verifizieren und verarbeiten. Dabei sind zwei Wege der Verifikation möglich: via Entity Statement, welches auf den Vertrauensanker des Federation Master rückführbar ist und via TI-PKI mit der TSL als Vertrauensanker. Dienste, die einen PoPP-Token verifizieren, werden im Folgenden als "PoPP-Verifier" bezeichnet.

A_27015 -PoPP-Verifier - Prüfung Signaturzertifikat via TI-PKI möglich

Der PoPP-Verifier KANN die Signatur des PoPP-Token auch gegen die TI-PKI verifizieren. [≤]

A_27016 -PoPP-Verifier - Prüfung Signaturzertifikat via TI-PKI - Vorgaben

Der PoPP-Verifier, der das Signaturzertifikat des PoPP-Token via TI-PKI prüft, MUSS dabei verifizieren, dass das für die Signatur des PoPP-Token verwendete Signaturzertifikat:

- im jwks des PoPP-Service mit der entsprechenden, im Header des PoPP-Token angegeben kid enthalten ist,
- aus der TI-PKI stammt,
- das Zertifikatsprofil oid_zd_sig (OID 1.2.276.0.76.4.287, "C.ZD.SIG") aufweist,
- die technische Rolle oid_popp-token (OID 1.2.276.0.76.4.320) aufweist und
- per OCSP von der TI-Komponenten-PKI als "good" beauskunftet wird.

[≤]

Hinweis: Nach A_27296 KANN der PoPP-Service in der TI-Föderation registriert sein. Ist das nicht der Fall, so kann ein PoPP-Verifier die URL des PoPP-Service aus dem Claim iss im PoPP-Token evaluieren (siehe "A_26452* - PoPP-Verifier - JWT Claims Validierung").*

Hinweis: Der OCSP-Responder ist im Internet erreichbar. Die Adresse des OCSP-Responders ist dem Authority Information Access (AIA) des Zertifikats zu entnehmen.

A_26449 -PoPP-Verifier - Verwendung von PoPP-Service JWK-Sets

Der Anbieter eines PoPP-Verifier MUSS in regelmäßigen Abständen die JWK-Set des PoPP-Service [RFC7517] über dem im Entity Statementmetadata.oauth_resource.signed_jwks_uri angegebenen URL abrufen und die öffentlichen Schlüssel zur Verifikation der PoPP-Token verwenden. Spätestens nach 24 Stunden MUSS das Entity Statement des PoPP-Service und das JWK-Set erneut abgerufen werden.

[≤]

A_27358 -PoPP-Verifier - Zugang zum Entity Statement des PoPP-Service

Der PoPP-Verifier KANN die URL zum Laden des Entity Statement des PoPP-Service ermitteln, indem er beim Federation Master die Liste der in der TI-Föderation registrierten Teilnehmer abrufen und daraus die Teilnehmer-URL des PoPP-Service extrahiert. Das Entity Statement ist dann unter <Teilnehmer-URL PoPP-Service>/ .well-known/openid-federation gemäß A_27296* abrufbar. [≤]

Hinweis: Die Liste der registrierten TI-Teilnehmer kann unter [\[https://app.federationmaster.de/federation/list\]](https://app.federationmaster.de/federation/list) abgerufen werden.

A_26534 -PoPP-Verifier - PoPP-Service JWK-Set Signatur Prüfung

Der PoPP-Verifier MUSS bei jedem Bezug des JWK-Sets dessen Signatur mit Hilfe des Entity Signing-Keys aus dem Entity Statement prüfen. [≤]

A_26450 -PoPP-Verifier - PoPP-Token Signaturprüfung

Der PoPP-Verifier MUSS sicherstellen, dass die JWT-Signatur des PoPP-Token gemäß [RFC7515] verifiziert wird. Folgende Header-Attribute müssen im signierten JWT enthalten sein:

- typ - fester Wert "vnd.telematik.popp+jwt",
- alg - fester Wert "ES256",
- kid - Key-ID des verwendeten Schlüssels.

Der öffentliche Schlüssel zur Verifikation der Signatur muss aus dem JWK-Set des PoPP-Service über das kid Header-Attribut des PoPP-Token ermittelt werden. [≤]

A_26452 -PoPP-Verifier - JWT Claims Validierung

Der PoPP-Verifier MUSS sicherstellen, dass die folgenden Claims im PoPP-Token vorhanden sind und deren Inhalt prüfen:

- iss - muss die URL desPoPP-Service aus dem Entity Statement sub-Attribut enthalten
- iat - Ausstellungszeitpunkt des PoPP-Token muss anwendungsspezifisch geprüft werden (bspw. nicht älter als 5 Minuten),
- actorId - Telematik-ID der LEI, für die der PoPP-Token ausgestellt wurde, muss abgeglichen werden, gegen die vom PoPP-Verifier authentifizierte LEI, die den PoPP-Token vorlegt

Alle weiteren Claims müssen entsprechend dem fachlichen Bedarfs ausgewertet werden. [≤]

Hinweis: Die vollständige Liste der Claims und ihrer Ausprägung sind in A_26431 spezifiziert.*

Ab diesem Punkt kann das PoPP-Token fachlich verarbeitet werden. Die Informationen aus dem PoPP-Token, insbesondere die Patienten- und Leistungserbringer-Identifikation, können zur Autorisierung von Zugriffen auf medizinische Daten verwendet werden.

5.2 Datenschutz und Sicherheit

Der PoPP-Service ist frei im Internet erreichbar und muss entsprechend seine Außenschnittstellen vor Angriffen und unberechtigten Zugriffen schützen. Dies geschieht für Zugriffe durch Leistungserbringerinstitutionen (LEI) bereits im Sinne der TI 2.0 über die Zero Trust Mechanismen bzw. konkret durch das Einbinden des von der gematik bereitgestellten ZETA Guard.

Der PoPP-Service verarbeitet Daten, aus denen genau nachvollziehbar ist, welche Versicherten bei welchen LEI in Behandlung sind. Bereits im Einzelnen, insbesondere aber in der Summe mehrerer solcher Daten pro Versicherten (Profilbildung) sind damit Rückschlüsse auf medizinische Sachverhalte möglich. Daher sind die vom PoPP-Service verarbeiteten Daten, als personenbezogene medizinische Daten zu bewerten. Zudem hat der PoPP-Service Zugriff auf den privaten Signaturschlüssel um PoPP-Token zu erstellen, über die wiederum Zugriffe auf medizinische Daten der Versicherten möglich werden. Daher müssen Zugriffe des Betreibers auf diese Daten mit technischen Mittel

verhindert werden, weshalb der PoPP-Service innerhalb einer Vertrauenswürdiges Ausführungsumgebung (VAU) laufen muss.

Da der ZETA Guard die von den LEI eingehenden TLS-Verbindungen terminieren muss, um die eingehenden Daten analysieren zu können, hat er rein technisch Zugriff auf alle übertragenen Daten im Klartext. Daher muss der ZETA Guard innerhalb der VAU laufen (vgl. [gemSpec_ZETA#A_25608]).

A_26469 -PoPP-Service - Ausschließlich TLS-Verbindungen

Der PoPP-Service MUSS sicherstellen, dass er nur TLS-geschützte Verbindungen zu allen externen Kommunikationspartnern herstellt. [≤]

A_27082 -PoPP-Service - DDoS-Protection

Der Anbieter des PoPP-Service MUSS Angriffe auf die Verfügbarkeit des PoPP-Service (DDoS) an seinen Schnittstellen zum Internet abwehren und dafür einen qualifizierten Dienstleister zum Schutz vor DDoS-Angriffen beauftragen, der im BSI-Dokument "Qualifizierte DDoS-Mitigation Dienstleister" ([BSI-QDDoS]) gelistet ist. [≤][≤]

A_27219 -PoPP-Service - Absicherung Internet-Schnittstellen mit Paketfiltern

Der Anbieter des PoPP-Service MUSS die Schnittstellen des PoPP-Service zum Internet durch Paketfilter absichern, welche ausschließlich die erforderlichen Protokolle weiterleiten und nicht auf denselben Systemen laufen wie der PoPP-Service selbst und dabei die Empfehlungen zu Paketfiltern [Kapitel 6.1.3, 6.2.3 und 7] in [BSI ISI-LANA] befolgen. [≤]

A_26470 -PoPP-Service - Schutz vor Angriffen auf Anwendungsebene

Der PoPP-Service MUSS sicherstellen, dass Angriffe auf Anwendungsebene erkannt und abgewehrt werden, indem er:

- für alle seine Schnittstellen mindestens die Daten und Parameter, die er empfängt, sicherheitstechnisch validiert, bevor sie fachlich verarbeitet werden und
- für Versichertenzugriffe zudem mindestens Maßnahmen zum Schutz vor den Risiken in der aktuellen Version der [OWASP-Top-10-Risiken] umsetzt.

Erkannte Angriffe MÜSSEN für das Zero Trust Security Monitoring berücksichtigt werden. [≤]

Hinweis: Der ZETA Guard führt bereits eine gewisse Angriffserkennung durch, jedoch kann er die eingehenden Daten nicht entsprechend den erwarteten Schemata/Mustern bewerten, da diese nur dem PoPP-Service bekannt sind (siehe Kommentar zu [gemSpec_ZETA#A_25406]). Zudem werden Anfragen von Versicherten (Zugriffe via PoPP-Modul) nicht über den ZETA Guard geleitet, sondern direkt vom Authorization Server des PoPP-Service entgegengenommen - solange ZETA noch keine Versichertenzugriffe unterstützt. Die Formulierung "alle seine Schnittstellen" inkludiert u.a. auch die technische Schnittstelle zu den TSPs der Kassen zur Befüllung der eGK-Hash-Datenbank. Es sind A_25421* und A_26612* zu berücksichtigen.*

A_26472 -PoPP-Service - Eingeschränkte Speicherung von Daten

Der PoPP-Service MUSS sicherstellen, dass keine Anwendungsdaten - weder temporär noch permanent - gespeichert werden, außer es wird durch Anforderungen explizit gefordert bzw. erlaubt. [≤]

Hinweis: Bzgl. verschlüsselter Speicherung inkl. Integritätsschutz siehe A_26603, A_26604*, A_26605*. Bzgl. Einträgen in die eGK-Hash-Datenbank siehe [6.1.1.9- eGK-Hash-Datenbank]. Von A_26472* ausgenommen ist zum einen für die Funktionalität notwendiges Schlüsselmateriale und zum anderen Monitoring-Daten, die durch andere Anforderungen gefordert werden.*

A_27613 -PoPP-Service - Maßnahmen gegen Datenverlust

Der PoPP-Service und der Anbieter des PoPP-Service MÜSSEN Maßnahmen zum Datenverlust umsetzen und dabei mindestens täglich für die Konfigurationsdaten und insbesondere die eGK-Hash-Datenbank hinsichtlich Vertraulichkeit und Integrität geschützte Sicherungskopien anlegen. Der PoPP-Service MUSS dies technisch unterstützen und der Anbieter MUSS die Sicherung durchführen und die Sicherungen geeignet verwahren. [≤]

A_26592 -PoPP-Service - Rollentrennung zwischen Hersteller und Anbieter

Der Hersteller bzw. Anbieter eines PoPP-Service MUSS sicherstellen, dass Personen, die an der Herstellung/Implementierung des PoPP-Service beteiligt sind (Rolle Hersteller), nicht zeitgleich am Betrieb des PoPP-Service beteiligt sind (Rolle Anbieter) und dass entsprechende Prozesse definiert und etabliert sind, die dies dauerhaft gewährleisten und eine regelmäßige Validierung der Umsetzung durchsetzen. Die Umsetzung des Rollenausschlusses MUSS die Weisungsbefugnis von Vorgesetzten berücksichtigen. Das bedeutet, dass kein Vorgesetzter direkte Weisungsbefugnis sowohl für Personen mit der Rolle "Hersteller" als auch für Personen mit der Rolle "Anbieter" haben darf. Eine Ausnahme bildet die Geschäftsführung des Unternehmens, wenn beide Rollen vom selben Unternehmen gestellt werden. [≤]

5.2.1 Vertrauenswürdige Ausführungsumgebung (VAU)

5.2.1.1 Allgemein

Der PoPP-Service wird innerhalb einer Vertrauenswürdigen Ausführungsumgebung (VAU) betrieben. Dies betrifft alle Komponenten des PoPP-Service: ZETA Guard und PoPP-Service Resource Server inkl. eGK-Hash-Datenbank.

Die VAU besteht aus der Summe von Maßnahmen in Software und Hardware, die den Schutz vor unberechtigten Zugriffen des Anbieters/Betreibers eines TI-Dienstes auf schützenswerte Daten und Schlüssel gewährleisten. Die Software besteht aus dem VAU-Image, das vom Hersteller bereitgestellt wird und aus dem die Verarbeitungskontexte der VAU im Betrieb gestartet werden. Innerhalb eines gestarteten Verarbeitungskontextes werden dann die schützenswerten Daten verarbeitet bzw. kann aus diesem Verarbeitungskontext auf schützenswerte Schlüssel zugegriffen werden. Diese schützenswerten Schlüssel werden in einem Hardware Security Module HSM gespeichert und ebenso werden im HSM Prüfungen zur Validierung von VAU-Image durchgeführt.

Die Sicherheitsleistung beruht u. a. auch auf der Nutzung Hardware (HW)-naher Funktionen, weshalb auch die zugrundeliegende Hardware zur VAU gehört und somit Teil des Produkts ist.

A_26471 -PoPP-Service - VAU - Umsetzung einer VAU

Der PoPP-Service MUSS eine VAU umsetzen und durchsetzen, dass:

1. nur innerhalb eines Verarbeitungskontextes der VAU eingehende Daten von Versicherten und LEIs im Klartext verarbeitet werden,
2. nur durch einen Verarbeitungskontext der VAU die folgenden Schlüssel im HSM nutzbar sind:
 - a. privater Schlüssel für die PoPP-Token-Signatur,
 - b. Schlüssel zur Authentisierung ggü. anderen Verarbeitungskontexten.
3. nur durch einen Verarbeitungskontext der VAU die folgenden Schlüssel im HSM erzeugbar und aus dem HSM exportierbar sind:

- a. privater Schlüssel PoPP-Service-TLS-Server-Identitäten (Richtung PS und PoPP-Modul integrierender App sowie eGK-CVC|X.509-Hashwert-Import),
 - b. privater Schlüssel PoPP-Service-TLS-Client-Identität (Richtung sektorialem IDP),
 - c. privater Schlüssel für die APDU-Paket-Signatur,
 - d. privater Schlüssel für die Entschlüsselung von ID Token,
 - e. privater Schlüssel für die Access Token-Signatur,
 - f. privater Schlüssel für die Access Token-Entschlüsselung.
4. nur Verarbeitungskontexte der VAU Persistenzschlüssel aus dem HSM ableiten können,
 5. innerhalb eines Verarbeitungskontextes erzeugte Daten entsprechend [A_26472*] ausschließlich mittels Persistenzschlüssel verschlüsselt außerhalb der VAU abgelegt werden.

[<=]

A_27042 -PoPP-Service - VAU - Gehärtete Schnittstellen für Anbieter

Die VAU des PoPP-Service MUSS die für den Anbieter zugänglichen Schnittstellen härten, was mindestens umfasst:

- Robustheit gegenüber versehentlichen und bewussten Fehleingaben,
- Robustheit gegenüber dem Import maliziöser Daten,
- Einschränkung der Schnittstellen auf die wesentlichen Konfigurationsfunktion,
- Einschränkung der Rechte, mit denen der Nutzer der Schnittstellen am System agiert,
- Verhindern von "low-level"-Zugängen wie bspw. einer Kommandozeile.

[<=]

Hinweis: Die Spezifikation fordert gewisse Konfigurations- und Import-Möglichkeiten an der VAU für den Anbieter.

Der Schutz vor Angriffen an den Außenschnittstellen erfolgt zum Teil über den ZETA Guard und ist zudem in A_26470 geregelt.*

5.2.1.2 Einbinden des ZETA Guard der gematik

Der PoPP-Service verwendet als TI 2.0-Service die Mechanismen des Zero Trusts für die Zugriffskontrolle. Dazu wird von der gematik zentral ein Software-Image für den ZETA Guard bereitgestellt, der von den Diensten der TI 2.0 eingebunden wird.

Für die Funktionsfähigkeit des ZETA Guard muss dieser den TLS-Kanal terminieren um die notwendigen Daten für die Zugriffsentscheidung zu erhalten und auswerten zu können. Bei PoPP fungiert TLS als VAU-Kanal, muss also innerhalb eines Verarbeitungskontextes (VK) der VAU terminieren. Entsprechend muss der ZETA Guard ebenso als Verarbeitungskontext (entweder als eigener VK oder als Teil eines Gesamt-VK mit der PoPP-Logik zusammen) in der VAU betrieben werden.

Da der von der gematik bereitgestellte ZETA Guard ein reines Docker-Image ist, muss es vom Hersteller des PoPP-Service in die Lage versetzt werden, als VAU-Image in der VAU des PoPP-Service importiert werden zu können und dort lauffähig zu sein.

Der ZETA Guard ist also so im Build-Prozess zu berücksichtigen, dass dieser ohne manuelle Anpassungen am Code automatisiert integriert werden kann. Da häufige Updates des ZETA Guard zu erwarten sind (insbesondere schnelle Patches bei neuen, relevanten CVE), ist ein manueller Anpassungsprozess zur Herstellung der Kompatibilität des ZETA Guard zur VAU des PoPP-Service inakzeptabel.

Im Folgenden wird das System, dass den Prozess zur Erzeugung von VAU-Image umsetzt und dabei automatisiert den ZETA Guard einbindet VAU-Image-Build-Pipeline genannt.

A_26468 -PoPP-Service - Bereitstellung VAU-Image-Build-Pipeline und automatisiertes Einbinden des ZETA Guard

Der Hersteller des PoPP-Service MUSS eine VAU-Image-Build-Pipeline bereitstellen und nutzen, von der:

1. das seitens gematik bereitgestellte ZETA Guard-Image entgegengenommen wird, wobei:
 - a. die gematik-Signatur des Images gegen den als vertrauenswürdig hinterlegten gematik-Signatur-Prüf Schlüssel verifiziert wird und
 - b. das Image genau nur bei erfolgreicher Signaturprüfung übernommen wird,und anschließend automatisiert:
 1. entweder aus der PoPP-Service-Logik und dem ZETA Guard-Image ein gemeinsames VAU-Image erzeugt wird
 2. oder aus jeweils PoPP-Service-Logik und ZETA Guard-Image ein eigenes VAU-Image erzeugt wird, wobei in jedes VAU-Image ein Vertrauensanker für die Authentifizierung anderer Verarbeitungskontexte hinterlegt wird,und anschließend automatisiert:
 1. zu jedem VAU-Image der signierte Attestierungswert mit der gleichen Methodik/Technik ermittelt wird, wie sie auch in der VAU im Betrieb verwendet wird und
 2. das/die VAU-Image(s) und die dazugehörigen signierten Attestierungswerte ausgegeben werden.

【<=】

Hinweis: Der in der zweiten Variante ("oder") genannte Vertrauensanker wird bei der gegenseitigen Authentifizierung bei der Kommunikation Verarbeitungskontext-zu-Verarbeitungskontext verwendet. Die Identitäten des jeweiligen Verarbeitungskontextes und die ausstellende CA sind auf dem HSM der VAU gespeichert A_26610- und werden bei der initialen Zeremonie A_26623-* erzeugt. Dabei wird der öffentliche Schlüssel der CA exportiert, der dann als Vertrauensanker in die VAU-Image eingebracht wird. Die Authentifizierung eines anderen Verarbeitungskontext ist in ersterer Variante ("entweder") nicht notwendig, da die Kommunikation ZETA Guard<=>PoPP-Service-Logik dort innerhalb des Verarbeitungskontext stattfindet.*

Ggf. ist zudem der Import eines Vertrauensankers für die Kommunikation zum HSM (Authentifizierung des HSM durch den Verarbeitungskontext) notwendig. Es kann hier dieselbe CA verwendet werden (so ist es im Hinweis unter A_26623- beschrieben). Grundsätzlich sind aber auch andere Methoden zur Etablierung eines beidseitig authentisierten Kanals zwischen Verarbeitungskontext und HSM möglich, solange der Verarbeitungskontext das HSM eindeutig authentifizieren kann.*

Die VAU-Image-Build-Pipeline muss im Rahmen der Produkt-Begutachtung des PoPP-Service geprüft werden. Der ZETA Guard selbst hat einen von der gematik abgenommenen Sicherheitsnachweis (Produktgutachten). Daher muss dieser bei dem beschriebenen Vorgehen nicht noch einmal sicherheitstechnisch betrachtet werden.

A_26822 -PoPP-Service - Sichere VAU-Image-Erzeugung (Prozess)

Der Hersteller des PoPP-Service MUSS einen sicheren Gesamtprozess zur VAU-Image-Erzeugung umsetzen und dabei:

1. die geprüfte VAU-Image-Build-Pipeline nutzen,

2. im 4-Augen-Prinzip abgesichert den gematik-Signaturprüfsschlüssel für den ZETA Guard in die VAU-Image-Build-Pipeline einbringen und
3. Abwehrmaßnahmen umsetzen gegen Manipulationen der VAU-Image-Build-Pipeline durch einen Innentäter, was auch das Verhindern des unberechtigten Einbringens von Signatur-Prüfsschlüsseln umfasst.

[<=]

5.2.1.3 Informative Erläuterung zu den Zielen der VAU und den konkreten Umsetzungshinweisen

Das Ziel der VAU ist der Ausschluss unberechtigter Zugriffe des Anbieters/Betreibers des Dienstes auf schützenswerte Daten und Schlüssel. Darüber hinaus sind die Sicherheitsanforderungen an die VAU und die Anforderungen an die Qualität der sicherheitstechnischen Begutachtung der genutzten Hardware und Software geeignet, auch gegen unberechtigte Zugriff anderer Angreifer zu schützen.

Bei PoPP sind die zu schützenden Daten zum einen die Information, welche Versicherten zu welchem Zeitpunkt welche LEI aufsuchen, woraus ein detailliertes Profil mit Rückschlüssen auf medizinische Daten des Versicherten erstellt werden kann. Zum anderen ist dies der PoPP-Token-Signaturschlüssel, mit dem PoPP-Token erzeugt werden können, die wiederum für den Zugriff auf medizinische Daten von Versicherten autorisieren.

Die zu berücksichtigen Angriffsszenarien schließen auch Zugriffe durch einzelne Innentäter beim Betriebspersonal ein. Personen aus diesem Kreis haben aufgrund Ihrer Position erhöhte Rechte und bessere Möglichkeiten grundsätzlich auf Daten zuzugreifen. Unabhängig von diesbezüglich getroffenen organisatorischen Maßnahmen sind zusätzliche technische Maßnahmen notwendig, um auch Angriffe von solchen Innentätern abzuwehren.

Ziel der VAU ist aber auch, trotz des umzusetzenden Betreiberausschlusses bei den Maßnahmen eine gute Balance zwischen Sicherheit und Betreibbarkeit zu finden. Insbesondere soll vermieden werden, dass das Einspielen von Updates jedes Mal mit aufwändigen Prozessen und der Beteiligung verschiedener Rollen verbunden ist. Im Gegensatz dazu ist ein aufwändiger Prozess unter Beteiligung mehrerer Rollen bei einer einmaligen oder zumindest sehr seltenen Zeremonie zur Inbetriebnahme vertretbar.

Entsprechend wird im Folgenden (vorrangig über Hinweise zu den Anforderungen) eine Umsetzung beschrieben, die sowohl die hier gestellten Anforderungen an die VAU erfüllt, als aber auch einen geringen Aufwand beim Einspielen von Updates im Betrieb erzeugt. Durch eine technisch durchgesetzte und geschützte Protokollierung sind dennoch jederzeit unberechtigte Veränderungen durch Dritte (gematik) eindeutig nachvollziehbar. In solchen Fällen werden dann entsprechende Maßnahmen zur Klärung und Behebung unternommen. Somit kann der Anbieter bzw. ein etwaige Innentäter auch ohne eine direktes 4-Augen-Prinzip nicht unbemerkt handeln.

Als Sicherheitsanker muss immer ein HSM zum Einsatz kommen, wobei durch die vorgeschlagene Umsetzung Firmwareanpassungen in Form eines HSM-Moduls notwendig werden. Auch diese Aufwände wirken sich nicht negativ auf den Betrieb aus, sondern fallen nur vor der Inbetriebnahme an.

Der kurz zusammengefasste Umsetzungsansatz

Der Anbieter kann eigenständig das VAU-Image (PoPP-Service und ZETA Guard) im HSM als Hashwert bekannt machen und das Image in die VAU einspielen. Das HSM prüft bei der Bekanntmachung, dass der Hashwert vom Hersteller signiert ist. Nach Übernahme eines Hashwerts protokolliert das HSM diesen und signiert das Protokoll, mit Schlüsseln, auf die nur das HSM zugreifen kann. Das Protokoll kann exportiert werden. Der Prüfsschlüssel für die Protokollsignatur liegt der gematik vor. Da die gematik jedes VAU-

Update vom Hersteller gemeldet bekommt und in diesem Zuge auch den Hashwert des fertigen VAU-Image erhält, kann jederzeit über das Protokoll aus dem HSM nachvollzogen werden, dass vom Anbieter nur die VAU-Image bekannt gemacht wurden, die auch vom Hersteller gemeldet worden sind. Ein aus einem VAU-Image gestarteter Verarbeitungskontext muss über Mittel der VAU attestiert werden und nur mit gültigen Attestierungsinformationen kann der Verarbeitungskontext auf Schlüssel im HSM zugreifen bzw. diese dort erzeugen lassen und exportieren. Das HSM erkennt zulässige Verarbeitungskontexte am Hashwert des VAU-Image, aus dem der Verarbeitungskontext gestartet wurde, da dieser Teil der Attestierungsinformationen ist und gegen die im HSM hinterlegten Hashwerte abgeglichen werden kann. Die Attestierungsinformationen sind von sicher in der VAU gespeicherten Schlüsseln signiert, sodass das HSM die Validität der präsentierten Informationen prüfen kann (wenn es zuvor die öffentlichen Prüfschlüssel für die sicher in der VAU gespeicherten Signatur-Schlüssel erhalten hat).

Ebenso wie VAU-Image bzw. deren Hashwert können dann auch erzeugte Schlüssel (bzw. deren öffentlicher Teil) protokolliert werden. Somit ist auch eine Verifikation möglich, dass für die verschiedenen Identitäten des PoPP-Service (bspw. für TLS) auch tatsächlich nur Schlüssel verwendet werden, die ausschließlich durch einen Verarbeitungskontext der VAU genutzt werden können.

Für das Vorgehen ist eine initial sichere Zeremonie notwendig, bei der u. a. alle notwendigen Prüfschlüssel in das HSM importiert werden und der Protokollprüfschlüssel exportiert wird. Zudem wird der Anbieter von allen Zugriffen auf das HSM ausgeschlossen, die nicht absolut notwendig sind (bspw. der Import signierter VAU-Image-Hashwerte ist für den Anbieter möglich).

In den Hinweisen zu einem Teil der Anforderungen wird im Folgenden dieses Vorgehen detailliert.

5.2.1.4 Lifecycle eines Verarbeitungskontextes

A_26594 -PoPP-Service - VAU - Import VAU-Image nur nach erfolgreicher Signaturprüfung

Die VAU des PoPP-Service MUSS vor der Übernahme eines importierten VAU-Image die Signatur des Images verifizieren und prüfen, dass diese vom Dienst-Hersteller ausgestellt wurde und das Image nur im Positivfall übernehmen. [≤]

A_26593 -PoPP-Service - VAU - Ausschluss Manipulationen der Software bei Start eines Verarbeitungskontextes

Die VAU des PoPP-Service MUSS technisch sicherstellen, dass ausschließlich das unveränderte, aktuelle, vom Dienst-Hersteller autorisierte VAU-Image (bzw. das jeweils aktuelle PoPP- und ZETA Guard-VAU-Image, wenn getrennte VAU-Image verwendet werden) als Verarbeitungskontext gestartet wird, wobei die Attestierungsinformationen durch die VAU ermittelt und kryptografisch geschützt werden müssen. Die Prüfung der Attestierungsinformationen, ob es sich um ein autorisiertes VAU-Image handelt, muss auf einem in einem HSM gespeicherten Vertrauensanker beruhen. [≤]

Hinweis: Eine technische Umsetzung von A_26593- ist die Ermittlung des Hashwerts des zu startenden VAU-Image durch die VAU und die Signatur dieses Hashwerts mit einem sicher gespeicherten (bspw. in einem TPM) Signaturschlüssel des Herstellers der HW-Plattform (Chip-Hersteller) oder des Herstellers des PoPP-Service. Diese Attestierungsinformation kann dann an prüfende Systeme, wie bspw. das HSM übermittelt werden, wobei das verwendete Protokoll vor Replay-Attacken schützen muss (das HSM muss erkennen können, dass die Attestierungsinformationen frisch für das HSM erzeugt wurden). Dem HSM müssen entsprechend über einen sicheren Prozess zum einen die Prüfschlüssel des Chip-Herstellers bzw. PoPP-Service-Herstellers und zum anderen die zulässigen Hashwerte von VAU-Image bekannt gemacht werden. Um einen späteren Rollback zu verhindern wird immer genau nur gegen den letzten hinzugefügten also den*

aktuellen Hashwert eines jeweiligen VAU-Image geprüft bzw. ist nur kurzzeitig nach dem Einspielen die Möglichkeit eines Rollback auf das letzte davor verwendete Image möglich (siehe A_27373*).

A_27373 -PoPP-Service - VAU - Temporäre Möglichkeit des Rollback auf vorherige Version

Die VAU des PoPP-Service MUSS es ermöglichen, dass nach dem Einspielen eines neuen VAU-Image, für eine Übergangszeit von drei Tagen ein Rollback auf das letzte davor verwendete VAU-Image durchgeführt werden kann.【<=】

Hinweis: Nach Einspielen eines neuen Hashwerts ins HSM akzeptiert dieses somit für drei Tage auch Verarbeitungskontexte, die in ihren Attestierungsinformationen den zweit jüngsten im HSM vorliegenden Hashwert vorweisen. Dies dient der Möglichkeit eines schnellen Rollback, falls nach Einspielen und Inbetriebnahme einer neuen Version massive Fehler im Betrieb auftreten.

A_26595 -PoPP-Service - VAU - Regelmäßiger Neustart der Verarbeitungskontexte

Die VAU des PoPP-Service MUSS durchsetzen, dass:

1. Ein Verarbeitungskontext maximal eine Stunde (1 h) verwendet wird und somit fortlaufend neue Verarbeitungskontexte frisch aus dem jeweiligen Image gestartet werden.
2. Dabei dürfen aktuell verwendete Verarbeitungsprozesse (Abarbeiten eines Operationsaufrufs läuft noch) nicht abgebrochen werden, sondern müssen entsprechend der funktionalen Vorgaben zu Ende geführt werden, bevor der Verarbeitungskontext geschlossen wird und
3. für parallel eintreffende neue Operationsaufrufe ein anderer, zeitlich noch gültiger Verarbeitungskontext verwendet wird.

Der VAU ist es auch erlaubt für jeden Operationsaufruf einen neuen eigenen Verarbeitungskontext zu starten und nach Abarbeiten des Aufrufs sofort wieder zu schließen.【<=】

Hinweis: Es wird davon ausgegangen, dass die VAU es leistet, dass mehrere Verarbeitungskontexte parallel ausführbar sind.

A_26596 -PoPP-Service - VAU - Attestierung durch Systeme außerhalb der VAU

Die VAU des PoPP-Service MUSS es technisch ermöglichen, dass es für Dritte außerhalb der VAU prüfbar ist, dass ein Verarbeitungskontext aus einem integren, autorisierten VAU-Image gestartet wurde.【<=】

Hinweis: Bei den in A_26596- genannten Dritten handelt es sich beispielsweise um die gematik, Systeme beim Anbieter PoPP-Service (bspw. andere Verarbeitungskontexte) oder auf den PoPP-Service zugreifende Clients.*

A_26597 -PoPP-Service - VAU - Erkennen von Manipulationen an der HW der VAU - Softwareanteil

Die VAU des PoPP-Service MUSS technisch sicherstellen, dass Maßnahmen zur Manipulationserkennung seitens der Hardware der VAU oder der die VAU umgebenden HW, von der VAU so unterstützt werden, dass Zugriffe auf die HW von der VAU erkannt werden und in diesem Fall sämtliche Verarbeitungskontexte beendet werden, so dass ein unberechtigter Zugriff auf Daten (Extraktion oder Manipulation) durch den physischen Zugriff auf die Hardwarekomponenten der VAU ausgeschlossen ist.【<=】

Hinweis: Es ist zulässig, dass physische Schutzmaßnahmen in der Hardware der VAU umgesetzt werden (bspw. in etwa im Ansatz vergleichbar mit denen eines HSM) und/oder in zusätzlicher vom Anbieter bereitgestellter Hardware (bspw. durch einen mit zusätzlichem Zutrittsschutz ausgestatteten Serverschrank). In jedem Fall besteht eine Kopplung dieser physischen Schutzmaßnahmen zur VAU-Software, damit bei durch diese

Schutzmaßnahmen erkannten Zugriffen eine Reaktion der VAU stattfindet, so dass Verarbeitungskontexte geschlossen und somit alle flüchtigen Daten aus dem System entfernt werden.

A_26598 -PoPP-Service - VAU - Erkennen von Manipulationen an der Hardware der VAU - Hardwareanteil

Die VAU des PoPP-Service oder der Anbieter des PoPP-Service MUSS technische Maßnahmen zum physischen Schutz umsetzen, die Zugriffe auf die Hardwarekomponenten der VAU erkennen und diese unmittelbar an die VAU weiterleiten (vgl. [A_26597*]). Auch bei einer Umsetzung durch den Anbieter MUSS die konkrete technische Umsetzung vollständig im Produktgutachten beschrieben werden.[<=]

A_26599 -PoPP-Service - VAU - Isolation der VAU von Datenverarbeitungsprozessen des Anbieters

Die VAU des PoPP-Service MUSS die in ihren Verarbeitungskontexten ablaufenden Datenverarbeitungsprozesse von allen sonstigen Datenverarbeitungsprozessen des Anbieters (unabhängig, ob für TI-Dienste oder andere Dienste) trennen und damit gewährleisten, dass diese Datenverarbeitungsprozesse sowie der Anbieter der VAU selbst technisch vom Zugriff auf die in den Verarbeitungskontexten verarbeiteten Daten ausgeschlossen sind.[<=]

A_26600 -PoPP-Service - VAU - Isolation zwischen Datenverarbeitungsprozessen mehrerer Verarbeitungskontexte der VAU

Die VAU des PoPP-Service MUSS mit technischen Mitteln ausschließen, dass sich die Verarbeitungen innerhalb eines Verarbeitungskontextes schadhaft auf die Verarbeitungen eines anderen Verarbeitungskontextes auswirken können.[<=]

A_26601 -PoPP-Service - VAU - Löschen aller Daten beim Beenden des Verarbeitungskontextes

Die VAU des PoPP-Service MUSS sicherstellen, dass beim Beenden eines Verarbeitungskontextes sämtliche Daten dieses Verarbeitungskontextes aus flüchtigen Speichern sicher gelöscht werden oder ein Zugriff auf diese Daten technisch ausgeschlossen ist.[<=]

5.2.1.5 Anforderungen an das HSM

A_26602 -PoPP-Service - VAU - Prüfungsfunktionalität und Schlüsselmanagement im HSM

Der Anbieter eines PoPP-Service MUSS über ein HSM verfügen, welches neben der sicheren Schlüsselspeicherung und -verwaltung, folgendes umsetzt:

- eine Funktionalität zur Validierung von VAU-Image und Attestierungsinformationen zu solchen Images, wobei zwischen verschiedenen Arten von VAU-Image unterschieden werden kann (konkret PoPP-VAU-Image und ZETA Guard-Image),
- für die aus diesen Images gestarteten verschiedenen Arten von Verarbeitungskontexten (PoPP-Service und ZETA Guard) jeweils genau nur die für diesen vorgesehenen Identitäten (Schlüssel) zur Nutzung freigeben bzw. Schlüsselerzeugung und -export ermöglichen und
- sonstige Schlüsselerzeugungen (abgesehen von etwaigen spezifizierten Ausnahmen) und den Import von Prüfschlüsseln nur im technisch abgesicherten Vier-Augen-Prinzip ermöglichen.

[<=]

Hinweis: Eine valide technische Umsetzung ist ein HSM-Modul, welches:

1. *Prüf Schlüssel importieren und an bestimmte Zwecke/Anwendungsfälle binden kann,*

2. *Schlüsselpaare erzeugen und diese an bestimmte Zwecke/Anwendungsfälle bzw. an bestimmte authentifizierte Nutzer (ZETA Guard- und PoPP-Verarbeitungskontexte) des HSM binden kann,*
3. *autorisierte VAU-Image in Form von signierten Hashwerten dieser Images importieren kann und dabei die Signatur gegen zuvor importierte Prüfschlüssel prüft, bevor Hashwerte übernommen werden,*
4. *eine beidseitig authentifizierte, verschlüsselte und integritätsgeschützte Verbindung zu Verarbeitungskontexten aufbauen kann,*
5. *Attestierungsprotokolle mit Schutz vor Replay-Attacken unterstützt und somit:*
 - a. *frische Attestierungsinformationen von Verarbeitungskontexten entgegennehmen kann, deren Signatur gegen zuvor importierte Prüfschlüssel prüft,*
 - b. *die Attestierungsinformation (Hashwert) gegen die aktuell als autorisiert hinterlegten Hashwerte von VAU-Image validiert,*
 - c. *ggf. die Art von VAU-Image detektiert (falls PoPP-VAU- oder ZETA Guard-Image getrennt sind) und nur im Erfolgsfall*
 - i. *private Schlüssel zur Nutzung für genau den attestierten Verarbeitungskontext freigibt, wobei nur die für diese Art von Verarbeitungskontext zulässigen Schlüssel freigegeben (Unterscheidung PoPP-Logik und ZETA-Logik sofern verschiedene Kontexte) werden und*
 - ii. *Schlüsselpaare für den attestierten Verarbeitungskontext erzeugt und an diesen übergibt*
6. *die Anmeldung von Benutzern am HSM/HSM-Modul zur Erzeugung von Schlüsseln (abgesehen von etwaigen spezifizierten Ausnahmen) und zum Import von Prüfschlüsseln nur über sichere Authentisierungsmittel, die zwischen mehreren Rollen aufgeteilt werden können, ermöglicht.*

5.2.1.6 Schlüsselnutzung direkt im Verarbeitungskontext

Der PoPP-Service verfügt über ein oder mehrere HSMs, welche sowohl die sichere Nutzung und Speicherung von Schlüsseln gewährleisten als auch eine performante Nutzung der Schlüssel ermöglichen. Nichtsdestotrotz ist die Nutzung von Schlüsseln im HSM weniger performant, als für Schlüssel die direkt im Verarbeitungskontext der VAU verfügbar sind. Dies liegt u.a. auch am Overhead für die sichere Kommunikation mit dem HSM, die bei jeder Nutzung eines Schlüssels im HSM anfällt.

Um Performanceprobleme beim PoPP-Service zu vermeiden soll daher die Anzahl der HSM-Zugriffe, die für die jeweiligen Anwendungsfälle notwendig sind gering gehalten werden. Da die VAU und deren Verarbeitungskontexte gerade für die Verarbeitung vertraulicher Daten konzipiert und implementiert wird, ist es sicherheitstechnisch grundsätzlich denkbar Schlüssel direkt im Verarbeitungskontext zu nutzen, statt sie im HSM zu speichern. Dabei muss der Zweck des Schlüssels und auch die Laufzeit betrachtet werden, da die Speicherung außerhalb des HSM ein leicht erhöhtes Restrisiko der Offenbarung des Schlüssel hervorruft und dessen Eintrittswahrscheinlichkeit mit der Zeit, in der der Schlüssel genutzt wird, steigt.

Entsprechend dieser Betrachtungen muss der private Schlüssel für die Signatur von PoPP-Token ausschließlich im HSM gespeichert werden. Private Schlüssel für bspw. die TLS-Identitäten sowie für die Signatur und Entschlüsselung von Access Token dürfen außerhalb des HSM direkt im Verarbeitungskontext der VAU verarbeitet werden, dürfen jedoch nicht länger drei Monate genutzt werden.

Für das Vorgehen, die genannten private Schlüssel direkt im Verarbeitungskontext zu nutzen, ist zu berücksichtigen, dass diese Schlüssel teilweise nicht vom PoPP-Service selbst sondern vom ZETA Guard verwendet werden. Eine Nutzung der Schlüssel direkt im Verarbeitungskontext ist in dem Fall nur möglich, wenn der ZETA Guard dies unterstützt. Sollte letzteres nicht der Fall sein, ist selbstverständlich auch die Nutzung der Schlüssel im HSM zulässig.

A_26687 -PoPP-Service - VAU - Schlüssel- und CSR-Erzeugung im HSM durch Verarbeitungskontext

Die VAU des PoPP-Service MUSS es dem Anbieter ermöglichen Schlüsselpaare für die folgenden Identitäten erzeugen zu können:

1. PoPP-Service-TLS-Server (Richtung PS und PoPP-Modul integrierender App sowie für eGK-Zertifikat-Hashwert-Import),
2. PoPP-Service-TLS-Client (Richtung sektorialem IDP),
3. APDU-Paket-Signatur,
4. ID Token-Verschlüsselung (durch den sektoralen IDP) und
5. Access Token-Signatur (durch den PoPP-AuthZ),
6. Access Token-Ver-/Entschlüsselung (durch AuthZ bzw. Resource Server).

und dabei wie folgt vorgehen:

1. Auslösen der Schlüsselerzeugung durch den Verarbeitungskontext am HSM unter Angabe des Zwecks bzw. der zu Erzeugenden Identität,
2. Export des Schlüsselpaars aus dem HSM in den Verarbeitungskontext,
3. mit dem Persistenzschlüssel verschlüsselte Ablage des Schlüsselpaars außerhalb des Verarbeitungskontext unter Angabe der zugehörigen Identität, des Zeitpunkts der Schlüsselerzeugung sowie des Status "neu" und
4. Ausgabe eines mit dem jeweiligen privaten Schlüssel signierten CSR für die TLS-Identitäten und die APDU-Paket-Signatur-Identität an den Anbieter.

[<=]

Hinweis: Für die TLS-Client-Identität (Richtung sektorialem IDP für die Stufe 2) ist ggf. kein CSR notwendig, da nicht zwingend ein Zertifikat erforderlich ist. Die Nachvollziehbarkeit, dass die genannten Schlüssel im HSM erzeugt wurden, ist entsprechend der Hinweise zur Umsetzung für die gematik über das signierte und exportierbare Protokoll des HSM möglich. Dafür wird im Rahmen der initialen Zeremonie ein Protokoll-Signatur-Schlüsselpaar erzeugt und von der gematik der öffentliche Signaturprüf Schlüssel exportiert. In diesem Protokoll werden - neben den Hashwerten der importierten VAU-Image - die öffentlichen Schlüssel der in A_26687- genannten Schlüsselpaare unter Angabe des Zwecks bzw. der zugehörigen Identität protokolliert.*

Neu erzeugte Schlüsselpaare sind nicht automatisch aktiv, da für einige Identitäten zunächst Zertifikate durch den Anbieter bezogen werden müssen und / oder öffentliche Schlüssel in JWKSs / Entity Statements veröffentlicht werden müssen, bevor die Identität einsatzbereit ist und aktiviert werden kann.

A_27038 -PoPP-Service - VAU - Aktivierung außerhalb des HSM gespeicherter Schlüssel

Die VAU des PoPP-Service MUSS es dem Anbieter ermöglichen,

1. für über den Verarbeitungskontext im HSM erzeugte und exportierte Identitäten (neue Schlüsselpaare) Zertifikate zu importieren sowie
2. die neuen Schlüsselpaare für die Nutzung zu aktivieren (Status "neu" > "aktiv")

und der PoPP-Service MUSS dabei

1. Zertifikate ausschließlich dann übernehmen, wenn diese zum für die jeweilige Identität vorhandenen neuen Schlüsselpaar passen,
2. etwaige für die Identität bereits vorhandene Schlüsselpaare deaktivieren (Status "aktiv" > "inaktiv")
3. fortan das neue Schlüsselpaar für die jeweilige Identität verwenden.

[<=]

Hinweis: Es ist aktuell nicht vorgesehen einmal deaktivierte Schlüssel reaktivieren zu können (Status "inaktiv" > "aktiv").

A_27039 -PoPP-Service - VAU - Nutzung außerhalb des HSMs gespeicherter Schlüssel

Die VAU des PoPP-Service MUSS beim Start eines Verarbeitungskontextes prüfen,

1. ob mit dem Persistenzschlüssel geschützte Identitäten (Schlüsselpaare) verfügbar sind,
2. ob diese als aktiv gekennzeichnet sind und
3. ob im Falle der Identitäten für TLS, Access Token-Signatur, Access Token-Verschlüsselung und ID Token-Verschlüsselung diese nicht älter als drei Monate sind

und nur im Positivfall diese Identitäten verwenden.**[<=]**

5.2.1.7 Speicherung von Daten

A_26603 -PoPP-Service - VAU - Verschlüsselung von Daten vor Speicherung außerhalb des Verarbeitungskontextes

Die VAU des PoPP-Service MUSS sicherstellen, dass schützenswerte Daten, insbesondere auch Schlüssel, die außerhalb eines Verarbeitungskontextes gespeichert werden sollen, ausschließlich mit einem Persistenzschlüssel verschlüsselt aus dem Verarbeitungskontext in Speichersysteme ausgeleitet werden und die Verschlüsselung einen Integritätsschutz inkludiert.**[<=]**

A_26604 -PoPP-Service - VAU - Ableitung Persistenzschlüssel durch ein HSM

Die VAU des PoPP-Service MUSS sicherstellen, dass der Verarbeitungskontext Persistenzschlüssel von im HSM gespeicherten Master-Schlüsseln im HSM ableitet.**[<=]**

A_26605 -PoPP-Service - VAU - Nutzung Persistenzschlüssel ausschließlich im Verarbeitungskontext

Die VAU des PoPP-Service MUSS sicherstellen, dass Persistenzschlüssel ausschließlich in einem Verarbeitungskontext genutzt werden.**[<=]**

5.2.1.8 Transport von Daten und Authentisierung/Authentifizierung bei Kommunikation

A_26606 -PoPP-Service - VAU - Sicherer VAU-Kanal vom Kommunikationspartner in den Verarbeitungskontext

Die VAU des PoPP-Service MUSS sicherstellen, dass schützenswerte Daten zwischen dem PoPP-Service und einem VAU-Client oder dem PoPP-Service und dem sektoralen IDP ausschließlich über einen TLS-Kanal übermittelt werden, der in einem Verarbeitungskontext der VAU des PoPP-Service terminiert.**[<=]**

Hinweis: Für PoPP wird der VAU-Kanal somit durch TLS realisiert. Der von VAU-Clients etablierte TLS-Kanal endet im PEP des ZETA Guard, der in einem Verarbeitungskontext läuft. Der TLS-Kanal zum sektoralen IDP wird aus einem Verarbeitungskontext des PoPP-Service aufgebaut.

A_26607 -PoPP-Service - VAU - Authentisierung gegenüber VAU-Clients

Die VAU des PoPP-Service MUSS sicherstellen, dass der Verarbeitungskontext sich gegenüber VAU-Clients mittels der PoPP-Service-spezifischen TLS-Server-Identität ausweist, deren privater Schlüssel nur im Verarbeitungskontext genutzt werden kann und außerhalb für den Verarbeitungskontext verschlüsselt gespeichert ist, so dass auf den Schlüssel nur attestierte Verarbeitungskontexte Zugriff haben. [≤]

Hinweis: Vorgaben zum TLS-Zertifikat werden in A_26497- definiert.*

A_26609 -PoPP-Service - VAU - Sichere Kommunikation zwischen Komponenten

Die VAU des PoPP-Service MUSS sicherstellen, dass alle Komponenten der VAU ausschließlich transportverschlüsselt, integritätsgeschützt und beidseitig authentisiert mit anderen Komponenten (außerhalb oder innerhalb) der VAU kommunizieren, einschließlich der Kommunikation zwischen dem ZETA Guard- und den PoPP-Verarbeitungskontexten (sofern diese nicht in einem gemeinsamen Verarbeitungskontext laufen) und der Kommunikation zum HSM. Die Verbindung müssen auch gegen Zugriffe durch den Anbieter geschützt sein. [≤]

A_26610 -PoPP-Service - VAU - Identitäten zur Authentisierung für Kommunikation zwischen Verarbeitungskontexten

Die VAU des PoPP-Service MUSS sicherstellen, dass - sofern eigene VAU-Image und somit auch Verarbeitungskontexte für PoPP-Fachlogik und ZETA Guard verwendet werden - für die Kommunikation zwischen den Verarbeitungskontexten beide Kontexte eine Identität verwenden, deren privater Schlüssel im HSM gespeichert ist und auf den der jeweilige Kontext ausschließlich nach Attestierung, dass der Kontext integer aus einem autorisierten Image gestartet wurde, im HSM zugegriffen werden kann. [≤]

A_26611 -PoPP-Service - VAU - Sichere Verbindung zwischen bekannten VAU-Image und HSM

Die VAU des PoPP-Service MUSS technisch sicherstellen, dass:

1. Zwischen einem Verarbeitungskontext der VAU und dem HSM eine beidseitige authentisierte, integritätsgeschützte und vertrauliche Verbindung besteht.
2. sich ausschließlich Verarbeitungskontexte mit dem HSM verbinden können, die Instanz eines VAU-Image sind, welches dem HSM bekannt gemacht wurde und
3. die Verarbeitungskontexte auch das HSM authentifizieren können.

[≤]

Hinweis: Eine technische Umsetzung des Aspekts "Verarbeitungskontext ist Instanz eines bekannten VAU-Image" ist die Bekanntmachung des vom Dienst-Hersteller signierten Hashwerts eines VAU-Image ggü. dem HSM. Das HSM kann die Signatur gegen Prüfschlüssel verifizieren, die über sichere Prozesse ins HSM eingebracht wurden. Im Zuge des Zugriffs eines Verarbeitungskontextes auf das HSM lässt dieses den Verarbeitungskontext von der VAU attestieren und kann die wiederum signierten Attestierungsinformationen (Quote) gegen den ebenfalls vorab eingebrachten Prüfschlüssel verifizieren. Die Attestierung muss Übergabe und Prüfung einer Challenge umfassen, um Replay-Attacken auszuschließen.

5.2.1.9 Protokollierung und Monitoring

A_26612 -PoPP-Service - VAU - Datenschutzkonformes Logging und Monitoring des Verarbeitungskontextes der VAU

Die VAU des PoPP-Service MUSS die für den Betrieb des PoPP-Service erforderlichen Logging- und Monitoring-Informationen in solcher Art und Weise erheben und verarbeiten, dass mit technischen Mitteln ausgeschlossen ist, dass dadurch Identitätsdaten von Versicherten und LEI offenbart werden. [≤]

A_26613 -PoPP-Service - VAU - Protokollierung VAU-Image-Hashwerte und öffentliche Schlüssel im HSM

Das HSM des PoPP-Service MUSS:

1. alle ihm bekanntgemachten VAU-Image (Hashwerte)
2. die öffentlichen Schlüssel von erzeugten:
 - a. PoPP-Token-Signatur-Identitäten,
 - b. APDU-Paket-Signatur-Identitäten,
 - c. ID Token-Verschlüsselungs-Identitäten,
 - d. Access Token-Signatur-Identitäten,
 - e. Access Token-Verschlüsselungs-Identitäten,
 - f. PoPP-Service-TLS-Server-Identitäten (Richtung PS und PoPP-Modul integrierender App sowie eGK-Zertifikats-Hash-Import) und
 - g. PoPP-Service-TLS-Client-Identitäten (Richtung sektorialem IDP),

zusammen mit dem jeweiligen Zweck / Identitätsbezeichnung und dem Zeitpunkt des Imports bzw. der Erzeugung so integritätsgeschützt protokollieren, dass eine Manipulation des Protokolls durch potentielle Innentäter beim Anbieter ausgeschlossen ist und Dritte den Integritätsschutz außerhalb des HSM prüfen können. Die jeweiligen Informationen MÜSSEN für mindestens drei Jahre im Protokoll vorgehalten werden. [≤]

Hinweis: Eine technische Umsetzung ist die Signatur des Protokolls mit einem Protokoll-Signatur-Schlüssel, der über sichere Prozesse im Rahmen der Inbetriebnahme im HSM erzeugt wurde, ausschließlich vom HSM selbst genutzt werden kann und dessen öffentlicher Schlüssel der gematik bekannt ist.

A_26614 -PoPP-Service - VAU - Exportierbarkeit Protokoll für VAU-Image-Hashwerte und öffentliche Schlüssel aus HSM

Das HSM des PoPP-Service MUSS das integritätsgeschützte Protokoll der ihm bekanntgemachten VAU-Image und der öffentlichen Schlüssel der im HSM erzeugten Identitäten exportieren können, damit Dritte prüfen können, dass ausschließlich autorisierte VAU-Image dem HSM bekanntgemacht wurden und welche öffentlichen Schlüssel in Entity Statements und Zertifikaten aufgeführt sein müssen. [≤]

A_27546 -PoPP-Service - VAU - Tägliche Übermittlung HSM-Protokoll an gematik

Der Anbieter des PoPP-Service MUSS das HSM-Protokoll (VAU-Image-Hashwerte und öffentliche Schlüssel) täglich aus dem HSM exportieren und an die gematik übermitteln. [≤]

5.2.1.10 Konfigurierbarkeit

A_26615 -PoPP-Service - VAU - Einspielen von VAU-Image durch den Anbieter

Die VAU des PoPP-Service MUSS es dem Anbieter ermöglichen die vom Hersteller übermittelten VAU-Image eigenständig einzuspielen. [≤]

Hinweis: VAU-Image werden entsprechend der Hinweise zur Umsetzung nur nach erfolgreicher Prüfung der Signatur übernommen und werden nur ausgeführt bzw. können nur auf die relevanten Schlüssel zugreifen, wenn diese vom HSM über entsprechende Attestierungsinformationen der VAU als autorisierte Images verifiziert werden. Die dafür notwendigen Hashwerte von Images kann der Anbieter ebenfalls eigenständig ins HSM importieren und auch diese werden nur übernommen, wenn sie vom Hersteller signiert wurden.

5.2.1.11 Anforderungen an den Hersteller

Bei der Verwendung von Attestierungsmechanismen der Chip-Plattform wird entsprechend Schlüsselmaterial verwendet, was von den Herstellern der Chips (also bspw. Intel oder AMD) sicher erzeugt und sicher in die Plattform eingebracht wird. Der Hersteller des PoPP-Service hat darauf keinen Einfluss, muss aber dann entsprechend diese Mechanismen sicher verwenden.

Verwendet der Hersteller des PoPP-Service andere Mechanismen muss er selber Schlüsselmaterial, was im Zuge der Attestierung verwendet wird (Attestierungs-Schlüssel), erzeugen und speichern. Die in den folgenden Anforderungen gemachten Vorgaben bzgl. Attestierungs-Schlüsseln beziehen sich auf dieses Szenario.

A_26617 -PoPP-Service - VAU - Hersteller - Schlüsselqualität Attestierungs- und Autorisierungs-Schlüssel

Der Hersteller des PoPP-Service MUSS

- Schlüssel, die für die Signatur von Attestierungsinformationen (Attestierungs-Schlüssel) - sofern er diese selber erzeugt - und
- Schlüssel, die zur Autorisierung (Signatur) von VAU-Image verwendet werden, entsprechend der Vorgaben aus [gemSpec_Krypt#GS-A_4368*] erzeugen. Dies gilt sowohl für die Schlüssel die direkt für Signaturen von Attestierungsinformationen und VAU-Image verwendet werden als auch für Schlüsselmaterial, dass diese Schlüssel bestätigt (CA-Schlüssel).[<=]

A_26618 -PoPP-Service - VAU - Hersteller - Einbringen und Speichern von Attestierungs-Schlüsseln in VAU

Der Hersteller des PoPP-Service MUSS Schlüssel, die für die Signatur von Attestierungsinformationen von VAU-Image verwendet werden (Attestierungs-Schlüssel), sofern er diese selber erzeugt und verwaltet, im 4-Augen-Prinzip zusammen mit der gematik in die Systeme der VAU einbringen und dort vor Auslesen geschützt speichern. [<=]

A_26619 -PoPP-Service - VAU - Hersteller - Speichern von Autorisierungs- und CA-Schlüsseln

Der Hersteller des PoPP-Service MUSS

- Schlüssel, die für die Autorisierung (Signatur) von VAU-Image verwendet werden,
- Schlüssel, die Attestierungs-Schlüssel bestätigen (CA-Schlüssel), sofern er diese selber verwaltet,
- Schlüssel, die Autorisierungs-Schlüssel bestätigen (CA-Schlüssel),

vor Auslesen und unberechtigten Zugriffen geschützt speichern, sodass diese nur im 4-Augen-Prinzip verwendet werden können.[<=]

A_26620 -PoPP-Service - VAU - Hersteller - Bereitstellung Prüfschlüssel für Attestierung und Autorisierung

Der Hersteller des PoPP-Service MUSS die zur Prüfung von VAU-Image-Attestierungsinformationen und VAU-Image-Autorisierungen (Signaturen) benötigten Prüfschlüssel (bspw. öffentlicher CA-Schlüssel) zur HSM-Einrichtung beim Anbieter mitbringen und vorab der gematik und dem Anbieter bereitstellen.[<=]

Hinweis: Die Bereitstellung von Attestierungs-Prüfschlüsseln an die gematik erfolgt für den Fall, dass der Hersteller diese selbst verwaltet, beim Hersteller vor Ort im Rahmen der gemeinsamen Einbringung der Attestierungs-Schlüssel in die VAU (vgl. A_26618).*

A_26621 -PoPP-Service - VAU - Hersteller - kryptografische Autorisierung von VAU-Image

Der Hersteller des PoPP-Service MUSS von ihm erstellte VAU-Image im 4-Augenprinzip mit dem Autorisierungsschlüssel kryptografisch autorisieren (signieren).[<=]

Hinweis: Eine technische Umsetzung ist die Erzeugung des Hashwerts des VAU-Image und die Signatur mit dem privaten Autorisierungsschlüssel.

A_26622 -PoPP-Service - VAU - Hersteller - Übermittlung autorisierter VAU-Image an Anbieter und gematik

Der Hersteller des PoPP-Service MUSS von ihm erstellte und autorisierte VAU-Image an den Anbieter übermitteln und der gematik mindestens die Autorisierungsinformation bereitstellen.[<=]

Hinweis: Eine technische Umsetzung der Bereitstellung an die gematik ist die Übermittlung des signierten Hashwerts an die gematik.

A_26692 -PoPP-Service - VAU - Hersteller - Protokollierung sicherheitsrelevanter Hersteller-Aktivitäten

Der Hersteller des PoPP-Service MUSS alle Aktivitäten, die den Schutz der Integrität von VAU-Image und die Nachvollziehbarkeit des Rollout von VAU-Image betreffen, protokollieren und dabei festhalten wann, warum, durch wen, welche Aktion durchgeführt wurde. Dies umfasst mindestens Schlüsselerzeugungen, Prüfschlüssel-Importe, VAU-Image-Erzeugung, VAU-Image-Signatur und VAU-Image-Auslieferung. Das Protokoll ist der gematik auf Nachfrage vorzulegen.[<=]

5.2.1.12 Anforderungen an den Anbieter

A_26623 -PoPP-Service - VAU - Gemeinsame Zeremonie zur HSM-Einrichtung

Der Anbieter des PoPP-Service MUSS eine Zeremonie organisieren und durchführen, bei der gemeinsam mit dem Hersteller und der gematik die Einrichtung des HSM vorgenommen wird.[<=]

Hinweis: Eine Zeremonie, die die folgenden Punkte berücksichtigt, setzt die Anforderung um, wobei etwaige Abhängigkeiten zum ZETA Guard hier ggf. noch nicht in Gänze berücksichtigt werden:

1. Das HSM befindet sich im Auslieferungszustand.
2. Auf dem HSM wird das HSM-Modul des Herstellers des PoPP-Service installiert.
3. Der Zugang zum HSM wird so konfiguriert, dass der Anbieter:
 - a. neue signierte Hashwerte von VAU-Image ins HSM importieren kann,
 - b. Schlüssel für TLS-Server- und TLS-Client-Identität erzeugen und dazugehörige CSRs exportieren kann,
 - c. Schlüssel für die ID Token-Verschlüsselung (durch den sektoralen IDP) erzeugen kann,
 - d. Schlüssel für die Access Token-Signatur (durch ZETA Guard) erzeugen kann,
 - e. Schlüssel für die Access Token-Ver-/Entschlüsselung (durch ZETA Guard) erzeugen kann,
 - f. darüber hinaus keine anderen Schlüssel erzeugen, exportieren oder nutzen sowie keine Prüfschlüssel importieren kann, sondern dies nur im Vier-Augen-Prinzip mit der gematik möglich ist.
(Der Export öffentlicher Schlüssel kann und muss immer möglich sein.)
4. Auf dem HSM werden Schlüssel für die folgenden Identitäten bzw. Zwecke erzeugt:
 - a. Identität PoPP-Token-Signatur,
 - b. CA-Schlüsselpaar zur Ableitung von Identitäts-Schlüsselpaaren ("CA-Ident."),

- c. *Identität HSM für Verarbeitungskontext-zu-HSM-Kommunikation (aus CA-Ident.),*
 - d. *Identität PoPP-Verarbeitungskontext (aus CA-Ident.),*
 - e. *Identität ZETA Guard-Verarbeitungskontext (aus CA-Ident.),*
 - f. *Protokoll-Signatur-Schlüssel (Signatur des Protokolls der VAU-Image-Hashwerte und öffentlichen Schlüssel von im HSM erzeugten Identitäten).*
5. *Aus dem HSM werden die öffentlichen Schlüssel für PoPP-Token-Signatur, CA-Ident. und Protokoll-Signatur exportiert.*
6. *Der öffentliche Schlüssel für die PoPP-Token-Signatur-Identität geht:*
- a. *an den Anbieter zur Bekanntmachung via Entity Statement und zur Beantragung der TI-Identität (der Export eines mit dem privaten Schlüssel signierten CSR muss möglich sein) und*
 - b. *an die gematik, um diesen später gegen den im Entity Statement und im TI-Zertifikat enthaltenen Schlüssel abgleichen zu können.*
7. *Der öffentliche Schlüssel der CA-Ident. geht an den Hersteller, der diese über die VAU-Image-Build-Pipeline A_26468-* in die VAU-Image einbindet.*
8. *Der öffentliche Protokoll-Signatur-Schlüssel geht an die gematik zur späteren Prüfung der vom Anbieter übertragenen Protokolle.*
9. *In das HSM werden die folgenden Prüfschlüssel - gebunden an den jeweiligen Zweck - importiert, nachdem sie vorab zwischen Hersteller, Anbieter und gematik nochmals abgeglichen wurden:*
- a. *Prüfschlüssel zur Signatur der Attestierungsinformationen (vom Hersteller; geht bei Eigenverwaltung dieser Schlüssel durch den Hersteller in einer vorgelagerten Zeremonie - zur Einbringung der Attestierungs-Schlüssel in die VAU - an die gematik),*
 - b. *Prüfschlüssel zur Signatur von VAU-Image-Hashwerten (vom Hersteller).*

A_26968 -PoPP-Service - VAU - Prozess für Vertrauensanker-Management

Der Anbieter des PoPP-Service MUSS einen Prozess entwerfen und umsetzen, der das Wechseln oder Hinzufügen von in das HSM importierten Vertrauensankern, sowie das Erneuern von im HSM erzeugten Vertrauensankern zusammen mit dem Hersteller des PoPP-Service und der gematik ermöglicht und diesen so anstoßen, dass mindestens eine Erneuerung von im HSM erzeugten Vertrauensankern erfolgt, bevor die Laufzeit der aktuellen Anker zwei Jahre überschreitet.【<=】

Hinweis: Da von den in das HSM importierten Vertrauensanker ggf. nicht alle unter der Kontrolle des Herstellers des PoPP-Service stehen, ist ein Wechsel oder Hinzufügen eines Vertrauensankers bereits deutlich vor Ablauf von zwei Jahren nicht auszuschließen.

A_26624 -PoPP-Service - VAU - Sichere Erzeugung und Speicherung privater und geheimer Schlüssel der VAU

Der Anbieter des PoPP-Service MUSS alle privaten und geheimen Schlüssel, die für den Betrieb des Dienstes und der VAU benötigt werden, in einem HSM erzeugen und im HSM oder innerhalb eines Verarbeitungskontextes anwenden, bspw. private bzw. geheime Schlüssel, die:

- 1. zur Authentisierung der Verarbeitungskontexte gegenüber von VAU-Clients und Diensten,
 - 2. zur Ver- und Entschlüsselung oder
 - 3. zur Signatur
- genutzt werden.【<=】

Hinweis: Der Schutz der Schlüssel wird entsprechend der Anforderungslage technisch vom Produkt durchgesetzt. Schlüssel werden entweder innerhalb der initialen Zeremonie gemeinsam mit Hersteller und gematik erzeugt oder deren Erzeugung wird vom Anbieter über entsprechende Schnittstellen der VAU ausgelöst. Ausnahmen bilden die Signaturschlüssel für Attestierungsinformationen die in der Hardware der VAU (bspw. TPM) sicher gespeichert sind. Der Anbieter erzeugt nicht selber Schlüssel. Explizit davon ausgenommen und hier in den Anforderungen nicht erwähnt sind die Schlüssel zum Signieren des Entity Statements des PoPP-Service. Diese können unter der Hoheit des Anbieters stehen. Die gematik ist über die Zeremonie zur HSM-Einrichtung und über das vom HSM signierte Protokoll jederzeit in der Lage, die Authentizität der in Entity Statements und Zertifikaten veröffentlichten Schlüssel zu verifizieren.

A_27041 -PoPP-Service - VAU - Prozesse zur Regelmäßigen Erneuerung von Schlüsseln und Zertifikaten

Der Anbieter des PoPP-Service MUSS einen Prozess entwerfen und umsetzen, der gewährleistet, dass:

1. die im Verarbeitungskontext der VAU verarbeiteten Schlüssel der Identitäten für TLS, Access Token-Signatur, Access Token-Verschlüsselung und ID Token-Verschlüsselung und die dafür ggf. notwendigen Zertifikate rechtzeitig vor dem Ablauf ihrer dreimonatigen Laufzeit erneuert werden,
2. die Schlüssel und Zertifikate für die Identitäten zur PoPP-Token-Signatur und APDU-Paket-Signatur rechtzeitig vor dem Ablauf von zwei Jahren (vgl. A_26968*) Laufzeit erneuert werden und
3. die notwendigen Veröffentlichungen der öffentlichen Schlüssel und Zertifikate in JWKs und Entity Statements erfolgt.

[<=]

Hinweis: Entsprechend A_27039- wird die maximale Laufzeit der im ersten Punkt genannten Schlüssel technisch von der VAU durchgesetzt. Entsprechend den Vorgaben der TI-PKI ist die Laufzeit fünf Jahre für die im zweiten Punkt genannten Zertifikate, wobei die Schlüssel abweichend davon nur zwei Jahre genutzt werden sollen. Für die Erneuerung der PoPP-Token-Signatur-Identität ist ein Mehraugenprinzip mit der gematik notwendig. Vergleiche dazu auch A_26968-* weiter oben.*

A_26625 -PoPP-Service - VAU - Eingeschränkte HSM Administration

Der Anbieter des PoPP-Service MUSS sicherstellen, dass der Zugriff auf das HSM so eingeschränkt ist, dass - abgesehen von spezifizierten Ausnahmen - nur im Vier-Augen-Prinzip mit der gematik folgende Aktionen möglich sind:

1. die Erstellung, Sicherung und Wiederherstellung von Schlüsseln,
2. der Import von Prüfschlüssel in das HSM und
3. die Administration des HSM.

[<=]

Hinweis: Die Einrichtung des HSM findet innerhalb der initialen Zeremonie mit der gematik statt. Durch die beschriebene Umsetzung, bei der neue VAU-Image über den Import von signierten Hashwerten der Images im HSM bekannt gemacht werden, ergibt sich keine Abhängigkeit des Anbieters zur gematik im Regelbetrieb, da dieser Import durch den Anbieter eigenständig vorgenommen werden kann.

A_26626 -PoPP-Service - VAU - Einsatz zertifizierter HSM

Der Anbieter des PoPP-Service MUSS HSMs verwenden, deren Eignung durch eine erfolgreiche Evaluierung nachgewiesen wurde. Als Evaluierungsschemata kommen dabei Common Criteria oder Federal Information Processing Standard (FIPS) in Frage. Die Prüftiefe MUSS mindestens:

1. FIPS 140-2/140-3 Level 3 oder
2. Common Criteria EAL 4+ mit hohem Angriffspotenzial entsprechen. [\leq]

A_26627 -PoPP-Service - VAU - Ausschluss von Manipulationen über physische Angriffe

Der Anbieter des PoPP-Service MUSS mit zusätzlichen organisatorischen Mitteln ausschließen, dass ein Angreifer aus dem betrieblichen Umfeld des Anbieters physische Angriffsmittel zur Kompromittierung der VAU zum Einsatz bringen kann.[\leq]

Hinweis: Die Anforderung besteht für den Anbieter zusätzlich zum inA_26598- geforderten Schutz. Die Umsetzung der Anforderung umfasst bspw. Maßnahmen in der Betriebsumgebung, die verhindern, dass Einzelne unbemerkt physischen Zugang bekommen oder dass technische Angriffs-Hilfsmittel (bspw. Flex) in die Betriebsumgebung eingebracht werden.*

A_26628 -PoPP-Service - VAU - Physischer Zugriff auf Systeme der VAU nur im 4-Augen-Prinzip

Der Anbieter des PoPP-Service MUSS gewährleisten, dass physischer Zugriff auf die Hardware der VAU nur im 4-Augen-Prinzip möglich ist.[\leq]

Hinweis: Die Anforderung besteht für den Anbieter zusätzlich zum inA_26598- und A_26597-* geforderten Schutz.*

5.2.1.13 Bereitstellung durch die gematik

Die gematik stellt den ZETA Guard als Docker-Container-Image und eine dazugehörige Signatur bereit.

Die gematik stellt den Prüfschlüssel für die Prüfung der Signatur des ZETA Guard bereit.

5.3 Identitäten und Zertifikate PoPP-Service

5.3.1 Überblick

Die folgende Tabelle gibt einen Überblick über die verwendeten Schlüssel, wo sie erzeugt werden und wo der private bzw. geheime Schlüssel gespeichert und verwendet wird. Öffentliche Schlüssel die als Prüfschlüssel/Vertrauensanker dienen, sind nicht mit aufgeführt. Je nach Umsetzung des ZETA Guard können sich für Schlüssel, die der ZETA Guard benötigt, noch Änderungen ergeben. (VK = Verarbeitungskontext)

Tabelle 6: Übersicht über die im PoPP-Service verwendeten Schlüssel

Schlüssel/Zertifikat & Zweck	Erzeugung	Speicherung	Verwendung im	Verwendung durch
TLS-Server (Richtung PS & PoPP-Modul integrierender App)	HSM der VAU	für VK verschlüsselt	VK ZETA	VK ZETA
PoPP-Token-Signatur	HSM der VAU	HSM der VAU	HSM der VAU	VK PoPP
APDU-Paket-Signatur	HSM der	für VK	VK PoPP	VK PoPP

	VAU	verschlüsselt		
Access Token-Signatur	HSM der VAU	für VK verschlüsselt	VK ZETA	VK ZETA
ID Token-Entschlüsselung	HSM der VAU	für VK verschlüsselt	VK ZETA	VK ZETA
Access Token-Entschlüsselung	HSM der VAU	für VK verschlüsselt	VK ZETA	VK ZETA
Entity Statement-Signatur	Anbieter-spezifisch	Anbieter-spezifisch	Anbieter-spezifisch	Anbieter-spezifisch
CV-Zertifikat	Hersteller-spezifisch	VAU-Image	VK PoPP	VK PoPP
HSM-Identität	HSM der VAU	HSM der VAU	HSM der VAU	HSM der VAU
PoPP-VK-Identität*	HSM der VAU	HSM der VAU	HSM der VAU	VK PoPP
ZETA-VK-Identität*	HSM der VAU	HSM der VAU	HSM der VAU	VK ZETA
HSM-Protokoll-Signatur	HSM der VAU	HSM der VAU	HSM der VAU	HSM der VAU

*sofern getrennte VAU-Image und somit VK verwendet werden

5.3.2 Algorithmus für Schlüsselpaare

A_26498 -PoPP-Service - Schlüsselpaare und X.509-Zertifikate immer auf Basis P-256

Der PoPP-Service, der Hersteller PoPP-Service und der Anbieter PoPP-Service MÜSSEN durchsetzen, dass die für die kryptografischen Operationen und Identitäten im Betrieb des PoPP-Service notwendigen asymmetrischen Schlüsselpaare sowie etwaige dafür erzeugte X.509-Zertifikate (Zertifikatssignaturschlüssel) alle auf der NIST Kurve P-256 basieren, siehe [SP800-186#3.2.1.3].[<=]

A_26954 -PoPP-Service - Schlüsselpaare für CV-Zertifikate immer auf Basis von Brainpool

Der PoPP-Service, der Hersteller PoPP-Service und der Anbieter PoPP-Service MÜSSEN durchsetzen, dass die für die kryptografischen Operationen und Identitäten im Betrieb des PoPP-Service notwendigen asymmetrischen Schlüsselpaare für CV-Zertifikate sowie die dafür erzeugten CV-Zertifikate (Zertifikatssignaturschlüssel) alle auf der folgenden Kurve aus [RFC5639] basieren: brainpoolP256r1.[<=]

5.3.3 Entity Statement

A_26828 -PoPP-Service - Sichere Erzeugung und Speicherung Entity Statement-Signaturschlüssel

Der Anbieter des PoPP-Service MUSS das Schlüsselpaar für die Signatur seines Entity Statements sicher erzeugen und so vor unberechtigtem Zugriff geschützt speichern, dass auch ein potentieller einzelner Innentäter den Signaturschlüssel nicht verwenden kann.
[<=]

5.3.4 PoPP-Service Signaturen

A_26495 -PoPP-Service - PoPP-Token-Signatur-Identität

Der Anbieter des PoPP-Service MUSS für das Signaturschlüsselpaar für PoPP-Token ein Zertifikat mit dem Profil C.ZD.SIG und der Rolle OID 1.2.276.0.76.4.320 (oid_popp-token) aus der auf der NIST-P-256-Kurve basierenden TI-Komponenten-CA, welche aus der gematik-X.509-Root-CA abgeleitet ist, beziehen und dem Hersteller des PoPP-Service zur Verfügung stellen.[<=]

Hinweis: Für den Token-Signatur-Schlüssel wird zusätzlich zur Veröffentlichung via Entity Statement/JWKS (siehe A_26434- und A_26533-*) ein Zertifikat mit entsprechender Rolle aus der Komponenten-PKI der TI ausgestellt und in die PoPP-Token eingebettet. Somit können FD entscheiden, ob sie die Token über den vom Federation Master oder den von der TI-PKI aufgespannten Vertrauensraum prüfen. Der Hersteller benötigt das Zertifikat um das Einbetten dieses Zertifikats in die PoPP-Token umsetzen zu können.*

A_26496 -PoPP-Service - APDU-Paket-Signatur-Zertifikat

Der Anbieter des PoPP-Service MUSS für das Signaturschlüsselpaar für APDU-Pakete ein Zertifikat mit Profil C.ZD.SIG und Rolle OID 1.2.276.0.76.4.293 (oid_popp) aus der auf der NIST-P-256-Kurve basierenden TI-Komponenten-CA beziehen und in den PoPP-Service importieren.[<=]

Hinweis: Der Hersteller benötigt die Information zum Signaturzertifikat um die Erzeugung des Request für die im Folgenden geforderte OCSP-Abfrage umsetzen zu können.

A_26631 -PoPP-Service - APDU-Paket-Signatur - Einbetten OCSP-Response

Der PoPP-Service MUSS in die Signatur von APDU-Paketen eine OCSP-Response für das verwendete Signaturzertifikat einbetten, die nicht älter als 10 min ist.[<=]

5.3.5 TLS

A_26497 -PoPP-Service - TLS-Server-Zertifikat an Client-Schnittstelle

Der Anbieter des PoPP-Service MUSS sicherstellen, dass für die Authentisierung als TLS-Server an allen seinen Client-Schnittstellen ein TLS-Zertifikat verwendet wird, das:

- von einem Herausgeber stammt, der Mitglied des [CAB Forum] ist,
- für die Domäne ausgestellt ist, die zur Verbindung in den Verarbeitungskontext der VAU vorgesehen ist, und
- eine Laufzeit von maximal drei Monaten aufweist.

[<=]

A_26616 -PoPP-Service - TLS-Server-Zertifikate - Certificate Transparency

Der Anbieter des PoPP-Service MUSS

1. das TLS-Zertifikat für seine Clientschnittstelle aus einer CA beziehen, welche Certificate Transparency gemäß [RFC 6962] / [RFC 9162] unterstützt,
2. täglich sicherstellen, dass für die Domäne, die zur Verbindung in den Verarbeitungskontext der VAU vorgesehen ist, keine unbekannten Zertifikate im Certificate Transparency Log gelistet werden und
3. im Fehlerfall (es wird ein unbekanntes Zertifikat gelistet) einen Security Incident entsprechend den Vorgaben zur betrieblichen Sicherheit aus [gemSpec_DS_Anbieter] erstellen.

[<=]

A_26827 -PoPP-Service - TLS-Server-Zertifikate - Certification Authority Authorization (CAA) Records

Der Anbieter des PoPP-Service MUSS für das TLS-Zertifikat für seine Clientschnittstelle Certification Authority Authorization (CAA) DNS Resource Records nach [RFC 6844] bereitstellen, welche die Validität der ausstellenden CA verifizieren.[<=]

Hinweis: Anforderungen zu Schlüsseln/Zertifikaten bei der TLS-Client-Authentisierung bei der Verbindung zum sektoralen IDP finden sich in [gemSpec_IDP_FD], wobei der PoPP-Service die dort genannte Rolle Authorization Server/Anbieter von FD einnimmt.

A_26975 -PoPP-Service - OCSP-Stapling an Client-Schnittstelle

Der PoPP-Service MUSS an der HTTPS-Schnittstelle zum Internet für Clients (PoPP-Modul integrierende Apps und PS) OCSP-Stapling [RFC 6066] unterstützen und dabei OCSP-Responses verwenden, die nicht älter als eine Stunde sind.[<=]

Hinweis: Die OCSP-Adresse ist im "Authority Information Access" (AIA) des Zertifikats zu finden.

5.3.6 CV-Zertifikat

Für die Verifikation der Versichertenidentität via kontaktbehaftet angebundener eGK benötigt der PoPP-Service ein CV-Zertifikat aus der TI-CVC-PKI.

A_26499 -PoPP-Service - CV-Identität für eGK-Kommunikation

Der Hersteller des PoPP-Service MUSS für die Durchführung der Card-to-Card-Authentisierung mit einer eGK ein entsprechendes Schlüsselpaar nach den Vorgaben für CV-Zertifikate in [gemSpec_Krypt] erzeugen und dafür ein CV-Zertifikat beziehen, dass:

1. aus der TI-CVC-PKI abgeleitet ist,
2. eine Flaglist besitzt, die ausschließlich Nullen enthält, und
3. als CHR (12 Byte Wert) einen Zähler verwendet, der mit jedem für den PoPP-Service ausgestellten CV-Zertifikat inkrementiert wird, beginnend mit '0000 80 987 00000 0000000001'.
4. einen öffentlichen Schlüssel auf der Kurve gemäß [RFC 5639] brainpoolP256r1 enthält und das CV-Zertifikat und den privaten Schlüssel in das VAU-Image des PoPP-Service einbringen.

[<=]

Hinweis: Die erste CHR ist in A_26499 mit '0000 80 987 00000 0000000001' spezifiziert. Diese CHR enthält mit KeyID='0000' einen Wert, der in keiner Smartcard verwendet wird. Der Anfang des ICCSN-Teils der CHR weist mit dem Wert '80' auf das Gesundheitswesen hin. Die folgenden drei Stellen '987' stehen als CountryCode zur applikationsspezifischen Verfügung und werden im Regelungsbereich der gematik für TI-interne Zwecke verwendet. Die folgenden fünf Zeichen '00000' stehen in einer ICCSN für den Herausgeber und codieren in der TI den PoPP-Service.

5.3.7 TSL-Handling

Für die Prüfung von eGK-CV- und X.509-Zertifikaten benötigt der PoPP-Service eine aktuelle TSL.

A_27150 -PoPP-Service - TSL - Aktualisierung

Der PoPP-Service MUSS stündlich über den Internet-Downloadpunkt des TSL-Dienstes (vgl. [gemSpec_TSL#A_17680*]) prüfen, ob eine neue TSL zum Download bereitsteht, wenn ja, diese beziehen, und dabei folgendes durchsetzen:

1. Prüfung des TLS-Server-Zertifikats des TSL-Dienstes, inkl. Hostname-Validierung und dass dieses von einer gängigen Internet-CA und auf die "gematik GmbH" ausgestellt ist (vgl. [gemSpec_TSL#TIP1-A_4058*]).
2. Abgleich des Hashwerts der aktuell im System vorliegenden TSL gegen den Hashwert am Downloadpunkt,
3. Wenn der Hashwert unterschiedlich ist, wird die TSL vom Downloadpunkt geladen.
4. Die TSL wird wie folgt geprüft:
 - a. Signaturprüfung:
 - i. Die Signatur ist gegen den in der TSL enthaltenen TSL-Signer gültig.
 - ii. Das Signaturzertifikat (TSL-Signer) ist auf den im System vorhandenen TSL-Vertrauensanker (TSL-Signer-CA-Zertifikat) rückführbar.
 - iii. Das Signaturzertifikat wird per OCSP als "good" beauskunftet.
 - iv. Die Signatur der OCSP-Response ist auf einen in der TSL enthaltenen Signer rückführbar.
 - v. Nur im Falle, dass alle Prüfungen positiv verlaufen sind, wird fortgefahren.
 - b. Die Sequenznummer der geladenen TSL ist höher als die in der im System vorhandenen TSL.
 - c. Die TSL ist zeitlich gültig.

Der PoPP-Service MUSS durchsetzen, dass ausschließlich nachdem die o.g. Punkte erfolgreich, positiv geprüft wurden, die TSL und etwaige mit der TSL transportierte TSL-Vertrauensanker übernommen werden.

Punkt 4. (Prüfung der TSL) des in dieser Anforderung genannten Ablaufs MUSS im Verarbeitungskontext der VAU durchgeführt werden (vgl. [A_27202*]).[<=]

Hinweis: Der OCSP-Responder des TSL-Dienstes ist im Internet erreichbar. Die Adresse des OCSP-Responder ist dem Authority Information Access (AIA) des Signaturzertifikats zu entnehmen.

A_27202 -PoPP-Service - TSL - Proxy für Verarbeitungskontexte

Der PoPP-Service SOLL einen TSL-Proxy umsetzen, der die TSL vom Internet-Downloadpunkt unter Berücksichtigung der Punkte 1. bis 3. aus [A_27150*] lädt und den Verarbeitungskontexten zur Verfügung stellt, um die Last auf dem TSL-Downloadpunkt gering zu halten.[<=]

Hinweis: Das SOLL in A_27202 ermöglicht auf einen solchen Proxy zu verzichten, wenn stets nur ein oder sehr wenige Verarbeitungskontexte des PoPP-Service laufen. Der Verzicht auf die Umsetzung ist mit der gematik abzustimmen.

A_27151 -PoPP-Service - TSL - Prüfung auf Aktualität

Der PoPP-Service MUSS stündlich die Aktualität der ihm vorliegenden TSL prüfen (zeitliche Gültigkeit) und den Anbieter alarmieren, wenn die Gültigkeit sieben Tage unterschreitet. [<=]

A_27152 -PoPP-Service - TSL - Keine abgelaufene TSL verwenden

Der PoPP-Service DARF eine TSL, die zeitlich abgelaufen ist, NICHT verwenden. [≤]

5.4 ZETA Guard im PoPP-Service

Der ZETA Guard im PoPP-Service übernimmt wesentliche Sicherheitsleistungen für den LEI-Zugang zum PoPP-Service (Versichertenzugriffe laufen aktuell noch nicht über ZETA; dies wird in einer Folgestufe von ZETA umgesetzt und dann vom PoPP-Service genutzt). Das Gegenstück zum ZETA Guard des PoPP-Service ist der ZETA Client, der - wie der PoPP-Client - Teil des PS ist. Der ZETA Guard erfüllt folgende Aufgaben:

Die **Policy-basierte Zugriffskontrolle** stellt sicher, dass alle Zugriffe auf den PoPP-Service sicher und autorisiert sind, indem er kontinuierlich die Aktivitäten überwacht und analysiert. Solange der ZETA Guard noch keine Versicherten Clients unterstützt (Stufe 2) werden lediglich die PoPP-Service Schnittstelle I_PoPP-Token_Generation durch den ZETA Guard abgesichert. Über diese findet die fachliche Kommunikation zwischen PoPP-Service und PS statt.

Die **LEI-Authentifizierung mittels SM(C)-B** erfolgt automatisch bei freigeschalteter SM(C)-B, indem der ZETA Client im PS ein ID Token erstellt und mit der SM(C)-B signiert. Dieses Token nutzt DPOP, um sicherzustellen, dass die Anfragen vom autorisierten PS stammen und Replay-Attacken verhindert werden.

Das **Session Management** für den PoPP-Service im ZETA Guard verwendet OAuth2, wobei Access- und Refresh-Token sicher verwaltet und bei Bedarf erneuert werden. Die Token sind durch DPOP an spezifische Client-Instanzen (sprich PS Instanzen) gebunden, um sicherzustellen, dass nur autorisierte Clients Zugriff haben. Bei abgelaufenen Sessions ist eine erneute Authentifizierung erforderlich.

Der PoPP-Service-Anbieter verwendet den von der gematik bereitgestellten ZETA Guard, der gemäß [gemSpec_ZETA] implementiert ist. Die Beschreibung des ZETA Guard sowie Anforderungen an PoPP-Service und PoPP-Service-Anbieter rund um die Einbindung und Verwendung des ZETA Guard finden sich in [gemSpec_ZETA]. In diesem Dokument sind PoPP-Service spezifische Anforderungen und Hinweise rund um das ZETA Guard enthalten (dieses Kapitel und [5.2- Datenschutz und Sicherheit]).

Alle konkreten Informationen und Regelungen zu Bereitstellung, Konfiguration und Verwendung des ZETA Guard sind dem Betriebshandbuch des ZETA Guard-Herstellers zu entnehmen.

5.4.1 Bereitstellung, Konfiguration und Verwendung vom ZETA Guard

Zusätzlich zu den Anforderungen aus [gemSpec_ZETA]- insbesondere A_25773* - gelten für den PoPP-Service-Anbieter die folgenden Anforderungen.

A_26539 -PoPP-Service-Anbieter - Informationspflicht via Betriebshandbuch ZETA Guard Hersteller

Der Anbieter des PoPP-Service MUSS alle Informationen und Regelungen zu Bereitstellung, Konfiguration und Verwendung des ZETA Guard dem Betriebshandbuch des ZETA Guard-Herstellers entnehmen und anwenden. [≤]

A_26540 -PoPP-Service - ZETA Guard - PoPP-Policy erstellen

Der Anbieter des PoPP-Service MUSS bei der Erstellung der Policy für den PoPP-Service mit der gematik zusammenarbeiten. Diese Policy wird über den PAP deployed und

durch die Policy-Engine verarbeitet.

Hinweis: Die PoPP-Service Policy wird in GIT erstellt (CI/CD-Prozess). [<=]

Hinweis: Die Gültigkeitsdauern für Access Token und Refresh Token werden über die PoPP-Service Policy gesteuert.

A_26543 -PoPP-Service - Kommunikation zu den Zero Trust Komponenten der gematik

Der Anbieter des PoPP-Service MUSS sicherstellen, dass die Kommunikation zwischen seinem ZETA Guard und den dem PIP/PAP Server der gematik sowie dem Zero Trust git-Repository der gematik möglich ist.

[<=]

Hinweis: Der Zugriff auf den PIP/ PAP-Server erfolgt über die in A_25670- festgelegte URL; für den PoPP-Service unter dem application Endpunkt popp.*

Die URL inklusive Endpunkt für den PoPP-Service für das Zero Trust git-repository der gematik wird organisatorisch übermittelt

Hinweis: Die Verfügbarkeit von ZETA Guard soll durch die Bereitstellung einer lokalen Container Registry erhöht werden, siehe auch A_28256-..*

5.5 Vertrauenswürdige Uhrzeit im PoPP-Service

Mit der Einführung der Telematikinfrastruktur TI1.0 wurde ein zentraler Zeitdienst eingeführt, mit dem sich alle zentralen Komponenten sowie Konnektoren regelmäßig synchronisieren müssen. Im Rahmen des Sicherheitskonzepts der TI1.0 - als geschlossenes System bekannter Nutzer - wird diese Uhrzeit als vertrauenswürdige angenommen.

Der PoPP-Service stellt für verschiedene (aktuelle und zukünftige) Anwendungsfälle ein Zeugnis über einen real existierenden Versorgungskontext zentral aus. Da der Dienst über das Internet angesprochen wird und Clientsysteme und ein PoPP-Token-nachnutzendes System (PoPP-Verifier) sich mit unterschiedlichen Zeit-Servern synchronisieren könnten als der PoPP-Service, muss die Uhrzeit des PoPP-Service vertrauenswürdige sein.

A_26508 -PoPP-Service - Vertrauenswürdige Uhrzeit

Der Anbieter des PoPP-Service MUSS seine lokale Systemzeit mindestens einmal täglich mit einem qualifizierten Zeitstempel eines eIDAS-Vertrauensdiensteanbieters synchronisieren und dafür Sorge tragen, dass die lokale Systemzeit zwischen zwei Synchronisierungen nie mehr als eine Sekunde vom Sollwert abweicht. [<=]

Hinweis: Das PoPP-Token muss in der Signatur nicht über einen qualifizierten Zeitstempel verfügen, weil das PoPP-Token auch im ersten Anwendungsfall des VSDM 2.0 keine Rolle in den abrechnungsbegründeten Informationen spielt. Durch den sicheren Betrieb des PoPP-Service und die Prüfung dieser Afo im Zulassungsverfahren des PoPP-Service kann die Uhrzeit eines ausgestellten PoPP-Token als zuverlässig betrachtet werden.

Hinweis: Die Bundesnetzagentur listet unter [AnbieterVZeitD] verschiedene Anbieter für qualifizierte Zeitstempel.

5.6 Federation Entity Statement

Der PoPP-Service stellt Kommunikationspartnern notwendige Informationen bereit, indem er ein Entity Statement gemäß [OpenID Federation 1.0] unter <Identifizier-URL>/ .well-known/openid-federation verfügbar macht.

Das Entity Statement beaufkuntet allgemeine Informationen wie:

1. Identifizier (iss),
2. Schlüssel, mit denen das Entity Statement signiert wird (jwks),
3. Ausstellungszeitpunkt (iat),

und Metadaten Informationen zur Konfiguration als:

1. OAuth Authorization Server (oauth_authorization_server),
2. Relying Party (openid_relying_party),
3. OAuth Protected Resource (oauth_resource),
4. Teilnehmer der TI-Föderation (federation_entity).

Die Metadaten enthalten u. a. die Endpunkte, unter denen der PoPP-Service Authorization Server erreichbar ist und Informationen zu Signatur- und Verschlüsselungsschlüssel. Die Tabelle "Entity Statement des PoPP-Service" im Anhang stellt das Entity Statement des PoPP-Service Authorization Server exemplarisch dar.

A_27294 -PoPP-Service - Bereitstellung .well-known für PoPP-Service

Der Anbieter eines PoPP-Service MUSS sicherstellen, dass unter <Identifizier-URL>/ .well-known/openid-federation ein Entity Statement gemäß [OpenID Federation 1.0] veröffentlicht ist. Das Entity Statement MUSS über das Internet erreichbar sein. Das Metadaten-Statement MUSS mindestens die folgenden Werte enthalten:

Tabelle 7: Attribute im well-known document des PoPP-Service

Name	Werte	Beispiel / Beschreibung
iss	URL	URL des PoPP-Service, Identifizier in der TI- Föderation
sub	URL	URL des PoPP-Service, (=iss)
jwks	JWKS Objekt	Federation Entity Key für die Signatur des Entity Statement [OpenID Federation 1.0]
iat	Alle time Werte in Sekunden seit 1970, [RFC7519#section-2]	1645484401 für 2022-02-22 00:00:01
exp	Alle time Werte in Sekunden seit 1970, [RFC7519#section-2]	1645570800 für Ablauf 24h nach iat
authority_hints	[string]	iss Bezeichnung des Federation

		Master (PU: "https://app.federationmaster.de")
--	--	---

[<=]

A 27296 -PoPP-Service - Bereitstellung .well-known als Teilnehmer der TI-Föderation

Der Anbieter eines PoPP-Service MUSS sicherstellen, dass in dem unter <Identifizierungs-URL>/ .well-known/openid-federation gemäß [\[OpenID Federation 1.0\]](#) bereitgestellten Entity Statement die Metadatenfederation_entity als Teilnehmer der TI-Föderation enthält.

Tabelle 8: Attribute des Metadatenblock federation_entity im well-known document des PoPP-Service

Name	Werte	Beispiel
organization_name	String (max. 128 Zeichen) Wertebereich: ^[ÄÖÜäöüß\w\ \.\&\+*V]{1,128}	"Anbieter PoPP-Service GmbH"

Hinweis: Weitere optionale Claims (z.B. contact) sind nach [\[OpenID Federation 1.0\]](#) möglich. [<=]

Hinweis: Anforderungen an die Erstellung und Pflege des Entity Statements als Relying Party der TI-Föderation finden sich in [\[gemSpec_IDP_FD#Kapitel Entity Statements\]](#). Die Tabelle "Entity Statement des PoPP-Service" im Anhang stellt das Entity Statement des PoPP-Service Authorization Server exemplarisch dar.

6 Funktionsmerkmale

6.1 PoPP-Service - Resource Server

Innerhalb des PoPP-Service Resource Server sind die Komponenten und Funktionalitäten zusammengefasst, die zur Erstellung der PoPP-Token als Nachweis des Versorgungskontexts beitragen.

Im Wesentlichen werden die LEI-Informationen (Telematik-ID) und die Versicherten Informationen (KVNR und IK-Nummer) verarbeitet, ggf. temporär gespeichert, und ein PoPP-Token erstellt.

Die Zugriffsautorisierung wird für Primärsysteme (PS) der LEI dabei vom ZETA Guard in Form von signierten Access Token erzeugt und über das DPoP-Verfahren an das Client-System gebunden.

Für den PoPP-Service nutzende Client-Systeme (mobile Apps) der Anwendungen in den Händen der Versicherten, stellt der PoPP-Service Authorization Server entsprechende Access Token aus.

Hinweis: Eine Vertrauensstellung des PoPP-Service (Resource Server) besteht hierbei zum ZETA Guard und zum Authorization Server des PoPP-Service.

A_27102 -PoPP-Service - Verwenden der LEI Daten im Resource Server

Der PoPP-Service MUSS die Telematik-ID und professionOID einer LEI aus dem durch den ZETA Guard gesetzten HTTP Header gemäß [A_25669* -PEP HTTP Proxy - Zusätzliche HTTP-Header] auslesen und verwenden. [≤]

6.1.1 eGK-Handling

6.1.1.1 eGK-Handling, Einführung

Ohne hier auf eine Zerlegung des Systems PoPP-Service einzugehen wird in diesem Kapitel beschrieben und spezifiziert, wie das System PoPP-Service Nachrichten mit einer Smartcard austauscht. Der Transport solcher Nachrichten wird in anderen Kapiteln behandelt. Deshalb ist das Kommunikationsmodell hier einfach: Der PoPP-Service schickt Kommando-APDU zu einer Smartcard und erhält von dort die korrespondierenden Antwort-APDU zur Auswertung.

Der PoPP-Service kommuniziert im Rahmen der folgenden Use Cases mit einer Smartcard:

1. Ein Versicherter "steckt" seine eGK in einen Kartenleser, der über einen PoPP-Client mit dem PoPP-Service kommuniziert. Unterpunkte:
 - a. Der Versicherte befindet sich in einer LEI und
 - i. eGK, kontaktbehaftet in einem eH-KT, oder
 - ii. eGK, kontaktbehaftet in einem Standard-Kartenleser, oder
 - iii. eGK, kontaktlos in einem eH-KT, oder
 - iv. eGK, kontaktlos in einem Standard-Kartenleser.

- b. Der Versicherte und ein LE befinden sich außerhalb einer LEI. Der LE verfügt über ein mobiles Endgerät mit PS und die eGK kommuniziert
 - i. kontaktbehaftet mit einem am PS angeschlossenen Standard-Kartenleser oder
 - ii. kontaktlos mit einem am PS angeschlossenen Standard-Kartenleser.

Im Rahmen der Erzeugung eines PoPP-Token verfolgt der PoPP-Service bei der Kommunikation mit einer Smartcard die folgenden Ziele:

1. Der PoPP-Service überzeugt sich, dass es sich bei der Smartcard um eine echte eGK handelt.
2. Der PoPP-Service überzeugt sich, dass die eGK gültig ist.
3. Der PoPP-Service liest aus der eGK Daten aus, die für die Erstellung des PoPP-Token relevant sind.

Die genannten Ziele werden bei kontaktbehafteter Kartenkommunikation mit einer eGK basierend auf [gemSpec_eGK_ObjSys_G2.1] wie folgt erreicht:

1. Der PoPP-Service baut einen Trusted Channel mit der Identität ID.C.eGK.AUT_CVC.E256 auf. Gelingt dies, dann ist die Echtheit der eGK bestätigt.
2. Der PoPP-Service liest innerhalb des Trusted Channel das X.509-Zertifikat aus der Datei EF.C.CH.AUT.E256 aus und überprüft dieses auf Gültigkeit.
3. Der PoPP-Service konsultiert eine Datenbank, welche die Frage beantwortet: Stammen das CV-Zertifikat der Echtheitsprüfung und das präsentierte X.509-Zertifikat aus ein und derselben eGK? So eine Datenbank wird in[6.1.1.9- eGK-Hash-Datenbank] beschrieben.
4. Der PoPP-Service entnimmt dem ausgelesenen X.509-Zertifikat die für das PoPP-Token notwendigen Informationen.

Die genannten Ziele werden bei kontaktloser Kartenkommunikation mit einer eGK basierend auf [gemSpec_eGK_ObjSys_G2.1](die einen PACE Kanal voraussetzt) wie folgt erreicht:

1. Der PoPP-Service authentisiert die eGK mit der Identität ID.C.eGK.AUT_CVC.E256. Wegen [gemSpec_COS#N107.235)b] ist es nicht möglich dabei einen Trusted Channel zwischen PoPP-Service und eGK zu etablieren.
2. Der PoPP-Service liest aus der eGK das X.509-Zertifikat aus der Datei EF.C.CH.AUT.E256 aus und überprüft dieses auf Gültigkeit. Da der Kommunikationskanal zwischen PoPP-Service und eGK im kontaktlosen Fall nicht Ende-zu-Ende gesichert ist, ist der PoPP-Service nicht ohne weiteres in der Lage zu beurteilen, ob das ihm präsentierte X.509-Zertifikat von derselben eGK stammt, deren Echtheit er mit der Identität ID.C.eGK.AUT_CVC.E256 überprüft hat. Deshalb konsultiert der PoPP-Service im kontaktlosen Fall eine Datenbank, welche die Frage beantwortet: Stammen das CV-Zertifikat der Echtheitsprüfung und das präsentierte X.509-Zertifikat aus ein und derselben eGK? So eine Datenbank wird in[6.1.1.9- eGK-Hash-Datenbank] beschrieben.
3. Der PoPP-Service entnimmt dem präsentierten X.509-Zertifikat die für das PoPP-Token notwendigen Informationen.

Der PoPP-Service schaltet in der eGK nichts frei und erwartet auch nicht, dass in der eGK etwas freigeschaltet ist, insbesondere weder durch Card-2-Card noch durch eine PIN-Eingabe. Technisch ist dies gleichbedeutend mit der Aussage, dass der PoPP-Service nur solche Kommando-APDU an eine eGK sendet, für die im Rahmen der

Objektsystemspezifikation die Zugriffsbedingung "ALWAYS" festgelegt ist ("ALWAYS" = jeder, der im Besitz der Karte ist, ist in der Lage diese Operation auszuführen).

Hinweis: Im kontaktlosen Fall funktioniert eine sinnvolle Kartenkommunikation mit der eGK nur nach Aufbau eines PACE-Kanals. Weil die dazu notwendige CAN auf der eGK aufgedruckt ist und somit jeder, der im Besitz der eGK ist so einen PACE-Kanal aufzubauen in der Lage ist, wird hier der Einfachheit halber die Etablierung eines PACE-Kanals und die damit verbundene Freischaltung von Funktionalität in der eGK auch unter "ALWAYS" subsumiert.

Hinweis: Im kontaktlosen Fall stehen nach Etablierung eines PACE-Kanals dieselben Daten und Funktionen in einer eGK zur Verfügung, wie im kontaktbehafteten Fall unmittelbar nach Stecken einer eGK.

Hinweis: Für die Generation 3 einer eGK ist geplant, dass eine eGK G3 eine Identität besitzt, die sich ohne PIN-Eingabe nutzen lässt und deren zugehöriges X.509-Zertifikat alle Informationen enthält, die für ein PoPP-Token relevant ist.



Abbildung 7: Zustandsdiagramm für die Verarbeitung einer eGK

Abbildung "Zustandsdiagramm für die Verarbeitung einer eGK" zeigt das Zustandsdiagramm für die Verarbeitung einer eGK. Der PoPP-Service wartet in jedem der gezeigten Zustände auf den Empfang einer Nachricht. Er bearbeitet die Nachricht, sendet eine passende Nachricht zurück und geht in den Folgezustand über. Berücksichtigt sind die kontaktbehaftete und die kontaktlose Handhabung einer eGK Generation (G)2.x, sowie die (geplante) Handhabung einer eGK G3 (für welche die Handhabung

kontaktbehaftet und kontaktlos identisch ist). Das Zustandsdiagramm berücksichtigt nur Gutfälle. Das bedeutet, Fehlerfälle, Abbrüche und ähnliches sind im abgebildeten Zustandsdiagramm nicht enthalten.

6.1.1.2 Szenario

In diesem Kapitel ist "Szenario" definiert als eine Abfolge von Elementen in einer Liste. Jedes Element enthält eine Kommando-APDU und eine Liste von Statuswörtern, die als Gutfall für eine Antwort-APDU gewertet werden.

Definition CommandApdu: Eine Kommando-APDU gemäß [gemSpec_COS].

Definition ExpectedStatusWord: Eine Menge mit einem oder mehreren Statuswörtern aus [gemSpec_COS].

Definition Step: Eine Aggregation von genau einer CommandApdu und einem Objekt ExpectedStatusWord.

Definition Szenario: Eine Liste mit keinem, einem oder mehreren Elementen des Typs ScenarioStep.

A_27000 -PoPP-Service, StandardScenarioMessage

Der PoPP-Service MUSS Objekte vom Typ StandardScenarioMessage generieren so, wie in der Schnittstellspezifikation [I_PoPP_Token_Generation.yaml] beschrieben, wobei das Attribut "steps" ein Szenario ist. [≤]

Hinweis: "StandardScenarioMessage.steps" lässt sich auffassen als ein Skript, welches abzuarbeiten ist. Ein "Interpreter" eines "StandardScenarioMessage.steps" wertet das Skript Element für Element aus. Die im Element enthaltene Kommando-APDU wird an eine Smartcard gesendet. Falls die korrespondierende Antwort-APDU der Smartcard ein Statuswort enthält, welches Element der Menge "ExpectedStatusWord" ist, dann fährt der Interpreter mit dem nächsten Listenelement fort, ansonsten bricht er die Bearbeitung des Skripts ab (Exceptionhandling).

Hinweis: Ein "sehr einfacher Interpreter" wird "ExpectedStatusWord" ignorieren und auf die Auswertung der Statuswörter in den Antwort-APDU verzichten. So ein Interpreter erkennt keine Fehlerfälle. Im Fehlerfall wird so ein Interpreter ein möglicherweise sehr langes Skript zeitintensiv auch noch dann fortsetzen, wenn ein "intelligenter Interpreter" erkannt hätte, dass ein Fortsetzen wegen eines Fehlers nicht sinnvoll ist.

A_27017 -PoPP-Service, Erlaubnis für abweichende Szenarien

Falls der PoPP-Service Szenarien verwendet, die von den in [gemSpec_PoPP_Service] beschriebenen abweichen, so MUSS der Hersteller des PoPP-Service vor deren Verwendung eine Erlaubnis der gematik einholen. [≤]

Gemäß Abbildung "Systemkontext PoPP-Lösung" schickt der PoPP-Service StandardScenarioMessages entweder an ein PS, wo die Szenarien vom PoPP-Client bearbeitet oder an einen Konnektor weitergeleitet werden, oder an ein anderes Gerät. Hier ist lediglich die Unterscheidung wichtig, ob eine StandardScenarioMessage von einem Konnektor verarbeitet wird oder nicht. Sobald sich ein Gerät mit dem PoPP-Service verbindet, teilt es dem PoPP-Service mit, ob Szenarien von einem Konnektor verarbeitet werden, oder nicht (Property "cardConnectionType" in der "StartMessage").

A_27128 -PoPP-Service, Codierung von Konnektor-Szenarien

Falls der PoPP-Service ein Szenario für einen Konnektor zusammenstellt, dann MUSS er das Szenario in eine ConnectorScenarioMessage gemäß [I_PoPP_Token_Generation.yaml] einstellen. [≤]

A_27129 -PoPP-Service, Codierung von Nicht-Konnektor-Szenarien

Falls der PoPP-Service ein Szenario nicht für einen Konnektor, sondern für ein anderes Gerät zusammenstellt, dann MUSS er das Szenario in eine StandardScenarioMessage gemäß [I_PoPP_Token_Generation.yaml] einstellen.【<=】

6.1.1.3 eGK öffnen

Dieses Kapitel beschreibt, wie ermittelt wird, ob es sich bei der präsentierten Smartcard um eine eGK handelt und welche Version diese hat. Die Vorgehensweise ist unabhängig davon, ob die Smartcard kontaktbehaftet oder kontaktlos betrieben wird. Deshalb wird hier auf eine diesbezügliche Unterscheidung verzichtet.

Die gematik stellt mit der Prüfkarte eGK eine elektronische Identität zur Überprüfung verschiedener Anwendungsfälle in der TI bereit. Diese wird vorrangig von Dienstleistern vor Ort (DVOs) genutzt. Die Prüfkarte eGK ist nicht für die Nutzung im regulären Versorgungsalltag von Leistungserbringern oder Versicherten vorgesehen. Um die Prüfkarte eGK auch mit PoPP nutzen zu können, muss sie im PoPP-Service wie eine normale eGK behandelt werden.

A_28040 -PoPP-Service - Verarbeiten von Prüfkarte eGK

Der PoPP-Service MUSS eine Prüfkarte eGK wie eine normale eGK behandeln.【<=】

A_27008 -PoPP-Service, Szenario SceOpenEgk, eGK öffnen

Der PoPP-Service MUSS im ersten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 a4 040c 07 D2760001448000'
 - b. ExpectedStatusWord: {'9000'}
2. Element:
 - a. CommandApdu: '00 b0 9100 00'
 - b. ExpectedStatusWord: {'9000', '6281'}

【<=】

Hinweis: Das erste Listenelement selektiert das Masterfile (MF) einer eGK. Dieses Kommando bringt eine eGK in einen für die nachfolgenden Kommandos definierten Zustand. Antwortet die Smartcard auf dieses Kommando mit einem erwarteten Statuswort, dann handelt es sich um eine eGK (oder um eine Smartcard, die vorgaukelt eine eGK zu sein). Wird irrtümlich eine Karte mit falschem Typ angesprochen (etwa ein HBA oder eine Bankkarte), dann wird das durch ein inakzeptables Statuswort erkannt. Falls eine Karte vorgaukelt eine eGK zu sein, dann wird dies in einem späteren Szenario erkannt.

Hinweis: Das zweite Listenelement liest den Inhalt von EF.Version2. Basierend auf den Versionsinformationen ist es möglich eGK unterschiedlicher Generationen zu erkennen, oder beispielsweise wegen Schwachstellen abzulehnen.

Definition: eGKIncludedPtvObjSys ist eine Menge mit Produkttypversionen von eGK Objektsystemen, die der PoPP-Service zu unterstützen hat. Produkttypversionen werden in diese Menge aufgenommen oder aus ihr entfernt, wenn sich die Anforderungslage (also die Vorgaben der gematik) ändern. Basierend auf den Erfahrungen der Vergangenheit ist davon auszugehen, dass sich diese Menge nur wenige Male pro Jahr ändert, während sich die Anzahl zugelassener Kartenprodukte häufiger ändert. Deshalb wird die Menge zulässiger Kartenprodukte über die Produkttypversion definiert.

Definition: eGKexcludedPiObjSys ist eine Menge mit Produktidentifikationen aktiver Objektsysteme, die von der Verwendung durch den PoPP-Service ausgeschlossen werden. Produktidentifikationen der Kartenhersteller werden in diese Menge aufgenommen, wenn es Schwierigkeiten mit einem konkreten Kartenprodukt gibt. Basierend auf den Erfahrungen der Vergangenheit ist davon auszugehen, dass die Mächtigkeit der Menge klein ist und sich nur selten ändert.

A_27018 -PoPP-Service, Zulässige eGK Objektsystemversionen

Der PoPP-Service MUSS Änderungen an der Menge eGKincludedPtvObjSys, die ihm ausschließlich durch die gematik angezeigt werden, innerhalb von sieben Tagen im Betrieb berücksichtigen. [\leq]

A_27019 -PoPP-Service, Unzulässige eGK Objektsystemversionen

Der PoPP-Service MUSS Änderungen an der Menge eGKexcludedPiObjSys, die ihm ausschließlich durch die gematik angezeigt werden, innerhalb von 24 Stunden im Betrieb berücksichtigen. [\leq]

Hinweis: Die Produkttypversion eines Kartenproduktes findet sich in der Datei EF.Version2. Basierend auf der Anforderungslage der gematik besteht die Menge eGKincludedObjSys im November 2025 aus folgenden Elementen: {'040400', '040401', '040500', '040501', '040502', '040600', '040700'}. Mit Einführung der eGK G3 wird die Menge (laut aktuellem Plan) um den Wert '050000' ergänzt.

Hinweis: Die Produktidentifikation des aktiven Objektsystems findet sich in der Datei EF.Version2. Basierend auf dem Kenntnisstand November 2025 ist die Menge eGKexcludedObjSys leer.

A_27009 -PoPP-Service, Auswertung SceOpenEgk

Der PoPP-Service MUSS die Kartenantworten auf das Szenario SceOpenEgk wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceOpenEgk beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Der Inhalt der Datei EF.Version2 wird gemäß [gemSpec_Karten_Fach_TIP_G2.1#2.1.1] auswerten:
 - a. Falls die Version des aktiven Objektsystem (PT_ObjSys) nicht Element der Menge eGKincludedPtvObjSys ist, dann bricht der Use Case mit der Fehlermeldung InvalidPtvObjectSystem ab.
 - b. Falls die Produktidentifikation des aktiven Objektsystem (PI_ObjSystem) Element der Menge eGKexcludedPiObjSys ist, dann bricht der Use Case mit der Fehlermeldung InvalidPiObjectSystem ab.
 - c. Falls die Version des aktiven Objektsystems (PT_ObjSys) eine eGK der
 - i. Generation 2 anzeigt, dann wird im
 - A. kontaktbehafteten Fall mit dem Szenario SceReadCvc aus [A_27001*] fortgefahren.
 - B. kontaktlosen Fall mit dem Szenario SceAuthG2 aus [A_27020*] fortgefahren.
 - ii. Generation 3 anzeigt, dann wird mit dem Szenario SceAuthG3 aus [A_27022*] fortgefahren.

[\leq]

6.1.1.4 eGK G2 kontaktbehaftet

Dieses Kapitel behandelt die kontaktbehaftete Kommunikation mit einer eGK gemäß [gemSpec_eGK_ObjSys_G2.1]. Dies deckt folgende Anwendungsfälle ab:

1. Versicherter in einer LEI, eGK kontaktbehaftet in einem eH-KT
2. Versicherter in einer LEI, eGK kontaktbehaftet in einem Standard-Kartenleser
3. Versicherter und LE außerhalb einer LEI, LE mit mobilem Endgerät mit PS und kontaktbehaftetem Standard-Kartenleser
4. Versicherter mit Versichertenendgerät am PoPP-Service und kontaktbehaftetem Standard-Kartenleser

Im Gutfall schickt der PoPP-Service drei weitere Szenarien mit jeweils mehreren Kommando-APDU an die Karte:

A_27001 -PoPP-Service, Szenario SceReadCvc

Der PoPP-Service MUSS bei kontaktbehafter Kommunikation mit einer eGK G2 im zweiten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 b0 8700 00'
 - b. ExpectedStatusWord: {'9000', '6281'}
2. Element:
 - a. CommandApdu: '00 b0 8600 00'
 - b. ExpectedStatusWord: {'9000', '6281'}
3. Element:
 - a. CommandApdu: '80 ca 0100 00 0000'
 - b. ExpectedStatusWord: {'9000'}

[<=]

Hinweis: Das erste Listenelement liest den Inhalt von EF.C.CA.CS.E256 mit CVC-Sub-CA aus. Der PoPP-Service benötigt dieses CV-Zertifikat zur Verifikation des End-Entity-CVC (sofern er es nicht aus anderen Quellen bereits kennt).

Hinweis: Das zweite Listenelement liest den Inhalt von EF.C.eGK.AUT_CVC.E256 mit dem End-Entity-CVC aus. Der PoPP-Service benötigt dieses CV-Zertifikat für die Berechnung der Sessionkeys.

Hinweis: Das dritte Listenelement (LIST PUBLIC KEY) liest die in der Smartcard verfügbaren öffentlichen Schlüssel aus. Basierend auf dieser Liste ist es dem PoPP-Service möglich eine optimale Chain von CV-Zertifikaten zusammenzustellen für den Import seines CV-Authentisierungsschlüssels (siehe Szenario SceTC1). Im besten Fall ist kein CV-Zertifikatsimport erforderlich, weil die eGK bereits über den CV-Authentisierungsschlüssel des PoPP-Service verfügt.

A_27010 -PoPP-Service, Auswertung SceReadCvc

Der PoPP-Service MUSS die Kartenantworten auf das Szenario SceReadCvc wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceReadCvc beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Der Use Case wird mit der Fehlermeldung InvalidCaCvc beendet, wenn eine Prüfung des CV-Zertifikates gemäß [gemSpec_COS#(N095.900)b] aus der

Datei EF.C.CA.CS.E256 gegen das zugehörige CVC-Root-CA aus der TSL-Liste mit einem Fehler endet, dabei gilt:

- a. affectedObjectaus [gemSpec_COS#(N095.900)b] entspricht dem öffentlichen Signaturprüfchlüssel für das CV-Zertifikat, wie er beispielsweise aus einem CVC-Root-CA entnehmbar ist.
 - b. pointInTime aus [gemSpec_COS] entspricht der lokalen, aktuellen Systemzeit.
3. Der Use Case wird mit der Fehlermeldung InvalidEndEntityCvc beendet, wenn eine Prüfung des CV-Zertifikates gemäß [gemSpec_COS#(N095.900)b] aus der Datei EF.C.eGK.AUT_CVC.E256 mit einem Fehler endet, dabei gilt:
- a. affectedObjectaus [gemSpec_COS#(N095.900)b] entspricht dem öffentlichen Signaturprüfchlüssel für das CV-Zertifikat, wie er beispielsweise aus dem CV-Zertifikat aus der Datei EF.C.CA.CS.E256 entnehmbar ist.
 - b. pointInTimeaus [gemSpec_COS] entspricht der lokalen, aktuellen Systemzeit.
4. Dem Antwortdatenfeld auf das LIST PUBLIC KEY Kommando wird eine Liste von Schlüsselreferenzen gemäß [gemSpec_COS#(N099.462)] entnommen und für die Verwendung im Szenario SceTC1 zwischengespeichert.

[<=]

A_27002 -PoPP-Service, Szenario SceTC1

Der PoPP-Service MUSS bei kontaktbehafteter Kommunikation mit einer eGK G2 im dritten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 22 41A4 06 (84-01-09) || (80-01-54)'
 - b. ExpectedStatusWord: {'9000'}
2. Kein, ein oder mehrere Paare von MSE SET und PSO Verify Certificate Kommandos zum Import des öffentlichen CV-Authentisierungsschlüssels des PoPP-Service. CAR bezeichnet das Feld CAR aus dem zu importierenden CV-Zertifikat, CvcTemplate bezeichnet das Template des zu importierenden CV-Zertifikates.
 - a. Element:
 - i. CommandApdu: '00 22 81 b6 0a (83-08-CAR)'
 - ii. ExpectedStatusWord: {'9000'}
 - b. Element:
 - i. CommandApdu: '00 2a 00 be xy CvcTemplate'
 - ii. ExpectedStatusWord: {'9000'}
3. Element:
 - a. CommandApdu: '10 86 0000 10 (7c-0e-(c3-0c-CHR) 00)'
 - b. ExpectedStatusWord: {'9000'}

[<=]

Hinweis: Das erste Listenelement ist ein MSE Set Kommando gemäß [gemSpec_COS#(N100.900)] zur Selektion des privaten Schlüssels PrK.eGK.AUT_CVC.E256 für den Algorithmus elcSessionkey4SM. Dieser Schlüssel ist kartenseitig am Aufbau des Trusted Channels beteiligt.

Hinweis: Jedes Paar aus MSE Set Kommando und PSO Verify Certificate Kommando importiert einen öffentlichen Schlüssel in die Smartcard.

1. *SceTC1 enthält kein solches Paar, wenn der öffentliche CV-Authentisierungsschlüssel des PoPP-Service bereits in der Smartcard gespeichert ist.*
2. *SceTC1 enthält genau ein solches Paar, wenn der öffentliche Schlüssel zur Verifikation des End-Entity-CVC des PoPP-Service bereits in der Smartcard gespeichert ist:
CVC-Chain = End-Entity-CVC.*
3. *SceTC1 enthält genau zwei solche Paare, wenn der öffentliche Root-Schlüssel in der Smartcard gespeichert ist, mit dem sich das CVC-Sub-CA zum End-Entity-CVC des PoPP-Service verifizieren lässt:
CVC-Chain = CVC-Sub-CA, End-Entity-CVC.*
4. *SceTC1 enthält genau drei solche Paare, wenn der öffentliche Root-Schlüssel in die Smartcard zu importieren ist, mit dem sich das CVC-Sub-CA zum End-Entity-CVC des PoPP-Service verifizieren lässt:
CVC-Chain = Link-CVC-Root, CVC-Sub-CA, End-Entity-CVC.*
5. *SceTC1 enthält mehr als drei solcher Paare, wenn mehr als ein CVC-Root-Schlüssel in die Smartcard zu importieren ist.*

Hinweis: MSE Set Kommando gemäß [gemSpec_COS#(N103.300)] zur Selektion eines öffentlichen Signaturprüfchlüssels in der Smartcard. Der Wert CAR entspricht dem Wertfeld des Elementes CAR des CV-Zertifikates, welches im nächsten Kommando importiert wird.

Hinweis: PSO VerifyCertificate Kommando gemäß [gemSpec_COS#(N095.410)] zum Import eines CV-Zertifikates. Der Wert CvcTemplate entspricht dem Wertfeld (value-field) des BER-TLV codierten CV-Zertifikates.

Hinweis: Das letzte Listenelement ist ein GENERAL AUTHENTICATE Kommando gemäß [gemSpec_COS#(N085.012)]. Der Wert CHR entspricht dem Wertfeld des Elementes CHR des End-Entity-CVC. Dies ist der erste Schritt zur Etablierung eines Trusted Channels zwischen PoPP-Service und Smartcard. Der zweite und letzte Schritt wird in SceReadX.509 ausgeführt.

A_27011 -PoPP-Service, Auswertung SceTC1

Der PoPP-Service MUSS die Kartenantworten auf das Szenario SceTC1 wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceTC1 beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Dem Antwortdatenfeld des GENERAL AUTHENTICATE Kommandos wird ein ephemerer, öffentlicher Schlüssel ephemeralPK_eGK gemäß [gemSpec_COS#(N085.052)h] entnommen.

[<=]

A_27003 -PoPP-Service, Szenario SceReadX.509

Der PoPP-Service MUSS bei kontaktbehafteter Kommunikation mit einer eGK G2 im vierten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 86 0000 xy (7c-xy-(85-xy-ephemeralPK_PoPP-Service))'
 - b. ExpectedStatusWord: {'9000'}
2. Element:
 - a. CommandApdu*: '00 a4 040c 0a a000000167455349474e'
 - b. ExpectedStatusWord: {'9000'}

3. Element:

- a. CommandApdu*: '00 b0 8400 00 0000'
- b. ExpectedStatusWord: {'9000', '6281'}

[<=]

Hinweis: Das erste Listenelement ist ein GENERAL AUTHENTICATE Kommando gemäß [gemSpec_COS#(N085.016)] mit ephemeralPK_PoPP-Service als ephemerem öffentlichen Schlüssel des PoPP-Service. Dies ist der zweite und letzte Schritt zur Etablierung eines Trusted Channels. Der PoPP-Service ist nun in der Lage mit dem von ihm erzeugten ephemeren Schlüsselpaar, seinem privaten CV-Authentisierungsschlüssel und ephemeralPK_eGK aus A_27011- Sessionkeys gemäß [gemSpec_COS#(N85.054)c] zu berechnen. Alle folgenden Kommandos sind mit den vereinbarten Sessionkeys zu sichern gemäß den Regeln aus [gemSpec_COS#13]. Dies wird in der Anforderung durch einen '*' hinter CommandApdu angedeutet.*

Hinweis: Das zweite Listenelement selektiert das Verzeichnis DF.ESIGN. Das Kommando ist in A_27003- im Klartext notiert. Im real verwendeten Szenario ist es gemäß den Regel aus [gemSpec_COS#13] zu sichern. Die zugehörige Antwort-APDU wird von der Smartcard gemäß [gemSpec_COS#13] gesichert.*

Hinweis: Das dritte Listenelement liest das X.509-Zertifikat aus der Datei EF.C.CH.AUT.E256. Das Kommando ist in A_27003- im Klartext notiert. Im real verwendeten Szenario ist es gemäß den Regel aus [gemSpec_COS#13] zu sichern. Die zugehörige Antwort-APDU wird von der Smartcard gemäß [gemSpec_COS#13] gesichert.*

A_27006 -PoPP-Service, Szenario mit APDU innerhalb eines Trusted Channels

Sobald der PoPP-Service einen Trusted Channel zu einer Smartcard etabliert hat, MUSS er jede nachfolgende Kommando-APDU gemäß den Regeln aus [gemSpec_COS#13] sichern und jede nachfolgende Antwort-APDU gemäß den Regeln aus [gemSpec_COS#13] in eine ungesicherte Antwort-APDU umwandeln und dabei die ausgehandelten Sessionkeys verwenden.[<=]

A_27013 -PoPP-Service, Auswertung SceReadX.509

Der PoPP-Service MUSS die Kartenantworten auf das Szenario SceReadX.509 wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceReadX509 beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Aus dem Antwortdatenfeld der letzten Antwortnachricht wird ein X.509-Zertifikat erzeugt. Der Use Case wird mit der Fehlermeldung InvalidX509 beendet, wenn die Prüfung dieses Zertifikates gemäß [A_27130*] fehlschlägt.
3. Das CV-Zertifikat aus der Datei EF.C.eGK.AUT_CVC.E256 sowie das X.509-Zertifikat werden der eGK-Hash-Datenbank (siehe [A_27046*]) in der Methode "check(cvc, x509, "T=1")" übergeben. Falls die Funktion mit "blocked" antwortet, wird der Use Case mit der Fehlermeldung InvalidCertificatePairT1 beendet.

[<=]

6.1.1.5 eGK G2 kontaktlos

Dieses Kapitel behandelt die kontaktlose Kommunikation einer eGK gemäß [gemSpec_eGK_ObjSys_G2.1]. Dies deckt folgende Anwendungsfälle ab:

1. Versicherter in einer LEI, eGK kontaktlos in einem eH-KT
2. Versicherter in einer LEI, eGK kontaktlos in einem Standard-Kartenleser

3. Versicherter und LE außerhalb einer LEI, LE mit mobilem Endgerät mit PS und kontaktlosem Standard-Kartenleser
4. Versicherter mit Versichertenendgerät am PoPP-Service und kontaktlosem Standard-Kartenleser

Im Gutfall schickt der PoPP-Service nach dem in A_27008-* beschriebenen Szenario ein weiteres Szenario mit einer Reihe von Kommando-APDU an die Karte:

A_27020 -PoPP-Service, Szenario SceAuthG2

Der PoPP-Service MUSS bei kontaktloser Kommunikation mit einer eGK G2 im zweiten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 b0 8700 00'
 - b. ExpectedStatusWord: {'9000', '6281'}
2. Element:
 - a. CommandApdu: '00 b0 8600 00'
 - b. ExpectedStatusWord: {'9000', '6281'}
3. Element:
 - a. CommandApdu: '00 a4 040c 0a a000000167455349474e'
 - b. ExpectedStatusWord: {'9000'}
4. Element:
 - a. CommandApdu: '00 22 41A4 06 (84-01-09) || (80-01-00)'
 - b. ExpectedStatusWord: {'9000'}
5. Element:
 - a. CommandApdu: '00 b0 8400 00 0000'
 - b. ExpectedStatusWord: {'9000', '6281'}
6. Element:
 - a. CommandApdu: '00 88 0000 18 token 00'
 - b. ExpectedStatusWord: {'9000'}

【<=】

Hinweis: Das erste Listenelement liest den Inhalt von EF.C.CA.CS.E256 mit CVC-Sub-CA aus. Der PoPP-Service benötigt dieses CV-Zertifikat zur Verifikation des End-Entity-CVC (sofern er es nicht aus anderen Quellen bereits kennt).

Hinweis: Das zweite Listenelement liest den Inhalt von EF.C.eGK.AUT_CVC.E256 mit dem End-Entity-CVC aus. Der PoPP-Service benötigt dieses CV-Zertifikat für die Authentisierung.

Hinweis: Das dritte Listenelement selektiert das Verzeichnis DF.ESIGN.

Hinweis: Das vierte Listenelement ist ein MSE Set Kommando gemäß [gemSpec_COS#(N100.900)] zur Selektion des privaten Schlüssels PrK.eGK.AUT_CVC.E256 für den Algorithmus elcRoleAuthentication.

Hinweis: Das fünfte Listenelement liest das X.509-Zertifikat aus der Datei EF.CH.AUT.E256.

Hinweis: Das sechste Listenelement ist ein INTERNAL AUTHENTICATE Kommando gemäß [gemSpec_COS#(N086.400)].

A_27021 -PoPP-Service, Auswertung SceAuthG2

Der PoPP-Service MUSS die Kartenantworten auf das Szenario SceAuthG2 wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceAuthG2 beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Der Use Case wird mit der Fehlermeldung InvalidCaCvc beendet, wenn eine Prüfung des CV-Zertifikates gemäß [gemSpec_COS#(N095.900)b] aus der Datei EF.C.CA.CS.E256 gegen das zugehörige CVC-Root-CA aus der TSL-Liste mit einem Fehler endet, dabei gilt:
 - a. affectedObjectaus [gemSpec_COS#(N095.900)b] entspricht dem öffentlichen Signaturprüfchlüssel für das CV-Zertifikat, wie er beispielsweise aus einem CVC-Root-CA entnehmbar ist.
 - b. pointInTimeaus [gemSpec_COS] entspricht der lokalen, aktuellen Systemzeit.
3. Der Use Case wird mit der Fehlermeldung InvalidEndEntityCvc beendet, wenn eine Prüfung des CV-Zertifikates gemäß [gemSpec_COS#(N095.900)b] aus der Datei EF.C.eGK.AUT_CVC.E256 mit einem Fehler endet, dabei gilt:
 - a. affectedObjectaus [gemSpec_COS#(N095.900)b] entspricht dem öffentlichen Signaturprüfchlüssel für das CV-Zertifikat, wie er beispielsweise aus dem CV-Zertifikat aus der Datei EF.C.CA.CS.E256 entnehmbar ist.
 - b. pointInTimeaus [gemSpec_COS] entspricht der lokalen, aktuellen Systemzeit.
4. Aus dem Antwortdatenfeld der vorletzten Antwortnachricht wird ein X.509-Zertifikat erzeugt. Der Use Case wird mit der Fehlermeldung InvalidX509 beendet, wenn die Prüfung dieses Zertifikates gemäß [A_27130*] fehlschlägt.
5. Dem Antwortdatenfeld der letzten Antwortnachricht wird eine Signatur entnommen. Die Signatur wird mit dem öffentlichen Schlüssel aus dem CV-Zertifikat aus der Datei EF.C.eGK.AUT_CVC.E256 gegen das Token aus A_27020 Punkt 6.a geprüft. Der Use Case bricht mit der Fehlermeldung InvalidAuthentication ab, wenn diese Signaturprüfung fehlschlägt.
6. Die eGK-Hash-Datenbank wird befragt, ob das CV-Zertifikat aus der Datei EF.C.eGK.AUT_CVC.E256 sowie das X.509-Zertifikat aus ein und derselben eGK stammen (siehe Funktion "check(cvc, x509, "T=CL") in [A_27046*]"). Falls die Funktion mit:
 - a. "unknown" antwortet, dann wird der Use Case mit der Fehlermeldung UnknownCertificates beendet.
 - b. "mismatch" antwortet, dann wird der Use Case mit der Fehlermeldung InvalidCertificatePairContactless beendet.

[<=]

6.1.1.6 eGK G3 kontaktbehaftet und kontaktlos

Hinweis: Dieses Kapitel (sowie einige andere Stellen in diesem Dokument) behandeln eine eGK der Generation 3, so wie es Stand März 2025 geplant war. Zwischenzeitlich hat sich die Planung der gematik dahingehend geändert, dass es Stand November 2025 keine konkreten Pläne für eine eGK-Generation 3 gibt. Es ist denkbar, dass in einer zukünftigen eGK-Spezifikation eine ohne PIN nutzbare X.509-Identität gibt, welche von PoPP nutzbar ist. Dieses Kapitel wird relevant, wenn eine solche Identität auf einer eGK verfügbar ist. In der TI werden derzeit eine Reihe von Use Cases diskutiert/etabliert, bei denen sich ein

Nutzer mit eGK ohne PIN authentisiert. Falls die eGK dabei über eine passende X.509 Identität verfügt, dann vereinfachen sich diese Use Cases aus technischer Sicht, weil Challenge Response-Verfahren dann Standard-Software Bibliotheken nutzen und eine Datenbank überflüssig ist, welche die Zusammengehörigkeit von CV-Zertifikat und X.509-Zertifikat bestätigt.

Dieses Kapitel behandelt die kontaktbehaftet und die kontaktlose Kommunikation einer eGK der Generation 3, so wie es Stand März 2025 geplant ist. Dies deckt alle Anwendungsfälle aus [6.1.1.1- eGK-Handling, Einführung] ab.

Im Gutfall schickt der PoPP-Service nach dem in A_27008-* beschriebenen Szenario ein weiteres Szenario mit einer Reihe von Kommando-APDU an die Karte:

A_27022 -PoPP-Service, Szenario SceAuthG3

Der PoPP-Service MUSS bei einer Kommunikation mit einer eGK G3 im zweiten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 a4 040c 0a a000000167455349474e'
 - b. ExpectedStatusWord: {'9000', '6281'}
2. Element:
 - a. CommandApdu: '00 22 41B6 06 (84-01-91) || (80-01-00)'
 - b. ExpectedStatusWord: {'9000', '6281'}
3. Element:
 - a. CommandApdu: '00 b0 9100 00 0000'
 - b. ExpectedStatusWord: {'9000', '6281'}
4. Element:
 - a. CommandApdu: '00 2a 9e9a xy hashChallenge 00'
 - b. ExpectedStatusWord: {'9000', '6281'}

[<=]

Hinweis: Das erste Listenelement selektiert das Verzeichnis DF.ESIGN.

Hinweis: Das zweite Listenelement ist ein MSE Set Kommando gemäß [gemSpec_COS#(N102.900)] zur Selektion des privaten Schlüssels PrK.CH.AutU.E256 für den Algorithmus signECDSA.

Hinweis: Das dritte Listenelement liest das X.509-Zertifikat aus der Datei EF.CH.AutU.E256.

Hinweis: Das vierte Listenelement ist ein PSO Compute Digital Signature Kommando gemäß [gemSpec_COS#(N087.500)] wobei das Kommandodatenfeld einen Hashwert über eine Challenge enthält. Die zugehörige Antwort-APDU wird im Erfolgsfall eine Signatur zur Challenge enthalten.

A_27023 -PoPP-Service, Auswertung SceAuthG3

Der PoPP-Service MUSS die Kartenantworten auf das Szenario SceAuthG3 wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceAuthG3 beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Aus dem Antwortdatenfeld der vorletzten Antwortnachricht wird ein X.509-Zertifikat erzeugt.

3. Die Signatur im Antwortdatenfeld der letzten Antwortnachricht wird mit dem Signaturprüf Schlüssel aus dem X.509-Zertifikat geprüft. Der Use Case wird mit der Fehlermeldung InvalidAuthentication beendet, wenn die Signaturprüfung oder die Prüfung des X.509-Zertifikates gemäß [A_27130*] fehlschlägt.

[<=]

6.1.1.7 Prüfung des X.509-Zertifikates einer eGK

In A_27013-*, A_27021-* und A_27023-* wird gefordert, dass der PoPP-Service das aus einer eGK ausgelesene X.509-Zertifikat zu prüfen hat. Dies geschieht wie folgt:

A_27130 -PoPP-Service, Prüfen von X.509-Zertifikaten einer eGK

Der PoPP-Service MUSS das aus einer eGK ausgelesene X.509-Zertifikat in Anlehnung an [gemSpec_PKI#TUC_PKI_018] mit den folgenden Eingangsparametern prüfen:

1. x509: das zu prüfende Zertifikat,
2. referenceTime: die aktuelle Systemzeit als Referenzzeitpunkt,
3. policyList = {policyIdentifier = <oid_egk_aut>},
4. keyUsage = digitalSignature,
5. extendedKeyUsage = {keyPurposeld = id-kp-clientAuth},
6. ocspGracePeriod = default ,
7. offlineModus = nein,
8. beigefügteOcspResponse: entfällt,
9. timeoutParameter = 10 Sekunden,
10. TOLERATE_OCSP_FAILURE = false,
11. prüfModus = OCSP.

Zu beachten ist A_23225*, wobei die Gültigkeitsdauer D auf 12 Stunden zu setzen ist.

[<=]

Hinweis: Der OCSP-Responder des TSP-eGK ist im Internet erreichbar. Die Adresse des OCSP-Responder ist dem Authority Information Access (AIA) des eGK-Zertifikats zu entnehmen.

6.1.1.8 eGK-Handling Fehlercodes

In den vorherigen Kapiteln zum eGK-Handling werden diverse Fehlermeldungen definiert. Dabei ist es das Ziel für jede Stelle, an der ein Fehler detektierbar ist, eine eigene Fehlermeldung zu definieren, für den Fall, dass es an der Außenschnittstelle relevant ist. Derart aussagekräftige Fehlermeldungen unterstützen in der Entwicklungs- und Testphase das Debugging. Für den realen Betrieb ist zweifelhaft, ob aussagekräftige Fehlermeldungen sinnvoll sind. Mitunter wird gewünscht einem Angreifer nicht zu viel darüber zu verraten, an welcher Stelle genau sein Angriff scheiterte. Konkret auf das eGK-Handling bezogen ist es aus Sicht menschlicher Nutzer eher so, dass eine eGK sich eignet um ein PoPP-Token zu erstellen, oder eben nicht. Die Tabelle in diesem Kapitel weist jeder (internen) Fehlermeldung des PoPP-Service eine Fehlermeldung zu, die an der Außenschnittstelle am Interface [I_PoPP-Token_Generation.yaml] oder [I_PoPP_CheckIn_ResourceServer.yaml] sichtbar sind.

A_27049 -PoPP-Service, Mapping von Smartcard Fehlercodes

Der PoPP-Service MUSS interne Fehlermeldungen während des eGK-Handling auf folgende an Außenschnittstellen sichtbare Fehlermeldungen abbilden:

Tabelle 9: Fehlermeldungen eGK-Handling

Interne Fehlermeldung	Fehlermeldung Außenschnittstelle
InvalidAuthentication	ErrorEgkHandling
InvalidCaCvc	ErrorEgkHandling
InvalidCertificatePairContactless	ErrorEgkHandling
InvalidCertificatePairT1	ErrorEgkBlocked
InvalidEndEntityCvc	ErrorEgkHandling
InvalidPiObjectSystem	ErrorEgkHandling
InvalidPtvObjectSystem	ErrorEgkHandling
InvalidX509	ErrorEgkHandling
UnexpectedStatusWordSceAuthG2	ErrorEgkHandling
UnexpectedStatusWordSceAuthG3	ErrorEgkHandling
UnexpectedStatusWordSceOpenEgk	ErrorEgkHandling
UnexpectedStatusWordSceReadCvc	ErrorEgkHandling
UnexpectedStatusWordSceReadX509	ErrorEgkHandling
UnexpectedStatusWordSceTC1	ErrorEgkHandling
UnknownCertificates	WarningUnknownCertificates

[<=]

6.1.1.9 eGK-Hash-Datenbank

Übersicht zu den weiteren Unterabschnitten:

1. "Einleitung und Mengengerüst": die eGK-Hash-Datenbank wird auf hoher Abstraktionsebene beschrieben
2. "Use Cases im laufenden Betrieb": Analyse der Fälle, die im laufenden Betrieb auftreten, das ist die Grundlage für die Spezifikation der "check(. . .)" Funktion in A_27622*

3. "Definition von Begriffen zur Wahrscheinlichkeit": Definition von Begriffen, die in Folgeabschnitten verwendet werden
4. "Use Cases zur Befüllung durch Kostenträger": Analyse der Fälle, die beim Befüllen durch Kostenträger (oder deren Dienstleister) auftreten, das ist die Grundlage für die Spezifikation der "import(. . .)" Methode in A_27623
5. "Weitere Anforderungen an die eGK-Hash-Datenbank": unter anderem Spezifikation der "check(. . .)" Funktion und der "import(. . .)" Methode, basierend auf der Analyse in vorherigen Abschnitten

6.1.1.9.1 Einleitung und Mengengerüst

Die eGK-Hash-Datenbank im PoPP-Service beantwortet die Frage: "Stammt ein vorgelegtes CV-Zertifikat und ein vorgelegtes X.509 AUT-Zertifikat aus ein und derselben eGK?"

So eine eGK-Hash-Datenbank wird im PoPP-Service für eGK der Generation 2.x benötigt. Die einfachste Art der technischen Umsetzung wäre eine (mathematische) Funktion, die jedem CV-Zertifikat genau ein AUT-Zertifikat zuordnet. Softwaretechnisch wäre das eine Tabelle (in Java ein Map<CVC, AUT>). In dem Fall enthielte die Tabelle personenbezogene Daten, was weder datensparsam noch datenschutzfreundlich wäre. Stattdessen verwendet die eGK-Hash-Datenbank eine MengeegkEntries, die statt der Zertifikate unter anderem Hashwerte der Zertifikate enthält. Wenn der eGK-Hash-Datenbank dann ein Paar aus CV-Zertifikat und AUT-Zertifikat präsentiert wird, beantwortet die eGK-Hash-Datenbank letztendlich die Frage: "Ist a) dem Hashwert des CV-Zertifikats der Hashwert des AUT-Zertifikats zugeordnet und b) wenn das CV-Zertifikat unbekannt ist, ist dem Hashwert des AUT-Zertifikats nicht bereits der Hashwert eines anderen CV-Zertifikats zugeordnet?"

Überlegungen zum Mengengerüst: Es gibt (Stand November 2025) fast 75 Millionen gesetzlich Versicherte mit eGK G2.x. Es ist davon auszugehen, dass die Anzahl "nicht abgelaufener eGK" darüber liegt, weil es Versicherte gibt, die über mehr als eine "nicht abgelaufene eGK" verfügen, beispielsweise Ersatz für defekte eGK oder infolge eines Kassenwechsels. Hier wird geschätzt, dass die eGK-Hash-Datenbank so zu dimensionieren ist, dass die Mächtigkeit der Menge egkEntries bis zu 150.000.000 (150 Millionen) reicht. Zweimal 150 Millionen SHA-256 Werte beanspruchen netto 9.400 Millionen Byte, also 9,4 Gigabyte. Das ist eine Größenordnung, die für eine moderne Infrastruktur keine besonderen Ansprüche stellt.

Überlegungen zur Befüllung der MengeegkEntries: Ein neues Element wird der MengeegkEntries hinzugefügt, wenn ein KTR (oder dessen Dienstleister) das neue Element dem PoPP-Service mitteilt, oder wenn eine eGK G2.x kontaktbehaftet vom PoPP-Service angesprochen wird ("trust on first (contact based) use", siehe A_27013-*). Bei der Anlieferung von Elementen für die MengeegkEntries verwendet der KTR (oder dessen Dienstleister) eine mTLS Verbindung und übermittelt die Elemente als signierte Nachricht.

Überlegungen zum Entfernen von Elementen aus egkEntries: Es ist vorgesehen, dass KTR (oder deren Dienstleister) in der Lage sind das Entfernen eines Elementes aus egkEntries zu veranlassen (remove). Zusätzlich ist vorgesehen, dass jedem Element ein Verfallsdatum zugeordnet ist, welches sich aus dem Element "notAfter" aus dem AUT-Zertifikat ergibt. Die eGK-Hash-Datenbank ist damit in der Lage "abgelaufene" Elemente aus der Menge egkEntries zu entfernen. Die eGK-Hash-Datenbank wird nicht dazu verpflichtet "abgelaufene" Elemente aus der Menge egkEntries zu entfernen. Falls die eGK-Hash-Datenbank in der Lage ist auch mit einer sehr großen Anzahl an Elementen in egkEntries performant umzugehen, dann ist ein Entfernen "abgelaufener" Werte auch nicht erforderlich. Die eGK-Hash-Datenbank beantwortet ja nur die Frage, ob ein vorgelegtes Paar aus einer eGK stammt, nicht aber, ob das AUT-Zertifikat des Paares noch gültig ist. Die Frage "ist ein AUT-Zertifikat noch gültig?" wird nicht von der eGK-Hash-

Datenbank, sondern an anderer Stelle beantwortet (siehe Prüfung des AUT-Zertifikats in A_27130-*).

Definition „Lieferant“: Mit „Lieferant“ ist in diesem Unterkapitel eine Instanz gemeint, die berechtigt ist dem PoPP-Service neue Elemente für die eGK-Hash-Datenbank zu übermitteln. Es ist möglich, dass die Rolle des „Lieferanten“ von einem Kostenträger selbst wahrgenommen wird. Vermutlich wird es Kostenträger geben, welche das Einliefern von neuen Elementen an ihre Dienstleister delegieren. „Lieferant“ ist im Folgenden damit eine verkürzte Form von „Kostenträger (oder deren Dienstleister)“. Es hat sich gezeigt, dass die TSP eGK die Rolle Lieferant einnehmen werden.

6.1.1.9.2 Use Cases im laufenden Betrieb

Dieser Abschnitt beschreibt die möglichen Konstellationen, die im PoPP-Service bei der Überprüfung von CV-Zertifikaten und AUT-Zertifikaten mittels der eGK-Hash-Datenbank auftreten, also im Rahmen der "check(. . .)" Funktion nach A_27622*. Es gilt folgende Nomenklatur:

1. Für das genutzte Übertragungsprotokoll zur eGK sind zwei Werte möglich:
 - a. "T=1": kontaktbehaftete Übertragung mit dem Protokoll T=1 aus ISO/IEC 7816-3.
 - b. Nicht "T=1", also "T=CL": kontaktlose Übertragung mit einem Protokoll aus der ISO/IEC 14443 Serie.
2. Für das CV-Zertifikat, welches dem PoPP-Service übermittelt wird, sind zwei Werte möglich:
 - a. "CVC bekannt": Das übermittelte CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
 - b. Nicht "CVC bekannt" also "CVC unbekannt": Das übermittelte CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
3. Für das AUT-Zertifikat, welches dem PoPP-Service übermittelt wird, sind zwei Werte möglich:
 - a. "AUT bekannt": Das übermittelte AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
 - b. Nicht "AUT bekannt" also "AUT unbekannt": Das übermittelte AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
4. Für das vorgelegte Paar aus CV-Zertifikat und AUT-Zertifikat sind zwei Werte möglich:
 - a. "match": Die eGK-Hash-Datenbank bestätigt, dass das vorgelegte CV-Zertifikat zum vorgelegten AUT-Zertifikat gehört.
 - b. Nicht "match" also "mismatch": Die eGK-Hash-Datenbank ist nicht in der Lage zu bestätigen, dass das vorgelegte CV-Zertifikat zum vorgelegten AUT-Zertifikat gehört. Folgende Ursachen sind möglich:
 - i. Felder 3 und 4: Weder das CV-Zertifikat noch das AUT-Zertifikat sind der eGK-Hashdatenbank bekannt.
 - ii. Felder 7 und 8: Nur das CV-Zertifikat ist der eGK-Hash-Datenbank bekannt.
 - iii. Felder 11 und 12: Sowohl das CV-Zertifikat als auch das AUT-Zertifikat sind der eGK-Hash-Datenbank bekannt, aber sie gehören nicht zusammen.
 - iv. Felder 15 und 16: Nur das AUT-Zertifikat ist der eGK-Hash-Datenbank bekannt.
5. Des Weiteren speichert die eGK-Hash-Datenbank zu jedem Element einen Zustand, der einen der folgenden Werte annimmt:

- "imported": Das Element wurde mittels "import(. . .)" Methode in die eGK-Hash-Datenbank aufgenommen.
- "ad hoc": Das Element wurde mittels "check(...)" Methode in die eGK-Hash-Datenbank aufgenommen.
- "blocked": Der PoPP-Service stellt keine PoPP-Token aus, wenn ein vorgelegtes CV-Zertifikat oder ein vorgelegtes AUT-Zertifikat in einem Element enthalten sind, dessen Zustand "blocked" ist.

Aus der obigen Liste folgt, dass fünf Variablen zu berücksichtigen sind und die fünfte (Zustand des Elementes) keine boolesche Variable ist. Leider sind Karnaugh-Veitch-Diagramme mit mehr als vier Variablen unübersichtlich und Karnaugh-Veitch-Diagramme berücksichtigen lediglich boolesche Variablen. Zwecks Komplexitätsreduktion wird das System deshalb wie folgt vereinfacht:

- Für alle Felder "CVC unbekannt" und "AUT unbekannt" liegt kein Element vor. Deshalb liegt auch kein Zustand vor. Daraus folgt, dass der Zustand "blocked" für solche Felder keine Rolle spielt.
- Für alle übrigen Felder in denen der Zustand des Elementes "blocked" ist generiert der PoPP-Service eine Fehlermeldung.
- Fazit: Der Zustand "blocked" eines Elementes ist für die weitere Betrachtung im Karnaugh-Veitch-Diagramm irrelevant. Deshalb sind hier lediglich binäre (boolesche) Variablen relevant. Zudem ist der Zustand des Elements nur für wenige Felder relevant, die dann waagerecht unterteilt werden.

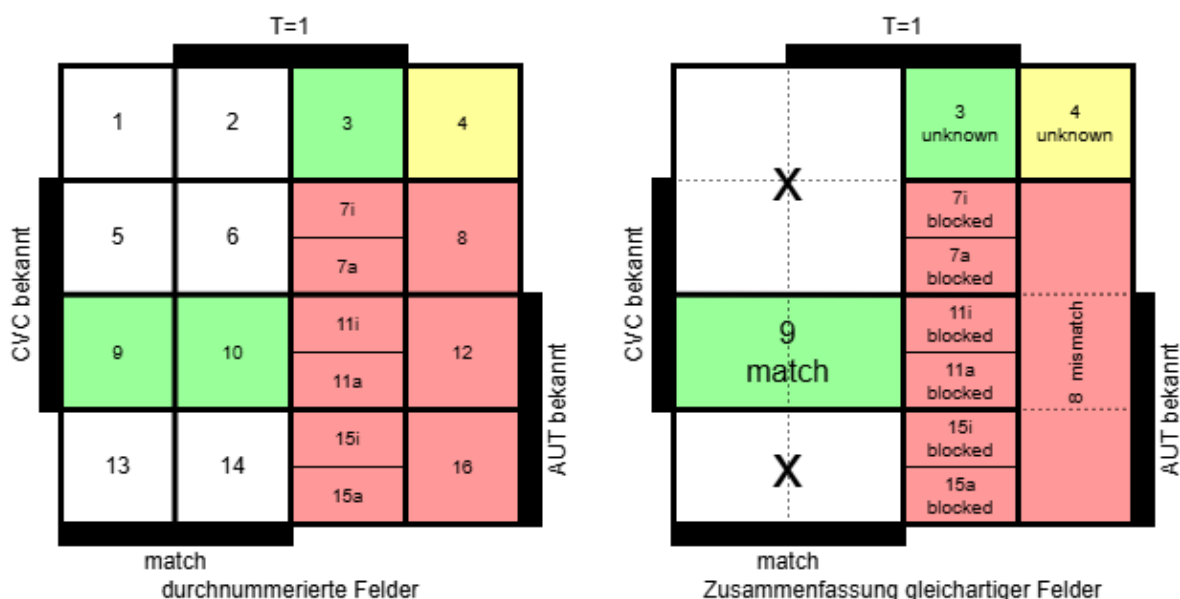


Abbildung 8: Karnaugh-Veitch Diagramm zur "check"-Funktion

Einträge in der eGK-Hash-Datenbank werden wie folgt dargestellt:

- cvcX** bezeichnet den Hashwert eines bestimmten CV-Zertifikates "X".
- autY** bezeichnet den Hashwert eines bestimmten AUT-Zertifikates "Y".
- {cvcX, autY, imported}** bezeichnet einen Eintrag in der eGK-Hash-Datenbank, der durch den Import gemäß A_27623-* entstand.
- {cvcX, autY, adHoc}** bezeichnet einen Eintrag in der eGK-Hash-Datenbank, der durch "trust on first (contact based) use" gemäß A_27622-* entstand.

5. **{cvcX, autY, blocked}** bezeichnet einen Eintrag in der eGK-Hash-Datenbank, der nicht zur Erzeugung eines PoPP-Token nutzbar ist.

Für das zugehörige Karnaugh-Veitch-Diagramm gilt:

1. **Felder 1, 2, 5, 6, 13, 14:** Es gibt sechs mit "x" gekennzeichnete "don't care" Felder, weil es unmöglich ist, dass "match" zutrifft, wenn "CVC unbekannt" oder "AUT unbekannt" ist.
2. **Feld 3:** Bei kontaktbehafteter Kommunikation ist "CVC unbekannt" und "AUT unbekannt". Gemäß "trust on first (contact based) use" wird hier "ad hoc" ein neuer Eintrag zur eGK-Hash-Datenbank hinzugefügt: {cvcX, autY, adHoc}.
3. **Feld 4:** Bei kontaktloser Kommunikation ist "CVC unbekannt" und "AUT unbekannt". Der PoPP-Service ist nicht in der Lage sicher zu entscheiden, ob CV-Zertifikat und AUT-Zertifikat zusammengehören. Statt eines PoPP-Token wird eine Fehlermeldung generiert.
4. **Feld 7i:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit "AUT unbekannt" vor und der Eintrag stammt aus einem Import. Aus "mismatch" folgt, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Da der Eintrag zum CV-Zertifikat "importiert" wurde ist sicher auszuschließen, dass in der Vergangenheit missbräuchlich PoPP-Token ausgestellt wurden. Wegen dieses Sicherheitsvorfalls wird sowohl das CV-Zertifikat, als auch die AUT-Zertifikate blockiert:
 - a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:
 - i. {cvcX, autX, imported}
 - ii. "kein Eintrag für autY"
 - c. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcX, autY, blocked}
5. **Feld 7a:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit "AUT unbekannt" vor und der Datenbankeintrag stammt aus einem "ad hoc" Vorgang. Vorgelegt werde das Paar (cvcX, autY). Der Datenbankeintrag laute vorher: {cvcX, autZ, adHoc}. Der PoPP-Service ist in diesem Fall nicht in der Lage zu entscheiden, ob cvcX zu autY oder zu autZ oder zu einem anderen AUT-Zertifikat gehört. Es ist möglich, dass in der Vergangenheit missbräuchlich PoPP-Token ausgestellt wurden. Es ist sicher, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Wegen dieses Sicherheitsvorfalls werden das CV-Zertifikat und die AUT-Zertifikate blockiert.
 - a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:
 - i. "kein Eintrag zu autY"
 - ii. {cvcX, autZ, adHoc}
 - c. Zustand hinterher:
 - i. {cvcX, autY, blocked}
 - ii. {cvcX, autZ, blocked}
6. **Felder 8, 12, 16:** Bei kontaktloser Kommunikation gibt es ein "mismatch". Neben dem unter Feld 7 beschriebenen Vorfall ist es hier auch möglich, dass ein Angreifer

zwei verschiedene eGK dem PoPP-Service präsentiert: Eine eGK für Authentisierung mit CV-Zertifikat und eine andere aus der das AUT-Zertifikat ausgelesen wird. Statt eines PoPP-Token wird eine Fehlermeldung generiert. Der Inhalt der eGK-Hash-Datenbank wird nicht verändert.

7. **Felder 9, 10:** Bei "CVC bekannt" und "AUT bekannt" liegt ein "match" vor. Das ist der Gutfall, der zur Ausstellung eines PoPP-Token führt (egal ob der Zustand des CV-Zertifikates "imported" oder "ad Hoc" ist).
8. **Feld 11i:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit "AUT bekannt" vor und der Eintrag stammt aus einem Import. Aus "mismatch" folgt, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Da der Eintrag zum CV-Zertifikat "importiert" wurde, ist sicher auszuschließen, dass in der Vergangenheit missbräuchlich PoPP-Token ausgestellt wurden. Wegen dieses Sicherheitsvorfalls werden sowohl die CV-Zertifikate als auch die AUT-Zertifikate blockiert:
 - a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:
 - i. {cvcX, autX, imported}
 - ii. {cvcY, autY, *=egal}
 - c. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcY, autY, blocked}
9. **Feld 11a:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit „AUT bekannt“ vor und der Datenbankeintrag stammt aus einem "ad hoc" Vorgang. Vorgelegt werde das Paar (cvcX, autY). Der Datenbankeintrag laute vorher: {cvcX, autZ, adHoc}. Der PoPP-Service ist in diesem Fall nicht in der Lage zu entscheiden, ob cvcX zu autY oder zu autZ oder zu einem anderen AUT-Zertifikat gehört. Es ist möglich, dass in der Vergangenheit missbräuchlich PoPP-Token ausgestellt wurden. Es ist sicher, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Wegen dieses Sicherheitsvorfalls werden sowohl die CV-Zertifikate als auch die AUT-Zertifikate blockiert:
 - a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:
 - i. {cvcX, autZ, adHoc}
 - ii. {cvcY, autY, *=egal}
 - c. Zustand hinterher:
 - i. {cvcX, autZ, blocked}
 - ii. {cvcY, autY, blocked}
10. **Feld 15i:** Bei kontaktbehafteter Kommunikation und "CVC unbekannt" gibt es ein "mismatch" und der Datenbankeintrag zum vorgelegten AUT-Zertifikat wurde importiert. Da der Eintrag zum AUT-Zertifikat "importiert" wurde ist sicher auszuschließen, dass in der Vergangenheit missbräuchlich PoPP-Token ausgestellt wurden. Wegen dieses Sicherheitsvorfalls wird sowohl das vorgelegte CV-Zertifikat als auch das vorgelegte AUT-Zertifikat blockiert:
 - a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:

- i. "kein Eintrag zu cvcX"
 - ii. {cvcY, autY, imported}
 - c. Zustand nachher:
 - i. {cvcX, autY, blocked}
 - ii. {cvcY, autY, blocked}
11. **Feld 15a:** Bei kontaktbehafteter Kommunikation und "CVC unbekannt" gibt es ein "mismatch" und der Datenbankeintrag zum vorgelegten AUT-Zertifikat stammt aus einem "ad hoc" Vorgang. Vorgelegt werde das Paar (cvcX, autY). Der Datenbankeintrag laute vorher: {cvcZ, autY, adHoc}. Der PoPP-Service ist in diesem Fall nicht in der Lage zu entscheiden, ob autY zu cvcX oder cvcZ oder zu einem anderen CV-Zertifikat gehört. Es ist möglich, dass in der Vergangenheit missbräuchlich PoPP-Token ausgestellt wurden. Es ist sicher, dass der private Schlüssel zu wenigstens einem der beteiligten CV-Zertifikate nicht wie vorgesehen verwendet wird. Wegen dieses Sicherheitsvorfalls werden beide Einträge blockiert.
- a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:
 - i. "kein Eintrag für cvcX".
 - ii. {cvcZ, autY, adHoc}.
 - c. Zustand hinterher:
 - i. {cvcX, autY, blocked}
 - ii. {cvcZ, autY, blocked}

6.1.1.9.3 Definition von Begriffen zur Wahrscheinlichkeit

Hier werden einige Begriffe zu Wahrscheinlichkeiten definiert, die in Folgekapiteln verwendet werden. Die Begriffe sind nach Wahrscheinlichkeiten von "sicher" bis "unmöglich" sortiert.

1. **sicher:** Ein Ereignis tritt mit einer Wahrscheinlichkeit von eins ein.
Beispiel: Eine Urne enthält nur rote Kugeln. Die Wahrscheinlichkeit aus dieser Urne eine rote Kugel zu ziehen ist eins.
2. **extrem wahrscheinlich:** Die Eintrittswahrscheinlichkeit ist fast eins. Theoretisch ist es möglich, dass das Ereignis nicht eintritt, aber in der Praxis muss der Nichteintritt nicht betrachtet werden.
Beispiel: Zwei Zufallszahlen der Länge 128 bit, die unabhängig voneinander generiert werden, sind verschieden.
3. **sehr wahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so hoch, dass in der Praxis das Gegenereignis nicht beobachtet wird.
Beispiel: Zu einer Nachricht *M* wird der SHA-256 Hashwert zweimal berechnet. Beide Ergebnisse stimmen überein.
Hinweis: In [<https://www.cs.toronto.edu/~bianca/papers/sigmetrics09.pdf>] werden RAM-Lesefehler untersucht. Bei Verwendung von ECC-RAM ist es sehr wahrscheinlich, dass RAM-Lesefehler entdeckt werden. Deshalb ist es sehr wahrscheinlich, dass eine zweimalige Hashwertberechnung dasselbe Ergebnis liefert.
4. **wahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so hoch, dass das Ereignis regelmäßig beobachtet wird.
5. **unwahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so niedrig, dass das Ereignis nur selten beobachtet wird. Obwohl das Ereignis selten eintritt, müssen Systeme mit

so einem Ereignis kontrolliert umgehen können, was durch Tests zu bestätigen ist.
Beispiel: Bitfehler im RAM eines Servers, beispielhafte Gegenmaßnahme ECC-RAM.

6. **sehr unwahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so niedrig, dass ein System gegen so ein Ereignis nicht gehärtet werden muss.
Beispiel: Zu einer Nachricht M wird der SHA-256 Hashwert zweimal berechnet. Beide Ergebnisse sind verschieden.
7. **extrem unwahrscheinlich:** Die Eintrittswahrscheinlichkeit ist fast null. Theoretisch ist es möglich, dass das Ereignis eintritt, aber in der Praxis muss der Eintritt nicht betrachtet werden.
Beispiel: Zwei Zufallszahlen der Länge 128 bit, die unabhängig voneinander gewürfelt werden, sind gleich.
8. **unmöglich:** Ein Ereignis tritt mit einer Wahrscheinlichkeit von null ein.
Beispiel: Eine Urne enthält nur rote Kugeln. Die Wahrscheinlichkeit aus dieser Urne eine grüne Kugel zu ziehen ist null.

6.1.1.9.4 Use Cases zur Befüllung durch Kostenträger

Dieses Kapitel betrachtet die Befüllung der eGK-Hash-Datenbank durch Lieferanten.

Annahmen:

1. Es ist "extrem unwahrscheinlich", dass zwei verschiedene Zertifikate denselben Hashwert haben. Deshalb werden derartige Fälle hier nicht weiter betrachtet.
2. Es ist "unwahrscheinlich" (aber denkbar), dass eGKs erst im Feld eingesetzt werden und für diese ein Eintrag in die eGK-Hash-Datenbank erfolgt (per "trust on first (contact based) use") und zu einem späteren Zeitpunkt liefert ein Lieferant für dieselbe eGK per Import einen Eintrag für die eGK-Hash-Datenbank. Daraus folgt, dass es zwar "unwahrscheinlich" aber möglich ist, dass zum Zeitpunkt des Imports bereits ein Eintrag in der eGK-Hash-Datenbank vorliegt. Dieses Kapitel betrachtet dann die dabei auftretenden Fälle und wie mit ihnen umzugehen ist.

Hinweis: Falls die Wahrscheinlichkeit des Ereignisses "eine eGK wird im Feld benutzt bevor ein Import ihrer Daten in die eGK-Hash-Datenbank" von "unwahrscheinlich" auf "sehr unwahrscheinlich" oder niedriger eingestuft wird, dann wird dieses Kapitel gegenstandslos. Derzeit ist die Annahme, dass dieses Ereignis lediglich "unwahrscheinlich" ist. Daraus folgt, dass dieses Kapitel relevant ist.

Ein neuer Eintrag {cvcX, autX, imported} wird der eGK-Hash-Datenbank neu hinzugefügt oder entfernt (remove).

1. Für das CV-Zertifikat, welches im Eintrag enthalten ist, sind zwei Werte möglich:
 - a. "CVC bekannt": Das CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
 - b. Nicht "CVC bekannt" also "CVC unbekannt": Das CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
2. Für das AUT-Zertifikat, welches im Eintrag enthalten ist, sind zwei Werte möglich:
 - a. "AUT bekannt": Das AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
 - b. Nicht "AUT bekannt" also "AUT unbekannt": Das AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
3. Für Einträge, die das CV-Zertifikat oder das AUT-Zertifikat (oder beide) enthalten, sind zwei Werte relevant:

- a. Wenigstens einer dieser Einträge ist im Zustand "blocked".
 - b. Keiner dieser Einträge ist "blocked" (also "unblocked").
4. Für den Vergleich zwischen dem Eintrag und einem bestehenden Eintrag in der eGK-Hash-Datenbank sind zwei Werte denkbar:
- a. "match": Die Hashwerte im Eintrag sind identisch zu den Hashwerten in einem bestehenden Eintrag.
 - b. Nicht "match" (also "mismatch"): Die Hashwerte des Eintrags sind nicht identisch zu irgendeinem bestehenden Eintrag.

Aus der obigen Liste folgt, dass vier binäre (boolesche) Variablen zu berücksichtigen sind. Für das zugehörige Karnaugh-Veitch Diagramm gilt:



Abbildung 9: Karnaugh-Veitch Diagramm zur "import"-Methode

1. **Felder 1, 4:** Der Fall, dass "CVC unbekannt" und "AUT unbekannt" aber "blocked" ist, kann nicht vorkommen. Deshalb sind diese Felder irrelevant (don't care).
2. **Felder 1, 2, 5, 6, 13, 14:** Der Fall, dass ein "match" vorliegt, wenn cvcX oder autX "unbekannt" sind, kann nicht vorkommen. Deshalb sind diese Felder irrelevant (don't care).
3. **Feld 3:** Weder cvcX, noch autX sind in der eGK-Hash-Datenbank enthalten.
 - a. "import": Ein neuer Eintrag wird der eGK-Hash-Datenbank hinzugefügt.
 - b. "remove": Der zu entfernende Eintrag ist nicht in der eGK-Hash-Datenbank enthalten. Die eGK-Hash-Datenbank wird nicht verändert.
4. **Felder 7, 8:** cvcX ist "bekannt" aber autX ist "unbekannt". Daraus folgt, dass der private Schlüssel des CV-Zertifikates nicht wie vorgesehen verwendet wird. In der Vergangenheit wurden missbräuchlich PoPP-Token ausgestellt. Der Eintrag wird gesperrt.
 - a. Zustand in der eGK-Hash-Datenbank vorher:
 - i. "kein Eintrag zu autX"
 - ii. {cvcX, autY, *=egal}

- b. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcX, autY, blocked}
- 5. **Feld 9:** cvcX ist "bekannt" aber "blocked" und es liegt ein "match" vor. Die eGK-Hash-Datenbank wird nicht verändert.
- 6. **Feld 10:** Sowohl cvcX, als auch autX sind "bekannt" und auch in der eGK-Hash-Datenbank sind cvcX und autX einander zugeordnet ("match") und der Zustand von cvcX in der eGK-Hash-Datenbank ist nicht "blocked".
 - a. "import": Ein neuer Eintrag wird der eGK-Hash-Datenbank hinzugefügt. Dann lassen sich folgende Fälle unterscheiden: Der Zustand von cvcX in der eGK-Hashdatenbank ist
 - i. "imported": Keine Aktion erforderlich, da die eGK-Hash-Datenbank den neuen Eintrag bereits enthält.
 - ii. "adHoc": Der Zustand des cvcX in der eGK-Hashdatenbank wird von "adHoc" auf "imported" geändert.
 - A. Zustand vorher: {cvcX, autX, adHoc}
 - B. Zustand nachher: {cvcX, autX, imported}
 - b. "remove": Der Eintrag wird aus der eGK-Hash-Datenbank entfernt.
- 7. **Felder 11, 12:** Sowohl cvcX, als auch autX sind "bekannt", aber in der eGK-Hash-Datenbank einander NICHT zugeordnet ("mismatch"). In der Vergangenheit wurden PoPP-Token missbräuchlich ausgestellt. Einträge werden gesperrt.
 - a. Zustand vorher:
 - i. {cvcX, autY, *=egal}
 - ii. {cvcZ, autX, *=egal}
 - b. Zustand nachher:
 - i. {cvcX, autY, blocked}
 - ii. {cvcZ, autX, blocked}
- 8. **Felder 15, 16:** cvcX ist "unbekannt", autX ist "bekannt". Daraus folgt, dass der private Schlüssel des CV-Zertifikates nicht wie vorgesehen verwendet wird. In der Vergangenheit wurden missbräuchlich PoPP-Token ausgestellt. Der vorhandene Eintrag wird gesperrt und ein neuer Eintrag angelegt.
 - a. Zustand vorher:
 - i. "kein Eintrag für cvcX"
 - ii. {cvcY, autX, *=egal}
 - b. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcY, autX, blocked}

A_27044 -PoPP-Service, Schnittstelle I_PoPP_EHC_CertHash_Import

Der PoPP-Service MUSS über eine Schnittstelle I_PoPP_EHC_CertHash_Import mit folgenden Eigenschaften verfügen:

- 1. Die Schnittstelle I_PoPP_EHC_CertHash_Import ist aus dem Internet erreichbar.

2. Die IP-Adresse über welche die Schnittstelle I_PoPP_EHC_CertHash_Import erreichbar ist wird bekanntgegeben.
3. Änderungen an der IP-Adresse für die Schnittstelle I_PoPP_EHC_CertHash_Import werden mindestens 30 Tage vor der Änderung allen Inhabern der Identitäten aus der Liste listImportClients bekanntgegeben.
4. Die Schnittstelle I_PoPP_EHC_CertHash_Import ist erst nach einem TLS-Handshake nutzbar. Der TLS-Handshake lässt ausschließlich Ciphersuiten gemäß [gemSpec_Krypt] zu. Der TLS-Handshake scheitert, wenn die Identität des Client nicht positiv entsprechend A_28547* geprüft werden konnte.
5. Über die Schnittstelle I_PoPP_EHC_CertHash_Import wird (nach dem TLS-Handshake auf Applikationsebene) eine oder mehrere signierte Nachrichten mit Einträgen für die eGK-Hash-Datenbank übertragen (siehe A_27046-* signedMessage).
6. Ein über die Schnittstelle importierter Eintrag ist spätestens nach 36 Stunden in der Menge egkEntries der eGK-Hash-Datenbank enthalten, sofern er erfolgreich geprüft wurde.

[<=]

A_27045 -PoPP-Service, Eintrag Lieferant

Ein Lieferant MUSS die Schnittstelle I_PoPP_EHC_CertHash_Import wie folgt bedienen:

1. Für neu produzierte eGK wird deren Eintrag spätestens 36 Stunden vor Auslieferung der eGK an den PoPP-Service übertragen.
2. Nach einem TLS-Handshake werden bis zu einem "close_notify"-Alert nicht mehr als 20.000.000 Einträge übertragen.
3. Die zu signierende Nachricht eContent besitzt folgende ASN.1 Struktur:

```
eContent ::= SEQUENCE {  
    version INTEGER  
    egkInfos SEQUENCE OF egkInfo (SIZE(1..20000000))  
}  
egkInfo ::= SET {  
    status INTEGER, -- 0=import, 1=remove  
    hashAut BIT STRING,  
    hashCvc OCTET STRING,  
    notAfter UTCTime -- attribute "notAfter" from X.509  
}
```
4. Die zu signierende Nachricht eContent wird DER codiert.
5. Als Versionsnummer wird der Wert 0 verwendet.
6. Die zu signierende Nachricht eContent wird mit einem gemäß [gemSpec_Krypt] zulässigen Verfahren wie in [RFC 5652#5] beschrieben signiert, wobei das Zertifikat des signierenden in die signierte Nachricht einzustellen ist.
7. Im Anschluss an den Import und als Antwort auf die signierte Nachricht eContent wird dem Importeur in derselben TLS-Session folgende Information zurückgemeldet:
 - a. Liste mit fehlerhaften Einträgen,
 - b. Liste mit ignorierten Einträgen,
 - c. Liste geblockter Einträge (weil der Eintrag bereits vorhanden war und geblockt war oder geblockt wurde).

[<=]

Hinweis: Die Beschränkung der Anzahl von Einträgen in A_27045, Punkt 5 auf 20 Millionen sorgt dafür, dass die signierte Nachricht kleiner als 2 GiByte = 2.147.483.647*

Byte bleibt. Größere Nachrichten lassen sich mit Standardbibliotheken unter gängigen Programmiersprachen (etwa Java) nicht verarbeiten. Falls ein Lieferant mehr Einträge anliefern möchte, dann verwendet er mehrere signierte Nachrichten.

A_28547 -PoPP-Service, eGK-Hash-Datenbank, Client-Zertifikatsprüfung im TLS-Handshake

Der PoPP-Service MUSS beim TLS-Verbindungsaufbau eines Lieferanten (TSP eGK) zur Befüllung der eGK-Hash-Datenbank eine Client-Authentisierung durchführen und dabei prüfen, dass der Client ein C.FD.TLS-C-Zertifikat der TI-Komponenten PKI-Prüfung gegen die TSL, vgl. A_27150* - mit Rolle oid_tsp_egk vorweist und dieses Zertifikat zeitlich gültig und nicht gesperrt ist. [**<=**]

6.1.1.9.5 Weitere Anforderungen an die eGK-Hash-Datenbank

A_27046 -PoPP-Service, eGK-Hash-Datenbank

Der PoPP-Service MUSS über eine eGK-Hash-Datenbank mit folgenden Eigenschaften verfügen:

1. Die eGK-Hash-Datenbank besitzt eine Menge egkEntries.
2. Die eGK-Hash-Datenbank ist in der Lage in der Menge egkEntries mindestens 150.000.000 Einträge zu speichern.
3. Die eGK-Hash-Datenbank besitzt eine Funktion mit der Signatur "String check(byte[] cvc, byte[] aut, String protocol)" gemäß A_27622*.
4. Die eGK-Hash-Datenbank besitzt eine Methode mit der Signatur "void import(byte[] SignedData)" gemäß A_27623*.
5. Wenn über die Funktionen "check(. . .)" oder "import(. . .)" mehr Einträge angeliefert werden, als in der eGK-Hash-Datenbank speicherbar sind, dann werden zusätzliche Einträge ignoriert.

[**<=**]

A_27622 -PoPP-Service, eGK-Hash-Datenbank, check-Funktion

Die eGK-Hash-Datenbank MUSS eine Funktion mit der Signatur "String check(byte[] cvc, byte[] aut, String protocol)" und folgendem Verhalten besitzen:

1. Der Parameter "cvc" ist ein Bytestring, dessen Inhalt identisch ist zum Inhalt der Datei EF.C.eGK.AUT_CVC.E256 einer eGK.
2. Der Parameter "aut" ist ein Bytestring, dessen Inhalt identisch ist zum Inhalt der Datei EF.C.CH.AUT.E256 einer eGK.
3. Der Parameter "protocol" kennzeichnet wie mit der eGK kommuniziert wurde:
 - a. "T=1" für die kontaktbehaftete Kommunikation
 - b. "T=CL" für die kontaktlose Kommunikation
4. Die Funktion check(. . .) berechnet SHA-256 Hashwerte gemäß [FIPS PUB 180-4] und extrahiert das Attribut "notAfter" aus dem AUT-Zertifikat, es gilt:
 - a. hashCvc = SHA-256(cvc)
 - b. hashAut = SHA-256(aut)
 - c. notAfter = Attribut "notAfter" aus dem AUT-Zertifikat.
5. Die Funktion check(. . .) führt folgende Schritte aus:

- a. Schritt 1, **"blocked"**, (alle Felder): Falls egkEntries einen Eintrag für hashCvc enthält und dessen Zustand ist "blocked", oder einen Eintrag für hashAut enthält und dessen Zustand ist „blocked“, dann gibt die Funktion den Wert "blocked" zurück.
- b. Schritt 2, **"match"**, (Felder 9, 10): Falls egkEntries einen Eintrag für hashCvc enthält und dieser Eintrag enthält hashAut, dann gibt die Funktion den Wert "match" zurück.
- c. Schritt 3, der Parameter "protocol" zeigt eine kontaktbehaftet angebundene eGK an. Dann gilt:
 - i. Schritt 3.1, **"unknown"**, (Feld 3): Falls egkEntries weder hashCvc noch hashAut enthält, dann gibt die Funktion "unknown" zurück. Zusätzlich wird ein neuer Eintrag in egkEntries erzeugt:
{hashCvc, hashAut, adHoc}
 - ii. Schritt 3.2, **"blocked"**, (Feld 7): Falls egkEntries den Wert hashCvc enthält aber dessen Eintrag enthält nicht hashAut (sondern autX), dann gibt die Funktion "blocked" zurück.
 - A. Zustand vorher:
 - I. {hashCvc, autX, *=egal}
 - II. "kein Eintrag für hashAut"
 - B. Zustand nachher:
 - I. {hashCvc, autX, blocked}
 - II. {hashCvc, hashAut, blocked}
 - iii. Schritt 3.3, **"blocked"**, (Feld 11): Falls egkEntries die Werte hashCvc und hashAut enthält, aber hashCvc ist nicht hashAut zugeordnet (sondern autX, „mismatch“), dann gibt die Funktion "blocked" zurück.
 - A. Zustand vorher:
 - I. {hashCvc, autX, *=egal}
 - II. {cvcY, hashAut, *=egal}
 - B. Zustand nachher:
 - I. {hashCvc, autX, blocked}
 - II. {cvcY, hashAut, blocked}
 - iv. Schritt 3.4, **"blocked"**, (Feld 15): Falls egkEntries den Wert hashCvc nicht enthält, aber den Wert hashAut enthält (innerhalb eines Eintrags zu cvcY), dann gibt die Funktion "blocked" zurück.
 - A. Zustand vorher:
 - I. "kein Eintrag für hashCvc"
 - II. {cvcY, hashAut, *=egal}
 - B. Zustand nachher:
 - I. {hashCvc, hashAut, blocked}
 - II. {cvcY, hashAut, blocked}
- d. Schritt 4, **"unknown"**, **"mismatch"**, (Felder 4, 8, 12, 16): Der Parameter "protocol" zeigt eine kontaktlos angebundene eGK an. Dann gilt: Falls egkEntries

weder hashCvc noch hashAut enthält, dann gibt die Funktion "unknown" zurück, sonst gibt die Funktion "mismatch" zurück. egkEntries wird nicht verändert.

- e. Logging innerhalb der „check(. . .)“-Funktion: Es wird ein Log-Eintrag mit hashCvc und hashAut erzeugt, falls während der Abarbeitung der „check(. . .)“-Funktion
 - i. der Wert "blocked" zurückgegeben wird.
 - ii. Einträge blockiert werden. In diesem Fall wird auch ein Kurzzeitprotokoll angelegt mit den Informationen ICCSN aus cvc und IK-Nummer aus x509. Anhand der IK-Nummer werden blockierte ICCSN spätestens am nächsten Werktag an den jeweiligen Kostenträger gemeldet. Gemeldete Informationen werden aus dem Kurzzeitprotokoll gelöscht.

[<=]

A_27623 -PoPP-Service, eGK-Hash-Datenbank, import-Funktion

Die eGK-Hash-Datenbank MUSS eine Methode mit der Signatur "void import(byte[] SignedData)" und folgendem Verhalten besitzen:

1. Schritt 1: Die Methode prüft, das Signaturzertifikat in SignedData entsprechend A_28548*. Endet die Prüfung nicht mit positivem Ergebnis, dann bricht die Methode ab, sonst fährt sie mit dem nächsten Schritt fort.
2. Schritt 2: Die Methode prüft die Signatur im Parameter SignedData gemäß [RFC 5652#5]. Falls die Signatur ungültig ist, dann bricht die Methode ab, sonst fährt sie mit dem nächsten Schritt fort.
3. Schritt 3: Die Methode entnimmt dem Parameter SignedData die darin enthaltene NachrichtContent.
4. Schritt 4: Die in der NachrichtContent enthaltenen Informationen beeinflussen die eGK-Hash-Datenbank wie folgt:
 - a. Falls eContent nicht die in A_27045* dargestellte Struktur besitzt, dann bricht die Methode ab.
 - b. Die Elemente der Liste egkInfos werden nacheinander bearbeitet. Falls ein Element egkInfo nicht die in A_27045* dargestellte Struktur besitzt, dann wird es übersprungen und der Zähler counterMalformedEgkInfo wird inkrementiert. Andernfalls werden die darin enthaltenen Informationen notAfter, hashCvc, hashAut und status in der eGK-Hash-Datenbank auf die in Schritt 5 beschriebene Art und Weise verarbeitet.
5. Schritt 5:
 - a. Felder 3, 15, 16: Falls hashCvc in egkEntries "unbekannt" ist und hashAut ist in egkEntries
 - i. ebenfalls "unbekannt" (Feld 3) und der Status ist
 - A. status = 0 = "import", dann wird ein neuer Eintrag erzeugt und der Zähler counterImported wird inkrementiert:
{hashCvc, hashAut, imported}
 - B. status = 1 = "remove", dann wird die eGK-Hash-Datenbank durch dieses Element egkInfo nicht verändert aber der Zähler counterRemoved wird inkrementiert.
 - ii. "bekannt" (Felder 15, 16), dann wird egkEntries wie folgt geändert und der Zähler counterBlocked wird inkrementiert:
 - A. vorher:

- I. "kein Eintrag für hashCvc"
 - II. {cvcY, hashAut, *=egal}
 - B. nachher:
 - I. {hashCvc, hashAut, blocked}
 - II. {cvcY, hashAut, blocked}
 - b. Felder 7, 8: Falls hashCvc in egkEntries "bekannt" ist und hashAut ist "unbekannt", dann werden in egkEntries folgende Änderungen vorgenommen und der Zähler counterBlocked wird inkrementiert:
 - i. vorher:
 - A. "kein Eintrag zu hashAut"
 - B. {hashCvc, autY, *=egal}
 - ii. nachher:
 - A. {hashCvc, hashAut, blocked}
 - B. {hashCvc, autY, blocked}
 - c. Feld 9: Falls hashCvc in egkEntries "bekannt" und "blocked" ist und es liegt ein "match" vor, dann wird dieses ElementegkInfo nicht weiterbearbeitet und der Zähler counterBlocked wird inkrementiert.
 - d. Feld 10: Falls hashCvc in egkEntries "bekannt" ist und es liegt ein "match" vor dann wird counterImported inkrementiert und der Status ist
 - i. status = 0 = import und der Zustand des CV-Zertifikates ist:
 - A. "imported", dann wird egkEntries nicht verändert.
 - B. "adHoc", dann wird nur dessen Zustand wie folgt geändert:
 - I. vorher: {hashCvc, hashAut, adHoc}
 - II. nachher: {hashCvc, hashAut, imported}
 - ii. status = 1 = "remove", dann werden Einträge mit hashCvc oder hashAut aus egkEntries entfernt.
 - e. Felder 11, 12: Falls sowohl hashCvc als auch hashAut in egkEntries "bekannt" sind und es liegt kein "match" vor, dann werden in egkEntries folgende Änderungen vorgenommen und der Zähler counterBlocked inkrementiert:
 - i. vorher:
 - A. {hashCvc, autY, *=egal}
 - B. {cvcZ, hashAut, *=egal}
 - ii. nachher:
 - A. {hashCvc, autY, blocked}
 - B. {cvcZ, hashAut, blocked}
6. Logging innerhalb der „import(. . .)“-Methode: Die „import(. . .)“- Methode loggt folgende Ereignisse:
- a. Es wird ein Log-Eintrag inklusive des Client-Zertifikats erzeugt, falls der TLS-Handshake fehlschlägt. Das ist dann der einzige Log-Eintrag für diesen Aufruf der "import(. . .)"-Methode.

- b. Es wird ein Log-Eintrag inklusive des Signaturzertifikates erzeugt, wenn die Signaturprüfung fehlschlägt. Das ist dann der einzige Log-Eintrag für diesen Aufruf der "import(. . .)"-Methode.
- c. Es wird ein Log-Eintrag mit hashCvc und hashAut erzeugt, falls während der Abarbeitung eines Elementes egkInfo Einträge blockiert werden.
- d. Falls irgendein Zähler aus der Menge {counterBlocked, counterMalformedEgkInfo, counterImported, counterRemoved} größer als Null ist, dann wird ein Log-Eintrag inklusive der folgenden Informationen erzeugt:
 - i. Lieferant
 - ii. Wert des Zählers counterBlocked
 - iii. Wert des Zählers counterMalformedEgkInfo
 - iv. Wert des Zählers counterImported
 - v. Wert des Zählers counterRemoved
- e. je Import (also gesammelt für alle Einträge im Paket) Absender/Client, Menge gesamt, Menge erfolgreich, Menge ignorierte, Menge fehlerhafte Einträge.

[<=]

A_28548 -PoPP-Service, eGK-Hash-Datenbank, Signatur-Zertifikatsprüfung

Der PoPP-Service MUSS bei der Prüfung von Signaturen von eGK-Hashwertpaketen (vgl. SignedData in A_27623-*) das Signaturzertifikat prüfen und dabei sicherstellen, dass dieses ein C.FD.ÖSIG-Zertifikat der TI-Komponenten PKI - Prüfung gegen die TSL, vgl. A_27150* - mit Rolle oid_tsp_egk ist und dieses Zertifikat zeitlich gültig und nicht gesperrt ist. **[<=]**

A_27624 -PoPP-Service, eGK-Hash-Datenbank, Löschen veralteter Einträge

Falls der PoPP-Service Einträge mit einem Ablaufdatum versieht und veraltete Einträge löscht, dann DARF er KEINE Einträge löschen, die als geblockt gekennzeichnet sind. **[<=]**

Hinweis: Es ist nicht erforderlich, dass die eGK-Hash-Datenbank für jeden Eintrag individuell ein Ablaufdatum speichert. Es ist beispielsweise möglich einem Ablaufdatum eine Menge von Einträgen zuzuordnen. Der Speicher der eGK-Hash-Datenbank enthält dieses Ablaufdatum dann nur einmal.

Hinweis: Jedes technische System hat eine Speichergrenze. Das Ignorieren von weiteren Einträgen, die sich nicht mehr speichern lassen, verhindernd undefinierte Zustände durch Speicherüberlauf.

Hinweis: Sowohl CV-Zertifikate, als auch AUT-Zertifikate verwenden aktuell SHA-256 als Hashverfahren. Deshalb ist es aus Sicherheitssicht ausreichend in der eGK-Hash-Datenbank ebenfalls SHA-256 Werte zu speichern. Insgesamt gibt es $2^{256} = 1,16 \times 10^{77}$ verschiedene SHA-256 Hashwerte. Angenommen jedem dieser Hashwerte wird ein Volumen zugeordnet, wie es ein Coronavirus einnimmt, dann käme im Mittel pro Würfel mit einer Kantenlänge von zwei Lichtjahren ein Coronavirus. Das bedeutet, dass die eGK-Hash-Datenbank im Vergleich zu allen möglichen Werten so dünn besetzt ist, dass es für einen Angreifer praktisch unmöglich ist ein Paar aus gültigem CV-Zertifikat und gültigem AUT-Zertifikat zu finden, welches nicht aus ein und derselben eGK stammt, aber trotzdem von der eGK-Hash-Datenbank eine "ja"-Antwort bekommt.

Hinweis: Die Codierung von SignedData wird derzeit mit den Kostenträgern und deren Dienstleistern abgestimmt.

A_27201 -PoPP-Service - eGK-Hash-Datenbank Aspekte für Produktgutachten

Der Hersteller des PoPP-Service MUSS die Umsetzung von:

1. Schritt 1 und Schritt 2 der import-Funktion aus A_27623-* (Signaturprüfung und Abbruch im Fehlerfall),
2. alle Schritte der check-Funktion aus A_27622-* und der import-Funktion aus A_27623-*, die zum Status bzw. zum Zustand "blocked" führen und
3. Schritt 4 aus A_27044-* (Client-Authentisierung im TLS-Handshake und Abbruch im Fehlerfall),

im Rahmend des Produktgutachtens prüfen lassen.【<=】

6.1.1.9.6 Anmerkungen zur Implementierung

Die Intention dieses Unterkapitels ist es, Hilfestellungen bei der Implementierung der eGK-Hash-Datenbank zu liefern. Dieses Unterkapitel enthält keine Anforderungen.

Am Anfang dieses Unterkapitels sei zunächst angenommen, dass der PoPP-Service und mit ihm die eGK-Hash-Datenbank in Betrieb seien. Aus Performancegründen erscheint es ratsam die eGK-Hash-Datenbank im RAM zu halten, weil Speicherzugriffe auf (volatile) RAM-Inhalte vielfach schneller ablaufen als solche auf ein persistentes Dateisystem.

Gemäß A_27622* „check(. . .)“-Funktion Punkt 5.c.i (Feld 3) ist es möglich im laufenden Betrieb beispielsweise durch "trust on first (contact based) use" der eGK-Hash-Datenbank neue Einträge hinzuzufügen. Zudem sind in A_27622* weitere Punkte enthalten, bei denen sich der Inhalt der eGK-Hash-Datenbank ändert. Deshalb erscheint es nicht hinreichend zu sein, den Inhalt der eGK-Hash-Datenbank ausschließlich im RAM vorzuhalten, damit er beispielsweise durch einen Stromausfall erhalten bleibt. Daraus folgt, dass es zur eGK-Hash-Datenbank im (volatilen) RAM auch eine persistente Variante der eGK-Hash-Datenbank gibt.

Gemäß A_27622* und A_27623 spielt die Reihenfolge in welche Einträge zur eGK-Hash-Datenbank hinzugefügt werden eine Rolle.

Szenario 1:

1. Die eGK-Hash-Datenbank enthalte einen Eintrag {cvc1, aut1, imported}.
2. Aus irgendeinem Grund werde cvc1 und damit auch aut1 blockiert.
3. Anschließend werde versucht der "check(...)"-Funktion {cvc1, aut2} zu präsentieren. Ohne die eGK-Hash-Datenbank zu ändern scheitert die "check(...)"-Funktion wegen blockiertem cvc1.
4. Anschließend werde {cvc2, aut2, "T=1"} der "check(...)"-Funktion präsentiert, was zu einem neuen Eintrag {cvc2, aut2, adHoc} führt.
5. Endzustand in Szenario 1:
 - a. {cvc1, aut1, blocked}
 - b. {cvc2, aut2, adHoc}

Szenario 2:

1. Die eGK-Hash-Datenbank enthalte einen Eintrag {cvc1, aut1, imported}.
2. Anschließend werde {cvc2, aut2, "T=1"} der "check(...)"-Funktion präsentiert, was zu einem neuen Eintrag {cvc2, aut2, adHoc} führt.
3. Anschließend werde versucht der "check(...)"-Funktion {cvc1, aut2, "T=1"} zu präsentieren. Das blockiert cvc1, cvc2, aut1 und aut2.
4. Endzustand in Szenario 2:
 - a. {cvc1, aut1, blocked}

- b. {cvc2, aut2, blocked}

Beispielhaftes Konzept für eine Implementierung der eGK-Hash-Datenbank:

1. Nach erfolgreicher Signaturprüfung speichert die "import(...)"-Methode die "SEQUENCE OF" egkInfos zunächst zusammen mit einem Zeitstempel persistent.
2. Falls im Rahmen der "check(...)"-Funktion ein neuer Eintrag angelegt wird (Feld 3), dann wird ebenfalls mit Zeitstempel eine „SEQUENCE OF“ egkInfos mit den Daten aus dem neuen Eintrag persistent gespeichert, mit status=2=adHoc.
3. Falls im Rahmen der "check(...)"-Funktion oder der "import(...)"-Methode ein oder mehr Einträge blockiert werden, dann werden alle dabei blockierten Einträge in einer „SEQUENCE OF“ egkInfos mit Zeitstempel persistent gespeichert, mit status=3=blocked.

Daraus ergibt sich, dass die eGK-Hash-Datenbank zweimal vorliegt: Volatil im RAM und als durch Zeitstempel geordnete Liste von „SEQUENCE OF“ egkInfos, die persistent gespeichert sind. Der Zustand im RAM lässt sich dabei jederzeit aus den persistenten Informationen eindeutig rekonstruieren, wenn die zeitlich geordneten „SEQUENCE OF“ egkInfos nacheinander verarbeitet werden.

Der Neuaufbau des RAM-Zustandes der eGK-Hash-Datenbank ist möglicherweise (je nach Implementierung) zeitintensiv. Dann bietet es sich an, von der RAM-Version der eGK-Hash-Datenbank einen persistenten Speicherabzug zu erstellen, der sich performanter ins RAM laden lässt.

6.1.2 Schnittstelle für Token Abrufe

Die technische Spezifikation der Schnittstelle `PoPP-Token-Generation` zum Abruf von PoPP-Token durch Clientsysteme veröffentlicht die gematik auf GitHub im OpenAPI-Format.

Zusätzlich gelten die folgenden Anforderungen.

A_26345 -PoPP-Service - WebSocket Interface zur PoPP-Token-Erstellung EHC

Der PoPP-Service MUSS sicherstellen, dass die Schnittstellen `PoPP-Token-Generation` zum Abruf von PoPP-Token bei Nutzung der eGK in einer LEI als WebSocket-Interface mit den Endpunkten gemäß der Schnittstellen-Spezifikation in `[I_PoPP-Token-Generation.yaml]` umgesetzt ist. [\leq]

A_26361 -PoPP-Service - Zero Trust Schutz des PoPP-Interfaces

Der PoPP-Service MUSS sicherstellen, dass der Zugang zu den Schnittstellen `PoPP-Token-Generation` zum Abruf von PoPP-Token mittels des ZETA Guard `[gemSpec_ZETA]` vor unberechtigten Zugriffen geschützt ist. [\leq]

A_26362 -PoPP-Service - Status Code im WebSocket-Interface

Der PoPP-Service MUSS sicherstellen, dass in der Beantwortung eingehender Request an den Schnittstellen `PoPP-Token-Generation` zum Abruf von PoPP-Token ausschließlich die http-Status Code gemäß der OpenAPI-Spezifikation auf `[I_PoPP-Token-Generation.yaml]` verwendet werden. [\leq]

6.2 PoPP-Client

Die Beschreibung des PoPP-Client erfolgt in `[gemILF_PoPP_Client]`.

6.3 Fehlerbehandlung

Error Code (HTTP Status Code, ergänzt um operationsspezifische Ergebnisse) werden in den jeweiligen Schnittstellen Dateien (Yaml-Files) erfasst und dokumentiert.

A_26500 -PoPP-Service - externe Fehlercodes

Der PoPP-Service MUSS Fehlercodes gemäß der Schnittstellenbeschreibung umsetzen:

- [I_PoPP-Token_Generation.yaml] und
- [pip-pap-service.yaml].

[<=]

A_27317 -PoPP-Service - interne Fehlercodes

Der PoPP-Service MUSS folgende interne Fehlercodes verwenden:

Tabelle 10: Tab_PoPP_Service_interne_Fehlercodes

BDE-Code	Errorcode Referenz	Beschreibung	Fehler-adressat
79030	MISSING_OR_INVALID_HEADER	The required header <header> is missing or invalid.	Clientsystem
79031	UNSUPPORTED_MEDIATYPE	The clientsystem asked for an unsupported media type <media type>.	Clientsystem
79032	UNSUPPORTED_ENCODING	The clientsystem asked for an unsupported encoding scheme <encoding scheme>.	Clientsystem
79040	INVALID_HTTP_OPERATION	ERROR	Clientsystem
79041	INVALID_ENDPOINT	ERROR	Clientsystem
79100	SERVICE_INTERNAL_SERVER_ERROR	Unexpected internal server error.	Clientsystem
79112	OCSP_NOTREACHABLE	Certificate validation services can not be reached	HTTP-Proxy
79113	OCSP_TIMEOUT	Certificate validation services timed out	HTTP-Proxy
79114	INVALID_ACCESSTOKEN	Signature verification of the presented	Clientsystem

		access token failed (FdV)	
79115	EXPIRED_ACCESSTOKEN	Access token has expired (FdV)	Clientsystem
79116	EGK_INVALID_CVC	eGK was not authentic (CVC check failed)	HTTP-Proxy
79117	EGK_INVALID_AUT	eGK C.CH.AUT was invalid	HTTP-Proxy
79118	EGK_MISMATCH_AUT	eGK C.CH.AUT did not match the CVC	HTTP-Proxy
79205	MISSING_HEADER_CLIENTDATA	Header ZTA-Client-Data fehlt.	HTTP-Proxy
79206	MISSING_HEADER_USERINFO	Header ZTA-User-Info fehlt.	HTTP-Proxy
79400	ERROR_HEADER_CLIENTDATA	Client-Data Daten können nicht verarbeitet werden.	HTTP-Proxy
79401	ERROR_HEADER_USERINFO	User-Info Daten können nicht verarbeitet werden.	HTTP-Proxy
79403	ZETA_DPOP_VALIDATION_ERROR	Signature verification of the DPoP-JWT failed	Clientsystem
79404	ZETA_INVALID_ACCESSTOKEN	Signature verification of the presented access token failed	Clientsystem
79405	ZETA_EXPIRED_ACCESSTOKEN	Access token has expired	Clientsystem

[<=]

7 Testanforderungen

Anforderungen an Test und Testbarkeit des PoPP-Service finden sich in [gemKPT_Test].

<< Die neuen Test-Festlegungen für PoPP sind im Änderungsantrag C_12019
zusammengeführt. >>

8 Betrieb

In diesem Kapitel werden übergreifende, betriebliche Anforderungen getroffen oder auf Kapitel mit speziellen Ausprägungen für den Anbieter PoPP-Service in normativen Querschnittsdokumenten verwiesen.

<<Die Festlegungen werden im Änderungseintrag C_11939 zusammengeführt.>>

A_27390 -Performance - PoPP-Service - Zugriff für den Nutzer

Der Anbieter PoPP-Service MUSS den Zugriff auf den Dienst über einer einzigen URL ermöglichen. Die Nutzung von diensteigenen Redundanzmechanismen zur Aufrechterhaltung der Verfügbarkeit im Falle eines Teilausfalls darf keine Nutzerinteraktion erfordern.【<=】

8.1 Schnittstellen und Anwendungsfälle

Die vom PoPP-Service zur Verfügung gestellten Schnittstellen und Anwendungsfälle werden im entsprechenden Kapitel von [gemKPT_Betr] dargestellt.

8.2 Leistungsanforderungen und Performance

Die vom PoPP-Service zu leistenden Performancevorgaben werden im entsprechenden Kapitel von [gemSpec_Perf] dargestellt. Dazu gehören insbesondere Vorgaben zur Verfügbarkeit, eingesetzten Redundanz und der Leistungsfähigkeit der Schnittstellenabrufe. Darüber hinaus werden Vorgaben zur Verarbeitung der eingesetzten Datenliefermodelle gemacht, die sich sowohl auf den Fachdienst, als auch organisatorisch auf den entsprechenden Anbieter beziehen, welcher diese Datenlieferungen gewährleisten muss.

9 Anhang A - Verzeichnisse

9.1 Abkürzungen

Tabelle 11: Im Dokument verwendete Abkürzungen

Kürzel	Erläuterung
APDU	Application Protocol Data Units
ASN.1	Abstract Syntax Notation 1, eine Variante zur Spezifikation von Datenaustauschformaten
AuthZ	Autorisierung
BDE	Betriebsdatenerfassung
CAN	Card Access Number
CID	Change, Integration, Deletion
CSR	Certificate Signing Request
CVC	Card Verifiable Certificate
CVE	Common Vulnerabilities and Exposures
DiGA	Digitale Gesundheitsanwendungen
DPoP	Demonstrating Proof of Possession
ECIES	Elliptic Curve Integrated Encryption Scheme
eGK	elektronische Gesundheitskarte
eH-KT	eHealth-Kartenterminal
eIDAS	electronic Identification, Authentication and trust Services
FD	Fachdienst
FIPS	Federal Information Processing Standard
HMAC	Hash-based Message Authentication Code

HSM	Hardware-Sicherheitsmodul
HSM(C)-B	Hardware Security Module Typ B
HW	Hardware
IANA	Internet Assigned Numbers Authority
IDP	Identity Provider
IK	Institutionskennzeichen
ITSEC	Information Technology Security Evaluation Criteria
JSON	JavaScript Object Notation
JWK	JSON Web Key
JWKS	JSON Web Key Set
JWT	JSON Web Token
KTR	Kostenträger
KVNR	Krankenversichertennummer
MFA	Medizinische Fachangestellte
NFC	Near Field Communication
NFD	Notfalldatensatz
OAuth	Open Authorization
OCSP	Online Certificate Status Protocol
OID	Object Identifier
P-256	elliptische Kurve mit Domainparametern gemäß [SP800-186#3.2.1.3]
PACE	Password Authenticated Connection Establishment
PAP	Policy Administration Point
PAR	Pushed Authorization Request
PDP	Policy Decision Point

PEP	Policy Enforcement Point
PIP	Policy Information Point
PKI	Public Key Infrastructure
PoPP	Proof of Patient Presence
PS	Primärsystem
SIEM	Security Information and Event Management
SM(C)-B	Security Module Typ B
SMC-B	Security Module Card Typ B
TAN	Transaktionsnummer
TI	Telematikinfrastuktur
TSL	Trust Service Status List
US1	Umsetzungsstufe 1 (gleich Stufe 1) des PoPP-Service
UX	User Experience
VAU	Vertrauenswürdige Ausführungsumgebung
VK	Verarbeitungskontext
VSDM	Versichertenstammdatenmanagement
ZETA	Zero Trust Access

9.2 Glossar

Tabelle 12: Glossar der explizit im Dokument verwendeten Begriffe

Begriff	Erläuterung
CVC-Root	Die CVC-Root ist die zentrale Root-CA der PKI für CV-Zertifikate in der TI. Die CVC-Root ist ein Produkttyp.
Distinguished Encoding Rules (DER)	Eine Variante zur Codierung von ASN.1 Objekten als Bytestring.
GesundheitsID	Die GesundheitsID ist die digitale Identität im

	Gesundheitswesen für Versicherte, welche durch die eigene Krankenversicherung bereitgestellt wird. Sie dient zur Anmeldung an TI-Anwendungen und weiteren versorgungsrelevanten Fachanwendungen und kann perspektivisch auch als Versicherungsnachweis - analog zur elektronischen Gesundheitskarte - verwendet werden.
Leistungserbringer (LE)	Ein Leistungserbringer gehört zu einem zugriffsberechtigten Personenkreis nach § 352 SGB V und erbringt Leistungen des Gesundheitswesens für Versicherte. Nach § 339 SGB V darf er auf Versichertendaten in Anwendungen der TI zugreifen.
Leistungserbringereinstitution (LEI)	Die in organisatorischen Einheiten oder juristischen Personen zusammengefassten Leistungserbringer (bspw. Arztpraxen, Krankenhäuser).
Mobiles PS	Mobiles Endgerät, auf dem ein PS-Client einer LEI installiert ist. Ein LE nutzt den mobilen PS-Client bei Anwendungsfällen außerhalb der LEI.
PoPP-Client	Eine Komponente im Primärsystem, die für die sichere Kommunikation zum PoPP-Service verantwortlich ist.
PoPP-Modul	Eine Komponente einer Kassen-(Authenticator)-App, welche beim mobilen Check-in die Kommunikation mit dem PoPP-Service übernimmt.
PoPP-Service	Zentraler Dienst in der Telematikinfrastruktur 2.0 (TI 2.0), der PoPP-Token generiert und verwaltet.
PoPP-Service Resource Server	Der Server der PoPP-Token für PoPP-Clients erzeugt.
PoPP-Token	Der PoPP-Token dient als Nachweis für einen Versorgungskontext im Gesundheitswesen. Er ist ein kryptografisch gesicherter Beleg, der die Verbindung zwischen zwei Identitäten im Gesundheitswesen darstellt: dem Versicherten, bzw. dessen eGK, und einer LEI.
Proof of Patient Presence (PoPP)	PoPP ist ein Nachweis, der belegt, dass ein Versicherter sich zu einem bestimmten Zeitpunkt in einem Versorgungskontext mit einer bestimmten LEI befindet.
Versorgungskontext (VK)	Der Versorgungskontext beschreibt die sichere und kryptografisch belegte Verbindung zwischen einem berechtigten Versicherten und einer authentifizierten LEI. Diese Verbindung autorisiert den Zugriff auf anwendungsbezogene Versicherungsdaten über die Telematikinfrastruktur (TI) Anwendungen. Ein Versorgungskontext besteht, wenn ein

	<p>Leistungserbringer und ein Versicherter zum Zweck einer Versorgung zusammenkommen. Dabei kann die Versorgung eine medizinische Behandlung, eine pflegerische Leistung oder eine andere Versorgungsleistung sein, beispielsweise in einer Apotheke. Das Zusammentreffen kann lokal in einer Leistungserbringerumgebung, mobil, beispielsweise bei einem Hausbesuch oder virtuell, beispielsweise bei einer Telefon- oder Videosprechstunde sein.</p> <p>Ein Versorgungskontext entsteht durch die erfolgreiche Authentifizierung des Versicherten mittels digitaler Identität oder durch die erfolgreiche Authentifizierung seiner eGK, und ist auch bei telemedizinischen Anwendungen relevant.</p>
ZETA Client	Zero Trust Client Komponente im Primärsystem; Client Komponente gegenüber der Zero Trust Server Komponente ZETA Guard.
ZETA Guard	Der beim Fachdienst einzubindende Zero Trust Cluster.

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

9.3 Abbildungsverzeichnis

Abbildung 1: EinfacherSystemüberblick.....	10
Abbildung 2: Rollen und Akteure bei Herstellung und Betrieb des PoPP-Service.....	13
Abbildung 3: Produkttypzerlegung.....	15
Abbildung 4: Systemkontext PoPP-Lösung (Stufe 1).....	16
Abbildung 5: Anwendungsfälle zur Attestierung des Versorgungskontexts für Stufe 1.....	20
Abbildung 6: Ausstellung PoPP-Token nach Stecken der eGK in LEI.....	23
Abbildung 7: Zustandsdiagramm für die Verarbeitung einer eGK.....	64
Abbildung 8: Karnaugh-Veitch Diagramm zur "check"-Funktion.....	80
Abbildung 9: Karnaugh-Veitch Diagramm zur "import"-Methode.....	85

9.4 Tabellenverzeichnis

Tabelle 1: PoPP-Use Cases (Business Sicht).....	11
Tabelle 2: Kurzbeschreibung der Komponenten in der PoPP-Lösung für die Stufe 1.....	17
Tabelle 3: Zuordnung der Anwendungsfälle zu den Use Cases.....	21
Tabelle 4: PoPP-Token Claims (informativ) berücksichtigt sind Stufe 1 und Stufe 2.....	28

Tabelle 5: PoPP-Token Header (informativ).....	30
Tabelle 6: Übersicht über die im PoPP-Service verwendeten Schlüssel.....	52
Tabelle 7: Attribute im well-known document des PoPP-Service.....	59
Tabelle 8: Attribute des Metadatenblock federation_entity im well-known document des PoPP-Service.....	60
Tabelle 9: Fehlermeldungen eGK-Handling.....	76
Tabelle 10: Tab_PoPP_Service_interne_Fehlercodes.....	95
Tabelle 11: Im Dokument verwendete Abkürzungen.....	99
Tabelle 12: Glossar der explizit im Dokument verwendeten Begriffe.....	101
Tabelle 13: Referenzierte Dokumente der gematik.....	104
Tabelle 14: Weitere Dokumente.....	106
Tabelle 15: Entity Statement des PoPP-Service.....	109

9.5 Referenzierte Dokumente

9.5.1 Dokumente der gematik

Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument referenzierten Dokumente der gematik zur Telematikinfrastruktur.

Tabelle 13: Referenzierte Dokumente der gematik

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Einführung der Gesundheitskarte – Glossar
[TI-Föderation]	Fachanwendungen der TI-Föderation (aus der Wissensdatenbank der gematik, zuletzt abgerufen am 21.08.2024) https://wiki.gematik.de/pages/viewpage.action?pageId=523502009
[gemILF_PoPP_Client]	Implementierungsleitfaden Primärsystemfunktionalität PoPP-Client https://github.com/gematik/spec-ilf-popp-client/tree/main (Version 1.0.0 vom 04.07.2025)
[gemKPT_Betr]	Betriebskonzept Online-Produktivbetrieb https://gemspec.gematik.de/docs/gemKPT/gemKPT_Betr/
[gemKPT_Test]	Testkonzept der TI https://gemspec.gematik.de/docs/gemKPT/gemKPT_Test/
[gemSpec_IDP_Dienst]	Spezifikation Identity Provider-Dienst https://gemspec.gematik.de/docs/gemSpec/gemSpec_IDP_D

	ienst/
[gemSpec_IDP_FD]	Spezifikation Identity Provider – Nutzungsspezifikation für Fachdienste https://gemspec.gematik.de/docs/gemSpec/gemSpec_IDP_FD/
[gemSpec_IDP_Sek]	Spezifikation Sektoraler Identity Provider https://gemspec.gematik.de/docs/gemSpec/gemSpec_IDP_Sek/
[gemSpec_ZETA]	Spezifikation Zero Trust Access (ZETA) https://gemspec.gematik.de/docs/gemSpec/gemSpec_ZETA
[gemSpec_PoPP_Modul]	Spezifikation PoPP-Modul (Dokument noch in Erstellung)
[gemSpec_DS_Anbieter]	Spezifikation Datenschutz- und Sicherheitsanforderungen der TI an Anbieter https://gemspec.gematik.de/docs/gemSpec/gemSpec_DS_Anbieter/
[gemSpec_OID]	Spezifikation Festlegung von OIDs https://gemspec.gematik.de/docs/gemSpec/gemSpec_OID/
[gemSpec_Krypt]	Übergreifende Spezifikation Verwendung kryptografischer Algorithmen in der Telematikinfrastruktur https://gemspec.gematik.de/docs/gemSpec/gemSpec_Krypt/
[gemAPI_ZT]	gematik: OpenAPI Schnittstellenspezifikation Zero Trust https://github.com/gematik/spec-t20r
[gemSpec_DS_Hersteller]	Spezifikation Datenschutz- u. Sicherheitsanforderungen der TI an Hersteller https://gemspec.gematik.de/docs/gemSpec/gemSpec_DS_Hersteller/
[gemSpec_TSL]	Spezifikation TSL-Dienst https://gemspec.gematik.de/docs/gemSpec/gemSpec_TSL/
[gemSpec_eGK_ObjSys_G2.1]	Spezifikation der elektronischen Gesundheitskarte eGK-Objektsystem https://gemspec.gematik.de/docs/gemSpec/gemSpec_eGK_ObjSys_G2_1/
[gemSpec_COS]	Spezifikation des Card Operating System (COS) https://gemspec.gematik.de/docs/gemSpec/gemSpec_COS/
[gemSpec_Kon]	Spezifikation Konnektor https://gemspec.gematik.de/docs/gemSpec/gemSpec_Kon/

[pip-pap-service.yaml]	OpenAPI Schnittstellenspezifikation für Policy Information Point und Policy Administration Point API https://raw.githubusercontent.com/gematik/spec-t20r/development/src/openapi/pip-pap-api.yaml
[I_PoPP_Token_Generation.yaml]	OpenAPI Schnittstellenspezifikation für PoPP-Service Resource Server für PoPP-Clients: https://github.com/gematik/api-popp/blob/publishInternalRelease-5/src/openapi/I_PoPP_Token_Generation.yaml

9.5.2 Weitere Dokumente

Tabelle 14: Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[AnbieterVZeitD]	Die Bundesnetzagentur listet unter https://www.elektronische-vertrauensdienste.de/EVD/DE/Uebersicht_eVD/Dienste/5_Zeitstempel.html?nn=691392 https://www.elektronische-vertrauensdienste.de/EVD/DE/Uebersicht_eVD/Dienste/5_Zeitstempel.html?nn=691392 verschiedene Anbieter für qualifizierte Zeitstempel.
[FIPS PUB 180-4]	FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, Secure Hash Standard (SHS), August 2015 http://dx.doi.org/10.6028/NIST.FIPS.180-4
[OpenID Federation 1.0]	OpenID Federation 1.0 https://openid.net/specs/openid-federation-1_0.html
[OpenID Connect Core 1.0]	OpenID Connect Core 1.0 https://openid.net/specs/openid-connect-core-1_0.html
[OWASP-Top-10-Risiken]	OWASP Top 10 https://owasp.org/www-project-top-ten/ (Abruf 01/2025)
[RFC7519#Sect.2]	JSON Web Token (JWT) https://datatracker.ietf.org/doc/html/rfc7519#section-2
[RFC1760]	The S/KEY One-Time Password System https://www.rfc-editor.org/rfc/rfc1760
[RFC2289]	A One-Time Password System https://www.rfc-editor.org/rfc/rfc2289
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels https://www.rfc-editor.org/rfc/rfc2119
[RFC 5639]	Elliptic Curve Cryptography (ECC) Brainpool Standard Curves

	and Curve Generation https://www.rfc-editor.org/rfc/rfc5639
[RFC 6962]	Certificate Transparency https://www.rfc-editor.org/rfc/rfc6962
[RFC 9162]	Certificate Transparency Version 2.0 https://www.rfc-editor.org/rfc/rfc9162
[CAB Forum]	https://cabforum.org/
[RFC7518]	JSON Web Algorithms (JWA) https://www.rfc-editor.org/rfc/rfc7518
[RFC7517]	JSON Web Key (JWK) https://www.rfc-editor.org/rfc/rfc7517
[RFC7638]	JSON Web Key (JWK) Thumbprint https://www.rfc-editor.org/rfc/rfc7638
[RFC7519]	JSON Web Token (JWT) https://www.rfc-editor.org/rfc/rfc7519
[RFC7515]	JSON Web Signature (JWS) https://www.rfc-editor.org/rfc/rfc7515
[RFC6749]	The OAuth 2.0 Authorization Framework https://datatracker.ietf.org/doc/html/rfc6749
[RFC 6844]	DNS Certification Authority Authorization (CAA) Resource Record https://www.rfc-editor.org/rfc/rfc6844
[RFC9396]	OAuth 2.0 Rich Authorization Requests https://www.rfc-editor.org/rfc/rfc9396.html
[RFC5639]	Elliptic Curve Cryptography (ECC) Brainpool Standard - Curves and Curve Generation https://datatracker.ietf.org/doc/html/rfc5639
[RFC 6066]	Transport Layer Security (TLS) Extensions: Extension Definitions https://datatracker.ietf.org/doc/html/rfc6066
[RFC5652]	Cryptographic Message Syntax (CMS) https://datatracker.ietf.org/doc/html/rfc5652
[RFC4122#3]	Namespace Registration Template https://datatracker.ietf.org/doc/html/rfc4122#section-3
[RFC 5280]	Validity

	https://www.rfc-editor.org/rfc/rfc5280#section-4.1.2.5
[SP800-186]	Chen L, Moody D, Regenscheid A, Robinson A, Randall K (2023) Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-186. https://doi.org/10.6028/NIST.SP.800-186
[jwk-set+jwt]	https://www.iana.org/assignments/media-types/application/jwk-set+jwt
[BSI-QDDoS]	BSI: Qualifizierte DDoS-Mitigation Dienstleister, aktuelle Fassung https://www.allianz-fuer-cybersicherheit.de/Webs/ACS/DE/Informationen-und-Empfehlungen/Informationen-und-weiterfuehrende-Angebote/Qualifizierte-Dienstleister/qualifizierte-dienstleister_node.html (zuletzt aufgerufen am 28.11.2024)
[BSI ISI-LANA]	BSI: Standards zur Internet-Sicherheit (ISi-Reihe) https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/ISi-Reihe/isi-reihe_node.html
[SP800-186#3.2.1.3]	Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters https://doi.org/10.6028/NIST.SP.800-186
[https://app.federationmaster.de/federation/list]	https://app.federationmaster.de/federation/list

9.6 Allgemeine Erläuterungen

9.6.1 Entity Statement des PoPP-Service

Tabelle 15: Entity Statement des PoPP-Service

Name	Werte	Beispiel	Anmerkungen
iss	URL	"https://popp-service.de"	URL des PoPP-Service, Identifier in der Föderation
sub	URL	"https://popp-service.de"	URL des PoPP-Service (=iss)

iat	Alle time Werte in Sekunden seit 1970, [RFC7519]#Sect .2	1645484401	2022-02-22 00:00:01
exp	Alle time Werte in Sekunden seit 1970, [RFC7519]#Sect .2	1645570800	Gültigkeit von 24 Stunden
jwks	JWKS Objekt		Federation Entity Key für die Signatur des Entity Statement [OpenID Federation 1.0]
authority_hints	[string]	"https://app.federationmaster.de"	iss Bezeichnung des Federation Master
<i>metadata {</i>			
<i>federation_entity{</i>			
organization_name	String	Hersteller PoPP-Service	Optional: Name der Organisation die hinter dem Fachdienst steht
contacts	[strings]	"support@popp-hersteller.de", "info@popp-hersteller.de"	Optional
homepage_uri	URL	"https://popp-hersteller.de"	Optional
logo_uri	URL	"https://popp-hersteller.de/logo.jpg"	Optional
<i>}</i>			
<i>oauth_resource {</i>			
signed_jwks_uri	URL	"https://popp.service.de/"	Schlüssel

		jwks.json"	welche die Protected Resource für die Signatur und für die Verschlüsselung verwendet.
authorization_server	URL	"https://popp.auth.de"	Identifizier (issuer) des PoPP-Service Authorization Server
}			
}			

9.7 Offene Punkte / Klärungsbedarf

- derzeit keine -