

Telematikinfrastruktur 2.0

Spezifikation PoPP (Proof of Patient Presence) -Modul

Version:	1.0.0- CC _CC4
Revision:	1494668 1713012
Stand:	26.01 03.09.2026
Status:	zur Abstimmung freigegeben
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemSpec_PoPP_Modul

Dokumenteninformationen

Gender-Hinweis

Aus Gründen der besseren Lesbarkeit wird in diesem Dokument überwiegend die männliche Form verwendet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

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
1.0.0 CC2	11.05.2026			gematik
1.0.0 CC3	05.08.2026		zur Abstimmung	gematik
1.0.0 CC4	03.09.2026		zur Abstimmung freigegeben	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes	8
1.1 Zielsetzung	8
1.2 Zielgruppe	8
1.3 Geltungsbereich	8
1.4 Abgrenzungen	9
1.5 Methodik	9
2 Systemüberblick/Systemkontext	10
2.1 Detailsicht "Online-Anwendungsfälle zur PoPP-Token-Erstellung"	10
3 Übersicht Funktionsmerkmale und Schnittstellen	22
3.1 Authentifizierung mit GesundheitsID	23
3.2 Authentifizierung einer eGK (in Fernversorgung)	24
3.3 Schnittstellen des PoPP-Moduls	25
3.3.1 initPoppTokenGeneration()	26
3.3.2 startPoppTokenErzeugung()	27
3.3.3 hasGesundheitsID()	28
3.3.4 init()	28
3.3.5 execute(httpLoadPractitionerInformationRequest)	28
3.3.6 callBackLoadPractitionerInformation	29
3.3.7 execute(httpReadEgkRequest)	30
3.3.8 callBackReadEgk — bei Authentifizierung einer eGK	31
3.3.9 sendAPDUCommands()	31
3.3.10 execute(httpCheckInGIDRequest)	32
3.3.11 callBackCheckInGID	33
3.3.12 execute(httpReadStatusPoPPRequest)	33
3.3.13 callBackReadStatusPoPP	35
3.3.14 callBackApp	35
3.4 PoPP-Modul in Drittanbieter APP	36
4 Übergreifende Festlegungen	38
4.1 Datenschutz und Informationssicherheit	39
5 Funktionsmerkmale	43
5.1 Allgemeine funktionale Anforderungen PoPP-Modul	43
5.2 Auswahl der LEI/ Bestimmen der Telematik ID	43
5.2.1 Favoriten	44
5.2.2 Verzeichnisdienstsuche	44
5.2.3 QR-Code	45
5.3 Auslösen der Erstellung eines PoPP-Token	46
5.3.1 Anforderungen PoPP-Modul bei Authentisierung eines Versicherten mit GesundheitsID	46

5.3.2 Anforderungen PoPP-Modul für eGK in Fernversorgung	47
5.3.3 Kommunikation PoPP-Modul mit ZETA-Client	49
5.3.4 Kommunikation PoPP-Modul mit Drittanbieter-Apps	54
5.4 Verarbeitung des Ergebnisses der PoPP-Token-Erzeugung	56
6 Informationsmodell	63
7 Implementierungsleitfaden	65
7.1 End-to-End-Flow (Happy Path)	65
7.2 Drittanbieter-Apps	67
7.2.1 Integriertes PoPP-Modul	68
7.2.2 Nachnutzung des PoPP-Moduls (App2App)	68
7.3 Alternative Flow-Schritte	69
7.3.1 Autorisierung	69
7.3.2 Auswahl der LEI	71
7.3.3 Statusinformation	76
7.3.4 Ersteinrichtung	76
7.4 Terminologie	79
8 Test	82
8.1 Schnittstelle für Testtreiber	82
9 Anhang A – Verzeichnisse	83
9.1 Abkürzungen	83
9.2 Glossar	84
9.3 Abbildungsverzeichnis	87
9.4 Tabellenverzeichnis	89
9.5 Referenzierte Dokumente	89
9.5.1 Dokumente der gematik	89
9.5.2 Weitere Dokumente	92
10 Anhang B – Ablaufbeschreibungen	94
10.1 Ablaufdiagramme	94
10.1.1 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung Versicherter über ihre GesundheitsID	94
10.1.2 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service	96
10.1.3 Detailablauf "ZETA-Client-Initialisierung"	97
10.1.4 Detailablauf "FHIR-VZD-Anfrage"	98
10.1.5 Detailablauf "Information zum Status der PoPP-Token-Generierung an das PoPP-Modul"	100
10.2 Flowdiagramme	101
10.2.1 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung Versicherter über ihre GesundheitsID	101
10.2.2 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung Versicherter mit ihrer GesundheitsID	107

10.2.3 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service	124
10.2.4 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung einer eGK	126
10.2.5 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service mit PoPP-Modul in einer Drittanbieter-App	176
10.2.6 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung einer eGK mit PoPP-Modul in einer Drittanbieter-App	178
11 Anhang C – Offene Punkte, Fragen	212
11.1 Offene Punkte	212
1 Einordnung des Dokumentes	8
1.1 Zielsetzung	8
1.2 Zielgruppe	8
1.3 Geltungsbereich	8
1.4 Abgrenzungen	9
1.5 Methodik	9
2 Systemüberblick/Systemkontext	10
2.1 Detailsicht "Online-Anwendungsfälle zur PoPP-Token-Erstellung"	10
3 Übersicht Funktionsmerkmale und Schnittstellen	22
3.1 Authentifizierung mit GesundheitsID	23
3.2 Schnittstellen des PoPP-Moduls	25
3.2.1 App interne Schnittstellen des PoPP-Modul	25
3.2.1.1 Initialisierung des ZETA Client	26
3.2.1.2 Ermitteln der Daten zu einer Telematik-ID (FHIR-VZD-Abruf)	29
3.2.1.3 Authentifizierung mit GesundheitsID	30
3.2.1.4 Abfrage Status PoPP-Token-Generierung	34
3.2.2 Schnittstellen des PoPP-Modul zum Aufruf aus anderen Anwendungen	35
3.2.2.1 HTTP-Request - start PoPPTokenGeneration	35
3.2.2.2 HTTP-Request - status PoPPTokenGeneration	35
4 Übergreifende Festlegungen	38
4.1 Datenschutz und Informationssicherheit	39
4.2 Barrierefreiheit	41
5 Funktionsmerkmale	43
5.1 Allgemeine funktionale Anforderungen PoPP-Modul	43
5.2 Anforderungen an die Kommunikation des PoPP-Modul mit dem ZETA Client	45
5.3 Bestimmen der Telematik-ID	50
5.3.1 QR-Code	50
5.3.2 Verzeichnisdienstsuche	51

5.4 Anforderungen PoPP-Modul bei Authentisierung einer VER mit GesundheitsID.....	52
5.5 Verarbeitung des Ergebnisses der PoPP-Token-Erzeugung	56
5.6 Kommunikation PoPP-Modul mit Anbieter-Apps	56
5.6.1 App2App-Flow	57
5.6.2 Web2App-Flow.....	58
5.6.3 2-Geräte-Flow	59
5.6.4 Anforderungen an die Kommunikation zwischen PoPP-Modul und Anbieter-Apps.....	60
6 Informationsmodell	63
7 Implementierungsleitfaden	65
7.1 End-to-End Flow (Referenzpfad)	65
7.2 Anbieter-Apps	67
7.3 Alternative Flow-Schritte.....	69
7.3.1 Erfassung der Telematik-ID	69
7.3.2 Statusinformation	76
7.3.3 Ersteinrichtung	76
7.4 Terminologie	79
8 Anhang A – Verzeichnisse	83
8.1 Abkürzungen	83
8.2 Glossar	84
8.3 Abbildungsverzeichnis.....	87
8.4 Tabellenverzeichnis	89
8.5 Referenzierte Dokumente	89
8.5.1 Dokumente der gematik.....	89
8.5.2 Weitere Dokumente.....	92
9 Anhang - Ablaufbeschreibungen	94
9.1 Gesamtüberblick Auslösung und Ablauf der PoPP-Token-Generierung	94
9.2 Detailablauf "ZETA Client Initialisierung"	97
9.3 Detailablauf "FHIR-VZD-Anfrage".....	98
9.4 Detailablauf "Authentifizierung mit GesundheitsID"	100
9.4.1 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung VER über ihre GesundheitsID.....	101
9.4.2 Detaillierter Ablauf des Online Check-in durch Authentifizierung VER über ihre GesundheitsID	103
9.4.3 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung VER mit ihrer GesundheitsID	107
9.5 Ablauf "PoPP-Token-Abruf aus einem Primärsystem"	175
9.5.1 Übersicht zum Ablauf des PoPP-Token-Abruf aus einem Primärsystem	175
9.5.2 Detaillierter Ablauf zum PoPP-Token-Abruf aus einem Primärsystem.....	176
9.5.3 Beschreibung der Schritte zum PoPP-Token-Abruf aus einem Primärsystem.....	178

9.6 Ablauf "Abfrage des Status zur PoPP-Token-Generierung"	196
9.6.1 Übersicht zum Ablauf der Abfrage des Status zur PoPP-Token-Generierung ..	196
9.6.2 Detaillierter Ablauf der Abfrage des Status zur PoPP-Token-Generierung	197
9.6.3 Beschreibung der Schritte des Ablaufs der Abfrage des Status zur PoPP-Token-Generierung	199
10 Anhang C – Offene Punkte, Fragen	212
10.1 Offene Punkte.....	212

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 **Personen (VER)** 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 Versicherte~~ eine bestimmte VER mit dem Inhaber einer bestimmten ~~Leistungserbringereinstitution (LEI)~~ Telematik-ID in einen Versorgungskontext getreten ist.

Neben dem PoPP-Modul, tragen weitere Komponenten zur PoPP-Lösung bei, ~~die in den referenzierten Dokumenten beschrieben werden.~~

- 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 **Kassen-Apps für Versicherte**, die ein PoPP-Modul ~~in ihre App~~ integrieren wollen, insbesondere an die Kassen und ihre Auftragnehmer, **sowie Drittanbieter an Hersteller und Betreiber von Anbieter-Apps. Anbieter-Apps nutzen das PoPP-Modul einer Kassen-App über App2App, integrieren aber selbst kein PoPP-Modul bei der Verwendung der GesundheitsID.**

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 98]).

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~~erfüllen funktionale und sicherheitstechnische Anforderungen der gematik-~~erfüllen. Welche Anforderungen ein PoPP-Modul erfüllen muss, wird von der, die~~ gematik in einem Produkttypsteckbrief bekanntgegeben-
werden. Die Erfüllung der Anforderungen ~~muss~~weist der Hersteller eines PoPP-Moduls der gematik in einem Zulassungsverfahren ~~nachweisender gematik nach~~.

~~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~~ Eine VER initiiert die Erstellung eines PoPP-Token an einer App (Kassen-App oder Anbieter-App). An der Erstellung des PoPP-Token sind mehrere Architekturkomponenten der TI beteiligt.

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

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~~den Inhaber der Telematik-ID, welchen die VER ausgewählt hat und für die er in eine Datenweitergabe eingewilligt hat.

Die folgende Abbildung zeigt die beteiligten Komponenten für den Online Check-in aus einer Anbieter-App mit Authentifizierung der VER über die GesundheitsID.

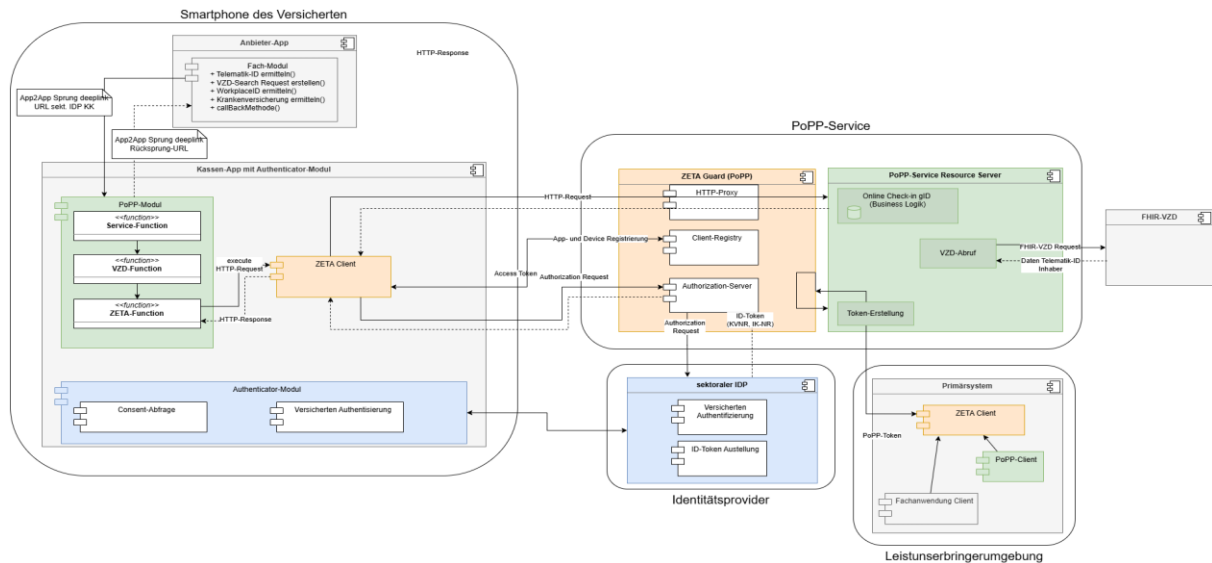
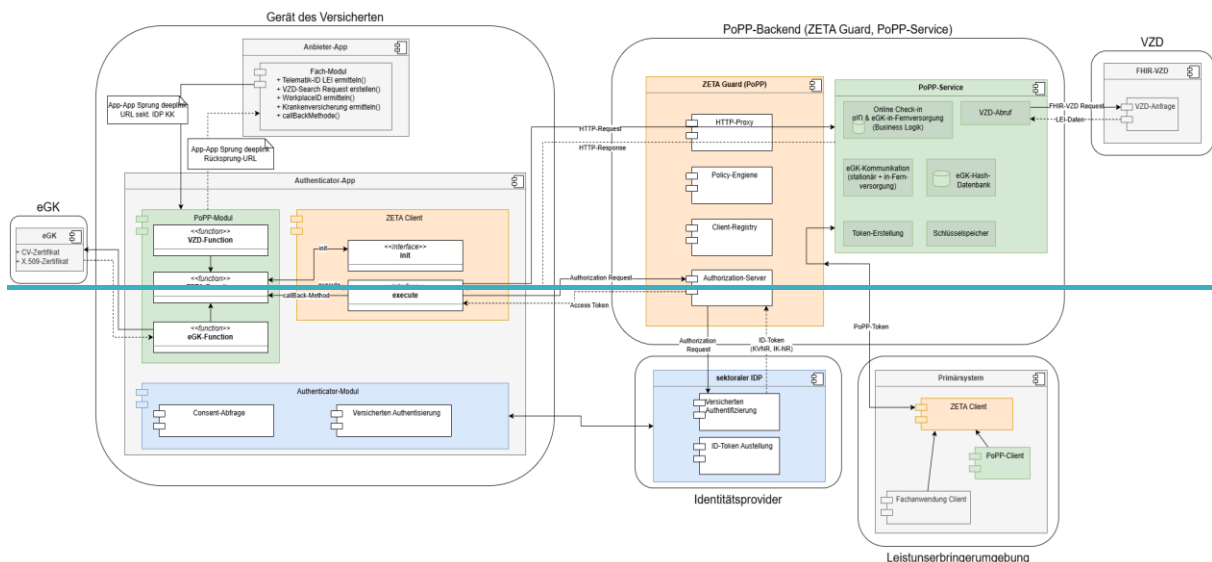


Abbildung 1: Verteilungssicht der beteiligten Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in App der Krankenversicherung und nutzender Anbieter-App auf dem Smartphone der VER

PoPP-Modul und ZETA Client müssen **dabei immer** gemeinsam in **eine einer Kassen-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 Das PoPP-Modul und ZETA Client in die Krankenversicherungs-App integriert sind, **welchewird** dabei in dem Anwendungsteil der Kassen-App bereitgestellt, **der** auch das Authenticator-Modul des sektoralen IDP der Krankenversicherung implementiert.



Die Abbildung 1.1, "Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in App der Krankenversicherung" zeigt die beteiligten Komponenten für diese Referenzarchitektur. Das PoPP-Modul, der ZETA Client und das Authenticator-Modul bilden dabei gemeinsam die zentrale Komponente für die Durchführung des Online-Check-ins auf dem Smartphone der VER.

Diese Konstellation unterstützt alle Online-Anwendungsfälle zur PoPP-Token-Erstellung ~~der. Bei Krankenversicherungen optimal~~, die eine Mehr-App-Strategie verfolgen, kann der Anwender abhängig vom jeweiligen Anwendungsfall in die Kassen-App mit PoPP-Modul weitergeleitet werden (App2App-Sprung).

Auch Anwendungsfälle von ~~Drittanbieter~~Anbieter-Apps werden ~~ebenfalls~~über einen App2App-Sprung in die Kassen-App mit PoPP-Modul unterstützt. ~~Im Ablauf findet~~Nach Abschluss des Online Check-ins erfolgt, sofern erforderlich, ein ~~App-Wechsel~~Rücksprung in die aufrufende Anwendung.

Anwendungen, aus denen ein Online Check-in ausgelöst wird, werden im Folgenden allgemein als „Anbieter-App“ bezeichnet. Dies umfasst sowohl Kassen-Apps ohne eigenes PoPP-Modul als auch Anwendungen von Drittanbietern. Anbieter-Apps initiieren den Online Check-in und verarbeiten dessen Ergebnis, führen jedoch bei der ~~Drittanbieter-App in die Krankenversicherungs-App und zurück statt. Es erfordert~~Verwendung der GesundheitsID selbst keine PoPP-spezifische Authentifizierung oder PoPP-Token-Erstellung durch.

Die Logik zur Authentifizierung der VER sowie zur Erstellung des PoPP-Token beschränkt sich vollständig auf die Kassen-App mit integriertem PoPP-Modul, ZETA Client und Authenticator-Modul.

Aus diesem Grund besteht keine Notwendigkeit, Anbieter-Apps hinsichtlich ihrer Implementierung des Online Check-ins durch die gematik ~~zuzulassende~~ Implementierung zuzulassen.

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 Smartphone der VER

Komponente	Beschreibung
Gerät des Versicherten Smartphone der VER	Auf dem Gerät des Versicherten (i.d.R. der VER (in der Regel ein Smartphone) sind Anwendungen (Apps) installiert, die für ihren fachlichen Ablauf die Erstellung eines PoPP-Token für eine LE den Inhaber einer Telematik-ID benötigen. Auf dem Gerät des Versicherten ist eine App installiert, welche PoPP-Modul, ZETA Client und Authenticator-Modul integriert (Krankenversicherungskassen-App, Authenticator-App der Krankenversicherung). Das Gerät verfügt über eine Menge von Eigenschaften, welche bei der sicheren Kommunikation mit den Backend-Systemen berücksichtigt werden müssen. 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. Bei den Eigenschaften handelt es sich u.a. um Gerätetyp, Gerätehersteller, Version des installierten Betriebssystems, Informationen zu Hardwareausstattung (z.B. Schlüsselspeicher). Die Anforderungen an die Geräte- und App-Attestierung sind in [gemSpec_ZETA] beschrieben.

Komponente	Beschreibung
Anwendung/App auf dem Gerät des Versicherten Smartphone der VER	Die auf dem Gerät des Versicherten Smartphone der VER installierten Anwendungen, welche die Erstellung eines PoPP-Token anfordern, können vielfältig ausgeprägt sein: 1. Personalisierte Anwendungen (KrankenversicherungKassen-Apps) für Versicherte VER zur Vermittlung einer Kommunikation zwischen Versicherten und LEIdem Inhabers einer Telematik-ID 2. Anwendung einer LEIeines Anbieters für Versicherte VER (z.B. Online-Apotheke) 3. Anwendung zur Vermittlung der Kommunikation zwischen Versicherten VER und LEIInhabern einer Telematik-ID (z.B. Telemedizin-Anwendungen) 4. Anwendungen für eine bestimmte Gruppe von Versicherten VER zur Kommunikation mit einer bestimmten Gruppe von LEITelematik-ID Inhabern (z.B. Patientenportale für Klinken) Bedingt durch die Vielfalt der Ausprägungen sind die Voraussetzungen für die Erstellung von PoPP-Token unterschiedlich: 1. Ist die Telematik-ID der LEIder Anwendung bereits bekannt oder muss sie erst ermittelt werden (Online-Apotheke vs. Versichertenportal)? 2. Kann die Telematik-ID ermittelt werden oder muss eine VZD-Suche der LEI mit Suchkriterien zum Inhaber einer Telematik-ID durchgeführt werden? 3. Ist die Krankenversicherung des Versicherten der VER bekannt oder muss diese ermittelt werden (KrankenversicherungKassen-App vs. Online-Apotheke)? Fehlende Informationen müssen durch die Anwendung selbst beschafft werden, bevor eine PoPP-Token-Erstellung durchgeführt werden kann.

Komponente	Beschreibung
AuthenticatorKassen-App /mit Authenticator-Modul auf dem Gerät des VersichertenSmartphone der VER	Werden VersicherteVER 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 mit einem der möglichen Authentisierungsmittel durchführen. Die AuthenticatorKassen-App mit Authenticator-Modul ist die App auf dem Gerät des VersichertenSmartphone der VER, welches das Authenticator-Modul des sektoralen IDP der Krankenversicherung des Versichertender VER implementiert. Es gibt zwei Ausprägungen der Umsetzung: 1. Das Authenticator-Modul ist in die Krankenversicherung-App/ePA-FdVeiner Kassen-App integriert, die weitere Funktionen (z.B. Frontends für ePA und eRP, weitere Funktionen der Krankenkasse) enthält. 2. Das Authenticator-Modul ist als eigene App der Krankenversicherung implementiert. 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.

PoPP-Modul	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 Smartphone der VER. • VZD-Function Funktion Für die Nutzerzustimmung zur Verwendung der Versichertendaten durch eine LE den Inhaber einer Telematik-ID wird vom PoPP-Modul über die ZETA-Komponenten eine Anfrage an den PoPP-Service Resource Server gestellt. Das PoPP-Modul formuliert dafür einen Anfrage-Request für das FHIR-VZD mit den zur Verfügung stehenden Parametern wie Name, PLZ des Inhabers der LE Telematik-ID oder übernimmt einen fertigen Anfrage-Request aus dem Aufruf der einer anderen App (bei Mehr-App-Strategie) oder Anbieter-App. Der PoPP-Service Resource Server 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 des Inhabers der LE Telematik-ID für die Nutzereinwilligung dar. • ZETA-Function Funktion Die Kommunikation zwischen PoPP-Modul und PoPP-Service Resource Server für die VZD VZD-Abfrage erfolgt über die ZETA-Komponenten. Bei einer Authentifizierung des Versicherten der VER über dessen GesundheitsID kommuniziert das PoPP-Modul ebenfalls nicht direkt mit dem PoPP-Service Resource Server. 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). Im Fall der Authentifizierung der eGK ("eGK in Fernversorgung") kommuniziert der PoPP-Service ebenfalls über die ZETA-Komponenten mit dem PoPP-Modul. • eGK-Function Funktion • Für das Auslesen der eGK ("eGK in Fernversorgung") agiert das PoPP-Modul als Proxy. Vom PoPP-Service-Function Das PoPP-Modul unterstützt die Anwender durch die Bereitstellung von Service
------------	--

Komponente	Beschreibung
	empfangende APDU Commands werden über die NFC-Schnittstelle des Gerätes direkt an Funktionen. Diese umfassen die eGK-gesendet. Die Antworten der eGK werden direkt an dem PoPP-Service gesendet. Des Weiteren speichert das PoPP-Modul Informationen zur Historie der PoPP-Anfragen und eine Favoritenliste von Leistungserbringer Institutionen.
ZETA Client	ZETA Client ist die Zero Trust-Komponente für ein FdVFrontend, 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. Im laufenden Prozess kapselt der ZETA Client die gesamte Kommunikation zwischen der Anwendung auf dem Gerät des VersichertenSmartphone der VER und den Komponenten im Backend, u.a. auch zwischen PoPP-Modul und PoPP-Service Resource Server. Der ZETA Client steuert die Client-Authentisierung am ZETA Guard Authorization-Server und unterstützt die Nutzerauthentifizierung mit GesundheitsID. ZETA Client ist in [gemSpec_ZETA] spezifiziert. Die gematik stellt ein ZETA SDK bereit, welche alle Anforderungen an ein ZETA Client Modul erfüllt und sowohl in Apps für Android als auch iOS Betriebssysteme integriert werden kann ([API ZETA Client]).
PoPP-Service ZETA Guard	
ZETA Guard-Client-Registry Registrierung der Kassen-App mit PoPP-Modul	Alle Clients bzw. AnwendungenDie Kassen-Apps auf dem Gerät des VersichertenSmartphone der VER, die ein ZETA Client implementieren, müssen an der am PoPP-Service ZETA Guard Client-Registry registriert werden. Der Prozess der Registrierung ist in [gemSpec_ZETA] beschrieben. Im laufenden Prozess wird überprüft dann der PoPP-Service ZETA Guard Client-Registry geprüft, ob ein Aufruf von einem bekannten bzw. registrierten ZETA Client kommt und lehnt bei Auffälligkeiten Aufrufe an den PoPP-Service Resource Server ab.

Komponente	Beschreibung
ZETA Guard Authorization-Server Ausstellen von Access-Token für Request des PoPP-Moduls an den PoPP-Service Resource Server	Bevor eine Anwendung über ein registriertes ZETA Client auf einen Fachdienst zugreifend das PoPP-Modul Requests an den PoPP-Service Resource Server senden kann, führt ZETA Guard Authorization-Server die Client-Authentifizierung durch und stellt dem ZETA Client in der Anwendung Kassen-App bei erfolgreicher Authentifizierung ein Access-Token aus. Wird von der Anwendung Kassen-App darüber hinaus die Authentifizierung des Nutzers der VER über die GesundheitsID angefragt, so stößt der ZETA Guard Authorization-Server den Authentifizierungsprozess beim jeweiligen Sektorens sektoralen IDP an. Der ZETA Guard Authorization-Server nimmt bei erfolgreicher Authentifizierung das vom sektoralen IDP ausgestellte ID-Token entgegen und verarbeitet dessen Inhalt für die spätere Verwendung beim Aufruf des PoPP-Service Resource Server. Dem ZETA-Client in der Kassen-App wird ein Access-Token ausgestellt.
ZETA Guard HTTP-Proxy Ausführung fachlicher Request vom PoPP-Modul an den PoPP-Service Resource Server	Fachliche Requests aus den Anwendungen werden vom ZETA Client der Anwendung an den ZETA Guard HTTP-Proxy geschickt, nach Prüfung des Clients gegen die . ZETA Guard Client-Registry und ggf. Durchführung von Prüfrege ln durch die ZETA Guard Policy Engine reichert der ZETA Guard HTTP-Proxy den Request mit Header -Informationen (z.B. über den aufrufenden Client oder aus dem ID-Token des sekt. IDP) an und sendet den Request dann an den eigentlichen Empfänger, den PoPP-Service Resource Server . Das Ergebnis des Requests wird dem ZETA Client der aufrufenden Anwendung Kassen-App zugestellt.
PoPP-Service	

~~eGK-Kommunikation (Stationär + in-Fernversorgung) Push-Notifications vom an den Resource Server an das PoPP-Modul~~

~~Versicherte haben die Möglichkeit eine eGK ("eGK in Fernversorgung") für die Erstellung eines PoPP-Token zu verwenden.~~
~~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.~~
~~Die Komponente "eGK-Kommunikation" nimmt die Antworten entgegen und wertet diese aus.~~
~~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.~~
~~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. Der Resource Server schickt Push-Notification über den ZETA Guard an die Kassen-App mit PoPP-Modul auf dem Gerät der VER.~~
~~Die Push-Nachricht wird dem ZETA-Client auf dem Gerät der VER zugestellt, das PoPP-Modul zeigt der VER die Nachricht an.~~

eGK-Hash-Datenbank	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.
VZD-Abfrage	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. Der PoPP-Service propagiert das Ergebnis der Suchanfrage unverändert über die ZETA-Komponenten an das PoPP-Modul.
Token-Erstellung	Die Komponente Token-Erstellung enthält die Funktionalität zum Erstellen des PoPP-Token. Beim 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	

Gelöscht

sektoraler IDP	Die Authentifizierung VersicherterVER mit GesundheitsID erfolgt durch den sektoralen IDP der Krankenversicherung des Versichertender VER. Der sektorale IDP hält einen Datensatz mit Daten zum Versichertenzur VER selbst (GesundheitsID) sowie Informationen zu den möglichen Authentisierungsmitteln und den Nutzerpräferenzen. Die Authentifizierung VersicherterVER erfolgt dann in Verbindung mit dem Authenticator-Modul auf dem Gerät des Versicherten (siehe auch Flow-Beschreibungen in Smartphone der [Wissensdatenbank] zurVER. Die Funktionsweise der TI-Föderation) ist in gemKPT_TI-Föderation beschrieben. Anforderungen an den sektoralen IDP und dessen Authenticator Modul sind in [gemSpec_IDP_Sek] festgelegt.
VZD	
FHIR-VZD	Der VZD liefert das Ergebnis einer Suchanfrage vom PoPP-Service- Resource Server. Der PoPP-Service Resource Server übergibt dem VZD-Suchparameter und liefert Datensätze zu gefundenen LEIsTelematik-IDs.

3 Übersicht Funktionsmerkmale und Schnittstellen

Apps mit unterschiedlichen fachlichen Ausrichtungen haben den Bedarf, einen Versorgungskontext zwischen einer versicherten Person und dem Inhaber einer Telematik-ID (Leistungserbringerinstitution, DiGA, Kostenträger) herzustellen.

Die Benutzerführung für die Erstellung eines Versorgungskontext zwischen einer versicherten Person und dem Inhaber einer Telematik-ID 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")?
- 1. Ist durch die Anbieter-App bereits festgelegt, welche Leistungserbringerinstitution (LEI) welcher Telematik-ID ein PoPP-Token benötigt oder muss die Telematik-ID über das PoPP-Modul ermittelt werden?
- 2. Welche Verfahren zur Ermittlung der Telematik-ID werden unterstützt?

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

Weitere Einflussfaktoren zur Ausgestaltung der Benutzerführung sind:

- 1. Wenn Ist in der Kassen-App auch das PoPP-Modul in ein ePA-Frontend des Versichertender VER (FdV) integriert ist, so kann für die Authentifizierung des Versichertender VER über seine GesundheitsID das Single-Sign-On (SSO) des ePA-FdV genutzt werden. (Das SSO auf Anwendungsebene ist derzeit nur für zugelassene ePA-FdV erlaubt - siehe [gemSpec_IDP_Sek]).
- 2. Wenn eine Anbieter-App, welche das PoPP-Modul benötigt, einen Online Check-in initiiert und diese auf demselben Gerät installiert ist wie die App mit dem Authenticator-Modul für die Authentifizierung mit GesundheitsID (entweder standalone oder integriert in Krankenkassen-App), erfolgt die Erstellung des PoPP-Token bei einer Authentifizierung mit GesundheitsID über das PoPP-Modul mit App-App2App-Kommunikation (App-App2App Sprung).
- 3. Wenn eine Anbieter-App, welche das PoPP-Modul benötigt, einen Online Check-in initiiert, und diese 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 elektronischen 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 Resource Server 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 **Resource Server** hergestellt.

Im Anhang des Dokuments sind die Abläufe der PoPP-Token-Erstellung über eine Authentifizierung von **Versicherten-VER** 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:

1. Auswahl der Krankenversicherung ~~des Versichertender~~ **VER** bzw. Ermittlung der ClientID des sektoralen IDP der Krankenversicherung in der TI-Föderation.
2. Ermittlung der Telematik-ID ~~der LEI~~, für **diesen** ein PoPP-Token erstellt werden soll.
3. **Wenn für den Anwendungsfall notwendig**, Ermittlung der WorkplaceID, für die ein PoPP-Token erstellt werden soll.
~~• Festlegung, ob zwingend eine Anmeldung mit "eGK in Fernversorgung" durchgeführt werden soll.~~
4. Aufruf eines **HTTP-Request** mit **Telematik-ID**den notwendigen Parametern und einer Rücksprung-URL (callbackURL) an ~~die einen festgelegten Pfad zur URL (ClientID)~~ des sektoralen IDP der Krankenversicherung in der TI-Föderation um den Ablauf in ~~das~~ **die App mit PoPP-Modul** zu verlagern.

Die WorkplaceID **ist eine optionale Information**. Sie dient dem Inhaber einer **LEI**, ~~welche~~ **Telematik-ID**, welcher ein PoPP-Token erhält, dieses einem konkreten Arbeitsplatz zuzuweisen. Die WorkplaceID kann durch ~~die LEI~~ **das Primärsystem des Inhabers der Telematik-ID** auch wie eine Session-ID verwendet werden, um das letztendlich erzeugte PoPP-Token dann ~~dem~~ **der** richtigen **Versicherten-VER** 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 geltend** darf durch das PoPP-Modul nicht gelöscht oder verändert werden.

~~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 **Kassen-App** integriertes PoPP-Modul gestartet. Für die Authentifizierung mit GesundheitsID gibt es ~~hier drei unterschiedliche~~ **diese unterschiedlichen** Umsetzungsvarianten:

1. PoPP-Modul und Authenticator-Modul des sektoralen IDP sind in ~~die~~ **eine** App der Krankenversicherung integriert, ~~welche auch das ePA-FdV integriert.~~ **die weitere digitalen Serviceleistungen umfasst.**

2. PoPP-Modul und Authenticator-Modul des sektoralen IDP sind in einer eigenen ~~AuthenticatorKassen~~-App implementiert und der Online -Check-in mit GesundheitsID wird aus einer ~~anderen-Anbieter~~-App gestartet (~~App-App2App~~ Sprung).

- ~~1. 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 allgemeinen~~ Ablauf der PoPP-Token-Generierung dar. Dabei wird davon ausgegangen, dass PoPP-Token-Generierung für den Fall dar, dass das PoPP-Modul, ZETA Client und ~~Autheticator~~Authenticator-Modul gemeinsam in einer Kassen-App integriert sind.

Der Start des Online-Check-in erfolgt in einer ~~DrittanbieterAnbieter~~-App oder in der Kassen-App ~~der Krankenversicherung. Ein Versicherter. Eine VER~~ wird über seine GesundheitsID authentifiziert. Die für die PoPP-Token-Erstellung notwendigen Daten, KVNR ~~des Versichertender VER~~ und IK-Nummer der Krankenversicherung ~~des Versichertender VER~~, 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.~~

3.3.2 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 **AuthenticatorKassen-App** sind API-Schnittstellen. Zwischen den über das Internet erreichbaren Komponenten sind REST-Schnittstellen.

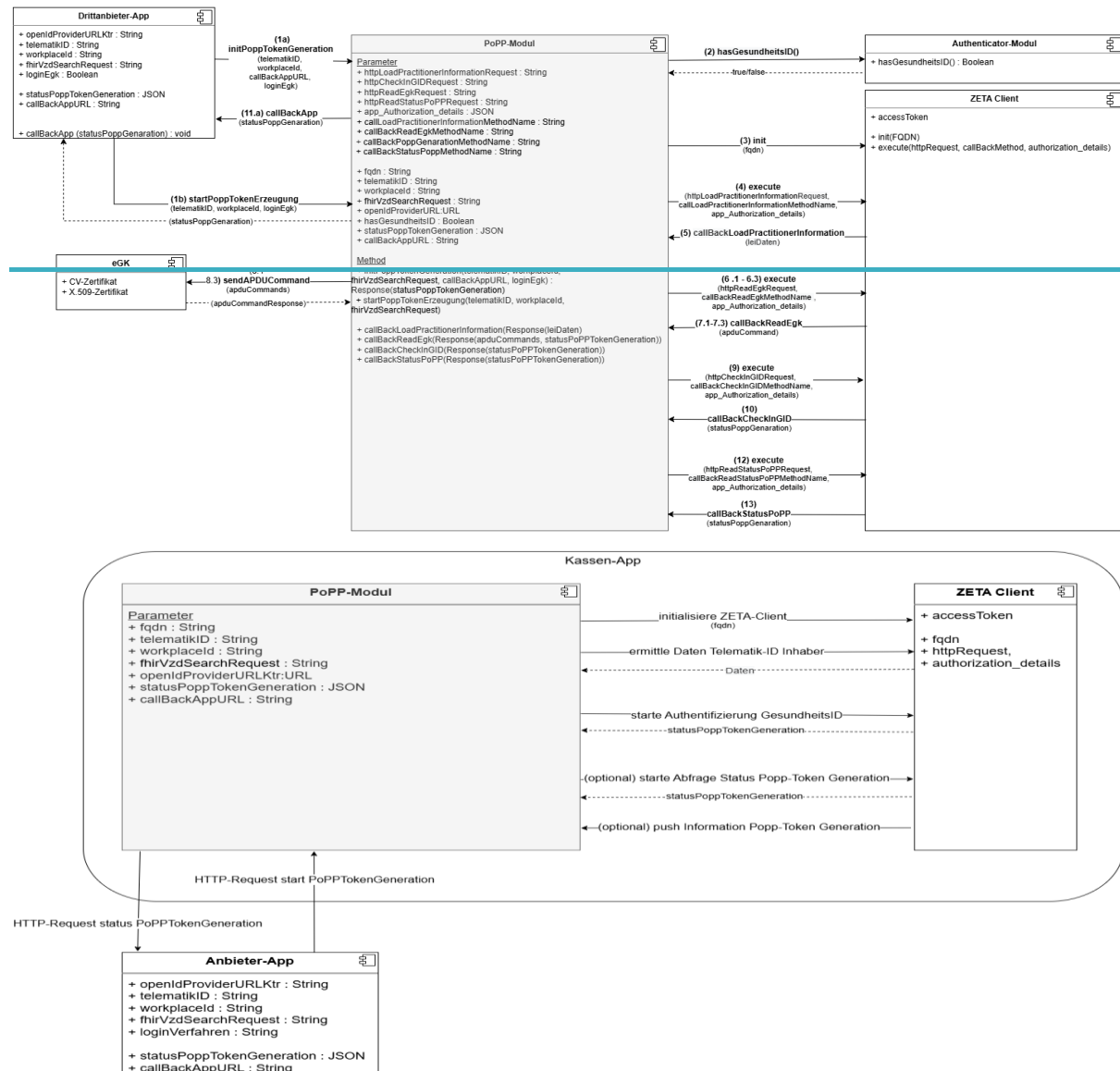


Abbildung 2: Schnittstellen vom und zum PoPP-Modul

3.2.1 App interne Schnittstellen des PoPP-Modul

Das PoPP-Modul tauscht einerseits Nachrichten in Richtung der VER aus. Andererseits tauscht es Nachrichten in Richtung PoPP-Service aus, die stets über einen ZETA Client laufen. Das Protokoll zwischen PoPP-Modul und ZETA Client ist implementierungsspezifisch. Wird das von der gematik bereitgestellte ZETA-SDK verwendet, so ist dessen API [ZETA-Client-SDK] zu verwenden.

3.3.1-Initialisierung des ZETA ClientinitPoppTokenGeneration()

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[<u>gemSpec-PKI</u>] – <u>"Aufbau der Telematik-ID"</u>
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 übergeben werden.	String	Der Search-Request muss ein am FHIR-VZD ausführbarer Request sein.
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 startPoppTokenErzeugung()

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"
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 übergeben werden.	String	Der Search-Request muss ein am FHIR-VZD ausführbarer Request sein.
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:

3.2.1.1 ~~Der execute-~~

Nach jedem Starten der Kassen-App wird deren ZETA Client initialisiert. Im Rahmen der Initialisierung werden bei Bedarf unter anderem Geräte- und App-Attestierung durchgeführt. Nach der ZETA-Initialisierung stellen die ZETA-Komponenten des PoPP-Service ein Client Access-Token aus, welchen im ZETA Client zur laufenden Session hinterlegt wird. Dieses Access-Token wird im weiteren Ablauf bei Request gegen den PoPP-Service Resource Server verwendet.

3.2.1.2 Ermitteln der Daten zu einer Telematik-ID (FHIR-VZD-Abruf)

Der Aufruf des PoPP-Modul am ZETA Client ist die Anweisung zur Durchführung eines HTTP-Request am PoPP-Service **Resource Server** für die Suche der Daten zu **einem Inhaber einer LEI/Telematik-ID**. Dafür wird ~~als Parameter~~ ein am FHIR-VZD ausführbarer Search-Request **als Parameter** mit übergeben.

~~Wurde~~ der Search-Request ~~wurde entweder~~ an der Schnittstelle zum PoPP-Modul (~~initPoPPTokenGeneration oder startPoPPTokenErzeugung~~) aus einer anderen App übergeben. ~~Nach der ZETA-Initialisierung stellen, so muss dieser verwendet werden.~~ Ansonsten erstellt das PoPP-Modul einen Search-Request mit der Telematik-ID, für die ~~ZETA-Komponenten ein PoPP-Token erstellt werden soll.~~

Der ZETA-Client ~~Access-Token aus und führt~~ den HTTP-Request ~~mit dem~~ unter Verwendung des Access-Token am PoPP-Service **Resource Server** aus. Nach Ermittlung der ~~LEI~~ Daten **zum Inhaber einer Telematik-ID** 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:

[I_PoPP_Service_mobile_CheckIn.yaml] enthält die API-Definition für die notwendige Implementierung im PoPP-Service Resource Server.

3.2.1.3 Authentifizierung mit GesundheitsID

Der Aufruf der ~~execute-Methode~~ mit dem Parameter ~~httpReadEgkRequest~~ `httpCheckInGIDRequest` wird nur bei der Authentifizierung einer eGK mit GesundheitsID 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	httpReadEgkRequest
callbackMethod	String	callbackReadEgk

Parameter	Typ	Wertebereich
app_authorization_details	JSON	<pre>"app_authorization_details": { "type": "urn:ti:gematik:apdu:command", "apdu_command_response": "<APDU-Command-Response>" }</pre> (Die app_authorization_details enthalten Informationen für den PoPP-Service)

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-Resource-Server zur Erstellung eines PoPP-Token nach Authentifizierung des Versichertender VER über seine GesundheitsID. Die ZETA-Komponenten prüfen, ob bereits eine Authentifizierung durchgeführt wurde. Ist das nicht der Fall, wird die Authentifizierung des Versichertender VER über den sektoralen IDP der Krankenversicherung der VER angestoßen. Die URL des sektoralen IDP wird in den `app_authorization_details` vom PoPP-Modul übergeben an der Schnittstelle ZETA als `authorization_details` gesetzt. Nach erfolgreicher Authentifizierung werden den ZETA-Komponenten die Daten des Versichertender VER als ID-Token übergeben. Die ZETA-Komponenten führen nun den `httpCheckInGIDRequest` am PoPP-Service-Resource-Server aus. Der PoPP-Service-Resource-Server erstellt einen Datensatz zur Generierung eines PoPP-Token. Den Status der PoPP-Token-Generierung liefert der PoPP-Service als Response den ZETA-Komponenten zurück. Den Status der PoPP-Token-Generierung liefert der PoPP-Service-Resource-Server 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>

Parameter	Typ	Wertebereich
app_authorization_details	JSON	<pre>"app_authorization_details": { "type": "urn:ti:gematik:auth:provider", "auth_preferences": { "openIdProviderUrl": "<URL zum Service, welche die Authentifizierung durchföhrt>" } }</pre> (Die app_authorization_details 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
statusPoppGenarationResponse	HTTP-Response	HTTP-OK(statusPoppGenaration), HTTP-<ErrorCode>

~~Rückgabewerte:~~ keiner

~~3.3.12 execute(httpReadStatusPoPPRequest)~~

~~Beschreibung:~~

[I_PoPP_Service_mobile_CheckIn.yaml] enthält die API-Definition für die notwendige Implementierung im PoPP-Service Resource Server.

Offener Punkt: OP-PoPP-2

Für die Authentisierung eines Versicherten mit GesundheitsID müssen Informationen zum verwendeten sektoralen Identity Provider der Krankenversicherung an die ZETA-Komponenten übergeben werden. Die Ausgestaltung dieser Übergabe und die hierfür verwendeten Schnittstellenparameter sind derzeit noch nicht abschließend festgelegt. (Stichwort: `idp_iss` via `authorization_details`.)

Vorgehen:

Klärung innerhalb der gematik gemeinsam mit den verantwortlichen ZETA-Teams parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die Spezifikationen eingearbeitet.

3.2.1.4 Abfrage Status PoPP-Token-Generierung

Der Aufruf ~~der execute-Methode~~ mit dem Parameter `httpReadStatusPoPPRequest` kann optional vom PoPP-Modul ~~aufgerufen~~ausgeführt werden, um ~~beim~~ PoPP-Service Resource Server den aktuelle Status zum Check-in abzurufen und der VER darzustellen. Dieser Aufruf ist dann sinnvoll, wenn der ursprüngliche Check-in noch nicht vollständig abgeschlossen werden konnte, da ~~die LE~~das Primärsystem den PoPP-Token noch nicht ~~online war~~abgerufen hat. 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 Resource Server aus. Den Status der PoPP-Token-Generierung liefert der PoPP-Service als Response den ZETA-Komponenten zurück. aus. Den Status der PoPP-Token-Generierung liefert der PoPP-Service als Response den Der ZETA-Komponenten zurück. Der ZETA-Client ruft die callback-Methode mit dem Client gibt das Response-Objekt beim an das PoPP-Modul aufweiter.~~

~~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
apduCommand statusPoppTokenGenerierung	HTTP- Response	HTTP-OK(apduCommand, statusPoppTokenGenerierung), HTTP- <ErrorCode>

~~Rückgabewerte: keiner~~

3.3.14 callBackApp

Beschreibung:

[I_PoPP_Service_mobile_CheckIn.yaml] enthält die API-Definition für die notwendige Implementierung im PoPP-Service Resource Server.

3.2.2 Schnittstellen des PoPP-Modul zum Aufruf aus anderen Anwendungen

3.2.2.1 HTTP-Request - start PoPPTokenGeneration

Wird die PoPP-Token-Erstellung über eine Anbieter-App initiiert, so ruft diese eine festgelegte URL des sektoralen IDP der Krankenversicherung der VER 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 Kassen-App mit implementiertem PoPP-Modul. Die im Request übergebenen Parameter werden an das PoPP-Modul übergeben.

[I_PoPP_Modul_mobile_CheckIn.yaml] enthält die API-Definition für die notwendige Implementierung im PoPP-Modul.

3.2.2.2 HTTP-Request - status PoPPTokenGenerierung

Ist die PoPP-Token-Erstellung über eine ~~Drittanbieter-App über App-AppApp2App-~~Kommunikation initiiert, muss der Rücksprung in die Anbieter-App erfolgen. Der App-Wechsel kann über Plattformmechanismen realisiert werden. Dazu ruft das PoPP-Modul die im ~~initPoppTokenGeneration~~startPoppTokenGeneration()-Request übergebene callback-URL auf. In der Response wird der Anbieter-App der Status der PoPP-Token Generierung ("success", "pending" oder "canceled") sowie die ID des Nachrichten-Datensatz übermittelt.

Ist die 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 (als deeplink /bzw. universal link)- eingerichtet, so führt der Aufruf zum Öffnen der Anbieter-App.

Protokoll: HTTP

Parameter:

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

Rückgabewerte:

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

3.4 [I_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 authetifiziert 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").

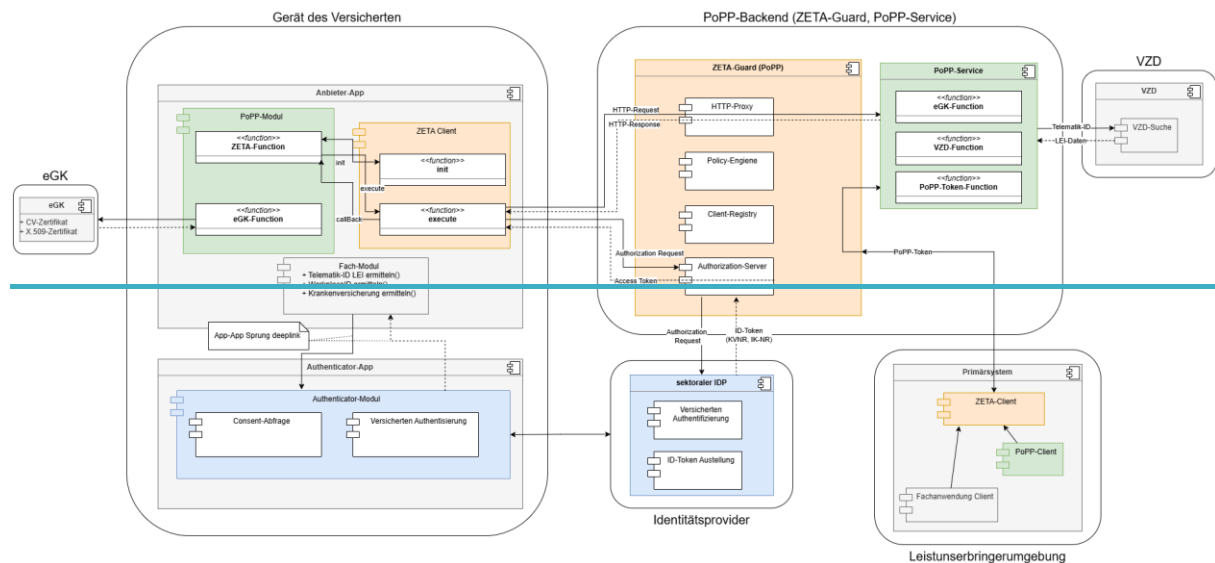


Abbildung 3: Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in Drittanbieter App

Da `mobile_CheckIn.yaml` enthält die API-Definition für die Authentifizierung einer eGK, nur notwendige Implementierung im 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.

Offener Punkt: OP-PoPP-3 (DeepLink)

Die derzeit spezifizierte Übergabe von Informationen zwischen Anbieter-App und Kassen-App mittels HTTP-POST über Deep Links ist hinsichtlich ihrer technischen Umsetzbarkeit noch nicht abschließend geklärt. Insbesondere ist offen, ob die vorgesehene Übergabeform durch die relevanten mobilen Betriebssysteme und die hierfür vorgesehenen Mechanismen zuverlässig unterstützt wird.

Für die Umsetzung des Online Check-in wird gegebenenfalls eine alternative Lösung auf Basis von App-Links und einer parameterbasierten Übergabe innerhalb der Ziel-URL erforderlich.

Vorgehen:

Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Die betroffenen Spezifikationsstellen werden identifiziert und konsistent angepasst. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet.

4 Übergreifende Festlegungen

A_30075 -PoPP-Modul - Zulassungsunterlagen

Der Hersteller des PoPP-Moduls MUSS für die Durchführung des funktionalen Zulassungstests der Zulassungsstelle der gematik folgende Unterlagen und Artefakte bereitstellen:

- Barrierefreiheitserklärung,
- Datenschutzerklärung,
- Datenschutzfolgenabschätzung (so vorhanden).

[<=]

A_30177 -PoPP-Modul – Integration in Kassen-App bei Nutzung der GesundheitsID

Das PoPP-Modul MUSS bei Durchführung eines Online Check-in unter Verwendung der GesundheitsID in einer Kassen-App auf dem Smartphone der versicherten Person integriert sein.

Das für den Online Check-in verwendete PoPP-Modul MUSS durch die gematik zugelassen sein.

[<=]

A_29040 -Implementierung des PoPP-Moduls in Kassen-App mit Authenticator-Modul des sektoralen IDP

Der Hersteller einer Kassen-App MUSS sicherstellen, dass das PoPP-Modul, das Authenticator-Modul des sektoralen IDP der Krankenkasse und ein ZETA Client integriert sind.

Das Frontend des Authenticator-Moduls und des PoPP-Moduls soll konsistent gestaltet sein, sich nahtlos ineinander fügen und die bestehenden UI-Patterns konsequent übernehmen.

Die Installation der App auf dem Gerät der VER MUSS auch ohne Durchlaufen des Identifikationsprozess und Anlage der GesundheitsID möglich sein, da für die Nutzung des PoPP-Modul nicht zwingend die Einrichtung der GesundheitsID notwendig ist.[<=]

Hinweis: Wird die Anforderung nicht umgesetzt, so kann die Funktion des App2App-Flow bzw. Web2App-Flow der GesundheitsID nicht nachgenutzt werden. Für die Verwendung der PoPP-Modul-Funktionen aus Anbieter-Apps müssen dann Alternativen entwickelt werden, die die Komplexität des Gesamtsystems erhöhen.

A_30012 -Unterstützung von Anbieter-Apps

Hersteller der App mit PoPP-Modul DÜRFEN Anbieter-Apps und gültige Telematik-IDs von der Nutzung des PoPP-Moduls NICHT ausschließen.

Dabei muss der Absprung von der Anbieter-App in die App mit PoPP-Modul ohne zusätzlichen Account und ohne nicht erforderliche Einrichtungsschritte möglich sein.

Davon ausgenommen sind gesetzlich notwendige Schritte, beispielsweise die Zustimmung zu Nutzungsbedingungen.[<=]

Hinweis: Wird die Anforderung nicht umgesetzt, so kann die Funktion des Online Check-In von Anbieter-Apps, wie z.B. Videosprechstunde und Online-Apotheke nicht genutzt werden.

4.1 Datenschutz und Informationssicherheit

A_30033 -PoPP-Modul - App - Unveränderte Kommunikation zum ZETA Client

~~A_27220 -PoPP-Modul (ZETA Client) - TLS-Verbindungsaufbau~~ Die App, die das PoPP-Modul bzw. der den ZETA Client integriert, MUSS beim TLS-Verbindungsaufbau zu allen Endpunkten des PoPP-Service das TLS-Server-Zertifikat des PoPP-Service validieren und dabei prüfendurchsetzen, 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 Kommunikation zwischen diesen Modulen direkt und unverändert stattfindet. [<= TLS-Verbindung aufbauen. [<=]~~

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

A_27621 -PoPP-Modul - Einmalige Einwilligung der VER in die Datennutzung durch PoPP-Service

~~Das PoPP-Modul —Einwilligung des Versicherten in die Datennutzung~~
Das PoPP-Modul MUSS MUSS, im Zuge vor der erstmaligen Durchführung einer Verifikation der Versichertenidentität Identität der VER sicherstellen, dass ~~der Versicherte, bevor seine~~ die VER, bevor ihre KVNR an den PoPP-Service übermittelt wird, ~~seine~~ ihre Einwilligung (Consent) zur Nutzung seiner KVNR durch ~~die LEI~~ den PoPP-Service erteilt. [<=]

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

A_30048 -PoPP-Modul - Einwilligung der VER in die Datennutzung durch SMC-B-Inhaber

Das PoPP-Modul MUSS, im Zuge der Verifikation der Identität der VER sicherstellen, dass die VER, bevor ihre KVNR an den PoPP-ServiceResource Server übermittelt wird, ihre Einwilligung (Consent) erteilt, dass der Inhaber der Telematik-ID berechtigt wird, auf Daten der VER zuzugreifen. Es ist zulässig, bei Inhabern einer Telematik-ID, die als Favoriten gespeichert sind und für die somit bereits eine Einwilligung stattgefunden hat, auf eine erneute Einwilligung zur Datennutzung zu verzichten.

~~Hinweis: Die Einwilligungsabfrage soll mit der Anzeige der Informationen aus dem VZD-Abruf (vgl. A_28488*) in einem Schritt erfolgen. Hat bereits eine Einwilligung stattgefunden und wird somit ggf. auf eine erneute Einwilligung zur Datennutzung~~

für den Inhaber der Telematik-ID verzichtet (als Favorit gespeichert), ist dennoch eine Anzeige und Bestätigung des Check-in entsprechend A_28488 immer zwingend erforderlich. [=]*

~~A_28488 - PoPP-Modul - Anzeige LEI oder DiGA Informationen aus VZDPoPP-Modul - Anzeige von VZD-Informationen zu einer Telematik-ID~~

~~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:~~

~~Das PoPP-Modul MUSS der VER vor dem Check-in immer die Klartextinformationen aus dem VZD zum Inhaber der Telematik-ID und den ggf. von der VER gesetzten Favoritenamen (vgl. A_28512*) anzeigen und durchsetzen, dass die VER den Check-in für diese Institution aktiv bestätigt.~~

~~Das PoPP-Modul MUSS dabei:~~

- ~~• aus dem "OrganizationDirectory" das Feld "type"/"coding"/"display" anzeigen (fachliche Rolle / Art der Institution),~~
- ~~• aus dem "OrganizationDirectory" das Feld "name" anzeigen (Anzeigenname der Institution),~~
- ~~• aus dem "OrganizationDirectory" das Feld "alias" anzeigen, sofern es vorhanden und befüllt 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 kann die VER die Anfrage mit geänderten Suchkriterien wiederholen oder den Check-in-Vorgang abbrechen, dem Versicherten einen verständlichen. Der VER ist bei Abbruch ein verständlicher Hinweis zum Fehler zu geben und ihm sie ist dabei auffordern aufzufordern, sich an die Institution zu wenden, für die ersie den Check-in durchführen wollte. [=]~~

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

~~A_28629 - PoPP-Modul - Prüfen der WorkplaceID~~

~~Das PoPP-Modul MUSS sicherstellen, dass die WorkplaceID aus maximal 64 alphanumerischen Zeichen besteht, wobei die folgenden Zeichen nicht zulässig sind: \ / : * ? " < > | [=]~~

Das PoPP-Modul sollte sicherstellen, dass der Check-in zu einer Telematik-ID von der VER nicht mehrfach für den gleichen Zeitraum ausgeführt werden kann. Der PoPP-Service Resource Server prüft zwar, ob im System noch ein PoPP-Datensatz im Status "pending" zur Telematik-ID und KVNR existiert, aber das PoPP-Modul sollte die UX so benutzerfreundlich gestalten, dass der mehrfache Check-in-Versuch unterbunden wird.

A_30130 -PoPP-Modul - Sichere Speicherung von Informationen

Das PoPP-Modul MUSS temporäre oder persistent zu speichernde Informationen so speichern, dass diese nicht durch andere Apps und Prozesse auf dem Gerät, auf dem das PoPP-Modul läuft, gelesen oder verändert werden können und dafür die Standardfunktionen der Plattform (Android, iOS) verwenden. [<=]

4.2 Barrierefreiheit

Die Entwicklung einer barrierefreien und gebrauchstauglichen Anwendung ist ein kontinuierlicher Prozess, der während der gesamten Lebensdauer der App berücksichtigt werden muss. Die Barrierefreiheitserklärung sollte mindestens einmal jährlich und bei jeder wesentlichen Änderung der mobilen Anwendung aktualisiert werden. Barrierefreiheit basiert auf den vier internationalen WCAG-Grundprinzipien - wahrnehmbar, bedienbar, verständlich und robust - die in allen relevanten Regelwerken verankert sind.

Gesetzliche Vorgaben

- Behindertengleichstellungsgesetz (BGG): Soll die Benachteiligung von Menschen mit Behinderungen beseitigen und verhindern und enthält spezifische Regelungen für barrierefreie Informationstechnik öffentlicher Stellen des Bundes
- BITV 2.0: regelt die konkrete Ausgestaltung der Barrierefreiheit in Deutschland für Webseiten und Apps öffentlicher Stellen gemäß dem BGG.
- Barrierefreiheitsstärkungsgesetz (BFSG): Verpflichtet seit 28. Juni 2025 private Unternehmen, ausgewählte digitale Produkte und Dienstleistungen – einschließlich Webseiten und Apps – barrierefrei zu gestalten. Es setzt die EU-Richtlinie 2019/882 („European Accessibility Act“) um.
- Verordnung zum Barrierefreiheitsstärkungsgesetz (BFSGV): Konkretisiert die Vorgaben des BFSG und beinhaltet unter anderem die Anforderungen an die barrierefreie Gestaltung von Benutzerschnittstellen und die Barrierefreiheitsanforderungen an Dienstleistungen im elektronischen Geschäftsverkehr (Onlineshops und Websites).
- EN 301 549: konkretisiert die technischen Anforderungen und verweist auf die WCAG 2.1 Level AA als Mindeststandard.

Relevante Normen

- ISO 9241-110 (Dialogprinzipien)
- ISO 9241-11 (Gebrauchstauglichkeit)

- ISO 9241-210 (Menschzentrierte Gestaltung)
- ISO 9241-171 (Barrierefreie Software)

Diese Normen der ISO-Normenreihe 9241 sind für ergonomische und barrierefreie Software maßgeblich und definieren wesentliche ergonomische und barrierefreie Vorgaben an interaktive Systeme.

Hieraus ergeben sich folgende Anforderungen:

Es ist ein einheitliches und normkonformes Niveau der digitalen Barrierefreiheit sicherzustellen.

A_30025 -PoPP-Modul - Umsetzung anerkannter Standards zur Barrierefreiheit

Die Anwendung, in welcher das PoPP-Modul integriert ist, MUSS die Benutzeroberflächen des PoPP-Moduls in Übereinstimmung mit den für den Anwendungsfall einschlägigen Anforderungen aus [BGG], [BITV 2.0], [BFSG], [BFSGV] und [EN 301 549] umsetzen. [≤]

5 Funktionsmerkmale

5.1 Allgemeine funktionale Anforderungen PoPP-Modul

A_28675 -PoPP-Modul - unterstützte Authentisierungsverfahren

Das PoPP-Modul MUSS ~~mindestens~~

- ~~die Authentifizierung des Versichertender VER mit dessen GesundheitsID~~
- ~~die Authentifizierung einer eGK (eGK in Fernversorgung)~~

unterstützen. [$\leq$]

A_29044 -PoPP-Modul - "Push Notifications"

Das PoPP-Modul MUSS den Anwendungsfall "Push Notifications" umsetzen [gemF_PushNotification] und sich dazu über das ZETA-API [API ZETA Client] beim PoPP-ServiceResource Server als Pusher registrieren. [$\leq$]

Hinweis: Die ZETA-Komponenten implementieren das "Client-FD Push API" [gemF_PushNotification] und stellen der nutzenden Komponente PoPP für die Verwendung der Push Notification selbst ein einfaches API zur Verfügung.

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

Das PoPP-Modul MUSS ~~Versicherten~~VER eine Funktion bereitstellen, über die ~~dieser~~diese die Historie der erstellten PoPP-Token einsehen kann. Die Historie MUSS Name ~~und~~, Anschrift ~~der LEI~~und Telematik-ID des Teilnehmers enthalten, für den das PoPP-Token ausgestellt wurde. [$\leq$; ~~;~~ $\leq$]

Hinweis 1: 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. [$\leq$]~~

~~Auswahl der LEI/~~*Hinweis 2: Die Telematik-ID des Teilnehmers muss den VER in der Historie nicht mit dargestellt werden.*

A_29036 -PoPP-Modul - Löschen von Einträgen in der Historie

Das PoPP-Modul MUSS sicherstellen, dass VER Einträge aus der Historie löschen können. Einträge, die älter als 12 Monate sind, MUSS das PoPP-Modul automatisch aus der Historie entfernen. [$\leq$]

~~5.21.1 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~~Ein durch das PoPP-Modul im VZD-Suche gefundene LEI gefundenen Eintrag muss der Nutzer die VER nach Anzeige und Bestätigung der VZD-Daten (vgl. A_28488* und A_27621*) als Favorit markieren können.
- Zu Favoriten wird der Name aus dem VZD-Eintrag "OrganizationDirectory"/"name"/"alias" und die Telematik-ID gespeichert.
- Nutzer/VER 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.~~

5.2.25.1.1 [~~<=>~~]

~~5.2.31.1.1 Verzeichnisdienstsuche~~

Hinweis: Es ist unerheblich, was

~~A_28515 -PoPP-Modul -Unterstützung~~ der Auslöser für die VZD-Suche ~~nach LEIs und DiGAs~~

~~Das PoPP-Modul MUSS den ist (Suche durch 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~~

~~5.2.45.1.2 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. [<=]~~

~~5.2.51.1.1 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"
im PoPP-Modul, wpid	WorkplaceID; String entsprechend A_28629* — optional	"REZEPTION-1"

~~Eingabe Telematik-ID durch die VER, Übergabe Telematik-ID beim App-Sprung, Scannen Telematik-ID per QR-Code usw.). Die Möglichkeit zum Speichern eines Favoriten muss immer gegeben sein. [<=]~~

~~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
- KVNR 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.

5.4 an

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.~~

~~5.4.1 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.

5.55.2 die Kommunikation ~~zwischen~~ des PoPP-Modul und PoPP-Service läuft über ~~diemit~~ dem ZETA-Komponenten: Client

~~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 das `apduCommand`.~~
- ~~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 (`statusPoPPGeneration`).~~
- ~~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.

5.5.1 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~~ einer Kassen-App. Der ZETA Client überträgt bei der Initialisierung der Anwendung die Geräte- und App-Informationen an den ZETA Guard ~~des PoPP-Service~~.

Im laufenden Prozess kapselt der ZETA Client die gesamte Kommunikation zwischen der Anwendung auf dem ~~Gerät des Versicherten-Smartphone der VER~~ und den Komponenten im Backend, u.a. auch zwischen PoPP-Modul und ~~PoPP-Service~~. `ServiceResource Server`. 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. ~~Wird das von der gematik bereitgestellte [ZETA-SDK] verwendet, muss~~ das PoPP-Modul ~~muss~~ die ~~bereitgestellten~~ API-Schnittstellen des ZETA-~~Client~~-SDK 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 ~~und dabei den FQDN des PoPP-Service Resource Server verwenden.~~ **[<=:[<=]**

5.3 Bestimmen der Telematik-ID

Die Telematik-ID, für den ein PoPP-Token erstellt werden soll, kann je nach Anwendungsfall über unterschiedliche Wege ermittelt werden:

- Übergabe der Telematik-ID an der Schnittstelle einer Anbieter-App zum PoPP-Modul - in diesem Fall ist die Ermittlung der Telematik-ID im Funktionsumfang der Anbieter-App
- Übergabe eines FHIR-VZD Search Request an der Schnittstelle einer Anbieter-App zum PoPP-Modul - in diesem Fall ist die Erstellung des FHIR-VZD Search Request im Funktionsumfang der Anbieter-App. Die Ausführung des FHIR-VZD Search Request und damit die Ermittlung der Telematik-ID ist im Funktionsumfang des PoPP-Modul.
- Erstellung des FHIR-VZD Search Request im PoPP-Modul - nach Ausführung des FHIR-VZD Search Request wird im PoPP-Modul die Telematik-ID aus dem Ergebnisdatensatz ermittelt
- Ermittlung der Telematik-ID aus einem QR-Code
- Verwendung der Telematik-ID aus einem früheren Check-in Prozess (Favoritenliste, Historie)

Weitere Verfahren sind denkbar aber nicht Teil der Spezifikationen (manuelle Eingabe, c&p aus Drittmedien).

5.3.1 QR-Code

Das PoPP-Modul bietet die Möglichkeit eine Telematik-ID aus einem QR-Code zu extrahieren. Dann gelten die in [gemILF_PoPP_Client] getroffenen Festlegungen zum Inhalt des QR-Codes und das PoPP-Modul prüft, ob diese Festlegungen eingehalten werden.

Tabelle 2: Tab_PoPP_Modul_Payload_stat_QRCode]: Payload QR-Code

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

Hinweis: Ob eine WorkplaceID verwendet wird, und wie diese gebildet ist, liegt in der Verantwortung des Telematik-ID-Inhabers.

A_30120 -PoPP-Modul, QR-Code einlesen

Das PoPP-Modul MUSS QR-Codes mindestens auf folgenden Wegen einlesen:

1. Scannen mit dem Smartphone der VER, direkt über die Schnittstellen des Betriebssystems (also gerade nicht über Schnittstellen anderer Apps),
2. Importieren eines QR-Codes aus einer Bilddatei auf dem Smartphone der VER.

[<=]

A_27595 -PoPP-Modul - QR-Code, Validierung und Warnung der VER

Das PoPP-Modul MUSS sicherstellen, dass:

- aus einem QR-Code extrahierte Informationen nur nach einer erfolgreichen, sicherheitstechnischen Validierung verwendet werden,
- die Inhalte dem erwarteten Schema entsprechen (siehe Tabelle [Tab_PoPP_Modul_Payload_stat_QRCode]),
- insbesondere keine Links aus dem QR-Code automatisch aufgelöst werden oder die VER weitergeleitet wird, falls der QR-Code eine URL enthält,
- im Falle eines Scheiterns der Validierung die VER
 - gewarnt wird und
 - an das Personal des Inhabers der Telematik-ID verwiesen wird, der den QR-Code bereitstellte.

[<=]

5.3.2 Verzeichnisdienstsuche

A_28515 -PoPP-Modul - Suche nach Inhabern einer Telematik-ID im FHIR-VZD

Das PoPP-Modul MUSS vor jedem Request zur PoPP-Token-Erstellung eine FHIR-VZD-Suche mit Suchkriterien oder einer Telematik-ID durchführen um sicherzustellen, dass der Inhaber der Telematik-ID valide und berechtigt ist, ein PoPP-Token zu erhalten. Die Ermittlung der VZD-Suchdaten erfolgt z.B. per QR-Code-Scan, bei externem App-Abruf übergebenen Telematik-ID oder durch Eingabe von Suchkriterien durch die VER.

Ist keine Telematik-ID vorhanden, MUSS das PoPP-Modul die VER eine Funktion anbieten, über die diese eine Suche nach Inhabern und Telematik-IDs im FHIR-VZD der TI durchführen kann.

Die gezielte Eingabe von Suchkriterien soll die VER bei einer erfolgreichen Suche der richtigen Gesundheitseinrichtung unterstützen. Dabei sollte die Suche wo sinnvoll und möglich durch folgende Kriterien unterstützt werden:

- Art der Einrichtung(Apotheke, Krankenhaus, Zahnarztpraxis, Kostenträger / Krankenversicherung usw. jedoch ausschließlich Leistungserbringerinstitutionen, DiGAs und Kostenträger)
- Ort
- PLZ
- Standort & Radius (nicht bei allen Einrichtungen verfügbar)
- Spezialisierung (Allgemeinmedizin, Neurologie, Sprachtherapie, ...)
- Physische Faktoren(Parkmöglichkeit, ÖPNV in der Nähe, Barrierefrei, Abholautomat)
- Öffnungszeiten, Notdienste, Sonderschließzeiten
- Apothekenservices(Pharmazeutische und medizinische Leistungen, Einlösewege)

Hinweis: Welche Suchkriterien den Anwendern zur Verfügung gestellt werden, hängt u.U. von bereits getroffen Auswahlkriterien ab. "Apothekenservices" ist beispielsweise nur bei der Vorauswahl "Art der Einrichtung = Apotheke" sinnvoll. [<=]

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

Hinweis: Welche Suchkriterien der VER angeboten werden ist Anwendungsfall spezifisch. So ist für einen Praxisbesuch eine Umkreissuche hilfreich, für Videosprechstunde aber nicht. Inhaltspunkte zu Suchkriterien sind im Implementierungsleitfaden beschrieben.

Hinweis: Die Schnittstelle ist in [II_PoPP_Service_mobile_CheckIn.yaml] definiert. Informationen zur Formulierung eines FHIR-VZD-Search Requests ind unter [\[https://github.com/gematik/api-vzd/blob/main/docs/FHIR_VZD_HOWTO_Search.adoc\]](https://github.com/gematik/api-vzd/blob/main/docs/FHIR_VZD_HOWTO_Search.adoc) zu finden. [$\leq$]

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~~ MUSS der nach A_28514* erzeugte HTTP-Request als Parameter ~~zu~~ übergeben werden. [$\leq$]

÷

A_28600 -Ergebnisverarbeitung der VZD-Abfrage

Das PoPP-Modul MUSS das Suchergebnis der FHIR-VZD-Abfrage aus dem Body der Response Objektes, welches vom ZETA-Client an das PoPP-Modul als Antwort auf den `httpLoadPractitionerInformationRequest` übergeben wird, extrahieren und hinsichtlich A_28488* auswerten.

Das PoPP-Modul MUSS sicherstellen, dass ausschließlich Leistungserbringerinstitutionen, DiGAs und Kostenträger in die Ergebnisliste übernommen werden - also insbesondere keine weiteren Organisationen und Personen.

Liefert die FHIR-VZD-Abfrage mehr als 50 Treffer MUSS das PoPP-Modul die VER darauf hinweisen, dass die Ergebnisliste unvollständig ist und für ein besseres Suchergebnis die Suche weiter eingegrenzt werden sollte. [$\leq$]

5.4 Anforderungen PoPP-Modul bei Authentisierung einer VER mit GesundheitsID

Die Abläufe der Erzeugung eines PoPP-Token über die Authentifizierung der VER 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 den Inhaber einer bestimmte Telematik-ID aus. Dabei prüft der ZETA Client, ob in der laufenden Session bereits eine Authentifizierung der VER 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 der VER wird vom ZETA Client der eigentliche HTTP-Request (siehe A_28501*) an den PoPP-ServiceResource Server 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 VER-Authentifizierung

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

- ~~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_27605 -PoPP-Modul - Authentifizierung mit SSO im ePA-FdV

Sind PoPP-Modul und ePA-FdV in der gleichen App der Krankenversicherung integriert, so KANN für die Authentifizierung der VER Single-Sign-On (SSO) gemäß [gemSpec_IDP_Sek] verwendet werden.[<=]

Hinweis: Das SSO auf Anwendungsebene ist derzeit nur für Apps mit zugelassenem ePA-FdV erlaubt (siehe [gemSpec_IDP_Sek]).

~~A_28619 -PoPP-Modul - Aufruf des ZETA-Client zum Auslösen der Erstellung eines PoPP-Token mit eGK~~*Hinweis: Da PoPP-Modul, ZETA Client und Authenticator-Modul Funktionen in und derselben App sind, handelt es sich immer um die 1-App-Strategie. Das PoPP-Modul MUSS, 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-GET-Request `httpCheckInGIDRequest` zum Auslösen der Erstellung eines PoPP-Token mit der nach Authentifizierung des Versicherten mit GesundheitsID erstellen. Dabei MÜSSEN:

- ~~einer eGK ausgelöst werden soll, die API des ZETA-Client gemäß [API-ZETA-Client] aufrufen. Beim Aufruf sind folgende~~Telematik-ID des Inhabers der Telematik-ID und
- die WorkplaceID des Inhabers der Telematik-ID (wenn vorhanden)

als Parameter ~~zu~~ übergeben werden.[<=]

÷

- ~~der nach A_28518* erzeugte HTTP-Request zur eGK-Prüfung; Hinweis: Die Schnittstelle ist in [PoPP_Service_mobile_CheckIn.yaml] definiert.~~
- ~~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_28502 -PoPP-Modul - Erstellen der "authorization_details" zum Auslösen der Erstellung eines PoPP-Token mit GesundheitsID

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

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

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 **der VER** 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`

[<=]

5.5.2 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.~~

A_28503 -PoPP-Modul - Ergebnisinterpretation nach Abschluss der Erstellung eines PoPP-Token mit GesundheitsID

Das PoPP-Modul MUSS das Ergebnis der der Erstellung eines PoPP-Token mit GesundheitsID aus dem Body der Response Objektes, welches vom ZETA Client an das PoPP-Modul als Antwort auf `denhttpCheckInGIDRequest` übergeben wird, extrahieren und auswerten.[<=]

~~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~~

Hinweis: Das Objekt zum Status der PoPP-Modul MUSS eine HTTP-Schnittstelle anbieten, über Generierung enthält die Drittanbieter-AppsInformation, ob die Erstellung eines des PoPP-Token erfolgreich war.

- *success - PoPP-Token initiieren können. Die HTTP-Schnittstelle muss wurde erzeugt und dem Inhaber der Telematik-ID zugestellt*
- *pending - PoPP-Token wurde erzeugt und noch nicht zugestellt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung wird weiter versucht.*
- *als FQDN die Client-ID des sektoralen IDP der Krankenkasse haben, dessen Authenticator-Modul auf dem Gerät installiert ist. canceled - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit von 72h abgebrochen*
- *Folgende Parameter umfassen:*
 - *Telematik-ID (mandatory),*
 - *WorkplaceID (optional),*
 - *FHIR-VZD-Search-Request (optional)*
 - *callbackMethod (mandatory),*
 - *loginEgk (optional).*
- **{<=>}**

Zur Identifikation des im PoPP-Service Resource Server angelegten Datensatzes ist die ID des Nachrichten-Datensatz ebenfalls im vom PoPP-Service erstellten JSON-Objekt enthalten.

A_29037 -PoPP-Modul - Informationen zum Nachrichten-Datensatz

~~**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 zur ID des Nachrichten-Datensatzes, welche vom PoPP-ServiceResource Server als Ergebnis der PoPP-Token-Erzeugung des online Check-in Prozesses 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 zurückgegeben wird, die zugehörigen VZD-Informationen speichern, um 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 ErgebnisVER den Status des online Check-in zuzur 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)~~

anzeigen zu können.

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

5.65.5 Verarbeitung des Ergebnisses der PoPP-Token-Erzeugung

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

Das PoPP-Modul MUSS ~~dem Versicherten~~ der VER 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 des Telematik-ID Inhabers> erfolgreich.". Im Fehlerfall könnte die Information lauten: "Die Berechtigung von <LEI-Name Telematik-ID Inhabers> 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 **Resource Server** abzurufen. Die ID des angelegten **PoPPNachrichten**-Datensatzes muss als Query-Parameter im HTTP-GET Request übergeben werden. [<=]

*Hinweis: Die ID des **PoPPNachrichten**-Datensatzes ist Teil der Ergebnisinformationen der vom PoPP-Modul ausgelösten PoPP-Token-Erstellung. Konnte ~~das~~ der PoPP-Datensatz erstellt aber ein PoPP-Token ~~dem Inhaber~~ der **LEI-Telematik-ID** 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-**ServiceResource Server** anfragen und den Benutzer entsprechend informieren.*

5.6 Kommunikation PoPP-Modul mit Anbieter-Apps

Anbieter-Apps 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 Anbieter-App aufgerufen werden kann.

Das Ergebnis der PoPP-Token-Erzeugung übermittelt das PoPP-Modul, indem es eine HTTP-Schnittstelle aufruft, welche die Anbieter-App zur Verfügung stellen muss.

Die Anwendung, aus der ein Check-in gestartet werden soll, kann als eigenständige App (App2App-Flow) oder als Web-App (Web2App-Flow) auf dem Smartphone der VER ausgeführt werden, auf dem auch die App mit PoPP-Modul, ZETA Client und Authenticator-Modul installiert ist. Ist die Anwendung, aus der ein Check-in gestartet werden soll auf einem anderen Gerät installiert, wird der 2-Geräte-Flow durchlaufen. Die genannten Abläufe werden im Folgenden kurz dargestellt. Eine detaillierte Ablaufbeschreibung und ein Flowdiagramm ist im Anhang dieses Dokuments dargestellt.

5.6.1 App2App-Flow

Beim App2App-Flow wird beim Aufruf des Check-in Requests aus der Anbieter-App direkt die App mit dem PoPP-Modul geöffnet. Dazu muss die App mit dem PoPP-Modul so konfiguriert sein, dass die Funktionen der jeweiligen Plattform (deeplink bzw. universal link) ausgenutzt werden.

Für das PoPP-Modul gelten hier die gleichen Rahmenbedingungen, wie für das Authenticator-Modul des sektoralen IDP. Da beide Module in der gleichen App implementiert sind, wird die Konfiguration bereits bei der Installation der App sichergestellt.

Der Rücksprung aus dem PoPP-Modul ist technisch identisch über die Konfiguration der Anbieter-App abgebildet. Die vom PoPP-Modul aufgerufene callBack-Url öffnet die Anbieter-App, die dann den eingegangenen Request verarbeitet und die Information für die VER aktualisiert.

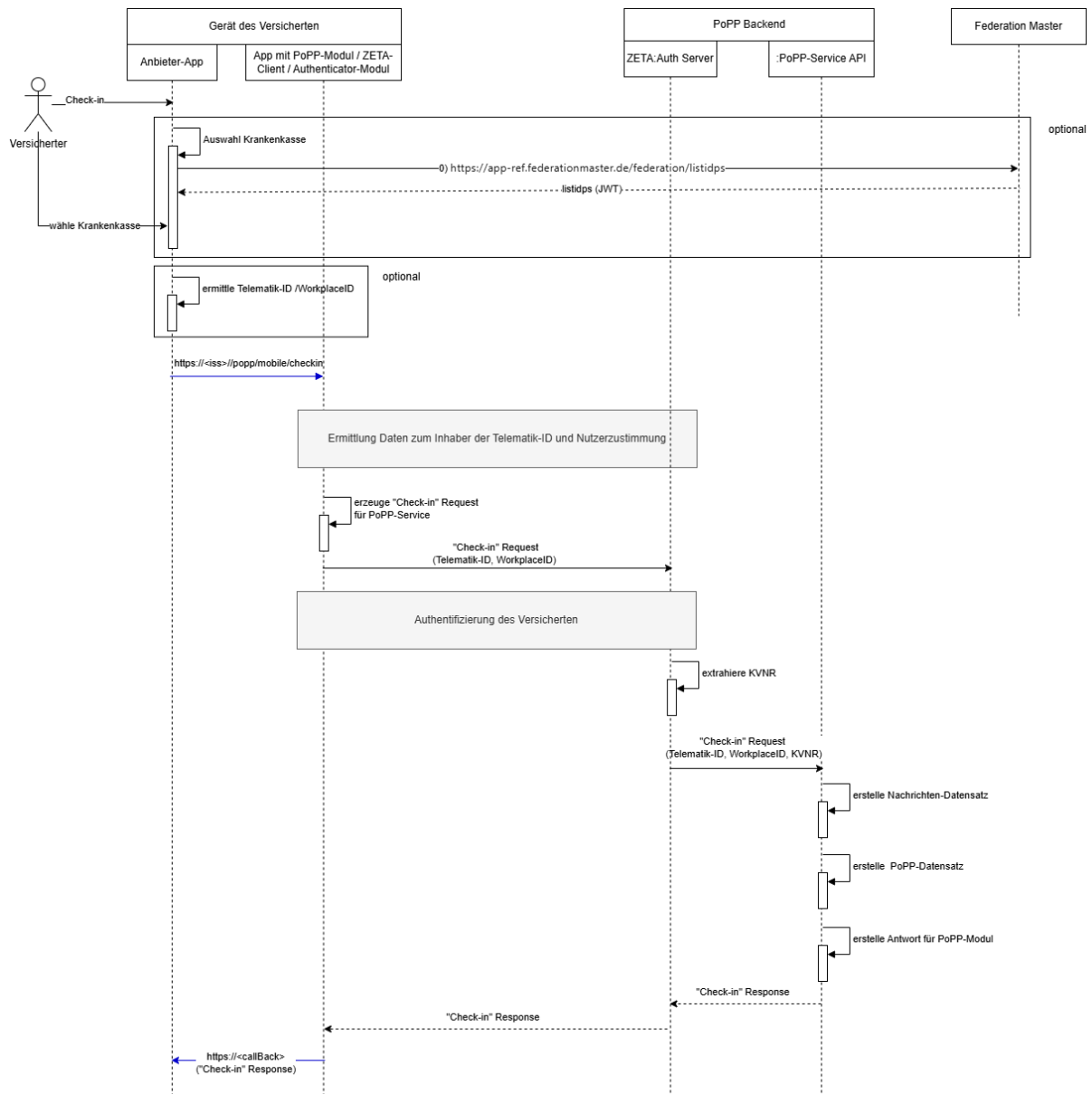


Abbildung 3 : Ablauf "Check-in" aus einer Anbieter-App

5.6.2 Web2App-Flow

Der Ablauf für den Fall, dass die Anwendung des Anbieters eine Web-App ist, also in einem Browser läuft, ist fast identisch zum App2App-Flow. Der Unterschied besteht lediglich im Rücksprung aus dem PoPP-Modul. Zur vom PoPP-Modul aufgerufene callback-URL gibt es keine Konfiguration einer App über deeplink bzw. universal link, so dass der Aufruf am Web-Backend der Anwendung erfolgt. Das Web-Backend wertet den Request aus und aktualisiert die Darstellung für den Anwender.

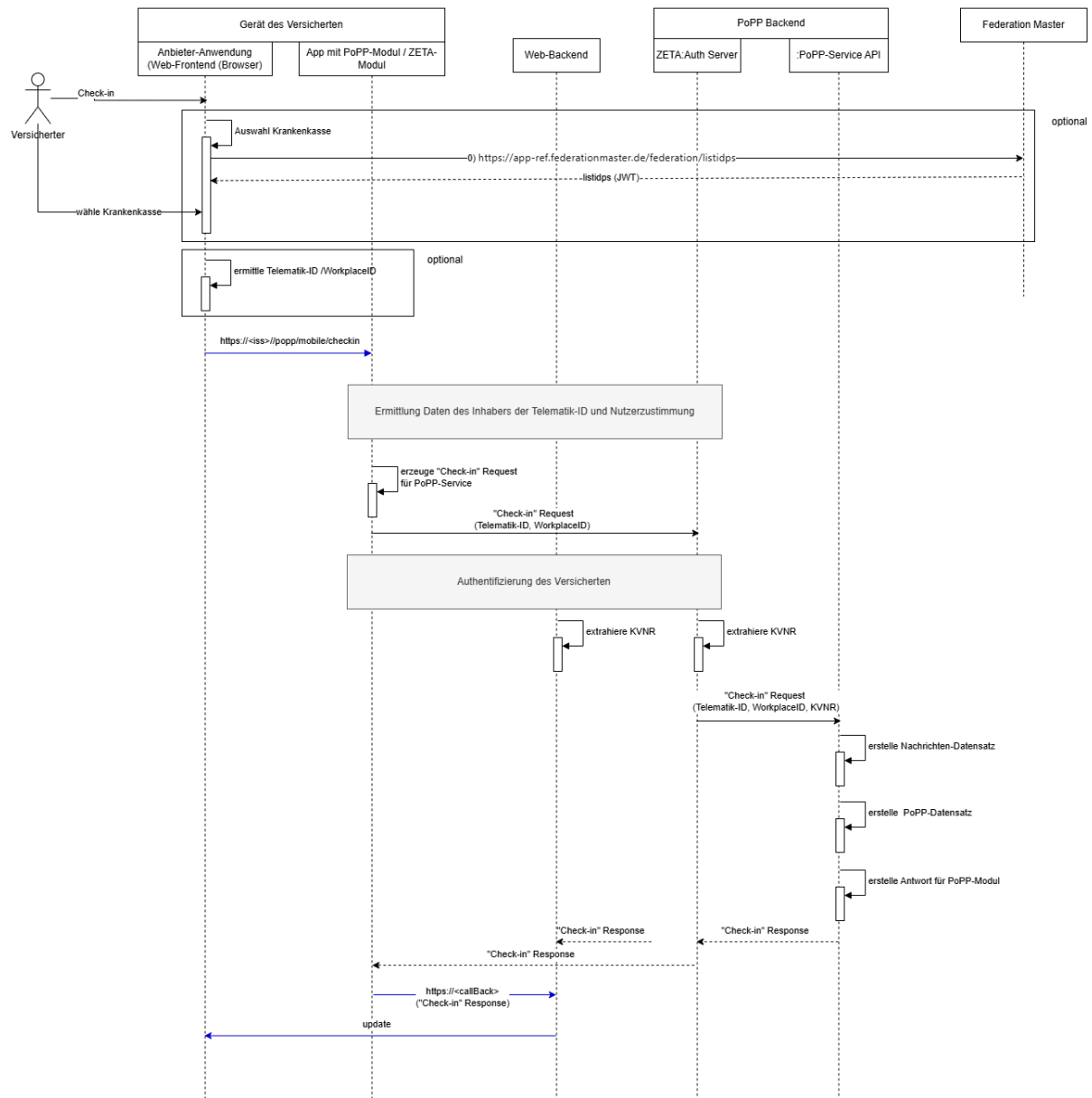


Abbildung 4 : Abbildung : Ablauf "Check-in" aus einer Anbieter-WebApp

5.6.3 2-Geräte-Flow

Offener Punkt: OP-PoPP-4 (2-Geräte-Flow)

Wie der 2-Geräte-Flow abzubilden ist, ist derzeit noch offen.

Vorgehen:

Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet.

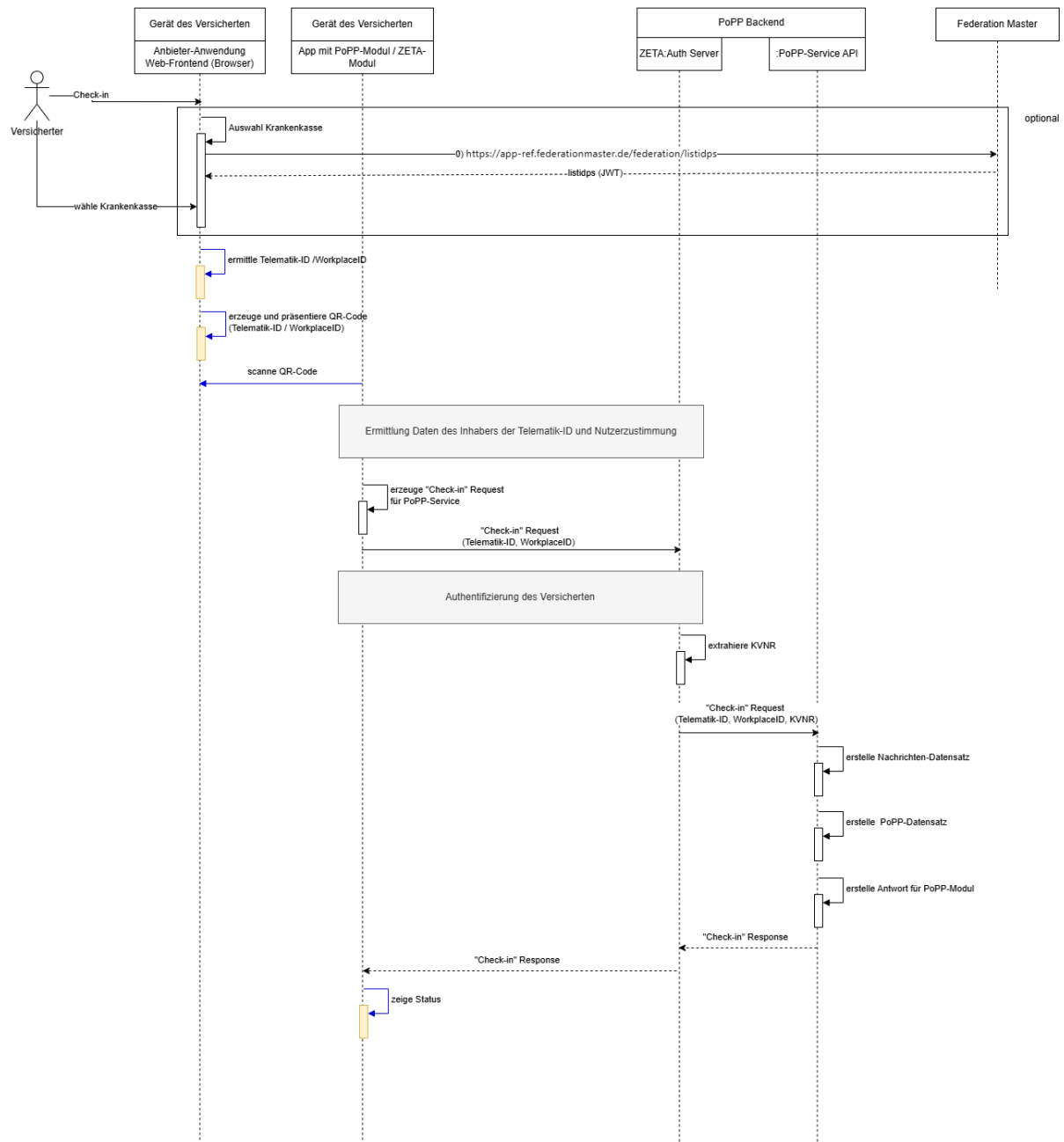


Abbildung 5 : Abbildung : Ablauf "Check-in" mit 2 Geräten

5.6.4 Anforderungen an die Kommunikation zwischen PoPP-Modul und Anbieter-Apps

A_29041 -Erreichbarkeit des PoPP-Modul aus Anbieter-App

Krankenversicherungen MÜSSEN sicherstellen, dass die App mit Authenticator-Modul ihres sektoralen IDP, PoPP-Modul und ZETA Client in ihrer Konfiguration je nach Betriebssystem Android/iOS einen deeplink/universal link hinterlegen. Der Aufruf des Links MUSS zum Öffnen der App führen. Dazu MUSS die App das Interface [I_PoPP_Modul_mobile_CheckIn.yaml] implementieren. Der Link MUSS der Struktur<iss>/<pfad> entsprechen, wobei iss die Client-ID des sektoralen IDP der Krankenversicherung in der TI-Föderation und pfad der Aufrufpfad gemäß [I_PoPP_Modul_mobile_CheckIn.yaml] ist. Die App MUSS bei Aufruf des Links das PoPP-Modul ansteuern und die Parameter im Aufruf interpretieren. [≤]

Hinweis: Wird kein deeplink/universal link hinterlegen, so wird der Anwender auf eine Web-Seite des sektoralen IDP der Krankenkasse geleitet. Dort erhält er Informationen zur Installation der App mit integriertem PoPP-Modul.

A_29045 -PoPP-Modul - Unveränderbarkeit der WorkplaceID

Das PoPP-Modul MUSS sicherstellen, dass eine von einer Anbieter-App gesetzte WorkplaceID nicht durch die VER gelöscht oder verändert werden kann. [≤]

A_28510 -PoPP-Modul - Erzeugung und Versand eines HTTP-Request an callbackURL einer Anbieter-App

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

A_30102 -Anbieter-APP - Implementierung der Callback Schnittstelle

Anbieter-APPs, die das Check-in über ein PoPP-Modul in einer Kassen-App verwenden wollen, MÜSSEN die Schnittstelle [I_AnbieterApp_mobile_CheckIn.yaml] implementieren. Diese wird vom PoPP-Modul zur Übergabe der Statusinformationen nach Beenden des Check-in-Prozess aufgerufen. [≤]

Ist die Abfrage der PoPP-Token beim PoPP-ServiceResource Server durch den Inhaber der Telematik-ID im synchronen Ablauf noch nicht erfolgt, so ist der Status der PoPP-Token-Generierung "pending".

Das PoPP-Modul stellt eine HTTP-Schnittstelle bereit, über die die Anbieter-App eine Anfrage zum Statusupdate der PoPP-Token-Generierung an das PoPP-Modul senden kann.

Das PoPP-Modul führt dann einen Aufruf an den PoPP-Service Resource Server durch, um den aktuellen Status der PoPP-Token-Generierung zu ermitteln.

Das Ergebnis der Statusabfrage übermittelt das PoPP-Modul, indem es eine HTTP-Schnittstelle aufruft, welche die Anbieter-App zur Verfügung stellen muss.

Offener Punkt: OP-PoPP-5 - (Anforderungen an Anbieter-App)

Im Zusammenspiel mit dem PoPP-Token-Abruf ergeben sich Anforderungen an die Anbieter-Apps - das sind zum Beispiel Videosprechstunden-Apps, die PoPP nutzen und selbst kein PoPP-Modul integrieren.

Es ist noch offen, ob es hierfür einen eigenen Steckbrief gibt, bzw. an wen und wie dieses Anforderungen im Anforderungsmanagement geführt werden.

Vorgehen:

Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet.

6 Informationsmodell

Das abgebildete Informationsmodell bezieht sich ausschließlich auf die Anteile des Gesamtmodells für das ~~Gerät des Versicherten~~ Smartphone der VER und des PoPP-ServiceServiceResource Server für das Ausstellen einen PoPP-Token nach Authentifizierung ~~des Versicherten~~ der VER mit GesundheitsID oder Authentifizierung einer eGK über das PoPP-Modul in einer ~~Krankenversicherung~~ Kassen-App auf dem ~~Gerät des Versicherten~~ Smartphone der VER.

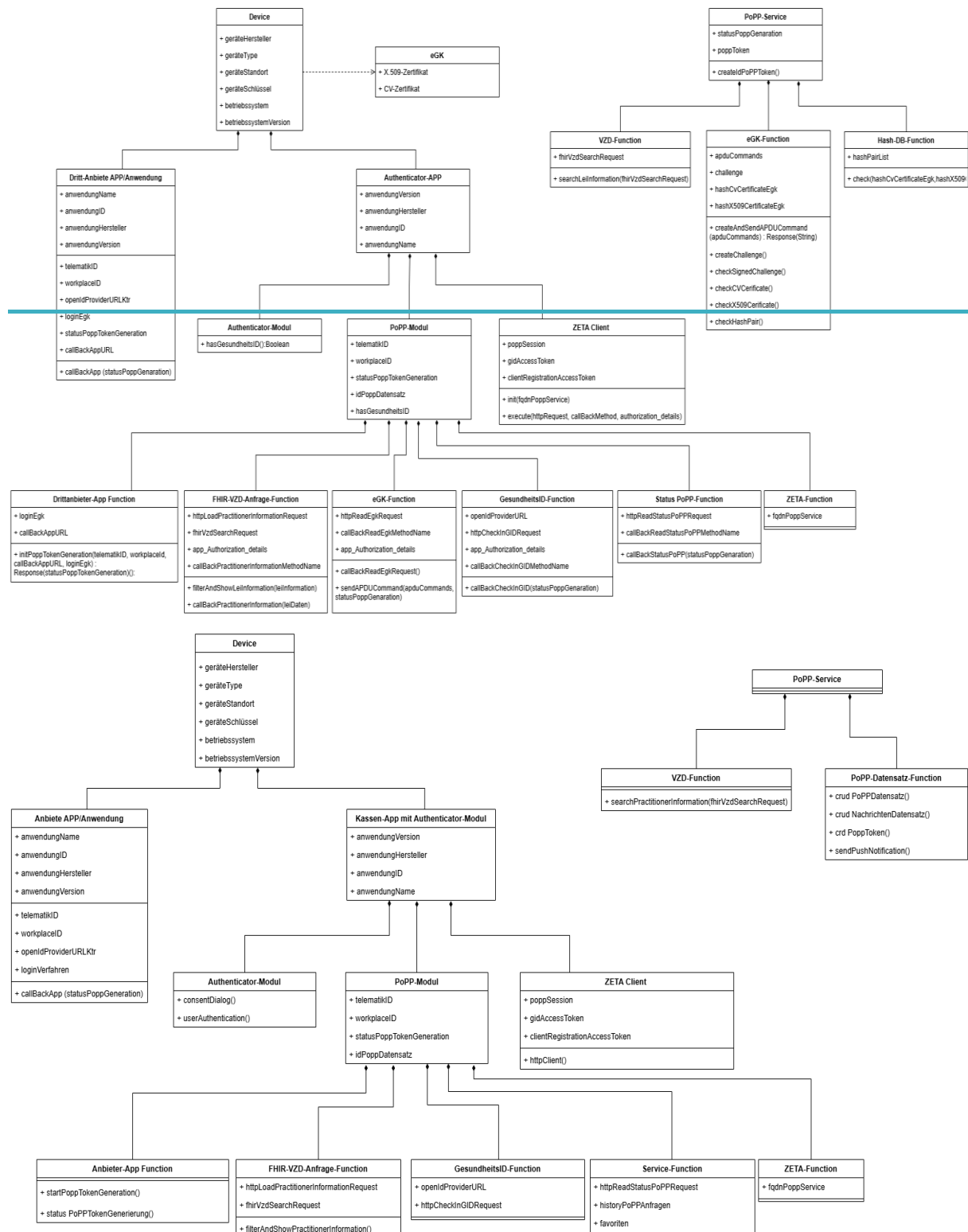


Abbildung 6: Bausteinsicht - Informationsmodell ~~Gerät des Versicherten Smartphone~~ der VER und PoPP-Service **Resource Server für Online-Anwendungsfälle zur PoPP-Token-Erstellung**

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-PathReferenzpfad**)

Zunächst wird der **Happy-PathReferenzpfad** beschrieben, der die Umsetzung des Online –Check-ins mit PoPP in einer **KrankenversicherungsKassen**-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:~~ Check-in mit der GesundheitsID in Apps umsetzen. Die Anbieter-App nutzt dafür das PoPP-Modul einer Kassen-App nach. Der eigentliche Online Check-in wird in der Kassen-App durchgeführt. Der Wechsel in den Kassen-App und wieder zurück erfolgt über Plattformmechanismen (deeplink/universal link).

- ~~• 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-PathReferenzpfads** werden Alternativen zu den einzelnen Flow-Schritten aus dem **Happy-PathReferenzpfad** aufgeführt, um die unterschiedliche Rahmenbedingungen von Use Cases zu berücksichtigen:

- ~~• Autorisierung: Variante zur Nutzung der eGK ohne PIN statt GesundheitsID~~
- ~~Auswahl der LEI~~Erfassung Telematik-ID: 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 **VorgabenEmpfehlungen** zur Terminologie genannt.

7.1 End-to-End Flow (**Happy-PathReferenzpfad**)

Der **Happy-PathReferenz** zeigt die Umsetzung des Online –Check-ins mit PoPP in einer **KrankenversicherungsKassen**-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 [KrankenversicherungsKassen](#)-App für einen Praxisbesuch mit GesundheitsID.

[Clickdummy]

[Clickdummy<https://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=2535-5273&viewport=597%2C4382%2C0.16&t=6QIfHZKYsaLuHn1P-9&scaling=scale-down&content-scaling=fixed&starting-point-node-id=2535%3A5273&show-protocol-sidebar=1>]

Ausgangssituation

Der [Happy PathReferenzpfad](#) umfasst nicht die Ersteinrichtung. Folgende Punkte werden entsprechend vorausgesetzt.

- [KrankenversicherungsKassen](#)-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.
- [LEITelematik-ID-Inhaber](#) 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 PathReferenzpfad](#).

	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

	ID	Handlung durch	Aktion
AuswahlErfassung der LEI-Telematik-ID	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 des SMC-B Inhabers wird angezeigt inkl. Adresse, tagesaktuelle Öffnungszeiten (sofern vorhanden) 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 [PoPPAuth-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 eingchecked

7.2 **DrittanbieterAnbieter-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 ~~DrittanbieterAnbieter-App mit integrierten~~, bei der das PoPP-Modul ~~heraus-~~ einer Kassen-App nachgenutzt wird (App2App), sind nicht alle im ~~Happy Path~~Referenzpfad aufgeführten Schritte erforderlich. Ausnahmen, wie bspw. Entfall des Flow-~~SchrittsAuswahl/SchrittsErfassung 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 eingecheckt 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 ~~DrittanbieterAnbieter-App~~ hinterlegt werden kann, sind in den jeweiligen Kapiteln benannt.

Clickdummy / Screenshot

Visuelle Umsetzung des Online -Check-ins in einer ~~DrittanbieterAnbieter-App~~ für ~~Online-Dienste einer Apotheke~~Videosprechstunden mit ~~eGKGesundheitsID~~.

~~[Clickdummy]~~

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

~~[ClickdummyClickdummyhttps://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=4822-55496&viewport=597%2C4382%2C0.16&t=6QIfHZKYsaLuHn1P-9&scaling=scale-down&content-scaling=fixed&starting-point-node-id=4822%3A55496&show-protocol-sidebar=1]~~

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~~Referenzpfad entfallen die Schritte Check-in und ~~AuswahlErfassung der LEI~~Telematik-ID.
- Der Einstieg sowie der Ausstieg können individuell durch die ~~DrittanbieterAnbieter-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 ~~DrittanbieterAnbieter~~-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-PathsReferenzpfads~~ beschrieben.

7.3.1 Autorisierung

~~Neben~~

7.3.1 Erfassung der ~~Autorisierung mittels GesundheitsID ist~~ Telematik-ID

Die Erfassung der Telematik-ID für den Online Check-in erfolgt in der Regel durch das Scannen eines QR-Codes. Alternativ kann auch eine ~~Autorisierung mit der eGK ohne PIN~~Anbieter-App die Telematik-ID bereits in der Anbieter-App hinterlegen, sodass die Erfassung der Telematik-ID durch die VER entfällt. Wird keine Anbieter-App verwendet oder ist das Scannen eines QR-Codes nicht möglich ~~Hierbei ändert sich nur der Flow-Schritt Autorisierung~~, sollen der VER alternative Varianten zu Erfassung der Telematik-ID zur Verfügung stehen.

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 eGKErfassung 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

	ID	Handlung durch	Aktion
	5.4	VER	Positioniert Karte
	5.5	APP [PoPP-Modul]	NFC-Dialog mit mindestens zwei Statusinformationen (Karte erkannt und Authentisierung erfolgreich)

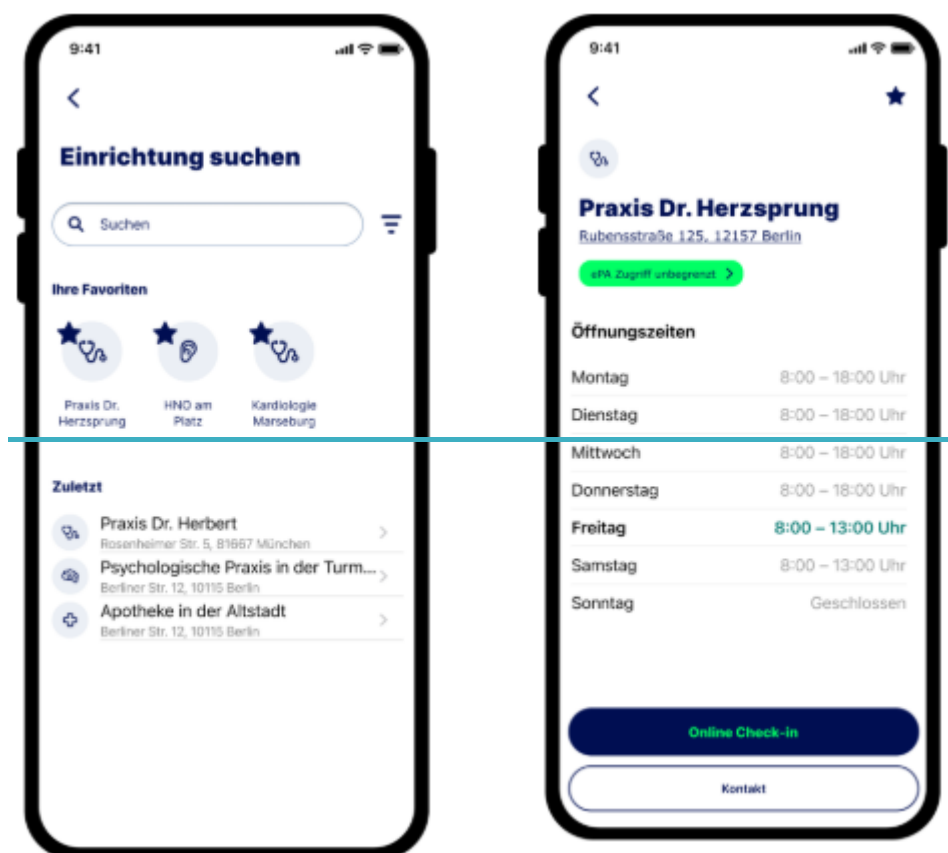
- Um der VER das kontaktlose Einlesen der eGK zu erleichtern, soll der VER die korrekte Positionierung der eGK in Abhängigkeit vom verwendeten Smartphone-Modell angezeigt werden. Für Smartphone-Modelle sollen allgemeine Hinweise zur optimalen Positionierung bereitgestellt werden, die auf den typischen 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/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 zu 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



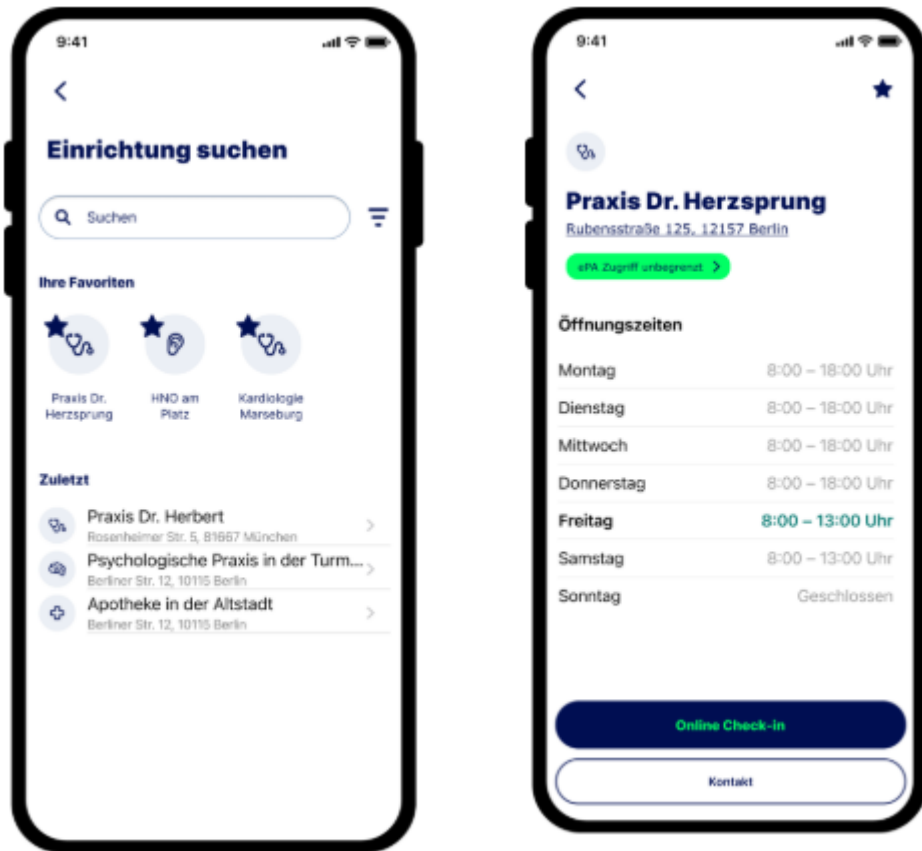


Abbildung 7: Screenshot des Flow-Schritts **AuswahlErfassung** der **LEITelematik-ID** mit VZD-Suche, Anzeige einer Historie der letzten Online -Check-ins, Favoriten und Detailscreens einer Einrichtung.

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 4: 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 anlegt wurden

Filter	Ergebnis	Hinweis
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, ...

Tabelle 5: 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.

Filter	Ergebnis	Hinweis
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 ausgewählten Faktoren an	Auswahl aus Liste und Suchfeld alphabetisch sortiert 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 6: 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

~~7.3.37.3.2~~ Statusinformation

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

~~Flow-Schritte und Vorgaben~~

Flow-Schritte und Vorgaben

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

~~7.3.47.3.3~~ 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 ~~KrankenversicherungsKassen~~-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]~~

~~[Clickdummy]~~<https://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=4729-53558&viewport=597%2C4382%2C0.16&t=6QIfHZKYsaLuHn1P-9&scaling=scale-down&content-scaling=fixed&starting-point-node-id=4729%3A53558&show-protocol-sidebar=1>

Visuelle Umsetzung des Online -Check-ins in einer ~~DrittanbieterAnbieter~~-App (App2App) für ~~einen Online-Dienst einer Apotheke~~ Videosprechstunden mit eGKGesundheitsID inkl. zusätzlicher Flow-Schritte für die Ersteinrichtung: Auswahl der ~~KrankenversicherungsKassen~~-App, ~~CAN-Eingabe~~ Einrichtung GesundheitsID und Zustimmung der VER zur Nutzung des Online-Check-ins.

~~[Clickdummy]~~

Visuelle Umsetzung des Online -Check-ins in einer Drittanbieter-App (~~Anwendung~~) (App2App) für Videosprechstunden mit GesundheitsID inkl. zusätzlicher Flow-Schritte für

die Ersteinrichtung: Auswahl der [Krankenversicherungskassen](#)-App, Einrichtung GesundheitsID und Zustimmung der VER zur Nutzung des Online -Check-ins.

[Clickdummy]

[Clickdummy<https://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=4836-56927&viewport=597%2C4382%2C0.16&t=6QIfHZKYsaLuHn1P-9&scaling=scale-down&content-scaling=fixed&starting-point-node-id=4836%3A56927&show-protocol-sidebar=1>]

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.

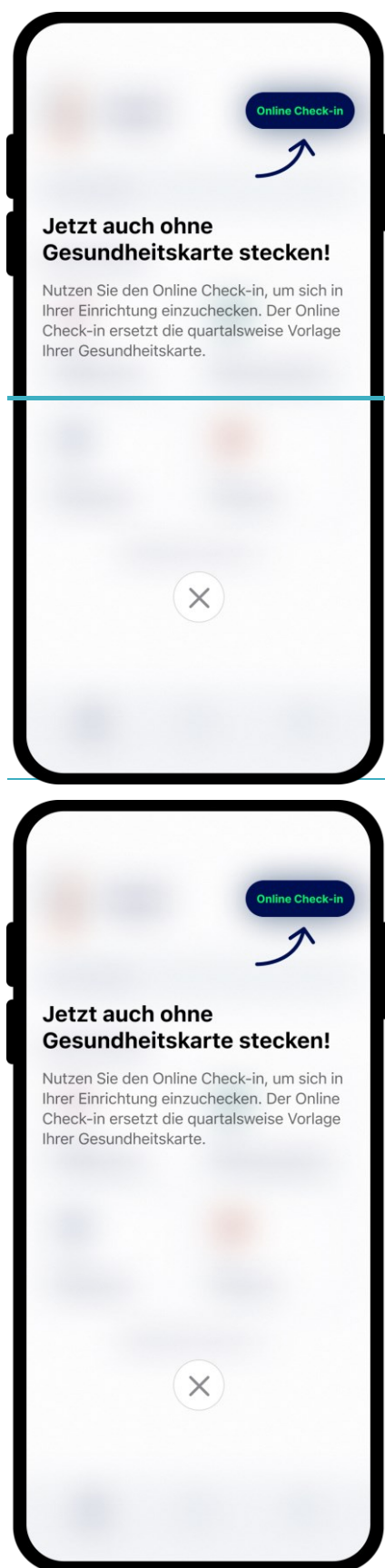


Abbildung 8: Screenshot der auf die neue Online -Check-in-Funktion hinweist.

Flow-Schritte und Vorgaben

~~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 ~~DrittanbieterAnbieter-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 ~~KrankenversicherungsKassen-App~~ ohne zusätzlichen Account und ohne nicht erforderliche Einrichtungsschritte möglich sein. Davon ausgenommen sind gesetzlich notwendige Schritte, beispielsweise die Zustimmung zu Nutzungsbedingungen.

7.4 Terminologie

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

Die Texte dienen als Empfehlung.

Thema	Text
Button für PoPP-Modul	Online -Check-in
Einwilligung [Auth-Modul] zur Übertragung KVRN, IK	Online -Check-in nutzen Wenn Sie den Online -Check-in nutzen möchten, werden folgende Daten an den Dienst der gematik GmbH übermittelt: • Ihre Versichertennummer • Name Ihrer Versicherung [Ablehnen] [Erlauben]

Thema	Text
Abfrage Einrichtung GesundheitsID nach eGK-Authentisierung	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]
Text bei fehlerhaften QR-Code	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]
Kein Internet bei Buttonclick: Online – Check-in	Kein Internet Wir konnten Sie leider nicht anmelden, da Sie keine Internetverbindung haben. [Abbrechen] [Erneut probieren]
Abgelehnte Kamerazugriffs-Berechtigung	Kamerazugriff benötigt Bitte erlauben Sie [der xxx App] den Kamerazugriff in den Einstellungen. [Abbrechen] [Einstellungen öffnen]
Abgelehnte App2App-Berechtigung (bei Aufruf aus andere App)	Öffnen anderer Apps erlauben Bitte erlauben Sie [der xxx App] andere Apps zu öffnen in den Einstellungen. [Abbrechen] [Einstellungen öffnen]
Erfolgreicher Check-in	Online –Check-in erfolgreich Sie haben sich erfolgreich bei [xxx] eingchecked. [Zur Startseite]
LEI Primärsystem des Inhabers der Telematik-ID noch nicht erreichbar	Online Check-in ausstehend Die Einrichtung, in der Sie sich einchecken möchten, ist zurzeit geschlossen. Bearbeitung Der Check-in wurde übermittelt und wird automatisch fortgesetzt, sobald die Einrichtung wieder geöffnet hat. Sie derzeit bearbeitet. Sie werden dazu benachrichtigt, sofern Siesobald der Check-in in der Einrichtung erfolgreich abgeschlossen wurde, vorausgesetzt, Push-Nachrichten erlaubt haben. Benachrichtigungen sind aktiviert. [Ok]

Thema	Text
LEI-Primärsystem des Inhabers der Telematik-ID nach 120h immer noch nicht erreichbar	Online 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]

8-Test

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

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.

98 Anhang A – Verzeichnisse

9.18.1 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 oder dessen Vertreter
VSDM	Versichertenstammdatenmanagement

9-28.2 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-ServiceResource Server wird mit dem Access-Token durch das ZETA Client ergänzt.
Anbieter-App	Der Begriff Anbieter-App verallgemeinert Apps, die kein PoPP-Modul implementieren. Das können andere Apps der Krankenkassen oder Drittanbieter-Anwendungen für VER sein.
Drittanbieter-AppAnwendung	Eine Drittanbieter-Apps bieten den Versicherten Anwendung ist eine Anwendung im Kontext des Gesundheitswesens, die Personen digitale Service zur Unterstützung medizinischer Anwendungsfälle an. Beispiele für Gesundheitsdienste bereitstellt und nicht von einer gesetzlichen oder privaten Krankenversicherung angeboten wird. Drittanbieter-Apps Anwendung können insbesondere von Leistungserbringern, deren Verbänden oder sonstigen Anbietern für Leistungen im Gesundheitswesen bereitgestellt werden. Beispiele sind Apotheken-Apps, Videosprechstunden-Apps oder DiGA-Apps. Die Apps von Krankenversicherungen aka Krankenversicherungskassen-Apps sind von keine Anbieter-Apps im Sinne dieser Spezifikation. Drittanbieter-Apps verschieden. Drittanbieter Apps können für beliebige Smartphone-Betriebssysteme wie Android oder iOS sowie für unterschiedliche Desktop-Betriebssysteme wie Windows oder Linux verfügbar Anwendungen können als native mobile Anwendungen oder browserbasierte Webanwendungen umgesetzt 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.

Begriff	Erläuterung
GesundheitsID	Die GesundheitsID ist die digitale Identität im Gesundheitswesen für Versicherte VER, 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.
KrankenversicherungsKassen- App	Mit einer Krankenversicherungs-App bieten Krankenversicherungen ihren Versicherten digitale Service zur Unterstützung medizinischer Anwendungsfälle an. Eine Kassen-App ist eine App, die von einer Krankenversicherung oder einem Dienstleister einer Krankenversicherung für VER bereitgestellt wird. Dazu zählen Apps wie das ePA-FdV, die Kassen-App mit Authenticator-Modul für die GesundheitsID sowie weitere Apps für digitale Serviceleistungen von Krankenversicherungen.
PoPP-Client	Eine Komponente im Primärsystem, die für die sichere Kommunikation zum PoPP- Service ServiceResource Server verantwortlich ist.
PoPP-Modul	Eine Komponente von Krankenversicherungs-App oder DrittanbieterKassen- App, welche für Online-Anwendungsfälle die Kommunikation mit dem PoPP- Service ServiceResource Server übernimmt. Das PoPP-Modul initiiert die Authentifizierung eines Versicherten einer VER 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 der VER, bzw. dessen elektronischer Gesundheitskarte (eGK), und dem Inhaber einer Leistungserbringerinstitution (LEI) -Telematik-ID.

Begriff	Erläuterung
Primärsystem	Primärsystem im Kontext der Spezifikation umfasst alle Systeme, die ein PoPP-Token vom PoPP-Service erhalten. Unter Primärsystem fallen demnach PVS, AVS und KIS sowie Systeme von Kostenträgern.
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)	Der Versorgungskontext beschreibt die sichere und kryptografisch belegte Verbindung zwischen einemeiner berechtigten VersichertenVER und einer authentifizierten Leistungserbringerinstitution. Diese Verbindung autorisiert den Zugriff auf anwendungsbezogene Versicherungsdaten über die Telematikinfrastruktur (TI) Anwendungen Ein Versorgungskontext besteht, wenn ein Leistungserbringer und ein Versichertereine VER 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. Ein Versorgungskontext entsteht durch die erfolgreiche Authentifizierung des Versichertender VER mittels digitaler Identität oder durch die erfolgreiche Authentifizierung seiner eGK in Fernversorgung, und ist auch bei telemedizinischen Anwendungen relevant
Versicherter VER (im Kontext PoPP)	Im Kontext von PoPP "Versicherter" "VER" eine Person, die ihre Identität im Gesundheitswesen einbringt, um einen Versorgungskontext mit dem Inhaber einer Leistungserbringerinstitution (LEI)Telematik-ID 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 VersicherteDie VER trägt zur Erstellung des Versorgungskontexts bei, indem erseinesie ihre Krankenversichertennummer (KVNR) bereitstellt und sich entweder über die GesundheitsID authentisiert, persönlich oder durch einen Vertreter mit der eGK beim Inhaber der LEITelematik-ID vorstellt, oder mobil per eGK bei beim Inhaber der LEITelematik-ID anmeldet.

Begriff	Erläuterung
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.

Offener Punkt: OP-PoPP-6 (Bezeichnung von Kassen-Systemen als Primärsysteme)

Die Erläuterung zum Eintrag Primärsystem bezeichnet auch die Systeme der Kassen, die einen PoPP-Client zum PoPP-Token-Abruf implementiert haben, als Primärsysteme. Konkret: "Primärsystem- Primärsystem im Kontext der Spezifikation umfasst alle Systeme, die ein PoPP-Token vom PoPP-Service erhalten. Unter Primärsystem fallen demnach PVS, AVS und KIS sowie Systeme von Kostenträgern."

Vorgehen:

Wenn es im Rahmen der Kommentierung keinen Widerspruch -von Kostenträgern- dazu gibt, würde der offene Punkt geschlossen werden und die aktuell gewählte Formulierung verwendet.

9.38.3 Abbildungsverzeichnis

Abbildung 1: Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in App der Krankenversicherung	11
Abbildung 2: Schnittstellen vom und zum PoPP-Modul.....	25
Abbildung 3: Verteilungssicht Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in Drittanbieter App.....	37
Abbildung 4: Bausteinsicht Informationsmodell Gerät des Versicherten und PoPP-Service für Online-Anwendungsfälle zur PoPP-Token-Erstellung.....	64
Abbildung 5: Screenshot des Flow-Schritt Autorisierung mit eGK ohne PIN.....	70
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.....	73
Abbildung 7: Screenshot der auf die neue Online-Check-in-Funktion hinweist.....	79
Abbildung 8: Laufzeitsicht Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID.....	96
Abbildung 9: Laufzeitsicht Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service.....	97
Abbildung 10: Laufzeitsicht Initialisierung ZETA Client	98
Abbildung 11: Laufzeitsicht Ermittlung Suchkriterien, VZD-Datenabruf mit Suchkriterien und Einwilligung in die Datenweitergabe.....	100

Abbildung 12: Laufzeitsicht – Information zum Status der PoPP-Token-Generierung an das PoPP-Modul.....	101
Abbildung 13: Laufzeitsicht – Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID	106
Abbildung 14: Laufzeitsicht – Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service	126
Abbildung 15: Detaillierter Ablauf der PoPP-Token-Generierung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service	178
Abbildung 1: Verteilungssicht der beteiligten Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in App der Krankenversicherung und nutzender Anbieter-App auf dem Smartphone der VER ...	11
Abbildung 2: Schnittstellen vom und zum PoPP-Modul.....	25
Abbildung 3 : Ablauf "Check-in" aus einer Anbieter-App	58
Abbildung 4 : Abbildung : Ablauf "Check-in" aus einer Anbieter-WebApp	59
Abbildung 5 : Abbildung : Ablauf "Check-in" mit 2 Geräten	60
Abbildung 6: Bausteinsicht - Informationsmodell Smartphone der VER und PoPP-Service Resource Server für Online-Anwendungsfälle zur PoPP-Token-Erstellung	64
Abbildung 7: Screenshot des Flow-Schritts Erfassung der Telematik-ID mit VZD-Suche, Anzeige einer Historie der letzten Online Check-ins, Favoriten und Detailscreens einer Einrichtung.	73
Abbildung 8: Screenshot der auf die neue Online Check-in-Funktion hinweist.....	79
Abbildung 9: Laufzeitsicht - Überblick Auslösung und Ablauf der PoPP-Token-Generierung	96
Abbildung 10: Laufzeitsicht - Initialisierung ZETA Client	98
Abbildung 11: Laufzeitsicht - Ermittlung Suchkriterien, VZD-Datenabruf mit Suchkriterien und Einwilligung in die Datenweitergabe.....	100
Abbildung 12: Laufzeitsicht - Ablauf der PoPP-Token-Generierung aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID	103
Abbildung 13: Laufzeitsicht - Detaillierter Ablauf des Online Check-in aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID.....	106
Abbildung 14: Laufzeitsicht - Ablauf Einlösen von PoPP-Token aus dem Online Check-in durch ein Primärsystem.....	176
Abbildung 15: Detaillierter Ablauf zum PoPP-Token-Abruf aus einem Primärsystem ...	178
Abbildung 16: Laufzeitsicht - Ablauf der Statusabfrage zur PoPP-Token-Generierung aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID	197
Abbildung 17: Laufzeitsicht - Detaillierter Ablauf zu Ermittlung des Status zum Online Check-in aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID	199

9.48.4 Tabellenverzeichnis

Tabelle 1: Beschreibung der wesentlichen Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung	12
Tabelle 2: Tab_PoPP_Modul_Payload_stat_QRCode]: Payload QR-Code	45
Tabelle 3: Flow-Schritte und Vorgaben an den Online-Check-in im Happy-Path	66
Tabelle 4: Flow-Schritte und Vorgaben an den Online-Check-in mit der Variante Autorisierung mittels eGK	70
Tabelle 5: Filterkriterien für Suche im VZD – Einträgen auswählen/suchen	73
Tabelle 6: Filterkriterien für Suche im VZD – Einträge eingrenzen	74
Tabelle 7: Filterkriterien für Suche im VZD – Einträge darstellen	75
Tabelle 8: Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung des Versicherten über seine GesundheitsID	107
Tabelle 9: Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App und Authentifizierung der eGK über den PoPP-Service	126
Tabelle 10: Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service	179
Tabelle 1: Beschreibung der wesentlichen Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung	12
Tabelle 2: Tab_PoPP_Modul_Payload_stat_QRCode]: Payload QR-Code	50
Tabelle 3: Flow-Schritte und Vorgaben an den Online-Check-in im Referenzpfad	66
Tabelle 4: Filterkriterien für Suche im VZD - Einträgen auswählen/suchen	73
Tabelle 5: Filterkriterien für Suche im VZD - Einträge eingrenzen	74
Tabelle 6: Filterkriterien für Suche im VZD - Einträge darstellen	75
Tabelle 7: Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung der VER über ihre GesundheitsID	126
Tabelle 8: Beschreibung der Schritte zur Generierung und zum Abruf von PoPP-Token aus dem Online-Check-in	179
Tabelle 9: Beschreibung der Schritte zum Abruf des Status eines zuvor durchgeführten Online-Check-in-Prozesses bei Authentifizierung der VER über ihre GesundheitsID	200

9.58.5 Referenzierte Dokumente

9.5.18.5.1 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)
[gemF_PushNotification]	gematik: Feature Spezifikation für PushNotification https://gemspec.gematik.de/prereleases/Draft_ePA_3_1_2/gemF_PushNotification_V1.0.0_CC/
[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/
[gemSpec_ZETA Stufe 2]	Aktuell ist die ZETA Spezifikation mit den Festlegungen für ZETA Stufe 2, insbesondere für die Authentisierung für Versicherte, noch nicht veröffentlicht. Vorab-Veröffentlichung vom 09.07.2026 https://gemspec.gematik.de/prereleases/Draft_ZETA_26_2/gemSpec_ZETA_V2.0.0_CC/
[ZETA-Client-SDK]	https://github.com/gematik/zeta-sdk
[API ZETA Client]	https://github.com/gematik/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_Modul_mobile_CheckIn_HID.yaml]	OpenAPI Schnittstellenspezifikation für Authentifizierung mit GesundheitsID und Anlegen eines Aufruf des PoPP-Datensatzes https://github.com/gematik/api-popp/blob/US-2-CC1/src/openapi/I_PoPP_CheckIn_HID.yaml Modul aus Anbieter-Apps todo: Anpassung github-Link
{I[_PoPP_Service_mobile_CheckIn_eGK_Verification.yaml]}	OpenAPI Schnittstellenspezifikation für Authentifizierung einer eGK und - den Aufruf des VZD-Suche am PoPP-ServiceResource Server - den Aufruf zum Anlegen eines PoPP-Datensatzes https://github.com/gematik/api-popp/blob/US-2-CC1/src/openapi/I_PoPP_CheckIn_eGK_Verification.yaml am PoPP-ServiceResource Server - den Aufruf zur Statusabfrage am PoPP-ServiceResource Server todo: Anpassung github-Link
{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_Service_mobile_Token_Generation_online_check_in.yaml]	OpenAPI Schnittstellenspezifikation zur Übertragung von PoPP-Token an den PoPP-Client, wenn die Übertragung durch den PoPP-ServiceResource Server 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 todo: Anpassung github-Link

[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
[I_AnbieterApp_mobile_CheckIn.yaml]	OpenAPI Schnittstellenspezifikation für den callback Aufruf vom PoPP-Modul an die Anbieter-App. todo: Anpassung github-Link

9-5-28.5.2 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
[BITV 2.0]	Barrierefreie Informationstechnik-Verordnung Bundesministerium der Justiz. (2023). Barrierefreie-Informationstechnik-Verordnung (BITV 2.0). https://www.gesetze-im-internet.de/bitv_2_0/

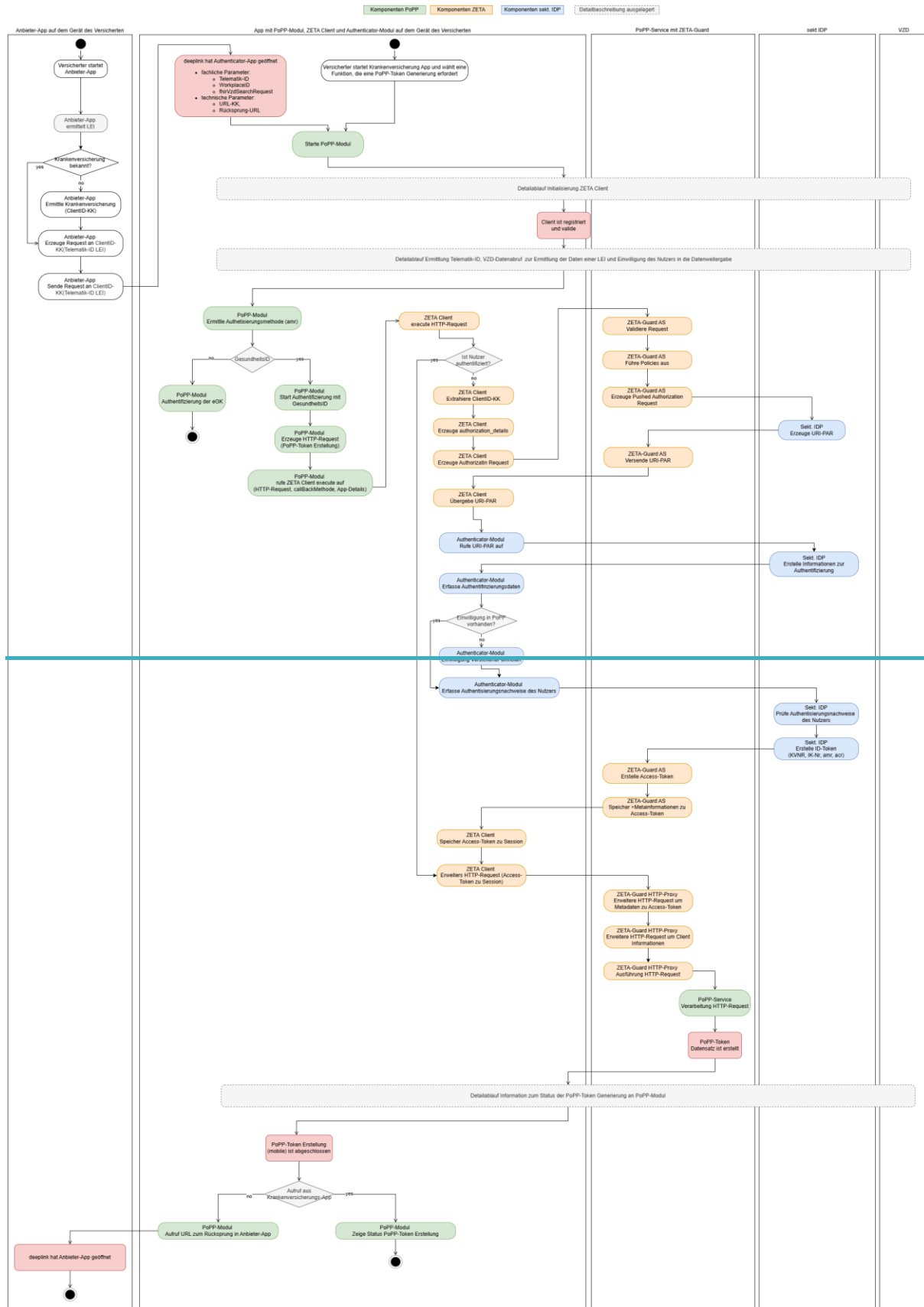
[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[DIN EN 301549]	DIN EN 301 549:2021-08 Barrierefreiheitsanforderungen für IKT-Produkte und - Dienstleistungen (deutsche Fassung EN 301 549:2021). Berlin: Beuth Verlag, 2021.

109 Anhang B - Ablaufbeschreibungen

10.1 Ablaufdiagramme

~~10.29.1 Allgemeiner Gesamtüberblick Auslösung und Ablauf der PoPP-Token-Generierung durch Authentifizierung Versicherter über ihre GesundheitsID~~

Spezifikation PoPP (Proof of Patient Presence) - Modul



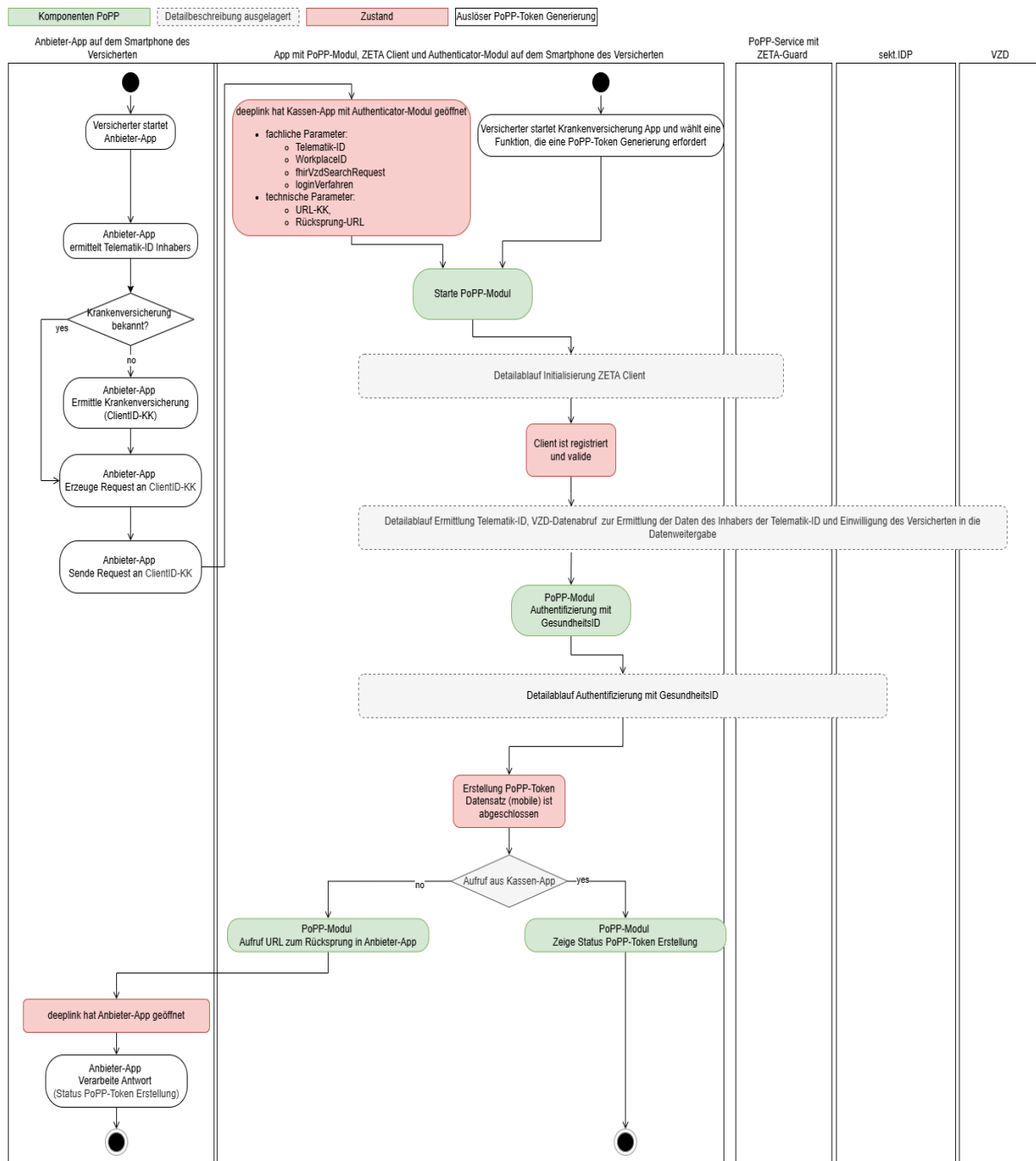


Abbildung 9: Laufzeitsicht - Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID

Allgemeiner Überblick Auslösung und Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service

Das nachfolgende Aktivitätsdiagramm stellt den allgemeinen Ablauf mit den beteiligten Komponenten dar.

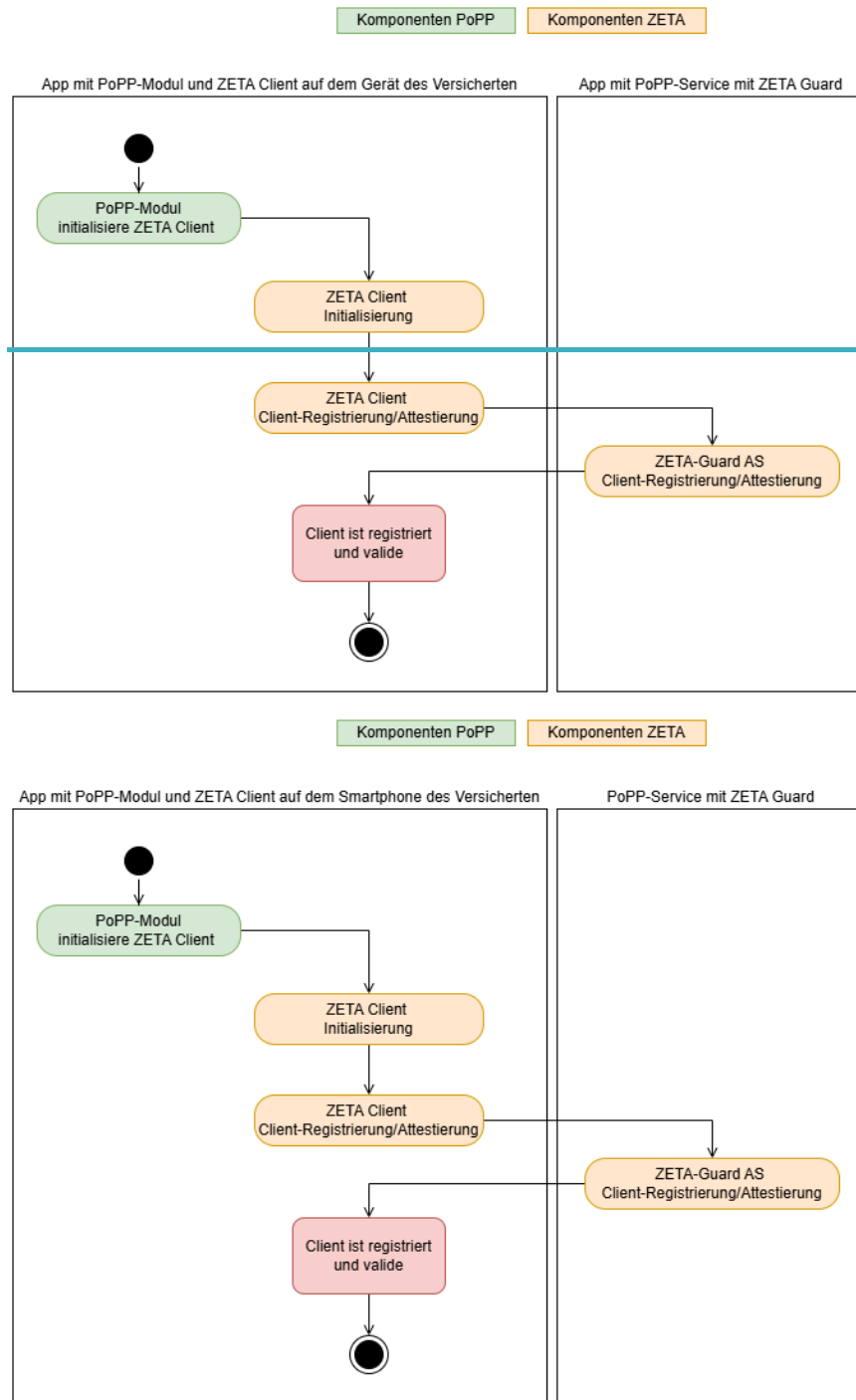


Abbildung 10: Laufzeitsicht - Initialisierung ZETA Client

10.49.3 Detailablauf "FHIR-VZD-Anfrage"

Die Abfrage des FHIR-VZD nach Daten zu [LEIsInhabern vonTelematik-IDs](#) muss die Anwendungsfälle unterstützen:

- Die Telematik-ID ist bekannt und es müssen für die Einwilligung ~~des Nutzers~~der VER weitere Daten ~~zur LEI~~zum Inhaber der Telematik-ID ermittelt werden.
- Die Telematik-ID ist nicht bekannt und es muss mit einer Suche auf Basis von bekannten Suchparametern (Name ~~der LEI~~, Adresse/PLZ ~~des Inhabers~~ der ~~LEI~~Telematik-ID, u.ä.) durchgeführt werden. Als Ergebnis werden alle Daten ~~des Inhabers der Telematik-ID inklusive~~ der ~~LEI~~Telematik-ID selbst für die Einwilligung ~~des Nutzers und die Telematik-ID~~ der ~~LEI~~VER 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- ~~Resource Server~~ . Das PoPP-Modul erstellt den eigentlichen FHIR-VZD Search Request auf Basis der im PoPP-Modul ~~vorhanden~~vorhandenen Daten ~~zur LEI~~zum Inhaber der Telematik-ID. Der FHIR-VZD Search Request wird über die ZETA-~~Komponenten~~Komponenten an den PoPP-Service ~~Resource Server~~ 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 ~~Resource Server~~ 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.

Das PoPP-Modul filtert das Suchergebnis und zeigt ~~dem Nutzer~~der VER den Datensatz zur Erteilung der Einwilligung in die Datennutzung an.

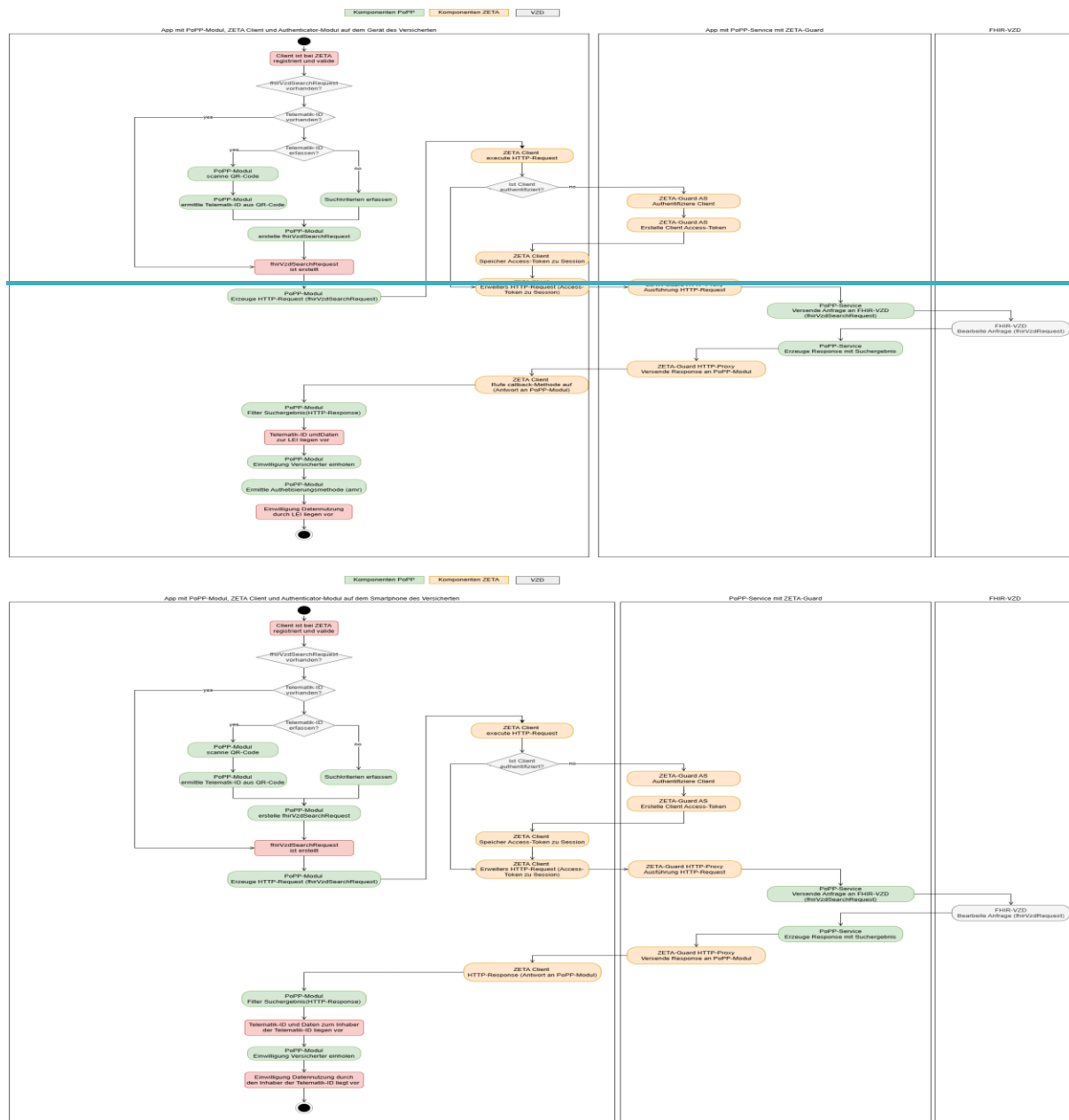


Abbildung 11: Laufzeitsicht - Ermittlung Suchkriterien, VZD-Datenabruf mit Suchkriterien und Einwilligung in die Datenweitergabe

10.59.4 Detailablauf "Information zum Status der PoPP-Token-Generierung an das PoPP-ModulAuthentifizierung mit GesundheitsID"

Nach Abschluss wird der Status der PoPP-Token-Generierung an das PoPP-Modul übermittelt. Der Status wird in der Response zum ausgeführten Request über den ZETA Guard HTTP-Proxy an den ZETA-Client zurückgegeben. Der ZETA-Client ruft daraufhin die vorgesehene callback-Methode am PoPP-Modul mit dem Response-Objekt als Parameter auf.

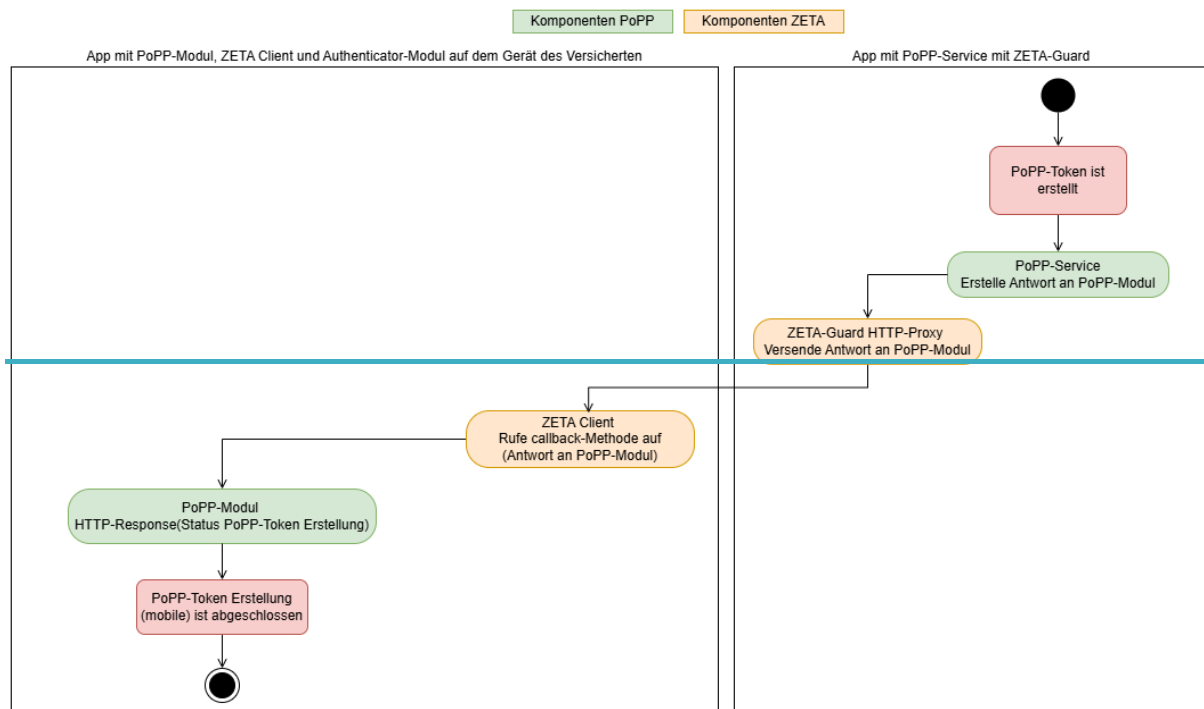


Abbildung 12: Laufzeitsicht – Information zum Status der PoPP-Token-Generierung an das PoPP-Modul

10.6 Flowdiagramme

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

Spezifikation PoPP (Proof of Patient Presence) - Modul

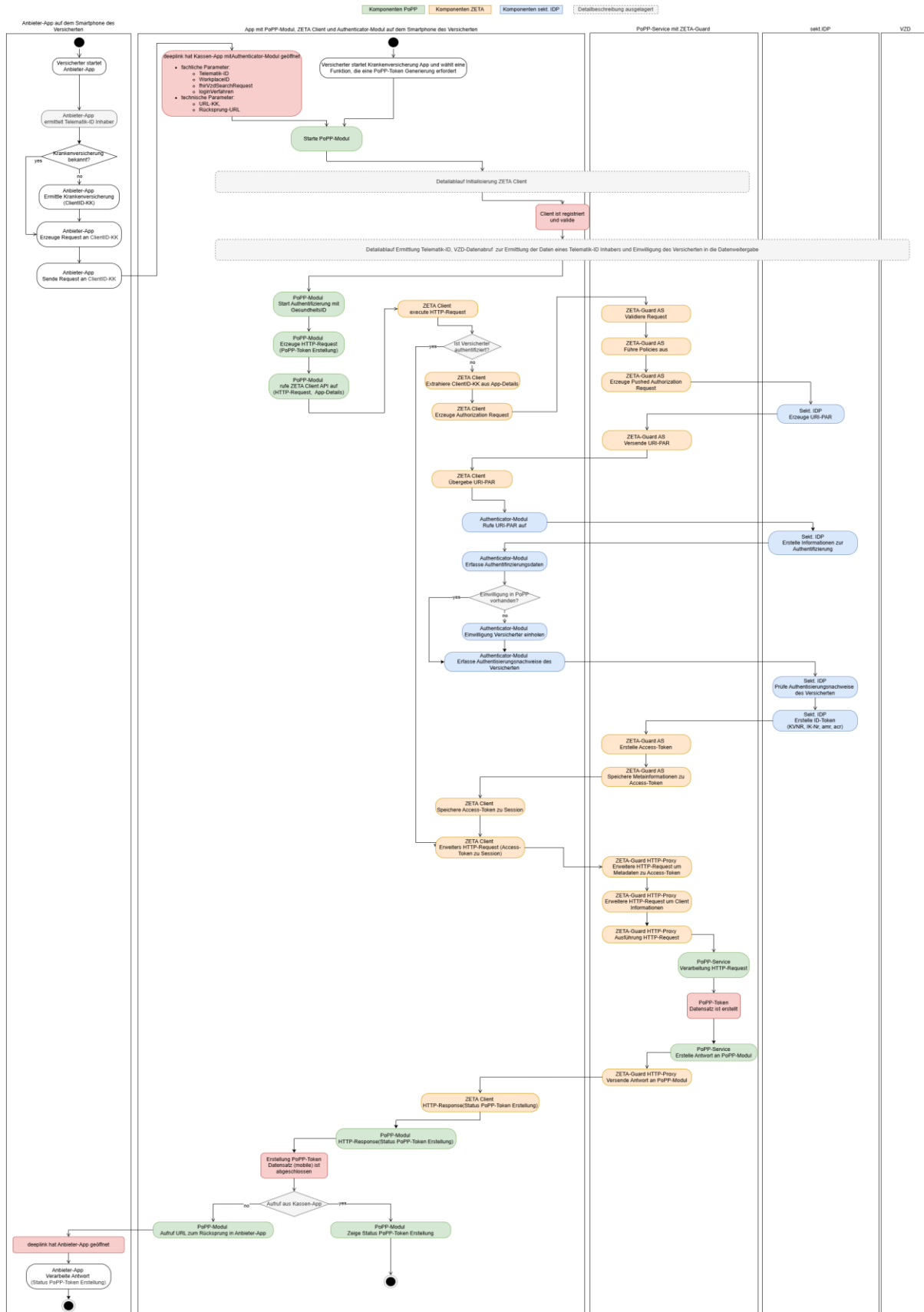
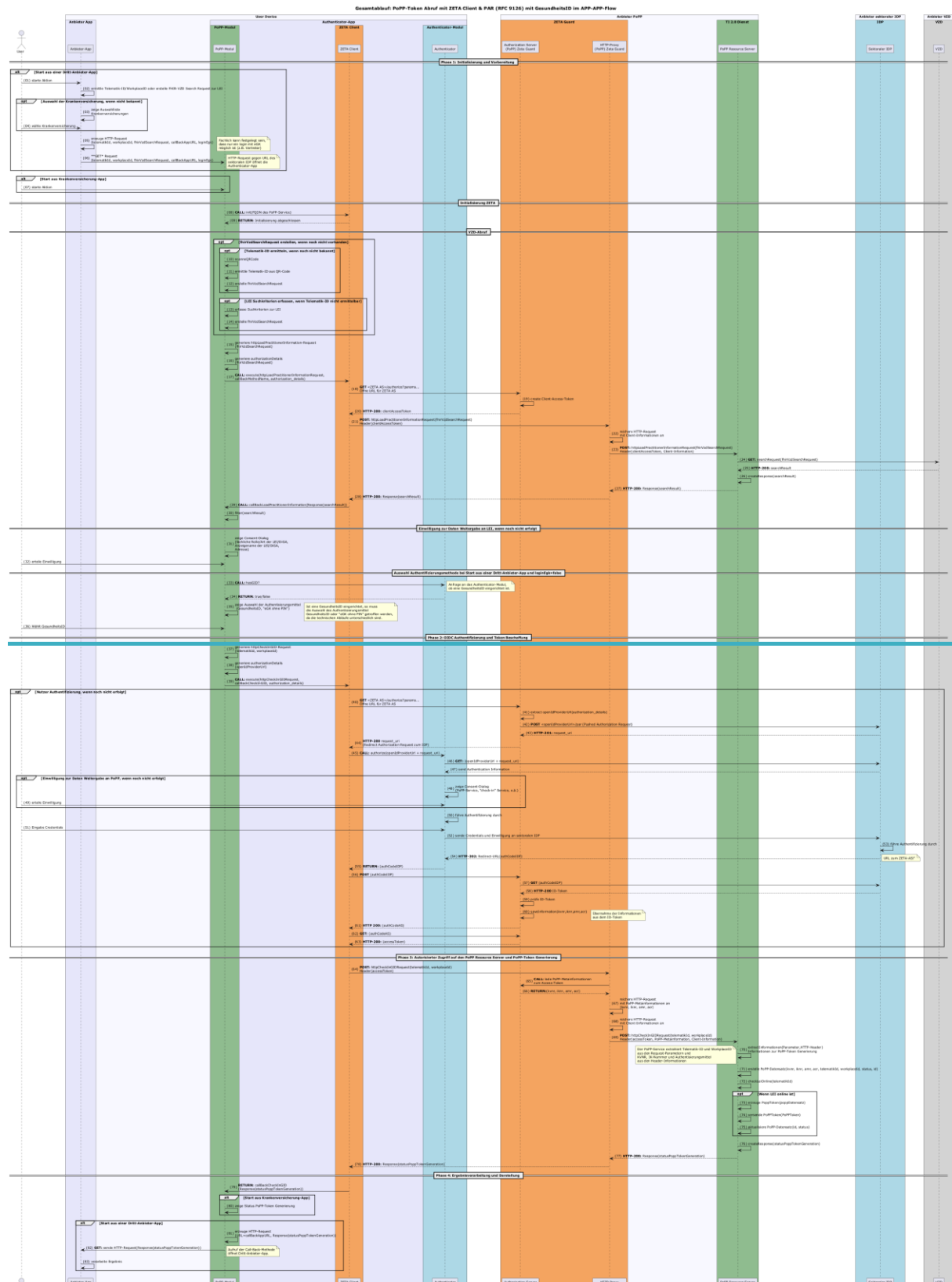


Abbildung 12: Laufzeitsicht - Ablauf der PoPP-Token-Generierung aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID

9.4.2 Detaillierter Ablauf des Online Check-in durch Authentifizierung VER über ihre GesundheitsID

Das Sequenzdiagramm "Detaillierter Ablauf ~~der PoPP-Token-Generierung~~ des Online Check-in aus einer Anbieter-App und Authentifizierung über die GesundheitsID" stellt den vollständigen Ablauf ~~der PoPP-Token-Generierung~~ für den Fall dar, dass ein Versicherter eine VER über seine GesundheitsID authentifiziert wird. Als Ergebnis des Ablaufs liegt im PoPP-ServiceResource Server ein PoPP-Datensatz und ein dazu gehörender Nachrichten-Datensatz. Die Datensätze dienen der späteren Ausstellung und Auslieferung des PoPP-Token.

Spezifikation PoPP (Proof of Patient Presence) - Modul



Der Status der PoPP-Token-Generierung wird dem PoPP-Modul zum Abschluss des Ablaufs vom PoPP-Service Resource Server zugesendet.

Die Schnittstellen sind den OpenAPI spezifiziert:

- [I_PoPP_Modul_mobile_CheckIn.yaml] - für den Aufrufs des PoPP-Modul aus einer Anbieter-App
- [I_PoPP_Service_mobile_CheckIn.yaml] -für die Aufrufe am PoPP-Service Resource Server vom PoPP-Modul
- [I_AnbieterApp_mobile_CheckIn.yaml] - für den callback-Aufruf der Anbieter-App aus demPoPP-Modul

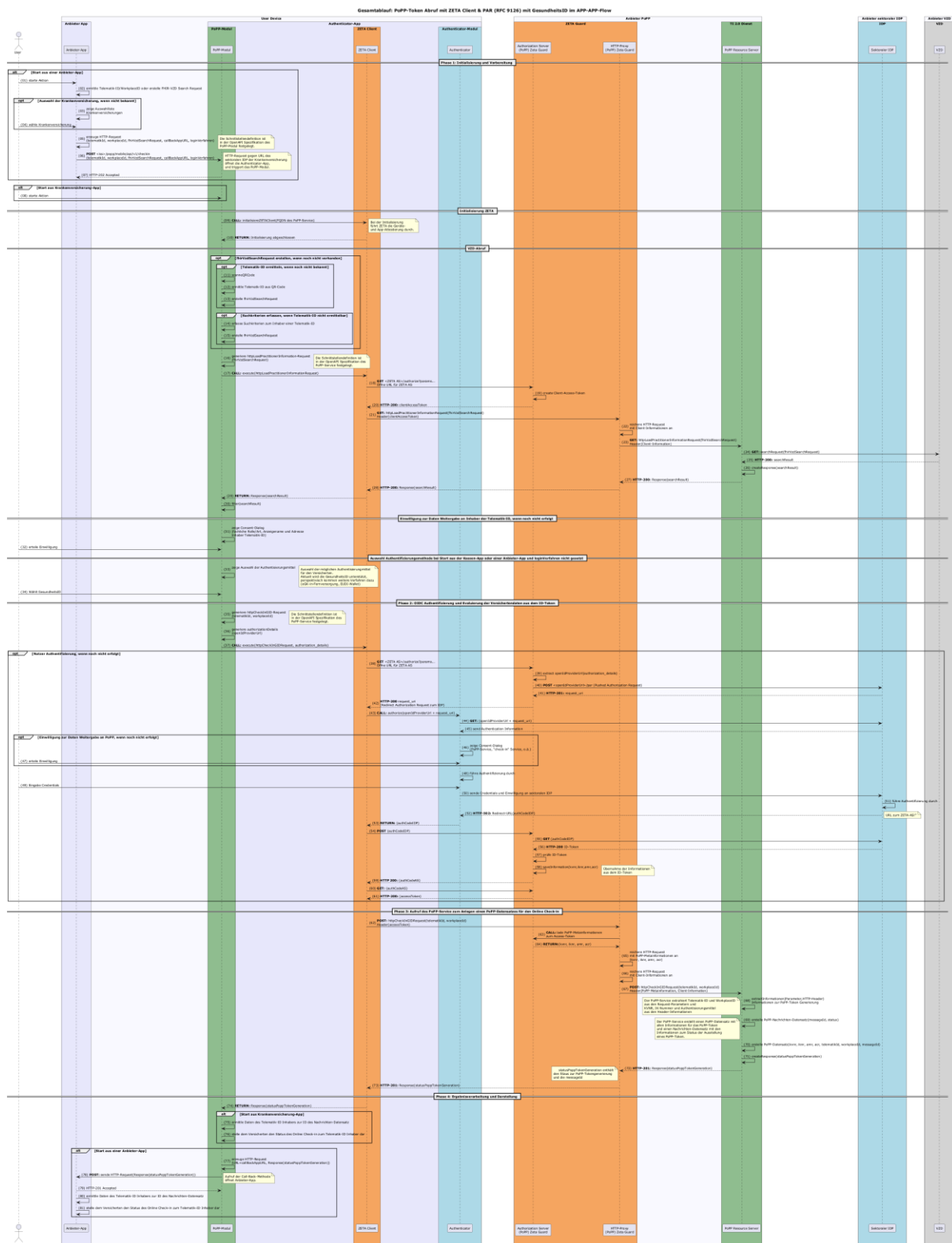


Abbildung 13: Laufzeitsicht - Detaillierter Ablauf der PoPP-Token-Generierung des Online Check-in aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID

~~10.6.2 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung Versicherter mit ihrer GesundheitsID~~

~~9.4.3 Die Tabelle [Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung des Versicherten-VER mit ihrer GesundheitsID~~

Die Tabelle "Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung der VER über seine ihre 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	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.: • 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

	Schritt	Beschreibung
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.

	Schritt	Beschreibung
5	erzeuge HTTP-GET-Request(<i>telematikId</i> , <i>workplaceId</i> , <i>fhirVzdSearchRequest</i> , <i>callbackAppURL</i> , <i>loginEgk</i>)	Die Anbieter-App erzeugt einen HTTP-GET-Request und übergibt als Parameter • <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(<i>telematikId</i> , <i>workplaceId</i> , <i>fhirVzdSearchRequest</i> , <i>callbackAppURL</i> , <i>loginEgk</i>)	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		

Schritt		Beschreibung
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		
± 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.
± ±	ermittle Telematik-ID aus QR-Code	Der QR-Code wird geprüft und die Telematik-ID sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.

Schritt		Beschreibung
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 httpLoadPractitionerInformationRequest(fhirVzdSearchRequest)	Das PoPP Modul generiert einen HTTP Request (httpLoadPractitionerInformationRequest) mit dem erstellten FHIR VZD Request als Parameter.
1 6	generiere authorizationDetails(fhirVzdSearchRequest)	Das PoPP Modul generiert Authorization_Details, diese enthalten den FHIR VZD Search Request zur LEI.

	Schritt	Beschreibung
1 7	CALL: execute(httpLoadPractitionerInformationRequest, callbackLoadPractitionerInformation, authorization_details)	Das PoPP-Modul ruft die execute-Methode am ZETA-Client mit den Parametern: • <i>httpLoadPractitionerInformationRequest</i> — erzeugter HTTP-Request • <i>callbackLoadPractitionerInformation</i> — callback-Methodenname der vom ZETA-Client aufzurufenden callback-Methode • <i>authorization_details</i> — Erzeugte Authorization Details auf.
1 8	GET <ZETA-AS>/authorize?params...	ZETA-Client sendet einen Authorization Request mit den Parametern des Authorization Code Flow und den <i>authorizationDetails{auth}</i> 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(fhirVsdSearchRequest) Header(clientAccessToken)	Der ZETA-Client ruft am ZETA-Guard HTTP-Proxy den ursprünglichen <i>httpLoadPractitionerInformationRequest</i> mit dem Client-Access-Token auf.
2 2	reichere HTTP-Request mit Client-Informationen an	Der ZETA-Guard HTTP-Proxy reicht den PoPP-Modul <i>httpLoadPractitionerInformationRequest</i> um Informationen zum Client (ZETA-Client) aus der ZETA-Guard-Client-Registry an.

	Schritt	Beschreibung
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-Anzeigergebnis(searchResult)	Das PoPP-Modul filtert das Anzeigergebnis, 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 KVN- und IK-Nummer für die LEI.

Schritt		Beschreibung
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.

	Schritt	Beschreibung
3 8	generiere <code>authorizationDetails(openIdProviderUrl)</code>	Das PoPP-Modul generiert <code>authorizationDetails</code> als JSON-Objekt welches die Client-ID des sekt. IDP (<code>openIdProviderUrl</code>) enthält: <pre>{ "app_details": { "auth_preferences": { "openIdProviderUrl": "<Client-ID des sekt. IDP>" } }}</pre>
3 9	CALL: <code>execute(httpCheckInGIDRequest, callBackCheckInGID, authorizationDetails)</code>	Das PoPP-Modul ruft am ZETA Client die Schnittstelle <code>execute</code> [<code>gemSpec_Zeta</code>] auf. Parameter sind: • <code>httpCheckInGIDRequest</code> vom PoPP-Modul erzeugter HTTP-CheckInGID-Request mit den Parametern Telematik-ID und WorkplaceID, • <code>callBackCheckInGID</code> Methode am PoPP-Modul welche der ZETA Client nach Abschluss der Bearbeitung aufruft, • <code>authorizationDetails</code> vom PoPP-Modul erzeugte <code>authorizationDetails</code> mit Informationen zur Versicherten-Authentifizierung.
4 0	GET <ZETA-AS>/authorize?params... \n Öffne URL für ZETA-AS	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.
4 1	<code>extract openIdProviderUrl(authorization_details)</code>	Da die Authentisierungsmethode "GesundheitsID" ist, extrahiert der ZETA-Guard-Authorization-Server aus den Authorization-Details die <code>openIdProviderUrl</code> des sektoralen IDP der gewählten Krankenversicherung.

	Schritt	Beschreibung
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 KVN durch den PoPP-Service. Verweigert der Versicherte die Einwilligung, wird der Prozess hier abgebrochen.

	Schritt	Beschreibung
50	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.
51	Eingabe Authentisierungsnachweise	Der Versicherte erfasst die Authentisierungsnachweise zur ausgewählten Authentifizierungsmethode.
52	Sende Authentisierungsnachweise und Einwilligung an IDP-Server	Das Authenticator-Modul sendet Einwilligung und Authentisierungsnachweis an den sekt. IDP.
53	führe Authentifizierung durch	Der sektorale IDP authentifiziert den Versicherten auf Basis der Authentisierungsnachweise und der ausgewählten Authentifizierungsmethode.
54	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.
55	RETURN: authCodeIDP	Das Authenticator-Modul antwortet dem Aufruf des ZETA-Client und übergibt den authCodeIDP.
56	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.
57	GET (authCodeIDP)	ZETA-AS ruft den ID-Token durch Übergabe des Authorization-Code (authCodeIDP) am Token-Endpoint des sektoralen IDP ab.

	Schritt	Beschreibung
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: • <code>urn:telematik:claims:id</code> mit dem Wert der KVNR des Versicherten, • <code>urn:telematik:claims:organization</code> mit dem Wert der IK-Nummer der Krankenversicherung, • <code>amr</code> mit dem Wert der angewandten Authentisierungsmethode, • <code>acr</code> 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.

	Schritt	Beschreibung
6 4	POST: httpCheckInGIDRequest(accessToken)	Der ZETA-Client ruft am ZETA Guard HTTP-Proxy den ursprünglichen 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 reichert 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 reichert 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.

	Schritt	Beschreibung
7 0	extractInformationen(appDetail, HTTP-Header)	Der PoPP-Service extrahiert: • aus dem Request-Header • die PoPP-Metainformationen zum Versicherten (KVNR, IK-Nummer) • die Informationen zur Authentifizierung (amr, acr) • Informationen zum aufrufenden Client • aus den Request-Parametern • Telematik-ID, WorkplaceID der LEI
7 1	erstelle-PoPP-Datensatz(kvnr, iknr, amr, acr, telematikId, workplaceId, status, id)	Der PoPP-Service legt einen PoPP-Datensatz mit allen relevanten Informationen für die Generierung eines PoPP-Token an. Zusätzlich enthält der Datensatz • id – eindeutiger Identifier des Datensatzes, dient dem Auffinden zur Abfrage oder Aktualisierung des Status zur PoPP-Token-Generierung • status – enthält den Status der PoPP-Token-Generierung • 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

Schritt		Beschreibung
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 liefert den PoPP-Token
7 5	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		
7 6	createResponse(statusPoppTokenGeneration)	Der PoPP-Service erstellt die Antwort an das PoPP-Modul • <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

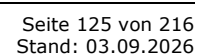
	Schritt	Beschreibung
7 7	HTTP 200: Response(statusPoPPTokenGeneration)	Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy mit HTTP-200 oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einem der Statuscodes: • <i>success</i> PoPP-Token wurde erzeugt 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 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.

	Schritt	Beschreibung
7 8	HTTP-200: Response(statusPoPPTokenGeneration)	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 einem der Statuscodes: • <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 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.
7 9	RETURN: callBackCheckInGID(Response(statusPoPPTokenGeneration))	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.
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.

Schritt		Beschreibung
Ende Alternative – Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App		
Start Alternative – Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App		
8 1	erzeuge HTTP-Request(URL=callbackAppURL, Response(statusPoPPTokenGeneration))	Das PoPP-Modul erzeugt einen HTTP-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(statusPoPPTokenGeneration))	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		

10.6.3 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.



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

~~10.6.4 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 7: Beschreibung der Schritte zur PoPP-Token-Erstellung aus einer Anbieter-App und bei Authentifizierung der eGK über den PoPP-Service VER über ihre GesundheitsID

	Schritt	Beschreibung
Start Alternative - Auslösen der Erzeugung eines PoPP-Tokens des Online Check-in aus einer Drittanbieter-Anbieter-App		
1	Aktion starten	Der Versicherte Die VER wählt eine Funktion in der App eines Anbieters (z.B. "Anmelden zum Arztbesuch", "Rezept einlösen").

	Schritt	Beschreibung
2	ermittle Telematik-ID/WorkplaceID oder erzeuge FHIR-VZD Search Request zur Datenermittlung zur LEI zum Inhaber einer Telematik-ID	In der Anbieter-App werden Informationen zur LEI zum Inhaber einer Telematik-ID ermittelt, für welche das PoPP-Token bestimmt ist. Dies ist Anbieter-App-spezifisch und kann auf unterschiedlichen Wegen erfolgen. Z.B. ist Bei der Anbieter-App einer LEI (online Apotheke) ist die Telematik-ID bekannt7. - Bei der Anbieter-App zur Vermittlung einer Videosprechstunde bietet die App entsprechende Auswahlfunktionen an7. - Eine Praxis kann die Telematik-ID als QR-Code bereitstellen. - Es werden Suchkriterien erfasst, mit denen eine FHIR-VZD – Anfrage durchgeführt werden kann

	Schritt	Beschreibung
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 Die VER wählt seine Krankenversicherung aus der Liste aus. Die Auswahl enthält auch die ClientID (OpenID-Provider URL, iss) des sektoralen IDP der Krankenversicherung.

5	erzeuge HTTP-GETPOST Request(<i>telematikId</i>, <i>workplaceId</i>, <i>fhirVzdSearchRequest</i>, <i>callbackAppURL</i>, loginEgkloginVerfahren)	Die Anbieter-App erzeugt einen HTTP-GET Request und übergibt als Parameter • <i>telematikId</i> - ist die Telematik-ID der LEIeines Telematik-ID Inhabers, für die ein PoPP-Token erstellt werden soll, • <i>workplaceId</i> (<i>optional</i>)-ist eine WorkplaceID (optional), wenn das PoPP-Token einem bestimmten Arbeitsplatz der LEIim Primärsystem des Inhabers der Telematik-ID zugeordnet werden soll, • <i>fhirVzdSearchRequest</i> (<i>optional</i>)-Search-Request für den Aufruf der FHIR-VZD API, um Daten zu einer LEIzum Inhaber der Telematik-ID zu erhalten • <i>callbackURL</i> (<i>Rücksprung-URL</i>)- welche das PoPP-Modul aufrufen
---	---	---

	Schritt	Beschreibung
		muss, wenn die Erzeugung eines 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. Werte bereich: "gID", wenn der Login durch die Authentifizierung mit GesundheitsID erfolgen muss. <i>Hinweise: Mit Einführung weiterer Verfahren wie eGK-in-Fernversorgung oder EUDI-Wallet wird der Wertebereich erweitert.</i>

	Schritt	Beschreibung
6	sende HTTP- GET POST Request(telematikId, workplaceId, fhirVzdSearchRequest, callBackAppURL, loginEgk loginVerfahren)	Die Anbieter-App sendet den HTTP- GET Request entsprechend der OpenAPI Spezifikation [I_PoPP_Modul_mobile_CheckIn.yaml] an die OpenID-Provider URL des sektoralen IDP der Krankenversicherung <iss>. 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 Kassen-App mit integriertem Authenticator-App-Modul und PoPP-Modul. Aufgrund der Parameter im Request des Requests wird das PoPP-Modul in der Authenticator Kassen-App aufgerufen.
7	sende HTTP-ACCEPTED	Ende Alternative – Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App Das PoPP-Modul bestätigt den Empfang mit einem HTTP-ACCEPTED.
Start Ende Alternative – - Auslösen der Erzeugung eines PoPP-Token des Online Check-in aus Krankenversicherung einer Anbieter-App		

Eingefü

Eingefü

	Schritt	Beschreibung
7	Aktion starten	Der Versicherte wählt eine Funktion in der Krankenversicherungs-App (z.B. "Praxis-Check-in")Start Alternative - Auslösen des Online Check-in aus einer Kassen-App
8	Aktion starten	Ende Alternative – Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung – AppDie VER wählt eine Funktion in der Kassen-App (z.B. "Praxis Check-in")
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.Ende Alternative - Auslösen des Online Check-in aus einer Kassen-App

Gelösch

Gelösch

Eingefü

Eingefü

Gelösch

Gelösch

	Schritt	Beschreibung
9	RETURN: Initialisierung abgeschlossen CALL: initialisiereZETAClient(FQDN des PoPP-Service Resource Server)	Nach Abschluss der Initialisierung kehrt die Anwendung in das PoPP-Modul zurück. Beim ersten Aufruf einer Session wird der ZETA Client initialisiert. Dazu wird der FQDN des PoPP-ServiceResource Server übergeben. Der FQDN des PoPP-Service Resource Server 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.
FHIR-VZD Search Request erzeugen, wenn nicht bereits vorhanden10	RETURN: Initialisierung abgeschlossen	Nach Abschluss der Initialisierung kehrt die Anwendung in das PoPP-Modul zurück.
Alternative: Telematik-ID ermittelnFHIR-VZD Search Request erzeugen, wenn nicht bereits vorhanden		
1 0	scanneQRCode	Alternative: Telematik-ID ermittelnIst 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.

Eingefü

Eingefü

Gelösch

Gelösch

	Schritt	Beschreibung
11	scanne QR-Code ermittle Telematik-ID aus QR-Code	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. Der QR-Code wird geprüft und die Telematik-ID sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.
12	ermittle Telematik-ID aus QR-Code FHIR VZD Search Request erstellen	Es Der QR-Code wird ein Search-Request mit dergeprüft und die Telematik-ID für den Aufruf der FHIR-VZD-API erstellt, um Daten zu einer LEI zu erhalten sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.
Alternative: Suchkriterien erfassen ¹³	erstelle FHIR-VZD Search Request	Es wird ein Search-Request mit der Telematik-ID für den Aufruf der FHIR-VZD-API erstellt, um Daten zum Inhaber einer Telematik-ID zu erhalten.
¹ ³ erfasse Suchkriterien zur LEI	In diesem Schritt erfolgt die Erfassung der Suchkriterien zur LEI Alternative: Suchkriterien erfassen. Suchkriterien können z.B. bestehen aus bekannten Angaben zur LEI (Name, Fachrichtung, PLZ) und zusätzlichen Informationen (Umkreissuche).	

Eingefü

Eingefü

Gelösch

Gelösch

	Schritt	Beschreibung
14	erstelle FHIR-VZD Search Requesterfasse Suchkriterien zum Inhaber einer Telematik-ID	In diesem Schritt erfolgt die Erfassung der Suchkriterien zum Inhaber einer Telematik-ID. Suchkriterien können z.B. Es wird ein Search-Request mit den Suchkriterien für den Aufruf der FHIR-VZD API erstellt, um Daten zu einer LEI zu erhalten bestehen aus bekannten Angaben zum Inhaber der Telematik-ID (Name, Fachrichtung, PLZ) und zusätzlichen Informationen (Umkreissuche).
15	erstelle FHIR-VZD Search Request	Der FHIR-VZD-Abruf wird über den ZETA-Ablauf an den PoPP-Service delegiert Es wird ein Search-Request mit den Suchkriterien für den Aufruf der FHIR-VZD API erstellt, um Daten zu einem Inhaber der Telematik-ID zu erhalten.
1 5	generiere httpLoadPractitionerInformationRequest(fhirVzdSearchRequest)	Das PoPP-Modul generiert einen HTTP-Request (httpLoadPractitionerInformationRequest) mit dem erstellten FHIR-VZD-Request als Parameter. Der FHIR-VZD-Abruf wird über den ZETA-Ablauf an den PoPP-ServiceResource Server delegiert

Eingefü

Eingefü

Gelösch

Gelösch

	Schritt	Beschreibung
16	generiere authorizationDetails <code>httpLoadPractitionerInformation-Request</code> (fhirVzdSearchRequest)	Das PoPP-Modul generiert Authorization_Details <code>Authorization_Details</code> , diese enthalten einen einen HTTP-Request (<code>httpLoadPractitionerInformationRequest</code>) mit dem erstellten FHIR-VZD-Search-Request zur LEI- <code>FHIR-VZD-Search-Request</code> als Parameter entsprechend der Schnittstellendefinition <code>[I_PoPP_Service_mobile_CheckIn.yaml]</code> .

	Schritt	Beschreibung
17	CALL: execute(httpLoadPractitionerInformationRequest, callbackLoadPractitionerInformation, authorization_details)	Das PoPP-Modul ruft die executeSchnittstelle n-Methode zur Ausführung des HTTP-Request am ZETA Client [gemSpec_ZETA] mit den Parametern • dem erzeugten HTTP-Request (httpLoadPractitionerInformationRequest —erzeugter HTTP-Request, • callbackLoadPractitionerInformation—callback-Methodenname der vom ZETA Client aufzurufenen callback-Methode, • authorization_details—Erzeugte Authorization Details) als Parameter auf.
18	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.

	Schritt	Beschreibung
19	create Client Access-Token	Auf Basis der Client-Registrierung wird vom ZETA Guard ein Client Access-Token erzeugt.
20	HTTP-200: clientAccessToken	Der ZETA Guard gibt das -antwortet dem ZETA-Client mit dem Access-Token- an ZETA-Client zurück.
21	POST GET: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken)	Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichen ursprünglichen httpLoadPractitionerInformationRequest mit dem Client Access-Token auf.
22	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.
23	POST GET: httpLoadPractitionerInformationRequest(fhirVzdSearchRequest) Header(clientAccessToken , Client-Information)	Der ZETA Guard HTTP-Proxy sendet den httpLoadPractitionerInformationRequest mit den erweiterten Client-Informationen an den PoPP-Service- Resource Server .
24	GET: searchRequest(fhirVzdSearchRequest)	Der PoPP-Service Resource Server führt den FHIR-VZD Search Request aus.

	Schritt	Beschreibung
25	HTTP-200: searchResult	Der FHIR-VZD liefert dem PoPP-Service Resource Server das SuchergebnisAnfrageergebnis zurück.
26	createResponse(searchResult)	Der PoPP-Service Resource Server erzeugt eine Response mit den ermittelten Daten zur LEIdem Anfrageergebnis.
27	HTTP-200: Response(searchResult)	Der PoPP-Service Resource Server antwortet dem ZETA Guard HTTP-Proxy mit der erstellten Response.
28	HTTP-200: Response(searchResult)	Der ZETA Guard HTTP-Proxy leitet die Response zum ZETA Client weiter.
29	CALL: callBackLoadPractitionerInformation(RETU RN: Response(searchResult))	Der ZETA Client ruft gibt als Antwort auf die callBack-Methode (callBackLoadPractitionerInformation) mit demAusführungsanweisung des httpLoadPractitionerInformationRequest das Response-Objekt als Parameter beim mit dem Anfrageergebnis an das PoPP-Modul aufzurück.

	Schritt	Beschreibung
30	Filter Anzeigeergebnis(searchResult)	Das PoPP-Modul filtert das Anzeigeergebnis, dass das nur für den Nutzer die VER relevante Daten zur Anzeige im Einwilligungsdialog kommen.
31	Zeige Consent-Dialog (Organisationsname, Organisationsanschrift)	Das PoPP-Modul zeigt dem Versicherten der VER einen Dialog mit den Detailinformationen zum Inhaber der LEI Telematik-ID Organisation (Name, Adresse, ggf. weiter Informationen) an und bittet um Zugriffserlaubnis auf KVNR und IK-Nummer für die LEI den Inhaber der Telematik-ID.
32	Erteilt Einwilligung	Der Versicherte Die VER erteilt seine ihre Einwilligung. Verweigert der Versicherte seine die VER ihre Einwilligung, so wird der Prozess abgebrochen.

	Schritt	Beschreibung
33	Zeige Auswahl der Authentisierungsmittel	Auswahl des Authentifizierungsverfahren— GesundheitsID oder eGK ohne PINDas PoPP-Modul zeigt der VER die Auswahl möglicher Authentisierungsmethoden an. Aktuell ist dies die GesundheitsID. Weitere mögliche Authentisierungsmethoden wären perspektivisch die eGK (in Fernversorgung) und die EUDI-Wallet.
3334	CALL: hasGID?Wählt GesundheitsID	Das PoPP-Modul ruft das Authenticator-Modul über dessen API auf um zu ermitteln, ob das Authenticator-Modul bereits an eine GesundheitsID gebunden ist. Die VER wählt "GesundheitsID" aus.
3 4	RETURN: true/false	Das Authenticator-Modul antwortet dem PoPP-Modul true/falseAblauf Erzeugung eines PoPP-Datensatzes

Eingefü

Eingefü

Gelösch

Gelösch

	Schritt	Beschreibung
35	Zeige Auswahl der Authentisierungsmittel (GesundheitsID, "eGK ohne PIN")generiere httpCheckInGID-Request(telematikId, workplaceId)	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 Authentisierungsmitteln "GesundheitsID" und "eGK" (ohne PIN) wählen kann. Diese Auswahl ist notwendig, da sich die jeweiligen Abläufe ab hier unterscheiden. Das PoPP-Modul generiert HTTP-CheckInGID-Request entsprechend der Schnittstellendefinition [I_PoPP_Service_mobile_CheckIn.yaml]. Telematik-ID und WorkplaceID sind Parameter des HTTP-CheckInGID-Requests.

	Schritt	Beschreibung
36	Wählt "eGK ohne PIN" generiere authorizationDetails(openIdProviderUrl)	Der Versicherte wählt in diesem Szenario das Authentisierungsmittel "eGK ohne PIN", also Authentifizierung einer eGK. Das PoPP-Modul generiert authorizationDetails als JSON-Objekt welches die Client-ID des sekt. IDP (openIdProviderURL) enthält. <pre>{ "app_details": { "auth_preferences" : { "openIdProviderUrl" : "<Client-ID des sekt. IDP>" } }</pre>

	Schritt	Beschreibung
37	CALL: execute(httpCheckInGIDRequest, authorizationDetails)	Ablauf-Erzeugung eines PoPP-Token Das PoPP-Modul ruft die Schnittstellen-Methode zur Ausführung des HTTP-Request am ZETA Client [gemSpec_ZETA] auf. Parameter sind: <i>httpCheckInGIDRequest</i> - vom PoPP-Modul erzeugter HTTP-CheckInGID-Request mit den Parametern Telematik-ID und WorkplaceID, <i>authorizationDetails</i> - vom PoPP-Modul erzeugte authorizationDetails mit Informationen zur Authentifizierung der VER.

Eingefü

Eingefü

	Schritt	Beschreibung
3738	generiere httpReadEgk-Request(telematikId, workplaceId)GET <ZETA AS>/authorize?params... Öffne URL für ZETA AS	Das PoPP-Modul generiert HTTP-Read-eGK-Request. Telematik-ID und WorkplaceID sind Parameter des HTTP-Read-eGK-Request.ZETA Client sendet einen Authorization Request mit den Parametern des Authorization Code Flow und den authorizationDetails{auth} an den ZETA Guard Authorization-Server.
3839	generiere authorizationDetails()extract openIdProviderUrl(authorization_details)	Die authorization_details enthalten für diesen Aufruf keine Werte. Da die Authentisierungsmethode "GesundheitsID" ist, extrahiert der ZETA Guard Authorization-Server aus den Authorization-Details die openIdProviderUrl des sektoralen IDP der gewählten Krankenversicherung.

	Schritt	Beschreibung
3940	<code>execute(httpReadEgkRequest, callbackReadEgk, authorizationDetails)POST <openIdProviderUrl>/par (Pushed Authorization Request)</code>	Das PoPP-Modul ruft amDer ZETA Client die Schnittstelle <code>execute</code> [gemSpec_Zeta] auf. Parameter sind: • <code>httpReadEgkRequest</code> vom PoPP-Modul erzeugter HTTP-Read-eGK-Guard Authorization-Server sendet einen Pushed Authorization Request mit den Parametern Telematik-ID und WorkplaceID, • <code>callbackReadEgk</code> Methode am PoPP-Modul welche PAR-Endpoint des sektoralen IDP der ZETA Client als Ergebnis des HTTP-Requests aufruft, <code>authorizationDetails</code> vom PoPP-Modul erzeugte <code>authorizationDetails</code>gewählten Krankenversicherung

	Schritt	Beschreibung
4041	<code>httpReadEgkRequest(telematikId, workplaceId)</code> <code>Header(clientAccessToken)HTTP-201: request_uri</code>	Der ZETA Client-sektoralen IDP der gewählten Krankenversicherung erzeugt eine <code>request_uri</code> (URI-PAR) und sendet den HTTP-Request (<code>httpReadEgkRequest</code>) mit dem Client Access-Token diese an den ZETA Guard HTTP-Proxy-Authorization-Server zurück.
4142	reichere HTTP-200: <code>request_uri</code> (Redirect Authorization Request mit Client-Informationen an zum IDP)	Der ZETA Guard HTTP-Proxy reicht den HTTP-Request um Informationen zum Authorization-Server antwortet dem ZETA Client aus der Client-Registrierung an mit einem Redirect auf die <code>request_uri</code> .
4243	<code>httpReadEgkRequest(telematikId, workplaceId)</code> <code>Header(clientAccessToken, Client-Information)CALL: authorize(openIdProviderUrl+request_uri)</code>	Der ZETA-Guard HTTP-Proxy sendet den <code>httpReadEgkRequest</code> mit den erweiterten Client-Informationen an den PoPP-Service. Der ZETA Client ruft eine Methode am Authenticator-Modul mit der <code>request_uri</code> als Parameter auf.
loop-Wiederholung, bis alle APDU-Commands bearbeitet sind 44	GET: <code>(openIdProviderUrl+request_uri)</code>	Die Kassen-App mit Authenticator-Modul führt den Authorization Request (<code>redirect_uri</code>) an den Auth-Endpunkt des sektoralen IDP aus.

Eingefügt

Eingefügt

	Schritt	Beschreibung
4345	<code>generateAPDUCommand()</code> send Authentication Information	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. Der sektorale IDP übermittelt dem Authenticator-Modul der Kassen-App die notwendigen Informationen zur Authentifizierung (z.B. Status Gerätebindung, Einwilligungen, Authentisierungsmittel).
4446	<code>createResponse(apduCommand)</code> Zeige Consent-Dialog (PoPP-Service, "Check-in" Service, o.ä.)	Der Wenn nicht erfolgt holt der Authenticator-Modul eine Consent-Freigabe für den PoPP-Service generiert eine Response mit dem APDU-Command ein.

	Schritt	Beschreibung
4547	HTTP-200-Response(apduCommand) Erteilt Einwilligung	Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy auf den Request mit HTTP-200 und der generierten Response. Die VER erteilt seine Einwilligung zur Nutzung der KVNR durch den PoPP-Service. Verweigert die VER die Einwilligung, wird der Prozess hier abgebrochen.
4648	HTTP-200-Response(apduCommand) Führe Authentifizierung durch	Der ZETA Guard HTTP-Proxy leitet die Antwort an den ZETA Client weiter. Das Authenticator-Modul führt die VER durch die Authentifizierung. Je nach Konfiguration kann die Auswahl der möglichen Authentisierungsmittel angezeigt werden oder direkt zur Erfassung der Authentisierungsnachweise der präferierten Authentifizierungsmethode.

	Schritt	Beschreibung
4749	callbackReadEgk(Response(apduCommand)) Eingabe Authentisierungsnachweise	Der ZETA-Client ruft die callback -Methode (callbackReadEgk) mit dem Response -Objekt am PoPP-Modul auf. Die VER erfasst die Authentisierungsnachweise zur ausgewählten Authentifizierungsmethode.
4850	sendeAPDUCommand(apduCommand) Sende Authentisierungsnachweise und Einwilligung an IDP Server	Das PoPPAuthenticator-Modul extrahiert das APDU-Command aus Response-Parameter und sendet es über die NFC-Schnittstelle des Gerätes Einwilligung und Authentisierungsnachweis an die eGK des sekt. IDP.
4951	erzeugeResponseApduCommand führe Authentifizierung durch	Die eGK verarbeitet das APDU-Command und generiert eine Antwort . Der sektorale IDP authentifiziert die VER auf Basis der Authentisierungsnachweise und der ausgewählten Authentifizierungsmethode.

	Schritt	Beschreibung
5052	RETURN responseApduCommandHTTP-302: Redirect-URL(authCodeIDP)	Die eGK beantwortet die Anfrage des PoPP-Modul (responseApduCommand)Der sektorale IDP erstellt einen Authorization-Code (authCodeIDP) und antwortet dem Authenticator-Modul mit einem Redirect auf die URL zum ZETA-AS
5153	generiere httpRequestEgk-Request(telematikId, workplaceId, responseApduCommand)RETURN: authCodeIDP	Das PoPPAuthenticator-Modul generiert einen neuen httpRequestEgk-Request mitantwortet dem zusätzlichen Parameter responseApduCommandAufruf des ZETA Client und übergibt den authCodeIDP
5254	generiere authorizationDetails(responseApduCommand)POST: (authCodeIDP)	Das PoPP-Modul generiert authorization_details mit dem responseApduCommandDer 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.

	Schritt	Beschreibung
5355	<code>execute(httpReadEgkRequest, callbackReadEgk, authorization_details)GET (authCodeIDP)</code>	Das PoPP-Modul ZETA-AS ruft am ZETA-Client die Schnittstelle <code>execute [gemSpec_Zeta]</code> auf. Parameter sind: • <code>httpReadEgkRequest</code> – vom PoPP-Modul erzeugter HTTP-Read-eGK-Request mit den Parametern Telematik-ID, WorkplaceID und der Antwort der eGK auf das APDU-Command (<code>responseApduCommand</code>); • <code>callbackReadEgk</code> – Methode am PoPP-Modul welche der ZETA-Client als Ergebnis-Token durch Übergabe des HTTP-Requests aufruft; <code>authorizationDetails</code> – vom PoPP-Modul erzeugte <code>authorizationDetails</code> Authorization-Code (authCodeIDP) am Token-Endpoint des sektoralen IDP ab.

	Schritt	Beschreibung
5456	<code>httpReadEgkRequest(telematikId, workplaceId, responseApduCommand) Header(clientAccessToken)HTTP-200: ID-Token</code>	Der ZETA Client sendet den HTTP-Request (<code>httpReadEgkRequest</code>) mit sektoraler IDP antwortet dem Client Access-Token an den ZETA Guard HTTP-ProxyAuthorization-Server mit einem ID-Token, welches u.a. die Claims enthält: • <i>urn:telematik:claims:id</i> mit dem Wert der KVNR der VER, • <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 Vertrauensniveaus, auf dem die Authentifizierung erfolgt ist.

	Schritt	Beschreibung
5557	httpReadEgkRequest(telematikId, workplaceId, responseApduCommand) Header(clientAccessToken, Client-Information)prüfe ID-Token	Der ZETA Guard HTTP-Proxy sendetAuthorization-Server prüft die Signatur des ID-Token und entschlüsselt den httpReadEgkRequest mit den erweiterten Client-Informationen an den PoPP-Service Inhalt.
Ende der Wiederholungsschritte58	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.
5659	prüfe signedChallengeHTTP-200: (authCodeAS)	Der PoPP-Service prüft die Antwort der eGK auf das APDU-Command zur Signatur einer Challenge mit dem CV-Zertifikat der eGK.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.

Eingefü

Eingefü

	Schritt	Beschreibung
5760	prüfe X.509-Zertifikat GET: (authCodeAS)	Der PoPP-Service prüft, ob das X.509-Zertifikat aus der Antwort der eGK gültig ist. Der ZETA Client ruft mit dem Authorization-Code-AS (authCodeAS) das Access-Token am Token-Endpoint des ZETA Guard Authorization-Server ab.
5861	prüfe CV-Zertifikat HTTP-200: accessToken	Der PoPP-Service prüft, ob das CV-Zertifikat aus der Antwort der eGK gültig ist. Der ZETA Guard Authorization-Server antwortet dem ZETA Client mit finalem Access-Token.
5962	RS: generiere Hashwerte(X.509-Zertifikat, CV-Zertifikat) GET: httpCheckInGIDRequest(accessToken)	Der PoPP-Service generiert Hashwerte zu den Zertifikaten. Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichem HTTP-CheckInGID-Request mit dem Access-Token auf.

	Schritt	Beschreibung
6063	check(Hash(X.509-Zertifikat), Hash(CV-Zertifikat)) CALL: lade PoPP-Metainformationen zum Access Token	Der PoPP-Service ruft die Hash-DB mit den Hashwerten der Zertifikate auf. 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.
6164	prüfe, ob Hash(X.509-Zertifikat) und Hash(CV-Zertifikat) zusammen passen RETURN: (kvnr, iknr, amr, acr)	Die Hash-DB prüft, ob die übergebenen Hashwerte als gültiges Paar hinterlegt sind. Ist das der Fall, so gehören die Zertifikate zur gleichen eGK. Der ZETA Guard Authorization-Server antwortet dem ZETA Guard HTTP-Proxy mit den Daten.
6265	RETURN ok/nok reichere HTTP-Request mit PoPP-Metainformationen an (kvnr, iknr, amr, acr)	Die Hash-DB liefert dem PoPP-Service das Ergebnis der Prüfung. Der ZETA Guard HTTP-Proxy reicht den PoPP-Modul HTTP-CheckInGID-Request um die PoPP-Metainformationen an.

	Schritt	Beschreibung
6366	extractInformationen(Parameter,reichere HTTP-Header)-Request mit Client-Informationen zur PoPP-Token Generierung an	Der ZETA Guard HTTP-Proxy reichert den PoPP-Service extrahiert: • aus dem Modul HTTP-CheckInGID-Request-Header • die PoPP-Metainformationen zum Versicherten (KVNR, IK-Nummer) • die um Informationen zur Authentifizierung (amr, acr) • Informationen zum aufrufenden Client (ZETA Client) aus den Request Parametern Telematik-ID, WorkplaceID der LEIZETA Guard Client-Registry an.

6467	erstelle PoPP-Datensatz(kvnr, iknr, amr, aer, GET: httpCheckInGIDRequest(telematikId, workplaceId, status, id) Header(accessToken, PoPP-Metainformation, Client-Information))	Der PoPP-Service legt einen PoPP-Datensatz mit allen relevanten Informationen für die Generierung eines PoPP-Token an. Zusätzlich enthält der Datensatz • id eindeutiger Identifier des Datensatzes, dient dem Auffinden zur Abfrage oder Aktualisierung des Status zur PoPP-Token-Generierung • status enthält den Status der PoPP-Token-Generierung • 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.
------	---	--

	Schritt	Beschreibung
		cancelled — Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen. Der ZETA Guard HTTP-Proxy ruft am PoPP-Service Resource Server vom ZETA Guard HTTP-Proxy erweiterten HTTP-CheckInGID-Request auf.

	Schritt	Beschreibung
6568	<code>checkLeiOnline(telematikId)extractInformationen(appDetail, HTTP-Header)</code>	Der PoPP-Service prüft, ob Resource Server extrahiert: • aus dem Request-Header • die LEIPoPP-Metainformationen zur VER (KVNR, IK-Nummer) • die Informationen zur Authentifizierung (amr, acr) • Informationen zum aufrufenden Client • aus den Request Parametern • Telematik-ID-online ist, Workplace ID des Inhabers der Telematik-ID

69	erstelle PoPP-Nachrichten-Datensatz(messageId, status)	Der PoPP-Service Resource Server legt einen Nachrichten Datensatz erstellt mit:: • <i>id</i> - eindeutiger Identifier des Nachrichten-Datensatzes. Diese dient dem Auffinden zur Abfrage oder Aktualisierung des Status zur PoPP-Token-Generierung • <i>status</i> - enthält den Status der PoPP-Token-Generierung • success - PoPP-Token wurde erzeugt und dem Inhaber der Telematik-ID zugestellt • pending - PoPP-Token wurde nicht erzeugt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung wird
----	--	---

Eingefü

Eingefü

	Schritt	Beschreibung
		weiter versucht. • canceled - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen Option: LEI ist online
6670	erzeuge PoppToken(poppDatenSatz) erstelle PoPP-Datensatz(kvnr, iknr, amr, acr, telematikId, workplaceId, messageId)	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. Der PoPP-Service Resource Server legt einen PoPP-Datensatz mit allen relevanten Informationen für die Generierung eines PoPP-Token an.

6771	versende PoPPToken(PoPPToken)createResponse(statusPoppTokenGeneration)	Das Primärsystem ließt den Der PoPP-Service Resource Server erstellt die Antwort an das PoPP-Modul • <i>messageId</i> - Identifier des Nachrichten-Datensatzes • <i>status</i> - Status der PoPP-Token-Generierung • <i>success</i> - PoPP-Token wurde erzeugt und dem Inhaber der Telematik-ID zugestellt • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung wird weiter versucht. • <i>canceled</i> - Prozess wurde aufgrund eines Fehlers oder wegen
------	---	---

	Schritt	Beschreibung
		Überschreitung der maximalen Wartezeit von 72h abgebrochen

6872	aktualisiere-PoPP-Datensatz(id, status)HTTP-201: Response(statusPoPPTokenGeneration)	Der PoPP-Service aktualisiertResource Server antwortet dem ZETA Guard HTTP-Proxy mit HTTP-201 (created) oder einem HTTP-Fehlercode und übergibt den Status im PoPP-Datensatz auf"der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes: • <i>success</i>" wenn - PoPP-Token wurde erzeugt und dem Inhaber der Telematik-ID zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung des PoPP-Token erfolgreich warwird weiter versucht, • <i>canceled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung g der maximalen Wartezeit (z.B. 72h) abgebrochen
------	---	--

	Schritt	Beschreibung
		sowie dem Identifier zum Nachrichten-Datensatz (<i>messageId</i>), welcher die Daten und den Status zur PoPP-Token-Generierung enthält.

73	HTTP-201: Response(statusPoPPTokenGeneration)	Ende der Option: LEI onlineDer ZETA Guard HTTP-Proxy leitet die Antwort vom PoPP-ServiceResource Server weiter, antwortet dem ZETA Client also ebenfalls mit HTTP-201 (created) oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes: • <i>success</i> - PoPP-Token wurde erzeugt und dem Inhaber der Telematik-ID zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung wird weiter versucht, • <i>canceled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen
----	--	--

Eingefügt

Eingefügt

	Schritt	Beschreibung
		sowie dem Identifier zum Nachrichten-Datensatz (<i>messageId</i>), welcher die Daten und den Status zur PoPP-Token-Generierung enthält.

	Schritt	Beschreibung
6974	<code>createResponse(statusPoPPTokenGeneration)</code> RETURN: <code>Response(statusPoPPTokenGeneration)</code>	Der PoPP-Service erstellt die-ZETA Client gibt als Antwort auf die Ausführungsanweisung des <code>httpCheckInGIDRequest</code> das Response-Objekt mit dem Status zu PoPP-Token Generierung an das PoPP-Modul: • 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; • message Detailinformationen zum Status-/ Fehlercodes. id Identifiziert des PoPP-Datensatzes zurück.

		Schritt	Beschreibung
7 0	HTTP-200: Response(status PoPPTokenGener ation)	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 einem der Statuscodes: • <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 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. Start Alternative - Ergebnisverarbeitung des Online Check-in Prozess in der Kassen-App	

Gelöscht

Gelöscht

7175	HTTP-200: Response(statusPoPPTokenGeneration)er mittelte Daten des Telematik-ID Inhabers zur ID des Nachrichten-Datensatz	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: • <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 Überschreitun g der maximalen Wartezeit (z.B. 72h) abgebrochen sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Datensatz, welcher die Daten und den Status zur
------	---	---

	Schritt	Beschreibung
		PoPP-Token-Generierung enthält. Das PoPP-Modul ermittelt aus der ID Nachrichten-Datensatzes die Daten des Telematik-ID Inhabers des ursprünglichen Online Check-in Prozesses.
7276	RETURN: callBackReadEgk(Response(statusPoPPTokenGeneration))stelle der VER den Status des Online Check-in zum Telematik-ID Inhaber dar	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. Das PoPP-Modul ergänzt die Information des Telematik-ID Inhabers um die übermittelte Statusinformation und stellt der VER eine verständliche Nachricht zum Online Check-in dar.
StartEnde Alternative - Auslösen Ergebnissverarbeitung des Online Check-in Prozess in der Erzeugung eines PoPP-Token aus Krankenversicherung Kassen-App		
73Start Alternative - Ergebnissverarbeitung des Online Check-in Prozess in der Anbieter-App	Zeige Status PoPP-Token-Generierung	Das PoPP-Modul wertet die Antwort aus und präsentiert dem Versicherten die Information.

Gelöscht

Gelöscht

	Schritt	Beschreibung
77	erzeuge HTTP-Request(URL=callBackAppURL, Response(statusPoPPTokenGeneration))	Das PoPP-Modul erzeugt einen HTTP-Request für den Rücksprung zur aufrufenden App. Der Status der PoPP-Token-Generierung wird als Request-Parameter gesetzt. Ende Alternative – Auslösen der Erzeugung eines PoPP-Token aus Krankenversicherung-App
78	POST: sende HTTP-POST Request(Response(statusPoPPTokenGeneration))	Start Alternative – Auslösen der Erzeugung eines PoPP-Token aus einer Drittanbieter-App Das PoPP-Modul sendet den HTTP-Request entsprechend der OpenAPI-Spezifikation [I_AnbieterApp_mobile_CheckIn.yaml] an die Rücksprung-URL aus dem initialen Aufruf der App. Der HTTP-Request öffnet über Plattformmechanismen (deeplink, universal link) die Anbieter-App.

Eingefü

Eingefü

	Schritt	Beschreibung
7479	erzeuge HTTP-Request(URL=callBackAppURL, Response(statusPoPPTokenGeneration))sende HTTP-ACCEPTED	Die Anbieter-App bestätigt den Empfang mit einem HTTP-ACCEPTED. Das PoPP-Modul erzeugt einen HTTP-Request für den Rücksprung zur aufrufenden App. Der Status der PoPP-Token-Generierung wird als Request-Parameter gesetzt.
7580	GET: sende HTTP-GET Request(Response(statusPoPPTokenGeneration))ermittle Daten des Telematik-ID Inhabers zur ID des Nachrichten-Datensatz	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. Das PoPP-Modul ermittelt aus der ID Nachrichten-Datensatzes die Daten des Telematik-ID-Inhabers des ursprünglichen Online Check-in Prozesses.

	Schritt	Beschreibung
7681	Verarbeite-Ergebnisstelle der VER den Status des Online Check-in zum Telematik-ID Inhaber dar	Die-Anbieter-App verarbeitet das Ergebnis der PoPP-Token-Generierung je nach Anwendungsfall. Das PoPP-Modul ergänzt die Information des Telematik-ID Inhabers um die übermittelte Statusinformation und stellt der VER eine verständliche Nachricht zum Online Check-in dar.
Ende Alternative - Auslösen Ergebnisverarbeitung des Online Check-in Prozess in der Erzeugung eines PoPP-Token aus einer Drittanbieter Anbieter-App		

9.5 Ablauf "PoPP-Token-Abruf aus einem Primärsystem"

9.5.1 Übersicht zum Ablauf des PoPP-Token-Abruf aus einem Primärsystem

Ist das Primärsystem online und hat sich der Telematik-ID-Inhaber mit seiner SM(C)-B authentifiziert, kann der Abruf der PoPP-Token aus dem Online Check-in durchgeführt werden.

Der PoPP-Service Resource Server ermittelt alle PoPP-Datensätze zur relevanten Telematik-ID, in denen der Status des zugeordneten Nachrichten-Datensatzes den Wert "pending" hat. Hat das Primärsystem zur Telematik-ID in der Anfrage zusätzliche eine Workplace-ID übermittelt, wird diese bei der Ermittlung der PoPP-Datensätze ebenfalls berücksichtigt.

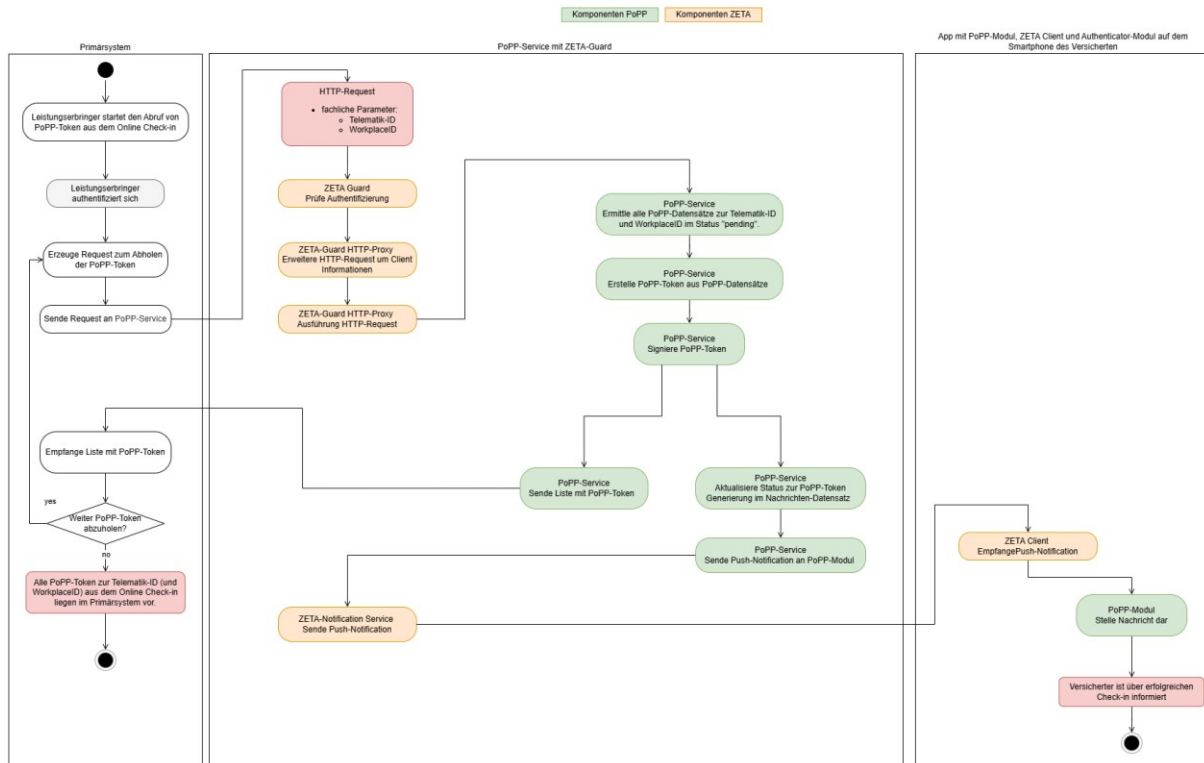


Abbildung 14: Laufzeitsicht - Ablauf Einlösen von PoPP-Token aus dem Online Check-in durch ein Primärsystem

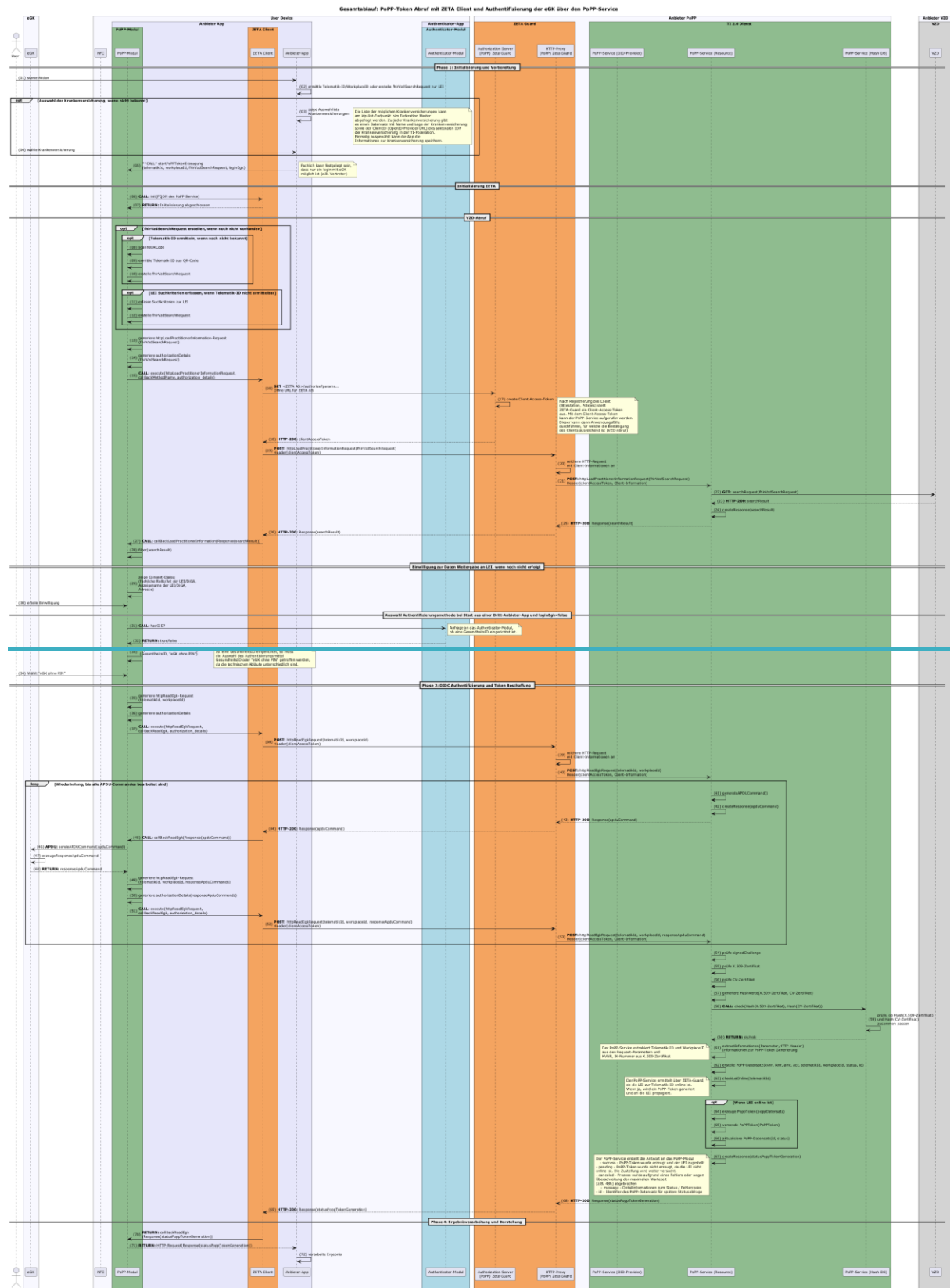
10.6.59.5.2 Detaillierter Ablauf der PoPP-Token-Generierung durch Authentifizierung der eGK über den PoPP-Service mit PoPP-Modul in einer Drittanbieter-App zum PoPP-Token-Abruf aus einem Primärsystem

Das Sequenzdiagramm "Detaillierter Ablauf der zum PoPP-Token-Generierung Abruf aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service in einem Primärsystem" stellt den vollständigen Ablauf der PoPP-Token-Generierung für den Fall dar, dass ein Versicherter den Abruf von PoPP-Token aus einem Primärsystem dar. Der Abruf ist so konzipiert, dass mit einem Aufruf begrenzt viele (limit) PoPP-Token zu einer Telematik-ID von einem Primärsystem beim PoPP-Service Resource Server abgerufen werden können.

Über Push Notification wird die Authentifizierung einer eGK durchgeführt und PoPP-Modul und ZETA-Client in der Drittanbieter-App implementiert sind, informiert, ob ihr Check-in Prozess erfolgreich war oder abgebrochen wurde.

In diesem Ablauf findet kein APP-Sprung statt.

Spezifikation PoPP (Proof of Patient Presence) - Modul



Die Schnittstellen ist in der OpenAPI I_PoPP_Service_mobile_Token_Generation.yaml] spezifiziert.

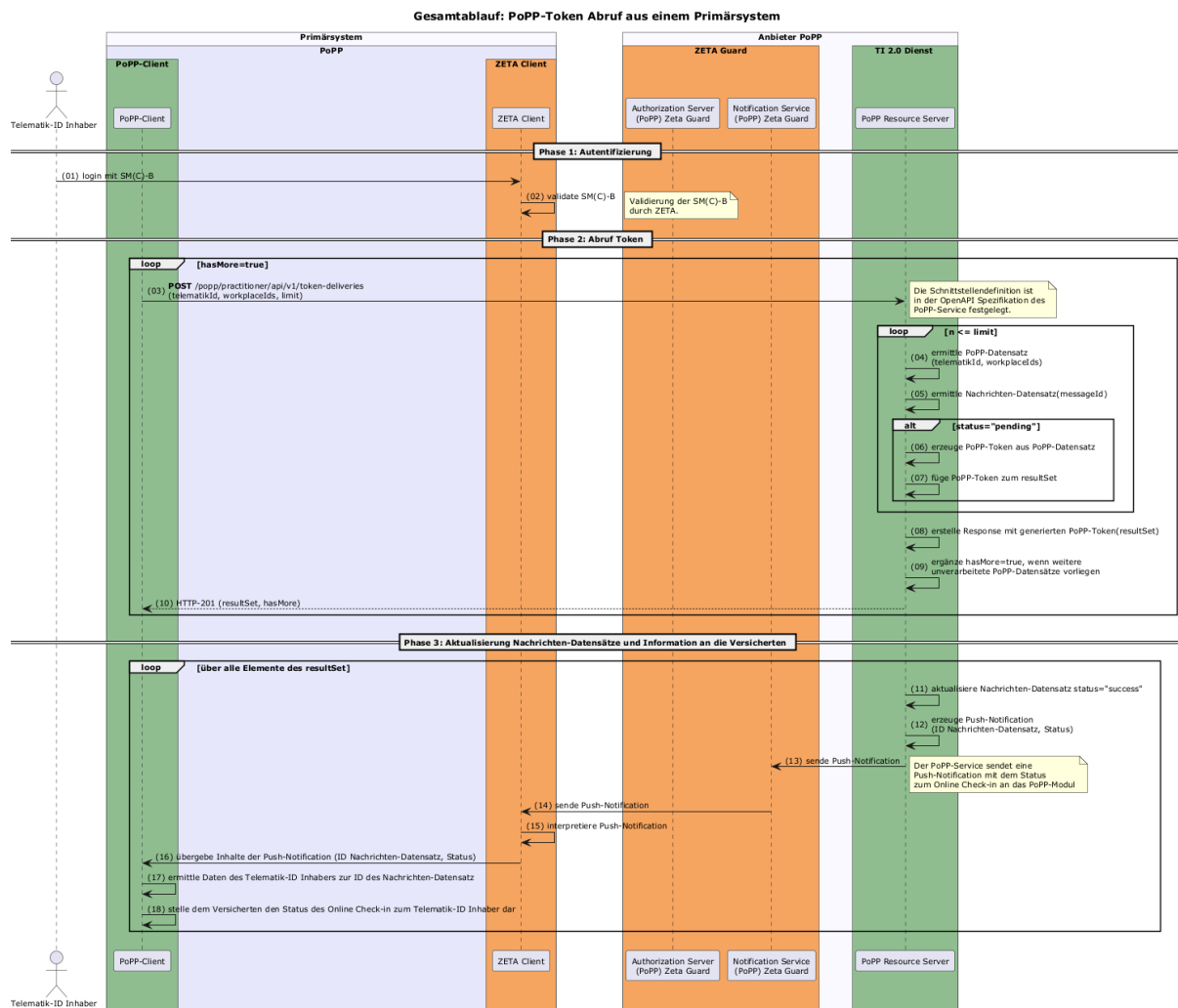


Abbildung 15: Detaillierter Ablauf der zum PoPP-Token-Generierung Abruf aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service in einem Primärsystem

10.6.69.5.3 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung einer eGK mit PoPP-Modul in einer Drittanbieter-App zum PoPP-Token-Abruf aus einem Primärsystem

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 8: Beschreibung der Schritte zur Generierung und zum Abruf von PoPP-Token-Erstellung aus einer Anbieter-App mit PoPP-Modul und Authentifizierung der eGK über den PoPP-Service dem Online Check-in

	Schritt	Beschreibung
Authentifizierung	Aktion starten	Der Versicherte wählt eine Funktion in der App eines Anbieters (z.B. "Anmelden zum Arztbesuch", "Rezept einlösen").
21	ermittle Telematik-ID/WorkplaceID oder erzeuge FHIR VZD Search Request zur Datenermittlung zur LEIlogin mit SM(C)-B	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.: • 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 Auswahlmöglichkeiten an, • eine Praxis kann die Telematik ID als QR-Code bereitstellen. Es werden Suchkriterien erfasst, mit denen eine FHIR VZD-Anfrage durchgeführt werden kann. Der Inhaber der telematik-I authentifiziert sich über sein Primärsystem mit seiner SM(C)-B.

Gelöscht

Gelöscht

	Schritt	Beschreibung
32	zeige Auswahl liste Krankenversicherungen valid ate SM(C)-B	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. Die Validierung der SM(C)-B erfolgt durch die ZETA Client Komponente im Primärsystem.
Ende Authentifizierung		
4	wähle Krankenversi cherung	Der Versicherte wählt seine Krankenversicherung aus der Liste aus. Die Auswahl enthält auch die ClientID (OpenID-Provider URL) des sektoralen IDP der Krankenversicherung. Der Start des PoPP-Token-Abrufs im Primärsystem eines Telematik-ID-Inhabers erfolgt manuell oder automatisiert

Gelösch

Gelösch

53	erzeuge HTTP-GET Request POST:/popp/practitioner/api/v1/token-deliveries\n(telematikId, workplaceId, fhirVzdSearchRequest, callBackAppURL, loginEgklimit)	- Die Anbieter-App ruft eine API-Methode mit dem Aufruf der Schnittstelle am PoPP-Modul zum startPoPPTokenErzeugung auf. Service Resource Server durch das Primärsystem wird Generierung und übergibt Abruf der PoPP-Token aus dem Online Check-in gestartet. - Der Aufruf enthält als Request-Parameter: <i>telematikId</i> ist die Telematik-ID, zu der LEI, für die ein PoPP-Token erstellt abgerufen werden soll, sollen - WorkplaceID (<i>optional</i>) ist eine WorkplaceID (<i>optional</i>), wenn das, zu der die PoPP-Token einem bestimmten Arbeitsplatz der LEI zugeordnet abgerufen werden soll, sollen <i>fhirVzdSearchRequest</i> (<i>optional</i>) Search-Request für den Aufruf der FHIR-VZD-API, um Daten zu einer LEI zu erhalten <i>loginEgk</i> (<i>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. Limit, maximale Anzahl der in einem Request abzuholenden PoPP-Token - Die Schnittstellendefinition ist in der OpenAPI Spezifikation des PoPP-ServiceResource Server [I_PoPP_Service_mobile-Token-Generation] festgelegt.
----	--	---

	Schritt	Beschreibung
64	CALL: init(FQDN-desermittle PoPP-ServiceDatensatz\n(telematikId, workplaceIds)	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. Der PoPP-Service Resource Server ermittelt die PoPP-Datensätze und Nachrichten-Datensätze zu übergebenen Telematik-ID und WorkplaceID, bei denen das Attribut "status" des Nachrichten-Datensatz den Wert "pending" hat. Es werden maximal soviel Datensätze extrahiert, bis "limit" erreicht ist.
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.
95	ermittle Telematik-ID aus QR-CodeNachrichten-Datensatz(messageId)	Der QR-Code wird geprüft und die Telematik-ID sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.
10	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		

Verbund

Verbund

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 httpLoadPractitionerInformationRequest(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: • httpLoadPractitionerInformationRequest — erzeugter HTTP Request • callbackLoadPractitionerInformation — callback-Methodenname der vom ZETA Client aufzurufenden callback-Methode • authorization_details — Erzeugte Authorization Details auf.
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 ursprünglichen httpLoadPractitionerInformationRequest mit dem Client-Access-Token auf.
2 0	reichere HTTP-Request mit Client-Informationen an	Der ZETA-Guard HTTP-Proxy reichert den PoPP-Modul httpLoadPractitionerInformationRequest 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 Einwilligungsdialog 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 KVNR 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, workplaceId)	Das PoPP-Modul generiert HTTP-Read-eGK-Request. Telematik-ID und WorkplaceID sind Parameter des HTTP-HTTP-Read-eGK-Request.
3 6	generiere authorizationDetails()	Die authorization_details enthalten für diesen Aufruf keine Werte.
376	execute(httpReadEgkRequest, callBackReadEgk, authorizationDetails)erzeuge PoPP-Token aus PoPP-Datensatz	DasAus jedem ermittelten PoPP-Modul ruft am ZETA-Client die Schnittstelle execute [gemSpec_Zeta] auf. Parameter sind: • httpReadEgkRequest vom Datensatz wird durch den PoPP-Modul erzeugter HTTP-Read-eGK-Request mit den Parametern Telematik-ID und WorkplaceID; • callBackReadEgk Methode amService Resource Server ein PoPP-Modul welche der ZETA-Client als Ergebnis des HTTP-Requests aufruft; authorizationDetails vom PoPP-Modul erzeugte authorizationDetailsToken generiert.
38	httpReadEgkRequest(telematikId, workplaceId) Header(clientAccessToken)	Der ZETA-Client sendet den HTTP-Request (httpReadEgkRequest) mit dem Client Access-Token an den ZETA Guard HTTP-Proxy.
39	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.
40	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		
41	generateAPDUCommand()	Wird eine Antwort der eGK auf das APDU-Command (responseApduCommand) 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.
42	createResponse(apduCommand)	Der PoPP-Service generiert eine Response mit dem APDU-Command.
43	HTTP-200-Response(apduCommand)	Der PoPP-Service antwortet dem ZETA Guard HTTP-Proxy auf den Request mit HTTP-200 und der generierten Response.
44	HTTP-200-Response(apduCommand)	Der ZETA Guard HTTP-Proxy leitet die Antwort an den ZETA Client weiter.
45	callBackReadEgk(Response(apduCommand))	Der ZETA-Client ruft die callBack-Methode (callBackReadEgk) mit dem Response-Objekt am PoPP-Modul auf.
46	sendeAPDUCommand(apduCommand)	Das PoPP-Modul extrahiert das APDU-Command aus Response-Parameter und sendet es über die NFC-Schnittstelle des Gerätes an die eGK.
47	erzeugeResponseApduCommand	Die eGK verarbeitet das APDU-Command und generiert eine Antwort.
48	RETURN responseApduCommand	Die eGK beantwortet die Anfrage des PoPP-Moduls (responseApduCommand).
49	generiere-httpReadEgk-Request(telematikId, workplaceId, responseApduCommand)	Das PoPP-Modul generiert einen neuen httpReadEgk-Request mit dem zusätzlichen Parameter responseApduCommand.

50	generiere authorizationDetails(responseAduCommand)	Das PoPP-Modul generiert authorization_details mit dem responseAduCommand.
517	execute(httpReadEgkRequest, callbackReadEgk, authorization_details)füge PoPP-Token zum resultSet	Das PoPP-Modul ruft am ZETA Client-Token und die Schnittstelle execute [gemSpec_Zeta] auf. Parameter sind: • httpReadEgkRequest vom PoPP-Modul erzeugter HTTP-Read-eGK-Request mit den Parametern Telematik-ID, zugeordnete WorkplaceID und (wenn vorhanden) werden der Antwort der eGK auf Ergebnismenge hinzugefügt, die an das APDU-Command (responseAduCommand); • callbackReadEgk Methode am PoPP-Modul welche der ZETA Client als Ergebnis des HTTP-Requests aufruft; authorizationDetails vom PoPP-Modul erzeugte authorizationDetailsPrimärsystem zu übermitteln ist.
52	httpReadEgkRequest(telematikId, workplaceId, responseAduCommand) Header(clientAccessToken)	Der ZETA-Client sendet den HTTP-Request (httpReadEgkRequest) mit dem Client-Access-Token an den ZETA-Guard HTTP-Proxy.
53	httpReadEgkRequest(telematikId, workplaceId, responseAduCommand) Header(clientAccessToken, Client-Information)	Der ZETA-Guard HTTP-Proxy sendet den httpReadEgkRequest mit den erweiterten Client-Informationen an den PoPP-Service.
Ende der Wiederholungsschritte		
54	prüfe signedChallenge	Der PoPP-Service prüft die Antwort der eGK auf das APDU-Command zur Signatur einer Challenge mit dem CV-Zertifikat der eGK.
55	prüfe X.509-Zertifikat	Der PoPP-Service prüft, ob das X.509-Zertifikat aus der Antwort der eGK gültig ist.

56	prüfe CV-Zertifikat	Der PoPP-Service prüft, ob das CV-Zertifikat aus der Antwort der eGK gültig ist.
57	RS: generiere Hashwerte(X.509-Zertifikat, CV-Zertifikat)	Der PoPP-Service generiert Hashwerte zu den Zertifikaten.
58	check(Hash(X.509-Zertifikat), Hash(CV-Zertifikat))	Der PoPP-Service ruft die Hash-DB mit den Hashwerten der Zertifikate auf.
59	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.
60	RETURN ok/nok	Die Hash-DB liefert dem PoPP-Service das Ergebnis der Prüfung.
61	extractInformationen(Parameter, HTTP-Header) Informationen zur PoPP-Token-Generierung	

628	erstelle Response mit generierten PoPP-Datensatz(kvnr, iknr, amr, acr, telematikId, workplaceId, status, idToken(resultSet))	Der PoPP-Service legt einen PoPP-Datensatz mit allen relevanten Informationen für Resource Server erstellt die Generierung eines Antwort auf den Anfrage-Request. Die Antwort enthält • Ein ResultSet mit • den generierten PoPP-Token • der WorkplaceID zum PoPP-Token an. Zusätzlich enthält der Datensatz (wenn vorhanden) • id eindeutiger Identifizier des Datensatzes, dient dem Auffinden zur Abfrage oder Aktualisierung des Status zur PoPP-Token-Generierung • status enthält den Status der PoPP-Token-Generierung • 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. Die Information, ob weitere PoPP-Datensätze zur Telematik-ID abrufbar sind.
9	ergänze hasMore=true, wenn weitere\unverarbeitete PoPP-Datensätze vorliegen	
6310	checkLeiOnline(telematikId) HTTP-201 (resultSet, hasMore)	Der PoPP-Service prüft, ob die LEI zur Telematik-ID online ist Resource Server antwortet auf dem Primärsystem-Aufruf.
Option: LEI ist online		

64	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-Ende PoPP-Token-Erstellung und Abruf	
65	versende PoPPToken(PoPPToken)	Das Primärsystem läßt den PoPP-TokenStartAktualisierung Nachrichten-Datensätze und Information an die VER nach erfolgreichem Versand der PoPP-Token	
6611		aktualisiere PoPPNachrichten-Datensatz(id, status)="success"	Nach erfolgreichem Versand der generierten PoPP-Token an das Primärsystem aktualisiert der PoPP-Service aktualisiert Resource Server den Status im PoPPNachrichten-Datensatz zu jedem gesendeten PoPP-Token indem der Wert des Attributs "status" auf "success", wenn die Zustellung des PoPP-Token erfolgreich war" gesetzt wird.
Ende der Option: LEI online			

Gelöscht

Gelöscht

67	createResponse(statusPoppTokenGeneration)	Der PoPP-Service erstellt die Antwort an das PoPP-Modul: • 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, • message Detailinformationen zum Status / Fehlercodes. • id Identifier des PoPP-Datensatzes
----	--	---

6812	HTTP-200:- Response(statusPoPPTokenGeneration)erzeuge Push-Notification\n(ID Nachrichten-Datensatz, Status)	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: • success PoPP-Token wurdeResource Server erzeugt und der LEI zugestellt, • pending PoPP-Token wurde nicht erzeugt, da die LEI nicht online ist.eine Push-Notification gemäß [gemSpec_ZETA]. Die Zustellung wird weiter versucht. • cancelled Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen sowie einer Message mit DetailinformationenPush-Notification enthält die Information zum Status bzw. zum Fehlercode und dem des Online Check-in zur Telematik-ID (und WorkplaceID) und den Identifier zum PoPP-Datensatz, welcher die Daten und den Status zur PoPP-Token-Generierung enthältdes Nachrichten-Datensatzes.
13	sende Push-Notification	Der PoPP-Service Resource Server sendet die Push-Notification den ZETA Notification Service.
14	sende Push-Notification	ZETA Notification Service sendet die Push-Notification an den ZETA-Client.

15	interpretiere Push-Notification	Der ZETA-Client interpretiert die Inhalte des Notification und extrahiert die Nachricht für das PoPP-Modul.
6916	HTTP-200: Response(statusPoPPTokenGeneration) übergebe Inhalte der Push-Notification (ID Nachrichten-Datensatz, Status)	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 die Inhalte der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes: • 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 sowie einer Message mit Detailinformationen zum Status bzw. zum Fehlercode und dem Identifier zum PoPP-Nachricht (ID Nachrichten-Datensatz, welcher die Daten und den Status zur) an das PoPP-Token-Generierung enthält Modul.

7017	RETURN: callBackReadEgk(Response(statusPoPPTokenGeneration))ermittle Daten des Telematik-ID Inhabers zur ID des Nachrichten-Datensatz	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. Das PoPP-Modul ermittelt aus derID Nachrichten-Datensatzes die Daten des Telematik-ID Inhabers des ursprünglichen Online Check-in Prozesses.
7118	RETURN: Request(Response(statusPoPPTokenGeneration))stelle der VER den Status des Online Check-in zum Telematik-ID Inhaber dar	Das PoPP-Modul sendet als Antwort auf den initialen Aufruf der App Response(statusPoPPTokenGeneration). Das PoPP-Modul ergänzt die Information des Telematik-ID Inhabers um die übermittelte Statusinformation und stellt der VER eine verständliche Nachricht zum Online Check-in dar.
72EndeAktualisierung Nachrichten-Datensätze und Information an die VER	Verarbeite Ergebnis	Die Anbieter App verarbeitet das Ergebnis der PoPP-Token-Generierung je nach Anwendungsfall.

Gelöscht

Gelöscht

Offener Punkt: OP-PoPP-7 (PoPP-Token Abruf -Step 16: Push Notification)

Die aktuelle ZETA-Spezifikation (in Kommentierung) legt fest, dass der ZETA Client bzw. das ZETA SDK die Push-Nachricht verarbeitet und den Klartext-Payload an die aufrufende App übergibt.

Dazu gibt es auf Detailebene noch offene Punkte, bspw. kann das PoPP-Modul den Inhalt der Nachricht modifizieren? Z.B. kommt die MessageID und der Status zu einem Online Check-in, angezeigt soll aber ein Text, z.B. "Der Check-in bei <LEI> war erfolgreich", wenn status = success zurück kommt.

Vorgehen:

Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet

9.6 Ablauf "Abfrage des Status zur PoPP-Token-Generierung"

9.6.1 Übersicht zum Ablauf der Abfrage des Status zur PoPP-Token-Generierung

Die Abfrage des aktuellen Status der PoPP-Token-Generierung erfolgt, indem an den PoPP-ServiceResource Server ein Request mit dem Identifier des Nachrichten-Datensatz gestellt wird. Der PoPP-Service Resource Server ermittelt den Nachrichten-Datensatz und gibt in seiner Antwort den dort gespeicherten Status-Wert zur PoPP-Token-Generierung zurück.

Ablauf gleicht in weiten Teilen dem zur Erstellung des PoPP-Datensatzes und des Nachrichten-Datensatzes. So sind die Authentifizierung des Nutzers mit seiner GesundheitsID über die ZETA-Komponenten und die Übermittlung des Ergebnis identisch. Abweichend sind lediglich die Requestparameter und die Funktion selbst, die der PoPP-Service Resource Server bei Eingang des Requests ausführt.

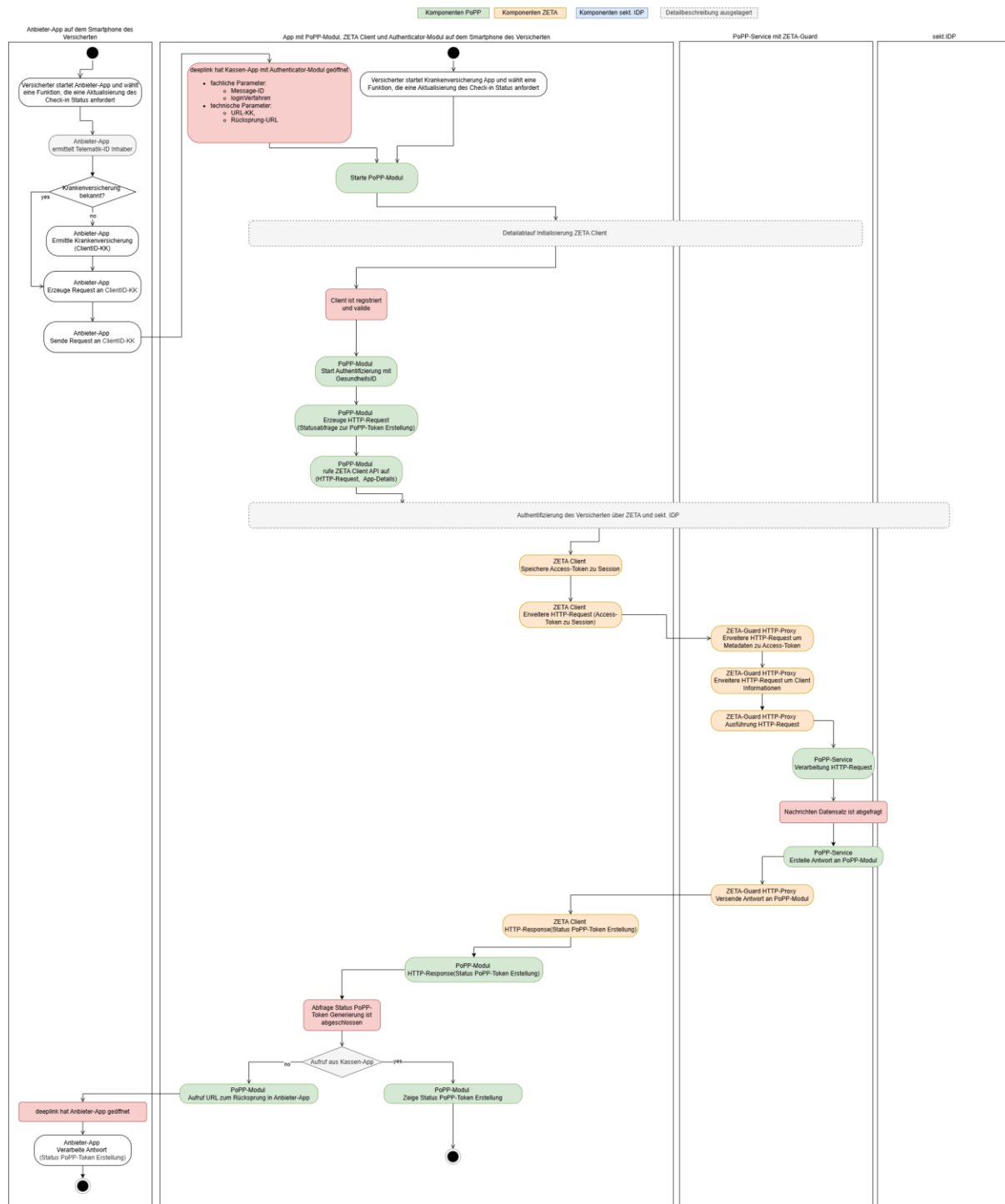


Abbildung 16: Laufzeitsicht - Ablauf der Statusabfrage zur PoPP-Token-Generierung aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID

9.6.2 Detaillierter Ablauf der Abfrage des Status zur PoPP-Token-Generierung

Das Sequenzdiagramm "Detaillierter Ablauf zu Ermittlung des Status zum Online Check-in aus einer Anbieter-App oder Kassen-App und Authentifizierung über die

GesundheitsID" stellt den vollständigen Ablauf für den Fall dar, dass der Status zu einem zuvor durchgeführten Online Check-in Prozess am PoPP-ServiceResource Server abgerufen wird. Der Abruf kann manuell durch aktive Einbeziehung der VER oder automatisiert im Hintergrund zu nicht abgeschlossenen Check-in Prozessen gestartet werden.

Der Status der PoPP-Token-Generierung wird dem PoPP-Modul zum Abschluss des Ablaufs vom PoPP-Service Resource Server zugesendet.

Die Schnittstellen sind den OpenAPI spezifiziert:

- [I_PoPP_Modul_mobile_CheckIn.yaml] - für den Aufrufs des PoPP-Modul aus einer Anbieter-App
- [I_PoPP_Service_mobile_CheckIn.yaml] -für die Aufrufe am PoPP-Service Resource Server vom PoPP-Modul
- [I_AnbieterApp_mobile_CheckIn.yaml] - für den callback-Aufruf der Anbieter-App aus dem PoPP-Modul



Seite 199 von 216
Stand: 03.09.2026

Tabelle 9: Beschreibung der Schritte zum Abruf des Status eines zuvor durchgeführten Online Check-in Prozesses bei Authentifizierung der VER über ihre GesundheitsID

	Schritt	Beschreibung
Start Alternative - Initialisierung Abruf des Status der PoPP-Token-Generierung aus einer Anbieter-App		
1	Aktion starten	Der Start der Abfrage kann manuell durch die VER über das User Interface der App oder automatisiert erfolgen.
2	ermittle ID Nachrichten-Datensatz	Die ermittelt den Identifier des Nachrichten-Datensatzes zu dem Online Check-in Vorgang, zu dem nun der Status abgerufen werden soll. Die ID des Nachrichten-Datensatzes hat das PoPP-Modul der Anbieter-App als Ergebnis des initialen Online Check-in Prozesses übermittelt.
3	zeige Auswahlliste Krankenversicherungen	Ist die Krankenversicherung der VER beim initialen Online Check-in nicht gespeichert worden, so muss sie interaktiv ermittelt werden. Dazu zeigt die Anbieter-App die Liste der möglichen Krankenversicherungen an. 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.
4	wähle Krankenversicherung	Die VER wählt seine Krankenversicherung aus der Liste aus. Die Auswahl enthält auch die ClientID (OpenID-Provider URL, iss) des sektoralen IDP der Krankenversicherung.

	Schritt	Beschreibung
5	erzeuge HTTP-POST Request(messageId, callBackAppURL, loginVerfahren)	Die Anbieter-App erzeugt einen HTTP-Request entsprechend der OpenAPI Spezifikation [I_PoPP_Modul_mobile_CheckIn.yam l] mit den Parametern: • <i>messageId</i> -ist die Telematik-ID eines Telematik-ID Inhabers, für die ein PoPP-Token erstellt werden soll, • <i>callBackURL</i> (Rücksprung-URL)- welche das PoPP-Modul aufrufen muss, wenn der Statusabruf abgeschlossen ist, • <i>Wertebereich: "gID", wenn der Login durch die Authentifizierung mit GesundheitsID erfolgen muss.</i> <i>Hinweise: Mit Einführung weiterer Verfahren wie eGK-in-Fernversorgung oder EUDI-Wallet wird der Wertebereich erweitert.</i>
6	sende HTTP-POST Request(messageId, callBackAppURL, loginVerfahren)	Die Anbieter-App sendet den HTTP-Request entsprechend der OpenAPI Spezifikation [I_PoPP_Modul_mobile_CheckIn.yam l] an OpenID-Provider URL des sektoralen IDP der Krankenversicherung <iss>. 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 Kassen-App mit integriertem Authenticator-Modul und PoPP-Modul. Aufgrund des Requests wird das PoPP-Modul in der Kassen-App aufgerufen.
7	sende HTTP-ACCEPTED	Das PoPP-Modul bestätigt den Empfang mit einem HTTP-ACCEPTED.

Schritt		Beschreibung
Ende Alternative - Initialisierung Abruf des Status der PoPP-Token-Generierung aus einer Anbieter-App		
Start Alternative - Initialisierung Abruf des Status der PoPP-Token-Generierung aus Kassen-App		
8	Aktion starten	Der Start der Abfrage kann manuell durch die VER über das User Interface der App oder automatisiert erfolgen.
9	ermittle ID Nachrichten-Datensatz	Die ermittelt den Identifier des Nachrichten-Datensatzes zu dem Online Check-in Vorgang, zu dem nun der Status abgerufen werden soll. Die ID des Nachrichten-Datensatzes hat das PoPP-Modul der Anbieter-App als Ergebnis des initialen Online Check-in Prozesses übermittelt.
Ende Alternative - Initialisierung Abruf des Status der PoPP-Token-Generierung aus Kassen-App		
1 0	CALL: initialisiereZETAClient(FQDN des PoPP-Service Resource Server)	Beim ersten Aufruf einer Session wird der ZETA Client initialisiert. Dazu wird der FQDN des PoPP-ServiceResource Server übergeben. Der FQDN des PoPP-Service Resource Server 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.
1 1	RETURN: Initialisierung abgeschlossen	Nach Abschluss der Initialisierung kehrt die Anwendung in das PoPP-Modul zurück.
Ablauf Abruf des Status der PoPP-Token-Generierung		

	Schritt	Beschreibung
1 2	Zeige Auswahl der Authentisierungsmittel	Das PoPP-Modul zeigt der VER die Auswahl möglicher Authentisierungsmethoden an. Aktuell ist dies die GesundheitsID. Weitere mögliche Authentisierungsmethoden wären perspektivisch die eGK (in Fernversorgung) und die EUDI-Wallet.
1 3	Wählt GesundheitsID	Die VER wählt "GesundheitsID" aus.
1 4	generiere httpReadStatusPoPP-Request(messaged)	Das PoPP-Modul generiert HTTP-Request entsprechend der Schnittstellendefinition [I_PoPP_Service_mobile_CheckIn.yaml] mit dem Identifier des Nachrichten-Datensatzes zum Online Check-in Prozess als Parameter des Requests.
1 5	generiere authorizationDetails(openIdProviderUrl)	Das PoPP-Modul generiert authorizationDetails als JSON-Objekt welches die Client-ID des sekt. IDP (openIdProviderURL) enthält. <pre>{ "app_details": { "auth_preferences" : { "openIdProviderUrl" : "<Client-ID des sekt. IDP>" } }</pre>

	Schritt	Beschreibung
1 6	CALL: execute(httpReadStatusPoPP, authorizationDetails)	Das PoPP-Modul ruft die Schnittstellen-Methode zur Ausführung des HTTP-Request am ZETA Client [gemSpec_ZETA] auf. Parameter sind: • <i>httpReadStatusPoPP</i>- vom PoPP-Modul erzeugter HTTP-httpReadStatusPoPP-Request mit dem Identifier des Nachrichten-Datensatzes zum Online Check-in Prozess als Parameter, • <i>authorizationDetails</i> - vom PoPP-Modul erzeugte authorizationDetails mit Informationen zur Authentifizierung der VER.
1 7	GET <ZETA AS>/authorize?params... Öffne 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.
1 8	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.
1 9	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.
2 0	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.

	Schritt	Beschreibung
2 1	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.
2 2	CALL: authorize(openIdProviderUrl+request_uri)	Der ZETA Client ruft eine Methode am Authenticator-Modul mit der request_uri als Parameter auf.
2 3	GET: (openIdProviderUrl+request_uri)	Die Kassen-App mit Authenticator-Modul führt den Authorization Request (redirect_uri) an den Auth-Endpunkt des sektoralen IDP aus.
2 4	send Authentication Information	Der sektorale IDP übermittelt dem Authenticator-Modul der Kassen-App die notwendigen Informationen zur Authentifizierung (z.B. Status Gerätebindung, Einwilligungen, Authentisierungsmittel).
2 5	Führe Authentifizierung durch	Das Authenticator-Modul führt die VER durch die Authentifizierung. Je nach Konfiguration kann die Auswahl der möglichen Authentisierungsmittel angezeigt werden oder direkt zur Erfassung der Authentisierungsnachweise der präferierten Authentifizierungsmethode.
2 6	Eingabe Authentisierungsnachweise	Die VER erfasst die Authentisierungsnachweise zur ausgewählten Authentifizierungsmethode.
2 7	Sende Authentisierungsnachweise und Einwilligung an IDP Server	Das Authenticator-Modul sendet Einwilligung und Authentisierungsnachweis an den sekt. IDP.
2 8	führe Authentifizierung durch	Der sektorale IDP authentifiziert die VER auf Basis der Authentisierungsnachweise und der ausgewählten Authentifizierungsmethode.

	Schritt	Beschreibung
2 9	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
3 0	RETURN: authCodeIDP	Das Authenticator-Modul antwortet dem Aufruf des ZETA Client und übergibt den authCodeIDP
3 1	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.
3 2	GET (authCodeIDP)	ZETA-AS ruft den ID-Token durch Übergabe des Authorization-Code (authCodeIDP) am Token-Endpoint des sektoralen IDP ab.
3 3	HTTP-200: ID-Token	Der sektorale IDP antwortet dem ZETA Guard Authorization-Server mit einem ID-Token.
3 4	prüfe ID-Token	Der ZETA Guard Authorization-Server prüft die Signatur des ID-Token und entschlüsselt den Inhalt.
3 5	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.
3 6	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.
3 7	GET: (authCodeAS)	Der ZETA Client ruft mit dem Authorization-Code-AS (authCodeAS) das Access-Token am Token-Endpoint des ZETA Guard Authorization-Server ab.

	Schritt	Beschreibung
3 8	HTTP-200: accessToken	Der ZETA Guard Authorization-Server antwortet dem ZETA Client mit finalem Access-Token.
3 9	GET: httpReadStatusPoPPRequest(accessToken)	Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichem HTTP-ReadStatusPoPP-Request mit dem Access-Token auf.
4 0	reichere HTTP-Request mit Client-Informationen an	Der ZETA Guard HTTP-Proxy reichert den PoPP-Modul HTTP-ReadStatusPoPP-Request um Informationen zum Client (ZETA Client) aus der ZETA Guard Client-Registry an.
4 1	GET: httpReadStatusPoPPRequest(messageId) Header(Client-Information)	Der ZETA Guard HTTP-Proxy ruft am PoPP-Service Resource Server vom ZETA Guard HTTP-Proxy erweiterten HTTP-ReadStatusPoPP-Request auf.
4 2	extractInformationen(appDetail, HTTP-Header)	Der PoPP-Service Resource Server extrahiert die ID des Nachrichten-Datensatzes (messageId) aus den Request Parameter.
4 3	ermittel PoPP-Nachrichten-Datensatz(messageId)	Der PoPP-Service Resource Server ermittelt den Nachrichten-Datensatz zur messageId.
4 4	ermittle des Statuswert im Nachrichten-Datensatz	Der PoPP-Service Resource Server ermittelt den Wert des Attributes "status" aus dem Nachrichten-Datensatz.

	Schritt	Beschreibung
4 5	createResponse(statusPoppTokenGeneration)	Der PoPP-Service Resource Server erstellt die Antwort an das PoPP-Modul • <i>messageId</i> - Identifier des Nachrichten-Datensatzes • <i>status</i> - Status der PoPP-Token Generierung • <i>success</i> - PoPP-Token wurde erzeugt und dem Inhaber der Telematik-ID zugestellt • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung wird weiter versucht. • <i>canceled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit von 72h abgebrochen

	Schritt	Beschreibung
4 6	HTTP-201: Response(statusPoPPTokenGeneration)	Der PoPP-Service Resource Server antwortet dem ZETA Guard HTTP-Proxy mit HTTP-201 (created) oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes: • <i>success</i> - PoPP-Token wurde erzeugt und dem Inhaber der Telematik-ID zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung wird weiter versucht, • <i>canceled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen sowie dem Identifier zum Nachrichten-Datensatz (<i>messageId</i>), welcher die Daten und den Status zur PoPP-Token-Generierung enthält.

	Schritt	Beschreibung
4 7	HTTP-201: Response(statusPoPPTokenGeneration)	Der ZETA Guard HTTP-Proxy leitet die Antwort vom PoPP-Service Resource Server weiter, antwortet dem ZETA Client also ebenfalls mit HTTP-201 (created) oder einem HTTP-Fehlercode und übergibt den Status der PoPP-Token-Generierung. Der Status besteht aus einen der Statuscodes: • <i>success</i> - PoPP-Token wurde erzeugt und dem Inhaber der Telematik-ID zugestellt, • <i>pending</i> - PoPP-Token wurde nicht erzeugt, da der Inhaber der Telematik-ID nicht online ist. Die Zustellung wird weiter versucht, • <i>canceled</i> - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen sowie dem Identifier zum Nachrichten-Datensatz (<i>messageId</i>), welcher die Daten und den Status zur PoPP-Token-Generierung enthält.
4 8	RETURN: Response(statusPoPPTokenGeneration)	Der ZETA Client gibt als Antwort auf die Ausführungsanweisung des <i>httpReadStatusPoPPRequest</i> das Response-Objekt mit dem Status zu PoPP-Token-Generierung an das PoPP-Modul zurück.
Start Alternative - Auswertung des Abruf zum Status des Online Check-in in der Kassen-App		
4 9	ermittle Daten des Telematik-ID Inhabers zur ID des Nachrichten-Datensatz	Das PoPP-Modul ermittelt aus derID des Nachrichten-Datensatzes die Daten des Telematik-ID-Inhabers des ursprünglichen Online Check-in Prozesses.

	Schritt	Beschreibung
50	stelle der VER den Status des Online Check-in zum Telematik-ID-Inhaber dar	Das PoPP-Modul ergänzt die Information des Telematik-ID-Inhabers um die übermittelte Statusinformation und stellt der VER eine verständliche Nachricht zum Online Check-in dar.
Ende Alternative - Auswertung des Abruf zum Status des Online Check-in in Kassen-App		
Start Alternative - Auswertung des Abruf zum Status des Online Check-in in einer Anbieter-App		
51	erzeuge HTTP-Request(URL=callBackAppURL, Response(statusPoPPTokenGeneration))	Das PoPP-Modul erzeugt einen HTTP-Request für den Rücksprung zur aufrufenden App. Der Status der PoPP-Token-Generierung wird als Request-Parameter gesetzt.
52	POST: sende HTTP-POST Request(Response(statusPoPPTokenGeneration))	Das PoPP-Modul sendet den HTTP-Request entsprechend der OpenAPI-Spezifikation [I_AnbieterApp_mobile_CheckIn.yaml] an die Rücksprung-URL aus dem initialen Aufruf der App. Der HTTP-Request öffnet über Plattformmechanismen (deeplink, universal link) die Anbieter-App.
53	sende HTTP-ACCEPTED	Die Anbieter-App bestätigt den Empfang mit einem HTTP-ACCEPTED.
54	ermittle Daten des Telematik-ID Inhabers zur ID des Nachrichten-Datensatz	Das PoPP-Modul ermittelt aus derID Nachrichten-Datensatzes die Daten des Telematik-ID-Inhabers des ursprünglichen Online Check-in Prozesses.
55	stelle der VER den Status des Online Check-in zum Telematik-ID-Inhaber dar	Das PoPP-Modul ergänzt die Information des Telematik-ID-Inhabers um die übermittelte Statusinformation und stellt der VER eine verständliche Nachricht zum Online Check-in dar.
Ende Alternative - Auswertung des Abruf zum Status des Online Check-in in einer Anbieter-App		

11.10 Anhang C – Offene Punkte, Fragen

11.10.1 Offene Punkte

ID	Fundstelle	Beschreibung
OP-PoP P-1	4.1.13 [gemSpec_PoPP-Service#4.5.3]: PoPP-Token bei Online Check-in	OP-PoPP-1 (zusätzliche Push-Notification für PS) Für PoPP Stufe 2 ist noch nicht festgelegt, ob und wie Primärsysteme über neu bereitgestellte PoPP-Token aktiv informiert werden oder diese ausschließlich durch regelmäßige Abrufe ermitteln. Die hierfür notwendigen Festlegungen befinden sich derzeit in Abstimmung. Vorgehen zur Auflösung: Klärung innerhalb der gematik gemeinsam mit den verantwortlichen ZETA-Teams parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die Spezifikationen eingearbeitet.
OP-PoP P-2	[gemSpec_Popp_Modul#3.2.1.3]	Offener Punkt: OP-PoPP-2 Für die Authentisierung eines Versicherten mit GesundheitsID müssen Informationen zum verwendeten sektoralen Identity Provider der Krankenversicherung an die ZETA-Komponenten übergeben werden. Die Ausgestaltung dieser Übergabe und die hierfür verwendeten Schnittstellenparameter sind derzeit noch nicht abschließend festgelegt. (Stichwort: idp_iss via authorization_details,) Vorgehen: Klärung innerhalb der gematik gemeinsam mit den verantwortlichen ZETA-Teams parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die Spezifikationen eingearbeitet.

Gelösch

ID	Fundstelle	Beschreibung	
OP-PoP-P-3	Kap. 2.1 (PoPP- [gemSpec_Popp_Modul}#3.2.2.2]	<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> Offener Punkt: OP-PoPP-3 (DeepLink) Die derzeit spezifizierte Übergabe von Informationen zwischen Anbieter-App und Kassen-App mittels HTTP-POST über Deep Links ist hinsichtlich ihrer technischen Umsetzbarkeit noch nicht abschließend geklärt. Insbesondere ist offen, ob die vorgesehene Übergabeform durch die relevanten mobilen Betriebssysteme und die hierfür vorgesehenen Mechanismen zuverlässig unterstützt wird. Für die Umsetzung des Online Check-in wird gegebenenfalls eine alternative Lösung auf Basis von App-Links und einer parameterbasierten Übergabe innerhalb der Ziel-URL erforderlich. Vorgehen: Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Die betroffenen Spezifikationsstellen werden identifiziert und konsistent angepasst. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet.	▲ ▲

Gelöscht

Gelöscht

ID	Fundstelle	Beschreibung	
OP-PoP P-4	[gemSpec_Popp_Modul#5.6.3]	Offener Punkt: OP-PoPP-4 (2-Geräte-Flow) Wie der 2-Geräte-Flow abzubilden ist, ist derzeit noch offen. Vorgehen: Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet.	
OP-PoP P-25	Kap 8.1 (Testkonzept)[gemSpec_Popp_Modul#5.6.4]	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. Offener Punkt: OP-PoPP-5 - (Anforderungen an Anbieter-App) Im Zusammenspiel mit dem PoPP-Token-Abruf ergeben sich Anforderungen an die Anbieter-Apps - das sind zum Beispiel Videosprechstunden-Apps, die PoPP nutzen und selbst kein PoPP-Modul integrieren. Es ist noch offen, ob es hierfür einen eigenen Steckbrief gibt, bzw. an wen und wie dieses Anforderungen im Anforderungsmanagement geführt werden. Vorgehen: Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet.	

ID	Fundstelle	Beschreibung
OP-PoP P-6	[gemSpec_Popp_Modul#8.2]	Offener Punkt: OP-PoPP-6 (Bezeichnung von Kassen-Systemen als Primärsysteme) Die Erläuterung zum Eintrag Primärsystem bezeichnet auch die Systeme der Kassen, die einen PoPP-Client zum PoPP-Token-Abruf implementiert haben, als Primärsysteme. Konkret: "Primärsystem- Primärsystem im Kontext der Spezifikation umfasst alle Systeme, die ein PoPP-Token vom PoPP-Service erhalten. Unter Primärsystem fallen demnach PVS, AVS und KIS sowie Systeme von Kostenträgern." Vorgehen: Wenn es im Rahmen der Kommentierung keinen Widerspruch -von Kostenträgern- dazu gibt, würde der offene Punkt geschlossen werden und die aktuell gewählte Formulierung verwendet.
OP-PoP P-7	[gemSpec_Popp_Modul#9.5.3]	Offener Punkt: OP-PoPP-7 (PoPP-Token Abruf -Step 16: Push Notification) Die aktuelle ZETA-Spezifikation (in Kommentierung) legt fest, dass der ZETA Client bzw. das ZETA SDK die Push-Nachricht verarbeitet und den Klartext-Payload an die aufrufende App übergibt. Dazu gibt es auf Detailebene noch offene Punkte, bspw. kann das PoPP-Modul den Inhalt der Nachricht modifizieren? Z.B. kommt die MessageID und der Status zu einem Online Check-in, angezeigt soll aber ein Text, z.B. "Der Check-in bei <LEI> war erfolgreich", wenn status = success zurück kommt. Vorgehen: Prüfung und Abstimmung innerhalb der gematik parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die betroffenen Spezifikationen eingearbeitet

Gelösch

ID	Fundstelle	Beschreibung
OP-PoP-P-8	4.1.13 [gemSpec_PoPP-Service#4.5.3]: PoPP-Token bei Online Check-in	Offener Punkt: OP-PoPP-8 (Berücksichtigung der WebSocket-Schnittstelle in AF_10390) Mit Einführung der Schnittstelle [I_PoPP_Service_mobile_Token_Generation_async.yaml] steht neben REST auch eine WebSocket-basierte Zustellung von PoPP-Token zur Verfügung. Der Anwendungsfall AF_10390 beschreibt bislang den REST-basierten Abruf. Zu prüfen ist, ob AF_10390 um die WebSocket-Nutzung erweitert werden kann oder ein eigener Anwendungsfall erforderlich ist. Dabei sind insbesondere die Token-Zustellung über bestehende Verbindungen, die Zuordnung von Telematik-ID und WorkplaceID sowie das Zusammenspiel von REST und WebSocket zu betrachten. Vorgehen: Erstellung eines neuen Anwendungsfalls oder Erweiterung von AF-10390 parallel zur Kommentierung. Das Ergebnis wird über einen Eigenkommentar der gematik veröffentlicht und in die Spezifikationen eingearbeitet.

Gelöscht