

Telematikinfrastruktur 2.0

Spezifikation PoPP (Proof of Patient Presence) -Modul

Version: 1.0.0_CC4
Revision: 1713012
Stand: 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	26.01.202 6		initiale Erstellung - zur Abstimmung freigegeben	gematik
1.0.0 CC2	11.05.202 6			gematik
1.0.0 CC3	05.08.202 6		zur Abstimmung	gematik
1.0.0 CC4	03.09.202 6		zur Abstimmung freigegeben	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes.....	6
1.1 Zielsetzung.....	6
1.2 Zielgruppe.....	6
1.3 Geltungsbereich.....	6
1.4 Abgrenzungen.....	7
1.5 Methodik.....	7
2 Systemüberblick/Systemkontext.....	8
2.1 Detailsicht "Online-Anwendungsfälle zur PoPP-Token-Erstellung".....	8
3 Übersicht Funktionsmerkmale und Schnittstellen.....	15
3.1 Authentifizierung mit GesundheitsID.....	16
3.2 Schnittstellen des PoPP-Moduls.....	16
3.2.1 App interne Schnittstellen des PoPP-Modul.....	17
3.2.1.1 Initialisierung des ZETA Client.....	17
3.2.1.2 Ermitteln der Daten zu einer Telematik-ID (FHIR-VZD-Abruf).....	17
3.2.1.3 Authentifizierung mit GesundheitsID.....	18
3.2.1.4 Abfrage Status PoPP-Token-Generierung.....	18
3.2.2 Schnittstellen des PoPP-Modul zum Aufruf aus anderen Anwendungen.....	19
3.2.2.1 HTTP-Request - start PoPPTokenGeneration.....	19
3.2.2.2 HTTP-Request - status PoPPTokenGenerierung.....	19
4 Übergreifende Festlegungen.....	20
4.1 Datenschutz und Informationssicherheit.....	21
4.2 Barrierefreiheit.....	22
5 Funktionsmerkmale.....	24
5.1 Allgemeine funktionale Anforderungen PoPP-Modul.....	24
5.2 Anforderungen an die Kommunikation des PoPP-Modul mit dem ZETA Client.....	25
5.3 Bestimmen der Telematik-ID.....	25
5.3.1 QR-Code.....	26
5.3.2 Verzeichnisdienstsuche.....	27
5.4 Anforderungen PoPP-Modul bei Authentisierung einer VER mit GesundheitsID.....	28
5.5 Verarbeitung des Ergebnisses der PoPP-Token-Erzeugung.....	30
5.6 Kommunikation PoPP-Modul mit Anbieter-Apps.....	30
5.6.1 App2App-Flow.....	31
5.6.2 Web2App-Flow.....	32
5.6.3 2-Geräte-Flow.....	33

79	5.6.4 Anforderungen an die Kommunikation zwischen PoPP-Modul und Anbieter-Apps	
80		34
81	6 Informationsmodell.....	37
82	7 Implementierungsleitfaden.....	38
83	7.1 End-to-End Flow (Referenzpfad).....	38
84	7.2 Anbieter-Apps.....	40
85	7.3 Alternative Flow-Schritte.....	40
86	7.3.1 Erfassung der Telematik-ID.....	41
87	7.3.2 Statusinformation.....	44
88	7.3.3 Ersteinrichtung.....	44
89	7.4 Terminologie.....	46
90	8 Anhang A - Verzeichnisse.....	48
91	8.1 Abkürzungen.....	48
92	8.2 Glossar.....	48
93	8.3 Abbildungsverzeichnis.....	51
94	8.4 Tabellenverzeichnis.....	52
95	8.5 Referenzierte Dokumente.....	53
96	8.5.1 Dokumente der gematik.....	53
97	8.5.2 Weitere Dokumente.....	54
98	9 Anhang - Ablaufbeschreibungen.....	56
99	9.1 Gesamtüberblick Auslösung und Ablauf der PoPP-Token-Generierung....	56
100	9.2 Detailablauf "ZETA Client Initialisierung".....	57
101	9.3 Detailablauf "FHIR-VZD-Anfrage".....	57
102	9.4 Detailablauf "Authentifizierung mit GesundheitsID".....	58
103	9.4.1 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung VER	
104	über ihre GesundheitsID.....	58
105	9.4.2 Detaillierter Ablauf des Online Check-in durch Authentifizierung VER über ihre	
106	GesundheitsID.....	60
107	9.4.3 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung VER	
108	mit ihrer GesundheitsID.....	62
109	9.5 Ablauf "PoPP-Token-Abruf aus einem Primärsystem".....	77
110	9.5.1 Übersicht zum Ablauf des PoPP-Token-Abruf aus einem Primärsystem.....	77
111	9.5.2 Detaillierter Ablauf zum PoPP-Token-Abruf aus einem Primärsystem.....	78
112	9.5.3 Beschreibung der Schritte zum PoPP-Token-Abruf aus einem Primärsystem....	78
113	9.6 Ablauf "Abfrage des Status zur PoPP-Token-Generierung".....	81
114	9.6.1 Übersicht zum Ablauf der Abfrage des Status zur PoPP-Token-Generierung....	81
115	9.6.2 Detaillierter Ablauf der Abfrage des Status zur PoPP-Token-Generierung.....	82
116	9.6.3 Beschreibung der Schritte des Ablaufs der Abfrage des Status zur PoPP-Token-	
117	Generierung.....	84
118	10 Anhang C - Offene Punkte, Fragen.....	95
119	10.1 Offene Punkte.....	95

120

121

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].

Das PoPP-Modul bietet versicherten Personen (VER) mit GesundheitsID den Zugang zur PoPP-Lösung. Der PoPP-Service erzeugt die Bestätigung eines Versorgungskontextes (VK) in Form eines kryptographisch gesicherten PoPP-Token. Dieses bestätigt, dass eine bestimmte VER mit dem Inhaber einer bestimmten 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].

1.2 Zielgruppe

Dieses Dokument richtet sich an Hersteller und Betreiber von Kassen-Apps, die ein PoPP-Modul integrieren wollen, insbesondere an die Kassen und ihre Auftragnehmer, sowie 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 8]).

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 erfüllen funktionale und sicherheitstechnische Anforderungen der gematik, die in einem Produkttypsteckbrief bekanntgegeben werden. Die Erfüllung der Anforderungen weist der Hersteller eines PoPP-Moduls in einem Zulassungsverfahren der gematik nach.

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

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) sollen die Vertrauenswürdigkeit des Smartphone der VER und der für PoPP zum Einsatz kommenden Anwendung auf 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 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.

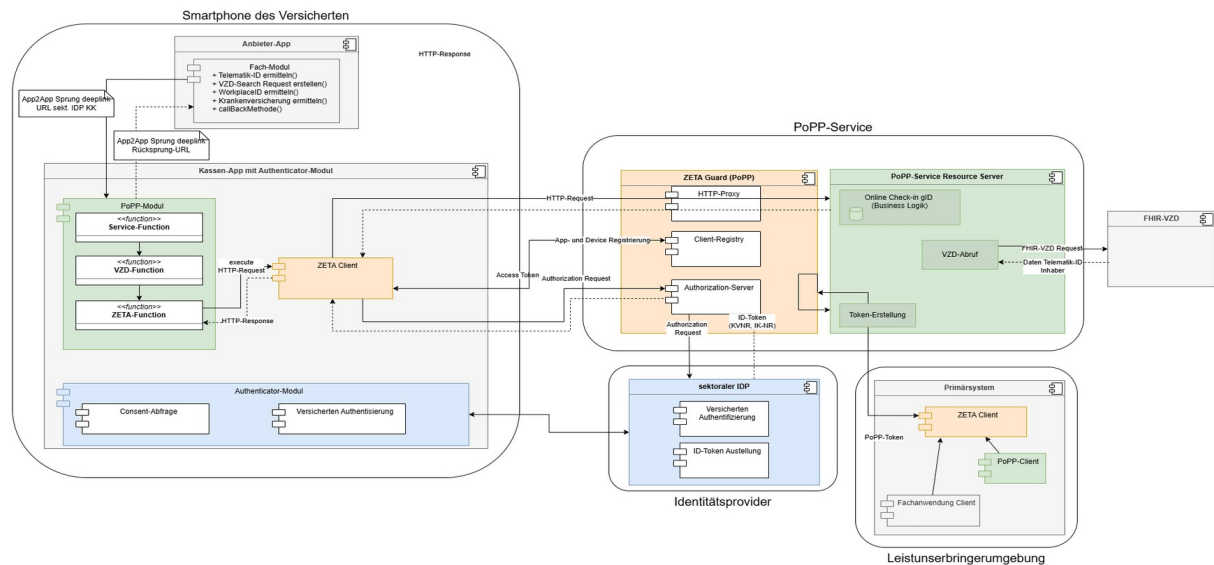


Abbildung 1: Verteilungssicht der beteiligten Komponenten für die Online-Anwendungsfälle zur PoPP-Token-Erstellung mit PoPP-Modul in App der Krankenversicherung und nutzender Anbieter-App auf dem Smartphone der VER

PoPP-Modul und ZETA Client müssen gemeinsam in einer Kassen-App integriert sein. Das PoPP-Modul wird dabei in dem Anwendungsteil der Kassen-App bereitgestellt, der auch das Authenticator-Modul des sektoralen IDP der Krankenversicherung implementiert. Die Abbildung „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. Bei Krankenversicherungen, 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 Anbieter-Apps werden über einen App2App-Sprung in die Kassen-App mit PoPP-Modul unterstützt. Nach Abschluss des Online Check-ins erfolgt, sofern erforderlich, ein 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 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 zuzulassen.

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

Komponente	Beschreibung
------------	--------------

Smartphone der VER	
Smartphone der VER	Auf dem Gerät der VER (in der Regel ein Smartphone) sind Anwendungen (Apps) installiert, die für ihren fachlichen Ablauf die Erstellung eines PoPP-Token für den Inhaber einer Telematik-ID benötigen. Auf dem Gerät ist eine App installiert, welche PoPP-Modul, ZETA Client und Authenticator-Modul integriert (Kassen-App der Krankenversicherung). Das Gerät verfügt über eine Menge von Eigenschaften, welche bei der sicheren Kommunikation mit den Backend-Systemen berücksichtigt werden. 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.
Anwendung/App auf dem Smartphone der VER	Die auf dem Smartphone der VER installierten Anwendungen, welche die Erstellung eines PoPP-Token anfordern, können vielfältig ausgeprägt sein: 1. Personalisierte Anwendungen (Kassen-Apps) für VER zur Vermittlung einer Kommunikation zwischen Versicherten und dem Inhabers einer Telematik-ID 2. Anwendung eines Anbieters für VER (z.B. Online-Apotheke) 3. Anwendung zur Vermittlung der Kommunikation zwischen VER und Inhabern einer Telematik-ID (z.B. Telemedizin-Anwendungen) 4. Anwendungen für eine bestimmte Gruppe von VER zur Kommunikation mit einer bestimmten Gruppe von Telematik-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 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 mit Suchkriterien zum Inhaber

	einer Telematik-ID durchgeführt werden? 3. Ist die Krankenversicherung der VER bekannt oder muss diese ermittelt werden (Kassen-App vs. Online-Apotheke)? Fehlende Informationen müssen durch die Anwendung selbst beschafft werden, bevor eine PoPP-Token-Erstellung durchgeführt werden kann.
Kassen-App mit Authenticator-Modul auf dem Smartphone der VER	Werden VER 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 Kassen-App mit Authenticator-Modul ist die App auf dem Smartphone der VER, welches das Authenticator-Modul des sektoralen IDP der Krankenversicherung der VER implementiert. Es gibt zwei Ausprägungen der Umsetzung: 1. Das Authenticator-Modul ist in einer 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 Smartphone der VER. • VZD-Funktion Für die Nutzerzustimmung zur Verwendung der Versichertendaten durch 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 Telematik-ID oder übernimmt einen fertigen Anfrage-Request aus dem Aufruf 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 Telematik-ID für die Nutzereinwilligung dar. • ZETA-Funktion Die Kommunikation zwischen PoPP-Modul und PoPP-Service Resource Server für die VZD-Abfrage erfolgt über die ZETA-Komponenten. Bei einer Authentifizierung 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). • Service-Funktion Das PoPP-Modul unterstützt die Anwender durch die Bereitstellung von Service-Funktionen. Diese umfassen die Informationen zur Historie der PoPP-Anfragen und eine Favoritenliste von Leistungserbringer Institutionen.
ZETA Client	ZETA Client ist die Zero Trust-Komponente für ein Frontend, 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 Smartphone 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	
Registrierung der Kassen-App mit PoPP-Modul	Die Kassen-Apps auf dem Smartphone der VER, die ein ZETA Client implementieren, müssen am PoPP-Service ZETA Guard registriert werden. Der Prozess der Registrierung ist in [gemSpec_ZETA] beschrieben.

	Im laufenden Prozess prüft dann der PoPP-Service ZETA Guard, ob ein Aufruf von einem bekannten bzw. registrierten ZETA Client kommt und lehnt bei Auffälligkeiten Aufrufe an den PoPP-Service Resource Server ab.
Ausstellen von Access-Token für Request des PoPP-Moduls an den PoPP-Service Resource Server	Bevor das PoPP-Modul Requests an den PoPP-Service Resource Server senden kann, führt ZETA Guard die Client-Authentifizierung durch und stellt dem ZETA Client in der Kassen-App bei erfolgreicher Authentifizierung ein Access-Token aus. Wird von der Kassen-App darüber hinaus die Authentifizierung der VER über die GesundheitsID angefragt, so stößt der ZETA Guard den Authentifizierungsprozess beim jeweiligen sektoralen IDP an. Der ZETA Guard 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.
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 geschickt. ZETA Guard reichert den Request mit 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 Kassen-App zugestellt.
Push-Notifications vom an den Resource Server an das PoPP-Modul	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.
sektorale IDPs	
sektoraler IDP	Die Authentifizierung der VER mit GesundheitsID erfolgt durch den sektoralen IDP der Krankenversicherung der VER. Der sektorale IDP hält einen Datensatz mit Daten zur VER selbst (GesundheitsID) sowie Informationen zu den möglichen Authentisierungsmitteln und den Nutzerpräferenzen. Die Authentifizierung der VER erfolgt dann in Verbindung mit dem Authenticator-Modul auf dem Smartphone

	der VER. 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 Telematik-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:

1. Ist durch die Anbieter-App bereits festgelegt, welcher Telematik-ID ein PoPP-Token benötigt oder muss die Telematik-ID über das PoPP-Modul ermittelt werden?
2. Welche Verfahren zur Ermittlung einer Telematik-ID werden unterstützt?

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

Weitere Einflussfaktoren zur Ausgestaltung der Benutzerführung sind:

1. Ist in der Kassen-App auch das ePA-Frontend der VER (FdV) integriert, so kann für die Authentifizierung der VER über ihre 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 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 Kassen-App), erfolgt die Erstellung des PoPP-Token bei einer Authentifizierung mit GesundheitsID über das PoPP-Modul mit App2App-Kommunikation (App2App Sprung).
3. Wenn eine Anbieter-App 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.

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 VER mit GesundheitsID 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 der VER bzw. Ermittlung der ClientID des sektoralen IDP der Krankenversicherung in der TI-Föderation.
2. Ermittlung der Telematik-ID, für den ein PoPP-Token erstellt werden soll.

3. Wenn für den Anwendungsfall notwendig, Ermittlung der WorkplaceID, für die ein PoPP-Token erstellt werden soll.

4. Aufruf eines HTTP-Request mit den notwendigen Parametern und einer Rücksprung-URL (callbackURL) an einen festgelegten Pfad zur URL (ClientID) des sektoralen IDP der Krankenversicherung in der TI-Föderation um den Ablauf in die App mit PoPP-Modul zu verlagern.

Die WorkplaceID ist eine optionale Information. Sie dient dem Inhaber einer Telematik-ID, welcher ein PoPP-Token erhält, dieses einem konkreten Arbeitsplatz zuzuweisen. Die WorkplaceID kann durch das Primärsystem des Inhabers der Telematik-ID auch wie eine Session-ID verwendet werden, um das letztendlich erzeugte PoPP-Token dann der richtigen VER zuordnen zu können. Dies kann bspw. im Online-Apotheken-Anwendungsfall notwendig werden. Die WorkplaceID darf durch das PoPP-Modul nicht gelöscht oder verändert werden.

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 diese unterschiedlichen Umsetzungsvarianten:

1. PoPP-Modul und Authenticator-Modul des sektoralen IDP sind in eine App der Krankenversicherung integriert, die weitere digitalen Serviceleistungen umfasst.
2. PoPP-Modul und Authenticator-Modul des sektoralen IDP sind in einer eigenen Kassen-App implementiert und der Online Check-in mit GesundheitsID wird aus einer Anbieter-App gestartet (App2App Sprung).

Das Aktivitätsdiagramm „Allgemeiner Ablauf der PoPP-Token-Generierung aus einer Anbieter-App und Authentifizierung über die GesundheitsID“ im Anhang stellt den allgemeinen Ablauf der PoPP-Token-Generierung dar. Dabei wird davon ausgegangen, dass PoPP-Modul, ZETA Client und Authenticator-Modul gemeinsam in einer Kassen-App integriert sind.

Der Start des Online-Check-in erfolgt in einer Anbieter-App oder in der Kassen-App. Eine VER wird über seine GesundheitsID authentifiziert. Die für die PoPP-Token-Erstellung notwendigen Daten, KVNR der VER und IK-Nummer der Krankenversicherung der 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 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 Kassen-App sind API-Schnittstellen. Zwischen den über das Internet erreichbaren Komponenten sind REST-Schnittstellen.

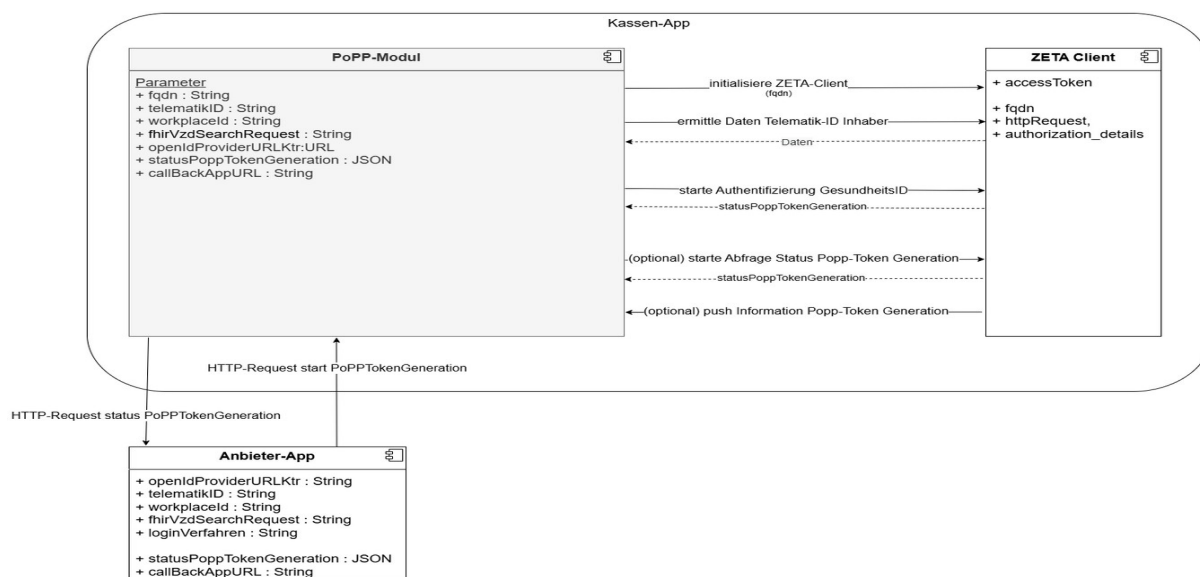


Abbildung 2: Schnittstellen vom und zum PoPP-Modul

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.

3.2.1.1 Initialisierung des ZETA Client

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 Telematik-ID. Dafür wird ein am FHIR-VZD ausführbarer Search-Request als Parameter mit übergeben.

Wurde der Search-Request an der Schnittstelle zum PoPP-Modul aus einer anderen App übergeben, so muss dieser verwendet werden. Ansonsten erstellt das PoPP-Modul einen Search-Request mit der Telematik-ID, für die ein PoPP-Token erstellt werden soll.

Der ZETA-Client führt den HTTP-Request unter Verwendung des Access-Token am PoPP-Service Resource Server aus. Nach Ermittlung der 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.

[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 mit dem Parameter `httpCheckInGIDRequest` wird nur bei der Authentifizierung mit GesundheitsID aufgerufen.

Der 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 der VER über ihre GesundheitsID. Die ZETA-Komponenten prüfen, ob bereits eine Authentifizierung durchgeführt wurde. Ist das nicht der Fall wird die Authentifizierung der VER über den sektoralen IDP der Krankenversicherung der VER angestoßen. Die URL des sektoralen IDP wird an der Schnittstelle ZETA als `authorization_details` gesetzt. Nach erfolgreicher Authentifizierung werden den ZETA-Komponenten die Daten der 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 Resource Server als Response den ZETA-Komponenten zurück. Der ZETA Client gibt das Response-Objekt an das PoPP-Modul weiter.

[`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 mit dem Parameter `httpReadStatusPoPPRequest` kann optional vom PoPP-Modul 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 das Primärsystem den PoPP-Token noch nicht abgerufen hat. In diesem Fall hat der Check-in-Prozess dem PoPP-Modul den Status "pending" zurückgeliefert.

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. Der ZETA Client gibt das Response-Objekt an das PoPP-Modul weiter.

[`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 App2App-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 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 als deeplink bzw. universal link eingerichtet, so führt der Aufruf zum Öffnen der Anbieter-App.

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

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

Die App, die das PoPP-Modul und den ZETA Client integriert, MUSS durchsetzen, dass die Kommunikation zwischen diesen Modulen direkt und unverändert stattfindet. [$\leq$]

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

Das PoPP-Modul MUSS, vor der erstmaligen Durchführung einer Verifikation der Identität der VER sicherstellen, dass die VER, bevor ihre KVNR an den PoPP-Service übermittelt wird, ihre Einwilligung (Consent) zur Nutzung seiner KVNR durch den PoPP-Service erteilt. [$\leq$]

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. [$\leq$]*

A_28488 -PoPP-Modul - Anzeige von VZD-Informationen zu einer Telematik-ID

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 (Anzeigename 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, kann die VER die Anfrage mit geänderten Suchkriterien wiederholen oder den Check-in-Vorgang abbrechen. Der VER ist bei Abbruch ein verständlicher Hinweis zu geben und sie ist dabei aufzufordern, sich an die Institution zu wenden, für die sie den Check-in durchführen wollte. [$\leq$]

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

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 KVN-R 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 die Authentifizierung der VER mit dessen GesundheitsID unterstützen. [≤]

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. [≤]

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 VER eine Funktion bereitstellen, über die diese die Historie der erstellten PoPP-Token einsehen kann. Die Historie MUSS Name, Anschrift und Telematik-ID des Teilnehmers enthalten, für den das PoPP-Token ausgestellt wurde. [≤]

Hinweis 1: Die Anzahl der angezeigten Einträge in der Historie sollte konfigurierbar sein. Empfohlen wird die Darstellung der letzten zehn Ereignisse.

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. [≤]

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

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

- Einen durch das PoPP-Modul im VZD gefundenen Eintrag muss 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.

- VER können zusätzlich einen alternativen Anzeigenamen des Favoriten, wie er in der Favoritenliste angezeigt wird, angeben und diesen ändern.

Hinweis: Es ist unerheblich, was der Auslöser für die VZD-Suche ist (Suche durch Versicherten im PoPP-Modul, 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. [≤]

5.2 Anforderungen an die Kommunikation des PoPP-Modul mit dem ZETA Client

ZETA Client ist die Zero Trust Komponente 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 Smartphone der VER und den Komponenten im Backend, u.a. auch zwischen PoPP-Modul und PoPP-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 dessen Verwendung spezifiziert. Wird das von der gematik bereitgestellte [ZETA-SDK] verwendet, muss das PoPP-Modul die API-Schnittstellen des ZETA-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 zu finden.[<=]

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 MUSS der nach A_28514* erzeugte HTTP-Request als Parameter übergeben werden.[<=]

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. [`<=`]

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 [[Detailierter 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.

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`]).

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-GET-Request `httpCheckInGIDRequest` zum Auslösen der Erstellung eines PoPP-Token nach Authentifizierung des Versicherten mit GesundheitsID erstellen. Dabei MÜSSEN:

- die Telematik-ID des Inhabers der Telematik-ID und
- die WorkplaceID des Inhabers der Telematik-ID (wenn vorhanden)

als Parameter übergeben werden.**[<=]**

Hinweis: Die Schnittstelle ist in `[_PoPP_Service_mobile_CheckIn.yaml]` definiert.

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 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
- die nach A_28502* erzeugten `app_authorization_details`

[<=]

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.**[<=]**

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 dem Inhaber der Telematik-ID zugestellt
- *pending* - PoPP-Token wurde erzeugt und noch nicht zugestellt, da derInhaber der Telematik-ID nicht online ist. Die Zustellung wird weiter versucht.
- *canceled* - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit von 72h abgebrochen

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

Das PoPP-Modul MUSS zur ID des Nachrichten-Datensatzes, welche vom PoPP-ServiceResource Server als Ergebnis des online Check-in Prozesses an das PoPP-Modul zurückgegeben wird, die zugehörigen VZD-Informationen speichern, um der VER den Status des online Check-in zu einer Telematik-ID anzeigen zu können.
[<=]

5.5 Verarbeitung des Ergebnisses der PoPP-Token-Erzeugung

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

Das PoPP-Modul MUSS 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 <Name des Telematik-ID Inhabers> erfolgreich.". Im Fehlerfall könnte die Information lauten: "Die Berechtigung von <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 Nachrichten-Datensatzes muss als Query-Parameter im HTTP-GET Request übergeben werden.[<=]

Hinweis: Die ID des Nachrichten-Datensatzes ist Teil der Ergebnisinformationen der vom PoPP-Modul ausgelösten PoPP-Token-Erstellung. Konnte der PoPP-Datensatz erstellt aber ein PoPP-Token dem Inhaber der 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.

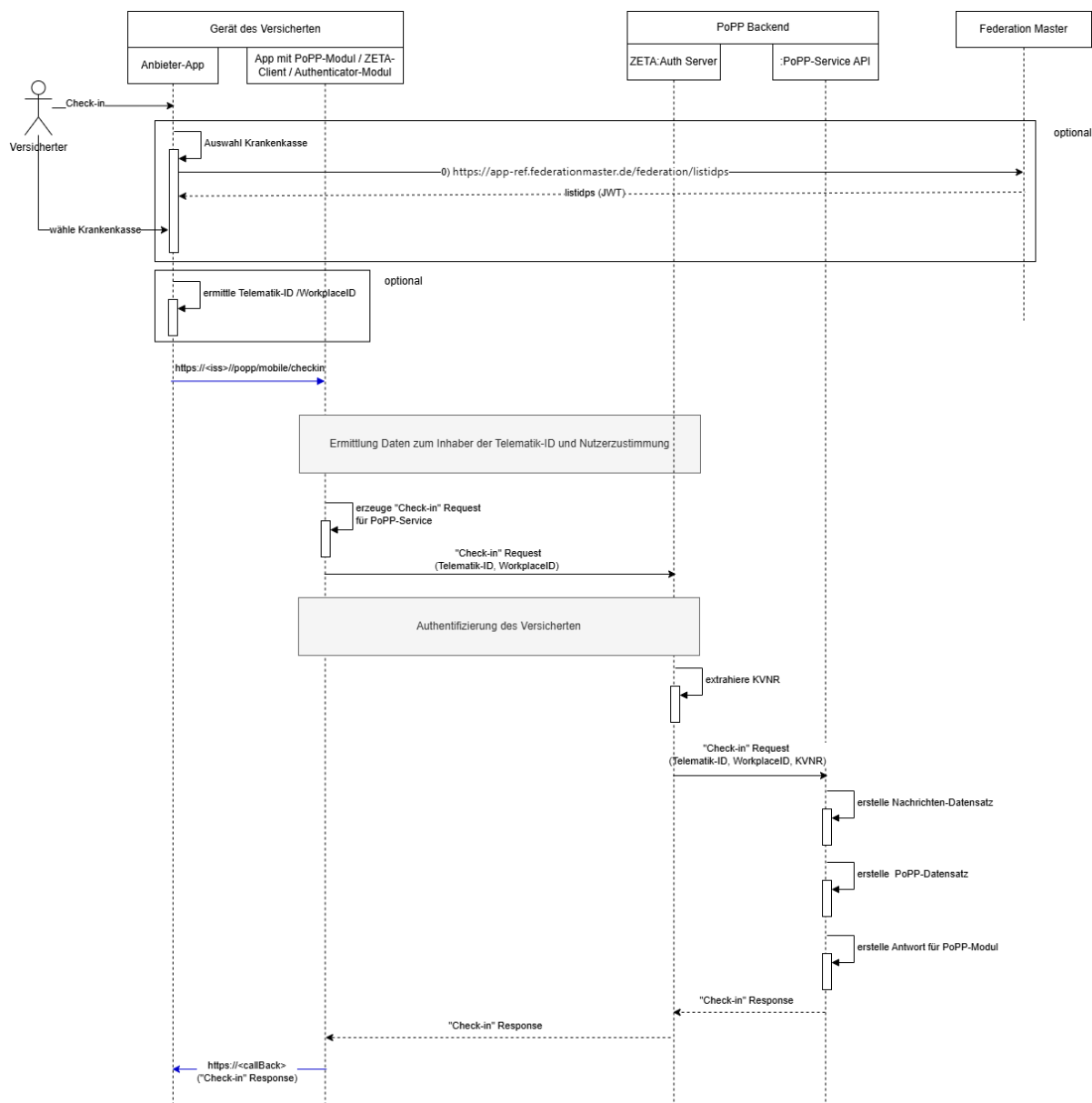


Abbildung 3 : Ablauf "Check-in" aus einer Anbieter-App

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.

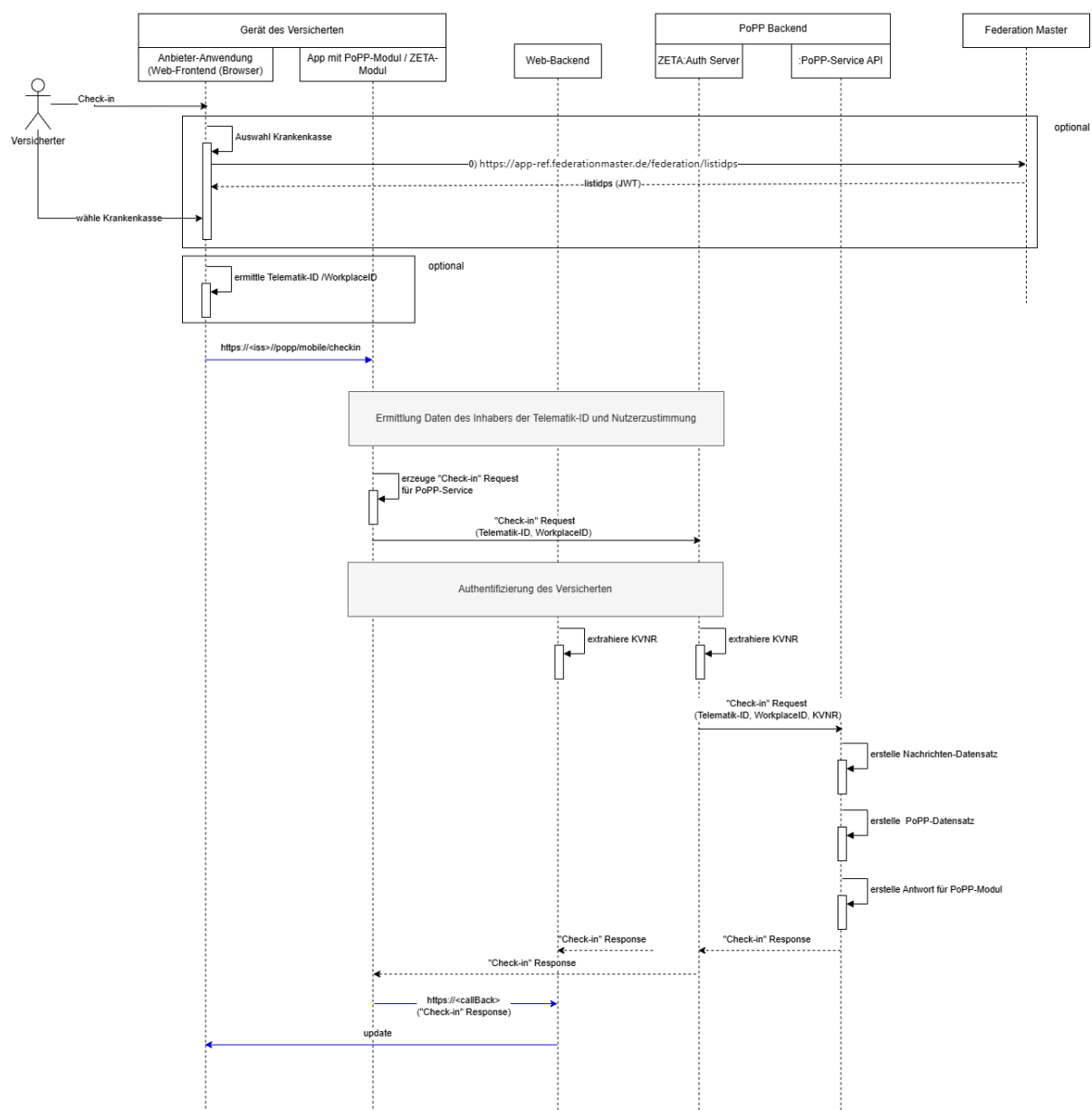


Abbildung 4 : Ablauf "Check-in" aus einer Anbieter-WebApp

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.

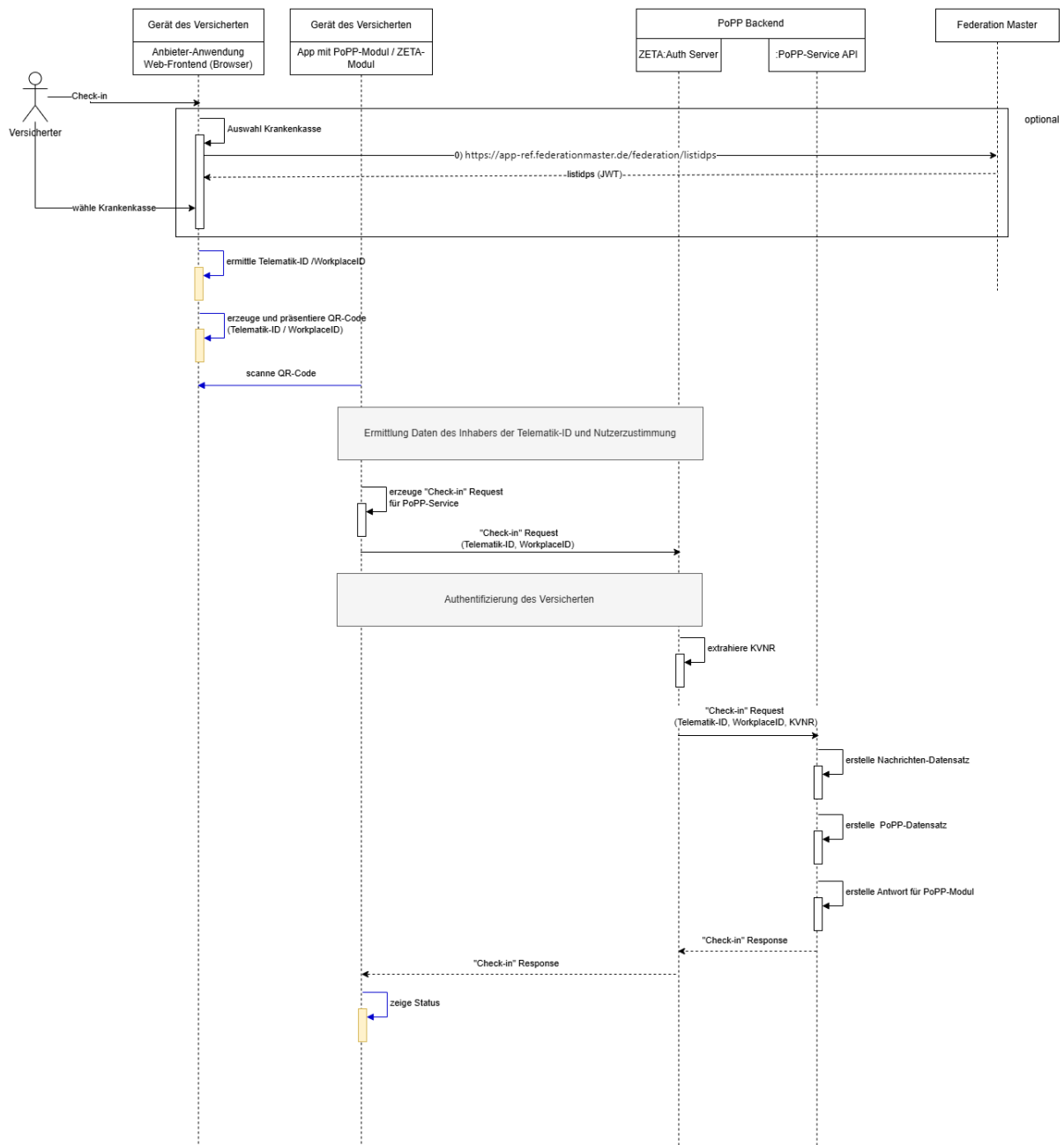


Abbildung 5 : Ablauf "Check-in" mit 2 Geräten

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 Smartphone der VER und des PoPP-ServiceResource Server für das Ausstellen eines PoPP-Token nach Authentifizierung der VER mit GesundheitsID oder Authentifizierung einer eGK über das PoPP-Modul in einer Kassen-App auf dem Smartphone der VER.

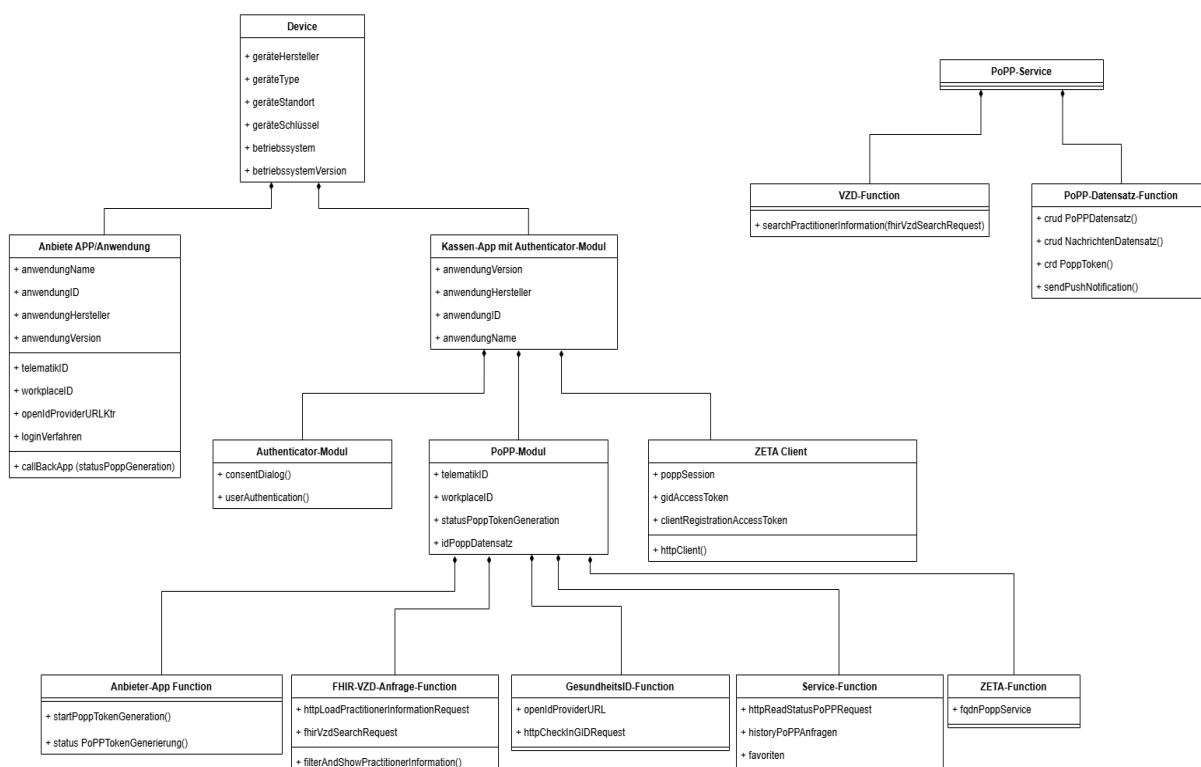


Abbildung 6: Bausteinsicht - Informationsmodell Smartphone der VER und PoPP-Service Resource Server für Online-Anwendungsfälle zur PoPP-Token-Erstellung

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 (Referenzpfad)

Zunächst wird der Referenzpfad beschrieben, der die Umsetzung des Online Check-ins mit PoPP in einer Kassen-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 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).

3. Optionen und Alternativen

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

- 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 Empfehlungen zur Terminologie genannt.

7.1 End-to-End Flow (Referenzpfad)

Der Referenz zeigt die Umsetzung des Online Check-ins mit PoPP in einer Kassen-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 Kassen-App für einen Praxisbesuch mit GesundheitsID.

[Clickdummy <https://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=2535-5273&viewport=597%2C4382%2C0.16&t=6QlfHZKYsaLuHn1P-9&scaling=scale-down&content-scaling=fixed&starting-point-node-id=2535%3A5273&show-proto-sidebar=1>]

Ausgangssituation

Der Referenzpfad umfasst nicht die Ersteinrichtung. Folgende Punkte werden entsprechend vorausgesetzt.

- Kassen-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.
- Telematik-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 Referenzpfad.

	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
Erfassung der 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 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 [Auth-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 eingecheckt

7.2 Anbieter-Apps

Bei einem Online Check-in aus einer Anbieter-App, bei der das PoPP-Modul einer Kassen-App nachgenutzt wird (App2App), sind nicht alle im Referenzpfad aufgeführten Schritte erforderlich. Ausnahmen, wie bspw. Entfall des Flow-Schritts *Erfassung der Telematik-ID*, da eine Telematik-ID direkt in der Anbieter-App hinterlegt werden kann, sind in den jeweiligen Kapiteln benannt.

Clickdummy / Screenshot

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

[Clickdummy <https://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=4822-55496&viewport=597%2C4382%2C0.16&t=6QlfHZKYsaLuHn1P-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 Referenzpfad entfallen die Schritte Check-in und Erfassung der Telematik-ID.
- Der Einstieg sowie der Ausstieg können individuell durch die Anbieter-App festgelegt werden.
- Die App der Krankenversicherung darf weder Anbieter-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 Referenzpfads beschrieben.

7.3.1 Erfassung der 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 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, sollen der VER alternative Varianten zu Erfassung der Telematik-ID zur Verfügung stehen.

Clickdummy/Screenshot

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

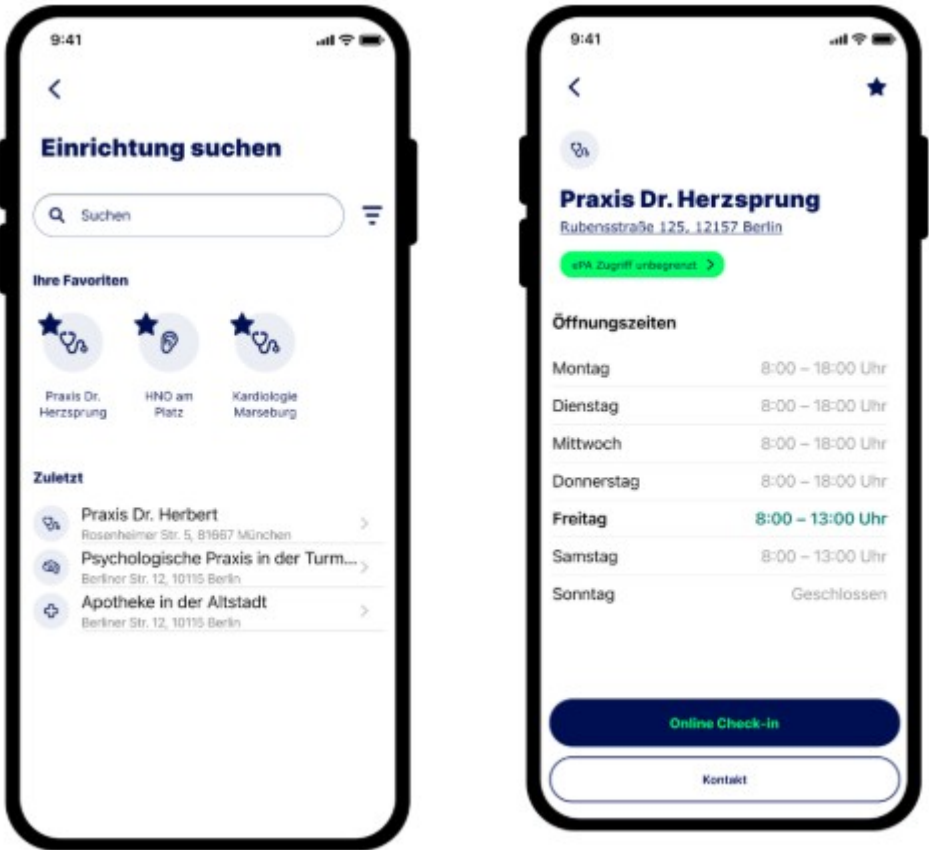


Abbildung 7: Screenshot des Flow-Schritts Erfassung der Telematik-ID mit VZD-Suche, Anzeige einer Historie der letzten Online Check-ins, Favoriten und Detailscreens einer Einrichtung.

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 angelegt wurden
Zuletzt genutzt	Liste der zuletzt durchgeführte Online Check-ins	chronologisch sortiert optional einblenden: weitere verwendete Einrichtungen aus anderen Modulen anzeigen: E-Rezept gesendet, ePA freigeschaltet, Widerspruch eingelegt, ...

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.
Spezialisierung (Allgemeinmedizin, Neurologie, Sprachtherapie, ...)	zeigt nur Einrichtungen, mit der	Auswahl aus Liste und Suchfeld alphabetisch sortiert

	Spezialisierung an	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, 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

7.3.2 Statusinformation

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

Flow-Schritte und Vorgaben

Die Benachrichtigung über den Check-in-Status müssen von der App am PoPP-ServiceResource Server abgefragt und der VER dargestellt werden.

7.3.3 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 Kassen-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<https://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=4729-53558&viewport=597%2C4382%2C0.16&t=6QlfHZKYsaLuHn1P-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 Anbieter-App (App2App) für Videosprechstunden mit GesundheitsID inkl. zusätzlicher Flow-Schritte für die Ersteinrichtung: Auswahl der Kassen-App, Einrichtung GesundheitsID und Zustimmung der VER zur Nutzung des Online-Check-ins.

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 Kassen-App, Einrichtung GesundheitsID und Zustimmung der VER zur Nutzung des Online Check-ins.

[Clickdummy<https://www.figma.com/proto/z8DtEV6loj6d00GD24V3dk/PoPP?page-id=2117%3A20775&node-id=4836-56927&viewport=597%2C4382%2C0.16&t=6QlfHZKYsaLuHn1P-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.

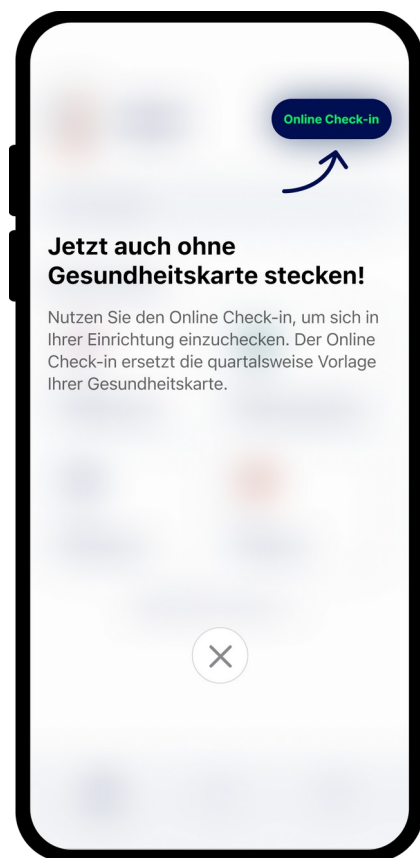


Abbildung 8: Screenshot der auf die neue Online Check-in-Funktion hinweist.
Flow-Schritte und Vorgaben

- 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 Anbieter-App, 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.
- Der Online Check-in muss in der Kassen-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 dienen als Empfehlung.

Thema	Text
Button für PoPP-Modul	Online Check-in
Einwilligung [Auth-Modul] zur Übertragung KVNR, 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]
Abfrage Einrichtung GesundheitsID	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]
Primärsystem des Inhabers der Telematik-ID noch nicht erreichbar	Online Check-in in Bearbeitung Der Check-in wurde übermittelt und wird derzeit bearbeitet. Sie werden benachrichtigt, sobald der Check-in in der Einrichtung erfolgreich abgeschlossen

	wurde, vorausgesetzt, Push-Benachrichtigungen sind aktiviert. [Ok]
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]

1135

1136

8 Anhang A - Verzeichnisse

8.1 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

8.2 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-Anwendung	Eine Drittanbieter-Anwendung ist eine Anwendung im Kontext des Gesundheitswesens, die Personen digitale Gesundheitsdienste bereitstellt und nicht von einer gesetzlichen oder privaten Krankenversicherung angeboten wird. Drittanbieter-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. Kassen-Apps sind keine Anbieter-Apps im Sinne dieser Spezifikation. Drittanbieter-Anwendungen können als native mobile Anwendungen oder browserbasierte Webanwendungen umgesetzt sein.
Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.
GesundheitsID	Die GesundheitsID ist die digitale Identität im Gesundheitswesen für 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.
Kassen-App	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-ServiceResource Server verantwortlich ist.
PoPP-Modul	Eine Komponente von Kassen-App, welche für Online-Anwendungsfälle die Kommunikation mit dem PoPP-ServiceResource Server übernimmt. Das PoPP-Modul initiiert

	die Authentifizierung einer VER mit GesundheitsID.
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: der VER, bzw. dessen elektronischer Gesundheitskarte (eGK), und dem Inhaber einer Telematik-ID.
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 einer berechtigten VER und einer authentifizierten Leistungserbringerinstitution. Diese Verbindung autorisiert den Zugriff auf anwendungsbezogene Versicherungsdaten über die Telematikinfrastruktur (TI) Anwendungen Ein Versorgungskontext besteht, wenn ein Leistungserbringer und eine 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 der VER mittels digitaler Identität und ist auch bei telemedizinischen Anwendungen relevant
VER (im Kontext PoPP)	Im Kontext von PoPP ist "VER" eine Person, die ihre Identität im Gesundheitswesen einbringt, um einen Versorgungskontext mit dem Inhaber einer 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. Die VER trägt zur Erstellung des Versorgungskontexts bei, indem sie ihre Krankenversicherungsnummer (KVNR) bereitstellt und sich

	entweder über die GesundheitsID authentisiert, persönlich oder durch einen Vertreter mit der eGK beim Inhaber der Telematik-ID vorstellt, oder mobil per eGK bei beim Inhaber der Telematik-ID anmeldet.
WorkplaceID	Die WorkplaceID ist ein Identifikator, um ein erstelltes PoPP-Token einen bestimmten Arbeitsplatz (z.B. in einem PVS) oder einem bestimmten Prozess (z.B. bei Online-Apotheken) zuordnen zu können.

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

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.

8.3 Abbildungsverzeichnis

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.....	9
Abbildung 2: Schnittstellen vom und zum PoPP-Modul.....	17
Abbildung 3 : Ablauf "Check-in" aus einer Anbieter-App.....	32
Abbildung 4 : Abbildung : Ablauf "Check-in" aus einer Anbieter-WebApp.....	33
Abbildung 5 : Abbildung : Ablauf "Check-in" mit 2 Geräten.....	34
Abbildung 6: Bausteinsicht - Informationsmodell Smartphone der VER und PoPP-Service Resource Server für Online-Anwendungsfälle zur PoPP-Token-Erstellung.....	37
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.....	41
Abbildung 8: Screenshot der auf die neue Online Check-in-Funktion hinweist.....	45
Abbildung 9: Laufzeitsicht - Überblick Auslösung und Ablauf der PoPP-Token-Generierung	56
Abbildung 10: Laufzeitsicht - Initialisierung ZETA Client.....	57
Abbildung 11: Laufzeitsicht - Ermittlung Suchkriterien, VZD-Datenabruf mit Suchkriterien und Einwilligung in die Datenweitergabe.....	58

1163	Abbildung 12: Laufzeitsicht - Ablauf der PoPP-Token-Generierung aus einer Anbieter-App	
1164	oder Kassen-App und Authentifizierung über die GesundheitsID.....	60
1165	Abbildung 13: Laufzeitsicht - Detaillierter Ablauf des Online Check-in aus einer Anbieter-	
1166	App oder Kassen-App und Authentifizierung über die GesundheitsID.....	61
1167	Abbildung 14: Laufzeitsicht - Ablauf Einlösen von PoPP-Token aus dem Online Check-in	
1168	durch ein Primärsystem.....	77
1169	Abbildung 15: Detaillierter Ablauf zum PoPP-Token-Abruf aus einem Primärsystem.....	78
1170	Abbildung 16: Laufzeitsicht - Ablauf der Statusabfrage zur PoPP-Token-Generierung aus	
1171	einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID.82	
1172	Abbildung 17: Laufzeitsicht - Detaillierter Ablauf zu Ermittlung des Status zum Online	
1173	Check-in aus einer Anbieter-App oder Kassen-App und Authentifizierung über die	
1174	GesundheitsID.....	84
1175		

1176 **8.4 Tabellenverzeichnis**

1177	Tabelle 1: Beschreibung der wesentlichen Komponenten für die Online-Anwendungsfälle	
1178	zur PoPP-Token-Erstellung.....	10
1179	Tabelle 2: Tab_PoPP_Modul_Payload_stat_QRCode]: Payload QR-Code.....	26
1180	Tabelle 3: Flow-Schritte und Vorgaben an den Online Check-in im Referenzpfad.....	39
1181	Tabelle 4: Filterkriterien für Suche im VZD - Einträgen auswählen/suchen.....	42
1182	Tabelle 5: Filterkriterien für Suche im VZD - Einträge eingrenzen.....	42
1183	Tabelle 6: Filterkriterien für Suche im VZD - Einträge darstellen.....	43
1184	Tabelle 7: Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung der	
1185	VERüber ihre GesundheitsID.....	62
1186	Tabelle 8: Beschreibung der Schritte zur Generierung und zum Abruf von PoPP-Token aus	
1187	dem Online Check-in.....	79
1188	Tabelle 9: Beschreibung der Schritte zum Abruf des Status eines zuvor durchgeführten	
1189	Online Check-in Prozesses bei Authentifizierung der VER über ihre GesundheitsID...84	
1190		

1191 **8.5 Referenzierte Dokumente**

1192 **8.5.1 Dokumente der gematik**

1193 Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument
1194 referenzierten Dokumente der gematik zur Telematikinfrastruktur.

1195

[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 fürPushNotification 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-ilm-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-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 den Aufruf des PoPP-Modul aus Anbieter-Apps todo: Anpassung github-Link
[_PoPP_Service_mobile_Checkin_HID.yaml]	OpenAPI Schnittstellenspezifikation für

In.yaml]	• den Aufruf des VZD-Suche am PoPP-ServiceResource Server • den Aufruf zum Anlegen eines PoPP-Datensatzes am PoPP-ServiceResource Server • den Aufruf zur Statusabfrage am PoPP-ServiceResource Server todo: Anpassung github-Link
[I_PoPP_Service_mobile_Token_Generation.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 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

1196

8.5.2 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]	OpenID Federation Standard

Federation 1.0]	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/
[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.

1197

9 Anhang - Ablaufbeschreibungen

9.1 Gesamtüberblick Auslösung und Ablauf der PoPP-Token-Generierung

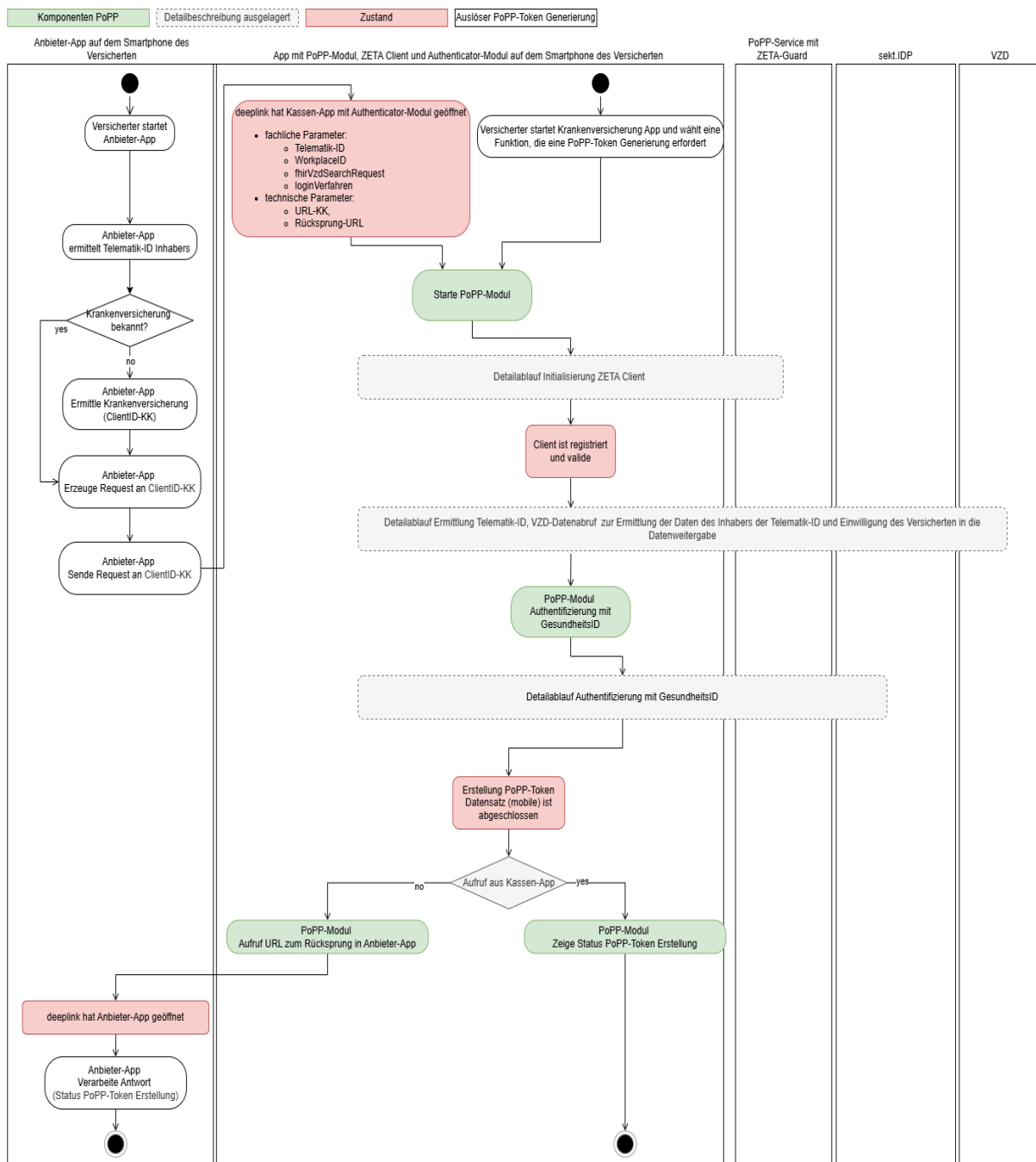


Abbildung 9: Laufzeitsicht - Überblick Auslösung und Ablauf der PoPP-Token-Generierung

9.2 Detailablauf "ZETA Client Initialisierung"

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

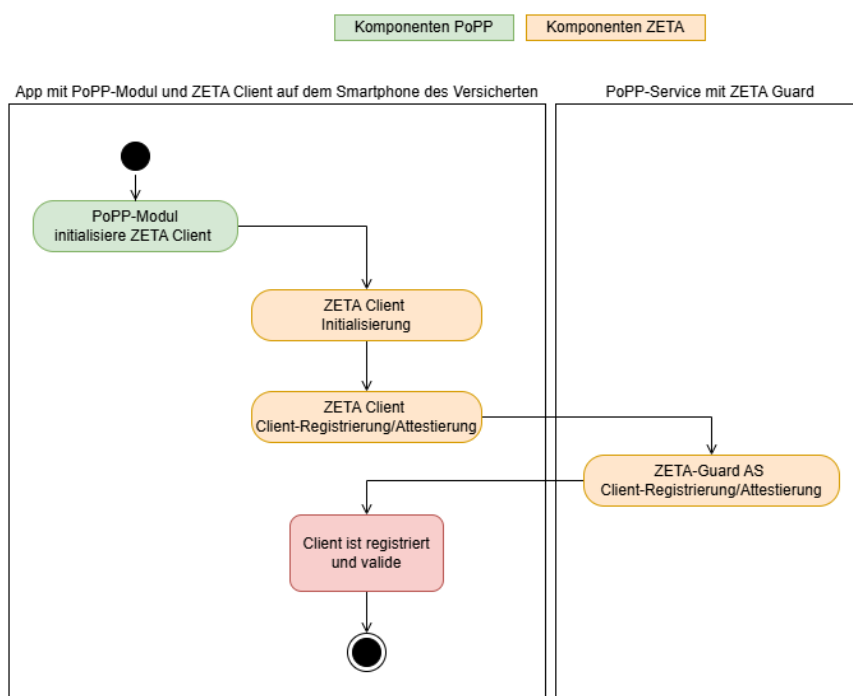


Abbildung 10: Laufzeitsicht - Initialisierung ZETA Client

9.3 Detailablauf "FHIR-VZD-Anfrage"

Die Abfrage des FHIR-VZD nach Daten zu Inhabern von Telematik-IDs muss die Anwendungsfälle unterstützen:

- Die Telematik-ID ist bekannt und es müssen für die Einwilligung der VER weitere Daten 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, Adresse/PLZ des Inhabers der Telematik-ID, u.ä.) durchgeführt werden. Als Ergebnis werden alle Daten des Inhabers der Telematik-ID inklusive der Telematik-ID selbst für die Einwilligung der 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 vorhandenen Daten zum Inhaber der Telematik-ID. Der FHIR-VZD Search Request wird über die ZETA-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 der VER den Datensatz zur Erteilung der Einwilligung in die Datennutzung an.

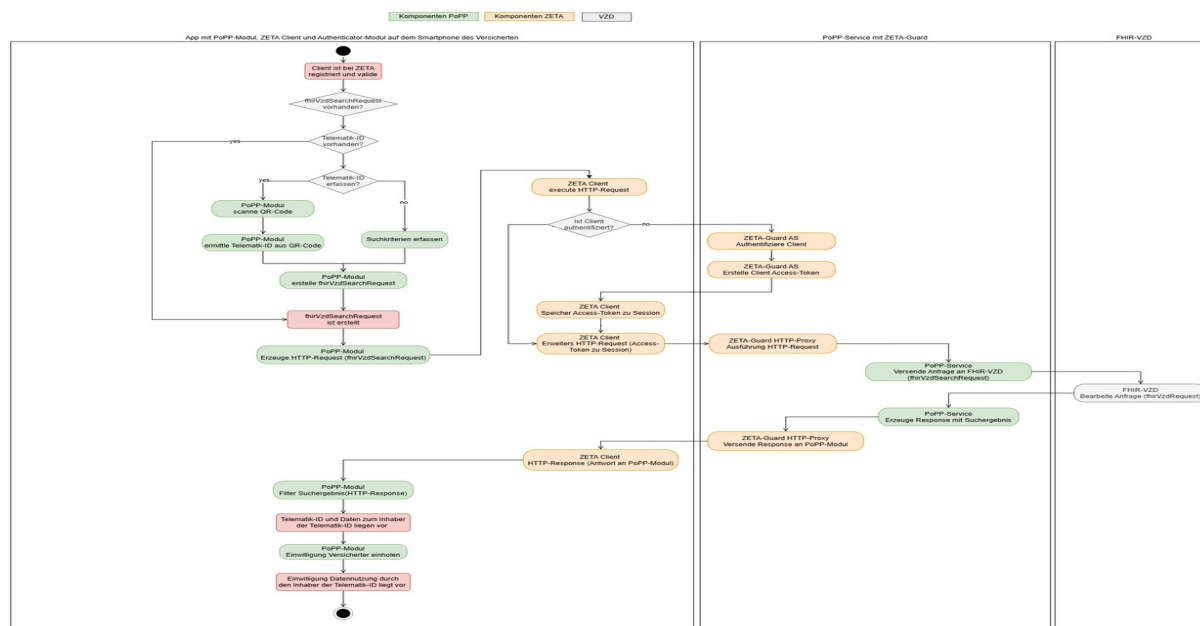


Abbildung 11: Laufzeitsicht - Ermittlung Suchkriterien, VZD-Datenabruf mit Suchkriterien und Einwilligung in die Datenweitergabe

9.4 Detailablauf "Authentifizierung mit GesundheitsID"

9.4.1 Allgemeiner Ablauf der PoPP-Token-Generierung durch Authentifizierung VER über ihre GesundheitsID

Spezifikation PoPP (Proof of Patient Presence) - Modul

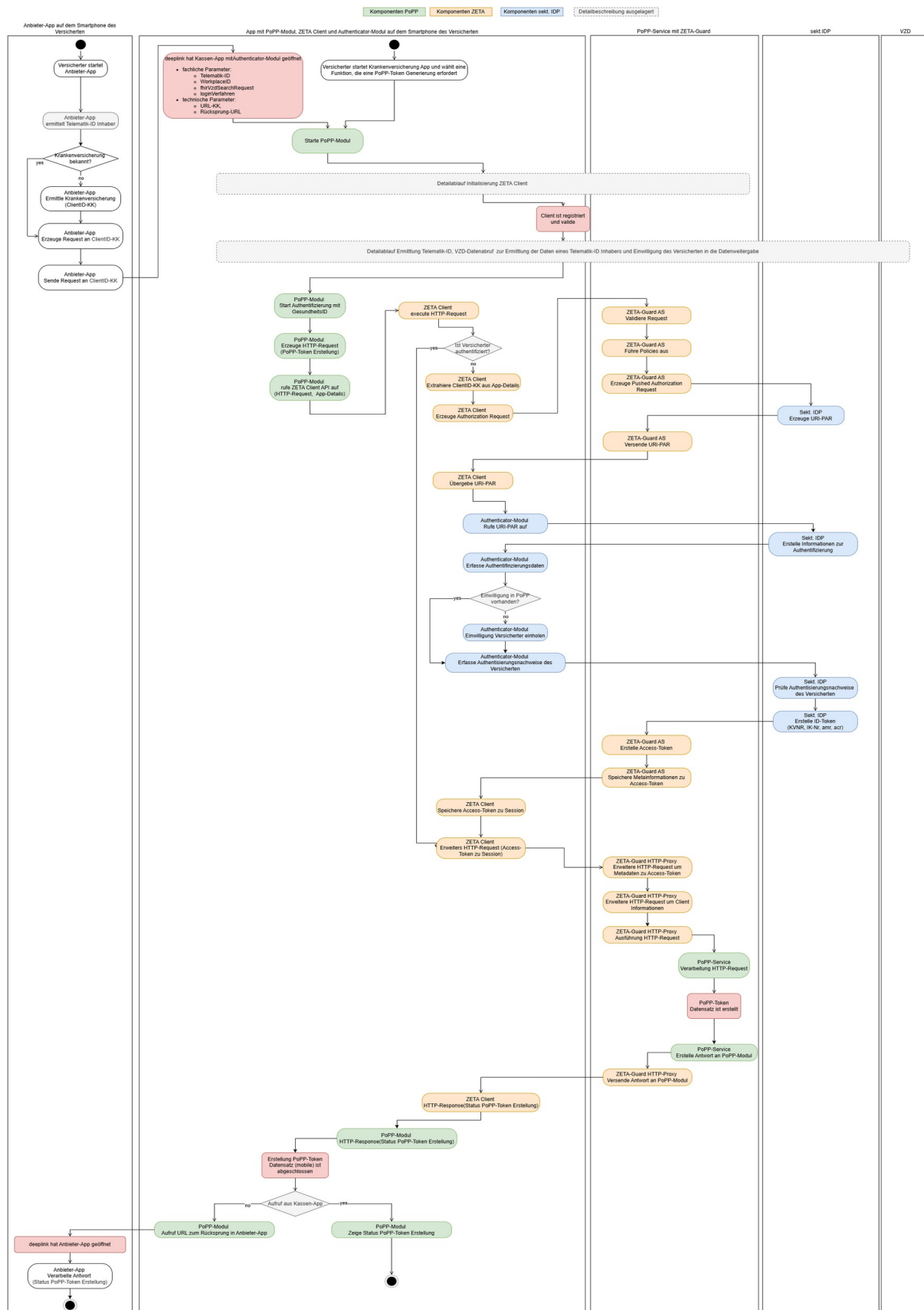


Abbildung 12: Laufzeitsicht - Ablauf der PoPP-Token-Generierung aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID

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

Das Sequenzdiagramm "Detaillierter Ablauf des Online Check-in aus einer Anbieter-App und Authentifizierung über die GesundheitsID" stellt den vollständigen Ablauf für den Fall dar, dass eine VER über ihre 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.

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 Aufruf 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

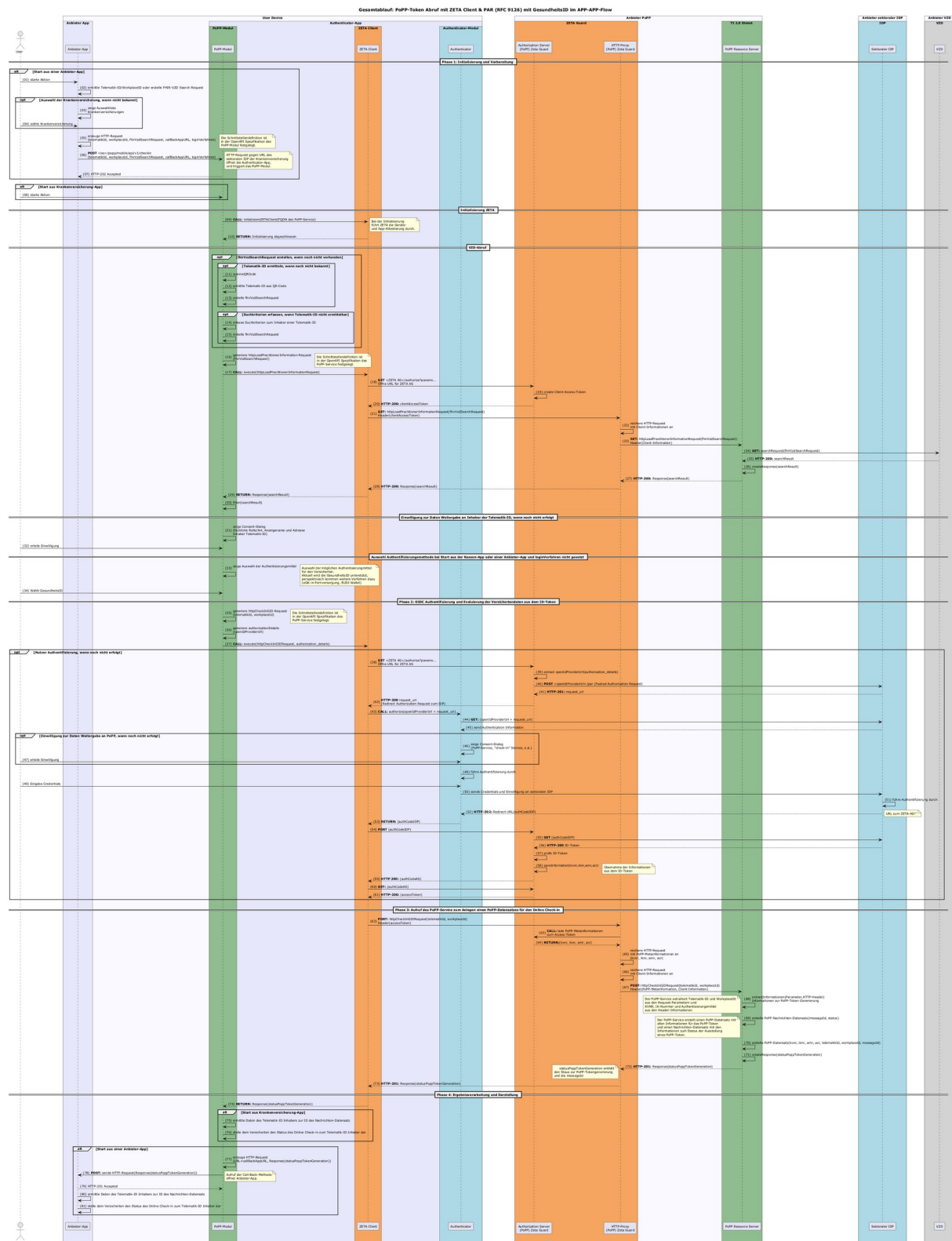


Abbildung 13: Laufzeitsicht - Detaillierter Ablauf des Online Check-in aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID

9.4.3 Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung VER mit ihrer GesundheitsID

Die Tabelle "Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung der VER über 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 7: Beschreibung der Schritte zur PoPP-Token-Erstellung bei Authentifizierung der VER über ihre GesundheitsID

	Schritt	Beschreibung
	Start Alternative - Auslösen des Online Check-in aus einer Anbieter-App	
1	Aktion starten	Die VER 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 zum Inhaber einer Telematik-ID	In der Anbieter-App werden Informationen 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. • Bei der Anbieter-App (online Apotheke) ist die Telematik-ID bekannt. • Bei der Anbieter-App zur Vermittlung einer Videosprechstunde bietet die App entsprechende Auswahlfunktionen an. • Eine Praxis kann die Telematik-ID als QR-Code bereitstellen. • Es werden Suchkriterien erfasst, mit denen eine FHIR-VZD Anfrage durchgeführt werden kann
3	zeige Auswahlliste Krankenversicherungen	Die Liste der möglichen Krankenversicherungen kann am idp-list-Endpoint 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	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-POST Request(telematikId, workplaceId, fhirVzdSearchRequest, callBackAppURL, loginVerfahren)	Die Anbieter-App erzeugt einen HTTP-Request und übergibt als Parameter • <i>telematikId</i> -ist die Telematik-ID eines Telematik-ID Inhabers, für die ein PoPP-Token erstellt werden soll, • <i>workplaceId (optional)</i>-ist eine WorkplaceID, wenn das PoPP-Token einem bestimmten Arbeitsplatz im Primärsystem des Inhabers der Telematik-ID zugeordnet werden soll, • <i>fhirVzdSearchRequest (optional)</i>- Search-Request für den Aufruf der FHIR-VZD API, um Daten zum Inhaber der Telematik-ID zu erhalten • <i>callBackURL (Rücksprung-URL)</i>- welche das PoPP-Modul aufrufen muss, wenn die Erzeugung eines PoPP-Token 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(telematikId, workplaceId, fhirVzdSearchRequest, callBackAppURL, loginVerfahren)	Die Anbieter-App sendet den HTTP-Request entsprechend der OpenAPI Spezifikation [I_PoPP_Modul_mobile_CheckIn.yaml] 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.
Ende Alternative - Auslösen des Online Check-in aus einer Anbieter-App		
Start Alternative - Auslösen des Online Check-in aus einer Kassen-App		
8	Aktion starten	Die VER wählt eine Funktion in der Kassen-App (z.B. "Praxis Check-in")
Ende Alternative - Auslösen des Online Check-in aus einer Kassen-App		
9	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.
10	RETURN: Initialisierung abgeschlossen	Nach Abschluss der Initialisierung kehrt die Anwendung in das PoPP-Modul zurück.
FHIR-VZD Search Request erzeugen, wenn nicht bereits vorhanden		
Alternative: Telematik-ID ermitteln		
1	scanneQRCode	Ist zu diesem Zeitpunkt noch keine

1		Telematik-ID erfasst, für die ein PoPP-Token erstellt werden soll, so muss dies in diesem Schritt erfolgen. Bei "Check-in" in einer Praxis wäre das z.B. das Scannen eines QR-Codes.
1 2	ermittle Telematik-ID aus QR-Code	Der QR-Code wird geprüft und die Telematik-ID sowie ggf. eine WorkplaceID aus dem QR-Code extrahiert.
1 3	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.
Alternative: Suchkriterien erfassen		
1 4	erfasse Suchkriterien zum Inhaber einer Telematik-ID	In diesem Schritt erfolgt die Erfassung der Suchkriterien zum Inhaber einer Telematik-ID. Suchkriterien können z.B. bestehen aus bekannten Angaben zum Inhaber der Telematik-ID (Name, Fachrichtung, PLZ) und zusätzlichen Informationen (Umkreissuche).
1 5	erstelle FHIR-VZD Search Request	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.
Der FHIR-VZD-Abruf wird über den ZETA-Ablauf an den PoPP-ServiceResource Server delegiert		
1 6	generiere httpLoadPractitionerInformation-Request(fhirVzdSearchRequest)	Das PoPP-Modul generiert einen HTTP-Request (httpLoadPractitionerInformationRequest) mit dem erstellten FHIR-VZD-Request als Parameter entsprechend der Schnittstellendefinition [I_PoPP_Service_mobile_CheckIn.yaml].
1 7	CALL: execute(httpLoadPractitionerInformationRequest)	Das PoPP-Modul ruft die Schnittstellen-Methode zur Ausführung des HTTP-Request am ZETA Client [gemSpec_ZETA] mit

		dem erzeugten HTTP-Request (<i>httpLoadPractitionerInformationRequest</i>) als Parameter 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: <i>clientAccessToken</i>	Der ZETA Guard antwortet dem ZETA-Client mit dem Access-Token.
2 1	GET: <i>httpLoadPractitionerInformationRequest(fhirVzdSearchRequest)</i> Header(<i>clientAccessToken</i>)	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 reichert den PoPP-Modul <i>httpLoadPractitionerInformationRequest</i> um Informationen zum Client (ZETA Client) aus der ZETA Guard Client-Registry an.
2 3	GET: <i>httpLoadPractitionerInformationRequest(fhirVzdSearchRequest)</i> Header(<i>Client-Information</i>)	Der ZETA Guard HTTP-Proxy sendet den <i>httpLoadPractitionerInformationRequest</i> mit den erweiterten Client-Informationen an den PoPP-Service Resource Server.
2 4	GET: <i>searchRequest(fhirVzdSearchRequest)</i>	Der PoPP-Service Resource Server führt den FHIR-VZD Search Request aus.
2 5	HTTP-200: <i>searchResult</i>	Der FHIR-VZD liefert dem PoPP-Service Resource Server das Anfrageergebnis zurück.
2 6	<i>createResponse(searchResult)</i>	Der PoPP-Service Resource Server erzeugt eine Response mit dem Anfrageergebnis.
2 7	HTTP-200: <i>Response(searchResult)</i>	Der PoPP-Service Resource Server 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	RETURN: Response(searchResult)	Der ZETA Client gibt als Antwort auf die Ausführungsanweisung des <i>httpLoadPractitionerInformationRequest</i> das Response-Objekt mit dem Anfrageergebnis an das PoPP-Modul zurück.
3 0	Filter Anzeigeergebnis(searchResult)	Das PoPP-Modul filtert das Anzeigeergebnis, dass das nur für die VER relevante Daten zur Anzeige im Einwilligungsdialo kommen.
3 1	Zeige Consent-Dialog (Organisationsname, Organisationsanschrift)	Das PoPP-Modul zeigt der VER einen Dialog mit den Detailinformationen zum Inhaber der Telematik-ID Organisation (Name, Adresse, ggf. weitere Informationen) an und bittet um Zugriffserlaubnis auf KVNR und IK-Nummer für den Inhaber der Telematik-ID.
3 2	Erteilt Einwilligung	Die VER erteilt ihre Einwilligung. Verweigert die VER ihre Einwilligung, so wird der Prozess abgebrochen.
3 3	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.
3 4	Wählt GesundheitsID	Die VER wählt "GesundheitsID" aus.
Ablauf Erzeugung eines PoPP-Datensatzes		
3 5	generiere httpCheckInGID-Request(telematikId, workplaceId)	Das PoPP-Modul generiert HTTP-CheckInGID-Request entsprechend der Schnittstellendefinition [I_PoPP_Service_mobile_CheckIn.ya

		ml]. Telematik-ID und WorkplaceID sind Parameter des HTTP-CheckInGID-Requests.
3 6	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>
3 7	CALL: execute(httpCheckInGIDRequest, authorizationDetails)	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.
3 8	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.
3 9	extract openIdProviderUrl(authorization_details)	Da die Authentisierungsmethode "GesundheitsID" ist, extrahiert der ZETA Guard Authorization-Server aus den Authorization-Details die openIdProviderUrl des sektoralen IDP der gewählten Krankenversicherung.
4 0	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 1	HTTP-201: request_uri	Der sektorale IDP der gewählten Krankenversicherung erzeugt eine request_uri (URI-PAR) und sendet diese an den ZETA Guard Authorization-Server zurück.
4 2	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 3	CALL: authorize(openIdProviderUrl+request_uri)	Der ZETA Client ruft eine Methode am Authenticator-Modul mit der request_uri als Parameter auf.
4 4	GET: (openIdProviderUrl+request_uri)	Die Kassen-App mit Authenticator-Modul führt den Authorization Request (redirect_uri) an den Auth-Endpunkt des sektoralen IDP aus.
4 5	send Authentication Information	Der sektorale IDP übermittelt dem Authenticator-Modul der Kassen-App die notwendigen Informationen zur Authentifizierung (z.B. Status Gerätebindung, Einwilligungen, Authentisierungsmittel).
4 6	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 7	Erteilt Einwilligung	Die VER erteilt seine Einwilligung zur Nutzung der KVNR durch den PoPP-Service. Verweigert die VER die Einwilligung, wird der Prozess hier abgebrochen.
4 8	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.
4 9	Eingabe Authentisierungsnachweise	Die VER erfasst die Authentisierungsnachweise zur ausgewählten Authentifizierungsmethode.
5	Sende Authentisierungsnachweise und	Das Authenticator-Modul sendet

0	Einwilligung an IDP Server	Einwilligung und Authentisierungsnachweis an den sekt. IDP.
5 1	führe Authentifizierung durch	Der sektorale IDP authentifiziert die VER auf Basis der Authentisierungsnachweise und der ausgewählten Authentifizierungsmethode.
5 2	HTTP-302: Redirect-URL(authCodeIDP)	Der sektorale IDP erstellt einen Authorization-Code (authCodeIDP) und antwortet dem Authenticator-Modul mit einem Redirect auf die URL zum ZETA-AS
5 3	RETURN: authCodeIDP	Das Authenticator-Modul antwortet dem Aufruf des ZETA Client und übergibt den authCodeIDP
5 4	POST: (authCodeIDP)	Der ZETA Client ruft die URL des ZETA-AS auf und übermittelt so den Authorization-Code für den Token Abruf am sektoralen IDP an den ZETA-AS.
5 5	GET (authCodeIDP)	ZETA-AS ruft den ID-Token durch Übergabe des Authorization-Code (authCodeIDP) am Token-Endpoint des sektoralen IDP ab.
5 6	HTTP-200: ID-Token	Der sektorale IDP antwortet dem ZETA Guard Authorization-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 Vertrauensniveau, auf dem die Authentifizierung erfolgt ist.
5 7	prüfe ID-Token	Der ZETA Guard Authorization-Server prüft die Signatur des ID-Token und entschlüsselt den Inhalt.

5 8	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.
5 9	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 0	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 1	HTTP-200: accessToken	Der ZETA Guard Authorization-Server antwortet dem ZETA Client mit finalem Access-Token.
6 2	GET: httpCheckInGIDRequest(accessToken)	Der ZETA Client ruft am ZETA Guard HTTP-Proxy den ursprünglichem HTTP-CheckInGID-Request mit dem Access-Token auf.
6 3	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 4	RETURN: (kvnr, iknr, amr, acr)	Der ZETA Guard Authorization-Server antwortet dem ZETA Guard HTTP-Proxy mit den Daten.
6 5	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 6	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	GET: httpCheckInGIDRequest(telematikId,	Der ZETA Guard HTTP-Proxy ruft

7	workplaceld) Header(accessToken, PoPP-Metainformation, Client-Information)	am PoPP-Service Resource Server vom ZETA Guard HTTP-Proxy erweiterten HTTP-CheckInGID- Request auf.
6 8	extractInformationen(appDetail, HTTP-Header)	Der PoPP-Service Resource Server extrahiert: • aus dem Request-Header • die PoPP-Metainformationen zur VER (KVNR, IK-Nummer) • die Informationen zur Authentifizierung (amr, acr) • Informationen zum aufrufenden Client • aus den Request Parametern • Telematik-ID, Workplaceld des Inhabers der Telematik- ID
6 9	erstelle PoPP-Nachrichten- Datensatz(messageld, 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 weiter versucht. • canceled - Prozess wurde aufgrund eines Fehlers oder wegen Überschreitung der maximalen Wartezeit (z.B. 72h) abgebrochen
7 0	erstelle PoPP-Datensatz(kvnr, iknr, amr, acr, telematikId, workplaceld, messageld)	Der PoPP-Service Resource Server legt einen PoPP-Datensatz mit allen relevanten Informationen für die

		Generierung eines PoPP-Token an.
7 1	createResponse(statusPoppTokenGeneration)	Der PoPP-Service Resource Server erstellt die Antwort an das PoPP-Modul • <i>messageld</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
7 2	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>messageld</i>), welcher die Daten und den Status zur PoPP-Token-Generierung enthält.

7 3	HTTP-201: Response(statusPoPPTokenGeneration)	Der 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 sowie dem Identifier zum Nachrichten-Datensatz (<i>messageId</i>), welcher die Daten und den Status zur PoPP-Token-Generierung enthält.
7 4	RETURN: Response(statusPoPPTokenGeneration)	Der ZETA Client gibt als Antwort auf die Ausführungsanweisung des <i>httpCheckInGIDRequest</i> das Response-Objekt mit dem Status zur PoPP-Token Generierung an das PoPP-Modul zurück.
Start Alternative - Ergebnisverarbeitung des Online Check-in Prozess in der Kassen-App		
7 5	ermittle Daten des Telematik-ID Inhabers zur ID des Nachrichten-Datensatz	Das PoPP-Modul ermittelt aus der ID Nachrichten-Datensatzes die Daten des Telematik-ID Inhabers des ursprünglichen Online Check-in Prozesses.
7 6	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 - Ergebnisverarbeitung des Online Check-in Prozess in der Kassen-App		

Start Alternative - Ergebnisverarbeitung des Online Check-in Prozess in der Anbieter-App		
7 7	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.
7 8	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.
7 9	sende HTTP-ACCEPTED	Die Anbieter-App bestätigt den Empfang mit einem HTTP-ACCEPTED.
8 0	ermittle Daten des Telematik-ID Inhabers zur ID des Nachrichten-Datensatz	Das PoPP-Modul ermittelt aus der ID Nachrichten-Datensatzes die Daten des Telematik-ID-Inhabers des ursprünglichen Online Check-in Prozesses.
8 1	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 - Ergebnisverarbeitung des Online Check-in Prozess in der Anbieter-App		

1272

1273 9.5 Ablauf "PoPP-Token-Abruf aus einem Primärsystem"

1274 9.5.1 Übersicht zum Ablauf des PoPP-Token-Abruf aus einem 1275 Primärsystem

1276 Ist das Primärsystem online und hat sich der Telematik-ID-Inhaber mit seiner SM(C)-B
1277 authentifiziert, kann der Abruf der PoPP-Token aus dem Online Check-in durchgeführt
1278 werden.

1279 Der PoPP-Service Resource Server ermittelt alle PoPP-Datensätze zur relevanten
1280 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.

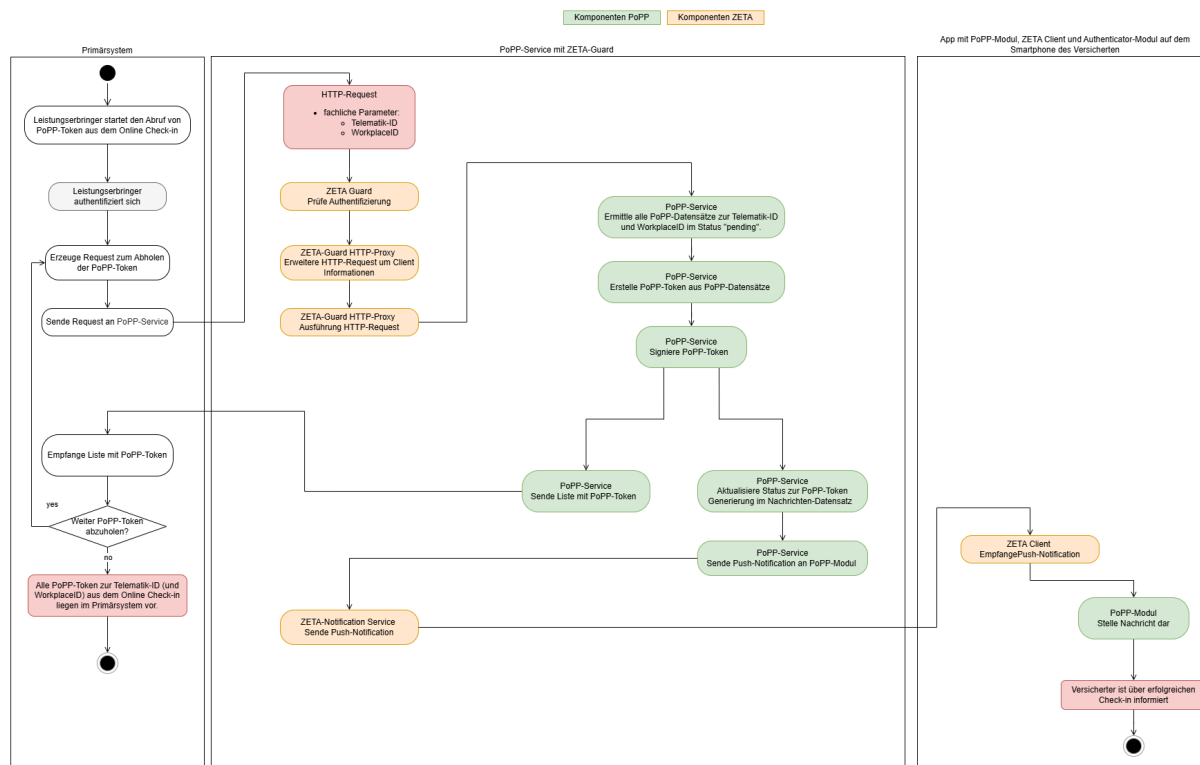


Abbildung 14: Laufzeitsicht - Ablauf Einlösen von PoPP-Token aus dem Online Check-in durch ein Primärsystem

9.5.2 Detaillierter Ablauf zum PoPP-Token-Abruf aus einem Primärsystem

Das Sequenzdiagramm "Detaillierter Ablauf zum PoPP-Token-Abruf aus einem Primärsystem" stellt den vollständigen Ablauf des Abrufs 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 VER informiert, ob ihr Check-in Prozess erfolgreich war oder abgebrochen wurde.

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

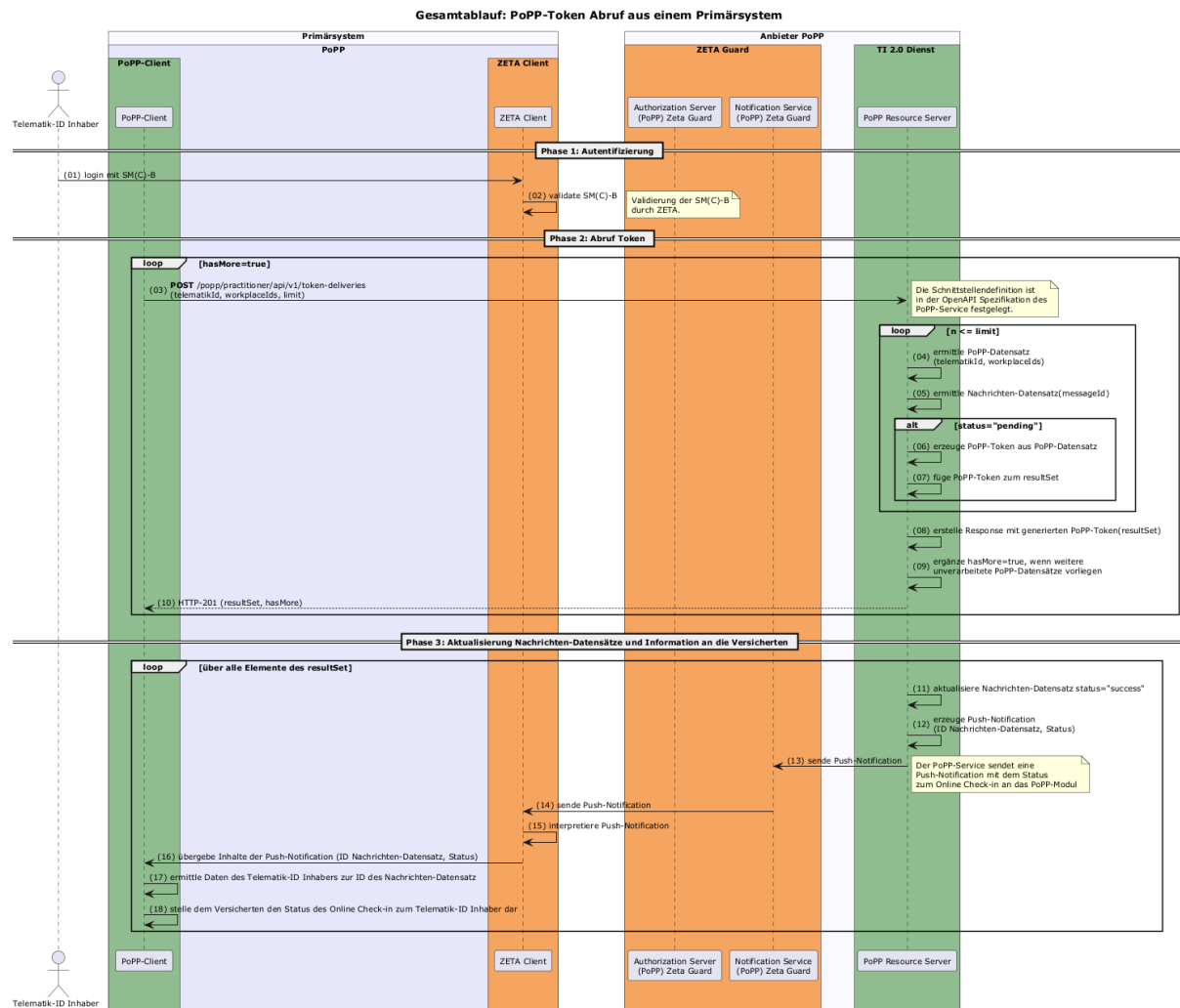


Abbildung 15: Detaillierter Ablauf zum PoPP-Token-Abruf aus einem Primärsystem

9.5.3 Beschreibung der Schritte zum PoPP-Token-Abruf aus einem Primärsystem

Tabelle 8: Beschreibung der Schritte zur Generierung und zum Abruf von PoPP-Token aus dem Online Check-in

Schritt	Beschreibung
Authentifizierung	
1	login mit SM(C)-B Der Inhaber der telematik-I authentifiziert sich über sein Primärsystem mit seiner SM(C)-B.
2	validate SM(C)-B Die Validierung der SM(C)-B erfolgt durch die ZETA Client Komponente im Primärsystem.

Ende Authentifizierung		
Der Start des PoPP-Token-Abrufs im Primärsystem eines Telematik-ID-Inhabers erfolgt manuell oder automatisiert		
3	POST:/popp/practitioner/api/v1/token-deliveries\n(telematikId, workplaceld, limit)	• Mit dem Aufruf der Schnittstelle am PoPP-Service Resource Server durch das Primärsystem wird Generierung und Abruf der PoPP-Token aus dem Online Check-in gestartet. • Der Aufruf enthält als Request-Parameter • Telematik-ID, zu der die PoPP-Token abgerufen werden sollen • Workplaceld, zu der die PoPP-Token abgerufen werden sollen • 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.
4	ermittle PoPP-Datensatz\n(telematikId, workplacelds)	Der PoPP-Service Resource Server ermittelt die PoPP-Datensätze und Nachrichten-Datensätze zu übergebenen Telematik-ID und Workplaceld, bei denen das Attribut "status" des Nachrichten-Datensatz den Wert "pending" hat. Es werden maximal soviel Datensätze extrahiert, bis "limit" erreicht ist.
5	ermittle Nachrichten-Datensatz(messageId)	
6	erzeuge PoPP-Token aus PoPP-Datensatz	Aus jedem ermittelten PoPP-Datensatz wird durch den PoPP-Service Resource Server ein PoPP-Token generiert.
7	füge PoPP-Token zum resultSet	Das PoPP-Token und die zugeordnete Workplaceld (wenn vorhanden) werden der Ergebnismenge hinzugefügt, die an das Primärsystem zu übermitteln ist.
8	erstelle Response mit generierten PoPP-Token(resultSet)	Der PoPP-Service Resource Server erstellt die Antwort auf den Anfrage-Request. Die Antwort enthält
9	ergänze hasMore=true, wenn weitere\nunverarbeitete PoPP-Datensätze vorliegen	• Ein ResultSet mit • den generierten PoPP-Token • der Workplaceld zum PoPP-Token (wenn vorhanden) • Die Information, ob weitere PoPP-Datensätze

		zur Telematik-ID abrufbar sind.
10	HTTP-201 (resultSet,hasMore)	Der PoPP-Service Resource Server antwortet auf dem Primärsystem-Aufruf.
Ende PoPP-Token-Erstellung und Abruf		
StartAktualisierung Nachrichten-Datensätze und Information an die VER nach erfolgreichem Versand der PoPP-Token		
11	aktualisiere Nachrichten-Datensatz status="success"	Nach erfolgreichem Versand der generierten PoPP-Token an das Primärsystem aktualisiert der PoPP-Service Resource Server den Nachrichten-Datensatz zu jedem gesendeten PoPP-Token indem der Wert des Attributs "status" auf "success" gesetzt wird.
12	erzeuge Push-Notification\n(ID Nachrichten-Datensatz, Status)	Der PoPP-Service Resource Server erzeugt eine Push-Notification gemäß [gemSpec_ZETA]. Die Push-Notification enthält die Information zum Status des Online Check-in zur Telematik-ID (und WorkplaceID) und den Identifier des 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.
16	übergebe Inhalte der Push-Notification (ID Nachrichten-Datensatz, Status)	Der ZETA-Client übergibt die Inhalte der Nachricht (ID Nachrichten-Datensatz, Status) an das PoPP-Modul.
17	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.
18	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.
EndeAktualisierung Nachrichten-Datensätze und Information an die VER		

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

1307

1308 **9.6 Ablauf "Abfrage des Status zur PoPP-Token-Generierung"**

1309 **9.6.1 Übersicht zum Ablauf der Abfrage des Status zur PoPP-**
1310 **Token-Generierung**

1311 Die Abfrage des aktuellen Status der PoPP-Token-Generierung erfolgt, indem an den
1312 PoPP-ServiceResource Server ein Request mit dem Identifier des Nachrichten-Datensatz
1313 gestellt wird. Der PoPP-Service Resource Server ermittelt den Nachrichten-Datensatz und
1314 gibt in seiner Antwort den dort gespeicherten Status-Wert zur PoPP-Token-Generierung
1315 zurück.

1316 Ablauf gleicht in weiten Teilen dem zur Erstellung des PoPP-Datensatzes und des
1317 Nachrichten-Datensatzes. So sind die Authentifizierung des Nutzers mit seiner
1318 GesundheitsID über die ZETA-Komponenten und die Übermittlung des Ergebnis identisch.
1319 Abweichend sind lediglich die Requestparameter und die Funktion selbst, die der PoPP-
1320 Service Resource Server bei Eingang des Requests ausführt.

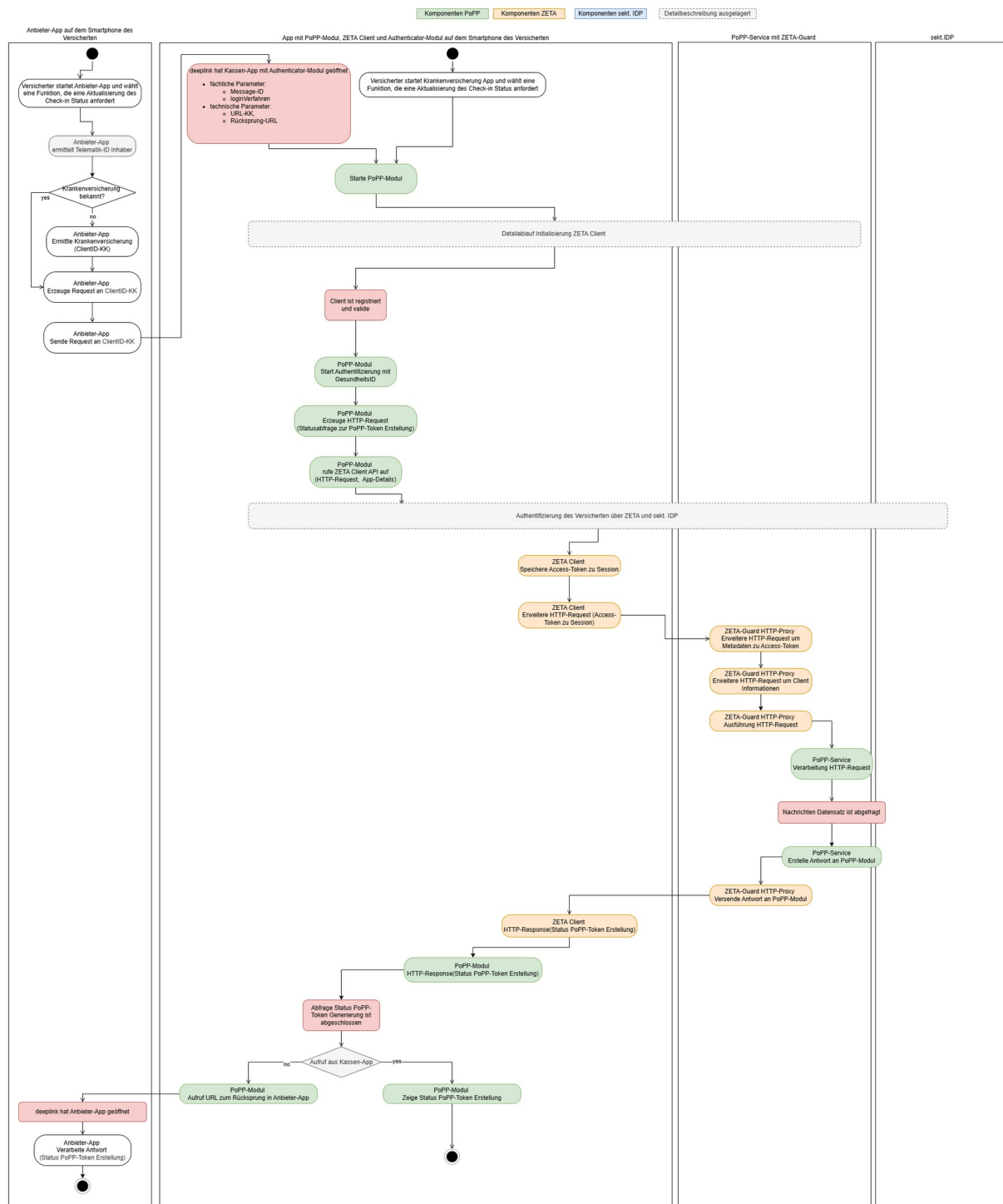


Abbildung 16: Laufzeitsicht - Ablauf der Statusabfrage zur PoPP-Token-Generierung aus einer Anbieter-App oder Kassen-App und Authentifizierung über die GesundheitsID

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"

1328 stellt den vollständigen Ablauf für den Fall dar, dass der Status zu einem zuvor
1329 durchgeführten Online Check-in Prozess am PoPP-ServiceResource Server abgerufen wird.
1330 Der Abruf kann manuell durch aktive Einbeziehung der VER oder automatisiert im
1331 Hintergrund zu nicht abgeschlossenen Check-in Prozessen gestartet werden.

1332 Der Status der PoPP-Token-Generierung wird dem PoPP-Modul zum Abschluss des Ablaufs
1333 vom PoPP-Service Resource Server zugesendet.

1334 Die Schnittstellen sind den OpenAPI spezifiziert:

- 1335 • [I_PoPP_Modul_mobile_CheckIn.yaml] - für den Aufrufs des PoPP-Modul aus einer
1336 Anbieter-App
- 1337 • [I_PoPP_Service_mobile_CheckIn.yaml] -für die Aufrufe am PoPP-Service Resource
1338 Server vom PoPP-Modul
- 1339 • [I_AnbieterApp_mobile_CheckIn.yaml] - für den callback-Aufruf der Anbieter-App aus
1340 dem PoPP-Modul



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

Seite 83 von 95
Stand: 03.09.2026

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.
5	erzeuge HTTP-POST Request(messageId, callBackAppURL, loginVerfahren)	Die Anbieter-App erzeugt einen HTTP-Request entsprechend der OpenAPI Spezifikation [I_PoPP_Modul_mobile_CheckIn.yaml] 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, • Wertebereich: "gID", wenn der Login durch die Authentifizierung mit GesundheitsID erfolgen muss. Hinweise: Mit Einführung weiterer Verfahren wie eGK-in-Fernversorgung oder EUDI-Wallet wird der Wertebereich erweitert.
6	sende HTTP-POST Request(messageId, callBackAppURL, loginVerfahren)	Die Anbieter-App sendet den HTTP-Request entsprechend der OpenAPI Spezifikation [I_PoPP_Modul_mobile_CheckIn.yaml] 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.
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		
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.yam I] 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>

		<pre> } } } } </pre>
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...\nÖffne URL für ZETA AS	ZETA Client sendet einen Authorization Request mit den Parametern des Authorization Code Flow und den authorizationDetails{auth} an den ZETA Guard Authorization-Server.
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.
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.
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.
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	GET:	Der ZETA Guard HTTP-Proxy ruft am

1	httpReadStatusPoPPRequest(messageld) Header(Client-Information)	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 (messageld) aus den Request Parameter.
4 3	ermittel PoPP-Nachrichten-Datensatz(messageld)	Der PoPP-Service Resource Server ermittelt den Nachrichten-Datensatz zur messageld.
4 4	ermittle des Statuswert im Nachrichten-Datensatz	Der PoPP-Service Resource Server ermittelt den Wert des Attributes "status" aus dem Nachrichten-Datensatz.
4 5	createResponse(statusPoppTokenGeneration)	Der PoPP-Service Resource Server erstellt die Antwort an das PoPP-Modul • <i>messageld</i> - Identifiziert 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
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.
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.
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 der ID 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		

1350

10 Anhang C - Offene Punkte, Fragen

10.1 Offene Punkte

OP-PoPP-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-PoPP-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.
OP-PoPP-3	[gemSpec_Popp_Modul#3.2.2.2]	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.
OP-PoPP-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-PoPP-5	[gemSpec_Popp_Modul#5.6.4]	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.
OP-PoPP-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-PoPP-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
OP-PoPP-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 WorkplacelD 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.

1354
1355
1356