

Telematikinfrastruktur 2.0

Spezifikation PoPP (Proof of Patient Presence) -Modul

Version: 1.0.0 CC
Revision: 1494668
Stand: 26.01.2026
Status: zur Abstimmung
freigegeben
Klassifizierung: öffentlich_Entwurf
Referenzierung: gemSpec_PoPP_Modul

Dokumenteninformationen

Änderungen zur Vorversion

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

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0 CC	26.01.2026		initiale Erstellung - zur Abstimmung freigegeben	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes.....	6
1.1 Zielsetzung.....	6
1.2 Zielgruppe.....	6
1.3 Geltungsbereich.....	6
1.4 Abgrenzungen.....	7
1.5 Methodik.....	7
2 Systemüberblick/Systemkontext.....	8
2.1 Detailsicht "Online-Anwendungsfälle zur PoPP-Token-Erstellung".....	8
3 Übersicht Funktionsmerkmale und Schnittstellen.....	15
3.1 Authentifizierung mit GesundheitsID.....	16
3.2 Authentifizierung einer eGK (in Fernversorgung).....	17
3.3 Schnittstellen des PoPP-Moduls.....	17
3.3.1 initPoppTokenGeneration().....	18
3.3.2 startPoppTokenErzeugung().....	19
3.3.3 hasGesundheitsID().....	20
3.3.4 init().....	20
3.3.5 execute(httpLoadPractitionerInformationRequest).....	21
3.3.6 callBackLoadPractitionerInformation.....	21
3.3.7 execute(httpReadEgkRequest).....	22
3.3.8 callBackReadEgk - bei Authentifizierung einer eGK.....	22
3.3.9 sendAPDUCommands().....	23
3.3.10 execute(httpCheckInGIDRequest).....	24
3.3.11 callBackCheckInGID.....	24
3.3.12 execute(httpReadStatusPoPPRequest).....	25
3.3.13 callBackReadStatusPoPP.....	25
3.3.14 callBackApp.....	26
3.4 PoPP-Modul in Drittanbieter-APP.....	26
4 Übergreifende Festlegungen.....	28
4.1 Datenschutz und Informationssicherheit.....	28
5 Funktionsmerkmale.....	30
5.1 Allgemeine funktionale Anforderungen PoPP-Modul.....	30
5.2 Auswahl der LEI/ Bestimmen der Telematik-ID.....	30
5.2.1 Favoriten.....	30
5.2.2 Verzeichnisdienstsuche.....	31
5.2.3 QR-Code.....	31
5.3 Auslösen der Erstellung eines PoPP-Token.....	32
5.3.1 Anforderungen PoPP-Modul bei Authentisierung eines Versicherten mit GesundheitsID.....	32

77	5.3.2 Anforderungen PoPP-Modul für eGK-in-Fernversorgung.....	34
78	5.3.3 Kommunikation PoPP-Modul mit ZETA Client.....	35
79	5.3.4 Kommunikation PoPP-Modul mit Drittanbieter-Apps.....	36
80	5.4 Verarbeitung des Ergebnisses der PoPP-Token-Erzeugung.....	37
81	6 Informationsmodell.....	39
82	7 Implementierungsleitfaden.....	40
83	7.1 End-to-End Flow (Happy Path).....	40
84	7.2 Drittanbieter-Apps.....	42
85	7.2.1 Integriertes PoPP-Modul.....	42
86	7.2.2 Nachnutzung des PoPP-Moduls (App2App).....	43
87	7.3 Alternative Flow-Schritte.....	43
88	7.3.1 Autorisierung.....	43
89	7.3.2 Auswahl der LEI.....	45
90	7.3.3 Statusinformation.....	49
91	7.3.4 Ersteinrichtung.....	49
92	7.4 Terminologie.....	51
93	8 Test.....	53
94	8.1 Schnittstelle für Testtreiber.....	53
95	9 Anhang A - Verzeichnisse.....	54
96	9.1 Abkürzungen.....	54
97	9.2 Glossar.....	54
98	9.3 Abbildungsverzeichnis.....	57
99	9.4 Tabellenverzeichnis.....	57
100	9.5 Referenzierte Dokumente.....	58
101	9.5.1 Dokumente der gematik.....	58
102	9.5.2 Weitere Dokumente.....	60
103	10 Anhang B - Ablaufbeschreibungen.....	61
104	10.1 Ablaufdiagramme.....	61
105	10.1.1 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung Versicherter über ihre GesundheitsID.....	61
106	10.1.2 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service.....	63
107	10.1.3 Detailablauf "ZETA Client Initialisierung".....	64
108	10.1.4 Detailablauf "FHIR-VZD-Anfrage".....	64
109	10.1.5 Detailablauf "Information zum Status der PoPP-Token-Generierung an das PoPP-Modul".....	65
110	10.2 Flowdiagramme.....	66
111	10.2.1 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung Versicherter über ihre GesundheitsID.....	66
112	10.2.2 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung Versicherter mit ihrer GesundheitsID.....	68
113	10.2.3 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service.....	82

120	10.2.4 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung	
121	einer eGK.....	84
122	10.2.5 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung der	
123	eGK über den PoPP-Service mit PoPP-Modul in einer Drittanbieter-App.....	97
124	10.2.6 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung	
125	einer eGK mit PoPP-Modul in einer Drittanbieter-App.....	99
126	11 Anhang C - Offene Punkte, Fragen.....	111
127	11.1 Offene Punkte.....	111
128		
129		

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die Anforderungen zur Herstellung, zum Test und zum Betrieb des Produkttyps Proof of Patient Presence (PoPP)-Modul. Diese bilden auf Versichertenseite den clientseitigen Gegenpart zu den serverseitigen Festlegungen in der Spezifikation [gemSpec_PoPP_Service]. In die Spezifikation sind die Ergebnisse intensiver Abstimmungen mit den Gesellschaftern der gematik als Konsumenten der gesamten PoPP-Lösung insbesondere zu Themen bei Nutzung der GesundheitsID eingeflossen.

Das PoPP-Modul bietet Versicherten mit GesundheitsID oder bei Nutzung der eGK-in-Fernversorgung den Zugang zur PoPP-Lösung. Der PoPP-Service erzeugt die Bestätigung eines Versorgungskontextes (VK) in Form eines kryptographisch gesicherten PoPP-Token. Dieses bestätigt, dass ein bestimmter Versicherter mit einer bestimmten Leistungserbringerinstitution (LEI) in einen Versorgungskontext getreten ist.

Neben dem PoPP-Modul, tragen weitere Komponenten zur PoPP-Lösung bei:

- Der PoPP-Service [gemSpec_PoPP_Service] ist der Server-Anteil der PoPP-Lösung.
- Die PoPP-Clients, die als Teil der Primärsysteme implementiert werden [gemILF_PoPP_Client].

Die Spezifikationen oder Beschreibungen dieser Komponenten erfolgt in den anderen Dokumenten.

1.2 Zielgruppe

Dieses Dokument richtet sich an Hersteller und Betreiber von Apps für Versicherte, die ein PoPP-Modul in ihre App integrieren wollen, insbesondere an die Kassen und ihre Auftragnehmer sowie Drittanbieter.

1.3 Geltungsbereich

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

In dem Dokument werden die von dem Produkttyp bereitgestellten (angebotenen) Schnittstellen spezifiziert. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert (siehe auch [Anhang 9]).

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

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 (Beispiel)

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

2 Systemüberblick/Systemkontext

Das Konzept für den PoPP Online-Check-in ist in der Feature Spezifikation [gemF_PoPP_Online_Check-in] beschrieben.

Ein ausführlicher Systemüberblick und eine Darstellung der Anwendungsfälle sind in [gemSpec_PoPP_Service] dargestellt.

PoPP-Module müssen funktionale und sicherheitstechnische Anforderungen der gematik erfüllen. Welche Anforderungen ein PoPP-Modul erfüllen muss, wird von der gematik in einem Produkttypsteckbrief bekanntgegeben.

Die Erfüllung der Anforderungen muss der Hersteller eines PoPP-Moduls der gematik in einem Zulassungsverfahren nachweisen.

Ein Systemüberblick und die Erläuterung dazu sind der in "Systemkontext PoPP-Lösung" dargestellten Komponenten findet sich in [gemSpec_PoPP_Service].

2.1 Detailsicht "Online-Anwendungsfälle zur PoPP-Token-Erstellung"

An der über eine App (Krankenversicherung-App, Drittanbieter-App) auf dem mobilen Endgerät des Versicherten initiierten Erstellung eines PoPP-Token sind mehrere Architekturkomponenten der TI beteiligt.

Die ZETA-Komponenten (Zero Trust-Architektur) garantieren die Vertrauenswürdigkeit des Geräts des Versicherten und der für PoPP zum Einsatz kommenden Anwendung auf diesem Gerät.

Die Komponenten der TI-Föderation garantieren die Authentifizierung der Person, welche die Anwendung nutzt, für den Fall, dass eine Authentifizierung über die GesundheitsID erfolgt.

Die Komponenten des PoPP-Service garantieren die Ausstellung eines PoPP-Token an die LEI, welche der Versicherte ausgewählt hat und für die er in eine Datenweitergabe eingewilligt hat.

PoPP-Modul und ZETA Client müssen dabei immer gemeinsam in eine App integriert sein. Die "Detailsicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token Erstellung mit PoPP-Modul in App der Krankenversicherung" zeigt die beteiligten Komponenten, wenn PoPP-Modul und ZETA Client in die Krankenversicherungs-App integriert sind, welche auch das Authenticator-Modul des sektoralen IDP der Krankenversicherung implementiert.

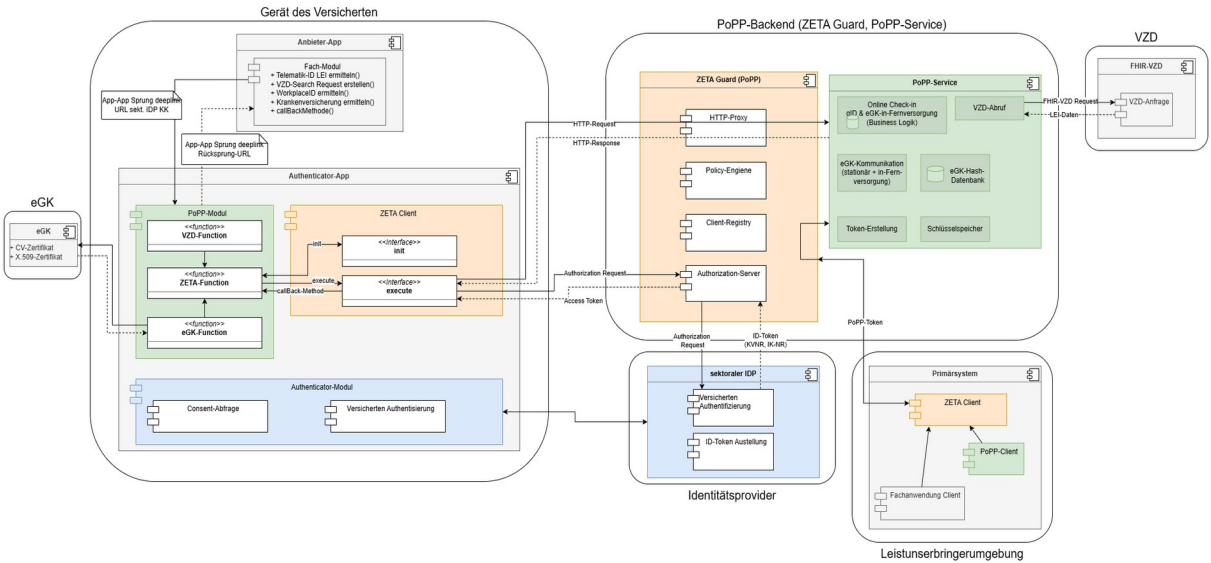


Abbildung 1: Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in App der Krankenversicherung

Diese Konstellation unterstützt alle Online-Anwendungsfälle zur PoPP-Token-Erstellung der Krankenversicherungen optimal. Anwendungsfälle von Drittanbieter-Apps werden ebenfalls unterstützt. Im Ablauf findet ein App-Wechsel von der Drittanbieter-App in die Krankenversicherungs-App und zurück statt. Es erfordert keine durch die gematik zuzulassende Implementierung.

Offener Punkt: OP-PoPP-3
Im Folgenden wird an verschiedenen Stellen auf die ZETA Client-Registrierung verwiesen. Diesbezüglich gibt es für die eGK-Anwendungsfälle einen offenen Punkt. Verwendet der Nutzer ausschließlich die eGK (also keine GesundheitsID) wird bzgl. ZETA eine rein Client-gebundene Client-Registrierung also ohne Authentifizierung des Nutzers benötigt. Dies wird aktuell noch in den ZETA-Spezifikationen umgesetzt. Die bisher in gemSpec_ZETA#5.2.3 beschriebene Client-Registrierung ist Nutzer-gebunden und erfordert daher zwingend die Authentifizierung des Nutzers. Letztere kann bei ausschließlichem Vorhandensein der eGK nicht stattfinden. Der Entwurf für die rein Client-gebundene Registrierung wird parallel bzw. zeitnah veröffentlicht.

Tabelle 1: Beschreibung der wesentlichen Komponenten für die Online Anwendungsfälle zur PoPP-Token Erstellung

Komponente	Beschreibung
Gerät des Versicherten	
Gerät des Versicherten	Auf dem Gerät des Versicherten (i.d.R. Smartphone) sind Anwendungen (Apps) installiert, die für ihren fachlichen Ablauf die Erstellung eines PoPP-Token für eine LEI benötigen. Auf dem Gerät des Versicherten ist eine App installiert, welche PoPP-Modul, ZETA Client und Authenticator-Modul integriert (Krankenversicherungs-App, Authenticator-App).

	<p>Das Gerät verfügt über eine Menge von Eigenschaften, welche bei der sicheren Kommunikation mit den Backend-Systemen berücksichtigt werden müssen.</p> <p>Im Rahmen der ZETA Guard Client-Registrierung wird das Gerät einer Attestierung (Device-Attestation) unterzogen, bei der die Eigenschaften im ZETA Guard hinterlegt und für Zugriffsentscheidungen auf Dienste herangezogen werden.</p> <p>Bei den Eigenschaften handelt es sich u.a. um Gerätetyp, Gerätehersteller, Version des installierten Betriebssystems, Informationen zu Hardwareausstattung (z.B. Schlüsselspeicher).</p>
Anwendung/App auf dem Gerät des Versicherten	<p>Die auf dem Gerät des Versicherten installierten Anwendungen, welche die Erstellung eines PoPP-Token anfordern, können vielfältig ausgeprägt sein:</p> <ul style="list-style-type: none"> • Personalisierte Anwendungen (Krankenversicherung-Apps) für Versicherte zur Vermittlung einer Kommunikation zwischen Versicherten und LEI • Anwendung einer LEI für Versicherte (z.B. Online-Apotheke) • Anwendung zur Vermittlung der Kommunikation zwischen Versicherten und LEI (z.B. Telemedizin-Anwendungen) • Anwendungen für eine bestimmte Gruppe von Versicherten zur Kommunikation mit einer bestimmten Gruppe von LEI (z.B. Patientenportale für Kliniken) <p>Bedingt durch die Vielfalt der Ausprägungen sind die Voraussetzungen für die Erstellung von PoPP-Token unterschiedlich:</p> <ul style="list-style-type: none"> • Ist die Telematik-ID der LEI der Anwendung bereits bekannt oder muss sie erst ermittelt werden (Online-Apotheke vs. Versichertenportal)? • Kann die Telematik-ID ermittelt werden oder muss eine VZD-Suche der LEI mit Suchkriterien durchgeführt werden? • Ist die Krankenversicherung des Versicherten bekannt oder muss diese ermittelt werden (Krankenversicherung-App vs. Online-Apotheke)? <p>Fehlende Informationen müssen durch die Anwendung selbst beschafft werden, bevor eine PoPP-Token-Erstellung durchgeführt werden kann.</p>
Authenticator-App / Authenticator-Modul auf dem Gerät des Versicherten	<p>Werden Versicherte über ihre GesundheitsID am sektoralen IDP ihrer Krankenversicherung authentifiziert, so läuft dies über das zum sektoralen IDP gehörende Authenticator-Modul. Das Authenticator-Modul ist das Frontend, über welches Versicherte die Authentisierung</p>

	<p>mit einem der möglichen Authentisierungsmittel durchführen.</p> <p>Die Authenticator-App ist die App auf dem Gerät des Versicherten, welches das Authenticator-Modul des sektoralen IDP der Krankenversicherung des Versicherten implementiert.</p> <p>Es gibt zwei Ausprägungen der Umsetzung:</p> <ol style="list-style-type: none"> 1. Das Authenticator-Modul ist in die Krankenversicherung-App/ePA-FdV-App integriert. 2. Das Authenticator-Modul ist als eigene App implementiert. <p>In den dargestellten Umsetzungsvarianten ist das PoPP-Modul zusammen mit dem ZETA Client in einer Anwendung integriert. Dabei ist für Krankenversicherungen immer die App gemeint, welche auch das Authenticator-Modul implementiert.</p>
PoPP-Modul	<p>Das PoPP-Modul implementiert alle für die Erstellung eines PoPP-Token notwendigen Prozessschritte auf dem Gerät des Versicherten. Die UX des PoPP-Moduls ist analog zum Authenticator-Modul anpassbar.</p> <ul style="list-style-type: none"> • VZD-Funktion <p>Für die Nutzerzustimmung zur Verwendung der Versichertendaten durch eine LEI wird vom PoPP-Modul über die ZETA-Komponenten eine Anfrage an den PoPP-Service gestellt.</p> <p>Das PoPP-Modul formuliert dafür einen Anfrage-Request für das FHIR-VZD mit den zur Verfügung stehenden Parametern wie Name, PLZ der LE oder übernimmt einen fertigen Anfrage-Request aus dem Aufruf der App.</p> <p>Der PoPP-Service führt den Anfrage-Request beim FHIR-VZD durch. Das Ergebnis wird vom PoPP-Service an das PoPP-Modul gesendet und hier nach Regeln gefiltert. Das PoPP-Modul stellt die Daten der LEI für die Nutzereinwilligung dar.</p> • ZETA-Funktion <p>Die Kommunikation zwischen PoPP-Modul und PoPP-Service für die VUD-Abfrage erfolgt über die ZETA-Komponenten.</p> <p>Bei einer Authentifizierung des Versicherten über dessen GesundheitsID kommuniziert das PoPP-Modul ebenfalls nicht direkt mit dem PoPP-Service.</p> <p>Die Kommunikation erfolgt hier auch über die Komponenten der Zero Trust-Architektur ZETA Client (auf der Frontend-Seite) sowie ZETA Guard Authorization-Server und ZETA Guard HTTP-Proxy (auf der Backend-Seite).</p> <p>Im Fall der Authentifizierung der eGK ("eGK-in-Fernversorgung") kommuniziert der PoPP-Service</p>

	<p>ebenfalls über die ZETA-Komponenten mit dem PoPP-Modul.</p> <ul style="list-style-type: none"> eGK-Funktion <p>Für das Auslesen der eGK ("eGK in Fernversorgung") agiert das PoPP-Modul als Proxy. Vom PoPP-Service empfangende APDU-Commands werden über die NFC-Schnittstelle des Gerätes direkt an die eGK gesendet. Die Antworten der eGK werden direkt an dem PoPP-Service gesendet.</p> <p>Des Weiteren speichert das PoPP-Modul Informationen zur Historie der PoPP-Anfragen.</p>
ZETA Client	<p>ZETA Client ist die Zero Trust-Komponente für ein FdV, das zusammen mit dem PoPP-Modul in einer App integriert sein muss. Der ZETA Client überträgt bei der Initialisierung der Anwendung die Geräte- und App-Informationen an den ZETA Guard.</p> <p>Im laufenden Prozess kapselt der ZETA Client die gesamte Kommunikation zwischen der Anwendung auf dem Gerät des Versicherten und den Komponenten im Backend, u.a. auch zwischen PoPP-Modul und PoPP-Service.</p> <p>Der ZETA Client steuert die Client-Authentisierung am ZETA Guard Authorization-Server und unterstützt die Nutzerauthentifizierung mit GesundheitsID.</p>
PoPP-Service ZETA Guard	
ZETA Guard Client-Registry	<p>Alle Clients bzw. Anwendungen auf dem Gerät des Versicherten, die ein ZETA Client implementieren, müssen an der ZETA Guard Client-Registry registriert werden. Der Prozess der Registrierung ist in [gemSpec_ZETA] beschrieben.</p> <p>Im laufenden Prozess wird über die ZETA Guard Client-Registry geprüft, ob ein Aufruf von einem bekannten bzw. registrierten ZETA Client kommt.</p>
ZETA Guard Authorization-Server	<p>Bevor eine Anwendung über ein registriertes ZETA Client auf einen Fachdienst zugreifen kann, führt ZETA Guard Authorization-Server die Client-Authentifizierung durch und stellt dem ZETA Client der Anwendung bei erfolgreicher Authentifizierung ein Access-Token aus.</p> <p>Wird von der Anwendung darüber hinaus die Authentifizierung des Nutzers über die GesundheitsID angefragt, so stößt der ZETA Guard Authorization-Server den Authentifizierungsprozess beim jeweiligen Sektoren IDP an. Der ZETA Guard Authorization-Server nimmt bei erfolgreicher Authentifizierung das ID-Token entgegen und verarbeitet dessen Inhalt.</p>
ZETA Guard HTTP-Proxy	<p>Fachliche Requests aus den Anwendungen werden vom ZETA Client der Anwendung an den ZETA Guard HTTP-</p>

	<p>Proxy geschickt, nach Prüfung des Clients gegen die ZETA Guard Client-Registry und ggf. Durchführung von Prüfregeln durch die ZETA Guard Policy-Engine reichert der ZETA Guard HTTP-Proxy den Request mit Header-Informationen an und sendet den Request dann an den eigentlichen Empfänger. Das Ergebnis des Requests wird dem ZETA Client der aufrufenden Anwendung zugestellt.</p>
PoPP-Service	
eGK-Kommunikation (Stationär + in-Fernversorgung)	<p>Versicherte haben die Möglichkeit eine eGK ("eGK-in-Fernversorgung") für die Erstellung eines PoPP-Token zu verwenden.</p> <p>Wählt der Versicherte "eGK-in-Fernversorgung", so werden durch die Komponente "eGK-Kommunikation" APDU-Commands erzeugt, die über die ZETA-Komponenten an das PoPP-Modul und von dort an die eGK propagiert werden.</p> <p>Die Komponente "eGK-Kommunikation" nimmt die Antworten entgegen und wertet diese aus.</p> <p>Dazu werden die von der eGK gelesenen Zertifikate auf Gültigkeit geprüft. Sind die Zertifikate selbst gültig wird geprüft, ob die Zertifikate zur ein und derselben eGK gehören. Diese Prüfung wird dann in der Hash-DB durchgeführt.</p> <p>Der Einsatz der eGK ohne PIN entspricht einer Autorisierung und erfüllt nicht die Voraussetzung für ein Vertrauensniveau nach [gemSpec_IDP_Sek]. Das ausgestellte PoPP-Token enthält diese Information im Attribut "proofMethod". Dienste, bei denen das PoPP-Token eingereicht wird, können entscheiden, ob und für was eine Autorisierung der LEI für den Datenzugriff erfolgen darf.</p>
eGK-Hash-Datenbank	<p>Die eGK-Hash-DB hält zu jeder ausgegeben eGK einen Datensatz bestehend aus dem Hashwert des X.509-Zertifikats und dem Hashwert des CV-Zertifikats. Zur Prüfung der Validität der von einer eGK gelesenen Daten, werden diese gegen die eGK-Hash-Datenbank abgeglichen.</p>
VZD-Abfrage	<p>Der PoPP-Service führt auf Anfrage von einem PoPP-Modul die Suche im FHIR-VZD mit den vom PoPP-Modul übertragenen Search-Request durch. Der PoPP-Service ist ein am VZD registrierter Client und hat für die VZD-Abfrage entweder ein Access-Token oder Client-Credentials zur Ausstellung eines Access-Token durch den FHIR-VZD.</p> <p>Der PoPP-Service propagiert das Ergebnis der Suchanfrage unverändert über die ZETA-Komponenten an das PoPP-Modul.</p>
Token-Erstellung	<p>Die Komponente Token-Erstellung enthält die Funktionalität zum Erstellen des PoPP-Token. Beim</p>

	Online-Check-in wurde zuvor ein PoPP-Datensatz mit allen PoPP-Token relevanten Daten erstellt, die vom PoPP-Modul beim Online-Check-in Vorgang an den PoPP-Service übermittelt wurden. Das PoPP-Token selbst wird erst beim Abruf durch das Primärsystem einer LEI aus dem PoPP-Datensatz erstellt.
Online Check-in gID & eGK-in- Fernversorgung	Die Komponente Online-Check-in enthält die steuernde Funktionalität, wenn ein Versicherter einen Online-Check-in durchführt. Sie implementiert die Schnittstellen, welche aus dem PoPP-Modul durch den ZETA HTTP-Proxy am PoPP-Service aufgerufen werden.
sektorale IDPs	
sektoraler IDP	Die Authentifizierung Versicherter mit GesundheitsID erfolgt durch den sektoralen IDP der Krankenversicherung des Versicherten. Der sektorale IDP hält einen Datensatz mit Daten zum Versicherten selbst (GesundheitsID) sowie Informationen zu den möglichen Authentisierungsmitteln und den Nutzerpräferenzen. Die Authentifizierung Versicherter erfolgt dann in Verbindung mit dem Authenticator-Modul auf dem Gerät des Versicherten (siehe auch Flow-Beschreibungen in der [Wissensdatenbank] zur TI-Föderation).
VZD	
FHIR-VZD	Der VZD liefert das Ergebnis einer Suchanfrage vom PoPP-Service. Der PoPP-Service übergibt dem VZD-Suchparameter und liefert Datensätze zu gefundenen LEIs.

3 Übersicht Funktionsmerkmale und Schnittstellen

Apps mit unterschiedlichen fachlichen Ausrichtungen haben den Bedarf, einen Versorgungskontext zwischen einer versicherten Person und einer Leistungserbringerinstitution herzustellen.

Die Benutzerführung für die Erstellung eines Versorgungskontext zwischen einer versicherten Person und einer LEI wird dabei wesentlich davon geprägt, welchen Zweck die jeweilige Anbieter-App verfolgt. Dabei spielen folgende Faktoren eine Rolle:

- Welche Authentisierungsverfahren sollen durch das PoPP-Modul unterstützt werden (GesundheitsID und/oder "eGK-in-Fernversorgung")?
- Ist durch die Anbieter-App bereits festgelegt, welche Leistungserbringerinstitution (LEI) ein PoPP-Token benötigt oder muss die LEI über das PoPP-Modul ermittelt werden?
- Welche Verfahren zur Ermittlung der LEI werden unterstützt?

Die Benutzerführung in den jeweiligen Apps muss die unterschiedlichen Einflussfaktoren berücksichtigen.

Weitere Einflussfaktoren zur Ausgestaltung der Benutzerführung sind:

- Wenn das PoPP-Modul in ein ePA-Frontend des Versicherten (FdV) integriert ist, so kann für die Authentifizierung des Versicherten über seine GesundheitsID das Single-Sign-On (SSO) des ePA-FdV genutzt werden.
- Wenn die Anbieter-App, welche das PoPP-Modul benötigt, auf demselben Gerät installiert ist wie die App mit dem Authenticator-Modul für die Authentifizierung mit GesundheitsID (entweder standalone oder integriert in Krankenversicherungs-App), erfolgt die Erstellung des PoPP-Token bei einer Authentifizierung mit GesundheitsID über das PoPP-Modul mit App-App-Kommunikation (App-Sprung). Wenn die Anbieter-App, welche das PoPP-Modul benötigt, nicht auf dem Gerät installiert ist, auf dem auch die App mit dem Authenticator-Modul läuft, so erfolgt die Erstellung des PoPP-Token bei einer Authentifizierung mit GesundheitsID über das PoPP-Modul durch eine 2-Geräte-Kommunikation.
- Ist das PoPP-Modul in eine Drittanbieter-App integriert und das Gerät unterstützt NFC, so kann eine Authentifizierung der eGK ohne PIN-Eingabe ("eGK-in-Fernversorgung") genutzt werden. In diesem Fall gibt es keinen App-Sprung.
- Ist das Gerät des Versicherten nicht Near Field Communication (NFC) fähig oder kann kein Standardkartenleser verwendet werden, so ist die Authentifizierung einer elektronischer Gesundheitskarte ("eGK-in-Fernversorgung") nicht möglich.

Jede App mit integriertem PoPP-Modul muss auch ein ZETA Client [gemSpec_ZETA] implementieren. Der ZETA Client als Frontend-Komponente der Zero Trust-Architektur stellt die sichere Kommunikation zwischen der App mit dem PoPP-Modul auf einem mobilen Endgerät und dem PoPP-Service sicher. Über den ZETA Client wird im Rahmen der Initialisierung sowohl das Gerät auf dem die App läuft als auch die App selbst in der ZETA Guard Client-Registry registriert. Unter anderem wird durch die Attestierung der App auf einem bestimmten Gerät das Vertrauensverhältnis zwischen PoPP-Modul und PoPP-Service hergestellt.

Im Anhang des Dokuments sind die Abläufe der PoPP-Token Erstellung über eine Authentifizierung von Versicherten mit GesundheitsID und über die Authentifizierung einer eGK ohne PIN-Eingabe (eGK-in-Fernversorgung) detailliert beschrieben.

Die Abläufe berücksichtigen auch den Fall, dass ein PoPP-Token von einer Anwendung angefordert wird, welche kein PoPP-Modul integriert.

Dabei muss die Anwendung Voraussetzungen erfüllen, um die PoPP-Token-Erstellung zu initiieren:

- Auswahl der Krankenversicherung des Versicherten bzw. Ermittlung der ClientID des sektoralen IDP der Krankenversicherung in der TI-Föderation.
- Ermittlung der Telematik-ID der LEI, für die ein PoPP-Token erstellt werden soll.
- Ermittlung der WorkplaceID, für die ein PoPP-Token erstellt werden soll.
- Festlegung, ob zwingend eine Anmeldung mit "eGK-in-Fernversorgung" durchgeführt werden soll.
- Aufruf eines Request mit Telematik-ID und einer Rücksprung-URL (callbackURL) an die ClientID des sektoralen IDP der Krankenversicherung in der TI-Föderation um den Ablauf in das PoPP-Modul zu verlagern.

Die WorkplaceID dient einer LEI, welche ein PoPP-Token erhält, dieses einem konkreten Arbeitsplatz zuzuweisen. Die WorkplaceID kann durch die LEI auch wie eine Session-ID verwendet werden, um das letztendlich erzeugte PoPP-Token dann dem richtigen Versicherten zuordnen zu können. Dies kann bspw. im Online-Apotheken-Anwendungsfall notwendig werden. Die LEI muss dabei beachten, dass die in A_28629* beschriebenen Einschränkungen für die WorkplaceID entsprechend genauso gelten.

Die Abläufe der Authentifizierung von Versicherten mit GesundheitsID und der Authentifizierung einer eGK ohne PIN-Eingabe (eGK-in-Fernversorgung) unterscheiden sich in den Schritten der konkret ablaufenden Authentifizierung und sind darüber hinaus z.B. im Hinblick auf die Initialisierung des ZETA Client, der Ermittlung der Daten einer LEI über den VZD, der Zustimmung zur Nutzung der Daten und der Ergebnisverarbeitung der PoPP-Token-Erstellung identisch.

3.1 Authentifizierung mit GesundheitsID

Die Erstellung eines PoPP-Token wird bei Online Szenarien über ein in eine App integriertes PoPP-Modul gestartet. Für die Authentifizierung mit GesundheitsID gibt es hier drei unterschiedliche Umsetzungsvarianten:

1. PoPP-Modul und Authenticator-Modul des sektoralen IDP sind in die App der Krankenversicherung integriert, welche auch das ePA-FdV integriert.
2. PoPP-Modul und Authenticator-Modul des sektoralen IDP sind in einer eigenen Authenticator-App und der Online-Check-in mit GesundheitsID wird aus einer anderen App gestartet (App-Sprung).
3. PoPP-Modul ist in einer Drittanbieter-App und das Authenticator-Modul ist in einer App der Krankenversicherung integriert (App-Sprung).

Das Aktivitätsdiagramm "Allgemeiner Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID" im Anhang stellt den Ablauf der PoPP-Token-Generierung für den Fall dar, dass das PoPP-Modul und Authenticator-Modul in einer App integriert sind. Der Start des Online-Check-in erfolgt in einer Drittanbieter-App oder in der App der Krankenversicherung. Ein Versicherter wird über seine GesundheitsID authentifiziert. Die für die PoPP-Token-Erstellung notwendigen Daten, KVN-R des Versicherten und IK-Nummer der Krankenversicherung des Versicherten werden bei erfolgreicher Authentifizierung aus dem vom sektoralen IDP ausgestellten ID-Token entnommen.

Die Informationen zum eingesetzten Authentisierungsmittel und zum Vertrauensniveau, auf dem die Authentifizierung durchgeführt wurde, werden ebenfalls aus dem ID-Token entnommen und fließen als "proofMethod" in das PoPP-Token ein.

3.2 Authentifizierung einer eGK (in Fernversorgung)

Mit einer Authentifizierung eGK-in-Fernversorgung können Anwendungsfälle unterstützt werden, bei denen das Sicherheitsniveau einer eGK-Authentifizierung ausreichend ist. Für die Authentifizierung einer eGK-in-Fernversorgung muss der Nutzer eine eGK an sein Smartphone halten. Über die NFC-Schnittstelle wird die eGK durch den PoPP-Service ausgelesen. Der PoPP-Service prüft die ausgelesenen Daten.

Wählt der Versicherte als Authentisierungsmittel "eGK-in-Fernversorgung" oder wird dies durch den Anwendungsfall vorgegeben (z.B. Rezept einlösen durch Vertreter), so wird die eGK über die Komponenten des PoPP-Service authentifiziert.

Für das Auslesen der eGK kommuniziert der PoPP-Service über die ZETA-Komponenten mit dem PoPP-Modul und dieses direkt mit der eGK. Das PoPP-Modul sendet über die NFC-Schnittstelle des Versicherten-Gerätes die vom PoPP-Service erzeugten APDU-Kommandos an die eGK und empfängt von dieser die jeweiligen Antworten. Die Antworten der eGK werden vom PoPP-Modul unverändert über die ZETA-Komponenten an den PoPP-Service geschickt. Über diesen Weg werden CV-Zertifikat und X.509-Zertifikat ausgelesen. Außerdem wird eine vom PoPP-Service erzeugte Challenge durch die eGK mit dem CV-Zertifikat der eGK signiert.

Der PoPP-Service führt dann die notwendigen Prüfungen der Zertifikate sowie der signierten Challenge durch. Der PoPP-Service erstellt nach erfolgreicher Prüfung das PoPP-Token. Die Daten zur LEI (Telematik-ID und WorkplaceID) kommen in diesem Fall aus einem vorherigen Request vom PoPP-Modul. Die KVNR des Versicherten und die IK-Nummer des Kostenträgers werden aus dem X.509-Zertifikat ermittelt.

Durch das proofMethod Attribut im PoPP-Token wird einer verwendenden Fachanwendung vermittelt, dass zur Erstellung des PoPP-Token nur eine niederschwellige Authentifizierung durchgeführt wurde.

3.3 Schnittstellen des PoPP-Moduls

Die Abbildung "Schnittstellen vom und zum PoPP-Modul" zeigt die Schnittstellen von und zum PoPP-Modul. Die Schnittstellen zwischen Komponenten der Authenticator-App sind API-Schnittstellen. Zwischen den über das Internet erreichbaren Komponenten sind REST-Schnittstellen.

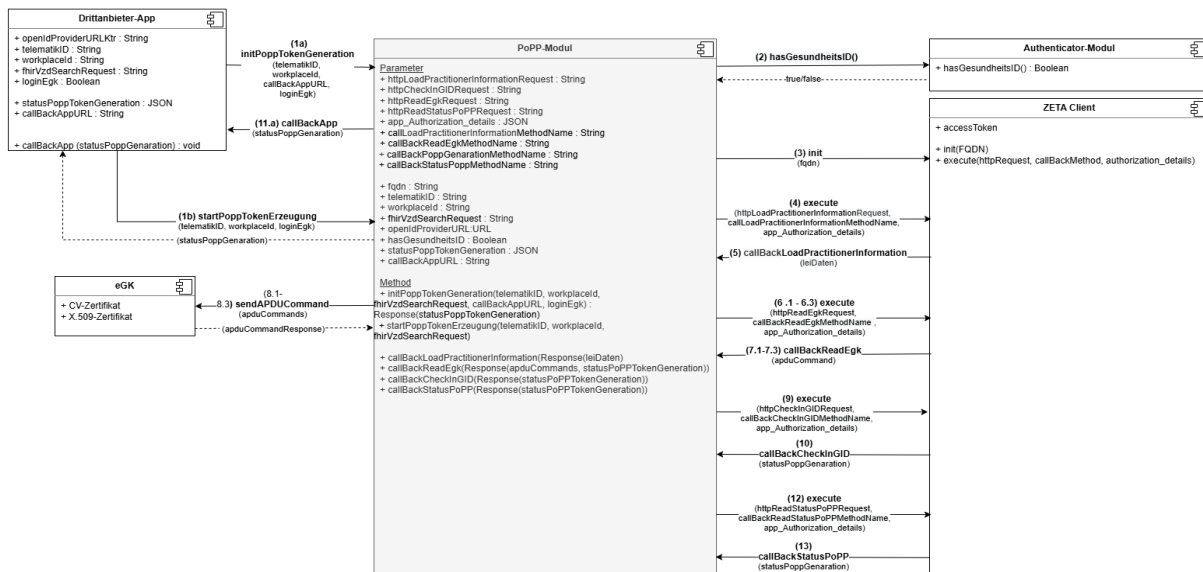


Abbildung 2: Schnittstellen vom und zum PoPP-Modul

3.3.1 initPopTokenGeneration()

Beschreibung:

Wird die PoPP-Token-Erstellung über eine Drittanbieter-App initiiert, so ruft diese die URL des sektoralen IDP der Krankenkasse des Versicherten mit den Parametern für das PoPP-Modul auf, um die Erstellung eines PoPP-Token für eine Telematik-ID zu initiieren.

Der Aufruf bewirkt das Öffnen der Authenticator-App mit implementierten PoPP-Modul. Aufgrund der übergebenen Parameter wird das PoPP-Modul aufgerufen.

Protokoll: HTTP

Parameter:

Parameter	Beschreibung	Typ	Wertebereich
telematikID	Telematik-ID der LEI, für die ein PoPP-Token generiert werden soll	String	Format gemäß Festlegungen [gemSpec_PKI] - "Aufbau der Telematik-ID"
workplaceID (optional)	Arbeitsplatz-Bezeichnung für einen konkreten Arbeitsplatz, an den der PoPP-Token geleitet werden soll	String	maximal 64 Zeichen, wobei folgende Zeichen nicht zulässig sind: \\ / : * ? " < >
fhirVzdSearchRequest	Ist die Telematik-ID nicht bekannt, kann in diesem Parameter ein Search-Request für das FHIR-VZD zur Ausführung	String	Der Search-Request muss ein am FHIR-VZD ausführbarer Request sein.

	übergeben werden.		
callBackAppURL	Rücksprung-URL, die vom PoPP-Modul aufgerufen wird, um in die Anbieter-App zurückzukehren	String	Der Parameter ist der String-Representation einer gültigen URL. Das Gerät muss so konfiguriert sein, dass der Aufruf der URL die Drittanbieter-App öffnet.
loginEgk (optional)	Der Parameter gibt an, ob der Login über die Authentifizierung einer eGK erfolgen muss (z.B. im Vertretungsfall)	Boolean	true, wenn der Login durch die Authentifizierung einer eGK erfolgen muss, sonst false. Ist der Parameter nicht vorhanden, so entspricht dies loginEgk= false;

Rückgabewerte:

Typ	Wertebereich
HTTP-Response	HTTP-OK, HTTP-<ErrorCode>

3.3.2 startPopTokenErzeugung()

Beschreibung:

Wird die PoPP-Token Erstellung über eine Drittanbieter-App initiiert, welche das PoPP-Modul und ZETA Client selbst implementieren, so ruft diese dazu eine API-Schnittstelle am PoPP-Modul auf.

Protokoll: API

Parameter:

Parameter	Beschreibung	Typ	Wertebereich
telematikID	Telematik-ID der LEI, für die ein PoPP-Token generiert werden soll	String	Format gemäß Festlegungen [gemSpec_PKI] - "Aufbau der Telematik-ID"
workplaceld (optional)	Arbeitsplatz-Bezeichnung für einen konkreten Arbeitsplatz, an den der PoPP-Token geleitet werden soll	String	maximal 64 Zeichen, wobei folgende Zeichen nicht zulässig sind: \\ / : * ? " < >
fhirVzdSearchRequest	Ist die Telematik-ID nicht bekannt, kann in diesem Parameter ein Search-Request für	String	Der Search-Request muss ein am FHIR-VZD ausführbarer Request sein.

	das FHIR-VZD zur Ausführung übergeben werden.		
loginEgk (optional)	Der Parameter gibt an, ob der Login über die Authentifizierung einer eGK erfolgen muss (z.B. im Vertretungsfall)	Boolean	true, wenn der Login durch die Authentifizierung einer eGK erfolgen muss, sonst false. Ist der Parameter nicht vorhanden, so entspricht dies loginEgk= false;

Rückgabewerte:

Typ	Wertebereich
Response-Objekt	HTTP-OK(statusPoppGeneration), HTTP-<ErrorCode>

3.3.3 hasGesundheitsID()

Beschreibung:

Die Schnittstelle dient der Abfrage an das Authenticator-Modul, ob eine GesundheitsID mit einer Identifikation des Versicherten auf dem Vertrauensniveau "gematik-ehealth-loa-high" eingerichtet wurde. Ist das nicht der Fall (Abfrage liefert "false"), so kann im weiteren Verlauf nur eine eGK-Authentifizierung über den PoPP-Service und nicht eine Versicherten-Authentifizierung über die sektoralen IDP erfolgen.

Protokoll: API

Parameter: keine

Rückgabewerte:

Typ	Wertebereich
Boolean	true, wenn eine GesundheitsID an das Authenticator-Modul gebunden ist, sonst false

3.3.4 init()

Beschreibung:

Der Aufruf initialisiert den ZETA Client.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
FQDN	String	Der Parameter ist die String-Representation des Fully Qualified Domain

		Name (FQDN) des PoPP-Service.
--	--	-------------------------------

Rückgabewerte: keine

3.3.5 execute(httpLoadPractitionerInformationRequest)

Beschreibung:

Der execute-Aufruf des PoPP-Modul am ZETA Client ist die Anweisung zur Durchführung eines HTTP-Request am PoPP-Service für die Suche der Daten zu einer LEI. Dafür wird als Parameter ein am FHIR-VZD ausführbarer Search-Request mit übergeben. Der Search-Request wurde entweder an der Schnittstelle zum PoPP-Modul (initPoppTokenGeneration oder startPoppTokenErzeugung) übergeben. Nach der ZETA-Initialisierung stellen die ZETA-Komponenten ein Client Access-Token aus und führen den HTTP-Request mit dem Access-Token am PoPP-Service aus. Nach Ermittlung der LEI-Daten über den FHIR-VZD wird das Suchergebnis als Response an ZETA zurückgegeben. Der ZETA Client ruft die callback-Methode mit dem Response-Objekt beim PoPP-Modul auf.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
HTTP-Request	String	httpLoadPractitionerInformationRequest Der Parameter ist die String-Representation des HTTP-Request, der an den PoPP-Service geschickt werden soll. Der HTTP-Request enthält die Suchkriterien für den VZD-Abruf als Query-Parameter
callbackMethod	String	callbackLoadPractitionerInformation Der Parameter ist die String-Representation der API-Methode, welche als Ergebnis der execute-Methode durch den ZETA Client beim PoPP-Modul aufzurufen ist.

Rückgabewerte: keine

3.3.6 callbackLoadPractitionerInformation

Beschreibung:

Das PoPP-Modul stellt die API-Methode bereit, die vom ZETA Client als Abschluss des ausgeführten execute-Auftrags zur Ermittlung der Daten einer LEI aufgerufen werden soll. Der ZETA Client übergibt die Antwort auf den HTTP-Request an das PoPP-Modul.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
LeiDaten	HTTP-Response	HTTP-OK(LeiDaten), HTTP-<ErrorCode>

Rückgabewerte: keiner

3.3.7 execute(httpReadEgkRequest)

Beschreibung:

Der Aufruf der execute-Methode mit dem Parameter `httpReadEgkRequest` wird nur bei der Authentifizierung einer eGK aufgerufen.

Der execute-Aufruf des PoPP-Modul am ZETA Client ist die Anweisung zur Durchführung eines HTTP-Request am PoPP-Service für die Prüfung einer eGK. Die ZETA-Komponenten führen mit dem Client-Access-Token aus den HTTP-Request am PoPP-Service aus. Der PoPP-Service erstellt einen APDU-Command, welches an die eGK verschickt werden soll. Der PoPP-Service gibt das APDU-Command in der Response an die ZETA-Komponenten zurück. Der ZETA Client ruft die `callBack`-Methode mit dem Response-Objekt beim PoPP-Modul auf.

Der Ablauf, Aufruf der `execute(httpReadEgkRequest)`-Methode und Antwort in der `callBackReadEgk()`-Methode wird so lange durchlaufen, bis alle vom PoPP-Service zu erstellenden APDU-Commands abgearbeitet sind. Die Antworten auf ein APDU-Command werden in den `app_authorization_details` des nächsten `execute()` Aufrufs an den PoPP-Service übertragen. Demzufolge sind keine Daten in den `app_authorization_details` im ersten Aufruf enthalten.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
HTTP-Request	String	<code>httpReadEgkRequest</code>
<code>callBackMethod</code>	String	<code>callBackReadEgk</code>
<code>app_authorization_details</code>	JSON	<pre>"app_authorization_details": [{ "type": "urn:ti:gematik:apdu:command", "apdu_command_response": "<APDU-Command-Response>" }]</pre> <p>(Die <code>app_authorization_details</code> enthalten Informationen für den PoPP-Service)</p>

Rückgabewerte: keine

3.3.8 callBackReadEgk - bei Authentifizierung einer eGK

Beschreibung:

Der `callBackReadEgk`-Methode wird vom ZETA Client nur bei der Authentifizierung einer eGK aufgerufen.

Das PoPP-Modul stellt die API-Methode bereit, die vom ZETA Client als Antwort auf ausgeführte execute-Aufträge zum Auslesen der Daten aus einer eGK aufgerufen werden soll. Der ZETA Client übergibt die Antwort auf den HTTP-Request an das PoPP-Modul. Das Response-Objekt enthält entweder den nächsten durchzuführenden APDU-Aufruf für die eGK oder einen Datensatz zum Status der PoPP-Token-Generierung, wenn die APDU-Command vollständig abgearbeitet ist und ein Status der PoPP-Token-Generierung vom PoPP-Service erstellt wurde.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
apduCommand statusPoppTokenGenerierung	HTTP- Response	HTTP-OK(apduCommand, statusPoppTokenGenerierung), HTTP- <ErrorCode>

Rückgabewerte: keiner

3.3.9 sendAPDUCommands()

Beschreibung:

Die sendAPDUCommand-Methode wird vom PoPP-Modul nur bei der Authentifizierung einer eGK aufgerufen.

Das PoPP-Modul sendet der eGK über die NFC-Schnittstelle die APDU-Command aus dem Response-Objekt, welche beim Aufruf der callReadEgk()-Methode übergeben wurde.

Die Antwort der eGK wird apdu_command_response in den app_authorization_details des nächsten execute(httpReadEgkRequest, callBackReadEgk, app_authorization_details) an den PoPP-Service übertragen.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
apduCommand	String	APDU-Command aus dem Response-Objekt der callReadEgk()-Methode

Rückgabewerte:

Typ	Wertebereich
APDU-Command Response	HTTP-OK(apduCommand, statusPoppTokenGenerierung), HTTP- <ErrorCode>

3.3.10 execute(httpCheckInGIDRequest)

Beschreibung:

Der Aufruf der execute-Methode mit dem Parameter httpCheckInGIDRequest wird nur bei der Authentifizierung mit GesundheitsID aufgerufen.

Der execute-Aufruf des PoPP-Modul am ZETA Client ist die Anweisung zur Durchführung eines HTTP-Request am PoPP-Service zur Erstellung eines PoPP-Token nach Authentifizierung des Versicherten über seine GesundheitsID. Die ZETA-Komponenten prüfen, ob bereits eine Authentifizierung durchgeführt wurde. Ist das nicht der Fall wird die Authentifizierung des Versicherten über den sektoralen IDP angestoßen. Die URL des sektoralen IDP wird in den app_authorization_details vom PoPP-Modul übergeben. Nach erfolgreicher Authentifizierung werden den ZETA-Komponenten die Daten des

Versicherten als ID-Token übergeben. Die ZETA-Komponenten führen nun den `httpCheckInGIDRequest` am PoPP-Service aus. Der PoPP-Service erstellt ein PoPP-Token. Den Status der PoPP-Token-Generierung liefert der PoPP-Service als Response den ZETA-Komponenten zurück. Der ZETA Client ruft die `callback`-Methode mit dem Response-Objekt beim PoPP-Modul auf.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
HTTP-Request	String	<code>httpCheckInGIDRequest</code>
<code>callbackMethod</code>	String	<code>callbackCheckInGID</code>
<code>app_authorization_details</code>	JSON	<pre>"app_authorization_details": [{ "type": "urn:ti:gematik:auth:provider", "auth_preferences": { "openIdProviderUrl": "<URL zum Service, welche die Authentifizierung durchführt>" } }]</pre> (Die <code>app_authorization_details</code> enthalten Informationen für den ZETA Client)

Rückgabewerte: keine

3.3.11 `callbackCheckInGID`

Beschreibung:

Die `callbackCheckInGID`-Methode wird vom ZETA Client nur bei der Authentifizierung mit GesundheitsID aufgerufen.

Das PoPP-Modul stellt die API-Methode bereit, die vom ZETA Client als Abschluss des ausgeführten `execute`-Auftrags aufgerufen werden soll. Der ZETA Client übergibt die Antwort auf den HTTP-Request an das PoPP-Modul.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
<code>statusPoppGenarationResponse</code>	HTTP-Response	HTTP-OK(<code>statusPoppGenaration</code>), HTTP- <code><ErrorCode></code>

Rückgabewerte: keiner

3.3.12 `execute(httpReadStatusPoPPRequest)`

Beschreibung:

Der Aufruf der execute-Methode mit dem Parameter `httpReadStatusPoPPRequest` kann vom PoPP-Modul aufgerufen werden um bei PoPP-Service den aktuelle Status zum Check-in abzurufen. Dieser Aufruf ist dann sinnvoll, wenn der ursprüngliche Check-in noch nicht vollständig abgeschlossen werden konnte, da die LEI nicht online war. In diesem Fall hat der Check-in-Prozess dem PoPP-Modul den Status "pending" zurückgeliefert (siehe A_28503*).

Der execute-Aufruf des PoPP-Modul am ZETA Client ist die Anweisung zur Durchführung eines HTTP-Request am PoPP-Service für den Abruf des aktuellen Status zur PoPP-Token Erstellung. Die ZETA-Komponenten führen mit dem Client Access-Token den HTTP-Request am PoPP-Service aus. Den Status der PoPP-Token-Generierung liefert der PoPP-Service als Response den ZETA-Komponenten zurück. Der ZETA Client ruft die `callback`-Methode mit dem Response-Objekt beim PoPP-Modul auf.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
HTTP-Request	String	<code>httpReadStatusPoPPRequest</code> Der Parameter ist die String-Representation des HTTP-Request, der an den PoPP-Service geschickt werden soll. Der HTTP-Request enthält die ID des im PoPP-Service angelegten PoPP-Datensatzes als Query-Parameter.
<code>callbackMethod</code>	String	<code>callbackReadStatusPoPP</code>

Rückgabewerte: keine

3.3.13 `callbackReadStatusPoPP`

Beschreibung:

Der `callReadStatus`-Methode wird vom ZETA Client als Ergebnis der Status-Abfrage zur PoPP-Token-Generierung aufgerufen.

Das PoPP-Modul stellt die API-Methode bereit, die vom ZETA Client als Antwort auf ausgeführte execute-Aufträge zum Auslesen der Daten aus einer eGK aufgerufen werden soll. Der ZETA Client übergibt die Antwort auf den HTTP-Request an das PoPP-Modul. Das Response-Objekt enthält einen Datensatz zum Status der PoPP-Token-Generierung.

Protokoll: API

Parameter:

Parameter	Typ	Wertebereich
<code>apduCommand</code> <code>statusPoppTokenGenerierung</code>	HTTP-Response	HTTP-OK(<code>apduCommand</code> , <code>statusPoppTokenGenerierung</code>), HTTP- <code><ErrorCode></code>

Rückgabewerte: keiner

3.3.14 callbackApp

Beschreibung:

Ist die PoPP-Token-Erstellung über eine Drittanbieter-App über App-App-Kommunikation initiiert, ruft das PoPP-Modul die im `initPoppTokenGeneration()`-Request übergebene `callbackAppURL` mit der HTTP-Response aus dem `callbackReadEgk()`-Request bzw. `callbackCheckInGID()`-Request auf. Das Gerät muss so konfiguriert sein, dass der Aufruf zum Öffnen der Drittanbieter-App führt (deeplink / universal link).

Protokoll: HTTP

Parameter:

Parameter	Typ	Wertebereich
<code>statusPoppGenarationResponse</code>	HTTP-Response	HTTP-OK(<code>statusPoppGenaration</code>), HTTP-<ErrorCode>

Rückgabewerte:

Typ	Wertebereich
HTTP-Response	HTTP-OK, HTTP-<ErrorCode>

3.4 PoPP-Modul in Drittanbieter-APP

Das spezielle Verfahren der Authentifizierung einer eGK wird über das Zusammenwirken von PoPP-Modul, ZETA-Komponenten und PoPP-Service ohne Beteiligung weiterer Komponenten abgebildet. PoPP-Token, die auf diesem niedrigen Vertrauensniveau ausgestellt werden, sind nicht für alle Fachverfahren zulässig. Bei der Authentifizierung Versicherter mit ihrer GesundheitsID werden noch die Komponenten des sektoralen IDP der Krankenversicherung benötigt. Die Authentifizierung der Versicherten über die sektoralen IDPs erfolgt hier auf hohem Vertrauensniveau (siehe [gemSpec_IDP_Sek] - "gematik-ehealth-loa-high" und "gematik-ehealth-loa-substantial"). Die im PoPP-Token enthaltenen Versichertendaten gehören eindeutig zum Versicherten, welcher sich authentifiziert hat.

Alternativ kann PoPP-Modul und ZETA Client auch in der Drittanbieter-App implementiert werden ("Detailsicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in Drittanbieter App").

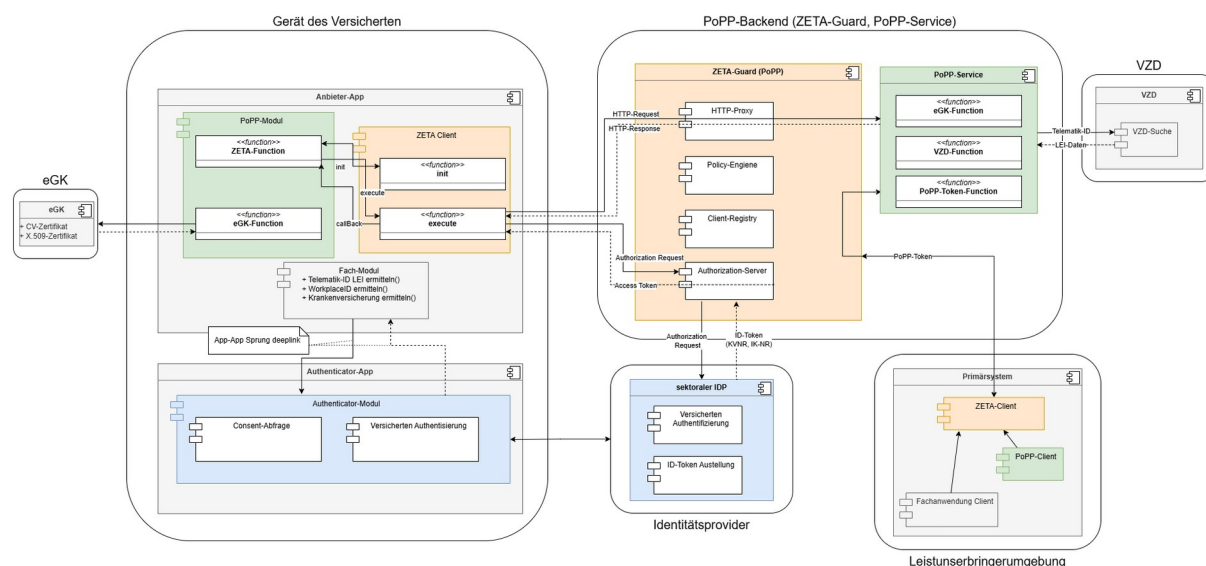


Abbildung 3: Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in Drittanbieter App

Da für die Authentifizierung einer eGK nur PoPP-Modul, ZETA-Komponenten und PoPP-Service benötigt werden, ist in diesem Fall kein App-Sprung notwendig. Der detaillierte Ablauf ist in [[Kapitel "Detaillierter Ablauf der PoPP-Token Generierung durch Authentifizierung der eGK über den PoPP-Service mit PoPP-Modul in einer Drittanbieter-App"](#)] dargestellt.

Die PoPP-Token Erstellung auf einem hohen Vertrauensniveau über Authentifizierung des Versicherten mit GesundheitsID muss allerdings immer über das Authenticator-Modul in der App der Krankenversicherung erfolgen.

4 Übergreifende Festlegungen

4.1 Datenschutz und Informationssicherheit

A_27220 -PoPP-Modul (ZETA Client) - TLS-Verbindungsaufbau

Das PoPP-Modul bzw. der ZETA Client MUSS beim TLS-Verbindungsaufbau zu allen Endpunkten des PoPP-Service das TLS-Server-Zertifikat des PoPP-Service validieren und dabei prüfen, dass:

- das Zertifikat von einem Herausgeber, der Mitglied im [CAB Forum] ist, ausgestellt wurde,
- das Ausstellerzertifikat aktuell gültig ist (bspw. Prüfung gegen den Truststore des Betriebssystems)
- der aufgerufene hostname im Zertifikat aufgeführt ist (Hostnamevalidierung),
- das Zertifikat zeitlich gültig ist,
- der OCSP Status "good" ist (OCSP-Stapling wird serverseitig unterstützt),
- die Signatur der OCSP-Response auf den selben Herausgeber rückführbar ist, wie das TLS-Zertifikat

und nur im Falle einer erfolgreichen, positiven Prüfung die TLS-Verbindung aufbauen.

[<=]

Hinweis: Das in gemSpec_ZETA#A_25340 geforderte Einbinden des TI-Vertrauensraums ist nicht erforderlich.*

A_27621 -PoPP-Modul - Einwilligung des Versicherten in die Datennutzung

Das PoPP-Modul MUSS, im Zuge der Verifikation der Versichertenidentität sicherstellen, dass der Versicherte, bevor seine KVNR an den PoPP-Service übermittelt wird, seine Einwilligung (Consent) zur Nutzung seiner KVNR durch die LEI erteilt.[<=]

Hinweis: Die Einwilligungsabfrage kann mit der Anzeige der LEI-Informationen (vgl. A_28488) in einem Schritt passieren.*

A_28488 -PoPP-Modul - Anzeige LEI oder DiGA Informationen aus VZD

Das PoPP-Modul MUSS im Falle von per QR-Code-Scan ermittelten oder bei externem App-Abruf übergebenen Telematik-ID oder zu erfassten Suchkriterien die dazugehörigen LEIs oder DiGAs oder Kostenträger über Volltextsuche im FHIR-Verzeichnisdienst der TI finden, dem Nutzer die Klartextinformationen zur LEI bzw. DiGA oder Kostenträger anzeigen und ihn den Check-in für diese Institution bestätigen lassen.

Das PoPP-Modul MUSS dabei die anzuzeigenden Informationen direkt an die entsprechende Schnittstelle des Betriebssystems übergeben also für die Anzeige gerade nicht interne Schnittstellen der umgebenden App nutzen.

Das PoPP-Modul MUSS sicherstellen, dass ausschließlich Einrichtungen (LEIs) und DiGAs und Kostenträger angezeigt werden - also insbesondere keine weiteren Organisationen und Personen. Dabei MUSS das PoPP-Modul:

- aus dem "OrganizationDirectory" das Feld "type"/"coding"/"display" anzeigen (fachliche Rolle / Art der Institution),

- aus dem "OrganizationDirectory" das Feld "name" anzeigen (Anzeigename der Institution),
- aus dem "OrganizationDirectory" das Feld "alias" anzeigen, sofern es vorhanden ist und sich vom Feld "name" unterscheidet,
- aus dem "LocationDirectory" aus dem Feld "address" die Adresse ermitteln und anzeigen für den Fall einer LEI.

Erhält das PoPP-Modul bei der VZD-Suche keinen Treffer zur Telematik-ID oder zu den erfassten Suchkriterien, MUSS es den Check-in-Vorgang abbrechen, dem Versicherten einen verständlichen Hinweis zum Fehler geben und ihn dabei auffordern, sich an die Institution zu wenden, für die er den Check-in durchführen wollte.【<=】

Hinweis: Über die Anzeige der LEI-Daten soll der Versicherte in die Lage versetzt werden, eine ggf. falsche Telematik-ID (gescannte oder übergebene Daten sind fehlerhaft oder manipuliert) zu erkennen, indem er die angezeigten Klartext-LEI-Informationen mit den erwarteten Informationen (LEI bei der er den Check-in durchführen möchte) abgleichen kann, so dass nur für die korrekte LEI die Einwilligung erteilt wird (vgl. A_27621).*

A_28629 -PoPP-Modul - Prüfen der WorkplaceID

DasPoPP-Modul MUSS sicherstellen, dass die WorkplaceID aus maximal 64 alphanumerischen Zeichen besteht, wobei die folgenden Zeichen nicht zulässig sind: \ / : * ? " < > | 【<=】

5 Funktionsmerkmale

5.1 Allgemeine funktionale Anforderungen PoPP-Modul

A_28675 -PoPP-Modul - unterstützte Authentisierungsverfahren

Das PoPP-Modul MUSS mindestens

- die Authentifizierung des Versicherten mit dessen GesundheitsID
- die Authentifizierung einer eGK (eGK-in-Fernversorgung)

unterstützen. [≤]

A_28511 -PoPP-Modul - Anzeige einer Historie der erstellten PoPP-Token

Das PoPP-Modul MUSS Versicherten eine Funktion bereitstellen, über die dieser die Historie der erstellten PoPP-Token einsehen kann. Die Historie MUSS Name und Anschrift der LEI enthalten. [≤]

Hinweis: Die Anzahl der angezeigten Einträge in der Historie sollte konfigurierbar sein. Empfohlen wird die Darstellung der letzten zehn Ereignisse.

A_28513 -PoPP-Modul - Unterstützung zum Einrichten einer GesundheitsID

Das PoPP-Modul MUSS dem Versicherten die Funktion zum Einrichten der GesundheitsID anbieten, wenn diese noch nicht eingerichtet ist. [≤]

5.2 Auswahl der LEI/ Bestimmen der Telematik-ID

5.2.1 Favoriten

A_28512 -PoPP-Modul - Anlage, Anzeige und Auswahl von Favoriten

Das PoPP-Modul MUSS die Funktion "Favoriten" anbieten, und diese wie folgt umsetzen:

- Per QR-Code gescannte, per bei App-Abruf übergebene und per bei VZD-Suche gefundene LEI muss der Nutzer nach Anzeige der VZD-Daten (vgl. A_28488*) als Favorit markieren können.
- Zu Favoriten wird der Name aus dem VZD-Eintrag "OrganizationDirectory"/"name" und die Telematik-ID gespeichert.
- Nutzer können zusätzlich einen alternativen Anzeigenamen des Favoriten, wie er in der Favoritenliste angezeigt wird, angeben und diesen ändern.
- Wählt ein Versicherter einen gesetzten Favoriten für eine erneute Erstellung eines PoPP-Token aus, so muss dafür kein erneuter VZD-Abruf durchgeführt werden.

[≤]

5.2.2 Verzeichnisdienstsuche

A_28515 -PoPP-Modul - Unterstützung der Suche nach LEIs und DiGAs

Das PoPP-Modul MUSS den Versicherten eine Funktion anbieten, über die dieser eine Suche nach LEIs und DiGAs im FHIR-VZD der TI durchführen kann. [≤]

A_28514 -PoPP-Modul - Erstellen eines HTTP-Request zur VZD-Abfrage

Das PoPP-Modul MUSS einen HTTP-Request (httpLoadPractitionerInformationRequest) zum Auslösen einer Abfrage des VZD durch den PoPP-Service erstellen.

Der HTTP-Request MUSS als Query-Parameter einen ausführbaren Search-Request für einen Aufruf des FHIR-VZD beinhalten.

Hinweis: Die Schnittstelle ist in [I_PoPP_Load_Practitioner_Information.yaml] definiert. [≤]

A_28600 -Bereitstellung callBack-Methode für Abschluss der VZD-Abfrage

Das PoPP-Modul MUSS in seinem API eine callBack-Methoden callBackLoadPractitionerInformation für den Abschluss der FHIR-VZD-Abfrage zur Verfügung stellen, die vom ZETA Client entsprechend [gemSpec_ZETA] aufgerufen werden kann. Die callBack-Methode MUSS als einzigen Parameter ein HTTP-Response Objekt beinhalten.

Das PoPP-Modul MUSS das Suchergebniss der VZD-Suche aus dem Body der Response Objektes extrahieren und hinsichtlich A_28488* auswerten. [≤]

5.2.3 QR-Code

Wenn die Auswahl einer LEI mittels des Auslesens eines QR-Codes der LEI geschieht, gelten folgende Festlegungen zum Lesen und Prüfen des QR-Codes.

Der Inhalt des statischen QR-Codes wird als JSON-Struktur gemäß RFC 8259 repräsentiert. Der verwendete Zeichensatz ist UTF-8 nach RFC 3629. Die Erstellung des QR-Codes erfolgt gemäß ISO/IEC 18004. Die Erstellung erfolgt im Primärsystem (siehe [gemILF_PoPP_Client] und verwendet die Formatvorgaben in Tabelle [Tab_PoPP_Modul_Payload_stat_QRCode].

Tabelle 2: Tab_PoPP_Modul_Payload_stat_QRCode]: Payload QR-Code

Name	Wert	Beispiel
tid	Telematik-ID der LEI (String, Format gemäß Festlegungen [gemSpec_PKI] - " Aufbau der Telematik-ID ")	"1-234567890"
typ	fester Wert: popp-checkin	"popp-checkin"
wpid	WorkplaceID; String entsprechend A_28629* - optional	"REZEPTION-1"

Hinweis zu wpid: Die Festlegung ob eine WorkplaceID verwendet wird, und wie diese gebildet ist, erfolgt in der Verantwortung der LEI.

A_27595 -PoPP-Modul - QR-Code, Validierung und Warnung des Nutzers

Das PoPP-Modul MUSS das Scannen von QR-Codes als Eingabe für die Telematik-ID und ggf. Arbeitsplatzinformationen umsetzen, und dabei sicherstellen, dass:

- für das Scannen direkt die Schnittstellen des Betriebssystems verwendet werden, also gerade nicht interne Schnittstellen der umgebenden App verwendet werden,

- gescannte Informationen nur nach erfolgreicher sicherheitstechnischer Validierung verwendet werden, wobei das PoPP-Modul prüfen MUSS, ob die Inhalte dem erwarteten Schema (vgl. Tabelle [Tab_PoPP_Modul_Payload_stat_QRCode]) entsprechen und keine unerwarteten Informationen enthalten,
- insbesondere keine Links automatisch aufgelöst werden, also der Nutzer nicht automatisch weitergeleitet wird, sollte der gescannte Code eine URL enthalten, und
- der Nutzer eindeutig gewarnt wird, wenn die gescannten Informationen nicht erfolgreich sicherheitstechnisch validiert werden konnten, wobei der Nutzer an das LEI-Personal verwiesen werden MUSS.

[<=]

5.3 Auslösen der Erstellung eines PoPP-Token

5.3.1 Anforderungen PoPP-Modul bei Authentisierung eines Versicherten mit GesundheitsID

Die Abläufe der Erzeugung eines PoPP-Token über die Authentifizierung des Nutzers mit GesundheitsID sind in Kapitel [[Detaillierter Ablauf der PoPP-Token Generierung durch Authentifizierung Versicherter über ihre GesundheitsID](#)] dargestellt.

Das PoPP-Modul einer Anwendung löst durch Aufrufen der ZETA Client Schnittstelle die Erzeugung eines PoPP-Token für eine bestimmte LEI aus. Dabei prüft der ZETA Client, ob in der laufenden Session bereits eine Nutzer-Authentifizierung über die GesundheitsID durchgeführt wurde. Ist das nicht der Fall, so stößt der ZETA Client die Authentifizierung des Versicherten an. Der Ablauf der Authentifizierung wird durch die ZETA-Komponenten und dem sektoralen IDP mit seinem Authenticator-Modul abgebildet.

Nach erfolgreicher Authentifizierung des Versicherten wird vom ZETA Client der eigentliche HTTP-Request (siehe A_28501*) an den PoPP-Service erstellt. Dabei wird der vom PoPP-Modul erstellte HTTP-Request vom ZETA Guard HTTP-Proxy um

- Informationen zum Client aus der Client-Registrierung und
- KVN- und IK-Nummer der Krankenversicherung aus dem ID-Token der Versicherten Authentifizierung

erweitert, bevor der eigentliche Request gegen den PoPP-Service erfolgt.

Bevor der Ablauf zur Authentifizierung über die GesundheitsID erfolgen kann, muss das PoPP-Modul prüfen, ob überhaupt eine GesundheitsID an das Authenticator-Modul gebunden ist.

A_28509 -PoPP-Modul - Aufruf des Authenticator-Modul zur Prüfung, ob eine GesundheitsID eingerichtet ist

Das PoPP-Modul MUSS, wenn die Versicherten-Authentifizierung über die GesundheitsID erfolgen soll, beim integrierten Authenticator-Modul erfragen, ob bereits eine GesundheitsID eingerichtet ist. Ist das nicht der Fall, so darf dem Versicherten eine Authentifizierung über GesundheitsID nicht angeboten werden; Versicherte können ausschließlich "eGK-in-Fernversorgung" nutzen.[<=]

A_27605 -PoPP-Modul - Nutzerauthentifizierung mit SSO im ePA-FdV

Ist ein PoPP-Modul in eine ePA-FdV integriert, so KANN für die Nutzerauthentifizierung das SSO gemäß [gemSpec_IDP_Sek] verwendet werden.[<=]

Hinweis: Da PoPP-Modul, ZETA Client und Authenticator-Modul Funktionen ein und derselben App sind, handelt es sich immer um die 1-App-Strategie, wenn diese Komponenten des ePA-FdV sind.

A_28501 -PoPP-Modul - Erstellen eines HTTP-Request zum Auslösen der Erstellung eines PoPP-Token mit GesundheitsID

Das PoPP-Modul MUSS einen HTTP-POST-Request zum Auslösen der Erstellung eines PoPP-Token erstellen. Dabei MÜSSEN:

- die Telematik-ID der LEI und
- die WorkplaceID der LEI (wenn vorhanden)

als Parameter übergeben werden.【<=】

Hinweis: Die Schnittstelle ist in [I_PoPP_CheckIn_HID.yaml] definiert.

A_28502 -PoPP-Modul - Erstellen der "app_authorization_details" zum Auslösen der Erstellung eines PoPP-Token mit GesundheitsID

Das PoPP-Modul MUSS authorization_details zur Versicherten-Authentifizierung gemäß [RFC9396] nach folgendem Muster erstellen:

```
"app_authorization_details": [{  
  "type": "urn:ti:gematik:auth:provider",  
  "auth_preferences": {  
    "openIdProviderUrl": "<URL zum Service, welche die Authentifizierung  
durchführt>",  
  }  
}]【<=】
```

【<=】

A_28503 -PoPP-Modul - Bereitstellung callBack-Methode für Abschluss der Erstellung eines PoPP-Token mit GesundheitsID

Das PoPP-Modul MUSS in seinem API eine callBack-Methoden callBackCheckInGID für den Abschluss der PoPP-Token-Erstellung zur Verfügung stellen, die vom ZETA Client entsprechend [gemSpec_ZETA] aufgerufen werden kann. Die callBack-Methode MUSS als einzigen Parameter ein HTTP-Response Objekt beinhalten.【<=】

Hinweis: Das Objekt zum Status der PoPP-Generierung enthält die Information, ob die Erstellung des PoPP-Token erfolgreich war.

- *success* - PoPP-Token wurde erzeugt und der LEI zugestellt
- *pending* - PoPP-Token wurde erzeugt und noch nicht zugestellt, da die LEI nicht online ist. Die Zustellung wird weiter versucht.
- *cancelled* - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen

Neben den Statusinformationen enthält das HTTP-Response-Objekt Detailinformationen zum Status oder zum aufgetretenen Fehler als "message". Die aufrufende Anwendung kann die Informationen verwenden, um z.B. den Nutzer zu informieren oder andere Anwendungsfälle anzubieten. Zur Identifikation des im PoPP-Service angelegten Datensatzes ist die ID des PoPP-Datensatz ebenfalls im vom PoPP-Service erstellten JSON-Objekt enthalten.

5.3.2 Anforderungen PoPP-Modul für eGK-in-Fernversorgung

Die Abläufe für die Erstellung eines PoPP-Token mit dem Authentisierungsmittel "eGK" sind in Kapitel "Authentifizierung mit "eGK-in-Fernversorgung" dargestellt.

Die Authentifizierung einer "eGK-in-Fernversorgung" erfolgt durch den PoPP-Service. Dieser erstellt eine Challenge, welche mit dem privaten Schlüssel des CV-Zertifikats der eGK zu signieren ist.

Der PoPP-Service erstellt APDU-Commands zur Signatur der Challenge sowie zum Auslesen der eGK und sendet diese sequentiell an das PoPP-Modul. Das PoPP-Modul sendet die APDU-Commands ungeändert an die eGK und deren Antwort zurück an den PoPP-Service.

Der PoPP-Service prüft die signierten Challenge und die von der eGK gelesenen Zertifikate (CV-Zertifikat, X.509-Zertifikat). Anschließend wird überprüft, ob CV-Zertifikat und X.509-Zertifikat zu ein und derselben eGK gehören. Zu diesem Zweck ruft der PoPP-Service die check-Schnittstelle der Hash-DB auf (siehe [gemSpec_PoPP_Service]).

Wird durch die eGK-Hash-DB die Zusammengehörigkeit der Zertifikate bestätigt, können die Versichertendaten durch den PoPP-Service aus dem X.509-Zertifikat extrahiert und für die Generierung des PoPP-Token verwendet werden.

Die Kommunikation zwischen PoPP-Modul und PoPP-Service läuft über die ZETA-Komponenten.

A_28518 -PoPP-Modul - Erstellen eines HTTP-Request zum Auslösen der Erstellung eines PoPP-Token mit eGK

Das PoPP-Modul MUSS einen HTTP-POST-Request zum Auslösen der Erstellung eines PoPP-Token erstellen. Dabei MÜSSEN

- die Telematik-ID der LEI und
- die WorkplaceID der LEI (wenn vorhanden)

als Parameter übergeben werden.【<=】

Hinweis: Die Schnittstelle ist in [I_PoPP_CheckIn_eGK_Verification.yaml] definiert.

A_28519 -PoPP-Modul - Erstellen der "app_authorization_details" zum Auslösen der Erstellung eines PoPP-Token mit eGK

Das PoPP-Modul MUSS authorization_details zur Versicherten Authentifizierung gemäß [RFC9396] nach folgendem Muster erstellen:

```
"app_authorization_details": [{  
  "type": "urn:ti:gematik:apdu:command",  
  "apdu_command_response": "<APDU-Command-Response>"  
}] .【<=】
```

A_28517 -PoPP-Modul - Bereitstellung callback-Methode für Abschluss der Erstellung eines PoPP-Token mit eGK

Das PoPP-Modul MUSS in seinem API eine callback-Methode callbackReadEgk für die Durchführung und den Abschluss des Auslesens der eGK, der Prüfung der eGK-Daten und der PoPP-Token-Erstellung zur Verfügung stellen, die vom ZETA Client entsprechend [gemSpec_ZETA] aufgerufen werden kann. Die callback-Methode MUSS als einzigen Parameter ein HTTP-Response Objekt beinhalten.

Das PoPP-Modul MUSS den Body des Response Objekts auslesen und nach folgenden Regeln interpretieren:

- Das Response Objekt enthält ein APDU-Command (apduCommand):
 - a. Das PoPP-Modul generiert eine Anfrage an die eGK und sendet dieser dasapduCommand.

- b. Das PoPP-Modul nimmt die Antwort der eGK entgegen und generiert app_authorization_details gemäß A_28519*
- c. Das PoPP-Modul ruft die execute()-Methode gemäß A_28619* auf
- Das Response Objekt enthält ein Objekt mit dem Status der PoPP-Generierung(statusPoPPGenaration):
- d. Das PoPP-Modul interpretiert den Inhalt des Objektes.

[<=]

Hinweis: Das Objekt zum Status der PoPP-Generierung enthält die Information, ob die Erstellung des PoPP-Token erfolgreich war.

- *success - PoPP-Token wurde erzeugt und der LEI zugestellt*
- *pending - PoPP-Token wurde erzeugt und noch nicht zugestellt, da die LEI nicht online ist. Die Zustellung wird weiter versucht.*
- *cancelled - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen*

Neben den Statusinformationen enthält das HTTP-Response Objekt Detailinformationen zum Status oder zum aufgetretenen Fehler als "message". Die aufrufende Anwendung kann die Informationen verwenden, um z.B. den Nutzer zu informieren oder andere Anwendungsfälle anzubieten.

5.3.3 Kommunikation PoPP-Modul mit ZETA Client

ZETA Client ist die Zero Trust Komponente für ein FdV, welches in die Krankenversicherungs-App bzw. Authenticator-App integriert ist. Der ZETA Client überträgt bei der Initialisierung der Anwendung die Geräte- und App-Informationen an den ZETA Guard.

Im laufenden Prozess kapselt der ZETA Client die gesamte Kommunikation zwischen der Anwendung auf dem Gerät des Versicherten und den Komponenten im Backend, u.a. auch zwischen PoPP-Modul und PoPP-Service. Der ZETA Client steuert die Client-Authentisierung am ZETA Guard Authorization-Server und unterstützt die Nutzerauthentifizierung mit GesundheitsID.

In [API ZETA Client] ist die API des ZETA Client und an dessen Verwendung spezifiziert. Das PoPP-Modul muss die bereitgestellten API-Schnittstellen des ZETA Client nutzen.

A_28507 -PoPP-Modul - Aufruf der Initialisierungs-Methode des ZETA Client

Das PoPP-Modul MUSS nach dem Start der Anwendung die Initialisierung des ZETA Client durch einen Methodenaufruf des ZETA Client gemäß [API ZETA Client] durchführen.**[<=]**

A_28618 -PoPP-Modul - Aufruf des ZETA Client für die FHIR-VZD-Anfrage

Das PoPP-Modul MUSS für eine FHIR-VZD-Anfrage die API des ZETA Client gemäß [API ZETA Client] aufrufen. Beim Aufruf sind folgende Parameter zu übergeben:

- der nach A_28514* erzeugte HTTP-Request zur FHIR-VZD-Anfrage,
- der Name der callBack-Methode, welche der ZETA Client nach Abschluss der Bearbeitung am PoPP-Modul aufruft (A_28600*).

[<=]

A_28619 -PoPP-Modul - Aufruf des ZETA Client zum Auslösen der Erstellung eines PoPP-Token mit eGK

Das PoPP-Modul MUSS, wenn die Erstellung eines PoPP-Token mit der Authentifizierung einer eGK ausgelöst werden soll, die API des ZETA Client gemäß [API ZETA Client] aufrufen. Beim Aufruf sind folgende Parameter zu übergeben:

- der nach A_28518* erzeugte HTTP-Request zur eGK-Prüfung,
- der Name der callBack-Methode, welche der ZETA Client nach Abschluss der Bearbeitung am PoPP-Modul aufruft (A_28517*),
- die nach A_28519* erzeugten app_authorization_details.

[<=]

A_28508 -PoPP-Modul - Aufruf des ZETA Client zum Auslösen der Erstellung eines PoPP-Token mit GesundheitsID

Das PoPP-Modul MUSS, wenn die Erstellung eines PoPP-Token mit der Versicherten Authentifizierung über die GesundheitsID erfolgen soll, die API des ZETA Client gemäß [API ZETA Client] aufrufen. Beim Aufruf sind folgende Parameter zu übergeben:

- der nach A_28501* erzeugte HTTP-Request zum Auslösen der PoPP-Token Erzeugung
- der Name der callBack-Methode, welche der ZETA Client nach Abschluss der Bearbeitung am PoPP-Modul aufruft (A_28503*)
- die nach A_28502* erzeugten app_authorization_details

[<=]

5.3.4 Kommunikation PoPP-Modul mit Drittanbieter-Apps

Drittanbieter-Apps, welche kein PoPP-Modul implementieren, können die Erstellung eines PoPP-Token über das PoPP-Modul in der App der Krankenversicherung anfordern. Dafür stellt das PoPP-Modul eine HTTP-Schnittstelle bereit, welche von der Drittanbieter-App aufgerufen werden kann.

Das Ergebnis der PoPP-Token-Erzeugung übermittelt das PoPP-Modul, indem es eine HTTP-Schnittstelle aufruft, welche die Drittanbieter-App zur Verfügung stellen muss (Ablaufbeschreibung siehe Anhang: [Ablaufdiagramme](#) und [Flowdiagramme](#)) .

A_28576 -PoPP-Modul - Bereitstellung init-Methode für die Initialisierung der Erstellung eines PoPP-Token durch Drittanbieter-Apps

Das PoPP-Modul MUSS eine HTTP-Schnittstelle anbieten, über die Drittanbieter-Apps die Erstellung eines PoPP-Token initiieren können. Die HTTP-Schnittstelle muss:

- als FQDN die Client-ID des sektoralen IDP der Krankenkasse haben, dessen Authenticator-Modul auf dem Gerät installiert ist.
- Folgende Parameter umfassen:
 - Telematik-ID (mandatory),
 - WorkplaceID (optional),
 - FHIR-VZD Search Request (optional)
 - callBackMethod (mandatory),
 - loginEgk (optional).

[<=]

A_28510 -PoPP-Modul - Erzeugung und Versand eines HTTP-Request an callBackURL einer Drittanbieter-App

Das PoPP-Modul MUSS, wenn es über eine Drittanbieter-App aufgerufen wurde, das Ergebnis der PoPP-Token-Erzeugung an die Drittanbieter-App übergeben, indem das PoPP-Modul die callback-URL der Drittanbieter-App aufruft. Das vom ZETA Client erhaltene Response-Objekt muss als Parameter beim Aufruf der callback-URL übergeben werden.【<=】

Für Drittanbieter-Apps, welche ein PoPP-Modul und der ZETA Client implementieren, stellt das PoPP-Modul eine API-Schnittstelle bereit. Drittanbieter-Apps können die Erstellung eines PoPP-Token über die API-Schnittstelle des PoPP-Modul anfordern und erhalten als Ergebnis den Status zur PoPP-Token-Erstellung.

A_28645 -PoPP-Modul - Bereitstellung einer Schnittstelle für die Initialisierung der Erstellung eines PoPP-Token durch Drittanbieter-Apps

Das PoPP-Modul MUSS eine Schnittstelle anbieten, über die Drittanbieter-Apps, welche das PoPP-Modul integrieren, die Erstellung eines PoPP-Token initiieren können. Das PoPP-Modul MUSS an der Schnittstelle folgende Parameter entgegennehmen:

- Telematik-ID (mandatory),
- WorkplaceID (optional),
- FHIR-VZD Search Request (optional)
- loginEgk (optional).

Das PoPP-Modul MUSS der Drittanbieter-App das vom ZETA Client erhaltene Response-Objekt als Ergebnis des Aufrufs zurückgeben.【<=】

5.4 Verarbeitung des Ergebnisses der PoPP-Token-Erzeugung

A_28516 -PoPP-Modul - Darstellung des Ergebnis der PoPP-Token-Erzeugung

Das PoPP-Modul MUSS dem Versicherten das Ergebnis der PoPP-Token-Erzeugung in einer für ihn verständlichen Sprache darstellen.【<=】

Hinweis: Bei erfolgreicher PoPP-Token-Erstellung kann z.B. folgende Nachricht angezeigt werden "Berechtigung von <LEI-Name> erfolgreich.". Im Fehlerfall könnte die Information lauten: "Die Berechtigung von <LEI-Name> wurde aufgrund eines Fehlers <Fehlerinformation> abgebrochen".

A_28677 -PoPP-Modul - Abfrage des Status der Erstellung eines PoPP-Token

Das PoPP-Modul KANN einen HTTP-GET-Request erstellen, um den Status der Erstellung eines PoPP-Token beim PoPP-Service abzurufen. Die ID des angelegten PoPP-Datensatzes muss als Query-Parameter im HTTP-GET Request übergeben werden.【<=】

Hinweis: Die ID des PoPP-Datensatzes ist Teil der Ergebnisinformationen der vom PoPP-Modul ausgelösten PoPP-Token-Erstellung. Konnte das PoPP-Datensatz erstellt aber ein PoPP-Token der LEI noch nicht zugestellt werden, ist der PoPP-Status "pending". In diesem Fall kann das PoPP-Modul über den HTTP-GET-Request zu einem späteren Zeitpunkt den Status beim PoPP-Service anfragen und den Benutzer entsprechend informieren.

6 Informationsmodell

Das abgebildete Informationsmodell bezieht sich ausschließlich auf die Anteile des Gesamtmodells für das Gerät des Versicherten und des PoPP-Service für das Ausstellen eines PoPP-Token nach Authentifizierung des Versicherten mit GesundheitsID oder Authentifizierung einer eGK über das PoPP-Modul in einer Krankenversicherungs-App auf dem Gerät des Versicherten.

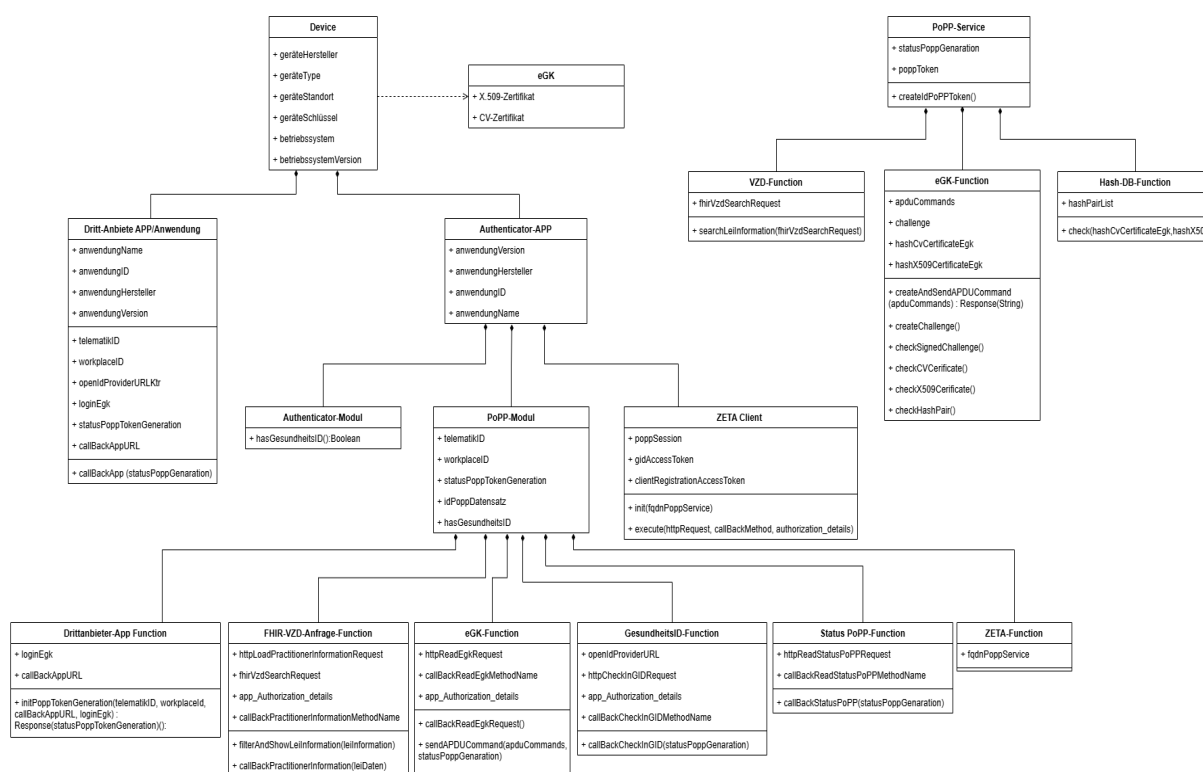


Abbildung 4: Bausteinsicht - Informationsmodell Gerät des Versicherten und PoPP-Service für Online-Anwendungsfälle zur PoPP-Token-Erstellung

7 Implementierungsleitfaden

Dieses Kapitel bietet Unterstützung für Hersteller, die den Online-Check-in mit PoPP in Apps umsetzen möchten. Ziel ist es, eine an den spezifischen Use Case angepasste Umsetzung zu ermöglichen. Der Inhalt gliedert sich wie folgt:

1. Einführung mit dem End-to-End Flow (Happy Path)

Zunächst wird der Happy Path beschrieben, der die Umsetzung des Online-Check-ins mit PoPP in einer Krankenversicherungs-App mit GesundheitsID und Scannen eines QR-Codes vor Ort darstellt. Die Vorgaben werden anhand der umzusetzenden Flow-Schritte aufgeführt.

2. Integration in Drittanbieter-Anwendungen

Neben Krankenversicherungen können auch Drittanbieter den Online-Check-in in Apps umsetzen. Hier werden zwei Umsetzungsvarianten mit unterschiedlichen Implementierungsaufwand für den Hersteller unterschieden:

- Integriertes PoPP-Modul: Das PoPP-Modul ist in der Drittanbieter-App integriert und ermöglicht die Durchführung des Online-Check-ins in der eigenen Drittanbieter-App (Bei Verwendung der GesundheitsID ist für den Vorgang der Autorisierung auch weiterhin ein App-Wechsel erforderlich).
- Nachnutzung eines anderen PoPP-Moduls (App2App): Die Drittanbieter-App nutzt das PoPP-Modul einer Krankenversicherungs-App nach und der eigentliche Online-Check-in wird in der Krankenversicherungs-App durchgeführt. Der Wechsel in den Krankenversicherungs-App und wieder zurück erfolgt hierbei automatisch.

3. Optionen und Alternativen

Neben der Darstellung des Happy Path werden Alternativen zu den einzelnen Flow-Schritten aus dem Happy Path aufgeführt, um die unterschiedliche Rahmenbedingungen von Use Cases zu berücksichtigen:

- Autorisierung: Variante zur Nutzung der eGK ohne PIN statt GesundheitsID
- Auswahl der LEI: Varianten, um Einrichtung auszuwählen statt QR-Code zu scannen
- Statusinformation: Varianten um über den Check-in-Status zu informieren
- Ersteinrichtung: Zusätzliche/Abweichende Flow-Schritt bei der erstmaligen Nutzung

4. Sprachliche Konsistenz und Terminologie

Abschließend werden Vorgaben zur Terminologie genannt.

7.1 End-to-End Flow (Happy Path)

Der Happy Path zeigt die Umsetzung des Online-Check-ins mit PoPP in einer Krankenversicherungs-App mit GesundheitsID und Scannen eines QR-Codes vor Ort. (Verweis auf PoPP-Service Spec. UC_PoPP_1a/UC_PoPP_1b).

Clickdummy / Screenshot

Visuelle Umsetzung des Online-Check-ins in einer Krankenversicherungs-App für einen Praxisbesuch mit GesundheitsID.

[\[Clickdummy\]](#)

Ausgangssituation

Der Happy Path umfasst nicht die Ersteinrichtung. Folgende Punkte werden entsprechend vorausgesetzt.

- Krankenversicherungs-App mit PoPP-Modul ist auf dem Smartphone einer VER installiert.
- [Auth-Modul]: GesundheitsID ist auf dem Smartphone der VER eingerichtet, GesundheitsID ist gültig, VER hat SSO zugestimmt, hat Komfortfeatures zugestimmt und benutzt Biometrie.
- [Auth-Modul]: VER hat Übermittlung von KVNR und IK-Nummer an den "Online-Check-in Service" einmalig zugestimmt.
- VER hat der App Zugriff auf die Kamera erteilt.
- VER ist mit dem Smartphone in einer Gesundheitseinrichtung vor Ort und hat Internetzugriff.
- LEI stellt für den Check-in einen QR-Code-Aufsteller am Empfang bereit.
- VER wird am Empfang aufgefordert, den QR-Code aus der App mit integriertem PoPP-Modul zu scannen.

Flow-Schritte und Vorgaben

Tabelle 3: Flow-Schritte und Vorgaben an den Online-Check-in im Happy Path.

	ID	Handlung durch	Aktion
Einstieg	1.1	VER	Wählt App-Symbol auf dem Smartphone
Check-in	2.1	APP [PoPP-Modul]	Homescreen mit Button: Online-Check-in
	2.2	VER	wählt Online-Check-in Button
Auswahl der LEI	3.1	APP [PoPP-Modul]	Screen mit Funktion QR-Code zu scannen und Buttons: Einrichtung manuell wählen und Fotomediathek
	3.2	VER	Hält Kamera auf den QR-Code.
Einwilligungen	4.1	APP [PoPP-Modul]	Namen der LEI wird angezeigt inkl. Adresse, tagesaktuelle Öffnungszeiten Adresse auswählbar, um externe Karten-App zu öffnen und Button: Bestätigen und Abbrechen und als Favorit hinzufügen
	4.2	VER	Wählt Bestätigen Button
Autorisierung	5.1	APP [PoPP-Modul]	Systemanzeige Biometrie (Fingerabdruck / Gesicht)

	5.2	VER	Zeigt Gesicht
Statusinformation	6.1	APP [PoPP-Modul]	Screen Online-Check-in erfolgreich und Button: Zurück zum Homescreen
Abschluss	7.1	VER	Ist eing_checked

7.2 Drittanbieter-Apps

Für Hersteller von Drittanbieter-Apps gibt es zwei Varianten einer versicherten Person den Online-Check-in zu ermöglichen. Die Umsetzungsvarianten sind dabei mit unterschiedlichen Implementierungsaufwand verbunden:

- Integriertes PoPP-Modul: Das PoPP-Modul ist in der Drittanbieter-App integriert und ermöglicht die Durchführung des Online-Check-ins in der eigenen Drittanbieter-App (Bei Verwendung der GesundheitsID ist für den Vorgang der Autorisierung auch weiterhin ein App-Wechsel erforderlich).
- Nachnutzung des PoPP-Moduls (App2App): Die Drittanbieter-App nutzt das PoPP-Modul einer Krankenversicherungs-App nach und der eigentliche Online-Check-in wird in der Krankenversicherungs-App durchgeführt. Der Wechsel in den Krankenversicherungs-App und wieder zurück erfolgt hierbei automatisch.

Neben der Umsetzung in Apps, ist auch die Umsetzung in (Web-) Anwendungen möglich.

7.2.1 Integriertes PoPP-Modul

Bei einem Online-Check-in aus einer Drittanbieter-App mit integrierten PoPP-Modul heraus sind nicht alle im Happy Path aufgeführten Schritte erforderlich. Ausnahmen, wie bspw. Entfall des Flow-Schritts *Auswahl der LEI*, da eine Telematik-ID direkt in der Drittanbieter-App hinterlegt werden kann, sind in den jeweiligen Kapiteln benannt.

Clickdummy/Screenshot

Visuelle Umsetzung des Online-Check-ins in einer Drittanbieter-App mit integriertem PoPP-Modul für Online-Dienste einer Apotheke mit eGK.

[Clickdummy]

Flowschritte und Vorgaben

- Abweichend vom Happy Path entfallen die Schritte Check-in und Auswahl der LEI.
- Der Einstieg sowie der Ausstieg können individuell durch die Drittanbieter-App festgelegt werden. Eine Ausnahme bildet der Sonderfall (nur eGK): Wenn für eine andere Person eing_checked werden soll, ist ein zusätzlicher Button "Für andere Person einchecken" in der Drittanbieter-App erforderlich.

7.2.2 Nachnutzung des PoPP-Moduls (App2App)

Bei einem Online-Check-in aus einer Drittanbieter-App, die das PoPP-Modul einer anderen App nachnutzen (App2App) sind nicht alle im Happy Path aufgeführten Schritte erforderlich. Ausnahmen, wie bspw. Entfall des Flowschritts *Auswahl der LEI*, da eine

Telematik-ID direkt in der Drittanbieter-App hinterlegt werden kann, sind in den jeweiligen Kapiteln benannt.

Clickdummy / Screenshot

Visuelle Umsetzung des Online-Check-ins in einer Drittanbieter-App für Online-Dienste einer Apotheke mit eGK.

[\[Clickdummy\]](#)

Visuelle Umsetzung des Online-Check-ins in einer Drittanbieter-App für Videosprechstunden mit GesundheitsID.

[\[Clickdummy\]](#)

Ausgangssituation

- VER hat vorab ausgewählt, für wen der Online-Check-in durchgeführt werden soll, für sich oder als Vertreterin oder Vertreter.

Flow-Schritte und Vorgaben

- Abweichend vom Happy Path entfallen die Schritte Check-in und Auswahl der LEI.
- Der Einstieg sowie der Ausstieg können individuell durch die Drittanbieter-App festgelegt werden. Eine Ausnahme bildet der Sonderfall (nur eGK): Wenn für eine andere Person eingecheckt werden soll, ist ein zusätzlicher Button "Für andere Person einchecken" in der Drittanbieter App erforderlich.
- Die App der Krankenversicherung darf weder Drittanbieter-Apps noch gültige Telematik-IDs von dem Online-Check-in ausschließen.

7.3 Alternative Flow-Schritte

Um die unterschiedliche Rahmenbedingungen von Use Cases an den Prozess zu berücksichtigen, werden im Folgenden Alternativen zu den Flow-Schritten abweichend des Happy Paths beschrieben.

7.3.1 Autorisierung

Neben der Autorisierung mittels GesundheitsID ist auch eine Autorisierung mit der eGK ohne PIN möglich. Hierbei ändert sich nur der Flow-Schritt *Autorisierung*.

Clickdummy/Screenshot

Visuelle Umsetzung des Flow-Schritts *Autorisierung* mit eGK ohne PIN als Screenshot.



Abbildung 5: Screenshot des Flow-Schritt Autorisierung mit eGK ohne PIN.

Ausgangssituation

- VER hat keine GesundheitsID eingerichtet und verwendet stattdessen die eGK oder VER möchte als Vertreterin oder Vertreter agieren und die eGK der zu vertretenden Person verwenden (s. Sonderfall).

Flow-Schritte und Vorgaben

Tabelle 4: Flow-Schritte und Vorgaben an den Online-Check-in mit der Variante Autorisierung mittels eGK.

	ID	Handlung durch	Aktion
Autorisierung	5.1	APP [PoPP-Modul]	Screen Zugangsnummer (CAN-Feld ist bereits ausgefüllt) Button: Weiter
	5.2	VER	Wählt Weiter Button
	5.3	APP [PoPP-Modul]	Screen zeigt Positionierung der eGK
	5.4	VER	Positioniert Karte

	5.5	APP [PoPP-Modul]	NFC-Dialog mit mindestens zwei Statusinformationen (Karte erkannt und Authentisierung erfolgreich)

- Um der VER das kontaktlose Einlesen der eGK zu erleichtern, soll der VER die korrekte Positionierung der eGK in Abhängigkeit vom verwendeten Smartphone-Modell angezeigt werden. Für Smartphone-Modelle sollen allgemeine Hinweise zur optimalen Positionierung bereitgestellt werden, die auf den typischen Positionen der NFC-Antennen basieren.
- Sofern eine GesundheitsID eingerichtet und gültig ist, soll immer diese verwendet werden.
- Sonderfall (Vertretung): Wenn die eincheckende Person als Vertreterin oder Vertreter handelt, muss diese Rolle bereits vorab in der App ausgewählt werden. In diesem Fall erfolgt die Autorisierung mit der eGK – auch dann, wenn eine GesundheitsID eingerichtet ist.
- Nur für Drittanbieter-Apps, die das PoPP-Modul einer anderen App nachnutzen (App2App): Weicht die Krankenversicherung der zu vertretenden Person ab, soll die Krankenversicherungs-App der Vertreterin oder des Vertreters verwendet werden, da diese in der Regel bereits auf dem Smartphone der VER installiert ist.

7.3.2 Auswahl der LEI

Die Auswahl der LEI für den Online-Check-in erfolgt in der Regel durch das Scannen eines QR-Codes. Alternativ kann auch eine Drittanbieter-App die Telematik-ID bereits in der Drittanbieter-App hinterlegen, sodass die Auswahl der LEI durch die VER entfällt. Wird keine Drittanbieter-App verwendet oder ist das Scannen eines QR-Codes nicht möglich, sollen der VER alternative Varianten zur Auswahl der LEI zur Verfügung stehen.

Clickdummy/Screenshot

Visuelle Umsetzung des Flow-Schritts *Auswahl der LEI* mit Verzeichnisdienst (VZD)
Suche, Anzeige einer Historie der letzten Online-Check-ins, Favoriten und Detailscreens einer Einrichtung

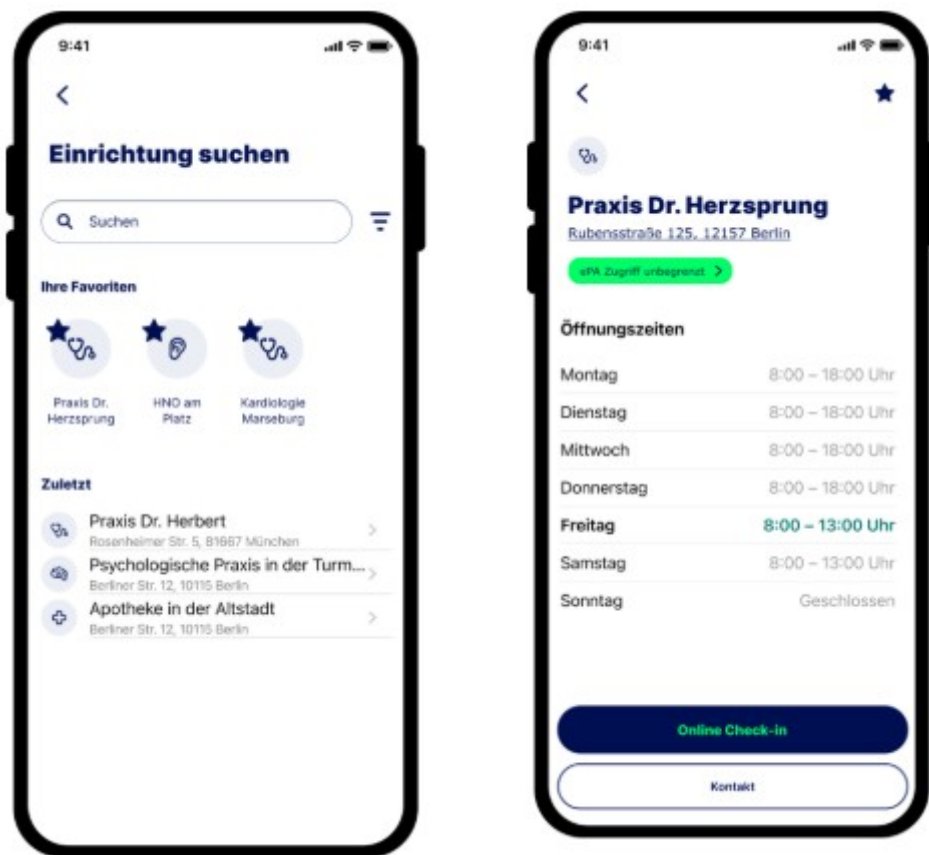


Abbildung 6: Screenshot des Flow-Schritts Auswahl der LEI mit VZD-Suche, Anzeige einer Historie der letzten Online-Check-ins, Favoriten und Detailscreens einer Einrichtung.

Flow-Schritte und Vorgaben

- Anzeige einer Historie der letzten zehn Online-Check-ins inkl. Adresse
- Möglichkeit, Favoriten anzulegen und anzuzeigen (keine Adresse erforderlich)
- Volltextsuche im VZD inkl. Filtermöglichkeiten (nur Einrichtungen, DiGA und Krankenversicherungen/Kostenträger dürfen angezeigt werden). Suchergebnisse inkl. Adresse anzeigen.

Tabelle 5: Filterkriterien für Suche im VZD - Einträgen auswählen/suchen

Filter	Ergebnis	Hinweis
Favoriten	Liste der Favoriten	alphabetisch sortiert VZD-Einträge, die durch die VER als Favoriten angelegt wurden
Zuletzt genutzt	Liste der zuletzt durchgeführte Online-Check-ins	chronologisch sortiert optional einblenden: weitere verwendete Einrichtungen aus anderen Modulen anzeigen: E-Rezept gesendet, ePA freigeschaltet, Widerspruch eingelegt, ...

1136

1137

Tabelle 6: Filterkriterien für Suche im VZD - Einträge eingrenzen

Filter	Ergebnis	Hinweis
Art der Einrichtung (Apotheke, Krankenhaus, Zahnarztpraxis, ...)	Zeigt nur die Einrichtungen mit der ausgewählten Art an	Auswahl aus Liste und Suchfeld alphabetisch sortiert Krankenversicherung und Kostenträger in einer Einrichtungsart zusammenführen)
Ort	Zeigt nur Einrichtungen mit dem ausgewählten Ort	Suchfeld alphabetisch sortiert
PLZ	Zeigt nur Einrichtungen mit der ausgewählten PLZ	Suchfeld numerisch sortiert
In meiner Nähe (ermittelter/gewählter Standort + Radius)	Zeigt nur Einrichtung in meiner Nähe an	Auswahl aus Kartenansicht und Liste Filter darf nur gezeigt werden, wenn Filter Art der Einrichtung = Apotheke Radius über einen Schieberegler festlegen, wobei die maximale Radiusgröße so zu begrenzen ist, dass weniger als 100 Einrichtungen angezeigt werden.
Spezialisierung (Allgemeinmedizin, Neurologie, Sprachtherapie, ...)	zeigt nur Einrichtungen, mit der Spezialisierung an	Auswahl aus Liste und Suchfeld alphabetisch sortiert Filter darf nur gezeigt werden, wenn Filter Art der Einrichtung = Praxis, Zahnarztpraxis, Krankenhaus, vom Typ Heil- und Hilfsmittel
Physische Faktoren (Parkmöglichkeit, ÖPNV in der Nähe, Barrierefrei, Abholautomat)	Zeigt nur Einrichtungen mit der	Auswahl aus Liste und Suchfeld alphabetisch sortiert

	ausgewählten Faktoren an	Filter darf nur gezeigt werden, wenn Filter Art der Einrichtung = Apotheke
Aktuell geöffnet (Öffnungszeiten, Notdienste, Sonderschließzeiten)	zeigt nur die Einrichtung, die offene haben	Filter darf nur gezeigt werden, wenn Filter Art der Einrichtung = Apotheke
Apothekenservices (Pharmazeutische und medizinische Leistungen, Einlösewege)	zeigt Einrichtungen mit ausgewählten Services	Auswahl aus Liste und Suchfeld alphabetisch sortiert Filter darf nur gezeigt werden, wenn Filter Art der Einrichtung = Apotheke vorher gesetzt

Tabelle 7: Filterkriterien für Suche im VZD - Einträge darstellen

Filter	Ergebnis	Hinweis
Ansicht	Listenansicht / Kartenansicht	Kartenansicht: Die Einrichtungen sollen auf einer Karte sichtbar sein. Wenn der Kartenausschnitt verschoben wird, soll eine erneute Suche im Kartenausschnitt möglich sein ("Hier suchen"); es soll möglich sein, an meinen Standort zu zoomen Kartenansicht darf nur gezeigt werden, wenn Filter Art der Einrichtung = Apotheke

Erhält die versicherte Person die Check-in-Informationen außerhalb einer App (z.B. per E-Mail oder über eine Website), ist das Scannen eines QR-Codes auf demselben Gerät, auf dem auch die App der Krankenversicherung installiert ist, nicht ohne weiteres möglich.

Daher sollen folgende Alternativen unterstützt werden:

- QR-Code als Bild speichern und das gespeicherte Bild zur Erkennung hochladen
- Telematik-ID direkt kopieren
- Telematik-ID oder QR-Code über das systemweite Share-Sheet teilen

7.3.3 Statusinformation

Die VER wird nach Abschluss des Online-Check-ins darüber informiert, wie der Status des Online-Check-ins ist.

Flow-Schritte und Vorgaben

Die Benachrichtigung über den Check-in-Status müssen kann von der App am PoPP-Service abgefragt und dem Nutzer dargestellt werden.

7.3.4 Ersteinrichtung

Bei der erstmaligen Verwendung des Online-Check-ins sind zusätzliche Flow-Schritte erforderlich, die sich je nach Rahmenbedingungen des Use Cases unterscheiden.

Clickdummy/Screenshot

Visuelle Umsetzung des Online-Check-ins in einer Krankenversicherungs-App für einen Praxisbesuch mit GesundheitsID inkl. zusätzlicher Flow-Schritte für die Ersteinrichtung: Einrichtung der GesundheitsID und Zustimmung der VER zur Nutzung des Online-Check-ins.

[\[Clickdummy\]](#)

Visuelle Umsetzung des Online-Check-ins in einer Drittanbieter-App (App2App) für einen Online-Dienst einer Apotheke mit eGK inkl. zusätzlicher Flow-Schritte für die Ersteinrichtung: Auswahl der Krankenversicherungs-App, CAN-Eingabe und Zustimmung der VER zur Nutzung des Online-Check-ins.

[\[Clickdummy\]](#)

Visuelle Umsetzung des Online-Check-ins in einer Drittanbieter-App (App2App) für Videosprechstunden mit GesundheitsID inkl. zusätzlicher Flow-Schritte für die Ersteinrichtung: Auswahl der Krankenversicherungs-App, Einrichtung GesundheitsID und Zustimmung der VER zur Nutzung des Online-Check-ins.

[\[Clickdummy\]](#)

Visuelle Umsetzung eines zusätzlichen Flow-Schritts nach erstmaliger Bereitstellung des Online-Check-in in einer App, der auf die neue Online-Check-in-Funktion hinweist.

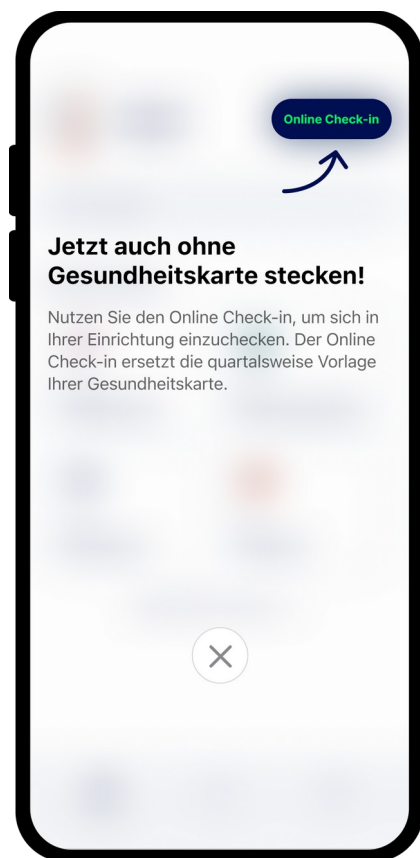


Abbildung 7: Screenshot der auf die neue Online-Check-in-Funktion hinweist.
Flow-Schritte und Vorgaben

- Bei der erstmaligen Bereitstellung des Online-Check-ins in der App soll der VER ein Tooltip eingeblendet werden, der auf die neue Online-Check-in-Funktion hinweist und sie kurz erklärt.
- Bei Nutzung des Online-Check-ins aus einer Drittanbieter-App, die das PoPP-Modul einer anderen App nachnutzt (App2App), muss in dieser zunächst die Krankenversicherung ausgewählt, damit die passende App der Krankenversicherung geöffnet wird. Ab der zweiten Nutzung soll die Krankenversicherung nicht mehr ausgewählt werden müssen.
- Bei der Erstverwendung muss die VER einwilligen, dass für den Online-Check-in erforderliche Daten an den Online-Check-in-Service übermittelt werden.
- Bei der Erstverwendung mit eGK muss die VER einmalig die CAN eingeben. Diese soll gespeichert werden, damit sie ab der zweiten Nutzung bereits vorausgefüllt ist.
- Wird die eigene eGK verwendet, soll der VER, die Möglichkeit angeboten werden, stattdessen auch eine GesundheitsID einzurichten. Nach der Authentisierung mit eGK soll eine Abfrage zur Einrichtung einer GesundheitsID erscheinen. Die Abfrage soll nur bei den ersten drei Online-Check-ins angezeigt werden.
- Der Online-Check-in muss in der Krankenversicherungs-APP ohne zusätzlichen Account und ohne nicht erforderliche Einrichtungsschritte möglich sein. Davon

1203 ausgenommen sind gesetzlich notwendige Schritte, beispielsweise die Zustimmung zu
1204 Nutzungsbedingungen.

1205 7.4 Terminologie

1206 Die Texte dürfen geringfügig angepasst werden, etwa wenn in der App das „Du“ genutzt
1207 wird.

Thema	Text
Button für PoPP-Modul	Online-Check-in
Einwilligung [Auth-Modul] zur Übertragung KVNR, IK	<p>Online-Check-in nutzen Wenn Sie den Online-Check-in nutzen möchten, werden folgende Daten an den Dienst der gematik GmbH übermittelt:</p> <ul style="list-style-type: none"> • Ihre Versichertennummer • Name Ihrer Versicherung <p>[Ablehnen] [Erlauben]</p>
Abfrage Einrichtung GesundheitsID nach eGK-Authentisierung	<p>GesundheitsID einrichten Sie können Ihre Anmeldedaten speichern, sodass Sie Ihre Gesundheitskarte nicht bei jeder Anmeldung benötigen. Hierfür brauchen Sie entweder die PIN Ihres Personalausweises oder die PIN Ihrer Gesundheitskarte. [GesundheitsID einrichten] [Vielleicht später]</p>
Text bei fehlerhaften QR-Code	<p>QR Code fehlerhaft Der gescannte QR-Code ist nicht lesbar, da er fehlerhafte Daten enthält. Bitte informieren Sie umgehend die Einrichtung bei der Sie sich einchecken wollten. [Ok]</p>
Kein Internet bei Buttonclick: Online-Check-in	<p>Kein Internet Wir konnten Sie leider nicht anmelden, da Sie keine Internetverbindung haben. [Abbrechen] [Erneut probieren]</p>
Abgelehnte Kamerazugriffs-Berechtigung	<p>Kamerazugriff benötigt Bitte erlauben Sie [der xxx App] den Kamerazugriff in den Einstellungen. [Abbrechen] [Einstellungen öffnen]</p>
Abgelehnte App2App-Berechtigung (bei Aufruf aus andere App)	<p>Öffnen anderer Apps erlauben Bitte erlauben Sie [der xxx App] andere Apps zu öffnen in den Einstellungen. [Abbrechen] [Einstellungen öffnen]</p>

Erfolgreicher Check-in	Online-Check-in erfolgreich Sie haben sich erfolgreich bei [xxx] eing_checked. [Zur Startseite]
LEI noch nicht erreichbar	Check-in ausstehend Die Einrichtung, in der Sie sich einchecken möchten, ist zurzeit geschlossen. Der Check-in wird automatisch fortgesetzt, sobald die Einrichtung wieder geöffnet hat. Sie werden dazu benachrichtigt, sofern Sie Push-Nachrichten erlaubt haben. [Ok]
LEI nach 120h immer noch nicht erreichbar	Check-in abgebrochen Die Einrichtung, in der Sie sich einchecken wollten, hat leider nicht reagiert. Bitte probieren Sie es erneut oder kontaktieren Sie bei wiederholtem Auftreten des Problems die Einrichtung. [Abbrechen] [Erneut probieren] [Einrichtung kontaktieren]
Angefragtes Authentifizierungsniveau „loa-high“, aber keine GesundheitsID eingerichtet	GesundheitsID erforderlich Für diesen Vorgang reicht ihre Gesundheitskarte nicht aus. Bitte richten Sie stattdessen eine GesundheitsID ein. [Abbrechen] [GesundheitsID einrichten]

1208

8 Test

1209 Um automatisierte Zulassungstests zu ermöglichen, muss das PoPP-Modul eine
1210 Testtreiber-Schnittstelle implementieren.

1211 **8.1 Schnittstelle für Testtreiber**

Offener Punkt: OP-PoPP-2

Es soll eine Testtreiberschnittstelle analog zum Vorgehen beim ePA-FdV für PoPP-Modul/ App-mit-Popp-Modul entwickelt werden. Die entsprechenden Festlegungen werden in [gemKPT_Test] oder anderswo an geeigneter Stelle aufgenommen werden.

9 Anhang A - Verzeichnisse

9.1 Abkürzungen

Kürzel	Erläuterung
eGK	elektronische Gesundheitskarte
ePA	elektronischen Patientenakte
FdV	Frontend des Versicherten
HW	Hardware
IDP	Identity Provider
LE	Leistungserbringer
LEI	Leistungserbringerinstitution
OCSP	Online Certificate Status Protocol
PAR	Pushed Authorization Request
PoPP	Proof of Patient Presence
PS	Primärsystem
QR-Code	Quick Response Code
URL	Uniform Resource Locator
VER	versicherte Person
VSDM	Versichertenstammdatenmanagement

9.2 Glossar

Begriff	Erläuterung
Access-Token	Im Kontext vom PoPP handelt es sich um Access Token, die vom ZETA Guard des PoPP-Service für Prozess zur Erstellung

	eines PoPP-Token ausgestellt wurden. Die Access Token werden an den ZETA Client gesendet und dort zur laufenden Session gespeichert. Jeder Aufruf des PoPP-Modul an den PoPP-Service wird mit dem Access-Token durch das ZETA Client ergänzt.
Drittanbieter-App	Drittanbieter-Apps bieten den Versicherten digitale Service zur Unterstützung medizinischer Anwendungsfälle an. Beispiele für Drittanbieter-Apps sind Apotheken-Apps, Videosprechstunden-Apps oder DiGA-Apps. Die Apps von Krankenversicherungen aka Krankenversicherungs-Apps sind von Drittanbieter-Apps verschieden. Drittanbieter-Apps können für beliebige Smartphone-Betriebssysteme wie Android oder iOS sowie für unterschiedliche Desktop-Betriebssysteme wie Windows oder Linux verfügbar sein. Zudem ist es möglich, dass Drittanbieter-Apps innerhalb beliebiger Internetbrowser laufen.
Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.
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.
Krankenversicherungs-App	Mit einer Krankenversicherungs-App bieten Krankenversicherungen ihren Versicherten digitale Service zur Unterstützung medizinischer Anwendungsfälle an.
PoPP-Client	Eine Komponente im Primärsystem, die für die sichere Kommunikation zum PoPP-Service verantwortlich ist.
PoPP-Modul	Eine Komponente von Krankenversicherungs-App oder Drittanbieter-App, welche für Online-Anwendungsfälle die Kommunikation mit dem PoPP-Service übernimmt. Das PoPP-Modul initiiert die Authentifizierung eines Versicherten mit GesundheitsID oder einer eGK-in-Fernversorgung.
PoPP-Service	Zentraler Dienst in der Telematikinfrastruktur 2.0 (TI 2.0), der PoPP-Token generiert und verwaltet.
PoPP-Service Resource Server	Komponente des PoPP-Service, 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 elektronischer Gesundheitskarte (eGK), und einer Leistungserbringerinstitution (LEI).
Telematik-ID	Die Telematik-ID ist die eindeutige elektronische Identität von Leistungserbringern und medizinischen Institutionen in der TI. Sie wird von den Sektoren des Gesundheitswesens zugewiesen und verwaltet.
Versorgungskontext (VK)	<p>Der Versorgungskontext beschreibt die sichere und kryptografisch belegte Verbindung zwischen einem berechtigten Versicherten und einer authentifizierten Leistungserbringerinstitution. Diese Verbindung autorisiert den Zugriff auf anwendungsbezogene Versicherungsdaten über die Telematikinfrastruktur (TI) Anwendungen</p> <p>Ein Versorgungskontext besteht, wenn ein 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-in-Fernversorgung, und ist auch bei telemedizinischen Anwendungen relevant</p>
Versicherter (im Kontext PoPP)	Im Kontext von PoPP "Versicherter" eine Person, die ihre Identität im Gesundheitswesen einbringt, um einen Versorgungskontext mit einer Leistungserbringerinstitution (LEI) zu etablieren. Dies erfolgt entweder durch die GesundheitsID oder die elektronische Gesundheitskarte (eGK). Bei der Verwendung der eGK kann auch ein Vertreter des Versicherten agieren. Der Versicherte trägt zur Erstellung des Versorgungskontexts bei, indem er seine Krankenversicherungsnummer (KVNR) bereitstellt und sich entweder über die GesundheitsID authentisiert, persönlich oder durch einen Vertreter mit der eGK bei der LEI vorstellt, oder mobil per eGK bei der LEI anmeldet.
WorkplaceID	Die WorkplaceID ist ein Identifikator, um ein erstelltes PoPP-Token einen bestimmten Arbeitsplatz (z.B. in einem PVS) oder einem bestimmten Prozess (z.B. bei Online-Apotheken) zuordnen zu können.

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

1215
1216

9.3 Abbildungsverzeichnis

Abbildung 1: Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in App der Krankenversicherung.....	9
Abbildung 2: Schnittstellen vom und zum PoPP-Modul.....	18
Abbildung 3: Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in Drittanbieter App.....	27
Abbildung 4: Bausteinsicht - Informationsmodell Gerät des Versicherten und PoPP-Service für Online-Anwendungsfälle zur PoPP-Token-Erstellung.....	39
Abbildung 5: Screenshot des Flow-Schritt Autorisierung mit eGK ohne PIN.....	44
Abbildung 6: Screenshot des Flow-Schritts Auswahl der LEI mit VZD-Suche, Anzeige einer Historie der letzten Online-Check-ins, Favoriten und Detailscreens einer Einrichtung.	46
Abbildung 7: Screenshot der auf die neue Online-Check-in-Funktion hinweist.....	50
Abbildung 8: Laufzeitsicht - Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID.....	63
Abbildung 9: Laufzeitsicht - Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service.....	63
Abbildung 10: Laufzeitsicht - Initialisierung ZETA Client.....	64
Abbildung 11: Laufzeitsicht - Ermittlung Suchkriterien, VZD-Datenabruf mit Suchkriterien und Einwilligung in die Datenweitergabe.....	65
Abbildung 12: Laufzeitsicht - Information zum Status der PoPP-Token-Generierung an das PoPP-Modul.....	66
Abbildung 13: Laufzeitsicht - Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID.....	67
Abbildung 14: Laufzeitsicht - Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service.....	84
Abbildung 15: Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service.....	98

9.4 Tabellenverzeichnis

Tabelle 1: Beschreibung der wesentlichen Komponenten für die Online Anwendungsfälle zur PoPP-Token Erstellung.....	9
Tabelle 2: Tab_PoPP_Modul_Payload_stat_QRCode]: Payload QR-Code.....	31
Tabelle 3: Flow-Schritte und Vorgaben an den Online-Check-in im Happy Path.....	41
Tabelle 4: Flow-Schritte und Vorgaben an den Online-Check-in mit der Variante Autorisierung mittels eGK.....	44
Tabelle 5: Filterkriterien für Suche im VZD - Einträgen auswählen/suchen.....	46
Tabelle 6: Filterkriterien für Suche im VZD - Einträge eingrenzen.....	47
Tabelle 7: Filterkriterien für Suche im VZD - Einträge darstellen.....	48

Tabelle 8: Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung des Versicherten über seine GesundheitsID.....	68
Tabelle 9: Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service.....	84
Tabelle 10: Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service.....	99

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.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Einführung der Gesundheitskarte - Glossar
[gemSpec_PoPP_Service]	gematik: Spezifikation Proof of Patient Presence (PoPP)-Service (Dokument in Freigabe)
[gemF_PoPP_Online_Check-in]	gematik: Feature Spezifikation für das PoPP Feature Online-Check-in (Dokument gemeinsam mit [gemSpec_PoPP_Modul] in Kommentierung)
[gemKPT_PoPP]	gematik: Technisches Konzept Proof of Patient Presence (PoPP) https://gemspec.gematik.de/docs/gemKPT/gemKPT_PoPP/
[gemSpec_Zeta]	gematik: Spezifikation Zero Trust Access (ZETA) https://gemspec.gematik.de/docs/gemSpec/gemSpec_ZETA/
[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)
[gemSpec_IDP_Sek]	gematik: Spezifikation Sektoraler Identity Provider https://gemspec.gematik.de/docs/gemSpec/gemSpec_IDP_Sek/

[API ZETA Client]	https://github.com/gematik/zeta
[api-popp]	OpenAPI Schnittstellenspezifikation des PoPP-Service Resource Server für Clients https://github.com/gematik/api-popp/tree/US-2_CC1
[I_PoPP_CheckIn_HID.yaml]	OpenAPI Schnittstellenspezifikation für Authentifizierung mit GesundheitsID und Anlegen eines PoPP-Datensatzes https://github.com/gematik/api-popp/blob/US-2_CC1/src/openapi/I_PoPP_CheckIn_HID.yaml
[I_PoPP_CheckIn_eGK_Verification.yaml]	OpenAPI Schnittstellenspezifikation für Authentifizierung einer eGK und Anlegen eines PoPP-Datensatzes https://github.com/gematik/api-popp/blob/US-2_CC1/src/openapi/I_PoPP_CheckIn_eGK_Verification.yaml
[I_PoPP_Load_Practitioner_Information.yaml]	OpenAPI Schnittstellenspezifikation für das Laden der Informationen zu einer LEI vom FHIR-VZD https://github.com/gematik/api-popp/blob/US-2_CC1/src/openapi/I_PoPP_Load_Practitioner_Information.yaml
[I_PoPP_Read_Status_PoPP.yaml]	OpenAPI Schnittstellenspezifikation für das Lesen des aktuellen Status zur PoPP-Token Erstellung und Auslieferung https://github.com/gematik/api-popp/blob/US-2_CC1/src/openapi/I_PoPP_Read_Status_PoPP.yaml
[I_PoPP_Token_Generation_online_check_in.yaml]	OpenAPI Schnittstellenspezifikation zur Übertragung von PoPP-Token an den PoPP-Client, wenn die Übertragung durch den PoPP-Service nach einem Online Check-in initiiert wird https://github.com/gematik/api-popp/blob/US-2_CC1/src/openapi/I_PoPP_Token_Generation_online_check_in.yaml
[I_PoPP_Token_Generation.yaml]	OpenAPI Schnittstellenspezifikation für PoPP-Service Resource Server für PoPP-Clients: https://github.com/gematik/api-popp/blob/US-2_CC1/src/openapi/I_PoPP_Token_Generation.yaml

9.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
----------	--

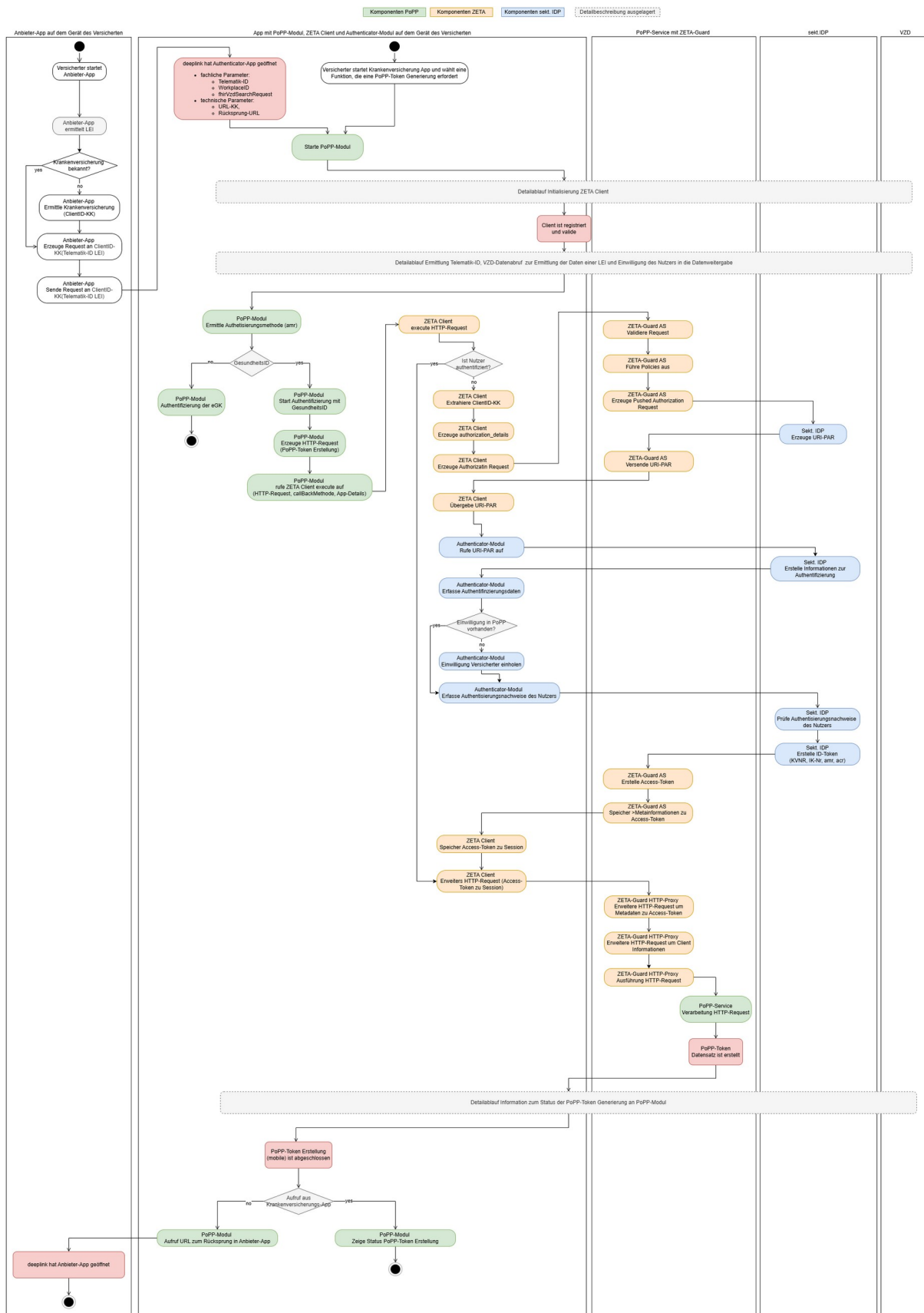
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels https://datatracker.ietf.org/doc/html/rfc2119
[RFC8414]	OAuth 2.0 Authorization Server Metadata https://datatracker.ietf.org/doc/html/rfc8414
[RFC7636]	Proof Key for Code Exchange by OAuth Public Clients https://datatracker.ietf.org/doc/html/rfc7636
[RFC8252]	OAuth 2.0 for Native Apps https://datatracker.ietf.org/doc/html/rfc8252
[RFC9396]	OAuth 2.0 Rich Authorization Requests https://www.rfc-editor.org/rfc/rfc9396
[RFC8259]	D. B. Crockford, "The JavaScript Object Notation (JSON)," RFC 8259, Dez. 2017. https://datatracker.ietf.org/doc/html/rfc8259
[OpenID Federation 1.0]	OpenID Federation Standard https://openid.net/specs/openid-federation-1_0.html
[RFC3629]	D. B. Cohen, "UTF-8, a transformation format of ISO 10646," RFC 3629, Nov. 2003. https://datatracker.ietf.org/doc/html/rfc3629

10 Anhang B - Ablaufbeschreibungen

10.1 Ablaufdiagramme

10.1.1 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung Versicherter über ihre GesundheitsID

Spezifikation Frontend des Versicherten für PoPP (Proof of Patient Presence)



10.1.2 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service

[illegible]

Seite 62 von 107
Stand: 26.01.2026

10.1.3 Detailablauf "ZETA Client Initialisierung"

Bei der Initialisierung des ZETA Client wird, wenn noch nicht bekannt, eine Client-Registrierung und App-Attestierung im ZETA Guard vorgenommen. Wenn der Ablauf zum PoPP-Modul zurückgekehrt ist, ist sowohl das Geräte als auch die App mit dem integrierten PoPP-Modul im ZETA Guard registriert.

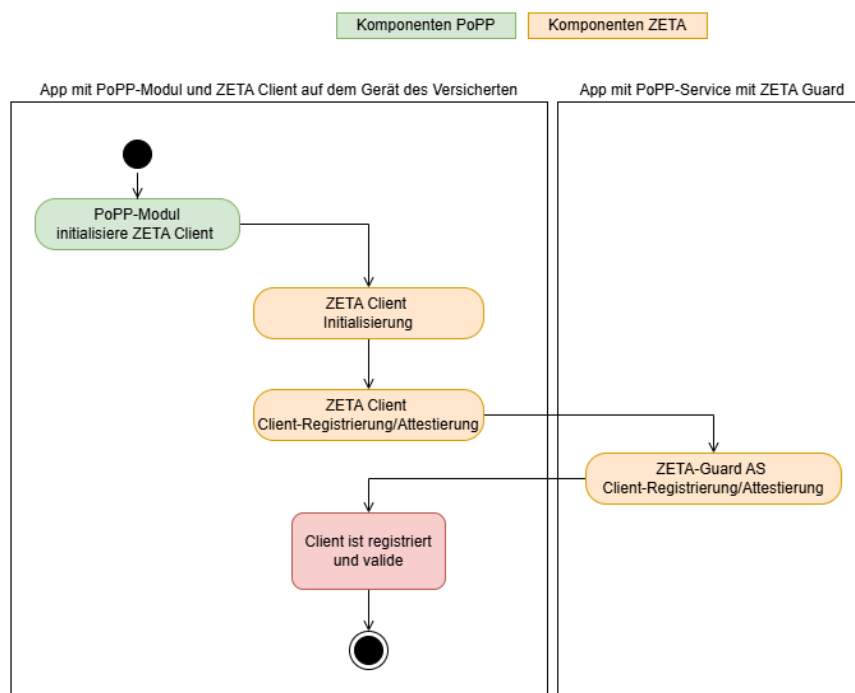


Abbildung 10: Laufzeitsicht - Initialisierung ZETA Client

10.1.4 Detailablauf "FHIR-VZD-Anfrage"

Die Abfrage des FHIR-VZD nach Daten zu LEIs muss die Anwendungsfälle unterstützen:

- Die Telematik-ID ist bekannt und es müssen für die Einwilligung des Nutzers weitere Daten zur LEI ermittelt werden.
- Die Telematik-ID ist nicht bekannt und es muss mit einer Suche auf Basis von bekannten Suchparametern (Name der LEI, Adresse/PLZ der LEI, u.ä.) durchgeführt werden. Als Ergebnis werden alle Daten der LEI für die Einwilligung des Nutzers und die Telematik-ID der LEI erwartet.

Die Unterscheidung beider genannten Fälle erfolgt lediglich aufgrund des formulierten Requests an den FHIR-VZD.

Die FHIR-VZD-Abfrage erfolgt durch das PoPP-Modul über die ZETA-Komponenten durch den PoPP-Service. Das PoPP-Modul erstellt den eigentlichen FHIR-VZD Search Request auf Basis der im PoPP-Modul vorhandenen Daten zur LEI. Der FHIR-VZD Search Request wird über die ZETA-Komponenten an den PoPP-Service propagiert. Dabei stellen die ZETA-Komponenten auf Basis der Client-Registrierung ein Client Access-Token aus, mit dem der Suchauftrag beim PoPP-Service angereichert wird.

Der PoPP-Service ist ein registrierter FHIR-VZD-Client und führt die Suche mit dem übergebenen FHIR-VZD Search Request aus. Die ZETA-Komponenten übertragen das Suchergebnis an das PoPP-Modul zur weiteren Verarbeitung.

1310
1311



1313
1314

1315
1316

1317
1318
1319
1320
1321

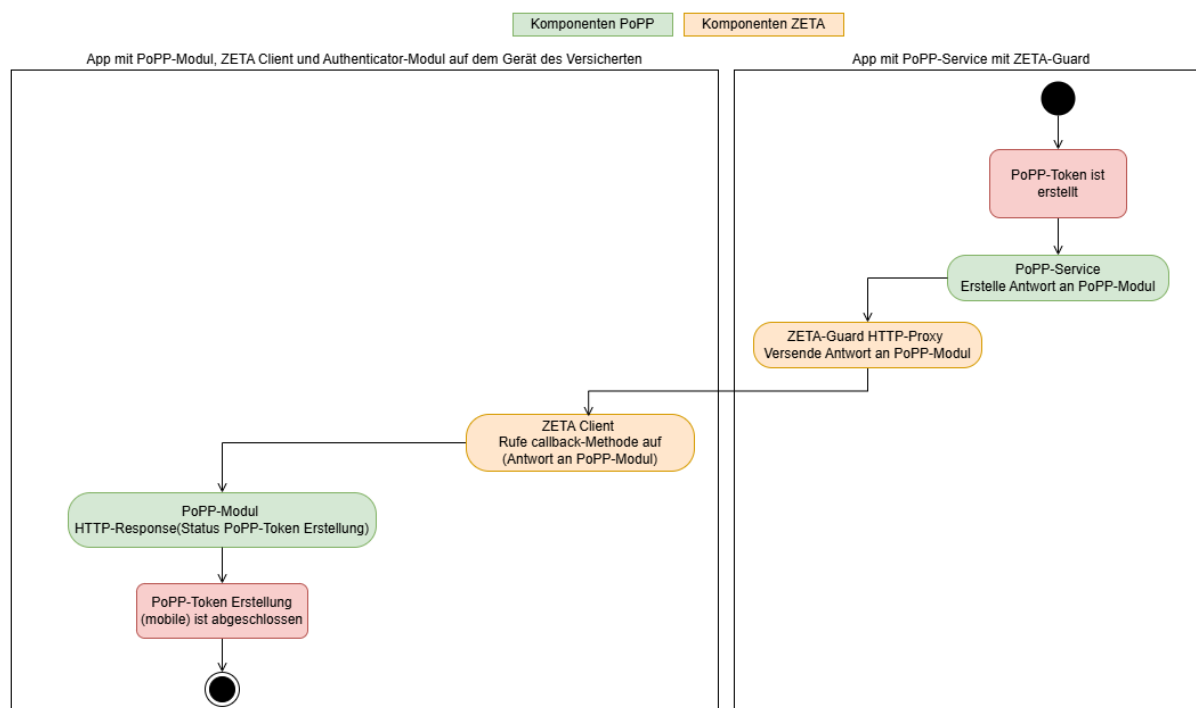


Abbildung 12: Laufzeitsicht - Information zum Status der PoPP-Token-Generierung an das PoPP-Modul

10.2 Flowdiagramme

10.2.1 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung Versicherter über ihre GesundheitsID

Das Sequenzdiagramm "Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID" stellt den vollständigen Ablauf der PoPP-Token-Generierung für den Fall dar, dass ein Versicherter über seine GesundheitsID authentifiziert wird.

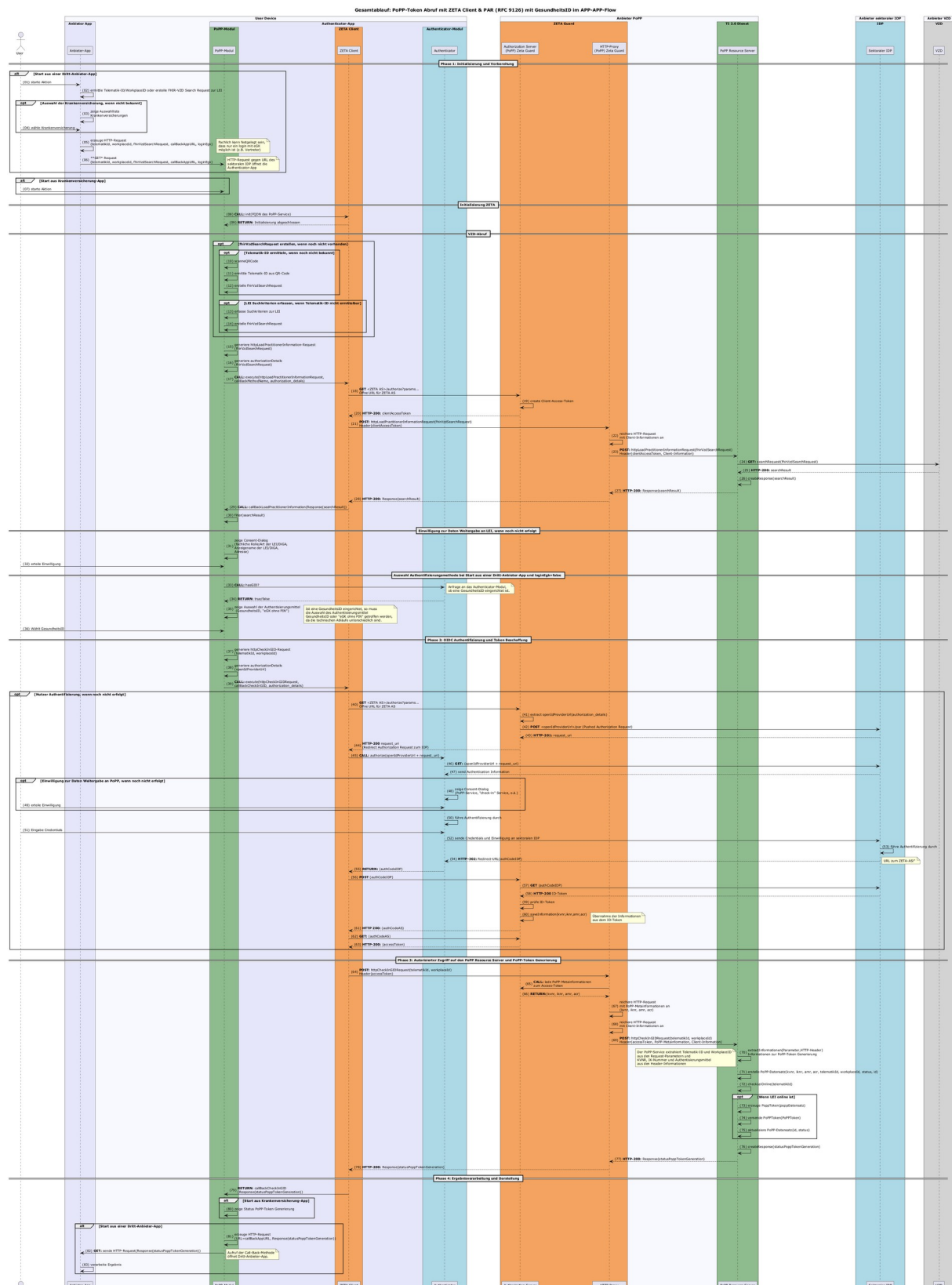


Abbildung 13: Laufzeitsicht - Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID

10.2.2 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung Versicherter mit ihrer GesundheitsID

Die Tabelle [Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung des Versicherten über seine GesundheitsID] beschreibt die einzelnen Schritte vom Start der PoPP-Token-Erstellung durch den Nutzer einer Fachanwendung bis zur Darstellung des Ergebnis der PoPP-Token-Erstellung.

Tabelle 8: Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung des Versicherten über seine GesundheitsID

	Schritt	Beschreibung
Start Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		
1	Aktion starten	Der Versicherte wählt eine Funktion in der App eines Anbieters (z.B. "Anmelden zum Arztbesuch", "Rezept einlösen").
2	ermittle Telematik-ID/WorkplaceID oder erzeuge FHIR-VZD Search Request zur Datenermittlung zur LEI	<p>In der Anbieter-App werden Informationen zur LEI ermittelt, für welche das PoPP-Token bestimmt ist. Dies ist Anbieter-App-spezifisch und kann auf unterschiedlichen Wegen erfolgen. Z.B.</p> <ul style="list-style-type: none"> • ist bei der Anbieter-App einer LEI (online Apotheke) die Telematik-ID bekannt, • bei der Anbieter-App zur Vermittlung einer Videosprechstunde bietet die App entsprechende Auswahlfunktionen an, • eine Praxis kann die Telematik-ID als QR-Code bereitstellen. • Es werden Suchkriterien erfasst, mit denen eine FHIR-VZD Anfrage durchgeführt werden kann
3	zeige Auswahlliste Krankenversicherungen	Die Liste der möglichen Krankenversicherungen kann am idp-list-Endpunkt beim Federation Master abgefragt werden. Zu jeder Krankenversicherung gibt es einen Datensatz mit Name und Logo der Krankenversicherung sowie der ClientID (OpenID-Provider URL) des sektoralen IDP der Krankenversicherung in der TI-Föderation. Einmalig ausgewählt kann die Anbieter-App die

		Informationen zur Krankenversicherung speichern.
4	wähle Krankenversicherung	Der Versicherte wählt seine Krankenversicherung aus der Liste aus. Die Auswahl enthält auch die ClientID (OpenID-Provider URL) des sektoralen IDP der Krankenversicherung.
5	erzeuge HTTP-GET Request(telematikId, workplaceId, fhirVzdSearchRequest, callBackAppURL, loginEgk)	<p>Die Anbieter-App erzeugt einen HTTP-GET Request und übergibt als Parameter</p> <ul style="list-style-type: none"> • <i>telematikId</i> -ist die Telematik-ID der LEI, für die ein PoPP-Token erstellt werden soll, • <i>workplaceId (optional)</i>-ist eine WorkplaceID, wenn das PoPP-Token einem bestimmten Arbeitsplatz der LEI zugeordnet werden soll, • <i>fhirVzdSearchRequest (optional)</i>- Search-Request für den Aufruf der FHIR-VZD API, um Daten zu einer LEI zu erhalten • <i>callBackURL (Rücksprung-URL)</i>- welche das PoPP-Modul aufrufen muss, wenn die Erzeugung eines PoPP-Token abgeschlossen ist, • <i>loginEgk (optional)</i> - ist die Information, ob das Login mit eGK erfolgen muss, z.B. wenn der Versicherte dies wünscht oder wenn es sich um Vertreter eines Versicherten handelt.
6	sende HTTP-GET Request(telematikId, workplaceId, fhirVzdSearchRequest, callBackAppURL, loginEgk)	<p>Die Anbieter-App sendet den HTTP-GET Request an die OpenID-Provider URL des sektoralen IDP. Die URL entspricht der ClientID des sektoralen IDP in der TI-Föderation und wurde zuvor über die Auswahl der Krankenversicherung ermittelt. Der HTTP-Request gegen die URL des sektoralen IDP öffnet die Authenticator-App.</p> <p>Aufgrund der Parameter im Request wird das PoPP-Modul in der Authenticator-App aufgerufen.</p>

Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		
Start Alternative -Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
7	Aktion starten	Der Versicherte wählt eine Funktion in der Krankenversicherungs-App (z.B. "Praxis Check-in")
Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
8	CALL: init(FQDN des PoPP-Service)	Beim ersten Aufruf einer Session wird der ZETA Client initialisiert. Dazu wird der FQDN des PoPP-Service übergeben. Der FQDN des PoPP-Service ist ein Konfigurationsparameter des PoPP-Modul. Die ZETA-Komponenten prüfen, ob der Client (Gerät und App) bereits registriert ist. Ist das nicht der Fall, findet eine Client-Registrierung statt (Attestation, Policies) statt.
9	RETURN: Initialisierung abgeschlossen	Nach Abschluss der Initialisierung kehrt die Anwendung in das PoPP-Modul zurück.
FHIR-VZD Search Request erzeugen, wenn nicht bereits vorhanden		
Alternative: Telematik-ID ermitteln		
1 0	scanneQRCode	Ist zu diesem Zeitpunkt noch keine Telematik-ID erfasst, für die ein PoPP-Token erstellt werden soll, so muss dies in diesem Schritt erfolgen. Bei "Check-in" in einer Praxis wäre das z.B. das Scannen eines QR-Codes.
1 1	ermittle Telematik-ID aus QR-Code	Der QR-Code wird geprüft und die Telematik-ID sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.
1 2	erstelle FHIR-VZD Search Request	Es wird ein Search-Request mit der Telematik-ID für den Aufruf der FHIR-VZD API erstellt, um Daten zu einer LEI zu erhalten

Alternative: Suchkriterien erfassen		
1 3	erfasse Suchkriterien zur LEI	In diesem Schritt erfolgt die Erfassung der Suchkriterien zur LEI. Suchkriterien können z.B. bestehen aus bekannten Angaben zur LEI (Name, Fachrichtung, PLZ) und zusätzlichen Informationen (Umkreissuche).
1 4	erstelle FHIR-VZD Search Request	Es wird ein Search-Request mit den Suchkriterien für den Aufruf der FHIR-VZD API erstellt, um Daten zu einer LEI zu erhalten
Der FHIR-VZD-Abruf wird über den ZETA-Ablauf an den PoPP-Service delegiert		
1 5	generiere <code>httpLoadPractitionerInformationRequest(fhirVzdSearchRequest)</code>	Das PoPP-Modul generiert einen HTTP-Request (<code>httpLoadPractitionerInformationRequest</code>) mit dem erstellten FHIR-VZD-Request als Parameter.
1 6	generiere <code>authorizationDetails(fhirVzdSearchRequest)</code>	Das PoPP-Modul generiert <code>Authorization_Details</code> , diese enthalten den FHIR-VZD Search Request zur LEI.
1 7	CALL: <code>execute(httpLoadPractitionerInformationRequest, callbackLoadPractitionerInformation, authorization_details)</code>	Das PoPP-Modul ruft die <code>execute</code> -Methode am ZETA Client mit den Parametern: <ul style="list-style-type: none"> • <code>httpLoadPractitionerInformationRequest</code> - erzeugter HTTP-Request • <code>callbackLoadPractitionerInformation</code> - <code>callback</code>-Methodenname der vom ZETA Client aufzurufenden <code>callback</code>-Methode • <code>authorization_details</code> - Erzeugte <code>Authorization Details</code> auf.
1 8	GET <ZETA AS>/authorize?params...	ZETA Client sendet einen Authorization Request mit den Parametern des Authorization Code Flow und den <code>authorizationDetails{auth}</code> an den ZETA Guard Authorization-Server.
1 9	create Client Access-Token	Auf Basis der Client-Registrierung wird vom ZETA Guard ein Client

		Access-Token erzeugt.
2 0	HTTP-200: clientAccessToken	
2 1	POST: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken)	Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichem httpLoadPractitionerInformationRequest mit dem Client Access-Token auf.
2 2	reichere HTTP-Request mit Client-Informationen an	Der ZETA Guard HTTP-Proxy reicht den PoPP-Modul httpLoadPractitionerInformation-Request um Informationen zum Client (ZETA Client) aus der ZETA Guard Client-Registry an.
2 3	POST: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken, Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpLoadPractitionerInformationRequest mit den erweiterten Client-Informationen an den PoPP-Service.
2 4	GET: searchRequest(fhirVzdSearchRequest)	Der PoPP-Service führt den FHIR-VZD Search Request aus.
2 5	HTTP-200: searchResult	Der FHIR-VZD liefert dem PoPP-Service das Anfrageergebnis zurück.
2 6	createResponse(searchResult)	Der PoPP-Service erzeugt eine Response mit dem Anfrageergebnis.
2 7	HTTP-200: Response(searchResult)	Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy mit der erstellten Response.
2 8	HTTP-200: Response(searchResult)	Der ZETA Guard HTTP-Proxy leitet die Response zum ZETA Client weiter.
2 9	CALL: callbackLoadPractitionerInformation(Response(searchResult))	Der ZETA Client ruft die callback-Methode (callbackLoadPractitionerInformation) mit dem Response-Objekt als Parameter beim PoPP-Modul auf.
3 0	Filter Anzeigeergebnis(searchResult)	Das PoPP-Modul filtert das Anzeigeergebnis, dass das nur für den Nutzer relevante Daten zur Anzeige im Einwilligungsdialog kommen.
3	Zeige Consent-Dialog (Organisationsname,	Das PoPP-Modul zeigt dem

1	Organisationsanschrift)	Versicherten einen Dialog mit den Detailinformationen der LEI Organisation (Name, Adresse, ggf. weiter Informationen) an und bittet um Zugriffserlaubnis auf KVR und IK-Nummer für die LEI.
3 2	Erteilt Einwilligung	Der Versicherte erteilt seine Einwilligung. Verweigert der Versicherte seine Einwilligung, so wird der Prozess abgebrochen.
Auswahl des Authentifizierungsverfahren - GesundheitsID oder eGK ohne PIN		
3 3	CALL: hasGID?	Das PoPP-Modul ruft das Authenticator-Modul über dessen API auf um zu ermitteln, ob das Authenticator-Modul bereits an eine GesundheitsID gebunden ist.
3 4	RETURN: true/false	Das Authenticator-Modul antwortet dem PoPP-Modul true/false
3 5	Zeige Auswahl der Authentisierungsmittel (GesundheitsID, "eGK ohne PIN")	Wenn der Aufruf der Anbieter-App nicht die Authentifizierung einer eGK erzwingt und eine GesundheitsID eingerichtet (das Authenticator-Modul an eine GesundheitsID gebunden) ist, zeigt das PoPP-Modul dem Versicherten einen Dialog, wo dieser zwischen den Authentisierungsmittel "GesundheitsID" und "eGK" (ohne PIN) wählen kann. Diese Auswahl ist notwendig, da sich die jeweiligen Abläufe ab hier unterscheiden.
3 6	Wählt GesundheitsID	Der Versicherte wählt in diesem Szenario das Authentisierungsmittel "GesundheitsID", also Authentifizierung durch den sekt. IDP seiner Krankenversicherung, aus.
Ablauf Erzeugung eines PoPP-Token		
3 7	generiere httpCheckInGID-Request(telematikId, workplaceId)	Das PoPP-Modul generiert HTTP-CheckInGID-Request. Telematik-ID und WorkplaceID sind Parameter des HTTP-CheckInGID-Requests.
3	generiere	Das PoPP-Modul generiert authorizationDetails als JSON-Objekt

8	authorizationDetails(openIdProviderUrl)	welches die Client-ID des sekt. IDP (openIdProviderURL) enthält. <pre>{ "app_details": { "auth_preferences" : { "openIdProviderUrl" : "<Client-ID des sekt. IDP>" } } }</pre>
3 9	CALL: execute(httpCheckInGIDRequest, callBackCheckInGID, authorizationDetails)	Das PoPP-Modul ruft am ZETA Client die Schnittstelle execute [gemSpec_Zeta] auf. Parameter sind: <ul style="list-style-type: none"> • <i>httpCheckInGIDRequest</i> - vom PoPP-Modul erzeugter HTTP-CheckInGID-Request mit den Parametern Telematik-ID und WorkplaceID, • <i>callBackCheckInGID</i> - Methode am PoPP-Modul welche der ZETA Client nach Abschluss der Bearbeitung aufruft, • <i>authorizationDetails</i> - vom PoPP-Modul erzeugte authorizationDetails mit Informationen zur Versicherten-Authentifizierung.
4 0	GET <ZETA AS>/authorize?params...\nÖffne URL für ZETA AS	ZETA Client sendet einen Authorization Request mit den Parametern des Authorization Code Flow und den authorizationDetails{auth} an den ZETA Guard Authorization-Server.
4 1	extract openIdProviderUrl(authorization_details)	Da die Authentisierungsmethode "GesundheitsID" ist, extrahiert der ZETA Guard Authorization-Server aus den Authorization-Details die openIdProviderUrl des sektoralen IDP der gewählten Krankenversicherung.
4 2	POST <openIdProviderUrl>/par (Pushed Authorization Request)	Der ZETA Guard Authorization-Server sendet einen Pushed Authorization Request an den PAR-Endpunkt des sektoralen IDP der gewählten Krankenversicherung.
4 3	HTTP-201: request_uri	Der sektoralen IDP der gewählten Krankenversicherung erzeugt eine request_uri (URI-PAR) und sendet diese an den ZETA Guard

		Authorization-Server zurück.
4 4	HTTP-200: request_uri (Redirect Authorization Request zum IDP)	Der ZETA Guard Authorization-Server antwortet dem ZETA Client mit einem Redirect auf die request_uri.
4 5	CALL: authorize(openIdProviderUrl+request_uri)	Der ZETA Client ruft eine Methode am Authenticator-Modul mit der request_uri als Parameter auf.
4 6	GET: (openIdProviderUrl+request_uri)	Die Authenticator-App führt den Authorization Request (redirect_uri) an den Auth-Endpunkt des sektoralen IDP aus.
4 7	send Authentication Information	Der sektorale IDP übermittelt dem Authenticator-Modul der Authenticator-App die notwendigen Informationen zur Authentifizierung (z.B. Status Gerätebindung, Einwilligungen, Authentisierungsmittel).
4 8	Zeige Consent-Dialog (PoPP-Service, "Check-in" Service, o.ä.)	Wenn nicht erfolgt holt der Authenticator-Modul eine Consent-Freigabe für den PoPP-Service ein.
4 9	Erteilt Einwilligung	Der Versicherte erteilt seine Einwilligung zur Nutzung der KVNR durch den PoPP-Service. Verweigert der Versicherte die Einwilligung, wird der Prozess hier abgebrochen.
5 0	Führe Authentifizierung durch	Das Authenticator-Modul führt den Versicherten durch die Authentifizierung. Je nach Konfiguration kann die Auswahl der möglichen Authentisierungsmittel angezeigt werden oder direkt zur Erfassung der Authentisierungsnachweise der präferierten Authentifizierungsmethode.
5 1	Eingabe Authentisierungsnachweise	Der Versicherte erfasst die Authentisierungsnachweise zur ausgewählten Authentifizierungsmethode.
5 2	Sende Authentisierungsnachweise und Einwilligung an IDP Server	Das Authenticator-Modul sendet Einwilligung und Authentisierungsnachweis an den

		sekt. IDP.
5 3	führe Authentifizierung durch	Der sektorale IDP authentifiziert den Versicherten auf Basis der Authentisierungsnachweise und der ausgewählten Authentifizierungsmethode.
5 4	HTTP-302: Redirect-URL(authCodeIDP)	Der sektorale IDP erstellt einen Authorization-Code (authCodeIDP) und antwortet dem Authenticator-Modul mit einem Redirect auf die URL zum ZETA-AS
5 5	RETURN: authCodeIDP	Das Authenticator-Modul antwortet dem Aufruf des ZETA Client und übergibt den authCodeIDP
5 6	POST: (authCodeIDP)	Der ZETA Client ruft die URL des ZETA-AS auf und übermittelt so den Authorization-Code für den Token Abruf am sektoralen IDP an den ZETA-AS.
5 7	GET (authCodeIDP)	ZETA-AS ruft den ID-Token durch Übergabe des Authorization-Code (authCodeIDP) am Token-Endpoint des sektoralen IDP ab.
5 8	HTTP-200: ID-Token	Der sektorale IDP antwortet dem ZETA Guard Authorization-Server mit einem ID-Token, welches u.a. die Claims enthält: <ul style="list-style-type: none"> • <i>urn:telematik:claims:id</i> mit dem Wert der KVNR des Versicherten, • <i>urn:telematik:claims:organization</i> mit dem Wert der IK-Nummer der Krankenversicherung, • <i>amr</i> mit dem Wert der angewandten Authentisierungsmethode, • <i>acr</i> mit dem Wert des Vertrauensniveau, auf dem die Authentifizierung erfolgt ist.
5 9	prüfe ID-Token	Der ZETA Guard Authorization-Server prüft die Signatur des ID-Token und entschlüsselt den Inhalt.
6 0	saveInformation(kvnr, iknr, amr, acr)	Der ZETA Guard Authorization-Server prüft das ID-Token und

		speichert die Werte für KVNR, IK-Nummer, amr und acr aus dem ID-Token zur laufenden Session.
6 1	HTTP-200: (authCodeAS)	Der ZETA Guard Authorization-Server erzeugt einen Authorization-Code-AS (authCodeAS) und gibt diesen als Redirect auf die hinterlegte redirect_url an den ZETA Client zurück.
6 2	GET: (authCodeAS)	Der ZETA Client ruft mit dem Authorization-Code-AS (authCodeAS) das Access-Token am Token-Endpoint des ZETA Guard Authorization-Server ab.
6 3	HTTP-200: accessToken	Der ZETA Guard Authorization-Server antwortet dem ZETA Client mit finalem Access-Token.
6 4	POST: httpCheckInGIDRequest(accessToken)	Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichem HTTP-CheckInGID-Request mit dem Access-Token auf.
6 5	CALL: lade PoPP-Metainformationen zum Access Token	Der ZETA Guard HTTP-Proxy ruft, nachdem das Access-Token geprüft wurde, die zur Session (bzw. zum Access-Token) gespeicherten Metainformationen KVNR, IK-Nummer, amr, acr am ZETA Guard Authorization-Server ab.
6 6	RETURN: (kvnr, iknr, amr, acr)	Der ZETA Guard Authorization-Server antwortet dem ZETA Guard HTTP-Proxy mit den Daten.
6 7	reichere HTTP-Request mit PoPP-Metainformationen an (kvnr, iknr, amr, acr)	Der ZETA Guard HTTP-Proxy reicht den PoPP-Modul HTTP-CheckInGID-Request um die PoPP-Metainformationen an.
6 8	reichere HTTP-Request mit Client-Informationen an	Der ZETA Guard HTTP-Proxy reicht den PoPP-Modul HTTP-CheckInGID-Request um Informationen zum Client (ZETA Client) aus der ZETA Guard Client-Registry an.
6 9	POST: httpCheckInGIDRequest(telematikId, workplaceId) Header(accessToken, PoPP-Metainformation, Client-Information)	Der ZETA Guard HTTP-Proxy ruft am PoPP-Service vom ZETA Guard HTTP-Proxy erweiterten HTTP-CheckInGID-Request auf.

7 0	extractInformationen(appDetail, HTTP-Header)	<p>Der PoPP-Service extrahiert:</p> <ul style="list-style-type: none"> • aus dem Request-Header <ul style="list-style-type: none"> • die PoPP-Metainformationen zum Versicherten (KVNR, IK-Nummer) • die Informationen zur Authentifizierung (amr, acr) • Informationen zum aufrufenden Client • aus den Request Parametern <ul style="list-style-type: none"> • Telematik-ID, WorkplaceID der LEI
7 1	erstelle PoPP-Datensatz(kvnr, iknr, amr, acr, telematikId, workplaceId, status, id)	<p>Der PoPP-Service legt einen PoPP-Datensatz mit allen relevanten Informationen für die Generierung eines PoPP-Token an. Zusätzlich enthält der Datensatz</p> <ul style="list-style-type: none"> • <i>id</i> - eindeutiger Identifier des Datensatzes, dient dem Auffinden zur Abfrage oder Aktualisierung des Status zur PoPP-Token-Generierung • <i>status</i> - enthält den Status der PoPP-Token-Generierung <ul style="list-style-type: none"> • success - PoPP-Token wurde erzeugt und der LEI zugestellt • pending - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • cancelled - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen
7 2	checkLeiOnline(telematikId)	Der PoPP-Service prüft, ob die LEI zur Telematik-ID online ist.
Option: LEI ist online		
7 3	erzeuge PoppToken(poppDatenSatz)	Ist die LEI online, so wird ein PoPP-Token erstellt und eine Nachricht an das PS der LEI geschickt. Das PS der LEI triggert dann das Abholen des PoPP-Token.

7 4	versende PoPPToken(PoPPToken)	Das Primärsystem läßt den PoPP-Token
7 5	aktualisiere PoPP-Datensatz(id, status)	Der PoPP-Service aktualisiert den Status im PoPP-Datensatz auf " <i>success</i> ", wenn die Zustellung des PoPP-Token erfolgreich war.
Ende der Option: LEI online		
7 6	createResponse(statusPopTokenGeneration)	<p>Der PoPP-Service erstellt die Antwort an das PoPP-Modul</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen • <i>message</i>- Detailinformationen zum Status/ Fehlercodes • <i>id</i> - Identifier des PoPP-Datensatzes
7 7	HTTP-200: Response(statusPoPPTokenGeneration)	<p>Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy mit HTTP-200 oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht, • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen <p>sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Datensatz, welcher die</p>

		Daten und den Status zur PoPP-Token-Generierung enthält.
7 8	HTTP-200: Response(statusPoPPTokenGeneration)	<p>Der ZETA Guard HTTP-Proxy leitet die Antwort vom PoPP-Service weiter, antwortet dem ZETA Client also ebenfalls mit HTTP-200 oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht, • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen <p>sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Datensatz, welcher die Daten und den Status zur PoPP-Token-Generierung enthält.</p>
7 9	RETURN: callBackCheckInGID(Response(statusPoPPTokenGeneration))	<p>Der ZETA Client ruft am PoPP-Modul die callBackMethod auf, welche das PoPP-Modul beim execute-Aufruf dem ZETA Client übergeben hat (callBackCheckInGID). Die Response vom PoPP-Service wird dem Aufruf als Parameter mitgegeben.</p>
Start Alternative - Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
8 0	Zeige Status PoPP-Token-Generierung	Das PoPP-Modul wertet die Antwort aus und präsentiert dem Versicherten die Information.
Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
Start Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		
8	erzeuge HTTP-Request(URL=callBackAppURL,	Das PoPP-Modul erzeugt einen HTTP-

1	Response(statusPoPPTokenGeneration))	Request für den Rücksprung zur aufrufenden App. Der Status der PoPP-Token Generierung wird als Request-Parameter gesetzt.
8 2	GET: sende HTTP-GET Request(Response(statusPoPPTokenGeneratio n))	Das PoPP-Modul sendet den HTTP-Request an die Rücksprung-URL aus dem initialen Aufruf der App. Der HTTP-Request öffnet über Plattformmechanismen (deeplink, universal link) die Anbieter-App.
8 3	Verarbeite Ergebnis	Die Anbieter-App verarbeitet das Ergebnis der PoPP-Token Generierung je nach Anwendungsfall.
Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		

10.2.3 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service

Das Sequenzdiagramm "Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service" stellt den vollständigen Ablauf der PoPP-Token-Generierung für den Fall dar, dass ein Versicherter die Authentifizierung einer eGK durchführt.

Spezifikation Frontend des Versicherten für PoPP (Proof of Patient Presence)

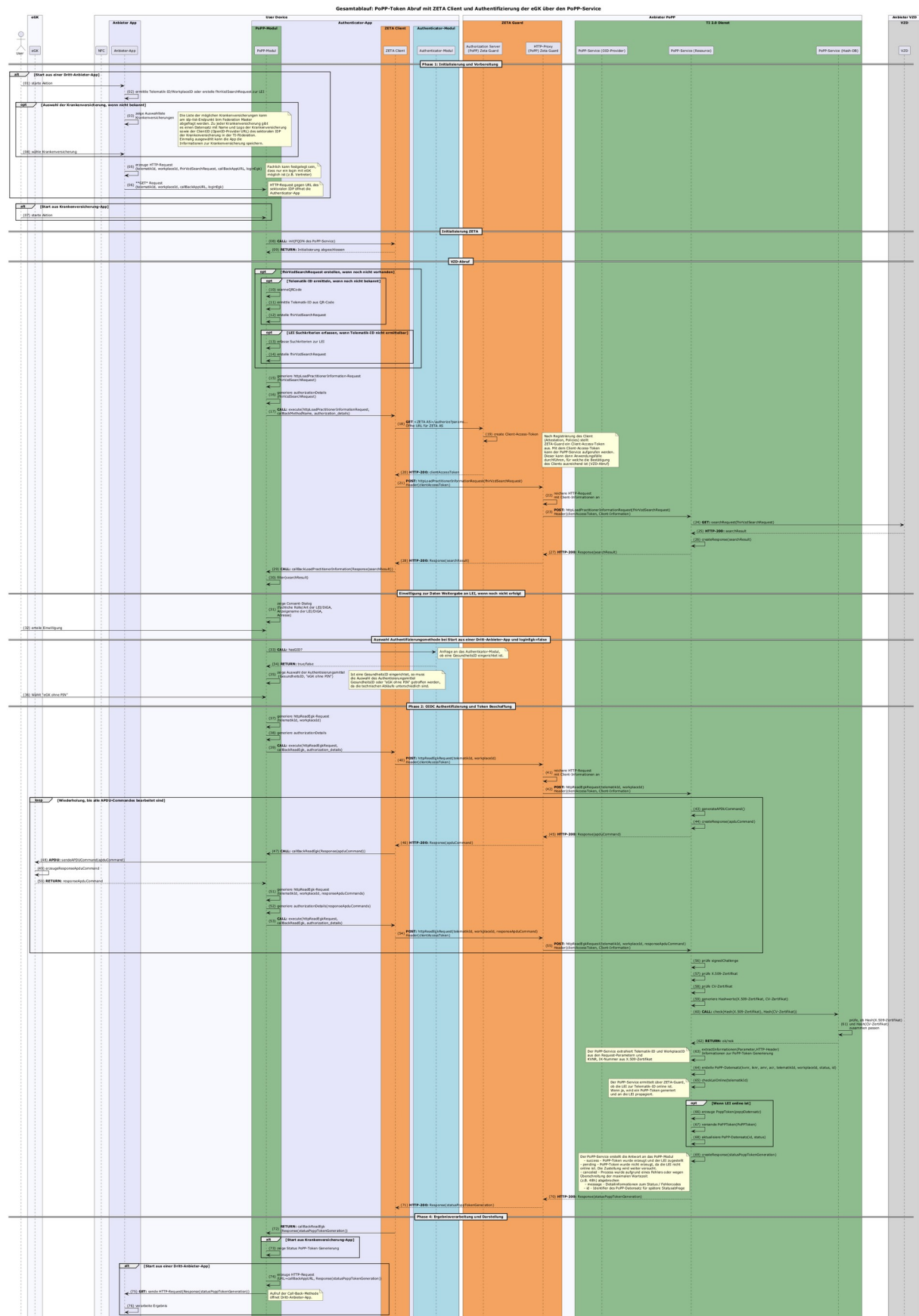


Abbildung 14: Laufzeitsicht - Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service

10.2.4 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung einer eGK

Die Tabelle [Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service] beschreibt die einzelnen Schritte vom Start der PoPP-Token-Erstellung durch den Nutzer einer Fachanwendung bis zur Darstellung des Ergebnis der PoPP-Token-Erstellung.

Tabelle 9: Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service

	Schritt	Beschreibung
Start Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		
1	Aktion starten	Der Versicherte wählt eine Funktion in der App eines Anbieters (z.B. "Anmelden zum Arztbesuch", "Rezept einlösen").
2	ermittle Telematik-ID/WorkplaceID oder erzeuge FHIR-VZD Search Request zur Datenermittlung zur LEI	<p>In der Anbieter-App werden Informationen zur LEI ermittelt, für welche das PoPP-Token bestimmt ist. Dies ist Anbieter-App spezifisch und kann auf unterschiedlichen Wegen erfolgen. z.B.</p> <ul style="list-style-type: none"> • ist bei der Anbieter-App einer LEI (online Apotheke) die Telematik-ID bekannt, • bei der Anbieter-App zur Vermittlung einer Videosprechstunde bietet die App entsprechende Auswahlfunktionen an, • eine Praxis kann die Telematik-ID als QR-Code bereitstellen. • Es werden Suchkriterien erfasst, mit denen eine FHIR-VZD-Anfrage durchgeführt werden kann
3	zeige Auswahlliste Krankenversicherungen	Die Liste der möglichen Krankenversicherungen kann am idp-list-Endpunkt beim Federation Master abgefragt werden. Zu jeder Krankenversicherung gibt es einen Datensatz mit Name und Logo der Krankenversicherung sowie der ClientID (OpenID-Provider URL) des

		sektoralen IDP der Krankenversicherung in der TI-Föderation. Einmalig ausgewählt kann die Anbieter-App die Informationen zur Krankenversicherung speichern.
4	wähle Krankenversicherung	Der Versicherte wählt seine Krankenversicherung aus der Liste aus. Die Auswahl enthält auch die ClientID (OpenID-Provider URL) des sektoralen IDP der Krankenversicherung.
5	erzeuge HTTP-GET Request(telematikId, workplaceId, fhirVzdSearchRequest callbackAppURL, loginEgk)	<p>Die Anbieter-App erzeugt einen HTTP-GET Request und übergibt als Parameter</p> <ul style="list-style-type: none"> • <i>telematikId</i> - ist die Telematik-ID der LEI, für die ein PoPP-Token erstellt werden soll • <i>workplaceId (optional)</i>- ist eine WorkplaceID (optional), wenn das PoPP-Token einem bestimmten Arbeitsplatz der LEI zugeordnet werden soll • <i>fhirVzdSearchRequest (optional)</i>- Search-Request für den Aufruf der FHIR-VZD API, um Daten zu einer LEI zu erhalten • <i>callbackURL (Rücksprung-URL)</i>- welche das PoPP-Modul aufrufen muss, wenn die PoPP-Token-Generierung abgeschlossen ist • <i>loginEgk (optional)</i>-ist die Information, ob das Login mit eGK erfolgen muss, z.B. wenn der Versicherte dies wünscht oder wenn es sich um Vertreter eines Versicherten handelt
6	sende HTTP-GET Request(telematikId, workplaceId, fhirVzdSearchRequest, callbackAppURL, loginEgk)	Die Anbieter-App sendet den HTTP-GET Request an die OpenID-Provider URL des sektoralen IDP. Die URL entspricht der ClientID des sektoralen IDP in der TI-Föderation und wurde zuvor über die Auswahl der Krankenversicherung ermittelt. Der HTTP-Request gegen die URL des sektoralen IDP öffnet die Authenticator-App. Aufgrund der Parameter im Request

		wird das PoPP-Modul in der Authenticator-App aufgerufen.
Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		
Start Alternative -Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
7	Aktion starten	Der Versicherte wählt eine Funktion in der Krankenversicherungs-App (z.B. "Praxis Check-in")
Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
8	CALL: init(FQDN des PoPP-Service)	Beim ersten Aufruf einer Session wird der ZETA Client initialisiert. Dazu wird der FQDN des PoPP-Service übergeben. Der FQDN des PoPP-Service ist ein Konfigurationsparameter des PoPP-Modul. Die ZETA-Komponenten prüfen, ob der Client (Gerät und App) bereits registriert ist. Ist das nicht der Fall, findet eine Client-Registrierung statt (Attestation, Policies) statt.
9	RETURN: Initialisierung abgeschlossen	Nach Abschluss der Initialisierung kehrt die Anwendung in das PoPP-Modul zurück.
FHIR-VZD Search Request erzeugen, wenn nicht bereits vorhanden		
Alternative: Telematik-ID ermitteln		
1 0	scanneQRCode	Ist zu diesem Zeitpunkt noch keine Telematik-ID erfasst, für die ein PoPP-Token erstellt werden soll, so muss dies in diesem Schritt erfolgen. Bei "Check-in" in einer Praxis wäre das z.B. das Scannen eines QR-Codes.
1 1	ermittle Telematik-ID aus QR-Code	Der QR-Code wird geprüft und die Telematik-ID sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.
1 2	erstelle FHIR-VZD Search Reques	Es wird ein Search-Request mit der Telematik-ID für den Aufruf der FHIR-

		VZD API erstellt, um Daten zu einer LEI zu erhalten.
Alternative: Suchkriterien erfassen		
1 3	erfasse Suchkriterien zur LEI	In diesem Schritt erfolgt die Erfassung der Suchkriterien zur LEI. Suchkriterien können z.B. bestehen aus bekannten Angaben zur LEI (Name, Fachrichtung, PLZ) und zusätzlichen Informationen (Umkreissuche).
1 4	erstelle FHIR-VZD Search Request	Es wird ein Search-Request mit den Suchkriterien für den Aufruf der FHIR-VZD API erstellt, um Daten zu einer LEI zu erhalten
Der FHIR-VZD-Abruf wird über den ZETA-Ablauf an den PoPP-Service delegiert		
1 5	generiere <code>httpLoadPractitionerInformationRequest(fhirVzdSearchRequest)</code>	Das PoPP-Modul generiert einen HTTP-Request (<code>httpLoadPractitionerInformationRequest</code>) mit dem erstellten FHIR-VZD-Request als Parameter.
1 6	generiere <code>authorizationDetails(fhirVzdSearchRequest)</code>	Das PoPP-Modul generiert <code>Authorization_Details</code> , diese enthalten den FHIR-VZD Search Request zur LEI.
1 7	CALL: <code>execute(httpLoadPractitionerInformationRequest, callbackLoadPractitionerInformation, authorization_details)</code>	Das PoPP-Modul ruft die <code>execute</code> -Methode am ZETA Client mit den Parametern <ul style="list-style-type: none"> <code>httpLoadPractitionerInformationRequest</code> - erzeugter HTTP-Request, <code>callbackLoadPractitionerInformation</code> - <code>callback</code>-Methodenname der vom ZETA Client aufzurufenden <code>callback</code>-Methode, <code>authorization_details</code> - Erzeugte <code>Authorization Details</code> auf.
1 8	GET <ZETA AS>/authorize?params...	ZETA Client sendet einen Authorization Request mit den Parametern des Authorization Code Flow und den <code>authorizationDetails{auth}</code> an den

		ZETA Guard Authorization-Server.
1 9	create Client Access-Token	Auf Basis der Client-Registrierung wird vom ZETA Guard ein Client Access-Token erzeugt.
2 0	HTTP-200: clientAccessToken	ZETA Guard gibt das Client Access-Token an ZETA Client zurück.
2 1	POST: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken)	Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichem httpLoadPractitionerInformationRequest mit dem Client Access-Token auf.
2 2	reichere HTTP-Request mit Client-Informationen an	Der ZETA Guard HTTP-Proxy reichert den PoPP-Modul httpLoadPractitionerInformation um Informationen zum Client (ZETA Client) aus der ZETA Guard Client-Registry an.
2 3	POST: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken, Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpLoadPractitionerInformationRequest mit den erweiterten Client-Informationen an den PoPP-Service.
2 4	GET: searchRequest(fhirVzdSearchRequest)	Der PoPP-Service führt den FHIR-VZD Search Request aus.
2 5	HTTP-200: searchResult	Der VZD liefert dem PoPP-Service das Suchergebnis zurück.
2 6	createResponse(searchResult)	Der PoPP-Service erzeugt eine Response mit den ermittelten Daten zur LEI.
2 7	HTTP-200: Response(searchResult)	Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy mit der erstellten Response.
2 8	HTTP-200: Response(searchResult)	Der ZETA Guard HTTP-Proxy leitet die Response zum ZETA Client weiter.
2 9	CALL: callBackLoadPractitionerInformation(Response(searchResult))	Der ZETA Client ruft die callBack-Methode (callBackLoadPractitionerInformation) mit dem Response-Objekt als Parameter beim PoPP-Modul auf.
3 0	Filter Anzeigeergebnis(searchResult)	Das PoPP-Modul filtert das Anzeigeergebnis, dass das nur für

		den Nutzer relevante Daten zur Anzeige im Einwilligungsdialog kommen.
3 1	Zeige Consent-Dialog (Organisationsname, Organisationsanschrift)	Das PoPP-Modul zeigt dem Versicherten einen Dialog mit den Detailinformationen der LEI Organisation (Name, Adresse, ggf. weitere Informationen) an und bittet um Zugriffserlaubnis auf KVNR und IK-Nummer für die LEI.
3 2	Erteilt Einwilligung	Der Versicherte erteilt seine Einwilligung. Verweigert der Versicherte seine Einwilligung, so wird der Prozess abgebrochen.
Auswahl des Authentifizierungsverfahren - GesundheitsID oder eGK ohne PIN		
3 3	CALL: hasGID?	Das PoPP-Modul ruft das Authenticator-Modul über dessen API auf um zu ermitteln, ob das Authenticator-Modul bereits an eine GesundheitsID gebunden ist.
3 4	RETURN: true/false	Das Authenticator-Modul antwortet dem PoPP-Modul true/false
3 5	Zeige Auswahl der Authentisierungsmittel (GesundheitsID, "eGK ohne PIN")	Wenn der Aufruf der Anbieter-App nicht die Authentifizierung einer eGK erzwingt und eine GesundheitsID eingerichtet (das Authenticator-Modul an eine GesundheitsID gebunden) ist, zeigt das PoPP-Modul dem Versicherten einen Dialog, wo dieser zwischen den Authentisierungsmittel "GesundheitsID" und "eGK" (ohne PIN) wählen kann. Diese Auswahl ist notwendig, da sich die jeweiligen Abläufe ab hier unterscheiden.
3 6	Wählt "eGK ohne PIN"	Der Versicherte wählt in diesem Szenario das Authentisierungsmittel "eGK ohne PIN", also Authentifizierung einer eGK.
Ablauf Erzeugung eines PoPP-Token		
3 7	generiere httpReadEgk-Request(telematikId, workplaceId)	Das PoPP-Modul generiert HTTP-Read-eGK-Request. Telematik-ID und WorkplaceID sind Parameter des HTTP-HTTP-Read-eGK-Request.

3 8	generiere authorizationDetails()	Die authorization_details enthalten für diesen Aufruf keine Werte.
3 9	execute(httpReadEgkRequest, callbackReadEgk, authorizationDetails)	<p>Das PoPP-Modul ruft am ZETA Client die Schnittstelle execute [gemSpec_Zeta] auf. Parameter sind:</p> <ul style="list-style-type: none"> • <i>httpReadEgkRequest</i> - vom PoPP-Modul erzeugter HTTP-Read-eGK-Request mit den Parametern Telematik-ID und WorkplaceID, • <i>callbackReadEgk</i> - Methode am PoPP-Modul welche der ZETA Client als Ergebnis des HTTP-Requests aufruft, • <i>authorizationDetails</i> - vom PoPP-Modul erzeugte authorizationDetails.
4 0	httpReadEgkRequest(telematikId, workplaceId) Header(clientAccessToken)	Der ZETA Client sendet den HTTP-Request (httpReadEgkRequest) mit dem Client Access-Token an den ZETA Guard HTTP-Proxy.
4 1	reichere HTTP-Request mit Client-Informationen an	Der ZETA Guard HTTP-Proxy reichert den HTTP-Request um Informationen zum Client aus der Client-Registrierung an.
4 2	httpReadEgkRequest(telematikId, workplaceId) Header(clientAccessToken, Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpReadEgkRequest mit den erweiterten Client-Informationen an den PoPP-Service.
loop Wiederholung, bis alle APDU-Commands bearbeitet sind		
4 3	generateAPDUCommand()	<p>Wird eine Antwort der eGK auf das APDU-Command (responseApduCommand) als Parameter des HTTP-Request übergeben, wertet der PoPP-Service diese aus.</p> <p>Der PoPP-Service generiert ein APDU-Command für die Ausführung an der eGK.</p>
4 4	createResponse(apduCommand)	Der PoPP-Service generiert eine Response mit dem APDU-Command
4	HTTP-200 Response(apduCommand)	Der PoPP-Service antwortet dem

5		ZETA Guard HTTP-Proxy auf den Request mit HTTP-200 und der generierten Response
4 6	HTTP-200 Response(apduCommand)	Der ZETA Guard HTTP-Proxy leitet die Antwort an den ZETA Client weiter
4 7	callBackReadEgk(Response(apduCommand))	Der ZETA Client ruft die callBack-Methode (callBackReadEgk) mit dem Response-Objekt am PoPP-Modul auf
4 8	sendeAPDUCommand(apduCommand)	Das PoPP-Modul extrahiert das APDU-Command aus Response-Parameter und sendet es über die NFC-Schnittstelle des Gerätes an die eGK
4 9	erzeugeResponseApduCommand	Die eGK verarbeitet das APDU-Command und generiert eine Antwort
5 0	RETURN responseApduCommand	Die eGK beantwortet die Anfrage des PoPP-Modul (responseApduCommand)
5 1	generiere httpReadEgk-Request(telematikId, workplaceId, responseApduCommand)	Das PoPP-Modul generiert einen neuen httpReadEgk-Request mit dem zusätzlichen Parameter responseApduCommand
5 2	generiere authorizationDetails(responseApduCommand)	Das PoPP-Modul generiert authorization_details mit dem responseApduCommand
5 3	execute(httpReadEgkRequest, callBackReadEgk, authorization_details)	<p>Das PoPP-Modul ruft am ZETA Client die Schnittstelle execute [gemSpec_Zet a] auf. Parameter sind:</p> <ul style="list-style-type: none"> • <i>httpReadEgkRequest</i> - vom PoPP-Modul erzeugter HTTP-Read-eGK-Request mit den Parametern Telematik-ID, WorkplaceID und der Antwort der eGK auf das APDU-Command (responseApduCommand), • <i>callBackReadEgk</i> - Methode am PoPP-Modul welche der ZETA Client als Ergebnis des HTTP-Requests aufruft, • <i>authorizationDetails</i> - vom PoPP-

		Modul erzeugte authorizationDetails.
5 4	httpReadEgkRequest(telematikId, workplaceld, responseApduCommand) Header(clientAccessToken)	Der ZETA Client sendet den HTTP- Request (httpReadEgkRequest) mit dem Client Access Token an den ZETA Guard HTTP-Proxy
5 5	httpReadEgkRequest(telematikId, workplaceld, responseApduCommand) Header(clientAccessToken, Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpReadEgkRequest mit den erweiterten Client-Informationen an den PoPP-Service
Ende der Wiederholungsschritte		
5 6	prüfe signedChallenge	Der PoPP-Service prüft die Antwort der eGK auf das APDU-Command zur Signatur einer Challenge mit dem CV-Zertifikat der eGK.
5 7	prüfe X.509-Zertifikat	Der PoPP-Service prüft, ob das X.509-Zertifikat aus der Antwort der eGK gültig ist.
5 8	prüfe CV-Zertifikat	Der PoPP-Service prüft, ob das CV- Zertifikat aus der Antwort der eGK gültig ist.
5 9	RS: generiere Hashwerte(X.509-Zertifikat, CV- Zertifikat)	Der PoPP-Service generiert Hashwerte zu den Zertifikaten.
6 0	check(Hash(X.509-Zertifikat), Hash(CV- Zertifikat))	Der PoPP-Service ruft die Hash-DB mit den Hashwerten der Zertifikate auf.
6 1	prüfe, ob Hash(X.509-Zertifikat) und Hash(CV- Zertifikat) zusammen passen	Die Hash-DB prüft, ob die übergebenen Hashwerte als gültiges Paar hinterlegt sind. Ist das der Fall, so gehören die Zertifikate zur gleichen eGK
6 2	RETURN ok/nok	Die Hash-DB liefert dem PoPP- Service das Ergebnis der Prüfung
6 3	extractInformationen(Parameter, HTTP- Header) Informationen zur PoPP-Token Generierung	Der PoPP-Service extrahiert: <ul style="list-style-type: none"> • aus dem Request-Header <ul style="list-style-type: none"> • die PoPP-Metainformationen zum Versicherten (KVNR, IK- Nummer) • die Informationen zur

		<p>Authentifizierung (amr, acr)</p> <ul style="list-style-type: none"> • Informationen zum aufrufenden Client • aus den Request Parametern • Telematik-ID, WorkplaceID der LEI
6 4	erstelle PoPP-Datensatz(kvnr, iknr, amr, acr, telematikId, workplaceId, status, id)	<p>Der PoPP-Service legt einen PoPP-Datensatz mit allen relevanten Informationen für die Generierung eines PoPP-Token an. Zusätzlich enthält der Datensatz</p> <ul style="list-style-type: none"> • <i>id</i> - eindeutiger Identifier des Datensatzes, dient dem Auffinden zur Abfrage oder Aktualisierung des Status zur PoPP-Token-Generierung • <i>status</i> - enthält den Status der PoPP-Token-Generierung <ul style="list-style-type: none"> • success - PoPP-Token wurde erzeugt und der LEI zugestellt • pending - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • cancelled - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen
6 5	checkLeiOnline(telematikId)	Der PoPP-Service prüft, ob die LEI zur Telematik-ID online ist.
Option: LEI ist online		
6 6	erzeuge PoppToken(poppDatenSatz)	Ist die LEI online, so wird ein PoPP-Token erstellt und eine Nachricht an das PS der LEI geschickt. Das PS der LEI triggert dann das Abholen des PoPP-Token.
6 7	versende PoPPToken(PoPPToken)	Das Primärsystem läßt den PoPP-Token
6 8	aktualisiere PoPP-Datensatz(id, status)	Der PoPP-Service aktualisiert den Status im PoPP-Datensatz auf "success", wenn die Zustellung des PoPP-Token erfolgreich war.

Ende der Option: LEI online		
6 9	createResponse(statusPopTokenGeneration)	<p>Der PoPP-Service erstellt die Antwort an das PoPP-Modul:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen, • <i>message</i> - Detailinformationen zum Status / Fehlercodes. • <i>id</i> - Identifier des PoPP-Datensatzes
7 0	HTTP-200: Response(statusPoPPTokenGeneration)	<p>Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy mit HTTP-200 oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht, • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen <p>sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Datensatz, welcher die Daten und den Status zur PoPP-Token-Generierung enthält.</p>
7 1	HTTP-200: Response(statusPoPPTokenGeneration)	<p>Der ZETA Guard HTTP-Proxy leitet die Antwort vom PoPP-Service weiter, antwortet dem ZETA Client also ebenfalls mit HTTP-200 oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-</p>

		<p>Generierung. Der Status besteht aus einen der Statuscodes:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen <p>sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Datensatz, welcher die Daten und den Status zur PoPP-Token-Generierung enthält.</p>
7 2	<p>RETURN: callBackReadEgk(Response(statusPoPPToken Generation))</p>	<p>Der ZETA Client ruft am PoPP-Modul die callBackMethod auf, welche das PoPP-Modul beim execute-Aufruf dem ZETA Client übergeben hat (callBackReadEgk). Die Response vom PoPP-Service wird dem Aufruf als Parameter mitgegeben.</p>
Start Alternative - Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
7 3	<p>Zeige Status PoPP-Token-Generierung</p>	<p>Das PoPP-Modul wertet die Antwort aus und präsentiert dem Versicherten die Information.</p>
Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
Start Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		
7 4	<p>erzeuge HTTP-Request(URL=callBackAppURL, Response(statusPoPPTokenGeneration))</p>	<p>Das PoPP-Modul erzeugt einen HTTP-Request für den Rücksprung zur aufrufenden App. Der Status der PoPP-Token-Generierung wird als Request-Parameter gesetzt.</p>
7 5	<p>GET: sende HTTP-GET Request(Response(statusPoPPTokenGeneratio n))</p>	<p>Das PoPP-Modul sendet den HTTP-Request an die Rücksprung-URL aus dem initialen Aufruf der App. Der HTTP-Request öffnet über</p>

		Plattformmechanismen (deeplink, universal link) die Anbieter-App.
7 6	Verarbeite Ergebnis	Die Anbieter-App verarbeitet das Ergebnis der PoPP-Token-Generierung je nach Anwendungsfall.
Ende Alternative - Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		

10.2.5 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service mit PoPP-Modul in einer Drittanbieter-App

Das Sequenzdiagramm "Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service" stellt den vollständigen Ablauf der PoPP-Token-Generierung für den Fall dar, dass ein Versicherter die Authentifizierung einer eGK durchführt und PoPP-Modul und ZETA Client in der Drittanbieter-App implementiert sind.

In diesem Ablauf findet kein APP-Sprung statt.

Spezifikation Frontend des Versicherten für PoPP (Proof of Patient Presence)

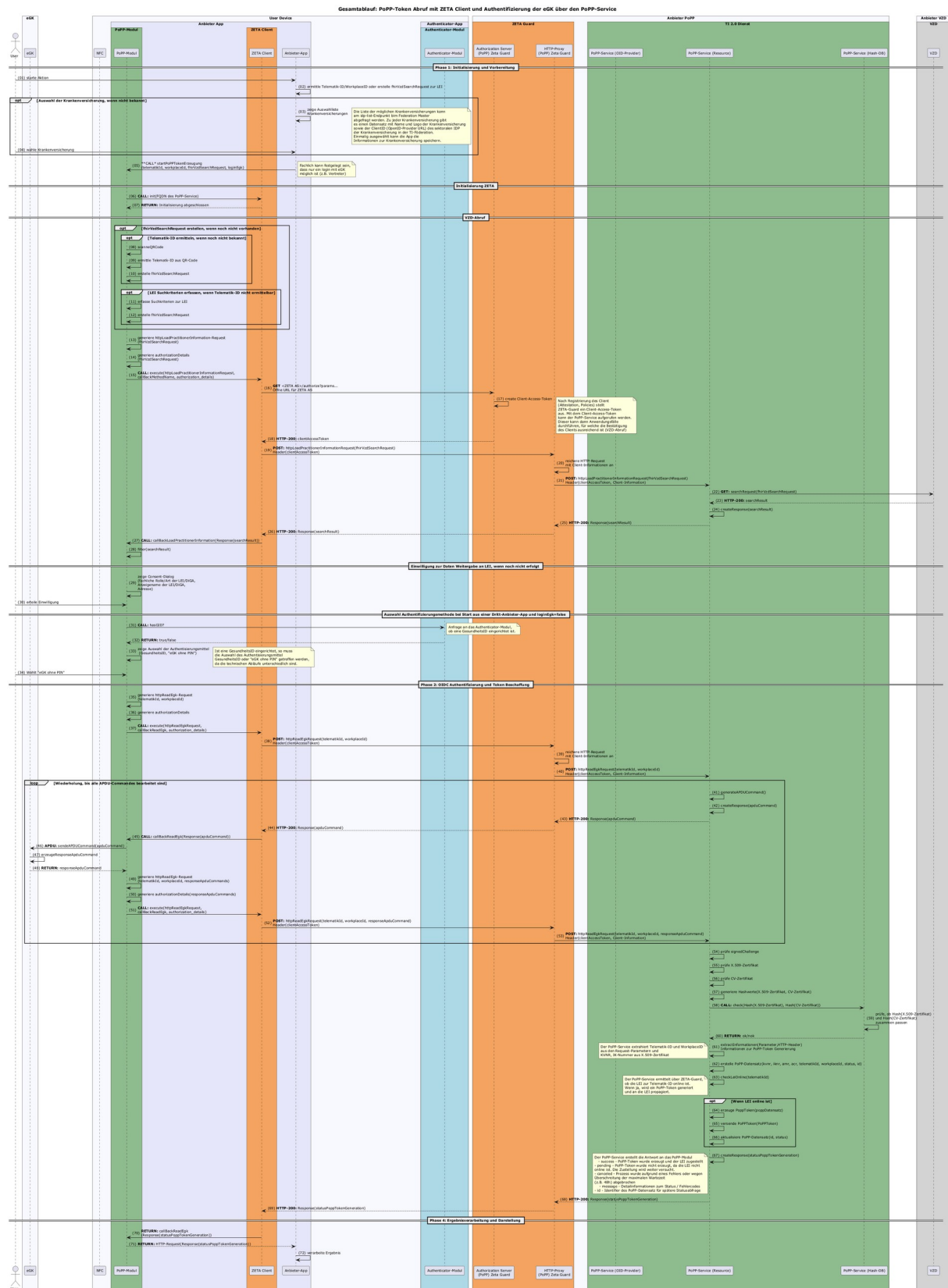


Abbildung 15: Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service

10.2.6 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung einer eGK mit PoPP-Modul in einer Drittanbieter-App

Die Tabelle [Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service] beschreibt die einzelnen Schritte vom Start der PoPP-Token-Erstellung durch den Nutzer einer Fachanwendung bis zur Darstellung des Ergebnisses der PoPP-Token-Erstellung.

Tabelle 10: Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service

Schritt	Beschreibung
1 Aktion starten	Der Versicherte wählt eine Funktion in der App eines Anbieters (z.B. "Anmelden zum Arztbesuch", "Rezept einlösen").
2 ermittle Telematik-ID/WorkplaceID oder erzeuge FHIR-VZD Search Request zur Datenermittlung zur LEI	<p>In der Anbieter-App werden Informationen zur LEI ermittelt, für welche das PoPP-Token bestimmt ist. Dies ist Anbieter-App spezifisch und kann auf unterschiedlichen Wegen erfolgen. z.B.:</p> <ul style="list-style-type: none"> • ist bei der Anbieter-App einer LEI (online Apotheke) die Telematik-ID bekannt, • bei der Anbieter-App zur Vermittlung einer Videosprechstunde bietet die App entsprechende Auswahlfunktionen an, • eine Praxis kann die Telematik-ID als QR-Code bereitstellen. • Es werden Suchkriterien erfasst, mit denen eine FHIR-VZD Anfrage durchgeführt werden kann.
3 zeige Auswahlliste Krankenversicherungen	Die Liste der möglichen Krankenversicherungen kann am idp-list-Endpunkt beim Federation Master abgefragt werden. Zu jeder Krankenversicherung gibt es einen Datensatz mit Namen und Logo der Krankenversicherung sowie der ClientID (OpenID-Provider URL) des sektoralen IDP der Krankenversicherung in der TI-Föderation. Einmalig ausgewählt kann die Anbieter-App die

		Informationen zur Krankenversicherung speichern.
4	wähle Krankenversicherung	Der Versicherte wählt seine Krankenversicherung aus der Liste aus. Die Auswahl enthält auch die ClientID (OpenID-Provider URL) des sektoralen IDP der Krankenversicherung.
5	erzeuge HTTP-GET Request(telematikId, workplaceId, fhirVzdSearchRequest, callBackAppURL, loginEgk)	<p>Die Anbieter-App ruft eine API-Methode am PoPP-Modul zum startPoPPTokenErzeugung auf und übergibt als Parameter:</p> <ul style="list-style-type: none"> • <i>telematikId</i> ist die Telematik-ID der LEI, für die ein PoPP-Token erstellt werden soll, • <i>workplaceId (optional)</i> ist eine WorkplaceID (optional), wenn das PoPP-Token einem bestimmten Arbeitsplatz der LEI zugeordnet werden soll, • <i>fhirVzdSearchRequest (optional)</i>-Search-Request für den Aufruf der FHIR-VZD API, um Daten zu einer LEI zu erhalten • <i>loginEgk (optional)</i> ist die Information, ob das Login mit eGK erfolgen muss, z.B. wenn der Versicherte dies wünscht oder wenn es sich um Vertreter eines Versicherten handelt.
6	CALL: init(FQDN des PoPP-Service)	<p>Beim ersten Aufruf einer Session wird der ZETA Client initialisiert. Dazu wird der FQDN des PoPP-Service übergeben. Der FQDN des PoPP-Service ist ein Konfigurationsparameter des PoPP-Modul.</p> <p>Die ZETA-Komponenten prüfen, ob der Client (Gerät und App) bereits registriert ist. Ist das nicht der Fall, findet eine Client-Registrierung statt (Attestation, Policies) statt.</p>
7	RETURN: Initialisierung abgeschlossen	Nach Abschluss der Initialisierung kehrt die Anwendung in das PoPP-Modul zurück.
FHIR-VZD Search Request erzeugen, wenn nicht bereits vorhanden		

Alternative: Telematik-ID ermitteln		
8	scanneQRCode	Ist zu diesem Zeitpunkt noch keine Telematik-ID erfasst, für die ein PoPP-Token erstellt werden soll, so muss dies in diesem Schritt erfolgen. Bei "Check-in" in einer Praxis wäre das z.B. das Scannen eines QR-Codes.
9	ermittle Telematik-ID aus QR-Code	Der QR-Code wird geprüft und die Telematik-ID sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.
1 0	erstelle FHIR-VZD Search Request	Es wird ein Search-Request mit der Telematik-ID für den Aufruf der FHIR-VZD API erstellt, um Daten zu einer LEI zu erhalten
Alternative: Suchkriterien erfassen		
1 1	erfasse Suchkriterien zur LEI	In diesem Schritt erfolgt die Erfassung der Suchkriterien zur LEI. Suchkriterien können z.B. bestehen aus bekannten Angaben zur LEI (Name, Fachrichtung, PLZ) und zusätzlichen Informationen (Umkreissuche).
1 2	erstelle FHIR-VZD Search Request	Es wird ein Search-Request mit den Suchkriterien für den Aufruf der FHIR-VZD API erstellt, um Daten zu einer LEI zu erhalten
Der FHIR-VZD-Abruf wird über den ZETA-Ablauf an den PoPP-Service delegiert		
1 3	generiere httpLoadPractitionerInformation-Request(fhirVzdSearchRequest)	Das PoPP-Modul generiert einen HTTP-Request (httpLoadPractitionerInformationRequest) mit dem erstellten FHIR-VZD-Request als Parameter.
1 4	generiere authorizationDetails(fhirVzdSearchRequest)	Das PoPP-Modul generiert Authorization_Details, diese enthalten den FHIR-VZD Search Request zur LEI.
1 5	CALL: execute(httpLoadPractitionerInformationRequest, callBackLoadPractitionerInformation, authorization_details)	Das PoPP-Modul ruft die execute-Methode am ZETA Client mit den Parametern: <ul style="list-style-type: none"> httpLoadPractitionerInformationR

		<p>equest - erzeugter HTTP-Request</p> <ul style="list-style-type: none"> • callbackLoadPractitionerInformation - callback-Methodenname der vom ZETA Client aufzurufenden callback-Methode • authorization_details - Erzeugte Authorization Details <p>auf.</p>
1 6	GET <ZETA AS>/authorize?params...	ZETA Client sendet einen Authorization Request mit den Parametern des Authorization Code Flow und den authorizationDetails{auth} an den ZETA Guard Authorization-Server.
1 7	create Client Access-Token	Auf Basis der Client-Registrierung wird vom ZETA Guard ein Client Access-Token erzeugt.
1 8	HTTP-200: clientAccessToken	
1 9	POST: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken)	Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichem httpLoadPractitionerInformationRequest mit dem Client Access-Token auf.
2 0	reichere HTTP-Request mit Client-Informationen an	Der ZETA Guard HTTP-Proxy reicht den PoPP-Modul httpLoadPractitionerInformation-Request um Informationen zum Client (ZETA Client) aus der ZETA Guard Client-Registry an.
2 1	POST: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken, Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpLoadPractitionerInformationRequest mit den erweiterten Client-Informationen an den PoPP-Service.
2 2	GET: searchRequest(fhirVzdSearchRequest)	Der PoPP-Service führt den FHIR-VZD Search Request aus.
2 3	HTTP-200: searchResult	Der FHIR-VZD liefert dem PoPP-Service das Anfrageergebnis zurück.
2 4	createResponse(searchResult)	Der PoPP-Service erzeugt eine Response mit dem Anfrageergebnis.
2 5	HTTP-200: Response(searchResult)	Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy mit der

		erstellten Response.
2 6	HTTP-200: Response(searchResult)	Der ZETA Guard HTTP-Proxy leitet die Response zum ZETA Client weiter.
2 7	CALL: callBackLoadPractitionerInformation(Response (searchResult))	Der ZETA Client ruft die callBack-Methode (callBackLoadPractitionerInformation) mit dem Response-Objekt als Parameter beim PoPP-Modul auf.
2 8	Filter Anzeigeergebnis(searchResult)	Das PoPP-Modul filtert das Anzeigeergebnis, dass das nur für den Nutzer relevante Daten zur Anzeige im Einwilligungsdialo kommen.
2 9	Zeige Consent-Dialog (Organisationsname, Organisationsanschrift)	Das PoPP-Modul zeigt dem Versicherten einen Dialog mit den Detailinformationen der LEI Organisation (Name, Adresse, ggf. weitere Informationen) an und bittet um Zugriffserlaubnis auf KVN und IK-Nummer für die LEI.
3 0	Erteilt Einwilligung	Der Versicherte erteilt seine Einwilligung. Verweigert der Versicherte seine Einwilligung, so wird der Prozess abgebrochen.
Auswahl des Authentifizierungsverfahren - GesundheitsID oder eGK ohne PIN		
3 1	CALL: hasGID?	Das PoPP-Modul ruft das Authenticator-Modul über dessen API auf um zu ermitteln, ob das Authenticator-Modul bereits an eine GesundheitsID gebunden ist.
3 2	RETURN: true/false	Das Authenticator-Modul antwortet dem PoPP-Modul true/false.
3 3	Zeige Auswahl der Authentisierungsmittel (GesundheitsID, "eGK ohne PIN")	Wenn der Aufruf der Anbieter-App nicht die Authentifizierung einer eGK erzwingt und eine GesundheitsID eingerichtet (das Authenticator-Modul an eine GesundheitsID gebunden) ist, zeigt das PoPP-Modul dem Versicherten einen Dialog, wo dieser zwischen den Authentisierungsmittel "GesundheitsID" und "eGK" (ohne PIN) wählen kann. Diese Auswahl ist

		notwendig, da sich die jeweiligen Abläufe ab hier unterscheiden.
3 4	Wählt "eGK ohne PIN"	Der Versicherte wählt in diesem Szenario das Authentisierungsmittel "eGK ohne PIN", also Authentifizierung einer eGK.
Ablauf Erzeugung eines PoPP-Token		
3 5	generiere httpReadEgk-Request(telematikId, workplaceld)	Das PoPP-Modul generiert HTTP-Read-eGK-Request. Telematik-ID und Workplaceld sind Parameter des HTTP-HTTP-Read-eGK-Request.
3 6	generiere authorizationDetails()	Die authorization_details enthalten für diesen Aufruf keine Werte.
3 7	execute(httpReadEgkRequest, callBackReadEgk, authorizationDetails)	Das PoPP-Modul ruft am ZETA Client die Schnittstelle execute [gemSpec_Zeta] auf. Parameter sind: <ul style="list-style-type: none"> • httpReadEgkRequest - vom PoPP-Modul erzeugter HTTP-Read-eGK-Request mit den Parametern Telematik-ID und Workplaceld, • callBackReadEgk - Methode am PoPP-Modul welche der ZETA Client als Ergebnis des HTTP-Requests aufruft, • authorizationDetails - vom PoPP-Modul erzeugte authorizationDetails.
3 8	httpReadEgkRequest(telematikId, workplaceld) Header(clientAccessToken)	Der ZETA Client sendet den HTTP-Request (httpReadEgkRequest) mit dem Client Access-Token an den ZETA Guard HTTP-Proxy.
3 9	reichere HTTP-Request mit Client-Informationen an	Der ZETA Guard HTTP-Proxy reicht den HTTP-Request um Informationen zum Client aus der Client-Registrierung an.
4 0	httpReadEgkRequest(telematikId, workplaceld) Header(clientAccessToken, Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpReadEgkRequest mit den erweiterten Client-Informationen an den PoPP-Service.
loop Wiederholung, bis alle APDU-Commands bearbeitet sind		

4 1	<code>generateAPDUCommand()</code>	Wird eine Antwort der eGK auf das APDU-Command (<code>responseApduCommand</code>) als Parameter des HTTP-Request übergeben, wertet der PoPP-Service diese aus. Der PoPP-Service generiert ein APDU-Command für die Ausführung an der eGK.
4 2	<code>createResponse(apduCommand)</code>	Der PoPP-Service generiert eine Response mit dem APDU-Command.
4 3	<code>HTTP-200 Response(apduCommand)</code>	Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy auf den Request mit HTTP-200 und der generierten Response.
4 4	<code>HTTP-200 Response(apduCommand)</code>	Der ZETA Guard HTTP-Proxy leitet die Antwort an den ZETA Client weiter.
4 5	<code>callBackReadEgk(Response(apduCommand))</code>	Der ZETA Client ruft die <code>callBack</code> -Methode (<code>callBackReadEgk</code>) mit dem Response-Objekt am PoPP-Modul auf.
4 6	<code>sendeAPDUCommand(apduCommand)</code>	Das PoPP-Modul extrahiert das APDU-Command aus Response-Parameter und sendet es über die NFC-Schnittstelle des Gerätes an die eGK.
4 7	<code>erzeugeResponseApduCommand</code>	Die eGK verarbeitet das APDU-Command und generiert eine Antwort.
4 8	<code>RETURN responseApduCommand</code>	Die eGK beantwortet die Anfrage des PoPP-Moduls (<code>responseApduCommand</code>).
4 9	<code>generiere httpReadEgk-Request(telematikId, workplaceld, responseApduCommand)</code>	Das PoPP-Modul generiert einen neuen <code>httpReadEgk-Request</code> mit dem zusätzlichen Parameter <code>responseApduCommand</code> .
5 0	<code>generiere authorizationDetails(responseApduCommand)</code>	Das PoPP-Modul generiert <code>authorization_details</code> mit dem <code>responseApduCommand</code> .
5 1	<code>execute(httpReadEgkRequest, callBackReadEgk, authorization_details)</code>	Das PoPP-Modul ruft am ZETA Client die

		<p>Schnittstelle execute [gemSpec_Zeta] auf. Parameter sind:</p> <ul style="list-style-type: none"> • httpReadEgkRequest - vom PoPP-Modul erzeugter HTTP-Read-eGK-Request mit den Parametern Telematik-ID, WorkplaceID und der Antwort der eGK auf das APDU-Command (responseApduCommand), • callBackReadEgk - Methode am PoPP-Modul welche der ZETA Client als Ergebnis des HTTP-Requests aufruft, • authorizationDetails - vom PoPP-Modul erzeugte authorizationDetails.
5 2	httpReadEgkRequest(telematikId, workplaceId, responseApduCommand) Header(clientAccessToken)	Der ZETA Client sendet den HTTP-Request (httpReadEgkRequest) mit dem Client Access-Token an den ZETA Guard HTTP-Proxy.
5 3	httpReadEgkRequest(telematikId, workplaceId, responseApduCommand) Header(clientAccessToken, Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpReadEgkRequest mit den erweiterten Client-Informationen an den PoPP-Service.
Ende der Wiederholungsschritte		
5 4	prüfe signedChallenge	Der PoPP-Service prüft die Antwort der eGK auf das APDU-Command zur Signatur einer Challenge mit dem CV-Zertifikat der eGK.
5 5	prüfe X.509-Zertifikat	Der PoPP-Service prüft, ob das X.509-Zertifikat aus der Antwort der eGK gültig ist.
5 6	prüfe CV-Zertifikat	Der PoPP-Service prüft, ob das CV-Zertifikat aus der Antwort der eGK gültig ist.
5 7	RS: generiere Hashwerte(X.509-Zertifikat, CV-Zertifikat)	Der PoPP-Service generiert Hashwerte zu den Zertifikaten.
5 8	check(Hash(X.509-Zertifikat), Hash(CV-Zertifikat))	Der PoPP-Service ruft die Hash-DB mit den Hashwerten der Zertifikate auf.
5 9	prüfe, ob Hash(X.509-Zertifikat) und Hash(CV-Zertifikat) zusammen passen	Die Hash-DB prüft, ob die übergebenen Hashwerte als gültiges

		Paar hinterlegt sind. Ist das der Fall, so gehören die Zertifikate zur gleichen eGK.
6 0	RETURN ok/nok	Die Hash-DB liefert dem PoPP-Service das Ergebnis der Prüfung.
6 1	extractInformationen(Parameter, HTTP-Header) Informationen zur PoPP-Token Generierung	
6 2	erstelle PoPP-Datensatz(kvnr, iknr, amr, acr, telematikId, workplaceld, status, id)	<p>Der PoPP-Service legt einen PoPP-Datensatz mit allen relevanten Informationen für die Generierung eines PoPP-Token an. Zusätzlich enthält der Datensatz</p> <ul style="list-style-type: none"> • <i>id</i> - eindeutiger Identifier des Datensatzes, dient dem Auffinden zur Abfrage oder Aktualisierung des Status zur PoPP-Token-Generierung • <i>status</i> - enthält den Status der PoPP-Token-Generierung <ul style="list-style-type: none"> • success - PoPP-Token wurde erzeugt und der LEI zugestellt • pending - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • cancelled - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen
6 3	checkLeiOnline(telematikId)	Der PoPP-Service prüft, ob die LEI zur Telematik-ID online ist.
Option: LEI ist online		
6 4	erzeuge PoppToken(poppDatenSatz)	Ist die LEI online, so wird ein PoPP-Token erstellt und eine Nachricht an das PS der LEI geschickt. Das PS der LEI triggert dann das Abholen des PoPP-Token.
6 5	versende PopPToken(PopPToken)	Das Primärsystem ließt den PoPP-Token
6	aktualisiere PoPP-Datensatz(id, status)	Der PoPP-Service aktualisiert den

6		Status im PoPP-Datensatz auf "success", wenn die Zustellung des PoPP-Token erfolgreich war.
Ende der Option: LEI online		
6 7	createResponse(statusPopTokenGeneration)	<p>Der PoPP-Service erstellt die Antwort an das PoPP-Modul:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen, • <i>message</i> - Detailinformationen zum Status / Fehlercodes. • <i>id</i> - Identifier des PoPP-Datensatzes
6 8	HTTP-200: Response(statusPoPPTokenGeneration)	<p>Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy mit HTTP-200 oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • <i>cancelled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen <p>sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Datensatz, welcher die Daten und den Status zur PoPP-Token Generierung enthält.</p>
6	HTTP-200:	Der ZETA Guard HTTP-Proxy leitet

9	Response(statusPoPPTokenGeneration)	<p>die Antwort vom PoPP-Service weiter, antwortet dem ZETA Client also ebenfalls mit HTTP-200 oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes:</p> <ul style="list-style-type: none"> • <i>success</i> - PoPP-Token wurde erzeugt und der LEI zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist. Die Zustellung wird weiter versucht. • <i>cancelled</i>- Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen <p>sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Datensatz, welcher die Daten und den Status zur PoPP-Token-Generierung enthält.</p>
7 0	RETURN: callBackReadEgk(Response(statusPoPPTokenGeneration))	<p>Der ZETA Client ruft am PoPP-Modul die callBackMethod auf, welche das PoPP-Modul beim execute-Aufruf dem ZETA Client übergeben hat (callBackReadEgk). Die Response vom PoPP-Service wird dem Aufruf als Parameter mitgegeben.</p>
7 1	RETURN: Request(Response(statusPoPPTokenGeneration))	<p>Das PoPP-Modul sendet als Antwort auf den initialen Aufruf der App Response(statusPoPPTokenGeneration).</p>
7 2	Verarbeite Ergebnis	<p>Die Anbieter-App verarbeitet das Ergebnis der PoPP-Token-Generierung je nach Anwendungsfall.</p>

1383

11 Anhang C - Offene Punkte, Fragen

11.1 Offene Punkte

ID	Fundstelle	Beschreibung	
OP-PoPP-3	Kap. 2.1 (PoPP-Modul)	<i>Im Folgenden wird an verschiedenen Stellen auf die ZETA Client-Registrierung verwiesen. Diesbezüglich gibt es für die eGK-Anwendungsfälle einen offenen Punkt. Verwendet der Nutzer ausschließlich die eGK (also keine GesundheitsID) wird bzgl. ZETA eine rein Client-gebundene Client-Registrierung also ohne Authentifizierung des Nutzers benötigt. Dies wird aktuell noch in den ZETA-Spezifikationen umgesetzt. Die bisher in gemSpec_ZETA#5.2.3 beschriebene Client-Registrierung ist Nutzer-gebunden und erfordert daher zwingend die Authentifizierung des Nutzers. Letztere kann bei ausschließlichem Vorhandensein der eGK nicht stattfinden. Der Entwurf für die rein Client-gebundene Registrierung wird parallel bzw. zeitnah veröffentlicht.</i>	
OP-PoPP-2	Kap 8.1 (Testkonzept)	<i>Es soll eine Testtreiberschnittstelle analog zum Vorgehen beim ePA-FdV für PoPP-Modul/ App-mit-Popp-Modul entwickelt werden. Die entsprechenden Festlegungen werden in [gemKPT_Test] an geeigneter Stelle aufgenommen werden..</i>	