Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation
Sektoraler Identity Provider

Version:	2.0.0 CC
Revision:	477815
Stand:	11.07.2022
Status:	zur Abstimmung freigegeben
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemSpec_IDP_Sek

Dokumentinformationen

Änderungen zur Vorversion

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

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	17.12.21		initiale Version	gematik
2.0.0 CC	11.07.22		Anpassung für die IDP Föderation	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die Anforderungen zu Herstellung, Test und Betrieb des Produkttyps sektoraler Identity Provider (IDP). Ein sektoraler IDP basiert auf den Standards OpenID Connect (OIDC), Open Authorization 2.0 (OAuth 2) und JSON Web Token (JWT). Die hier beschriebenen Schnittstellen werden vom Authenticator-Modul und von Clients für eine Authentifikation eines Nutzers genutzt. Diese Authentifikation ist die Voraussetzung, damit ein Client Zugang zu Fachdaten und Prozessen eines Fachdienstes erlangen kann. Ein sektoraler IDP verwaltet und steuert den Authentifizierungsprozess für Anwendungen der Telematikinfrastruktur (TI).

1.2 Zielgruppe

Das Dokument richtet sich an Hersteller und Anbieter von Identity Providern, welche die Funktionen eines sektoralen IDP für die TI realisieren wollen.

1.3 Geltungsbereich

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

Nicht Bestandteil des vorliegenden Dokumentes ist die konkrete Umsetzung der Authentifizierung eines Nutzers durch einen sektoralem IDP.

Als Umsetzungsleitlinie ist [OpenID Connect Core 1.0] heranzuziehen. Die TI-weit übergreifenden Festlegungen – insbesondere aus Dokumenten wie beispielsweise [gemSpec_Krypt] bezüglich Algorithmen und Schlüsselstärken sowie [gemSpec_PKI] bezüglich zu verwendender Zertifikatstypen und deren Attributausprägungen – haben Bestand, sind weiterhin bindend und werden nicht in diesem Dokument beschrieben. Die konkreten, für das Produkt relevanten Anforderungen finden sich in den entsprechenden Steckbriefen.

1.5 Methodik

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

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

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

Hinweis auf offene Punkte

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

2 Systemkontext

2.1 Allgemeiner Überblick

Zentrales Merkmal der zu entwickelnden Gesamtlösung der sektoralen IDP ist das Prinzip der Föderation. Die Funktionalität des IDP wird nicht von einem einzigen zentralen Dienst bereitgestellt, sondern „kollektiv" durch eine Menge von sektoralen IDP, für die jeweils die entsprechenden identitätsherausgebenden Institutionen verantwortlich sind, welche auch für die jeweiligen Nutzergruppen zuständig sind.

Um eine Gesamtlösung sicherzustellen, bei der Anwendungen in möglichst einfacher Weise die verschiedenen sektoralen IDP nutzen können, sind in bestimmten Bereichen einheitliche Vorgaben für die technische und organisatorische Umsetzung zu erstellen:

- Einheitliche Identitätsattribute für die Nutzergruppen (`scopes`)
- Einheitliche Verfahren zum Auffinden von sektoralen IDP (IDP Discovery)
- Grundstruktur der Vertrauensbeziehungen der Föderierung (Zwischen Fachdiensten und IDP)
- Einheitliche Vertrauensniveaus (Trust Framework).

Die Grundidee der Föderation ist die Erstellung eines Vertrauensraum, in dem mehrere Anwendungen und IDP abgesichert über Vertrauensketten (Trust chain) miteinander kommunizieren. Grundlage für die Föderation sind die Standards für Autorisierung und Authentisierung von Anwendungen und Nutzern OAuth 2.0 und OIDC.

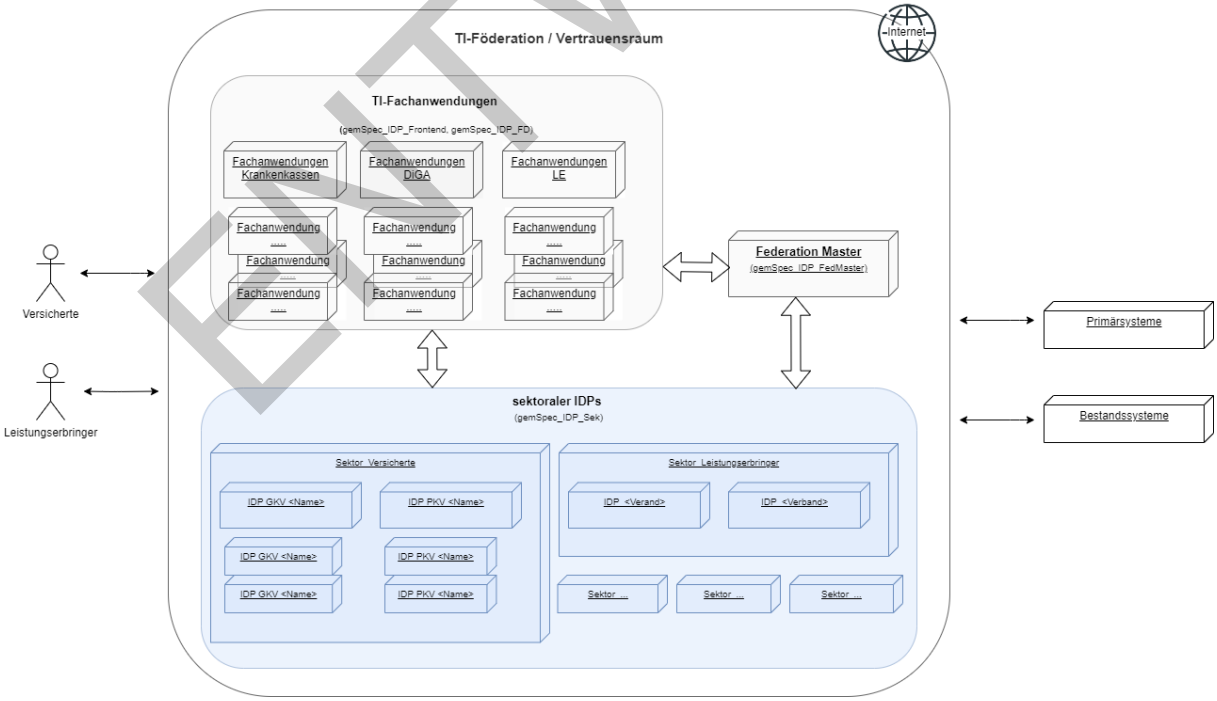

Abbildung 1 : Überblick TI-Föderation

Das Konzept der sektoralen IDP sieht vor, dass diese nicht ausschließlich von Fachanwendungen der TI zur Authentifizierung von Anwendern zu verwenden sind. Vielmehr können (und sollen) auch Anwendungen außerhalb der TI (z. B. Anwendungen

der Krankenkassen) die sektoralen Identity Provider zur Nutzerauthentifizierung und Attributübertragung verwenden. Die in den Spezifikationen der gematik festgelegten Anforderungen sind jedoch für diese Anwendungen und den Anmeldungsflow am sektoralen IDP nicht bindend. So sind beispielsweise `scopes` und `claims` frei wählbar und eine Registrierung am Federation Master für diese Anwendungen nicht zwingend notwendig. Die Fachanwendungen müssen sich lediglich OIDC konform am sektoralen IDP (also dem OpenID-Provider) registrieren. Der sektorale Identity Provider kann für diese Anwendungen auch zugleich als Authorization-Service agieren und `ACCESS_TOKEN` ausstellen.

2.2 Detaillierter Überblick

Die untere Abbildung beschreibt den Systemkontext aus Sicht des sektoralen IDP. Das Anwendungsfrontend des Fachdienstes stellt die Anfrage zur Authentifizierung des Nutzers an den Authorization-Service des Fachdienstes. Dieser generiert eine `CODE_CHALLENGE` und stellt einen Pushed Authorization Request (PAR) an den entsprechenden sektoralen IDP. Der Fachdienst agiert diesem gegenüber als Client. Über das Authenticator-Modul des sektoralen IDP findet dann die Authentifizierung des Nutzers statt. Anschließend erhält der Authorization-Service des Fachdienstes einen `AUTHORIZATION_CODE`, welchen er bei Token-Endpunkt des sektoralen IDP gegen einen `ID_TOKEN` eintauscht. Der Authorization-Service des Fachdienstes erstellt nun ein `ACCESS_TOKEN` für das Anwendungsfrontend, mit welchem dieses auf die, für den Nutzer freigegebenen, Ressourcen des Fachdienstes zugreifen kann. Die Kommunikation zwischen Anwendungsfrontend und Authorization-Service des Fachdienstes kann ebenfalls über einen eigenen `AUTHORIZATION_CODE` abgesichert werden.

Der Fachdienst und der sektoralen IDP müssen sich zuvor beim Federation Master in Form eines organisatorischen Prozesses registriert haben.

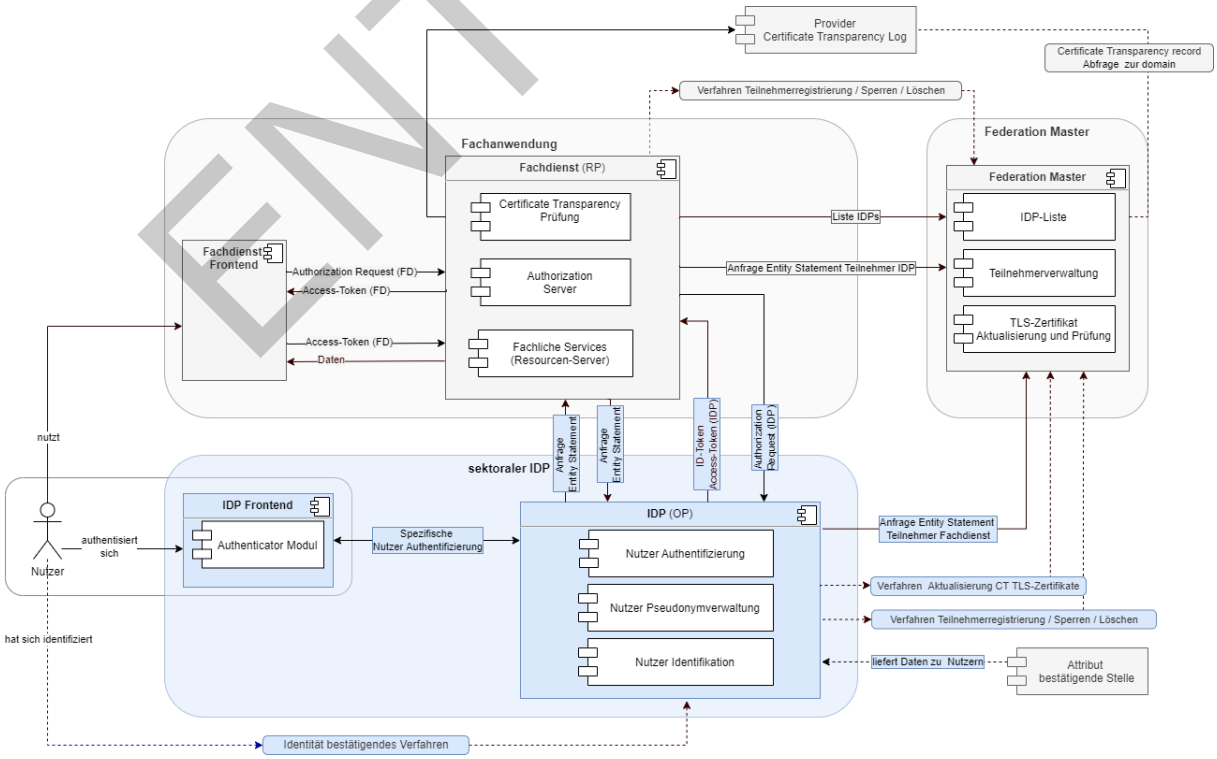

Abbildung 2: Systemkontext

2.3 Zerlegung des Produkttyps

Der Produkttyp besteht aus der zentralen Komponente des sektoralen IDP. Dieser wird bei der Durchführung des Authentifizierungsprozesses vom Authenticator-Modul unterstützt. Das Authenticator-Modul übernimmt die Ausführung der Nutzerauthentisierung.

Der sektoraler IDP stellt die zentralisierte Identitätsprüfung der auf die Fachdienste zugreifenden Nutzer bereit. Als weitere Teile der Gesamtlösung sind neben dem sektoraler IDP die Clients (Anwendungsfrontend) und die Fachdienste zu nennen, auf denen Fachdaten für den Zugriff durch die Nutzer (z. B. Versicherte oder Bediener eines AVS, PVS oder KVS) bereitgestellt werden. Ein sektoraler IDP bietet seine Dienste Fachdiensten an, auf welche Millionen Nutzer zeitgleich zugreifen.

2.4 Akteure und Rollen

Als sektoraler IDP wird ein Dienst bezeichnet, welcher die Nutzerauthentisierung durchführt. Nach erfolgreicher Nutzerauthentisierung stellt der sektoraler IDP Identitätsinformationen zum Nutzer bereit.
Die Identitätsinformationen werden von den Fachdiensten zur Durchführung einer Nutzerautorisierung verwendet, also zur Feststellung, auf welche Fachdaten und -prozesse des Fachdienstes dem Nutzer Zugriff gewährt wird. Die bereitgestellten Identitätsinformationen sind spezifisch für die unterschiedlichen Gruppen von Nutzern bzw. Sektoren innerhalb der TI des Gesundheitswesens. Einen Sektor stellen insbesondere die Krankenkassen mit den Versicherten als Nutzer dar.Zukünftig werden allerdings auch andere Personengruppen wie z. B. Ärzte oder Pflegeinstitutionen über sektorale IDP angebunden.

Im Systemkontext eines sektoralen IDP interagieren verschiedene Akteure (Nutzer und aktive Komponenten) in unterschiedlichen OAuth2-Rollen gemäß [The OAuth 2.0 Authorization Framework (section-1.1)] und OpenID-Connect-Rollen gemäß [OpenID Connect Core 1.0] und [OpenID Connect Federation 1.0].

Die Abläufe zur Nutzerauthentisierung für einen Fachdienst sowie der Herausgabe der Identitätsinformationen durch den sektoralen IDP sind als innere Flow und der äußere Flow in Abschnitt 2.5 erläutert.

Tabelle 1 : Akteure und Rollen

Akteur	Rolle "OAuth2"	Rolle "OIDC"
Nutzer (z. B. Versicherte)	Resource Owner	Resource Owner
Fachdienst - Authorization-Server	Authorization-Server	Relying Party (RP)
Fachdienst - Fachliche Services (Fachdaten und -Prozesse)	Protected Resource	-

Fachdienst - App-Frontend	Client, Nutzerschnittstelle als App	-
Fachdienst - Web-Frontend	Client, Nutzerschnittstelle als Web-Anwendung	-
Fachdienst - UI-Backend	Client, Services der UI-Bereitstellung für Web-Anwendung	-
Authenticator-Modul des sektoralen IDP	-	Frontend des sektoralen IDP
sektoraler IDP	-	OpenID Provider (OP)
Attributbestätigende Stelle	-	-
Federation Master	-	Vertrauensanker (Trust Anchor)

Nutzer (Rolle: Resource Owner)

Der Resource Owner ist eine natürliche Person, welcher auf die beim Fachdienst (Resource Server) für ihn bereitgestellten Daten und Prozesse (Protected Resource) zugreift.

Der Resource Owner verfügt über die folgenden Komponenten:

- Endgerät des Nutzers
- Authenticator-Modul
- Anwendungsfrontend des Fachdienstes.

Fachdienst (Rolle: Authorization-Server)

Der Authorization-Server des Fachdienstes (OIDC Relying Party) stößt die Authentifizierung des Nutzers beim sektoralen IDP an und erhält als Ergebnis einen Authorization Code, den er gegen ein `ID_TOKEN` und `ACCESS_TOKEN` beim sektoralen IDP eintauschen kann. Der Authorization-Server des Fachdienstes verwendet die Informationen aus dem `ID_TOKEN` für die Feststellung der Zugriffsrechte des Anwendungsfrontend auf die Ressourcen des Fachdienstes. Der Authorization-Server des Fachdienstes stellt eigene `ACCESS_TOKEN` und `REFRESH_TOKEN` für das Anwendungsfrontend aus.

Fachdienst (Rolle: Resource Server)

Der Resource Server ist der Fachdienst, der dem Nutzer (Resource Owner) Zugriff auf seine Fachdaten und Prozesse (Protected Resource) gewährt. Der Fachdienst, der die geschützten Fachdaten (Protected Resources) anbietet, ist in der Lage, auf Basis von ACCESS_TOKEN Zugriff für Clients zu gewähren. Ein solches Token repräsentiert die delegierte Identifikation des Resource Owner.

Anwendungsfrontend (Rolle: Client)

Das Anwendungsfrontend (OAuth2 Client) greift auf Fachdienste (Resource Server) und ihre geschützten Fachdaten (Protected Resource) zu. Das Anwendungsfrontend kann auf einem Server als Webanwendung (Primärsystem als Terminalserver), auf einem Desktop-PC oder einem mobilen Gerät (z. B. Smartphone) oder als App auf einem mobilen Gerät ausgeführt werden. Ist das Anwendungsfrontend eine Webanwendung, so ist die Backend-Komponente, welche die UI für die Visualisierung im Browser auf dem Gerät des Nutzers realisiert, ebenfalls Teil des Clients.

Sektoraler IDP mit dem Authenticator-Modul als Frontend (Rolle: OpenID Provider)

Der Authorization-Server des sektoralen IDP authentifiziert den Resource Owner (Nutzer) und stellt einen Authorization Code aus. Dieser Authorization Code kann später gegen ein `ID_TOKEN` beim sektoralen IDP eingetauscht werden. Das `ID_TOKEN` enthält die Informationen für den vom Resource Owner erlaubten Anwendungsbereich (`scope`).

Weitere Akteure im Kontext des sektoralen IDP sind:

Fachdaten und Prozesse (Rolle: Protected Resource)

Die geschützten Fachdaten und Prozesse, welche vom Fachdienst (Resource Server) angeboten werden.

Attributbestätigende Stelle

Attributbestätigende Stellen sind legitimierte Organisationen, welche die Korrektheit der Attribute verantworten, die durch sie für einen Nutzer beim sektoralen IDP bestätigt werden.

Als Teilprozess der Registrierung ist die zuverlässige und eindeutige Identifikation der Nutzer zwingend notwendig. Hierbei werden eindeutige Identifikationsmerkmale der realen Identitäten benötigt und letztlich als Identitätsinformationen dem sektoralen IDP zur Verfügung gestellt.

Die eindeutigen Identitäten von natürlichen Personen (Versicherte, Leistungserbringer) bzw. juristischen Personen (medizinische Institutionen, Gesellschafterorganisations- und Kostenträgergeschäftsstellen) werden innerhalb der TI über die Krankenversichertennummer des Versicherten und die Telematik-ID eines Leistungserbringers bzw. einer medizinischen Institution oder Organisation des Gesundheitswesens repräsentiert.

Federation Master

Der Federation Master ist eine zentrale Komponente der TI. Alle Teilnehmer der Föderation müssen beim Federation Master registriert sein. Teilnehmer der Föderation sind in diesem Kontext alle Fachdienste und sektoralen IDP. Die Registrierung erfolgt durch einen organisatorischen Prozess. Der Federation Master verwaltet die öffentlichen Schlüssel aller Teilnehmer. Der Federation Master stellt auf Anfrage Teilnehmerbestätigungen in Form von Entity Statements aus. Der Federation Master agiert als Trust Anchor im Sinne der OpenID-Connect-Federation Spezifikation.

2.5 Nachbarsysteme und Interaktion

Ein sektoraler IDP bietet zahlreiche Schnittstellen gegenüber unterschiedlichen Akteuren an, weswegen es notwendig ist, die einzelnen Schnittstellen so zu beschreiben, dass andere Akteure deren Funktionsweise leichter verstehen können.

Vorbereitende Maßnahmen:

- Der Fachdienst hat bei der Registrierung am Federation Master seine öffentlichen Schlüssel hinterlegt.
- Der Fachdienst hat bei der Registrierung am Federation Master sie `scopes` hinterlegt, welche er für die Autorisierung eines Nutzers zwingend benötigt
- Der Fachdienst kennt das Entity Statement der sektoralen IDP und hat bei der Registrierung dort seine öffentlichen Schlüssel hinterlegt.

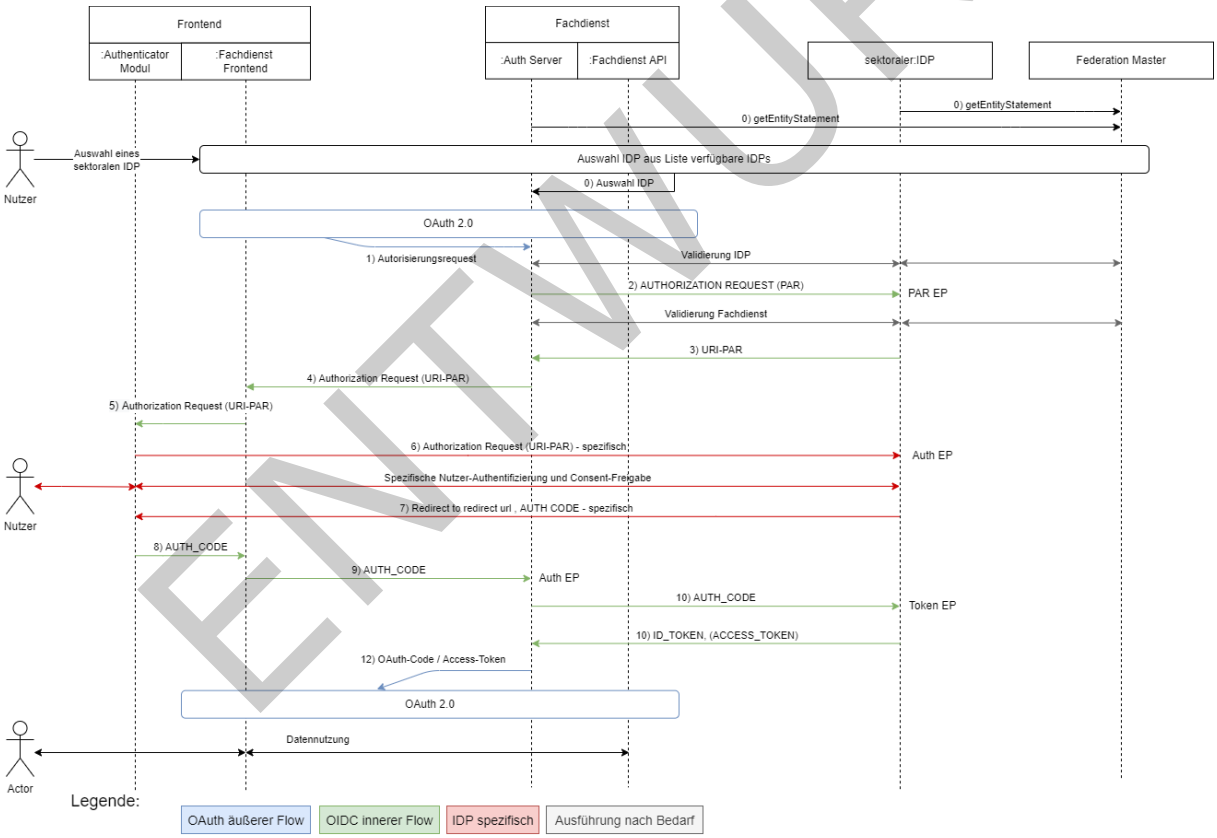

Abbildung 3 : OAuth- und OIDC-Flow

Der gesamte Authentifizierungsprozess (Abbildung: *"OAuth- und OIDC-Flow"*) basiert aus Gründen der Entkoppelung zwischen den Authentifizierungsmethoden und Token-Formaten der sektoralen IDP und des Fachdienstes aus zwei ineinander geschachtelten OAuth2-Flows vom Typ `grant_type`= *authorization_code*.

Im äußeren Flow (Schritt 1) wendet sich das Anwendungsfrontend als Client initial an den Authorization-Server des Fachdienstes und signalisiert diesem über einen zusätzlichen Parameter `idp_iss` (siehe Kapitel 7.1.4 Detailinformationen zum Flow) den zur Authentifizierung zu verwendenden sektoralen IDP. Der innere Flow beginnt mit einem Authorization Request in Schritt 2 und endet mit Schritt 11, der Herausgabe eines `ID_TOKEN` und `ACCESS_TOKEN` vom sektoralen IDP an den Authorization-Server des Fachdienstes.

Die erste Anfrage an den sektoralen IDP geht am PAR-Endpunkt [OAuth 2.0 Pushed Authorization Requests (section-2)] ein. Der Authorization-Server des Fachdienstes reicht dort am Endpunkt den Authorization Request zur Authentifizierung des Nutzers und zur Bestätigung des `scope` der anfragenden Anwendung sowie eine `CODE_CHALLENGE` ein. Der `scope` der angefragten Nutzdaten ist im Entity Statement des Fachdienstes hinterlegt. Dieses ist dem sektoralen IDP bekannt. Ist das nicht der Fall, so wird das Entity Statement des Fachdienstes durch den sektoralen IDP abgefragt und durch den Federation Master bestätigt. Der Authorization-Server des Fachdienstes tritt bzgl. des inneren Flow als Client auf.

Im Weiteren Ablauf wird der Nutzer wird dann aufgefordert sich, unter Nutzung des Authenticator-Moduls des sektoralen IDP, zu authentisieren. Dies erfolgt über eine Schnittstelle zwischen dem Authenticator-Modul und Authorization-Endpunkt des sektoralen IDP. Der sektorale IDP kann auf eine erneute Authentisierung des Nutzers verzichten, wenn diese bereits vor kurzem erfolgte. Überschreitet die letzte Authentisierung den in der Spezifikation festgelegten Zeitraum oder wird das Authenticator-Modul zum ersten Mal gestartet, muss eine Authentisierung des Nutzers erfolgen.

Nach erfolgreicher Authentisierung und der Consent-Freigabe durch den Nutzer erstellt der sektorale IDP den `AUTHORIZATION_CODE`. Dieser wird an den Authorization-Server des Fachdienstes übermittelt, welcher ihn am Token-Endpunkt [The OAuth 2.0 Authorization Framework (section-3.2)] des sektoralen IDP einreicht. Der sektorale IDP überprüft den `AUTHORIZATION_CODE` und stellt bei positiver Validierung einen `ID_TOKEN` und ein `ACCESS_TOKEN` aus.

Anschließend erstellt der Authorization-Server des Fachdienstes einen `AUTHORIZATION_CODE`, der an das Anwendungsfrontend zurückgegeben wird. Der äußere Flow endet mit der Herausgabe eines `ACCESS_TOKEN` an das Anwendungsfrontend bzw. im Fall von Web-Anwendungen an das Web-Backend des Anwendungsfrontends. Der weitere fachliche Ablauf zum Einreichen der Token und zur Nutzung der Fachdaten und Prozesse ist anwendungsspezifisch.

Tabelle 2 : Schritte OAuth- und OIDC-Flow

Schritt	Beschreibung
optional	Die Auswahl eines sektoralen IDP durch den Anwender am Anwendungsfrontend ist erforderlich, wenn der dem Fachdienst (z. B. aus früheren Sitzungen) nicht bekannt ist.
1	Das Anwendungsfrontend sendet einen Authorization Request mit dem zur Anmeldung gewünschten sektoralen IDP an den Autorisierungsserver des Fachdienstes.

optional	Falls der Autorisierungsserver das Entity Statement des sektoralen IDP noch nicht kennt, lädt er dies herunter. (/.well-known/openid-federation). Der sektorale IDP sendet sein Entity Statement zurück. Der sektorale IDP wird gegen den Federation Master validiert indem der Fachdienst das Entity Statement zum sektoralen IDP beim Federation Master abruft.
2	Der Autorisierungsserver sendet einen Pushed Authorization Request (PAR) inkl. Code-Challenge, benötigter `claims` bzw. `scope` und eines `private_key_jwt` an den sektoralen IDP. Die Erzeugung der Code-Challenge erfolgt durch den Autorisierungsserver entsprechende der Spezifikation [RFC7636 - Proof Key for Code Exchange by OAuth Public Clients] (PKCE) über die Generierung eines Zufallswertes (Codeverifier) und die Erzeugung eines Hashwert für den Codeverifier. Die Code-Challenge der base64-codierte Hashwert des Codeverifier.
optional	Falls der sektorale IDP das Entity Statement des Autorisierungsservers noch nicht kennt, lädt er dies herunter. (/.well-known/openid-federation). Der Autorisierungsserver sendet sein Entity Statement zurück und der sektorale IDP registriert ihn als Client. Der Fachdienst wird gegen den Federation Master validiert indem der sektorale IDP das Entity Statement zum Fachdienst/Autorisierungsserver beim Federation Master abruft.
3	Der sektorale IDP sendet eine Request-URI (mit Bezug zum vorherigen AUTHORIZATION_REQUEST) an den Autorisierungsserver.
4	Der Autorisierungsserver sendet die Request-URI und Client ID an das Anwendungsfrontend zur Weiterleitung an die Adresse des Authenticator des sektoralen IDP.
5	Anwendungsfrontend öffnet den Authenticator für die eigentliche Authentifizierung des Anwenders (Deep-Link/Universal-Link).
6	Das Authenticator-Modul leitet den Authentication Request an den sektoralen IDP weiter.
spezifisch	Der Ablauf der Authentifizierung des Nutzers ist IDP spezifisch.
7	Der Authorization-Endpunkt des sektoralen IDP antwortet dem Authenticator-Modul mit dem `AUTHORIZATION_CODE` und einem Redirect zum Fachdienst.
8	Das Authenticator-Modul des sektoralen IDP ruft über einen App-Link bzw. Universal-Link entsprechend der Redirect-URL das Anwendungsfrontend auf (eigentlich ein Redirect zum Fachdienst aber das Frontend ist auf die Adresse registriert) und übergibt den `AUTHORIZATION_CODE`
9	Die Anwendungsfrontend leitet den `AUTHORIZATION_CODE` (IDP) an den Autorisierungsserver.

10	Der Autorisierungsserver reicht den `AUTHORIZATION_CODE` (IDP), den `CODE_VERIFIER` und seinen `private_key_jwt` beim Token-Endpunkt des sektoralen IDP ein.
11	Der Autorisierungsserver erhält vom Token-Endpunkt des sektoralen IDP einen `ID_TOKEN` und `ACCESS_TOKEN` mit den gewünschten `claims`, der mit dem öffentlichen Schlüssel aus der Registrierung verschlüsselt ist.
	Der weitere Ablauf entspricht dem OAuth-Flow und unterscheidet sich in Details je nach Ausprägung des Anwendungsfrontend als App oder Web-Anwendung.

395

396 Die Abläufe für App-App Kommunikation, Web-App Kommunikation und Kommunikation
397 unter Beteiligung von zwei Geräten sind im Anhang B detailliert beschrieben.

3 Übergreifende Festlegungen

Der sektorale IDP muss die folgenden übergreifenden Anforderungen erfüllen.

A_22838 - Entgegennahme von Sperrmeldungen
Der Anbieter des sektoralen IDP MUSS Sperrmeldungen von Sperrberechtigten, zu von ihm verantworteten Authentisierungsmitteln, jederzeit entgegennehmen und das betroffene Authentisierungsmittel oder auch den gesamten Zugang des Nutzers daraufhin unverzüglich sperren.**[<=]**

Hinweis: Dies bezieht sich nicht auf für eine Authentisierung verwendete eGK oder den elektronischen Identitätsnachweis (online-Ausweisfunktion).

A_23101 - Durchsetzung der eGK-Sperrfrist
Der Anbieter des sektoralen IDP MUSS sicherstellen, dass lediglich elektronische Gesundheitskarten, deren Herausgeber eine Sperrfrist von 60 Minuten zwischen Sperrmeldung und Sperrung am OCSP-Responder einhält, zur Identifikation und Authentifizierung genutzt werden können.**[<=]**

A_22690 - Darstellen der Voraussetzungen für sicheren Betrieb des Produkts im Betriebshandbuch
Der Hersteller des sektoralen IDP MUSS für sein Produkt im dazugehörigen Betriebshandbuch leicht ersichtlich darstellen, welche Voraussetzungen vom Betreiber und der Betriebsumgebung erfüllt werden müssen, damit ein sicherer Betrieb des Produktes gewährleistet werden kann.**[<=]**

A_22691 - Sicherer Betrieb des Produkts nach Betriebshandbuch
Der Anbieter eines sektoralen IDP MUSS die im Betriebshandbuch des eingesetzten sektoralen IDP beschriebenen Voraussetzungen für den sicheren Betrieb des Produktes gewährleisten.**[<=]**

A_23044 - Unterstützung von Diensten außerhalb der TI
Der Anbieter des sektoralen IDP KANN die Anmeldung an weiteren Diensten außerhalb der Föderation unterstützen und diesen die Authentisierung von Nutzern auf Basis der bestehenden digitalen Identitäten anbieten.**[<=]**

3.1 Sicherheitsanforderungen für den operativen Betrieb

A_22239 - Schützenswerte Objekte
Der Anbieter eines sektoralen Identity Provider MUSS die folgenden kryptographischen Objekte als schützenswerte Objekte in seinem Sicherheitskonzept berücksichtigen: (a) Private Schlüssel, (b) Öffentlicher Schlüssel, (c) Öffentliche Schlüssel von registrierten Clients, (d) Datensätze zu den einzelnen Nutzern, (e) Authentisierungsinformationen von Sperrberechtigten, (f) Dokumentation über beauftragte und durchgeführte Sperrungen, (g) Statusinformationen, (h) Authentisierungsinformationen zur Authentisierung von internen Akteuren bzw. Rollen, (i) Protokolldaten, (j) Konfigurationsdaten.**[<=]**

A_22240 - Berücksichtigung OWASP-Top-10-Risiken

Der Anbieter des sektoralen Identity Provider MUSS Maßnahmen zum Schutz vor den zum Zulassungszeitpunkt aktuellen OWASP-Top-10-Risiken umsetzen und dokumentieren, wie es vorgesehen ist, ebenfalls auf die nach dem Zulassungszeitpunkt aktuellen OWASP-Top-10-Risiken zu reagieren.**[<=]**

Hinweis: Die Nichtanwendbarkeit eines OWASP-Top-10-Risikos ist zu begründen. Für Informationen zum Umgang mit den OWASP-Top-10-Risiken wird auf den aktuellen [OWASP Top 10 Report] und die darin enthaltenen Vorgehensweisen für z. B. Entwickler und Tester verwiesen.

A_22241 - Interner Datenaustausch der Komponenten des sektoralen Identity Provider
Der Anbieter eines sektoralen Identity Provider MUSS beim internen Datenaustausch die Integrität, Authentizität und Vertraulichkeit der Daten sichern.**[<=]**

A_22242-01 - Gesicherte externe Schnittstellen des sektoralen Identity Provider
Der Anbieter eines sektoralen Identity Provider MUSS für den Datenaustausch mit anderen Rollen und Diensten Mechanismen zur Sicherung der Datenintegrität, der Authentizität und der Vertraulichkeit der Daten zur Verfügung stellen. Hierzu gehören z.B. die Schnittstellen vom Anbieter eines sektoralen Identity Provider zur Attributbestätigenden Stelle für die Übermittlung der Attribute bei der Einrichtung eines Nutzers sowie von Supportfälle.
[<=]

Hinweis: Die Attributbestätigende Stelle (z. B. der Kostenträger für Versicherte) verantwortet die Korrektheit dieser Daten.

A_22243-01 - Nutzung bestehender SGB-Datensätze bei Registrierung für Endanwender (Versicherte)
Der Anbieter des sektorale Identity Provider SOLL für die Registrierung der Endanwender die bestehenden Datensätze der Endanwender (Versicherte) beim Kostenträger verwenden, so wie sie im Rahmen der Vorgaben des Sozialgesetzbuches (SGB) erhoben wurden.
[<=]

A_22244 - Trennung der Betriebsumgebungen
Der Anbieter eines sektoralen Identity Provider MUSS sicherstellen, dass das Testsystem von dem Produktivsystem technisch, organisatorisch und betrieblich so getrennt wird, dass keine gegenseitige Beeinflussung und keine Verwechslung möglich sind. **[<=]**

A_22245 - Datenschutzgerechte Einrichtungs- und Sperrprozesse
Der Anbieter eines sektoralen Identity Provider MUSS die Einrichtungs- und Sperrprozesse datenschutzgerecht ausgestalten, d.h. insbesondere sind für die Verarbeitung der Antrags- und Sperrauftragsdaten die Datenschutzgrundsätze gemäß Art. 5 DSGVO zu berücksichtigen, sowie die technischen und organisatorischen Maßnahmen nach Art. 25 und Art. 32 DSGVO zu treffen.**[<=]**

A_22246 - Löschung von Nutzerinformationen
Der Anbieter eines sektoralen Identity Provider MUSS die Attributsdaten und Sperraufträge zu einem Nutzer unverzüglich löschen, sobald die gesetzlichen oder vertraglichen Aufbewahrungsfristen erreicht sind.**[<=]**

A_22839 - Fehlerprotokollierung

Falls der Anbieter eines sektoralen IDP eine Protokollierung zum Zwecke der Fehler- bzw. Störungsbehebung durchführen muss, MÜSSEN die Protokolldaten entsprechend dem Datenschutzgrundsatz der Datenminimierung (gemäß Art. 5 Abs. 1 Satz 1 lit.c DSGVO unter Berücksichtigung der Art. 25, 32 DSGVO) derart gestaltet sein, dass nur personenbezogene Daten in der Art und dem Umfang enthalten sind, wie sie zur Behebung erforderlich sind. Insbesondere MUSS der Anbieter eines sektoralen IDP sicherstellen, dass ein Bezug zwischen Nutzer und Fachdienst aus den Protokollen nicht ersichtlich sein. **[<=]**

A_23021 - Trennung von Diensten der Föderation und weiteren unterstützten Anwendungen
Wenn der Anbieter eines sektoralen Identity Providers die Anmeldung an weiteren Dienste außerhalb der Föderation unterstützt MUSS sichergestellt sein, dass diese die Verfügbarkeit, Performance und Sicherheit der Schnittstellen für Fachdienste der Föderation nicht beeinflussen können.**[<=]**

A_23023 - Sicherung externen Schnittstellen gegen bösartige Eingaben
Der sektorale IDP MUSS sicherstellen, dass alle Eingabewerte, welche vom sektoralen IDP über externe Schnittstellen entgegengenommen und verarbeitet werden, auf schadhafte Werte geprüft werden.
[<=]

Hinweis: Eine Prüfung der Eingabewerte muss produktseitig bereitgestellt werden und sollte mindestens Prüfungen auf Länge, Character Set, Schlüsselwörter und Steuerzeichen enthalten. Ein Fuzzing im Rahmen des Produkttests bzw. der Inbetriebnahme ist durchzuführen.

A_23022 - Prozesse zum Ändern oder Löschen personenbezogener Daten
Der Anbieter des sektoralen IDP MUSS sicherstellen, dass operative Prozesse, welche personenbezogene Daten ändern oder löschen können, ausschließlich im 4-Augen-Prinzip ausgeführt werden.
[<=]

A_22824 - Verhalten bei Vollauslastung
Der Anbieter eines sektoralen Identity Provider MUSS den Dienst so konfigurieren, dass bei Vollauslastung der Systemressourcen im sektoralen Identity Provider keine weiteren Verbindungen angenommen werden und dieser stattdessen mit dem HTTP-Statuscode "429 - Too Many Requests" antwortet.**[<=]**

Hinweis: Durch die Zurückweisung von Verbindungen wird sichergestellt, dass Clients einen Verbindungsaufbau mit einer anderen Instanz des Dienstes versuchen, bei der die erforderlichen Systemressourcen zur Verfügung stehen.

A_22692 - Kriterien für die Standortwahl von Rechenzentren
Der Anbieter des sektoralen IDP MUSS nachweisen, dass er die aktuellen Empfehlungen des BSI bei der Standortwahl seiner Rechenzentren vollumfänglich umsetzt. Der Anbieter des sektoralen IDP MUSS Unterschreitungen der Empfehlungen des BSI begründen und die Abmilderung der Risiken begründet nachweisen. Der Anbieter des sektoralen IDP MUSS einen Prozess für die Umsetzung zukünftige Empfehlungen des BSI bei der Standortwahl seiner Rechenzentren nachweisen.**[<=]**

Hinweis: Weitere Informationen finden Sie unter:
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/RZ-Sicherheit/Standort-Kriterien_Rechenzentren.pdf

A_22250 - Schutz der Verbindung zum sektoralen Identity Provider
Der Anbieter eines sektoralen Identity Provider MUSS sicherstellen, dass die Schnittstellen des sektoralen Identity Provider nur über eine gegen Abhören, Manipulation und Replay-Angriffe geschützte Verbindung genutzt werden können.**[<=]**

A_22512 - Schutz der Schnittstellen des sektoralen Identity Provider ins Internet
Der Anbieter eines sektoralen IDP MUSS sicherstellen, dass seine Schnittstellen ins Internet an allen Standorten durch einen DDoS-mitigierenden Dienstleister geschützt werden.**[<=]**

Hinweis: Die Informationen zu den Empfehlungen des BSI sind zu berücksichtigen:
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Dienstleister-DDos-Mitigation.html
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Dienstleister-DDos-Mitigation-Liste.pdf;

A_23099 - Datenverarbeitung innerhalb Europa
Der Anbieter eines sektoralen IDP MUSS im Sinne der vollständigen DSGVO-Konformität sicherstellen, dass die Datenverarbeitung innerhalb Europas erfolgt und dieses auch nachweisen.**[<=]**

A_23100 - Sicherstellung des Datenverkehrs ausschließlich in europäische Union
Der Anbieter eines sektoralen IDP MUSS sicherstellen, dass der Datenverkehr auch bei Einsatz eines externen Dienstleisters die europäische Union nicht verlässt.**[<=]**

A_22694 - Georedundanz des sektoralen Identity Provider
Der Anbieter des sektoralen Identity Provider MUSS diesen an mindestens zwei Standorten betreiben.
Jeder Standort MUSS dabei die Performancevorgaben allein erfüllen.
Eine getrennte Adressierung durch zugreifende Anwendungsfrontends und Fachdiensten MUSS hierdurch möglich sein - alternativ ist diese Adressierung auch durch den DDoS-mitigierenden Dienstleister erlaubt.
[<=]

A_22695 - Mindestabstand für Georedundanz des sektoralen Identity Provider
Ab dem 31.12.2023 MUSS der Anbieter des sektoralen Identity Provider seinen Dienst an zwei Standorten gemäß A_22692 betreiben, wobei eine Unterschreitung des Abstandes von 100 km gemäß A_22692 nicht zulässig ist.**[<=]**

A_22506 - Unabhängiges Betriebspersonal pro Standort des sektoralen Identity Provider
Der Anbieter des sektoralen Identity Provider MUSS pro Standort ein unabhängiges Betriebspersonal vorhalten, um die Risiken der Standortvorgaben des BSI tragen zu können.**[<=]**

A_22508 - Ausschluss von nicht erlaubten Authenticator-Modul Versionen (Rohdatenerfassung v.02)
Der sektorale Identity Provider MUSS von ihm nicht erlaubte Authenticator-Module (anhand der Versionsnummern) mit dem Status-Code 79105 ablehnen, von einer Kommunikation mit dem sektoralen Identity Provider ausschließen und diesen Verbindungsversuch in den Rohdaten protokollieren.**[<=]**

A_22509 - Ausschluss bei fehlenden Authenticator-Modul Versionsnummern (Rohdatenerfassung v.02)
Der sektorale Identity Provider MUSS Authenticator-Module mit fehlenden Versionsnummern mit dem HTTP-Status-Code 403 ablehnen, von einer Kommunikation mit dem sektoralen Identity Provider ausschließen und diesen Verbindungsversuch in den Rohdaten protokollieren.**[<=]**

A_22931 - Zu verwendender HTTP-Header user-agent des Clientsystems (App, Primärsystem)
Das Clientsystem MUSS in alle HTTP-Requests an Dienste der TI im äußeren Http-Request den HTTP-Header user-agent gemäß [RFC7231] im JSON-Format mit:
`{"HN": "$Herstellername", "PN": "$Produktname", "PV": "$Produktversion"}` gemäß der Produktidentifikation befüllen:

- <Herstellername> gemäß eigener Definition, Länge 1-20 Zeichen, Zeichenvorrat[0-9a-zA-Z\-\.]
- <Produktname> gemäß eigener Definition, Länge 1-20 Zeichen, Zeichenvorrat[0-9a-zA-Z\-\.]
- <Produktversion> gemäß Produktidentifikation im Format "H.N.U-P": Hauptnummer, Nebennummer, Unternummer, Patchlevel (jeweils maximal zweistellig, numerisch)

[<=]

Hinweis:
Die client_id wird dem IDP außerhalb des user-agent im Verbindungsaufbau bekannt gegeben.
Bei Erhöhung der übergeordneten Nummer in der Produktversion, werden alle untergeordneten Nummern auf "0" gesetzt.

A_22253 - Ausschluss bestimmter Authenticator-Modul Versionen von der Kommunikation
Der sektorale Identity Provider MUSS die vom Authenticator-Modul mitgeteilte Versionsnummer erkennen und festgelegte Versionsnummern über eine blockinglist von einer Kommunikation ausschließen können. Der sektorale Identity Provider MUSS in diesen Fällen mit dem HTTP-Status-Code 79105 ablehnen eine entsprechende Fehlermeldung an das Clientsystem geben.**[<=]**

A_22254-01 - Ausschluss von Authenticator-Modul Versionen (Rohdatenerfassung v.02)
Der sektorale Identity Provider MUSS auf Anweisung der gematik Authenticator-Module mit bestimmten Versionsnummern von einer Kommunikation mit dem sektoralen Identity Provider ausschließen und diesen Verbindungsversuch in den Rohdaten protokollieren.**[<=]**

3.2 Vertrauenswürdige Ausführungsumgebung

In diesem Abschnitt werden die Anforderungen an den sektoralen IDP zur Umsetzung einer Vertrauenswürdigen Ausführungsumgebung (VAU) dargestellt. Die VAU dient der datenschutzrechtlich zulässigen und sicheren Verarbeitung von schützenswerten Klartextdaten innerhalb des sektoralen IDP sowie dem technischen Ausschluss einer Profilbildung durch den Anbieter bzw. Betreiber. Sie verhindert ein Eingreifen des Anbieters in den sicheren Betrieb und die Manipulation von Daten. Die VAU stellt dazu

Verarbeitungskontexte (d. h. Instanzen der VAU) bereit, in denen die Verarbeitung sensibler Daten im Klartext erfolgen kann. Diese Verarbeitungskontexte sind entsprechend zu schützen. Im Anhang C ist u.a. ein Beispiel für eine RZ-Lösung einer VAU beschrieben.

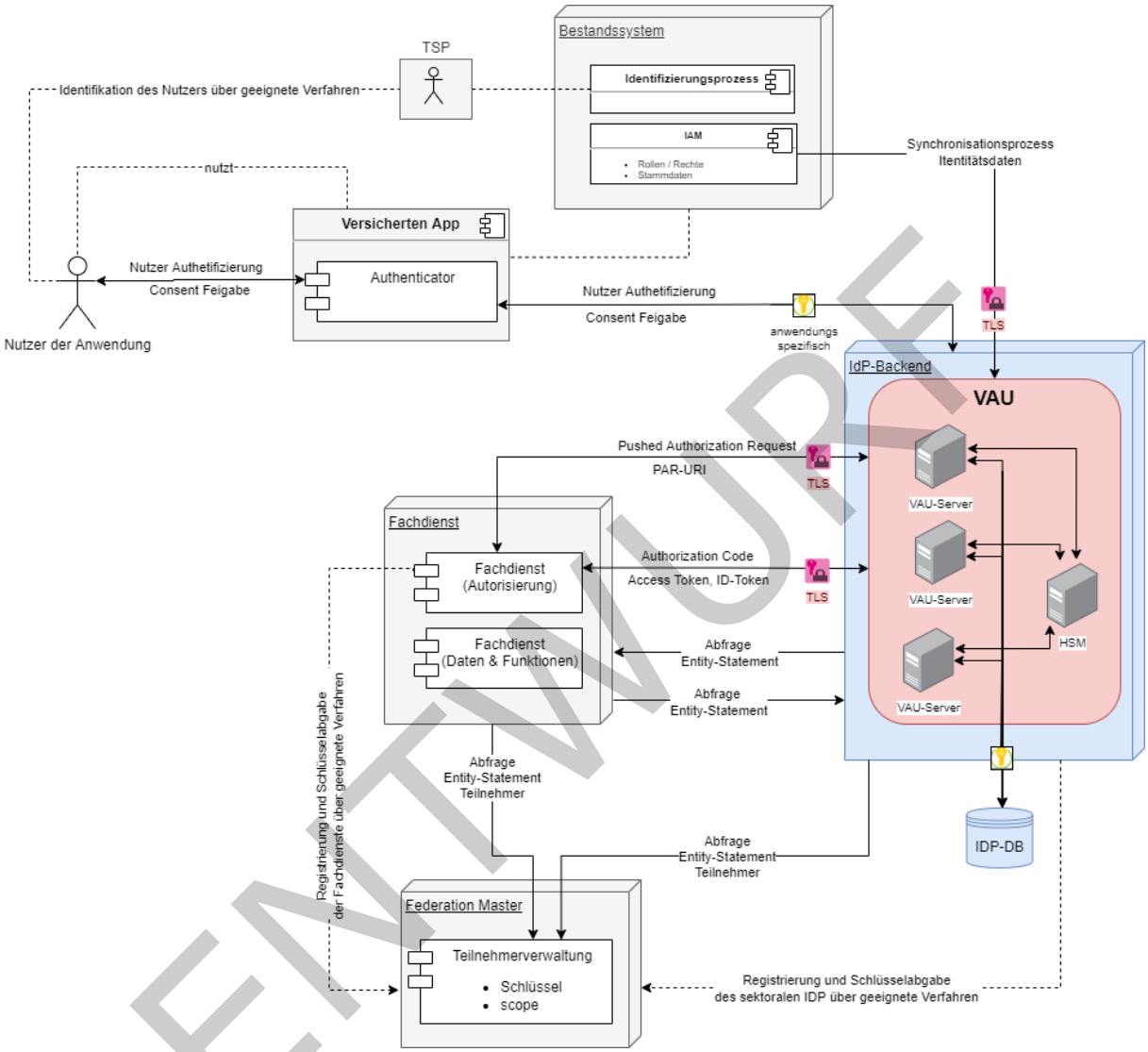

Abbildung 4: Schnittstellen der in der VAU laufenden Komponente des sektoralen IDP

Tabelle 3: Vorgaben für die im sektoralen IDP befindlichen Endpunkte zur Ausführung in einer VAU

Schnittstelle	Gegenstelle	Beschreibung	VAU Ausführung

Pushed Authorization Request (PAR)	Fachdienst Authorization-Server	Der Pushed Authorization Request enthält Informationen zum anfragenden Fachdienst und zum `scope` der angeforderten Daten des Nutzers.	zwingend
Einlösen des Authorization Code	Fachdienst Authorization-Server	Der Token-Request zum Einlösen des Authorization Code enthält Informationen zum Fachdienst. Der Response auf den Request enthält Informationen zum Nutzer.	zwingend
Abruf selbstsigniertes Entity Statement	Fachdienst Authorization-Server	Der Fachdienst bezieht die Konfigurationsparameter, Adressen und Schlüssel des sektoralen IDP	optional
Abruf Entity Statement zur Teilnehmerauskunft	Federation Master	Der Schlüssel des Federation Master zum Verifizieren der von diesem signierten Entity Statements wird sicher verwahrt.	optional
Authentifizierung	Authenticator-Modul auf Endgerät des Nutzers	Die Ausprägung der Schnittstelle kann anwendungsspezifisch gestaltet werden.	optional
Consent-Freigabe und Initialer Authorization Request	Authenticator-Modul auf Endgerät des Nutzers	Es muss nachprüfbar gewährleistet sein, dass der Betreiber des sektoralen IDP keinen Zugriff auf die über die Schnittstelle transportierten Inhalte bezüglich des Anfragenden Dienstes erlangen kann.	zwingend

Aktualisierung der Identitätsdaten im sektoralen IDP	Anwendungssystem, welchen die Identitäten der Versicherten verwaltet	Die Aktualisierung des Datenbestandes des sektoralen IDP erfolgt durch das Bestandssystem der jeweiligen attributbestätigenden Stelle.	zwingend
Ablage und Abfrage der vom sektoralen IDP verwalteten Identitäts- und Nutzerdaten	Datenbank des sektoralen IDP	Die vom sektoralen IDP verwalteten Identitäts- und Nutzerdaten liegen verschlüsselt in einer Datenbank außerhalb der Vertrauenswürdigen Ausführungsumgebung.	zwingend

3.2.1 Verarbeitungskontext

Die Gesamtheit aus der für eine Klartextverarbeitung erforderlichen Software, dem für eine Klartextverarbeitung genutzten physikalischen System sowie den für die Integrität einer Klartextverarbeitung erforderlichen organisatorischen und physischen Rahmenbedingungen bildet den Verarbeitungskontext der Vertrauenswürdigen Ausführungsumgebung (VAU). Zur Vertrauenswürdigen Ausführungsumgebung gehören neben den Verarbeitungskontexten alle für ihre Erreichbarkeit und betriebliche Steuerung erforderlichen Komponenten. Der Verarbeitungskontext grenzt sich von allen weiteren, im betrieblichen Kontext beim Anbieter des sektoralen IDP vorhandenen Systemen und Prozessen dadurch ab, dass die sensiblen Klartextdaten von Komponenten innerhalb des Verarbeitungskontextes aus erreichbar sind oder sein können, während sie dies von außerhalb des Verarbeitungskontextes nicht sind. Sensible Daten verlassen den Verarbeitungskontext ausschließlich gemäß wohldefinierten (Zugriffs-)Regeln und in verschlüsselter Form.

Umsetzungsempfehlungen für die Realisierung einer Vertrauenswürdigen Ausführungsumgebung finden sich im Anhang C.

Offener Punkt: Die Prüfung der Anforderung an den Betrieb VAU und Umsetzungsvorschläge bzw. -hinweise in cloud-Infrastrukturen sind derzeit in Arbeit. Details dazu werden diesem Kapitel später hinzugefügt.

A_22864 - Umsetzung der fachlichen Operationen in einer Vertrauenswürdigen Ausführungsumgebung (VAU)
Der sektoraler IDP MUSS die Verarbeitung aller fachlichen Operationen in einer Vertrauenswürdigen Ausführungsumgebung umsetzen. Die HTTP-Verbindungen zwischen Fachdiensten und sektoralem IDP MÜSSEN als TLS-Verbindungen ausgelegt

werden, welche innerhalb der VAU terminieren.
[<=]

A_23018 - Anforderungen an die Umsetzung der fachlichen Operationen in einer Vertrauenswürdigen Ausführungsumgebung (VAU)

- Aus den Daten, welche zum Zweck eines Reporting an die gematik erstellt werden, DARF es NICHT möglich sein, dass eine Zuordnung von Fachdiensten zu einzelnen Authentisierungen oder Nutzern durchgeführt werden kann.
- Die Verknüpfung einer Nutzeridentität / Authentisierung mit zu bestätigenden Attributen DARF sowohl für Dritte als auch den Betreiber selbst NICHT einsehbar sein.

[<=]

A_23002 - sicherer Betrieb der fachlichen Operationen in einer Vertrauenswürdigen Ausführungsumgebung (VAU)
Der Anbieter des sektoraler IDP MUSS die Verarbeitung aller fachlichen Operationen in einer Vertrauenswürdigen Ausführungsumgebung umsetzen.**[<=]**

A_23019 - Anforderungen an den sicherer Betrieb der fachlichen Operationen in einer Vertrauenswürdigen Ausführungsumgebung (VAU)

- Sowohl Dritte als auch der Betreiber selbst DARF NICHT Zugriff auf das Schlüsselmaterial der TLS-Verbindungen haben.
- Sowohl Dritte als auch der Betreiber selbst DARF NICHT Zugriff auf die für die Signatur von `ID_TOKEN` verwendeten Schlüssel haben.
- Sowohl Dritte als auch der Betreiber selbst DARF NICHT Zugriff auf die im `AUTHORIZATION_CODE` und in der Request-URI kodierten Informationen haben.

[<=]

Hinweis 1: Siehe in diesem Zusammenhang auch A_23031 - TLS-Verbindung Authenticator-Modul - Vertrauenswürdige Ausführungsumgebung.

Hinweis 2: Ein Logging zur Betriebsüberwachung und Fehleranalyse ist zulässig, darf jedoch keine Identifikation des genutzten Fachdienstes zulassen.

A_22959 - Prozess zur Consent-Freigabe durch den Nutzer
Der Prozess zur Freigabe des Consent durch den Nutzer MUSS zwischen Authenticator-Modul und sektoralem IDP verschlüsselt und nicht einsehbar für Dritte oder den Betreiber selbst erfolgen.
[<=]

3.2.2 Ausschluss von nicht autorisierten Zugriffen aus dem Betriebsumfeld

Der Schutzbedarf der in der VAU verarbeiteten Klartextdaten erfordert den technischen Ausschluss von Zugriffen des Anbieters. Dies umfasst insbesondere Zugriffe durch Personen aus dem betrieblichen Umfeld des Anbieters.

A_22829 - Anbieter sektoraler IDP Speicherung Schlüsselmaterial in HSM

Der Anbieter des sektoralen IDP MUSS das private Schlüsselmaterial für kryptografische Verfahren in einem HSM speichern, dessen Eignung durch eine erfolgreiche Evaluierung nachgewiesen wurde. Als Evaluierungsschemata kommen dabei Common Criteria, ITSEC oder Federal Information Processing Standard (FIPS) in Frage.
Die Prüftiefe MUSS mindestens:

1. FIPS 140-2 Level 3,
2. Common Criteria EAL 4+ mit hohem Angriffspotenzial oder
3. ITSEC E3 der Stärke „hoch" entsprechen.

[<=]

A_22830 - sektoraler IDP – Verarbeitungskontext der VAU
Der Verarbeitungskontext des sektoralen IDP MUSS sämtliche physikalischen Systemkomponenten sowie sämtliche Softwarekomponenten umfassen, deren Sicherheitseigenschaften sich auf den Schutz der personenbezogenen Daten vor Zugriff durch Unbefugte bei ihrer Verarbeitung im Klartext auswirken können.**[<=]**

A_22840 - Verschlüsselung von außerhalb des Verarbeitungskontextes der VAU gespeicherten Daten
Der Verarbeitungskontext des sektoralen IDP MUSS sicherstellen, dass sämtliche schützenswerten Daten vor einer Speicherung außerhalb der VAU verschlüsselt werden. Dies betrifft auch Daten zu Logging und Protokollierung.
[<=]

A_22841 - Ableitung der Persistenzschlüssel durch ein HSM
Der Verarbeitungskontext des sektoralen IDP MUSS die zur Verschlüsselung, von außerhalb des Verarbeitungskontextes der VAU gespeicherten Daten, verwendeten Schlüssel von einem HSM innerhalb der VAU abrufen.
[<=]

A_22842 - Ableitung der Persistenzschlüssel
Das HSM der VAU des sektoralen IDP MUSS eine Schnittstelle zur Ableitung eines symmetrischen Schlüssels für die Persistierung der Daten bereitstellen. Das HSM der VAU des sektoralen IDP MUSS zur Ableitung des jeweiligen Schlüssels ein nach der ersten Erstellung unveränderliches Merkmal als Ableitungsparameter verwenden.
[<=]

A_22843 - Geschützte Weitergabe von Daten an autorisierte Nutzer durch die VAU
Der Verarbeitungskontext des sektoralen IDP MUSS sicherstellen, dass sämtliche schützenswerten Daten ausschließlich über sichere Verbindungen an autorisierte Nutzer und Fachdienste weitergegeben werden.
[<=]

A_22844 - Transportverschlüsselte Übertragung von Daten mit Fachdiensten
Der Verarbeitungskontext des sektoralen IDP MUSS sicherstellen, dass er nur transportverschlüsselt mit Fachdiensten und Authenticator-Modulen kommuniziert.
[<=]

A_22845 - Transportverschlüsselte Übertragung von Daten mit Quellsystemen
Der Verarbeitungskontext des sektoralen IDP MUSS sicherstellen, dass er nur transportverschlüsselt mit den Quellsystemen der von ihm verwalteten Identitäten kommuniziert.
[<=]

Hinweis: für die Qualität der Transportverschlüsselung gelten die Anforderungen aus [gemSpec_Krypt].

A_22847 - Authentisierung gegenüber Clients
Der Verarbeitungskontext des sektoralen IDP MUSS sich gegenüber Clients, welche mit ihm kommunizieren, mit einem TLS-Zertifikat ausweisen, auf dessen privaten Schlüssel der Betreiber des sektoralen IDP keinen Zugriff hat.
[<=]

A_23006 - Subdomäne für Webservice-Endpunkte in der VAU
Der Verarbeitungskontext des sektoralen IDP MUSS diese Endpunkte anbieten:

- Endpunkt für Authorization Requests
- Endpunkt für Pushed Authorization Requests
- Token-Endpunkt

Für die Endpunkte der VAU MUSS eine eigene Subdomäne, welche keine Wildcard-Domäne ist, erstellt werden.
[<=]

Hinweis: Die Erstellung einer eigenen Subdomäne für die VAU eines sektoralen IDP ist notwendig um die Certificate Transparency TLS-Zertifikate im Federation Master effektiv prüfen zu können.

A_22943 - Richtlinien zum TLS-Verbindungsaufbau
Der Anbieter des sektoralen IDP MUSS dafür sorgen, dass der Verarbeitungskontext des sektoralen IDP sich beim TLS-Verbindungsaufbau über das Transportnetz gegenüber dem Client mit einem TLS-Zertifikat eines Herausgebers gemäß [CAB-Forum] authentisiert. Der Anbieter MUSS dafür sorgen, dass das Zertifikat sich an die jeweilige Schnittstelle des Eingangspunkts bindet, damit Clientsysteme beim TLS-Verbindungsaufbau eine vereinfachte Zertifikatsprüfung mit TLS-Standardbibliotheken durchführen können. **[<=]**

A_22980 - Grundlage zur Prüfung der TLS-Zertifikate mittels Certificate Transparency
Der Anbieter des sektoralen IDP MUSS die TLS-Zertifikate, welche in seinem Verarbeitungskontext terminieren, aus einer CA beziehen, welche Certificate Transparency gemäß RFC 6962 / RFC 9162 unterstützt und täglich prüfen und sicherstellen, dass für die zur Verbindungen in den Verarbeitungskontext der VAU vorgesehen Domänen keine unbekannten Zertifikate im Certificate Transparency Log gelistet werden. Im Fehlerfall MUSS ein "Security Incident" (gemäß 3.4 gemRL_Betr_TI) erstellt werden. <=**[<=]**

A_22981 - Grundlage zur Prüfung der TLS-Zertifikate mittels Certification Authority Authorization (CAA) Records
Der Anbieter des sektoralen IDP MUSS für die TLS-Zertifikate welche in seinem Verarbeitungskontext terminieren Certification Authority Authorization (CAA) DNS Resource Records nach RFC 6844 bereitstellen, welche die Validität der ausstellenden CA verifizieren. <=**[<=]**

A_22982 - Bereitstellung der öffentlichen Schlüssel der TLS-Zertifikate
Der Anbieter des sektoralen IDP MUSS die öffentlichen Schlüssel der TLS-Zertifikate, welche in seinem Verarbeitungskontext terminieren, dem Federation Master bereitstellen. Der organisatorische Prozess zur Schlüsselübergabe ist in [gemSpec_IDP_FedMaster] beschrieben.**[<=]**

Hinweis: Auf diesem Weg kann der Federation Master verifizieren, dass keine TLS-Zertifikate für diese Adressen erstellt werden deren privater Schlüssel nicht nachgewiesenermaßen im Verarbeitungskontext der VAU liegt. Der Federation Master bietet hierzu einen organisatorischen Prozess an.

A_22848 - Isolation zwischen Datenverarbeitungsprozessen mehrerer Verarbeitungskontexte der VAU
Die VAU des sektoralen IDP MUSS die in ihr ablaufenden Verarbeitungen für die Daten eines Verarbeitungskontextes von den Verarbeitungen für die Daten anderer Verarbeitungskontexte in solcher Weise trennen, dass mit technischen Mitteln ausgeschlossen wird, dass die Verarbeitungen eines Verarbeitungskontextes schadhaft auf die Verarbeitungen eines anderen Verarbeitungskontextes einwirken können.
[<=]

Hinweis: Da der Verarbeitungskontext der VAU des sektoralen IDP für jede fachliche Operation neu aufgebaut werden muss, ist ein Low-Level-Mechanismus zur Kontextseparation aus Gründen der Performance bzw. Skalierung nicht zwingend vorgeschrieben. Der hier geforderte Separationsmechanismus kann daher auch als Server-Thread, Worker, o. Ä. ausgeführt sein, solange für den dadurch gebildeten Verarbeitungskontext die geforderte Separation nachgewiesen werden kann. Dies setzt voraus, dass die Umsetzung der Verarbeitungskontexte und der in ihnen ablaufenden Verarbeitungsvorgänge technisch möglichst einfach und nachvollziehbar gestaltet ist.

A_22849 - Isolation der VAU von Datenverarbeitungsprozessen des Anbieters
Die VAU des sektoralen IDP MUSS die in ihren Verarbeitungskontexten ablaufenden Datenverarbeitungsprozesse von allen sonstigen Datenverarbeitungsprozessen des Anbieters trennen und damit gewährleisten, dass sowohl Dritte als auch der Betreiber des sektoralen IDP selbst vom Zugriff auf die in der VAU verarbeiteten schützenswerten Daten ausgeschlossen ist.
[<=]

Hinweis: Für die Separation zwischen Verarbeitungskontexten und Verarbeitungsprozessen des Anbieters, die der betrieblichen Steuerung des Systems dienen, ist eine Low-Level Separation anzustreben, da - im Unterschied zur Separation zwischen Verarbeitungskontexten - hier technisch sehr verschiedene Aspekte getrennt werden müssen.

A_22850 - Ausschluss von Manipulationen an der Software der VAU
Die VAU des sektoralen IDP MUSS eine Manipulation der eingesetzten Software erkennen und eine Ausführung der manipulierten Software verhindern.
[<=]

A_22851 - Ausschluss von Manipulationen an der Hardware der VAU
Die VAU des sektoralen IDP MUSS die Integrität der eingesetzten Hardware schützen und damit insbesondere Manipulationen an der Hardware sowohl durch Dritte als auch der Betreiber des sektoralen IDP ausschließen.
[<=]

A_22852 - Kontinuierliche Wirksamkeit des Manipulationsschutzes der VAU
Die VAU des sektoralen IDP MUSS den Ausschluss von Manipulationen an der Hardware und der Software sowohl durch Dritte als auch der Betreiber des sektoralen IDP mit Mitteln umsetzen, deren dauerhafte und kontinuierliche Wirksamkeit gewährleistet werden kann.
[<=]

A_22853 - Kein physischer Zugang des Anbieters zu Systemen der VAU
Die VAU des sektoralen IDP MUSS mit technischen und/oder organisatorischen Mitteln sicherstellen, dass niemand, auch nicht der Anbieter des sektoralen IDP, während der

Verarbeitung personenbezogener Daten Zugriff auf physische Schnittstellen der Systeme erlangen kann, auf denen eine VAU ausgeführt wird.
[<=]

A_22854 - Nutzdatenbereinigung vor physischem Zugang zu Systemen der VAU
Die VAU des sektoralen IDP MUSS mit technischen und/oder organisatorischen Mitteln sicherstellen, dass physischer Zugang zu Hardware-Komponenten der Verarbeitungskontexte nur erfolgen kann, nachdem gewährleistet ist, dass aus ihnen keine Nutzdaten extrahiert werden können.
[<=]

A_22868 - Private Schlüssel im HSM
Der sektorale IDP MUSS folgende private Schlüssel in einem Hardware Security Module (HSM) erzeugen und anwenden:

- die Schlüssel zur Signatur von Token und Entity Statements
- die Schlüssel der TLS-Zertifikate für die sichere Verbindung zum Verarbeitungskontext

Die Prüftiefe des HSM MUSS dabei den in [A_22829] angegebenen Standards entsprechen.
[<=]

A_22855 - HSM-Kryptographieschnittstelle verfügbar nur für Instanzen der VAU
Die VAU des sektoralen IDP MUSS mit technischen Mitteln, die Manipulationen sowohl durch Dritte als auch der Betreiber des sektoralen IDP ausschließen, gewährleisten, dass nur Instanzen der VAU Zugriff auf die Kryptographieschnittstelle des HSM zur Nutzung des privaten Schlüsselmaterials für ihre TLS-Zertifikate und die Signaturschlüssel für die Token und Entity Statements erhalten können.
[<=]

Hinweis: Falls die Verarbeitungskontexte als Threads, Workers, o. ä. umgesetzt sind und daher gemeinsam in einem übergreifenden Anwendungsprozess ausgeführt werden, kann dieser Anwendungsprozess eine authentisierte Verbindung zur Kryptographieschnittstelle des HSM herstellen und aufrechterhalten, um darüber die Kryptographieschnittstelle des HSM den Verarbeitungskontexten (und nur diesen) lokal zur Verfügung zu stellen.

3.2.3 Konsistenz des Systemzustands, Logging und Monitoring

A_22856 - Konsistenter Systemzustand des Verarbeitungskontextes der VAU
Die VAU des sektoralen IDP MUSS sicherstellen, dass ein konsistenter Zustand des Verarbeitungskontextes auch bei Bedienfehlern oder technischen Problemen immer erhalten bleibt bzw. wiederhergestellt werden kann.
[<=]

Hinweis: Wenn ein Ablauf bzw. Prozess aufgrund eines Fehlers neu gestartet wird, dann entspricht der Zustand des Verarbeitungskontextes exakt dem vor dem Prozesstart vor der Fehlersituation.

A_22857 - Datenschutzkonformes Logging und Monitoring des Verarbeitungskontextes der VAU
Die VAU des sektoralen IDP MUSS die für den Betrieb eines Fachdienstes erforderlichen Logging- und Monitoring-Informationen in solcher Art und Weise erheben und verarbeiten, dass mit technischen Mitteln ausgeschlossen ist, dass dem Anbieter des

sektoralen IDP oder Dritten vertrauliche oder zur Profilbildung geeignete Daten zur Kenntnis gelangen.
[<=]

3.3 Betriebliche Unterstützung des Probings

A_22567 - Informationsverpflichtung über Mandanten des Anbieters sektoraler IDP
Der Anbieter des sektoralen IDP MUSS der gematik initial zur Zulassung und danach jeweils bei Änderungen tagesaktuell die Liste der Mandanten (Versicherungen) mitteilen, für deren Versicherte er den Dienst anbietet.
Dabei MUSS der Anbieter des sektoralen IDP die IK-Nummer der Kasse und die Telematik-ID der Kasse aus dem "Verfahren zur Beantragung*Basis-Consumer*" nennen, in welchem der GKV-SV die Echtheit der Kasse bereits bestätigt hat.
Der Meldung ist die Beauftragung durch die Kasse als Erklärung schriftlich beizufügen.
[<=]

Hinweis: Die Benachrichtigung an die gematik kann per E-Mail (S/MIME) an transition@gematik.de erfolgen.

A_22534 - Bereitstellung von Validierungsidentitäten durch den sektoralen IDP für das Service Monitoring
Der Anbieter des sektoralen IDP MUSS der gematik mindestens in Summe zwei Validierungsidentitäten zur Verfügung stellen. Mit diesen gültigen Validierungsidentitäten MUSS die Erreichbarkeit und Funktionalität des sektoralen IDP im zyklischen 5-Minuten Intervall durch die gematik prüfbar sein.
[<=]

A_22535 - Kein Personenbezug der Validierungsidentitäten
Der Anbieter des sektoralen IDP DARF in den herauszugebenen Validierungsidentitäten nur Daten verwenden welche NICHT zu echten Personen korreliert werden können.**[<=]**

Hinweis: Dies kann beispielsweise durch eine administrativ ungültige Telematik-ID / KVNR erfolgen. Die Prüfziffer muss dabei korrekt bleiben.

3.4 Testseitige Vorgaben an den sektoralen IDP

Föderiertes Identitätsmanagement stellt einen der ersten Schritte auf dem Weg von der bestehenden TI 1.0 mit ihren drei getrennten Umgebungen hin zu einer cloudbasierten TI 2.0, in der die Dienste über das Internet erreichbar sind, dar. Daher müssen die sektoralen IDPs einerseits mit bestehenden Strukturen und Konzepten verträglich sein, andererseits zukünftige Entwicklungen unterstützen. Dieser Konflikt zeigt sich deutlich in den testseitigen Anforderungen an den Anbieter und das Produkt. Übergreifende Anforderungen aus [gemKPT_Test] passten inhaltlich nicht mehr vollständig und neue Anforderungen werden notwendig.

3.4.1 Testinstanzen

Damit die gematik Zulassungstests durchführen und andere Hersteller frühzeitig mit den sektoralen IDP integrieren können, werden neben der produktiven Instanz auch Testinstanzen benötigt. Das sind eine Instanz für entwicklungsbegleitende Integrationstest - analog zur RU in der TI 1.0 und im folgenden Integrationsinstanz genannt - und eine sehr produktionsnahe Instanz für Abnahmen und Zulassungstests - analog zur TU in der TI 1.0 und im folgenden Staginginstanz genannt.

A_23096 - Übertragung RU/TU auf Testinstanzen im Internet
Der Hersteller des sektoralen IDPs MUSS die dem Produkt zugeordneten Anforderungen aus dem [gemKPT_Test] sinngemäß von den bestehenden Formulierungen mit Test- und Referenzumgebung auf Staging- bzw. Integrationsinstanz übertragen. Unklare Fälle sind mit dem Testmanagement der gematik zu klären.**[<=]**

3.4.1.1 zentrale Komponente

A_23053 - Bereitstellung von Testinstanzen
Der Anbieter des sektoralen IDPs MUSS nach der Zulassung neben der produktiven Instanz weitere Testinstanzen des sektoralen IDPs bereitstellen. Das sind zunächst eine Integrationsinstanz und eine Staginginstanz.
[<=]

A_23054 - Skalierung von Testinstanzen
Der Anbieter des sektoralen IDPs MUSS seine Testinstanzen skalieren können. Das heißt, es müssen nach Anforderung der gematik temporär weitere Instanzen bereitgestellt und vorhandene Instanzen an steigende Lasten angepasst werden können.
[<=]

A_23055 - Aufbau Integrationsinstanz
Der Hersteller des sektoralen IDPs MUSS die Integrationsinstanz iterativ aufbauen und der gematik frühzeitig einen Zugriff auf Zwischenstände ermöglichen.
[<=]

A_23056 - Bereitstellung von Staginginstanzen
Der Hersteller des sektoralen IDPs MUSS die Staginginstanz zur Zulassung bereitstellen.
[<=]

A_23057 - Version der Staginginstanz
Der Anbieter des sektoralen IDP SOLL dafür sorgen, dass die Version der Staginginstanz - außer für die Abnahme einer neuen Version - der der produktiven Instanz entspricht.
[<=]

A_23058 - Änderung der Version der Integrationsinstanz
Der Hersteller des sektoralen IDPs KANN die Version der Integrationsinstanz während der Entwicklung nach Absprache mit der gematik ohne Change-Prozess ändern. Downtimes MÜSSEN dabei der gematik ankündigt werden. Dadurch sollen schnelle Feedbackschleifen während der Entwicklung ermöglicht werden.
[<=]

A_23059 - Bereitstellung Simulation (Mock) der Außenschnittstellen
Der Anbieter des sektoralen IDPs MUSS der gematik spätestens unmittelbar nach Zulassung eine Simulation seines sektoralen IDPs mit identischen Außenschnittstellen bereitstellen, damit andere Hersteller diese frühzeitig und offline in ihrem Entwicklungsprozess verwenden können.
[<=]

3.4.1.2 Authenticator-Modul

A_23060 - Testversion Authenticator-Moduls für Testinstanzen
Der Anbieter des sektoralen IDPs MUSS eine Testversion seines Authenticator-Moduls in einer App bereitstellen, die mit allen Testinstanzen des sektoralen IDPs genutzt werden kann. Das können verschiedene Apps oder eine konfigurierbare sein. Die Testversion MUSS kurzfristig und auf Anfrage an die gematik, aber auch an Dritte - z. B. DIGA-Hersteller - bereitgestellt werden.
[<=]

A_23061 - Betriebssysteme der Testversion des Authenticator-Moduls
Der Anbieter des sektoralen IDPs MUSS eine Testversion seines Authenticator-Moduls in einer App für alle von ihm produktiv unterstützten Betriebssysteme bereitstellen.
[<=]

A_23062 - Funktionsumfang der Testversion des Authenticator-Moduls
Der Anbieter des sektoralen IDPs MUSS eine Testversion des Authenticator-Moduls bereitstellen, die funktional der Produktivversion entspricht.
[<=]

3.4.2 Testidentitäten

Um eine Nutzung der Testinstanzen eines sektoralen IDPs in produktübergreifenden Integrationstests anderer Hersteller und bei Zulassungstests durch die gematik zu ermöglichen, müssen sie einen Satz an Testidentitäten bereitstellen.

A_23063 - Bereitstellung Testidentitäten
Der Anbieter des sektoralen IDPs MUSS Testidentitäten in allen seinen Testinstanzen bereitstellen. Dabei werden sowohl bereits aktivierte Identitäten (also Identitäten, für die die Identifikation durchgeführt wurde und die direkt über das Authenticator-Modul nutzbar sind) als auch noch zu aktivierende Identitäten benötigt. Der Anbieter des sektoralen IDPs MUSS zunächst je 50 aktivierte und noch zu aktivierende Testidentitäten bereitstellen. Auf Anfrage der gematik MÜSSEN bei Bedarf weitere Testidentitäten angelegt werden.
[<=]

A_23065 - Bereitstellung Authentisierungsmöglichkeiten für den Einsatz in automatisierten Tests
Der Anbieter des sektoralen IDPs MUSS zusätzlich zu den produktiv verwendeten Authentisierungsverfahren "einfache" Authentisierungsmöglichkeiten, die einen Einsatz in automatisierten Tests ermöglichen, für die Testidentitäten im Zusammenspiel mit seinem Authenticator-Modul bereitstellen.
[<=]

4 Funktionsmerkmale

4.1 Entity Statement des sektoralen IDP

Das Entity Statement enthält die Metadaten und Adressen des sektoralen IDP, sowie seine verwendeten kryptographischen Identitäten.

A_22662 - Registrierung beim Federation Master durch organisatorischen Prozess
Der Anbieter des sektoralen IDP MUSS seine öffentlichen Schlüssel für die Signatur des selbst signierten Entity Statement über einen vom Federation Master angebotenen organisatorischen Prozess bei diesem bekannt machen.**[<=]**

Hinweis: Schlüssel für Signatur von Entity Statement müssen nicht in VAU liegen. Hier kann der Anbieter einen nicht weiter vorgegenen Prozess etablieren.

A_22643 - Entity Statement des sektoralen IDP
Der sektorale IDP MUSS eine selbst signiertes Entity Statement gemäß [OpenID Connect Federation 1.0#entity-statement] bereitstellen und im Internet verfügbar machen. Mindestens die in den Tabellen *Header Entity Statement des sektoralen IDP* und *Body Entity Statement des sektoralen IDP* in 7.1.4- Detailinformationen zum App-App-Flow genannten Daten und Werte MÜSSEN im Entity Statement enthalten sein.

[<=]

A_22710 - Vorlaufzeit bei geplantem Schlüsselwechsel
Der Anbieter des sektoralen IDP MUSS Signaturschlüssel im Rahmen eines geplanten Schlüsselwechsels mindestens 24 Stunden vor Verwendung in seinem jwks-Schlüsselsatz veröffentlichen und beim Federation Master über einen organisatorischen Prozess hinterlegen.**[<=]**

Hinweis: Nicht betroffen von dieser Anforderung sind kurzfristig notwendige Schlüsselwechsel, z. B. aufgrund von Sicherheitsvorfällen. Diese Maßnahmen sind beispielsweise über security incidence abzuwickeln. Die Bearbeitung solcher kurzfristigen Schlüsselwechsel muss die Aktualisierung beim Federation mitberücksichtigen, da ansonsten zu Verarbeitungsfehlern wegen ungültiger Schlüssel kommen kann.

A_22711 - Regelmäßige Erneuerung des Statement
Das Entity Statement des sektoralen IDP MUSS täglich neu ausgestellt werden.
[<=]

A_23010 - Maximale Gültigkeitsdauer eines Entity Statement
Die maximale Gültigkeitsdauer eines Entity Statement (Attributewerte `iat` und `exp` im Entity Statement) DARF 30 Stunden NICHT überschreiten.**[<=]**

A_22644 - Entity Statement - Prüfung angebotener URLs
Der sektorale IDP MUSS alle von ihm im Entity Statement angebotenen URLs stündlich auf bloße Erreichbarkeit prüfen.**[<=]**

4.2 API-Endpunkte des sektoralen IDP

4.2.1 Anforderung an die Schnittstelle zum Authorization-Server des Fachdienstes

A_22649 - Anfragen bekannter Clients
Der Produkttyp sektoraler IDP MUSS Authorization Request von Clients mit dem http-Statuscode 401 (Unauthorized) ablehnen, wenn diese nicht in der Föderation registriert sind.**[<=]**

A_22921 - Ablehnung eines nicht identifizierten Authenticator durch den IDP
Der Produkttyp sektoraler IDP MUSS die Anfrage mit dem http-Statuscode 412 (Precondition failed) ablehnen, sofern vom Client der UserAgent fehlerhaft (entgegen A_20015-01) geliefert wird und deshalb durch die gematik gesperrt wurden. **[<=]**

A_22922 - Anfragen veralteter Authenticator Versionen
Der Produkttyp sektoraler IDP MUSS Authorization Request von veralteten Authenticator Versionen mit dem http-Statuscode 426 (Upgrade Required) ablehnen, wenn diese aus Kompatibilitätsgründen durch die gematik gesperrt wurden.
[<=]

A_22650 - automatische Registration von Fachdiensten
Der sektorale IDP MUSS eine automatische Registrierung eines Fachdienstes bei Übermittlung eines Authorization Request mit `private_key_jwt` gemäß [OpenID Connect Federation 1.0 (section-10.1)] durchführen. Sofern dieser Dienst nicht bereits am IDP Registriert wurde. Nach Abruf des Entity Statement des Fachdienstes beim Fachdienst selbst MUSS der sektorale IDP beim Federation Master dessen Entity Statement zum Fachdienst gemäß [OpenID Connect Federation 1.0 (section-7.1)] abrufen und seine Zugehörigkeit zur Föderation bestätigen zu lassen. Anschließend registriert der sektorale IDP den Fachdienst und importiert dessen Schlüssel für die Verschlüsselung von Token.
[<=]

Hinweis: Wenn eine signed_jwks_uri im Entity Statement des Fachdienstes angegeben ist, müssen auch diese Schlüssel importiert werden. Sowohl dies als auch die Nutzung des jwks im Statement selbst muss unterstützt werden.

4.2.2 PAR - Endpunkt

Am PAR-Endpunkt des sektoralen IDP werden Anfragen der Authorization-Servern eines Fachdienstes eingereicht und verifiziert. Inhalt der Anfrage ist unter anderem:

- Die `client_id` des anfragenden Fachdienstes sowie dessen öffentlicher Authentierungsschlüssel.
- Die `redirect_uri`, an welche der Authorization Request beantwortet werden soll.
- Der über das eigene `CODE_VERIFIER` [Proof Key for Code Exchange by OAuth Public Clients (section-4.1)] gebildete HASH `CODE_CHALLENGE` [Proof Key for Code Exchange by OAuth Public Clients (section-4.2)] mit Angabe des Algorithmus

`code_challenge_method` [Proof Key for Code Exchange by OAuth Public Clients (section-4.3)] entsprechend dem gewählten Authorization Code Flow (response_type=code).

- Der `state`-Parameter [OAuth 2.0 for Native Apps (section-8.9)] wird genutzt, um CSRF (Cross-Site-Request-Forgery) zu verhindern.
- Der `scope` der Anfrage, welcher einen definierten Satz von benötigten Attributen für die entsprechende Anwendung beinhaltet.

Der PAR-Endpunkt des sektoralen IDP nimmt den Pushed Authorization Request [OAuth 2.0 Pushed Authorization Requests] des Authorization-Server eines Fachdienstes entgegen. Der am PAR-Endpunkt des sektoralen IDP eingehende Request wird validiert, um frühzeitig unautorisierte Abfragen zu verhindern.

Ist der Pushed Authorization Request geprüft und valide, erstellt der PAR-Endpunkt des sektoralen IDP eine Request-URI. Diese wird im weiten Ablauf für die Nutzerauthetifizierung benötigt. Die Request-URI und deren Gültigkeitsdauer sind Parameter der Antwort des sektorale IDP auf den eingegangenen Request.

4.2.2.1 PAR-Endpunkt Eingangsdaten

A_22651 - Parameter des Pushed Authorization Request durch den sektoralen IDP
Der sektorale IDP MUSS die Annahme von Pushed Authorization Request gemäß [OAuth 2.0 Pushed Authorization Requests#section-2] unterstützen und den Endpoint für Pushed Authorization Request im Entity Statement des sektoralen IDP `pushed_authorization_request_endpoint` bekanntgeben.
Der sektorale IDP MUSS mindestens die in der 7.1.4- Detailinformationen zum App-App-Flow Tabelle *Parameter Pushed Authorization Request* dargestellten Parameter im Pushed Authorization Request des Authorization-Server eines Fachdienstes annehmen.

[<=]

A_22966 - Prüfung eingehender Pushed Authorization Request durch den sektoralen IDP
Der sektorale IDP MUSS die eingehende Pushed Authorization Request validieren und invalide Request gemäß [OAuth 2.0 Pushed Authorization Requests#section-2.3] mit einer Fehlernmeldung quittieren. Die Validierung des eingegangenen Pushed Authorization Request schließt die Prüfung der im Request enthaltenen Werte für `redirect_uri` und `scope` gegen die für den Fachdienst zulässigen (d.h. bei der Registrierung gemeldeten) Werte ein.
[<=]

Hinweis: Nach [OpenID Connect Core 1.0# AuthRequest] ist es zulässig, dass ein Client mehrere `redirect_uri` bei der Registrierung hinterlegt. Der sektorale IDP muss laut der OIDC-Spezifikation prüfen, ob die im Request gelieferte `redirect_uri` mit exakt einer der hinterlegten `redirect_uri` übereinstimmt. Die Prüfung muss über eine 'Simple String Comparison' nach [Uniform Resource Identifier (URI)#section-6.2.1] erfolgen.

A_22991 - Prüfung "private_key_jwt" am PAR-Endpunkt des sektoralen IDP
Der PAR-Endpunkt des sektoralen IDP MUSS den im `client_assertion`-Parameter übertragenen `private_key_jwt` wie folgt überprüfen:

- Der Parameter `iss` MUSS der Client-ID des registrierten Fachdienstes entsprechen.
- Der Parameter `aud` MUSS der Issuer-URL des jeweiligen sektoralen IDP entsprechen.
- Die aktuelle Zeit MUSS kleiner als der im Parameter `exp` angegebene Zeitpunkt sein.

[<=]

A_23009 - Prüfung "private_key_jwt" am PAR-Endpunkt des sektoralen IDP - Replay-Schutz
Der am PAR-Endpunkt des sektoralen IDP im `client_assertion`-Parameter übertragenen `private_key_jwt` (`jti claim`) DARF NICHT bereits eingereicht worden sein (Replay-Schutz).
[<=]

4.2.2.2 PAR-Endpunkt Ausgangsdaten

A_22992 - Antwort auf einen eingehenden Pushed Authorization Request durch den sektoralen IDP
Der sektorale IDP MUSS auf einen validen Pushed Authorization Request mit einem http-Statuscode 201 gemäß OAuth 2.0 Pushed Authorization Requests (section-2.2) antworten.
[<=]

A_22993 - Gültigkeit der vom sektoralen IDP erstellten Request-URI
Die Gültigkeit der vom sektoralen IDP erstellten Request-URI DARF NICHT 90 Sekunden überschreiten.
[<=]

4.2.3 Authorization-Endpunkt

Am Authorization-Endpunkt des sektoralen IDP wird in Kommunikation mit dem Authenticator-Modul die Authentisierung des Nutzers durchgeführt.

4.2.3.1 Schnittstelle Authorization-Endpunkt

A_22312-01 - Einhaltung der Standards bei der Realisierung des Authorization-Endpunkts
Der sektorale Identity Provider MUSS die Schnittstelle Authorization-Endpunkt gemäß [RFC6749 - The OAuth 2.0 Authorization Framework (section-3.1)], [RFC8252 - OAuth 2.0 for Native Apps] und [RFC9126 - OAuth 2.0 Pushed Authorization Requests (section-

4)] sowie weitere darin festgelegte Standards implementieren. Hierbei MÜSSEN nur im Rahmen der gemSpec_IDP_Sek relevante Aspekte (Authorization Code Flow ohne User Info Endpoint) berücksichtigt werden.
[<=]

4.2.3.2 Authorization-Endpunkt Ausgangsdaten

Sind alle im `scope` geforderten Attribute vorhanden und die Gültigkeit der Attribute geprüft sowie eine erfolgreiche Authentifizierung des Nutzers erfolgt, erstellt der Authorization-Endpunkt des sektoralen IDP einen `AUTHORIZATION_CODE` und sendet diesen an das Authorization-Server eines Fachdienstes.

A_22324 - Verwendung des Attributes "state" durch sektoralen IDP
Der Authorization-Endpunkt des sektoralen Identity Provider MUSS den `state`-Parameter [RFC6749 # section-10.12] der Anfrage in allen darauf basierenden Responses verwenden.**[<=]**

A_22325-01 - Übermitteln des "AUTHORIZATION_CODE" an den Sender des Requests
Der sektorale Identity Provider MUSS den `AUTHORIZATION_CODE` und den `state` auf demselben Kanal beantworten, auf dem er den Authorization Request empfangen hat.
[<=]

Hinweis: Im Fall des App-App-Flow (7.1 App-App-Flow) und des Web-App-Flow (7.2 Wep-App-Flow) wird der Request durch das Authenticator Modul angenommen und an den sektoralen IDP gestellt. Im Fall des Zwei-Geräte-Flow (7.3 Zwei-Geräte-Flow) wird der Request direkt über den Browser gestellt und damit auch an diesen zurückgeliefert.

4.2.4 Token-Endpunkt

Der Token-Endpunkt des sektoralen IDP nimmt die Anfrage des Authorization-Server eines Fachdienstes entgegen und prüft neben deren Integrität, ob der eingereichte `CODE_VERIFIER` bei Nutzung des Hash-Verfahrens S256 (nach [Proof Key for Code Exchange by OAuth Public Clients (section-4.2)]) zum bitgleichen Hash-Wert führt. Stimmt der Hash-Wert aus dem initialen Aufruf über das Authenticator-Modul - die Code-Challenge - mit dem gebildeten Hash-Wert überein, ist sichergestellt, dass Aufrufer und Initiator identisch sind. Der Token-Endpunkt gibt daraufhin das `ID_TOKEN` an den Authorization-Server des Fachdienstes heraus.

4.2.4.1 Token-Endpunkt Eingangsdaten

A_22653 - Annahme von "AUTHORIZATION_CODE" und "CODE_VERIFIER"
Der Token-Endpunkt des sektoralen IDP MUSS die vom Authorization-Server eines Fachdienstes übertragenen `AUTHORIZATION_CODE` und `CODE_VERIFIER` annehmen. **[<=]**

A_23007 - Gültigkeit des "AUTHORIZATION_CODE"
Die Gültigkeitsdauer eines `AUTHORIZATION_CODE` DARF 90 Sekunden NICHT überschreiten.**[<=]**

A_22321 - Prüfung des "CODE_VERIFIER"

Der Token-Endpunkt des sektoralen Identity Provider MUSS die Überprüfung des `CODE_VERIFIER` gegen die `CODE_CHALLENGE` mit S256 (Algorithmus nach [RFC7636 # section-4.2]) durchführen.**[<=]**

A_22654 - Prüfung "private_key_jwt" am Token-Endpunkt des sektoralen IDP
Der Token-Endpunkt des sektoralen IDP MUSS den im `client_assertion`-Parameter übertragenen `private_key_jwt` wie folgt überprüfen:

- Der Parameter `iss` MUSS der Client-ID des registrierten Fachdienstes entsprechen.
- Der Parameter `aud` MUSS der Issuer-URL des jeweiligen sektoralen IDP entsprechen.
- Die aktuelle Zeit MUSS kleiner als der im Parameter `exp` angegebene Zeitpunkt sein.

[<=]

A_23008 - Prüfung "private_key_jwt" am Token-Endpunkt des sektoralen IDP - Replay-Schutz
Der am Token-Endpunkt des sektoralen IDP im `client_assertion`-Parameter übertragenen `private_key_jwt` (`jti claim`) DARF NICHT bereits eingereicht worden sein (Replay-Schutz).**[<=]**

A_22323 - Protokollierung der Token-Ausgabe in allen Fällen
Der Token-Endpunkt des sektoralen Identity Provider MUSS im Positivfall die Herausgabe der Token und im Negativfall die Token-Anfrage protokollieren. **[<=]**

Das Protokoll wird intern und ggf. für Audits verwendet.

4.2.4.2 Token-Endpunkt Ausgangsdaten

A_22316 - Maximale Gültigkeitsdauer von "ID_TOKEN"
Der sektorale Identity Provider DARF `ID_TOKEN` mit einer Gültigkeitsdauer von mehr als 300 Sekunden (5 Minuten) NICHT ausstellen.**[<=]**

A_22706 - "ID_TOKEN" des sektoralen IDP
Der sektorale IDP MUSS nach erfolgreicher Prüfung des erhaltenen `AUTHORIZATION_CODE` dem aufrufenden Authorization-Server des Fachdienstes ein `ID_TOKEN` gemäß OIDC Standard OpenID Connect Core 1.0 (section-2) mit den angefragten `claims`zurückgeben. **[<=]**

Hinweis: Für nicht vorhandene `scopes` und `claims` deren Weitergabe der Nutzer nicht zugestimmt hat werden im `ID_TOKEN` keine Werte gesetzt.

A_22655 - Signatur des "ID_TOKEN" des sektoralen IDP
Der sektorale IDP MUSS die `ID_TOKEN` unter Verwendung eines privaten Schlüssels der im Entity Statement unter `signed_jwks_uri` referenzierten öffentlichen Schlüssel signieren [OpenID Connect Federation 1.0 (section-4.2)]. **[<=]**

A_22983 - Signaturverfahren für Signatur des "ID_TOKEN" des sektoralen IDP

Das für die Signatur der `ID_TOKEN` zu verwendende Verfahren MUSS ECDSA auf Basis der NIST-Kurve P-256 sein. (vergleiche JSON Web Signature (section-3)).
[<=]

A_22989 - "scopes" und "claims" des sektoralen IDP für Versicherte
Ein sektoraler IDP, welcher den Identitäten für Versicherte verwaltet MUSS mindestens die folgenden `scopes` und `claims` unterstützen:

Tabelle 4: `scopes` und `claims`

Scope	Claim	Wert	Beschreibung
urn:telematik:geburtsdatum	birthdate	string	End-User's birthday, represented as an ISO 8601:2004 [ISO8601-2004] YYYY-MM-DD format. The year MAY be 0000, indicating that it is omitted. To represent only the year, YYYY format is allowed. Note that depending on the underlying platform's date related function, providing just year can result in varying month and day, so the implementers need to take this factor into account to correctly process the dates.
urn:telematik:alter	urn:telematik:claims:alter	string	Alter der Person in Jahren zum Zeitpunkt der Erstellung des Tokens
urn:telematik:display_name	urn:telematik:claims:display_name	string	Analog zu name gemäß Standard Zweck ist die Anzeige in Anwendungen, um eine einheitliche Benutzeransprache zu ermöglichen (End-User's full name in displayable form including all name parts, possibly including titles and suffixes)

urn:telematik:geschlecht	urn:telematik:claims:geschlecht	string	Analog VSDM M =männlich, W = weiblich, X = unbestimmt, D = divers
email	email	string	E-Mail Addresse des Versicherten, wenn bekannt.
	email_verified	boolean	"true", wenn die E-Mail Addresse durch den sektoralen IDP oder die Attributsbestätigende Stelle verifiziert wurde, ansonsten "false".
urn:telematik:versicherter	urn:telematik:claims:profession	string	Für Versicherte 1.2.276.0.76.4.49
	urn:telematik:claims:id	string	Für Versicherte der unveränderliche Anteil der KVNR
	urn:telematik:claims:organization	string	ID oder Name der Attributsbestätigenden Stelle (IK-Nummer der Kasse)

[<=]

A_22990 - Umgang mit Fehlender oder verwehrten Informationen
Ein sektoraler IDP DARF, wenn ihm der Wert eines `claims` nicht vorliegt oder der Nutzer dessen Weitergabe im Consent verweigert hat, diesen `claim` im `ID_TOKEN` NICHT setzen.**[<=]**

4.3 Identifizierung und Authentifizierung des Nutzers

Die Durchführungsverordnung (EU) 2015/1502 [eIDAS 2015/1502] gemäß Artikel 8 Absatz 3 der Verordnung (EU) Nr. 910/2014 [eIDAS 910/2014] legt die Mindestanforderungen an technische Spezifikationen und Verfahren für Vertrauensniveaus elektronischer Identifizierungsmittel fest. Die Vertrauensniveaus der [TR-03107-1] entsprechen im Wesentlichen den eIDAS LOA [TR-03107-1#Anhang A#Tabelle 13].

Im Rahmen der Anbieterzulassung prüft der unabhängige Sicherheitsgutachter, dass die vom Anbieter verwendeten Mechanismen die Mindestanforderungen des jeweiligen Vertrauensniveaus erfüllen.

A_23024 - Interpretation "gematik-ehealth-loa-substancial"
Der Anbieter des sektoralen IDP MUSS `gematik-ehealth-loa-substancial` wie folgt interpretieren.
Der Wert `gematik-ehealth-loa-substancial` entspricht dem Widerstandspotential gegen das Angriffspotential "moderate" nach [ISO18045]. Zertifizierungen von Prozessen oder Prozessbestandteilen auf einem Sicherheitsniveau entsprechend, z. B. nachVerordnung (EU) Nr. 910/2014 an elektronische Identifizierungsmittel (eIDAS) "substancial", BSI TR-03107-1 "substantiell", [ISO29115] LoA 3 oder vergleichbar, können nachgenutzt werden.
[<=]

A_23025 - Interpretation "gematik-ehealth-loa-high"
Der Anbieter des sektoralen IDP MUSS `gematik-ehealth-loa-high` wie folgt interpretieren.
Der Wert `gematik-ehealth-loa-high` entspricht dem Widerstandspotential gegen das Angriffspotential "high" nach [ISO18045]. Zertifizierungen von Prozessen oder Prozessbestandteilen auf einem Sicherheitsniveau entsprechend, z. B. nachVerordnung (EU) Nr. 910/2014 an elektronische Identifizierungsmittel (eIDAS) "High", BSI TR-03107-1 "Hoch", ETSI EN 319 411-1 (REG-6.2.2-05), [ISO29115] LoA 4 oder vergleichbar, können nachgenutzt werden.
[<=]

Hinweis: Im Folgenden wird an den relevanten Stellen ausschließlich `gematik-ehealth-loa-high` oder/und `gematik-ehealth-loa-substancial` verwendet.

A_22987 - Mindestmaß für eine "gematik-ehealth-loa-substantial" Authentisierungsstärke
Der Anbieter des sektoralen IDP MUSS sicherstellen das der `claim acr` nur auf den Wert `gematik-ehealth-loa-substancial` gesetzt wird wenn der Nutzer mindestens auf dem Niveau "substanziell" identifiziert und authentisiert wurde (siehe 4.3-1- Interpretation "gematik-ehealth-loa-substancial").**[<=]**

A_22988 - Mindestmaß für eine "gematik-ehealth-loa-high" Authentisierungsstärke
Der Anbieter des sektoralen IDP MUSS sicherstellen das der `claim acr` nur auf den Wert `gematik-ehealth-loa-high` gesetzt wird wenn der Nutzer auf dem Niveau "hoch" Identifiziert und authentisiert wurde (siehe4.3-2- Interpretation "gematik-ehealth-loa-high"). **[<=]**

Hinweis: weitere Werte des `claim acr` sind zulässig aber werden nicht spezifiziert und auch nicht im Rahmen von Anwendungen der Telematikinfrastruktur genutzt.

Im Rahmen von Anwendungen der TI kommt aktuell nur das Niveau `gematik-ehealth-loa-high` zum Einsatz.

4.3.1 Identifikation des Nutzers

Eine Notifizierung des elektronischen Identifizierungssystems, welches die elektronischen Identifizierungsmittel ausstellt, ist nicht gefordert. Ebenso ist nicht gefordert, dass der Anbieter ein qualifizierter oder nicht-qualifizierter Vertrauensdiensteanbieter gemäß Verordnung (EU) Nr. 910/2014 ist.

A_22865 - Verpflichtende Verfahren zur Identifikation von Nutzern
Der Anbieter des sektoralen IDP MUSS einen organisatorischen Prozess zur Identifikation von Nutzern mit mindestens diesen Identifikationsverfahren zur Nutzeridentifikation anbieten:

- Identifikation mittels elektronischem Identitätsnachweis (online Ausweisfunktion)
- Identifikation mittelt eGK und PIN

[<=]

A_22334-01 - Verifikation des Versicherten vor erster Nutzung
Der Anbieter des sektoralen IDP MUSS sicherstellen, dass der Zuordnungsprozess zwischen einer natürlichen Person und den Daten der Attributbestätigenden Stelle eindeutig ist.
[<=]

A_23102 - Weitere Verfahren zur Identifikation von Nutzern
Wenn der Anbieter des sektoralen IDP weitere Identifikationsverfahren anbietet, dann MUSS derIdentifikationsprozess derart gestaltet werden, dass typischen Angriffen, wie etwa

- Verwendung anderer, abgelaufener, gefälschter oder fremder Identmedien oder
- Manipulationen von digitalen Übertragungswegen bei der Prüfung der natürlichen Person oder des Identmediums,

von Angreifern mit hohem Angriffspotential (vgl. z. B. ISO 18045 Annex B oder vergleichbare Einstufung/Äquivalenzen) standgehalten wird.

Bestehende Zertifizierungen von Prozessen oder Prozessbestandteilen, z. B. nach eIDAS LoA "High", BSI TR-03107-1 "Hoch", ETSI oder vergleichbar, können nachgenutzt werden. **[<=]**

4.3.2 Authentifizierungsverfahren

Nach [TR-03107-1] *"Elektronische Identitäten und Vertrauensdienste im E-Government Teil 1#Tabelle 2: Grundlegende Kriterien für die Vertrauensniveaus"* werden je nach Vertrauensniveau unterschiedliche Anforderungen an die Authentisierung von Nutzern gestellt."

A_22712 - Unterstützung von NFC eGK und PIN
Der Hersteller eines sektoralen IDP MUSS, korrekte Prozessimplementierung zur Erreichung des Vertrauensniveaus „hoch" vorausgesetzt, ein Authentifizierungsverfahren über NFC mittels eGK und PIN unterstützen.**[<=]**

A_22713 - Unterstützung des elektronischen Identitätsnachweis (online-Ausweisfunktion)
Der Hersteller eines sektoralen IDP MUSS ein Authentifizierungsverfahren mittels elektronischem Identitätsnachweis (online-Ausweisfunktion) unterstützen.**[<=]**

A_23026 - Entfernen von Authentifizierungsverfahren, welche die Vorgaben nicht mehr erfüllen
Der Anbieter eines sektoralen IDP MUSS sicherstellen, dass Authentifikationsverfahren entfernt/ausgeschlossen werden, wenn sie das entsprechende Sicherheitsniveau `gematik-ehealth-loa-high` bzw. `gematik-ehealth-loa-substancial` nicht mehr erfüllen.
[<=]

A_22867 - Niederschwellige Authentifizierungsverfahren
Der Anbieter des sektoralen IDP DARF den `claim acr` auf den Wert `gematik-ehealth-loa-high` setzen obwohl des Sicherheitsniveau des Authentisierungsverfahren das Niveau `gematik-ehealth-loa-high` nicht ganz erreicht, wenn eine Einwilligung des Nutzer gemäß Art. 7 DSGVO vorliegt. In diesen Fällen ist der Nutzer über die Risiken einer Abstufung des Authentifizierungsverfahren ausreichend aufzuklären.

Das verwendete Verfahren darf dabei nicht die Anforderungen des Niveau `gematik-ehealth-loa-substancial` unterschreiten.
[<=]

A_23103 - Nutzer Einwilligung Vertrauensniveauabsenkung
Der Anbieter MUSS sich vor Freischaltung der Vertrauensniveauabsenkung nach A_22867 versichern, dass die Einwilligung des Nutzers hierzu insbesondere aufgeklärt, vollständig freiwillig, unter Hervorhebung sichererer Verfahren und widerrufbar erfolgt. Der Nutzer muss dem niederschwelligen Authentifizierungsverfahren aktiv zustimmen.**[<=]**

4.3.2.1 Gerätenutzung

Die unterschiedliche Ausstattung der mobilen Geräte erfordert unterschiedliche Anforderung hinsichtlich der Authentifizierungsverfahren. Unterschieden werden:

- Geräte ohne Hardware Keystore
- Geräte mit Hardware Keystore
- Geräte mit zertifiziertem Secure Element.

Das Vorhandensein eines Hardware Keystore wird hierbei wie folgt definiert:

- Apple - Entscheidend ist das Vorhandensein eines "Secure Enclave" [support.apple.com/guide/security]. Diese ist Bestandteil der A7 und neueren Chips von Apple. Die A7 Serie wurde erstmals 2013 mit dem iPhone 5s eingeführt.
- Android - Ab Android 9 gibt es den Systemaufruf [KeyInfo#getSecurityLevel()], um die Speicherung eines Schlüssels im Hardware Keystore abzufragen. Die Rückgabewerte `KeyProperties.SECURITY_LEVEL_TRUSTED_ENVIRONMENT` oder `KeyProperties.SECURITY_LEVEL_STRONGBOX` sind zulässig. Ältere Systeme bieten die Schnittstelle [KeyInfo#isInsideSecureHardware()] an. Hier ist der Rückgabewert `true` zulässig.

A_22750 - Gerätebindung und Authentisierung

Abhängig von der Geräteausstattung des Nutzers ist eine Gerätebindung für einen festgelegten Zeitraum ohne Erneuerung gültig. Der Anbieter des sektoralen IDP MUSS, wenn er eine Gerätebindung im Rahmen eines Authentisierungsverfahren nutzt, die Zeitrahmen der Gültigkeit für die Gerätebindung gemäß Tabelle "*Übersicht Gerätebindung*" berücksichtigen. Die

Gerätebindung MUSS durch den Nutzer dementsprechend erneuert werden. Das Vertrauensniveau einer Gerätebindung als Authentisierungsfaktor entspricht dem für die Einrichtung verwendeten Identifikationsverfahren.

Tabelle 5: Übersicht Gerätebindung

Gerätenutzung	Authentisierung
ohne Hardware Keystore	• Gerätebindung kann durch Identifikation, welche dem Niveau "hoch" entspricht, angelegt werden. • Gerätebindung kann mit einer 2FA, welche dem Niveau "hoch" entspricht, angelegt werden. • Geräte kann für 24 Stunden mit Biometrie oder PIN/Passwort zur Authentisierung genutzt werden.
mit Hardware Keystore	• Gerätebindung kann durch Identifikation, welche dem Niveau "hoch" entspricht, angelegt werden. • Gerätebindung kann mit einer 2FA, welche dem Niveau "hoch" entspricht, angelegt werden. • Geräte kann für 6 Monate mit Biometrie oder PIN/Passwort zur Authentisierung genutzt werden.
mit zertifiziertem Secure Element	• Gerätebindung kann durch Identifikation, welche dem Niveau "hoch" entspricht, angelegt werden. • Gerätebindung kann mit einer 2FA, welche dem Niveau "hoch" entspricht, angelegt werden. • Gerät kann unbegrenzt mit Biometrie oder PIN/Passwort zur Authentisierung genutzt werden.

[<=]

Hinweis: Die Überprüfung der Gültigkeit der Gerätebindung kann im Authenticator-Modul selbst erfolgen oder alternativ beim sektoralen IDP durch Übermittlung der notwendigen Informationen vom Authenticator-Modul an den sektoralen IDP.

Hinweis: In der Praxis erlaubt dieses Vorgehen primär die Nutzung von nicht zertifizierten Hardware Keystores für eine Authentisierung auf dem formalen Sicherheitsniveau "gematik-ehealth-loa-high".

4.3.2.2 Anforderungen an die Authentisierung der Nutzer

Offener Punkt: Single-Sign-On (SSO)
".... Nach Einschätzung des BSI erheben und verarbeiten sowohl DiGAs, als auch DiPAs Gesundheitsdaten. Anwendungen, die Gesundheitsdaten erheben und verarbeiten, werden dem Vertrauensniveau „hoch" zugeordnet. Somit ist SSO für DiGAs und DiPAs ausgeschlossen. ...".
SSO ist ein wichtiger Bestandteil der Nutzerfreundlichkeit und damit der Akzeptanz. AFOs zu SSO werden deshalb erstellt, sind aber aufgrund der fehlenden Randbedingungen noch nicht final.
Afo - Single-Sign-On (SSO) für TI-Anwendungen innerhalb einer App
Der Hersteller eines sektoralen IDP MUSS ein Single-Sign-On (SSO) für TI-Anwendungen unterstützen.
Afo - Gültigkeitsdauer von SSO
Der Hersteller eines sektoralen IDP MUSS sicherstellen, dass die Gültigkeitsdauer eines SSO <Wert> nicht überschreitet.

A_22744 - Authenticator auf Zweitgerät
Der Hersteller eines sektoralen IDP MUSS ein technisches Verfahren für den Fall etablieren, dass ein Nutzer das Authenticator-Modul auf einem anderen Gerät betreibt als die Fachanwendung, welche eine Authentifizierung beim sektoralen IDP angefragt hat. In diesem Fall MUSS der sektorale IDP für den Auth-Endpunkt ein WebFrontend bieten, welches die weitere Authentisierung über einen Authenticator auf einem anderen Gerät ermöglicht. **[<=]**

Hinweis: Für das Veranlassen zum Öffnen des Authenticator-Moduls durch den Nutzer gibt es unterschiedliche technische Möglichkeiten (z. B. scannen QR-Code von Web-Seite und Codeeingabe im Authenticator, 1-Faktor Login auf Web-Seite und push an Authenticator, u. a.). Diesbezüglich werden keine Anforderungen formuliert, es können auch mehrere Verfahren angeboten werden.

A_22306 - Information des Nutzers bei fehlender Installation des gewählten Authenticator-Moduls
Der Anbieter des sektoralen Identity Provider MUSS auf der unter `redirect_uri` des Authenticator-Moduls erreichbaren Webseite darstellen, aus welcher Quelle das jeweilige Authenticator-Modul des sektoralen Identity Provider zu beziehen ist, auf welchen Geräten/Plattformen er installiert werden kann und welche Voraussetzungen für die Verwendung zur Authentifizierung zu erfüllen sind (z. B. erforderliche Registrierungsprozeduren beim Anbieter des sektoralen Identity Provider).**[<=]**

A_22345 - Maximale Gültigkeit einer Authentifikation
Der sektorale Identity Provider MUSS sicherstellen, dass nach erfolgreicher Authentifikation des Nutzers die Session maximal 12 Stunden ohne erneute Authentifikation gültig bleibt.**[<=]**

A_22257 - Operationsaufruf erfordert erfolgreiche Authentifizierung

Der sektorale Identity Provider MUSS sicherstellen, dass Authorization Request nur nach vorheriger erfolgreicher Authentifikation des Nutzers mit einem `AUTHORIZATION_CODE` beantwortet werden.**[<=]**

A_22235 - Information des Versicherten über Änderungen an Authentifizierungsfaktoren
Der Anbieter des sektoralen Identity Provider MUSS den Versicherten über Änderungen an Authentifizierungsfaktoren informieren.
Die Information des Versicherten kann dabei auch über die Attributbestätigende Stelle erfolgen, welche den Anbieter des sektoralen Identity Provider mit der Erstellung des elektronischen Identifizierungsmittels beauftragt hat.**[<=]**

Hinweis: Dies könnten z. B. Änderungen von E-Mail-Adressen, Mobilfunknummern, registrierten Geräten oder Kennwörtern sein. Die Informationen sollen über entsprechende Anzeige im Authenticator-Modul erfolgen.

A_22236 - Auskunft an Versicherten
Der Anbieter des sektoralen Identity Provider MUSS dem Versicherten auf dessen Verlangen Auskunft geben über

- erfolgte Zugriffe auf das elektronische Identifizierungsmittel des Versicherten und
- Änderungen der Authentifizierungsfaktoren des Versicherten.

[<=]

Hinweis: Die Auskunft könnte z. B. über eine Protokollfunktion im Authenticator-Modul erfolgen. Die Auskunft des Versicherten kann auch über die Attributbestätigende Stelle erfolgen, der den Anbieter des sektoralen IDP mit der Erstellung des elektronischen Identifizierungsmittels beauftragt hat.

5 Anforderungen an Authenticator-Module sektoraler IDPs

5.1 Schnittstellen des Authenticator-Moduls

Schnittstellen des Authenticator-Moduls sind diejenigen, an welchen es Anfragen durch das Anwendungsfrontend oder Web-Frontend empfängt und jene, welche das Authenticator-Modul selbst verwendet, um mit dem Authorization-Endpunkt des sektoralen IDPs in Kontakt zu treten.

Das Authenticator-Modul nimmt die Authentifizierungs-Anfrage des Anwendungsfrontend entgegen und nutzt den Authorization-Endpunkt des sektoralen IDPs, um die Anfrage einzureichen. Der Authorization-Endpunkt des sektoralen IDPs antwortet – nach positiver Validierung der Anfrage und Authentisierung des Nutzers am Authenticator Modul – mit einem `AUTHORIZATION_CODE`. Das Authenticator-Modul empfängt den `AUTHORIZATION_CODE` und leitet diesen an das Anwendungsfrontend weiter. Nachfolgende Abbildung skizziert die Schnittstellen des Authenticator-Moduls und weiterer relevanter Komponenten. Siehe auch 7.1.2- Flow-Diagramm App-App-Flow Abbildung *App-App-Flow*.

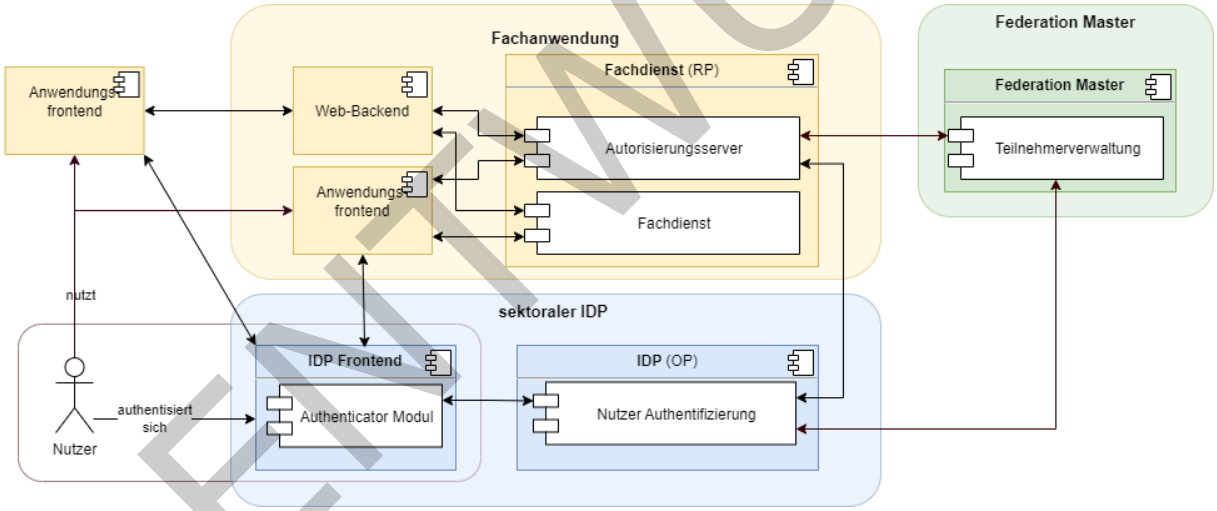

Abbildung 5: Systemkontext Authenticator-Modul

5.2 Funktionsmerkmale Authenticator-Modul

Die folgende Beschreibung in diesem Kapitel gilt für Authenticator-Module sektoraler Identity Provider im Rahmen der Föderation. Entsprechende Vorgaben für die Authenticator-Modul des IDP-Dienstes finden sich in [gemSpec_IDP_Dienst].

Das Authenticator-Modul ist ein Modul, welches in einer Applikation für mobile Endgeräte wie Smartphones bereitgestellt wird. Bei Nutzung eines Primärsystems wird die Funktionalität des Authenticator-Moduls vom Primärsystem selbst realisiert.

Die Bereitstellung des Authenticator-Moduls erfolgt über die dem jeweiligen Betriebssystem üblicherweise zur Verfügung stehenden Portale in einer sicheren, für den Nutzer kostenfreien Form.

Aufgabe des Authenticator-Moduls ist die Nutzerauthentifizierung gegenüber dem sektoralen IDP, bei welchem der Nutzer als Identität hinterlegt ist. Eine weitere Aufgabe ist das Einholen der Zustimmung des Nutzers (Resource Owner) für den Zugriff durch Fachdienste auf Ressourcen des Nutzers (Consent -Freigabe).

Es können je sektoralem IDP ein oder mehrere Authenticator-Module existieren, welche die Authentisierung des Benutzers durchführen. Über die generellen Vorgaben zum Authentifizierungsverfahren hinaus werden hier keine funktionalen Vorgaben gemacht.

Der Anbieter des sektoralen IDP ist für seine Authenticator-Module zuständig. Eine organisatorische Zuständigkeitstrennung zwischen Authenticator-Modulen und Anbietern sektoraler IDPs ist möglich. Ansprechpartner und verantwortlich bleibt in jedem Fall der Anbieter des sektoralen IDP - auch für Produkte von anderen Herstellern.

Aufgabe des Authenticator-Moduls ist, den zum Abruf der `ID_TOKEN` und `ACCESS_TOKEN` benötigten `AUTHORIZATION_CODE`, mit Zustimmung des Nutzers (Resource Owner) und nach eingehender Überprüfung dessen Identität, zu beantragen. Dazu nimmt das Authenticator-Modul die Authentifizierungs-Anfrage des Anwendungsfrontend entgegen und reicht diese am Authorization-Endpunkt des sektoralen IDPs ein. Der Authorization-Endpunkt des sektoralen IDPs antwortet – nach positiver Validierung der Anfrage – mit einem `AUTHORIZATION_CODE`. Das Authenticator-Modul nimmt den `AUTHORIZATION_CODE` und leitet diesen an den Autorisierungsserver bzw. an das Anwendungsfrontend weiter. Durch Übergabe des `AUTHORIZATION_CODE` erhält der Autorisierungsserver bzw. Anwendungsfrontend am Token-Endpunkt das `ID_TOKEN` und `ACCESS_TOKEN` (siehe auch 7.1.2- Flow-Diagramm App-App-Flow).

Schnittstellen des Authenticator-Moduls sind diejenigen, an welchen es Anfragen durch das Anwendungsfrontend oder Web-Frontend empfängt und jene, welche das Authenticator-Modul selbst verwendet, um mit dem Authorization-Endpunkt des sektoralen IDPs in Kontakt zu treten.

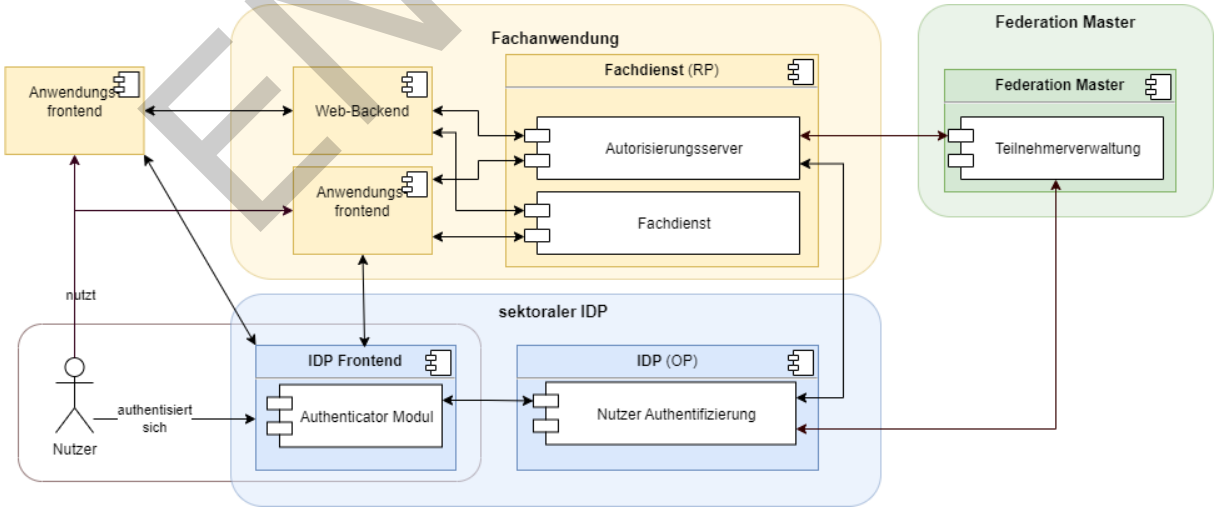

Abbildung 6 Systemkontext Authenticator-Modul

A_22832 - Authenticator-Modul: Anzeige des "user_consent"

Authenticator-Module des sektoralen IDP MUSS die Willenserklärung des Nutzers zur Übermittlung seiner in den `claims` angeforderten Daten zum anfragenden Fachdienst über ein für den Betreiber des sektoralen IDP nicht einsehbares Verfahren einholen. **[<=]**

Hinweis: Die erfolgte Zustimmung des Nutzers darf gespeichert werden und weitere Abfragen können entfallen.

A_22939 - Widerspruch zur Weitergabe einzelner Daten
Authenticator-Module des sektoralen IDP MÜSSEN dem Nutzer die Möglichkeit geben einem Dienst einzelne `claims` nicht zu übermitteln. Auch auf das Risiko hin das dieser Dienst dann nicht verwendet werden kann.**[<=]**

A_23051 - Authenticator-Module für Android und iOS
Der Anbieter des sektoralen Identity Provider MUSS den Nutzern Authenticator-Module für Android und iOS bereitstellen.**[<=]**

A_22277 - Authenticator-Modul: Schutz vor überalterter Software
Der Anbieter des sektoralen Identity Provider MUSS dafür Sorge tragen, dass die von ihm in App Stores veröffentlichten Authenticator-Modulen bei Änderungen automatisiert aktualisiert werden.**[<=]**

A_22659 - Realisierung der App2App-Kommunikation im Fall Android
Im Kontext von Android-Anwendungen MÜSSEN Authenticator-Module zu sektoralen IDP für die wechselseitige Verlinkung den unter [ANDROIDAPPLINKS] beschriebenen App-Link-Mechanismus verwenden und damit Aufrufe an die Adresse des Authorization-EP ermöglichen.**[<=]**

A_22660 - Realisierung der App2App-Kommunikation im Fall Apple/iOS
Im Kontext von iOS-Anwendungen MÜSSEN Authenticator-Module zu sektoralen IDP für die wechselseitige Verlinkung den unter [APPLEUNIVERSAL] beschriebenen Universal-Link-Mechanismus verwenden und damit Aufrufe an die Adresse des Authorization-EP ermöglichen.**[<=]**

A_22661 - Serverseitige Registrierungsdaten
Anbieter von sektoralen IDP MÜSSEN sicherstellen, dass die durch das Betriebssystem notwendigen Voraussetzungen für die Funktionsfähigkeit ihres Authenticator-Moduls erfüllt sind (z. B. Registrierung der Anwendung zur App2App-Kommunikation entsprechend der Mechanismen unter [ANDROIDAPPLINKS] bzw. [APPLEUNIVERSAL] zur Verknüpfung der Anwendung mit einer Webseite).**[<=]**

A_22308-01 - Beschränkung des Authenticator-Moduls eines sektoralen IDP auf die Authentifizierung
Das Authenticator-Modul beim Aufruf durch das Anwendungsfrontend DARF NICHT weitere/andere Funktionalitäten anbieten als solche, die direkt oder indirekt zur Authentifizierung des Nutzers dienen (z. B. Einrichtung, Registrierung, dafür relevante Informationen). Insbesondere Werbung für andere Leistungen oder Funktionen DARF NICHT angezeigt werden.**[<=]**

A_22311 - Verwendung der ursprünglichen Adresse zur Übergabe des "AUTHORIZATION_CODE"
Authenticator-Module von sektoralen Identity Provider MÜSSEN die bei der Übergabe des Authorization Request erhaltene `redirect_uri` für die Übergabe des `AUTHORIZATION_CODE` verwenden. Außer für diesen Aufruf DARF er NICHT an andere Anwendungen übergeben werden.**[<=]**

A_22978 - Aufbereiten von Geräteinformationen

Authenticator-Module von sektoralen IDP SOLLEN Informationen zum verwendeten Endgerät des Nutzers erheben können welche die Inhalte des Datentyps "Device_Type" abbilden.**[<=]**

Der Datentyp "Device_Type" wird perspektivisch zur Übertragung von Informationen über einen Gerätetyp vom Authenticator-Modul zum sektoralenIDP verwendet. Der Datensatz wird vom Authenticator-Modul produziert und soll dem sektoralen IDP dazu dienen TI-Weite Vorgaben zur Zulässigkeit von mobilen Endgeräten bei der Authentisierung umzusetzen. Der Datentyp umfasst die Elemente des folgenden Schema:

Tabelle 6 : Schema Datentyp "Device_Type"

Name	Type	Hinweise
`device_type_data_version`	JSON/String, konstant "1.1"	-
`manufacturer`	JSON/String	Name des Herstellers eines Geräts
`product`	JSON/String	Produktname des Geräts gegenüber dem Endkunden
`model`	JSON/String	Name des Modells
`keystore`	JSON/Boolean	Ist ein Hardware Keystore vorhanden?
`os`	JSON/String	Betriebssystem
`os_version`	JSON/String	Version des Betriebssystems
`security_patchlevel`	JSON/String	Format "YYYY-MM-DD"

A_23031 - Authenticator-Modul: OAuth 2.0 Pushed Authorization Request (PAR)
Das Authenticator-Modul MUSS mittels App2App-Kommunikation übertragene Anfragen entsprechend [OAuth 2.0 Pushed Authorization Requests (section-4)] annehmen und gewährleisten, dass der Request TLS-gesichert in die vertrauenswürdige Ausführungsumgebung des sektoralen IDP übermittelt wird.
[<=]

A_23052 - Authenticator-Modul mit Webfrontend
Das Authenticator Modul MUSS einen Mechanismus zur Durchführung von Authentifizierungsvorgänge welche am Webfrontend ausgelöst wurden anbieten.
*Hinweis: Siehe Anmerkungen zur Umsetzung von A_22744.***[<=]**

A_20527 - Authenticator-Modul: Übertragung des "AUTHORIZATION_CODE" an das Anwendungsfrontend
Das Authenticator-Modul MUSS den vom Authorization-Endpunkt empfangenen `AUTHORIZATION_CODE` an das Anwendungsfrontend übertragen.**[<=]**

Hinweis: Der Authorization-Endpunkt liefert den `AUTHORIZATION_CODE` innerhalb einer HTTP-Redirection (HTTP-Status Code 302) an das Authenticator-Modul zurück. Bei der Verwendung eines Anwendungsfrontends als App ist der Wert des Attributs `location` der HTTP 302 Response dessen im mobilen Betriebssystem registrierte URI. Beim Aufruf der URI wird automatisch das Anwendungsfrontend mit der Verarbeitung der URI gestartet.

6 Anhang A – Verzeichnisse

6.1 Abkürzungen

Kürzel	Erläuterung
AVS	Apothekenverwaltungssystem (ein Primärsystem)
IDP	Identity Provider
JWT	JSON Web Token
KVS	Krankenhausverwaltungssystem (ein Primärsystem)
OAuth 2	Open Authorization 2.0
OIDC	OpenID Connect
PAR	Pushed Authorization Request
PVS	Praxisverwaltungssystem (ein Primärsystem)
SGB	Sozialgesetzbuch
TI	Telematikinfrastruktur
VAU	Vertrauenswürdige Ausführungsumgebung

6.2 Glossar

Begriff	Erläuterung
ACCESS_TOKEN	Ein ACCESS_TOKEN (nach [The OAuth 2.0 Authorization Framework (section-1.4)]) wird vom Client (Anwendungsfrontend) benötigt, um auf geschützte Daten eines Resource Servers zuzugreifen. Die Repräsentation kann als JSON Web Token erfolgen.
Anwendungsfrontend	Die Applikation durch welche ein Nutzer die Dienste einer Anwendung der TI wie etwa das E-Rezept nutzt.

App2App-Kommunikation	Eine direkte Nachrichtenübertragung zwischen zwei Anwendungen auf einem Endgerät, welche durch Mechanismen des Betriebssystems ermöglicht wird.
Authenticator-Modul	Komponente, durch welche der Nutzer die Authentifizierung gegenüber dem IDP vornimmt.
Authentifizierung des Nutzers am Gerät oder lokale Authentifizierung	Authentifizierungsmittel des Nutzers zur Nutzung eines Kontos auf einem Mobilgerät.
Authorization-Endpunkt	Der Authorization-Endpunkt führt nach der initialen Anfrage die Authentifizierung des Nutzers durch und stellt einen `AUTHORIZATION_CODE` aus, welcher zum Abrufen der eigentlichen Token verwendet wird.
Authorization Request	Der Client fordert die Autorisierung vom Ressourceneigentümer durch einen Authorization Request an. Der Authorization Request kann direkt an den Ressourcenbesitzer oder indirekt über die Autorisierung Server als Vermittler gestellt werden (siehe[The OAuth 2.0 Authorization Framework (section-4.1.1)]).
Authorization-Server	OAuth2-Rolle (siehe [The OAuth 2.0 Authorization Framework (section-1.1)]): Der Authorization-Server ist Teil des sektoralen IDP. Der Server authentifiziert den Resource Owner (Nutzer) und stellt Access Token für den vom Resource Owner erlaubten Anwendungsbereich (`scope`) für einen Resource Server bzw. eine auf einem Resource Server existierende Protected Resource aus.
Autorisierte Anwendung eines Schlüssels	Anwendung eines kryptographischen Schlüssels auf Daten durch einen berechtigten Nutzer.
Betriebssystem (oder Plattform)	Der Name des Betriebssystems eines Geräts.
Besitz (eines Geräts)	Verwendungshoheit eines Nutzers über ein Mobilgerät.
`claim`	Ein Key/Value-Paar im Payload eines JSON Web Token.
Client	OAuth2-Rolle (siehe [The OAuth 2.0 Authorization Framework (section-1.1)]): Eine Anwendung (Relying Party), die auf geschützte Ressourcen des Resource Owner zugreifen möchte, die vom Resource Server bereitgestellt werden. Der Client kann auf einem Server (Webanwendung), Desktop-PC, mobilen Gerät etc. ausgeführt werden. Im Fokus der aktuellen Spezifikationen liegt jedoch allein die Kommunikation mit dem E-Rezept-FdV.

Consent	Zustimmung des Nutzers zur Verarbeitung der angezeigten Daten. Der Consent umfasst die Attribute, welche vom sektoralen IDP bezogen auf die im `claim` des jeweiligen Fachdienstes eingeforderten Attribute zusammenfasst. Es besteht Einigkeit zwischen dem, was gefordert wird, und welche Attribute im Token bestätigt werden.
Entity Statement	Ein Entity Statement wird von einer Entity ausgegeben, die sich auf eine Intermediate Entity und Leaf Entity bezieht. Ein Entity Statement ist immer ein signiertes JWT. *Hinweis: Definition Entity Statement, Entity, Intermediate Entity, Leaf Entity siehe [OpenID Connect Federation 1.0 (section-1.2)]*
Federation Master	Der Federation Master basiert auf den Standards OpenID Connect (OIDC), Open Authorization 2.0 (OAuth 2) und JSON Web Token (JWT). Der Federation Master ist einerseits der Trust Anchor des Vertrauensbereichs der Föderation. Andererseits stellt der Federation Master Schnittstellen bereit, welche Auskunft über die in der Föderation registrierten sektoralen IDP gibt.
Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.
Gerät	Alle Arten von mobilen oder stationären Endgeräten.
`ID_TOKEN`	Ein auf JSON basiertes und nach [JSON Web Token (JWT)] genormtes Identitäts-Token, mit dem ein Client (Anwendungsfrontend) die Identität eines Nutzers überprüfen kann.
JSON Web Token	Ein auf JSON basiertes und nach [JSON Web Token (JWT)] genormtes `ACCESS_TOKEN`. Das JWT ermöglicht den Austausch von verifizierbaren `claims` innerhalb seines Payloads.
Löschung	Unter Löschung eines Schlüssels sollen pauschal alle Operationen verstanden werden, die einer Anwendung einen kryptographischen Schlüssel dauerhaft entziehen.
Name (eines Geräts)	Ein vom Nutzer vergebener Name eines Geräts.
Open Authorization 2.0	Ein Protokoll zur Autorisierung für Web-, Desktop und Mobile Anwendungen. Dabei wird es einem Endbenutzer (Resource Owner) ermöglicht, einer Anwendung (Client) den Zugriff auf Daten oder Dienste (Resources) zu ermöglichen, die von einem Dritten (Resource Server) bereitgestellt werden.
OpenID Connect	OpenID Connect (OIDC) ist eine Authentifizierungsschicht, die auf dem Autorisierungsframework OAuth 2.0 basiert. Es

	ermöglicht Clients, die Identität des Nutzers anhand der Authentifizierung durch einen Autorisierungsserver zu überprüfen (siehe [OpenID Connect Core 1.0]).
Pushed Authorization Request (PAR)	Der Pushed Authorization Request (PAR) ermöglicht es Clients, eine OAuth 2.0-Autorisierungsanforderung direkte an den Autorisierungsserver des sektoralen IDP zu senden. Die übergeben redirect-URI ist Autorisierungsendpunkt und wird im weiteren Flow verwendet. https://datatracker.ietf.org/doc/html/rfc9126
Resource Owner	OAuth2-Rolle (siehe [The OAuth 2.0 Authorization Framework (section-1.1)]): Eine Entität (Nutzer), die einem Dritten den Zugriff auf ihre geschützten Ressourcen gewähren kann. Diese Ressourcen werden durch den Resource Server bereitgestellt. Ist der Resource Owner eine Person, wird dieser als Nutzer bezeichnet.
Resource Server	OAuth2 Rolle (siehe [The OAuth 2.0 Authorization Framework (section-1.1)]): Der Server (Dienst), auf dem die geschützten Ressourcen (Protected Resources) liegen. Er ist in der Lage, auf Basis von Access Tokens darauf Zugriff zu gewähren. Ein solcher Token repräsentiert die delegierte Autorisierung des Resource Owner.
sektoraler Identity Provider (IDP)	Als sektoraler Identity Provider wird ein Dienst bezeichnet, welcher nach vorheriger Authentifizierung Identitätsinformationen für eine bestimmte Gruppe von Nutzern innerhalb der TI des Gesundheitswesens bereitstellt, welche anschließend verwendet werden, um auf verschiedene Fachdienste und deren Fachdaten und -prozesse zuzugreifen.
Token-Endpunkt	Ein Endpunkt des Authorization-Servers, welcher für die Ausstellung von Token (`ID_TOKEN` und `ACCESS_TOKEN`) zuständig ist.
Verarbeitungskontext	Die Gesamtheit aus der für eine Klartextverarbeitung erforderlichen Software, dem für eine Klartextverarbeitung genutzten physikalischen System sowie den für die Integrität einer Klartextverarbeitung erforderlichen organisatorischen und physischen Rahmenbedingungen bildet den Verarbeitungskontext der Vertrauenswürdigen Ausführungsumgebung (VAU).

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

6.3 Abbildungsverzeichnis

6.4 Tabellenverzeichnis

6.5 Referenzierte Dokumente

6.5.1 Dokumente der gematik

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

[Quelle]	Herausgeber: Titel

[gemGlossar]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Glossar der Telematikinfrastruktur
[gemSpec_Krypt]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Übergreifende Spezifikation Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur
[gemSpec_PKI]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Übergreifende Spezifikation PKI
[gemSpec_IDP_FedMaster]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Spezifikation Federation Master
[gemSpec_IDP_Dienst]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Spezifikation Identity Provider-Dienst
[gemSpec_IDP_FD]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Spezifikation Identity Provider – Nutzungsspezifikation für Fachdienste
[gemSpec_IDP_Frontend]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Spezifikation Identity Provider - Frontend
[gemSpec_OM]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Übergreifende Spezifikation Operations und Maintenance
[gemSpec_SST_LD_BD]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Spezifikation Logdaten- und Betriebsdatenerfassung
[gemKPT_Test]	Elektronische Gesundheitskarte und Telematikinfrastruktur - Testkonzept der TI

6.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[ANDROIDAPPLINKS]	https://developer.android.com/studio/write/app-link-indexing
[APPLEUNIVERSAL]	https://developer.apple.com/ios/universal-links/
Verordnung (EU) Nr. 910/2014 auch eIDAS Verordnung genannt	VERORDNUNG (EU) Nr. 910/2014 DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 23. Juli 2014 über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt und zur Aufhebung der Richtlinie 1999/93/EG

Duchführungsverordnung (EU) 2015/1502	DURCHFÜHRUNGSVERORDNUNG (EU) 2015/1502 DER KOMMISSION vom 8. September 2015 zur Festlegung von Mindestanforderungen an technische Spezifikationen und Verfahren für Sicherheitsniveaus elektronischer Identifizierungsmittel gemäß Artikel 8 Absatz 3 der Verordnung (EU) Nr. 910/2014 des Europäischen Parlaments und des Rates über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt
GKV-SV Richtlinie "Kontakt mit Versicherten"	Richtlinie des GKV-Spitzenverbandes zu Maßnahmen zum Schutz von Sozialdaten der Versicherten vor unbefugter Kenntnisnahme nach § 217f Absatz 4b SGB V (GKV-SV Richtlinie Kontakt mit Versicherten) vom 14.12.2018
[Uniform Resource Identifier (URI)]	https://datatracker.ietf.org/doc/html/rfc3986
[The OAuth 2.0 Authorization Framework]	https://datatracker.ietf.org/doc/html/rfc6749
[Hypertext Transfer Protocol (HTTP/1.1)]	https://datatracker.ietf.org/doc/html/rfc7231
[JSON Web Token (JWT)]	https://datatracker.ietf.org/doc/html/rfc7519
[Proof Key for Code Exchange by OAuth Public Clients]	https://datatracker.ietf.org/doc/html/rfc7636
[OAuth 2.0 for Native Apps]	https://datatracker.ietf.org/doc/html/rfc8252
[OAuth 2.0 Pushed Authorization Requests]	https://www.rfc-editor.org/rfc/rfc9126.html
[OpenID Connect Core 1.0]	https://openid.net/specs/openid-connect-core-1_0.html
[OpenID Connect Federation 1.0]	https://openid.net/specs/openid-connect-federation-1_0.html
[OAuth 2.0 Pushed Authorization Request]	https://datatracker.ietf.org/doc/html/rfc9126
[ISO18045]	https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

[ISO29115]	ISO/IEC 29115:2013 Information technology — Security techniques — Entity authentication assurance framework
[TR-03107-1]	https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03107/TR-03107-1.pdf;jsessionid=FFBC05B6EE23EE8461127AC755D621FC.internet461?__blob=publicationFile&v=1
[KeyInfo#getSecurityLevel()]	https://developer.android.com/reference/android/security/keystore/KeyInfo#getSecurityLevel()
[KeyInfo#isInsideSecureHardware()]	https://developer.android.com/reference/android/security/keystore/KeyInfo#isInsideSecureHardware()
[support.apple.com/guide/security]	https://support.apple.com/de-de/guide/security/sec59b0b31ff/web
[OpenID Connect Native SSO for Mobile Apps 1.0]	https://openid.net/specs/openid-connect-native-sso-1_0.html

1728
1729

7 Anhang B - Abläufe

7.1 App-App-Flow

Der App-App-Flow beschreibt die Einzelschritte für die Authentifizierung eines Nutzers im Rahmen einer Fachanwendung wobei die Fachanwendung ein App ist, welche auf demselben Gerät wie die Authenticator-App installiert ist.

7.1.1 Vorbedingungen App-App-Flow

- Registrierung des App-Link/Universal-Link für das Frontend auf dem Gerät des Nutzers (auf redirect Adresse des Fachdienstes) - oder einreichen über Web.
- Registrierung des App-Link/Universal-Link für das Authenticator-Modul des IDP auf dem Gerät des Nutzers (auf Adresse des IDP) - oder anfragen über Web.
- Aktueller Signaturschlüssel des Federation Master ist bekannt und vertrauenswürdig bei IDP und Fachdienst eingebracht worden.

7.1.2 Flow-Diagramm App-App-Flow

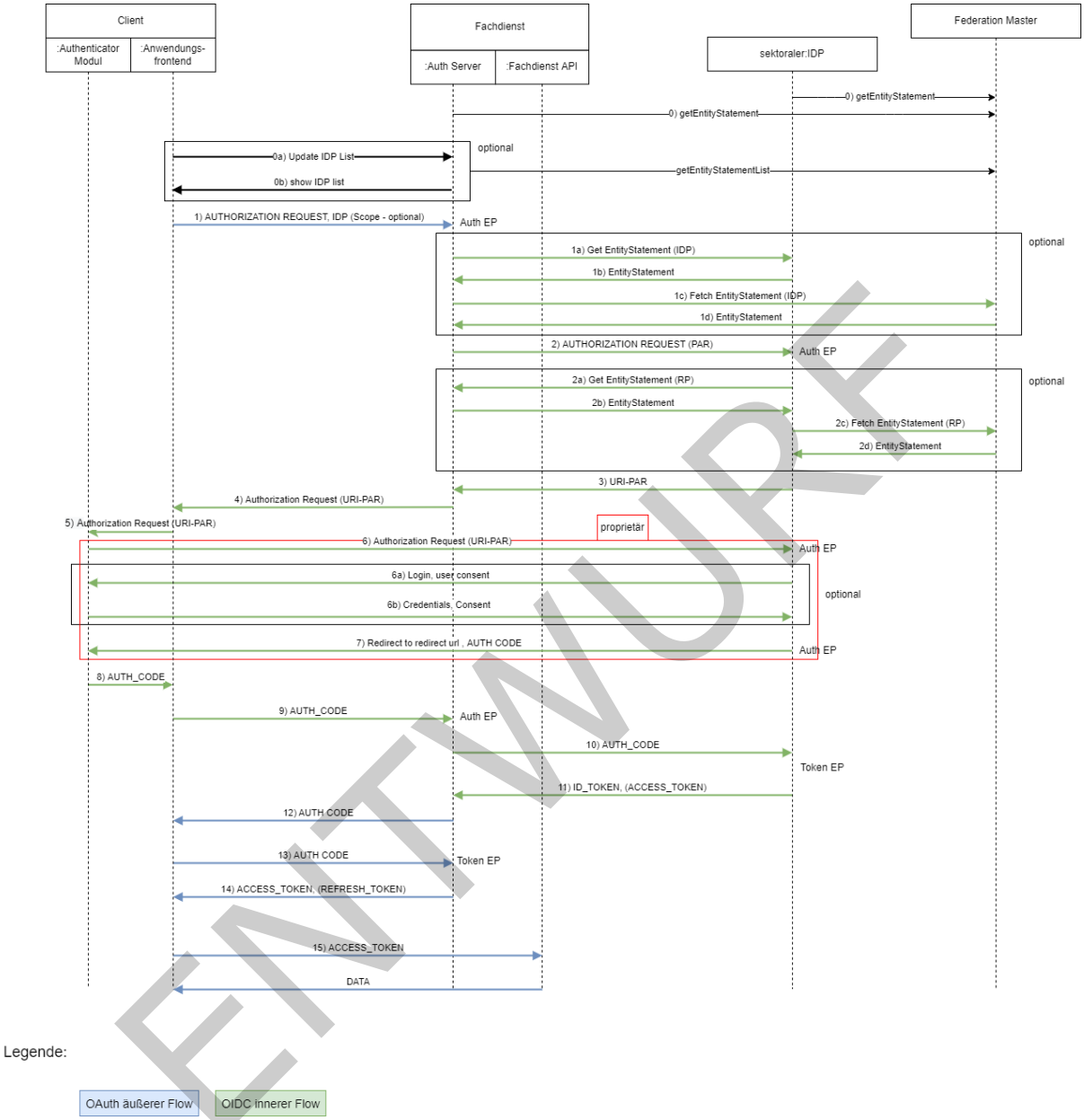

Abbildung 7 : App-App-Flow

7.1.3 Ablaufbeschreibung App-App-Flow

Tabelle 7 : Ablaufbeschreibung App-App-Flow

Schritt	Teilschritt	Beschreibung	Standard

0		Bezug des Entity Statement des Federation Master unter Nutzung des bekannten Signaturschlüssels. (Folgeschlüssel können über das signierte Entity Statement transportiert werden) Hintergrund: *Das Entity Statement wird von einer Entität eines IDP (im föd. Verbund) ausgestellt und betrifft eine Subjekt-Entität und Blatt-Entitäten in einer Föderation. Ein Entity Statement ist immer ein signiertes JSON Web Token (JWT).*	• Entity Statement → OpenID Connect Federation 1.0 (section-3.1) • Key Rollover for a Trust Anchor → OpenID Connect Federation 1.0 (section-9.2)
	0-a	Bei Bedarf ruft das Anwendungsfrontend beim Autorisierungsserver die Liste aller IDPs ab. Die Ermittlung der registrierten IDPs erfolgt über den Federation Master. Beim Federation Master sind die Entity Statements aller registrierten IDP hinterlegt. Die Bereitstellung der Liste kann über zwei Wege erfolgen: a) Der Fachdienst verwendet das OIDC Federation API . Der Fachdienst muss dann aus dem Response die für eine Auswahl notwendigen Informationen extrahieren und seinen Anwendungsfrontends zur Verfügung stellen. b) Der Federation Master stellt ein zusätzliches API neben dem Standard-API bereit und liefert hier nur die für eine Auswahl notwendigen Informationen (Name der Organisation/Kasse, Icon, weitere Informationen für Folge-Request zur Ermittlung des vollständigen Entity Statement). Die Adresse des API ist als custom-metadata im Entity	• Entity Listings Request → OpenID Connect Federation 1.0 (section-7.3.1) • OP-Metadata organisation_name → OpenID Connect Federation 1.0 (section-4.2) • Metadata Erweiterung → OpenID Connect Federation 1.0 (section-4)

		Statement des Federation Master hinterlegt.	
	0-b	• Der Autorisierungsserver antwortet dem Anwendungsfrontend mit der Liste aller IDPs. • Das Anwendungsfrontend zeigt dem Nutzer eine Suchfunktion an, in der er in der Liste seine Kasse per Name und mit Icon auswählen kann. • Die Auswahl kann am Anwendungsfrontend gespeichert werden, so dass bei folgenden Anmeldungen der Nutzer diese manuelle Auswahl nicht mehr durchführen muss.	

1		Das Anwendungsfrontend sendet dem Autorisierungsserver einen AUTHORIZATION_REQUEST und eine Code-Challenge sowie den zur Anmeldung gewünschten IDP. (Wenn die Wahl des IDP nicht im Anwendungsfrontend getroffen wurde (0-a) kann der Autorisierungsserver in diesem Schritt einen Auswahldialog anzeigen lassen.)	• Authorization Request → The OAuth 2.0 Authorization Framework (section-4.1.1) • PKCE/Code-Challenge → Proof Key for Code Exchange by OAuth Public Clients (section-4.3)
	1-a	Falls der Autorisierungsserver das Entity Statement des IDP noch nicht kennt, lädt er dies herunter. (/.well-known/openid-federation)	Federation Entity Configuration Request → OpenID Connect Federation 1.0 (section-6.1)
	1-b	Der IDP sendet sein Entity Statement zurück.	• Federation Entity Configuration Response → OpenID Connect Federation 1.0 (section-6.2) • OAuth 2.0 Pushed Authorization Request → OAuth 2.0 Pushed Authorization Requests (section-5)
	1-c	Der Autorisierungsserver fragt das Entity Statement des Federation Master über den IDP an.	Entity Statement-Request → OpenID Connect Federation 1.0 (section-7.1.1)
	1-d	Der Federation Master sendet sein Entity Statement über den IDP zurück.	• Federation Entity Configuration Response → OpenID Connect Federation 1.0 (section-6.2) • Validation trust chain → OpenID Connect Federation 1.0 (section-8)

2		Der Autorisierungsserver sendet einen Pushed Authorization Request (PAR) inkl. Code-Challenge / PKCE, benötigter `claims` bzw. `scope` und eines `private_key_jwt` an den IDP.	• OAuth 2.0 Pushed Authorization Requests → OAuth 2.0 Pushed Authorization Requests (section-2.1) • Authentication Request → OpenID Connect Federation 1.0 (section-3.1.2.1) • PKCE/Code-Challenge → Proof Key for Code Exchange by OAuth Public Clients (section-4.3) • Client Authentication → OpenID Connect Federation 1.0 (section-9)
	2-a	Falls der IDP das Entity Statement des Autorisierungsservers noch nicht kennt, lädt er dies herunter. (/.well-known/openid-federation).	Federation Entity Configuration Request → OpenID Connect Federation 1.0 (section-6.1)
	2-b	Der Autorisierungsserver sendet sein Entity Statement zurück und der IDP registriert ihn als Client.	• Federation Entity Configuration Response → OpenID Connect Federation 1.0 (section-6.2) • RP Metadata → OpenID Connect Federation 1.0 (section-4.1) • Entity Statement → OpenID Connect Federation 1.0 (section-3.1) • OAuth 2.0 Pushed Authorization Requests → OAuth 2.0 Pushed Authorization Requests (section-6)
	2-c	Abruf des Entity Statement zum Fachdienst/Autorisierungsserver beim Federation Master.	Entity Statement-Request → OpenID Connect

			Federation 1.0 (section-7.1.1)
	2-d	Der Federation Master sendet sein Entity Statement über den Fachdienst/Autorisierungsserver zurück.	• Federation Entity Configuration Response → OpenID Connect Federation 1.0 (section-6.2) • Automatic Registration → OpenID Connect Federation 1.0 (section-10.1) • Validation trust chain → OpenID Connect Federation 1.0 (section-8.2) • Entity Statement → OpenID Connect Federation 1.0 (section-3.1)
3		Der IDP sendet eine Request-URI (mit Bezug zum vorherigen AUTHORIZATION_REQUEST) an den Autorisierungsserver.	Request-URI → OAuth 2.0 Pushed Authorization Requests (section-2.2)
4		Der Autorisierungsserver sendet die Request-URI und Client ID an das Anwendungsfrontend zur Weiterleitung an die Adresse des Authenticator des IDP.	
5		Anwendungsfrontend öffnet den Authenticator für die eigentliche Authentifizierung des Anwenders (Deep-Link/Universal-Link).	
6		Das Authenticator-Modul leitet den Authentication Request an den IDP weiter (proprietär).	

	6-a	• Der IDP Prüft anhand der URI ob der Request zu einem vorherigen AUTHORIZATION_REQUEST gehört (proprietär). • Der Authorization-Endpunkt des IDP stellt (wenn nötig) entsprechend den angefragten `claims` einen Consent (Zustimmung des Nutzers zur Verarbeitung der angezeigten Daten) zusammen (proprietär). • Der Authorization-Endpunkt des IDP überträgt (wenn nötig) Consent-Abfrage und ggf. für die Authentisierung des Nutzers notwendige Daten zu dem Authenticator-Modul (proprietär).	
	6-b	• Das Authenticator-Modul des IDP fordert den Nutzer (wenn nötig) zur Consent-Zustimmung auf und führt die Authentisierung des Nutzers nach den Verfahren des IDP durch. Das notwendige Vertrauensniveau steht im Request (acr-`claim`). • Das Authenticator-Modul des IDP bestätigt dem IDP die erfolgreiche Durchführung der Authentisierung (proprietär). • Der Authorization-Endpunkt des IDP erstellt den `AUTHORIZATION_CODE`.	
7		Der Authorization-Endpunkt des IDP antwortet dem Authenticator-Modul mit dem `AUTHORIZATION_CODE` und einem Redirect zum Fachdienst (proprietär).	

8		Das Authenticator-Modul des IDP ruft über einen App-Link bzw. Universal-Link entsprechend der Redirect-URL das Anwendungsfrontend auf (eigentlich ein Redirect zum Fachdienst aber das Frontend ist auf die Adresse registriert) und übergibt den `AUTHORIZATION_CODE`.	
9		Die Anwendungsfrontend leitet den `AUTHORIZATION_CODE`(IDP) an den Autorisierungsserver.	
10		Der Autorisierungsserver reicht den `AUTHORIZATION_CODE`(IDP), den `CODE_VERIFIER` und seinen `private_key_jwt` beim Token-Endpunkt des IDP ein.	• `AUTHORIZATION_CODE` und `CODE_VERIFIER` → Proof Key for Code Exchange by OAuth Public Clients (section-4.5) • Client Authentication → OpenID Connect Core 1.0 (section-9)
11		• Der Autorisierungsserver erhält vom Token-Endpunkt des IDP einen `ID_TOKEN` und `ACCESS_TOKEN` mit den gewünschten `claims` , der mit dem öffentlichen Schlüssel aus der Registrierung verschlüsselt ist. • Der Autorisierungsserver entschlüsselt das `ACCESS_TOKEN`. • Der Autorisierungsserver prüft den Herausgeber `iss`, validiert die Signatur des `ID_TOKEN` gegen den zur KID passenden Schlüssel aus den JWKS des IDP und zieht die `claims` (d. h. die Key/Value-Paare im Payload eines Tokens) der authentisierten Identität aus dem `ID_TOKEN`.	

12		Zum weiteren Zugriff erstellt der Autorisierungsserver ein AUTHORIZATION_CODE(AS) und sendet diese an das Anwendungsfrontend.	
13		Anwendungsfrontend übergibt dem Autorisierungsserver den AUTHORIZATION_CODE(AS) sowie den CODE_VERIFIER.	
14		Anwendungsfrontend erhält ACCESS_TOKEN und REFRESH_TOKEN mit den notwendigen Daten vom Autorisierungsserver.	
15		• Das Anwendungsfrontend greift auf die Fachdienst API zu und übergibt dabei das ACCESS_TOKEN. • Nach erfolgreicher Validierung des ACCESS_TOKEN gibt die Fachdienst API den Zugriff auf die Fachdaten dieser Identität frei.	

7.1.4 Detailinformationen zum App-App-Flow

Abruf der Schlüssel des Federation Master

Dazu wird das selbst Entity Statement des Federation Master abgerufen und gegen den vorher bekanntgemachten Signaturschlüssel des Federation Master geprüft.

Response auf GET an die Adresse "http://master0815.de/.well-known/openid-federation"

HTTP 200 mit Content-Type: application/jose

Folgende Werte müssen im Header des selbst signierten Entity Statement des Federation Master auftauchen:

Tabelle 8 : Header Entity Statement des Federation Master

Name	Werte	Beispiel	Anmerkungen
alg	*ES256*	-	
kid	wie aus jwks im Body des Dokumentes	"master0815-1"	Identifier des verwendeten Schlüssels aus dem jwks im Body des Statement

typ	*JWT*	-	

Folgende Werte müssen im Body des selbst signierten Entity Statement des Federation Master enthalten sein:

Tabelle 9 : Body Entity Statement des Federation Master

iss	URL	"http://master0815.de"	iss anstelle issuer ist hier Spec konform = URL des Federation Master (wird definiert)
sub	URL	"http://master0815.de"	URL des Federation Master (wird definiert) = iss
iat	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2	1645398001	2022-02-21 00:00:01
exp	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2	1646002800	Beispielhafte Gültigkeit von 7 Tagen
jwks	JWKS Objekt	unter anderem "master0815-1"	Schlüssel für die Signatur des Entity Statement Gemäß OpenID Connect Federation 1.0 (section-9.2) werden hier auch Schlüssel für einen Key-Rollover transportiert.

metadata {			
federation_entity {			
`federation_api_endpoint`	URL	"http://master0815.de/federation_api_endpoint"	Adresse des Endpunktes zum Abrufen einzelner oder aller Statements des Masters über IDPs und Fachdienste
`idp_list_endpoint`	URL	"http://master0815.de/idp_list.jws"	non Standard `claim` - ggf. auch als reine Konfiguration machbar z.B /.well-known/entity_listing
}}			

IDP-Liste

Es wird vom Nutzer einer Anwendung der Telematikinfrastruktur erwartet, dass dieser die Institution kennt, welche seine Identität herausgibt (bei einem Versicherten wäre dies z. B. seine Krankenkasse).

Tabelle 10 : Beispiel vorliegender Identitätsdaten

Begriff	Erläuterung	Beispiel
Identität	Von Institution gemanagte ID	

Jede Kasse wird als eigener IDP mit eigenen Endpunkten und Entity Statements geführt. Ein Dienstleister kann dahinter aber denselben Dienst stehen haben und die Kassen als Mandanten pflegen. Damit bleibt es auch möglich für die Kasse bei

fehlender Installation auf einer eigenen Infoseite zu ihren Apps zu verweisen. Kassen geben die Freigabe für ihren Eintrag in der Föderation frei.

Die Liste der Kassen wird aus der Föderation generiert und am Federation Master zum Abruf bereitgestellt. Die Integrität der Liste wird mittels Signatur über einen Schlüssel aus dessen Keyset sichergestellt.

(0-a) Anwendungsfrontend fragt die Liste aller IDPs ab

Das Anwendungsfrontend fragt die Liste aller IDPs ab, oder der Autorisierungsserver lässt diese Liste selbst im Frontend anzeigen (Webview). Die Kommunikation zwischen Anwendungsfrontend und Fachdienst ist anwendungsspezifisch und wird hier nicht weiter spezifiziert.

(0-b) Autorisierungsserver antwortet dem Anwendungsfrontend mit der Liste aller IDPs

Der Autorisierungsserver antwortet dem Anwendungsfrontend mit der Liste aller IDPs oder der Autorisierungsserver lässt diese Liste selbst im Frontend anzeigen. Diese Liste wird als JWS formatiert und mittels eines Schlüssels des Federation Master signiert. Das Frontend lässt den Nutzer die Wahl seines IDP (seiner Kasse) treffen oder diese Auswahl erfolgt über eine Webseite des Fachdienstes. Die notwendigen Informationen können aus den Entity Statements gelesen werden. Das signierte JWS der IDP-Liste hat folgende Inhalte:

Tabelle 11 : Attribute der IDP-Liste

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	"http://master0815.de"	`iss` anstelle issuer ist hier Spec konform = URL des Federation Master (wird definiert)
`iat`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2	1645398001	2022-02-21 00:00:01
`exp`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2	1646002800	Beispielhafte Gültigkeit von 7 Tagen
idp_entity {			
`organization_name`	String (max. 128 Zeichen	"IDP 4711"	Der Name des IDP zur Anzeige für den Benutzer ist die Definition

			von `organization_name` im Entity Statement des IDP
`iss`	URI	"https://idp4711.de"	issuer Wert des jeweiligen sektoralen Identitiy Provider (URL) - sollte nach Vorgaben der Föderation der Adresse für die Authentisierung entsprechen
`logo_uri`	URI	„https://idp4711.de/logo.png“	Parameter `logo_uri` aus dem Entity Statement des IDP
`user_type_supported`	[HCI = Health Care Institution, HP = Health Professional, IP = Insured Person]	"IP"	Parameter `user_type_supported` aus dem Entity Statement des IDP
}			

Folgende Werte müssen im Header der vom Federation Master signierten IDP-Liste auftauchen:

Name	Werte	Beispiel	Anmerkungen
`alg`	*ES256*	-	
`kid`	wie aus jwks im Body des Entity Statement	"master0815-1"	Identifier des verwendeten Schlüssels aus dem jwks im Body des Statement
`typ`	*JWT*	-	

(1) Authorization Request von Anwendungsfrontend zum Authentication-Endpunkt (Auth EP) des Autorisierungsservers des Fachdienstes

Das Anwendungsfrontend sendet ein HTTP-GET an den Authorisation Server des Fachdienstes. Die folgenden GET-Parameter werden im query string verwendet:

1800 **Tabelle 12 Authorization Request von Anwendungsfrontend zum Autorisierungsservers**

Name	Werte	Beispiel	Anmerkungen
client_id	VSCHAR (max. 32 Zeichen)	"eRezeptApp"	kein ";" und kein "┼" (definiert gem. Unicode U+253C (95 32)), kein Leerzeichen
state	VSCHAR (max. 32 Zeichen)	af0ifjsldkj	
redirect_uri	URL	"https://Fachdienst007.de"	Adresse des Fachdienstes weil da soll der ACCESS_TOKEN am Ende landen.
code_challenge	Hash über CODE_VERIFIER	K2-ltc83acc4h0c9w6ESC_rEMTJ3bww-uCHaoeK1t8U	
code_challenge_method	*S256*	-	
response_type	*code*	-	
scope	String	"e-rezept"	Anwendungsspezifisch zu definieren, kein *openid*
claims			weitere claims
idp_iss	URL	"https://idp4711.de"	• nicht Standard Parameter, • iss URL des IDP den der Nutzer für die Authentisierung ausgewählt hat, • optional - nötig, wenn Auswahl des IDP im Frontend passiert.

(1-a) Falls der Autorisierungsserver des Fachdienstes das Entity Statement des IDP noch nicht kennt, lädt er dies herunter

Request:

HTTP-GET

Adresse: "https://idp4711.de/.well-known/openid-federation"

(1- b) Der IDP sendet sein Entity Statement zurück

Der Autorisierungsserver verifiziert die Signatur des Entity Statement gegen einen Schlüssel aus dem Entity Statement des Federation Master über diesen issuer (OpenID Connect Federation 1.0 (section-8.2)).

Response:

HTTP 200 mit Content-Type: application/jose

Folgende Werte müssen im Header des selbst signierten Entity Statement des sektoralen IDP auftauchen:

Tabelle 13 : Header Entity Statement des sektoralen IDP

Name	Werte	Beispiel	Anmerkungen
`alg`	*ES256*	-	
`kid`	wie aus jwks im Body des Dokumentes	"idp4711-3"	Identifier des verwendeten Schlüssels aus dem jwks im Body des Entity Statement
`typ`	*JWT*	-	

Folgende Werte müssen im Body selbst signierten Entity Statement des sektoralen IDP-Dienstes enthalten sein:

Tabelle 14 : Body Entity Statement des sektoralen IDP

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	"https://idp4711.de"	`iss` anstelle issuer ist hier Spec konform = URL des IDP (variabel je Mandant/Kasse)
`sub`	URL	"https://idp4711.de"	URL des IDP (variabel je Mandant/Kasse) = `iss`
`iat`	Alle time Werte in Sekunden seit 1970, RFC	1645484401	2022-02-22 00:00:01

	7519 Sect.2		
exp	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2	1645570800	Gültigkeit von 24 Stunden
jwks	JWKS Objekt	unter anderem "idp4711-3"	Schlüssel für die Signatur des Entity Statement und optional weitere Schlüssel des IDP
authority_hints	[string]	"http://master0815.de"	iss Bezeichnung des Federation Master
metadata {			
openid_provider {			
issuer	URL	"https://idp4711.de"	URL des IDP (variabel je Mandant/Kasse)
signed_jwks_uri	URL	"https://idp4711.de/jws.json"	Optional - Ablageort für weitere Schlüssel des IDP etwa die zur Signatur seiner Token Alternativ liegen alle Schlüssel im jwks des Entity Statement. Wenn eine signed_jwks_uri im Entity Statement angegeben ist müssen auch diese Schlüssel importiert werden. Beides sollte unterstützt werden.
organization_name			Name des IDP - wird genutzt in der Auswahlliste für den Benutzer (Alternativ name im Feld federation_entity nutzen)
logo_uri	URL	„https://idp4711.de/logo.png“	Attribut ist nicht im Standard, ist nach OpenID Connect Discovery 1.0 - aber in Federation Spec auch für ein OP gelistet

authorization_endpoint	URL	„https://idp4711.de/Auth“	Adresse des IDP-Endpunkt (im Internet)
token_endpoint	URL	„https://idp4711.de/Token“	Adresse des IDP-Endpunkt (im Internet)
pushed_authorization_request_endpoint	URL	„https://idp4711.de/PAR_Auth“	Adresse des IDP-Endpunkt (im Internet) nach OAuth 2.0 Pushed Authorization Requests (section-5)
client_registration_types_supported	[*automatic*]	-	
subject_types_supported	[*pairwise*]	-	
response_types_supported	[*code*]		Weitere Werte sind möglich.
scopes_supported	[openid profile email telematik]		Weitere Werte sind möglich - The OAuth 2.0 Authorization Framework (section-3.3)
response_modes_supported	[*query*]		
grant_types_supported	[*authorization_code*]		
require_pushed_authorization_requests	true		OAuth 2.0 Pushed Authorization Requests (section-5)
token_endpoint_auth_methods_supported	[*private_key_jwt*]		Weitere Werte sind möglich.
token_endpoint_auth_signing_alg_values_supported	[*ES256*]	-	Weitere Werte sind möglich.

request_authentication_methods_supported	*{ "ar": ["none"], "par": ["private_key_jwt"] }*	-	
request_object_signing_alg_values_supported	[*ES256*]	-	OpenID Foundation - Issue: https://bitbucket.org/openid/connect/issues/1474/request_authentication_signing_alg_values
id_token_signing_alg_values_supported	[*ES256*]	-	Weitere Werte sind möglich.
id_token_encryption_alg_values_supported	[ECDH-ES]		
id_token_encryption_enc_values_supported			
}			
federation_entity {			
name	String	"IDP 4711"	Name des IDP - wird genutzt in der Auswahlliste für den Benutzer (alternativ organization_name aus Metadata nutzen)
contacts	[string]	"support@idp4711.de"	optional
homepage_uri	URL	"https://idp4711.de"	optional
}}			

signed_jwks

Ablageort für weitere Schlüssel des IDP etwa die zur Signatur seiner Token. Wenn eine signed_jwks_uri im Entity Statement angegeben ist, müssen auch diese Schlüssel importiert werden.

Folgende Werte müssen im Header des selbst signierten KeySet des sektoralen IDP auftauchen:

Tabelle 15 : Header des KeySet des sektoralen IDP

Name	Werte	Beispiel	Anmerkungen
alg	*ES256*		
kid	wie aus jwks im Body des Entity Statement		Identifier des verwendeten Schlüssels aus dem jwks im Body des Entity Statement
typ	*JWT*		

Folgende Werte müssen im Body enthalten sein:

Tabelle 16 : Body des KeySet des sektoralen IDP

Name	Werte	Beispiel	Anmerkungen
keys {			
kty		EC	
kid		idp4711-3	
crv		P-256	
x		qAOdPQROkHfZY1daGofOmSNQWpYK8c9G2m2Rbkpbd4c	
y		G_7fF-T8n2vONKM15Mzj4KR_shvHBxKGjMosF6FdoPY	
use		sig	nach JSON Web Key (section-4.2)
}			
iss	URL	"https://idp4711.de"	URL des IDP (variabel je Mandant/Kasse)
iat	Alle time Werte in Sekunden seit 1970, RF	1645484401	2022-02-22 00:00:01

	C 7519 Sect.2		

(1- c) Der Autorisierungsserver des Fachdienstes ruft das Entity Statement zum IDP beim Federation Master ab

Request:

HTTP-GET

Adresse: "http://master0815.de/federation_api_endpoint"

HTTPS GET Request an den federation_api_endpoint aus dem Entity Statement des Federation Master mit dem folgenden Parameter:

Tabelle 17 : Parameter HTTPS GET Request an den Federation Master API

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	"http://master0815.de"	issuer des Federation Master - verpflichtender Parameter für unser Szenario aber ohne Relevanz
`sub`	URL	"https://idp4711.de"	issuer des angefragten sektoralen IDP

(1-d) Der Federation Master sendet sein Entity Statement über den angefragten sektoralen IDP zurück

Response:

HTTP 200 mit Content-Type: application/jose

Folgende Werte müssen im Header des Entity Statement des Federation Master über den sektoralen IDP-Dienst enthalten sein:

Tabelle 18 : Header HTTP-Response

Name	Werte	Beispiel	Anmerkungen
`alg`	*ES256*		
`kid`	wie aus jwks im Body des Dokumentes	"master0815-1"	Identifier des verwendeten Schlüssels aus dem jwks im Body des Statement
`typ`	*JWT*		

Folgende Werte müssen im Body des Entity Statement des Federation Master über den sektoralen IDP-Dienst enthalten sein:

Tabelle 19 : Body HTTP-Response

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	"http://master0815.de"	URL des Federation Master
`sub`	URL	"https://idp4711.de"	URL des angefragten IDP (variabel je Mandant/Kasse)
`iat`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2	1645398001	2022-02-21 00:00:01
`exp`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2	1645480801	Beispielhafte Gültigkeit von 1 Tag um schneller Sperrungen durchzuführen
`jwks`	JWKS Objekt	unter anderem "idp4711-3"	Schlüssel für die Signatur des Entity Statement des IDP

Als Ergebnis des Schritts (d2-d) kennt der Autorisierungsserver des Fachdienstes kennt die öffentlichen Schlüssel für Signaturen des IDP.

(2) Der Autorisierungsserver des Fachdienstes sendet einen Pushed Authorization Request (PAR) an den Authentication-Endpunkt (Auth EP) des sektoralen IDP

Der innere Flow startet mit dem Pushed Authorization Request (rfc9126) des Fachdienstes an den sektoralen IDP. Als `client_assertion` wird private_key_jwt verwendet (siehe OIDC Standard OpenID Connect Core 1.0 (section-9)).

Anmerkung: Dies passiert als Folge des Authorization Request des Anwendungsfrontends.

HTTP-POST

Der Authorization Request des Fachdienstes zum sektoralen IDP enthält die folgenden Parameter:

Tabelle 20 : Parameter Pushed Authorization Request

Name	Werte	Beispiel	Anmerkungen
`client_id`	URL	"https://Fachdienst007.de"	kein ";" und kein "┼" (definiert gem. Unicode U+253C (9532)), kein Leerzeichen

state	VSCHAR (max 32 Zeichen)	bg1jgktmelk	Generierter Wert, ist ein anderer state als in dem OAUTH Request des Frontendan den Fachdienst
redirect_uri	URL	https://Fachdienst007.de/AS	Adresse des Fachdienstes Authorization-Server
code_challenge	Hash über CODE_VERIFIER des Fachdienstes	K2-mvd94bdd5i1d0x7FTD_sFNRK4cxx-vDIbpfL2u9W	CODE_VERIFIER ist ein beliebiger Wert, über den der Hash gebildet wird.
code_challenge_method	*S256*	-	
response_type	*code*	-	
nonce	(max. 32 Zeichen)	274312:dj83hs9s	Beliebig generierter Wert, hier wird auch die nonce genutzt, die mit dem ID_TOKEN abgeglichen wird.
scope	[string]	"profile telematik openid"	The OAuth 2.0 Authorization Framework (section-3.3)
acr_values	"gematik-ehealth-loa-high" oder "gematik-ehealth-loa-substantial"	"gematik-ehealth-loa-high"	
client_assertion_type	"*urn:ietf:params:oauth:client-assertion-type:jwt-bearer*"	-	Notwendiger Parameter nach OpenID Connect Core 1.0 (section-9)
client_assertion	*private_key_jwt*	siehe unten	

max_age	Number	"0"	Zulässige Zeit in Sekunden seit der letzten Authentisierung des Nutzers. Diesen Wert auf 0 zu setzen erzwingt immer eine erneute Authentisierung des Nutzers, auch dann, wenn zukünftig Single-Sign-On Mechanismen zulässig werden würden.

1865

1866 **private_key_jwt**

1867 Das private_key_jwt ist mittels ES256 signiert und der Header hat folgende Inhalte:

1868 **Tabelle 21 : Header des private_key_jwt**

Name	Werte	Beispiel	Anmerkungen
alg	*ES256*	-	
typ	*JWT*	-	
kid	wie aus jwks (oder signed_jwks) im Body des Entity Statement	"Fachdienst007-42"	Der öffentliche Schlüssel muss auch im Entity Statement des Fachdienstes stehen. (der Einfachheit halber wird im Beispiel derselbe Key für alle Signaturen genutzt)

1869 Das eigentliche Datenobjekt sieht wie folgt aus:

1870 **Tabelle 22 : Inhalt des private_key_jwt**

Name	Wert	Beispiel	Anmerkungen
iss	URL	"https://Fachdienst007.de"	client_id des Fachdienstes

sub	URL	"https://Fachdienst007.de"	client_id des Fachdienstes
aud	URL	"https://idp4711.de/PAR_Auth"	URL des Pushed_Authorization_Endpunkts des sektoralen IDPs
jti	random max. 32 Zeichen	on7W8ltV2F7mDzp10zThzrors8BSBM4b	
exp	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645565032	Vorgesehen ist eine Gültigkeit von jeweils 90 Sekunden
iat	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645564942	2022-02-22 22:22:22

Zu den scopes und claims bzgl. der Identitäten für Versicherte siehe Kapitel 4.2.3.2.

(2-a) Falls der IDP das Entity Statement des Autorisierungsservers des Fachdienstes noch nicht kennt, lädt er dies herunter

Request:

HTTP-GET

Adresse: "https://Fachdienst007.de/.well-known/openid-federation"

(2-b) Der Autorisierungsserver des Fachdienstes sendet sein Entity Statement zurück und der IDP registriert ihn als Client (Automatic Registration)

Der IDP verifiziert die Signatur des Entity Statement gegen einen Schlüssel aus dem Entity Statement des Federation Master über einen Dienst gemäß den Standards:

- OpenID Connect Federation 1.0 (section-10.1)
- OpenID Connect Federation 1.0 (section-8.2)

Response:

HTTP 200 mit Content-Type: application/jose

Folgende claims müssen im Header des selbst signierten Entity Statement des Fachdienstes auftauchen:

Tabelle 23 : Header des Entity Statement des Fachdienstes

Name	Werte	Beispiel	Anmerkungen
`alg`	*ES256*	-	
`kid`	wie aus jwks im Body des Dokumentes	"Fachdienst007-42"	Identifier des verwendeten Schlüssels aus dem jwks im Body des Statement
`typ`	*JWT*	-	

Folgende Body-`claims` müssen im selbst signierten Entity Statement des Fachdienstes enthalten sein:

Tabelle 24 : Body des Entity Statement des Fachdienstes

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	"https://Fachdienst007.de"	iss anstelle issuer ist hier Spec konform = URL des Fachdienstes
`sub`	URL	"https://Fachdienst007.de"	URL des Fachdienstes (variabel je Mandant/Kasse) = iss
`iat`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645484401	2022-02-22 00:00:01
`exp`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645570800	//Gültigkeit von 24 Stunden

`jwks`	JWKS Objekt	unter anderem "Fachdienst007-42" "Fachdienst007-69", wenn nicht im signed_jwks transportiert	Schlüssel für die Signatur des Entity Statement Es können auch weitere Schlüssel des Fachdienstes etwa die zur Signatur seiner `private_key_jwt` Authentisierungs objekte (`use` = "sig") oder für die Verschlüsselung der`ID_TOKEN` (`use` = "enc") hier enthalten sein.
`authority_hints`	[string]	"http://master0815.de "	iss Bezeichnung des Federation Master
metadata {			
openid_relying_party {			
`signed_jwks_uri`	URL	https://Fachdienst007.de/jws.json	Optional: (es können auch alle Schlüssel im Statement stehen) Wenn eine `signed_jwks_uri` im Entity Statement angegeben ist müssen auch diese Schlüssel importiert werden Enthält Schlüssel für die Signatur des Entity Statement, der `private_key_jwt` Authentisierungs objekte (`use` = `sig`) und für die Verschlüsselung der `ID_TOKEN` (`use` = `enc`)

`organization_name`	String	007 GmbH	Optional: Name der Organisation die hinter dem Fachdienst steht
`client_name`	String	Fachdienst007	Name des Fachdienstes (redundant zum `name` in den "Federation Entity"`claims`)
`logo_uri`	URL	https://Fachdienst007.de/logo.jpg	Optional: Wenn vorhanden zur Darstellung der Anfrage durch den Authenticator/IDP zu verwendet
`redirect_uris`	[URLs]	https://Fachdienst007.de/client	One of these registered Redirection URI values MUST exactly match the redirect_uri parameter value used in each Authorization Request
`response_types`	[*code*]	-	
`client_registration_types`	[*automatic*]	-	gemäß OpenID Connect Federation 1.0 (section-4.1)
`grant_types`	[*authorization_code*]	-	OpenID Connect Dynamic Client Registration 1.0 (section-2)
`require_pushed_authorization_requests`	*true*	-	OAuth 2.0 Pushed Authorization Requests (section-6)

`token_endpoint_auth_method`	*private_key_jwt*	-	
`token_endpoint_auth_signing_alg`	*ES256*	-	
`default_max_age`	*0*	-	Default Wert um immer eine erneute Authentisierung des Nutzers zu erzwingen
`default_acr_values`	"gematik-ehealth-loa-high" oder "gematik-ehealth-loa-substancial"	"gematik-ehealth-loa-high"	
`id_token_signed_response_alg`	*ES256*	-	Weitere Werte sind möglich.
`id_token_encrypted_response_alg`	*ECDH-ES*	-	Weitere Werte sind möglich.
`id_token_encrypted_response_enc`	*A256GCM*	-	Weitere Werte sind möglich.
`scope`	[string]	[profile telematik openid]	String mit Space-delimited `scope` Values
}			
federation_entity{			
`name`	string	"Fachdienst007"	Optional: Name des Fachdienstes - wird z. B., genutzt in der

			Consent-Freigabe des Benutzers (redundant zum client_name)
`contacts`	strings	"Support@Fachdienst007.de "	Optional
`homepage_uri`	URL	"https://Fachdienst007.de"	Optional
}}			

Weitere Informationen zu den Inhalten zur Client-Registrierung finden sich in den Spezifikationen zum OIDC Standard:

- OpenID Connect Federation 1.0 (section-3.1)
- OAuth 2.0 Dynamic Client Registration Protocol (section-2)
- OpenID Connect Dynamic Client Registration 1.0
- The OAuth 2.0 Authorization Framework (section-3.3) https://openid.net/specs/openid-connect-federation-1_0.html - OpenID.Registration

signed_jwks

Ablageort für weitere Schlüssel des Fachdienstes etwa die zur Signatur seiner `private_key_jwt` Authentisierungsobjekte (`use` = "sig") oder für die Verschlüsselung der`ID_Token` (`use` = "enc").

Wenn eine `signed_jwks_uri` im Entity Statement angegeben ist müssen auch diese Schlüssel importiert werden.

Folgende Werte müssen im Header des selbst signierten KeySet des Fachdienstes auftauchen:

Tabelle 25 : Header des KeySet des Fachdienstes

Name	Werte	Beispiel	Anmerkungen
`alg`	*ES256*	-	
`kid`	wie aus jwks im Body des Dokumentes	"Fachdienst007-42"	Identifier des verwendeten Schlüssels aus dem jwks im Body des Entity Statement

typ	*JWT*	-	

1911 Folgende Werte müssen im Body enthalten sein:

1912 **Tabelle 26 : Body des KeySet des Fachdienstes**

Name	Werte	Beispiel	Anmerkungen
keys {			
kty		EC	
kid		Fachdienst007-42 / Fachdienst007-69	
crv		P-256 / P-256	
x		qAOdPQROkHfZY1daGofOmSNQWpYK8c9G2m2Rbkpbd4c /	
y		G_7fF-T8n2vONKM15Mzj4KR_shvHBxKGjMosF6FdoPY / ...	
use		sig / enc	nach JSON Web Key (section-4.2) Der Fachdienst listet sowohl sig als auch enc Schlüssel
}			
iss	URL	"https://Fachdienst007.de"	URL des IDP (variabel je Mandant/Kasse)
iat	Alle time Werte in Sekunden	1645484401	

	seit 1970, RFC 7519 Sect.2,		

(2-c) Abruf des Entity Statement zum Fachdienst beim Federation Master

Request:

HTTP-GET

Adresse: "http://master0815.de/federation_api_endpoint"

HTTPS GET Request an den federation_api_endpoint aus dem Entity Statement des Federation Master mit dem folgenden Parameter:

Tabelle 27 : Parameter HTTPS GET Request an Federation Master API

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	"http://master0815.de"	issuer des Federation Master - Verpflichtender Parameter für unser Szenario aber ohne Relevanz
`sub`	URL	"https://Fachdienst007.de"	issuer des angefragten Fachdienst

(2-d) Der Federation Master sendet sein Entity Statement über den Fachdienst zurück

Response:

HTTP 200 mit Content-Type: application/jose

Folgende Werte müssen im Header zum Entity Statement des Federation Master über den Fachdienst enthalten sein:

Tabelle 28 : Header zum Entity Statement des Federation Master über den Fachdienst

Name	Werte	Beispiel	Anmerkungen
`alg`	ES256	-	
`kid`	wie aus jwks im Body des Dokumentes	"master0815-1"	Identifier des verwendeten Schlüssels aus dem jwks im Body des Statement
`typ`	*JWT*	-	

Folgende Werte müssen im Body des Entity Statement des Federation Master über den Fachdienst enthalten sein:

Tabelle 29 : Body zum Entity Statement des Federation Master über den Fachdienst

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	"http://master0815.de"	URL des Federation Master
`sub`	URL	"https://Fachdienst007.de"	URL des angefragten Fachdienstes
`iat`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645398001	2022-02-21 00:00:01
`exp`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645480801	Beispielhafte Gültigkeit von 1 Tag für Möglichkeit der Sperrung
`jwks`	JWKS Objekt	unter anderem "Fachdienst007-42"	Schlüssel für die Signatur des EntityStatement

Als Ergebnis des Schritts (2-d) kennt der IDP die öffentlichen Keys des Fachdienstes für Verschlüsselung und Authentisierung.

(3) Der Authentication-Endpunkt (Auth EP) des sektoralen IDP antwortet dem AS des Fachdienstes mit einer Request URI

Zuvor verifiziert der IDP die Signatur des private_key_jwt gegen einen Schlüssel aus dem Entity Statement des Fachdienstes.
Response:

HTTP 201 mit Content-Type: application/json

Tabelle 30 : Parameter der HTTP-Response

Name	Werte	Beispiel	Anmerkungen
`request_uri`	URI	urn:Fachdienst007:bwc4JK-ESC0w8acc191e-Y1LTC2	URI zur späteren Identifikation des Request
`expires_in`	Gültigkeitsdauer der URI	90	nach RFC 6749 - max. 90 Sekunden scheint praktikabel

Diese URI wird als redirect an das Anwendungsfrontend gesendet um über das Authenticator-Modul den IDP zu erreichen.

(4) Der Authorization-Server des Fachdienstes antwortet dem Frontend mit einem redirect und seiner Request URI

HTTP-302,

Mit mindestens den folgenden HTTP-Header Elementen:

- Location

Die Location setzt sich zusammen aus:

<target_url><authorization request IDP Dienst zu sektoralem IDP>

Die target_url entspricht dabei der Adresse des Authorization-Endpunktes des sektoralen IDP entsprechend dem Entity Statement, welche auf dem Gerät auf das Authenticator-Modul weitergeleitet wird.

Der Request des Fachdienstes AS zum sektoralen IDP enthält dabei die folgenden Parameter:

Tabelle 31 : Request Parameter des Fachdienstes zum sektoralen IDP

Name	Werte	Beispiel	Anmerkungen
`client_id`	VSCHAR (max 32 Zeichen)	"https://Fachdienst007.de"	Hier muss die URL des Fachdienstes eingetragen werden = seine `client_id` in der Föderation
`request_uri`	URI	urn:Fachdienst007:bwc4JK-ESC0w8acc191e-Y1LTC2	URI zur späteren Identifikation des Request

(5) Das Anwendungsfrontend sendet den Authentication Request an die URI des IDP und leitet ihn somit an das Authenticator-Modul weiter

Das Anwendungsfrontend sendet ein HTTP-GET an den Authorization-Endpunkt des sektoralen IDP.

Die GET-Parameter entsprechen dem Request des Fachdienstes aus Schritt 4.

Das Authenticator-Modul des sektoralen IDP fängt diesen Request dadurch, dass er diese Adresse für App2App Kommunikation im Betriebssystem registriert hat.

(6) Das Authenticator-Modul leitet den Authentication Request an den IDP weiter (proprietär)

Die Schritte zur Nutzer-Authentifizierung und zur Erstellung des `AUTHORIZATION_CODE` durch den IDP sind anwendungsspezifisch und werden hier nicht weiter spezifiziert.

(7) Der Authorization-Endpunkt des sektoralen IDP antwortet dem Authenticator-Modul mit einem Redirect zum Fachdienst (proprietär)

Beispielsweise

HTTP-302,

Mit mindestens den folgenden HTTP-Header Elementen:

- Location

Die Location setzt sich zusammen aus:

<uri_Fachdienst_AS>?code=<AUTHORIZATION_CODE_IDP>&state=<state_Fachdienst>

Tabelle 32 : Parameter des Redirect-Request

Name	Werte	Beispiel	Anmerkungen
`uri_Fachdienst_AS`	URI	https://Fachdienst007.de/AS	`redirect_uri` aus der Anfrage in Schritt 2
`code`	max. 2000 Zeichen	AUTHORIZATION_CODE_IDP	`AUTHORIZATION_CODE` des sektoralen IDP
`state`	VSCHHAR (max 32 Zeichen)	state_Fachdienst	`state` des Fachdienstes um den Code zu dereferenzieren

(8) Das Authenticator-Modul des IDP ruft über einen App-Link bzw. Universal-Link entsprechend der Redirect-URL das Anwendungsfrontend auf und übergibt den `AUTHORIZATION_CODE`

Der App-Link bzw. Universal-Link Aufruf des Authenticator-Modul ist anwendungsspezifisch und wird hier nicht weiter spezifiziert.

Das Anwendungsfrontend fängt diesen Request dadurch, dass er diese Adresse für App2App Kommunikation im Betriebssystem registriert hat.

(9) Das Anwendungsfrontend leitet den `AUTHORIZATION_CODE` an den Autorisierungsserver des Fachdienstes

HTTP-POST (Content-Type: application/x-www-form-urlencoded) nach uri_Fachdienst_AS

Der Request des enthält dabei die folgenden Parameter:

Tabelle 33 : Parameter des POST-Request

Name	Werte	Beispiel	Anmerkungen
`code`	maximal 2000 Zeichen	AUTHORIZATION_CODE_IDP	`AUTHORIZATION_CODE` des sektoralen Identity Provider

state	VSCHAR (max 32 Zeichen)	state_Fachdienst	state des Fachdienstes um den Code zu dereferenzieren

(10) Der Autorisierungsserver reicht den AUTHORIZATION_CODE(IDP), den CODE_VERIFIER und seinen private_key_jwt beim Token-Endpunkt des IDP ein

HTTP POST mit Content-Type: application/x-www-form-urlencoded

Die folgenden Parameter werden im payload verwendet:

Tabelle 34 : HTTP-POST Parameter

Name	Werte	Beispiel	Anmerkungen
grant_type	*authorization_code*	-	
code	<AUTHORIZATION_CODE des sektoralen IDP base64-kodiert> - max. 2000 Zeichen	AUTHORIZATION_CODE_IDP	AUTHORIZATION_CODE des sektoralen IDP
code_verifier	<CODE_VERIFIER des Fachdienstes>	code_verifier_Fachdienst	
client_id	URL	"https://Fachdienst007.de"	URL des Fachdienstes = seine client_id
redirect_uri	URL	"https://Fachdienst007.de/AS"	
client_assertion_type	"*urn:ietf:params:oauth:client-assertion-type:jwt-bearer*"	-	Notwendiger Parameter nach OpenID Connect Core 1.0 (section-9)
client_assertion	private_key_jwt	siehe unten	

private_key_jwt

Das private_key_jwt ist mittels ES256 signiert und der Header hat folgende Inhalte:

Tabelle 35 : Header des private_key_jwt

Name	Werte	Beispiel	Anmerkungen

alg	ES256	-	
typ	JWT	-	
kid	wie aus jwks (oder signed_jwks) im Body des Entity Statement	"Fachdienst007-42"	Der öffentliche Schlüssel muss auch im Entity Statement des Fachdienstes stehen (der Einfachheit halber wird im Beispiel derselbe Schlüssel für alle Signaturen genutzt)

Das eigentliche Datenobjekt sieht wie folgt aus:

Tabelle 36 : Inhalt des private_key_jwt

Name	Werte	Beispiel	Anmerkungen
iss	URL	"https://Fachdienst007.de"	client_id des Fachdienstes
sub	URL	"https://Fachdienst007.de"	client_id des IDP Dienstes
aud	URL	"https://idp4711.de/Token"	URL des Token-Endpunkts des sektoralen IDPs
jti	random max. 32 Zeichen	PR3CQWQQXvPoLxy8CoAfMpBI28F2hxBf	
exp	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645565043	Vorgesehen ist eine Gültigkeit von jeweils 90 Sekunden
iat	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645564953	2022-02-22 22:22:33

Siehe OpenID Connect Core 1.0 (section-9)

(11) Der Autorisierungsserver erhält vom Token-Endpunkt des IDP einen ID_TOKEN und ACCESS_TOKEN mit den gewünschten claims, der mit dem öffentlichen Schlüssel aus der Registrierung verschlüsselt ist

Der Autorisierungsserver des Fachdienstes entschlüsselt den ID_TOKEN und verifiziert anschließend dessen Signatur. Damit endet der innere Flow.

HTTP-200:

- Content-Type=application/json
- Cache-Control=no-store
- Pragma=no-cache.

Die JSON-Struktur sieht so aus:

{
"access_token": <ACCESS_TOKEN>,
"id_token": <ID_TOKEN>,
"token_type": "Bearer",
"expires_in": 300, (Gültigkeit desACCESS_TOKEN in Sekunden, The OAuth 2.0 Authorization Framework (section-4.2.2))
}

Der ACCESS_TOKEN wird ignoriert.

Der Encryption Header-claims des ID_TOKEN sieht dabei wie folgt aus:

Tabelle 37 : Header-claims des ID_TOKEN des sektoralen IDP

Name	Werte	Beispiel	Anmerkungen
alg	*ECDH-ES*	-	
enc	*A256GCM*	-	
kid	wie aus signed_jwks	"Fachdienst007-69"	Ein Schlüssel mit der use="enc" aus demsigned_jwks des Fachdienstes
cty	JWT	-	

Signature Header-claimsdes ID_TOKEN sind genau die folgenden:

Tabelle 38 : Signature Header-claims des ID_TOKEN des sektoralen IDP

Name	Werte	Anmerkungen
alg	*ES256*	P256 wird zugelassen
typ	*JWT*	
kid	wie aus jwks in Entity Statement des sektoralen IDP	Für die Signatur des Token verwendeter Schlüssel

2033 Die Body-`claims` ür den `ID_TOKEN` des sektoralen IDP sind Beispielsweise die folgenden:

2034 **Tabelle 39 : Body-`claims` für den `ID_TOKEN` des sektoralen IDP**

Name	Werte	Beispiel	Anmerkungen
`iss`	URL	https://idp4711.de	Adresse des sektoralen IDP / reicht als Authentizitätsnachweis
`sub`	Beliebig, aber eindeutig je Nutzer und fest je Fachdienst.	"UserC3PO-666"	Wird als pseudonymer Identifier verwendet und ist einzig relevanter `claim` für Dienste die keiner Nutzerdaten erhalten sollen oder wollen.
`professionOID`	OID	1.2.276.0.76.4.49	Wird immer mit der OID des Versicherten belegt Abhängig von `scope`/`claims`
`iat`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645565035	2022-02-22 22:23:55
`exp`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645565335	Zeitliche Gültigkeit des Token von 5 Minuten
`given_name`	max. 64 Zeichen	-	Wird zur Anzeige verwendet und durch Kassen belegt. Möglich ist hier z. B. die Verwendung des Wertes von "givenName" wie im X.509-Zertifikat der eGK (spezifiziert in [gemSpec_PKI_V2] Kap. 5.1.2 in GS-A_4593)

			Abhängig von `scope/claims`
`family_name`	max. 64 Zeichen	-	Wird zur Anzeige verwendet und durch Kassen belegt. Möglich ist hier z. B. die Verwendung des Wertes von "surname" wie im X.509-Zertifikat der eGK (spezifiziert in [gemSpec_PKI_V2] Kap. 5.1.2 in GS-A_4592) Abhängig von `scope/claims`
`organization_name`	max. 64 Zeichen	-	IK-Nummer der Kasse. Abhängig von `scope/claims`
`idNummer`	10 Zeichen (für KVNR)	-	Hier muss die KVNR rein Abhängig von `scope/claims`
`aud`	URL	"https://Fachdienst007.de"	Die `client_id` des Fachdienstes - dieser hat die Anfrage gestellt.
`nonce`	max. 32 Zeichen	274312:dj83hs9s	
`auth_time`	Alle time Werte in Sekunden seit 1970, RFC 7519 Sect.2,	1645568630	Wenn max_age angefragt wurde ist dieser `claim` verpflichtend
`acr`	"gematik-ehealth-loa-high" oder "gematik-ehealth-loa-substancial"	gematik-ehealth-loa-high	Stärke der durch den IDP durchgeführten Authentisierung des Nutzers

weitere `claims`			Weitere `claims` sind optional und werden nicht ausgewertet.

(12) Der Autorisierungsserver des Fachdienstes erstellt ein `AUTHORIZATION_CODE` und sendet diesen an das Anwendungsfrontend zum Einreichen beim Token Endpunkt

Beispielsweise

HTTP-302,

Mit mindestens den folgenden HTTP-Header Elementen:

- Location

Die Location setzt sich zusammen aus:

< https://Fachdienst007.de/Token>?code=<authorization code AS>&state=<state Frontend>

Tabelle 40 : Parameter des Redirect-Request

Name	Werte	Beispiel	Anmerkungen
`code`	max. 2000 Zeichen	AUTHORIZATION_CODE_AS	`AUTHORIZATION_CODE` des Fachdienstes
`state`	VSCHAR (max 32 Zeichen)	af0ifjsldkj	`state` des Frontend um den Code zu dereferenzieren

(13) Anwendungsfrontend übergibt dem Autorisierungsserver den `AUTHORIZATION_CODE` sowie den `CODE_VERIFIER`

HTTP POST mit Content-Type: application/x-www-form-urlencoded

Die folgenden Parameter werden im payload verwendet:

Tabelle 41 : Parameter HTTP-POST

Name	Werte	Beispiel	Anmerkungen
`grant_type`	*authorization_code*	-	
`code`	<`AUTHORIZATION_CODE` des Fachdienstes base64-kodiert> - max. 2000 Zeichen	AUTHORIZATION_CODE_AS	`AUTHORIZATION_CODE` des Fachdienstes
`code_verifier`	<`CODE_VERIFIER` des Fachdienstes>	code_verifier_Frontend	

client_id	VSCHAR (max 32 Zeichen)	"eRezeptApp"	
redirect_uri	URI	"https://Fachdienst007.de"	

(14) Anwendungsfrontend erhält ACCESS_TOKEN und REFRESH_TOKEN mit den notwendigen Daten vom Autorisierungsserver des Fachdienstes

HTTP-200:

- Content-Type=application/json
- Cache-Control=no-store
- Pragma=no-cache.

Die JSON-Struktur sieht so aus:

{

"access_token": <ACCESS_TOKEN>,

"refresh_token": <REFRESH_TOKEN>,

"token_type": "Bearer",

"scope": "e-rezept",

"expires_in": 300, (Gültigkeit desACCESS_TOKEN in Sekunden, The OAuth 2.0 Authorization Framework (section-4.2.2))

}

(15) Das Anwendungsfrontend greift auf die Fachdienst API zu und übergibt dabei das ACCESS_TOKEN

Die Kommunikation zwischen Anwendungsfrontend und Fachdienst ist anwendungsspezifisch und wird hier nicht weiter spezifiziert.

7.2 Web-App-Flow

Der Web-App-Flow beschreibt die Einzelschritte für die Authentifizierung eines Nutzers im Rahmen einer Web-Anwendung, welche im Browser des selben Geräts ausgeführt wird, auf dem auch die Authenticator-App installiert ist.

7.2.1 Vorbedingungen Web-App-Flow

- Registrierung der Fachanwendung als Relying Party (RP) beim Federation Master.
- Registrierung des App-Link/Universal-Link für das Authenticator-Modul des IDP auf dem Gerät des Nutzers (auf Adresse des IDP) - oder anfragen über Web.
- Aktueller Signaturschlüssel des Federation Master ist bekannt und vertrauenswürdig bei IDP und Fachdienst eingebracht worden.

- Sektorale IDP ist Teil des TI-Vertrauensraums und beim Federation Master registriert.
- Der Fachdienst besitzt ein Web-Backend welches Anwendungslogik realisiert.

7.2.2 Flow-Diagramm Web-App-Flow

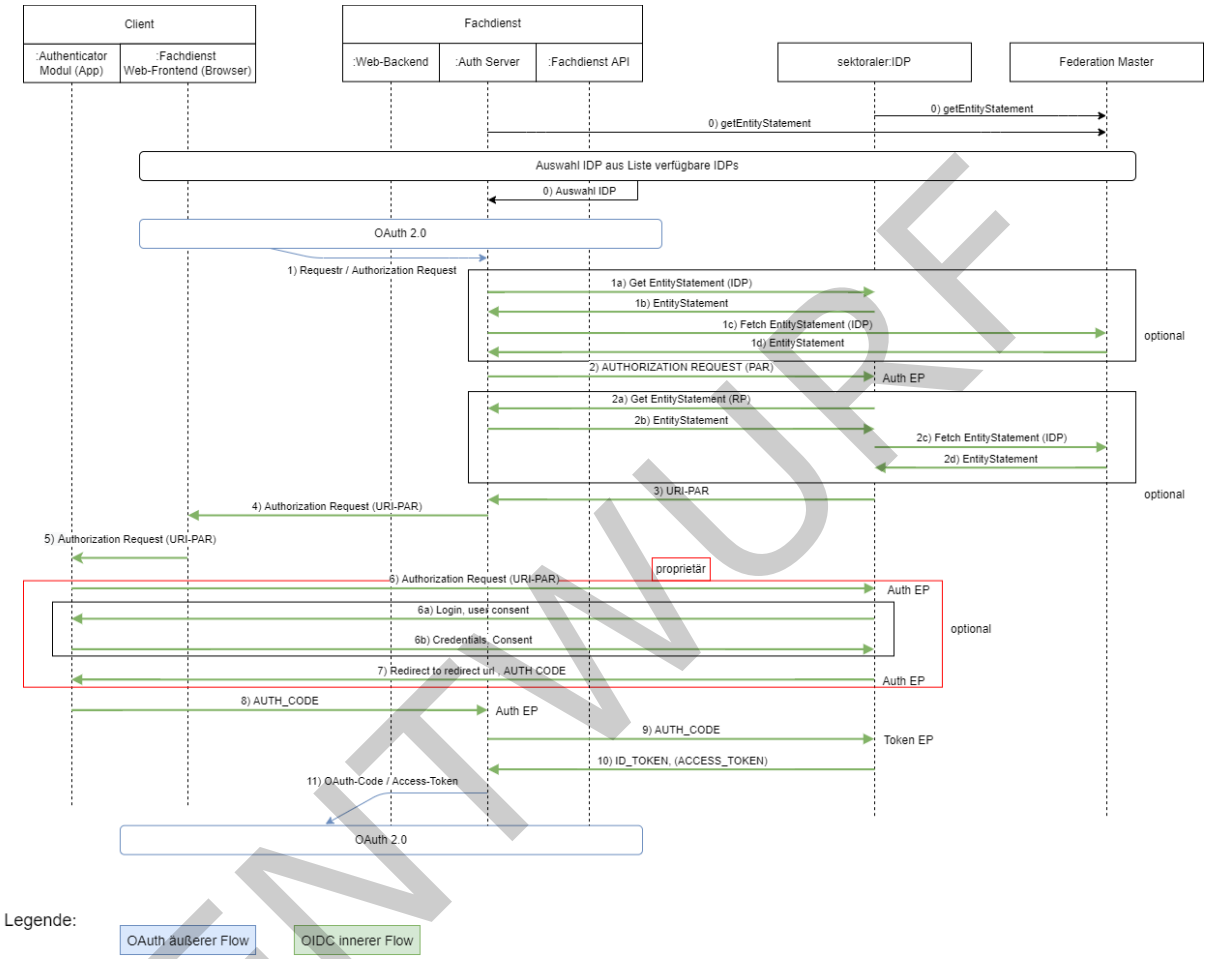

Abbildung 8 : Web-App-Flow

7.2.3 Ablaufbeschreibung Web-App-Flow

Tabelle 42 : Ablaufbeschreibung Web-App-Flow

Schritt	Teilschritt	Beschreibung

0		• Abruf der Schlüssel des Federation Master
		• Flow zur Auswahl des IDP:
		  • Die Auswahl des richtigen IDP ist optional. Ist der IDP bekannt (z. B. durch eine frühere Autorisierung) entfällt der Schritt
		  • Es gibt unterschiedliche Möglichkeiten den Ablauf der IDP-Ermittlung zu gestalten. Spätestens zum Schritt (1a) muss der Ziel-IDP bekannt sein

1		Abweichend vom App/App-Flow kommt der Request vom Web-Backend der Anwendung und nicht von einem Anwendungsfrontend (App)
	1-a	Schnittstellendetails analog App-zu-App Flow (1a)
	1-b	Schnittstellendetails analog App-zu-App Flow (1b)
	1-c	Schnittstellendetails analog App-zu-App Flow (1c)
	1-d	Schnittstellendetails analog App-zu-App Flow (1d)
2		Schnittstellendetails analog App-zu-App Flow (2)
	2-a	Schnittstellendetails analog App-zu-App Flow (2a)
	2-b	Schnittstellendetails analog App-zu-App Flow (2b)
	2-c	Schnittstellendetails analog App-zu-App Flow (2c)
	2-d	Schnittstellendetails analog App-zu-App Flow (2d)
3		Schnittstellendetails analog App-zu-App Flow (3)
4		Abweichend vom App/App-Flow läuft der Redirect über das Web-Backend zum Web-Frontend. Schnittstellendetails analog App-zu-App Flow (4)
5		Schnittstellendetails analog App-zu-App Flow (5)

6		Schnittstellendetails analog App-zu-App Flow (6)
	6-a	Schnittstellendetails analog App-zu-App Flow (6a)
	6-b	Schnittstellendetails analog App-zu-App Flow (6b)
7		Schnittstellendetails analog App-zu-App Flow (7)
8		Abweichend vom App/App Flow führt das Authenticator-Modul des IDP den Redirect zum Authorization-Service des Fachdienstes aus und übergibt den `AUTHORIZATION_CODE`. Schnittstellendetails analog App-zu-App Flow (9)
9		Schnittstellendetails analog App-zu-App Flow (10)
10		Schnittstellendetails analog App-zu-App Flow (11)
11		Der Autorisierungsserver des Fachdienstes reicht `ACCESS_TOKEN` und `REFRESH_TOKEN` an das Web-Backend der Anwendung weiter. Diese liegen zu keiner Zeit im Browser des Nutzers.
12		Der `ACCESS_TOKEN` (`REFRESH_TOKEN`) wird im Web-Backend der Anwendung persistiert. Die Kommunikation zwischen Web-Frontend und Web-Backend ist implementierungsspezifisch. Der Zugriff auf das Fachdienst-API erfolgt über das Web-Backend. Der `ACCESS_TOKEN` muss bei jedem Zugriff mitgegeben werden.

7.2.4 Detailinformationen zum Web-App-Flow

(1) Authorization Request von Web-Backend zum Authentication-Endpunkt (Auth ES) des Autorisierungsservers des Fachdienstes

Die Kommunikation zwischen Web-Frontend und Web-Backend ist anwendungsspezifisch. Das Web-Backend des Fachdienstes sendet einen Request an den Autorisierungsserver des Fachdienstes. Dieser Request ist ebenfalls anwendungsspezifisch. Damit der weitere Ablauf OIDC konform und weitest gehend identisch zum App-zu-App Flow ablaufen kann, muss der Request einigen Festlegungen genügen.

Das Web-Backend sendet ein HTTP-GET an den AS des Fachdienstes.

Die folgenden GET-Parameter werden im query string verwendet:

2106 **Tabelle 43 : Parameter des GET-Requests**

Name	Werte	Beispiel	Anmerkungen
client_id	VSCHAR (max 32 Zeichen)	"digaxy"	kein ";" und kein "┼" (definiert gem. Unicode U+253C (9532)), kein Leerzeichen
state	VSCHAR (max 32 Zeichen)	af0ifjsldkj	optional
redirect_uri	URL	"https://Fachdienst007.de"	Adresse des Fachdienstes weil da soll der ACCESS_TOKEN am Ende landen.
code_challenge	Hash über CODE_VERIFIER	K2-ltc83acc4h0c9w6ESC_rEMTJ3bww-uCHaoeK1t8U	PKCE optional weil Kommunikation innerhalb der Anwendung und nichts zum Browser fließt oder Redirects folgt.
code_challenge_method	*S256*	-	PKCE optional, siehe oben
response_type	*code*	-	CODE Flow optional, wenn andere Mechanismen die Verbindung schützen
scope	[string]	"e-rezept"	anwendungsspezifisch zu definieren kein openid

weitere `claims`			weitere `claims` können vereinbart werden
`kk_app_id`	max. 32 VSCHAR	kk_app_4711	
`idp_iss`	URL	"https://idp4711.de"	nicht Standard Parameter iss URL des IDP den der Nutzer für die Authentisierung ausgewählt hat. Optional - nötig, wenn Auswahl des IDP im Frontend passiert.

(1-a) Falls der Autorisierungsserver des Fachdienstes das Entity Statement des IDP noch nicht kennt, lädt er dies herunter

Request analog App-zu-App Flow (1a).

(1-b) Der IDP sendet sein Entity Statement zurück

Response analog App-zu-App Flow (1b)

signed_jwks

Die Werte sind analog zu App-zu-App Flow (1-signed_jwks).

(1-c) Der Autorisierungsserver des Fachdienstes ruft das Entity Statement zum IDP beim Federation Master ab

Request analog App-zu-App Flow (1c).

(1-d) Der Federation Master sendet sein Entity Statement über den angefragten sektoralen IDP zurück

Response analog App-zu-App Flow (1d).

(2) Der Autorisierungsserver des Fachdienstes sendet ein Pushed Authorization Request an den Authentication-Endpunkt (Auth ES) des sektoralen IDP

HTTP-POST analog App-zu-App Flow (2).

private_key_jwt

Das private_key_jwt ist analog zu App-zu-App Flow (2-private_key_jwt).

(2-a) Falls der IDP das Entity Statement des Autorisierungsservers des Fachdienstes noch nicht kennt, lädt er dies herunter

Request analog zu App-zu-App Flow (2a).

(2-b) Der Autorisierungsserver des Fachdienstes sendet sein Entity Statement zurück und der IDP registriert ihn als Client

Response analog zu App-zu-App Flow (2b).

signed_jwks

Die Werte sind analog zu App-zu-App Flow (2b-signed_jwks).

(2-c) Abruf des Entity Statement zum Fachdienst beim Federation Master

Request analog zu App-zu-App Flow (2c).

(2-d) Der Federation Master sendet sein Entity Statement über den Fachdienst zurück

Response analog zu App-zu-App Flow (2d).

(3) Der Authentication-Endpunkt (Auth EP) des sektoralen IDP antwortet dem AS des Fachdienstes mit einer Request URI

Response analog zu App-zu-App Flow (3).

(4) Der Authorization-Server des Fachdienstes antwortet dem Frontend mit einem redirect und seiner Request URI

Abweichend vom App/App-Flow läuft der Redirect zum Web-Frontend.

Redirect analog zu App-zu-App Flow (4).

(5) Das Web-Frontend sendet den Authentication Request an die URI des IDP und leitet ihn somit an das Authenticator-Modul weiter

HTTP-GET analog zu App-zu-App Flow (5).

(6) Das Authenticator-Modul leitet den Authentication Request an den IDP weiter (proprietär)

Die Schritte zur Nutzer-Authentifizierung und zur Erstellung des `AUTHORIZATION_CODE` durch den IDP sind anwendungsspezifisch und werden hier nicht weiter spezifiziert.

(7) Der Authorization-Endpunkt des sektoralen IDP antwortet dem Authenticator-Modul mit einem Redirect zum Fachdienst (proprietär)

Redirect analog zu App-zu-App Flow (7).

(8) Das Authenticator-Modul des IDP ruft über die Redirect-URL den Autorisierungsserver des Fachdienstes auf und übergibt den `AUTHORIZATION_CODE`

Abweichend vom App/App Flow führt das Authenticator-Modul des IDP den Redirect zum Authorization-Service des Fachdienstes aus und übergibt den `AUTHORIZATION_CODE`. Der Request mit einem HTTP-OK quittiert.

- Zu klären ist, ob dieser Request den Browser wieder in den Fokus bringt (je nach Technologie iOS/Android unterschiedlich) oder ob der Nutzer hier selbst wechseln muss und nur nach Erhalt des HTTP-OK dazu aufgefordert wird, zur Anwendung im Browser zurückzukehren.
- Der Fachdienst-AS könnte hier auch mit einem Redirect zur Fachdienstadresse antworten, wenn das einen wieder in dieselbe Websession bringt.

HTTP-POST analog zu App-zu-App Flow (9).

(9) Der Autorisierungsserver reicht den `AUTHORIZATION_CODE`, den `CODE_VERIFIER` und seinen `private_key_jwt` beim Token-Endpunkt des IDP ein

HTTP POST analog zu App-zu-App Flow (10).

private_key_jwt

Das private_key_jwt ist analog App-zu-App Flow (10-private_key_jwt).

(10) Der Autorisierungsserver erhält vom Token-Endpunkt des IDP einen `ID_TOKEN` und `ACCESS_TOKEN` mit den gewünschten `claims`, der mit dem öffentlichen Schlüssel aus der Registrierung verschlüsselt ist

Response analog zu App-zu-App Flow (11).

(11) Der Autorisierungsserver des Fachdienstes reicht das ACCESS_TOKEN und REFRESH_TOKEN an das Web-Backend der Anwendung weiter

HTTP-200:

- Content-Type=application/json
- Cache-Control=no-store
- Pragma=no-cache.

Die JSON-Struktur sieht so aus:

{

"access_token": <ACCESS_TOKEN>,

"refresh_token": <REFRESH_TOKEN>,

"token_type": "Bearer",

"scope": "e-rezept",

"expires_in": 300, (Gültigkeit desACCESS_TOKEN in Sekunden, [The OAuth 2.0 Authorization Framework#section 4.2.2])

}

(12) Kommunikation Web-Frontend, Web-Backend der Anwendung und Fachdienst-API

Das Web-Backend persistiert ACCESS_TOKEN und REFRESH_TOKEN. Das Web-Backend benötigt diese für die autorisierte Kommunikation mit dem Fachdienst-API. Die Kommunikation zwischen Web-Frontend und Web-Backend ist implementierungsspezifisch. ACCESS_TOKEN und/oder REFRESH_TOKEN werden nicht an das Frontend weitergereicht.

Das Web-Backend verwendet das ACCESS_TOKEN für die Kommunikation mit dem Fachdienst-API. Das Fachdienst-API prüft den ACCESS_TOKEN bevor Anfragen entsprechend quittiert werden.

GET /resource/1 HTTP/1.1 Host: example.com Authorization: Bearer <ACCESS_TOKEN>

7.3 Zwei-Geräte-Flow

Der Zwei-Geräte-Flow beschreibt die Einzelschritte für die Authentifizierung eines Nutzers im Rahmen einer Fachanwendung wobei die Fachanwendung eine App oder Web-Anwendung ist, welche auf einem anderen Gerät als die Authenticator-App ausgeführt wird.

7.3.1 Vorbedingungen Zwei-Geräte-Flow

- Registrierung der Fachanwendung als Relying Party (RP) beim Federation Master
- Registrierung des App-Link/Universal-Link für das Authenticator-Modul des IDP auf dem Gerät des Nutzers (auf Adresse des IDP) - oder anfragen über Web.

- Aktueller Signaturschlüssel des Federation Master ist bekannt und vertrauenswürdig bei IDP und Fachdienst eingebracht worden.
- Sektorale IDP ist Teil des TI-Vertrauensraums und beim Federation Master registriert.
- Der Fachdienst besitzt ein Web-Backend welches Anwendungslogik realisiert.
- Authenticator-Modul des IDP (App) läuft auf einem anderen Gerät als die Fachanwendung (z. B. App → Smartphone, Anwendung → PC-Browser)

7.3.2 Flow-Diagramm Zwei-Geräte-Flow

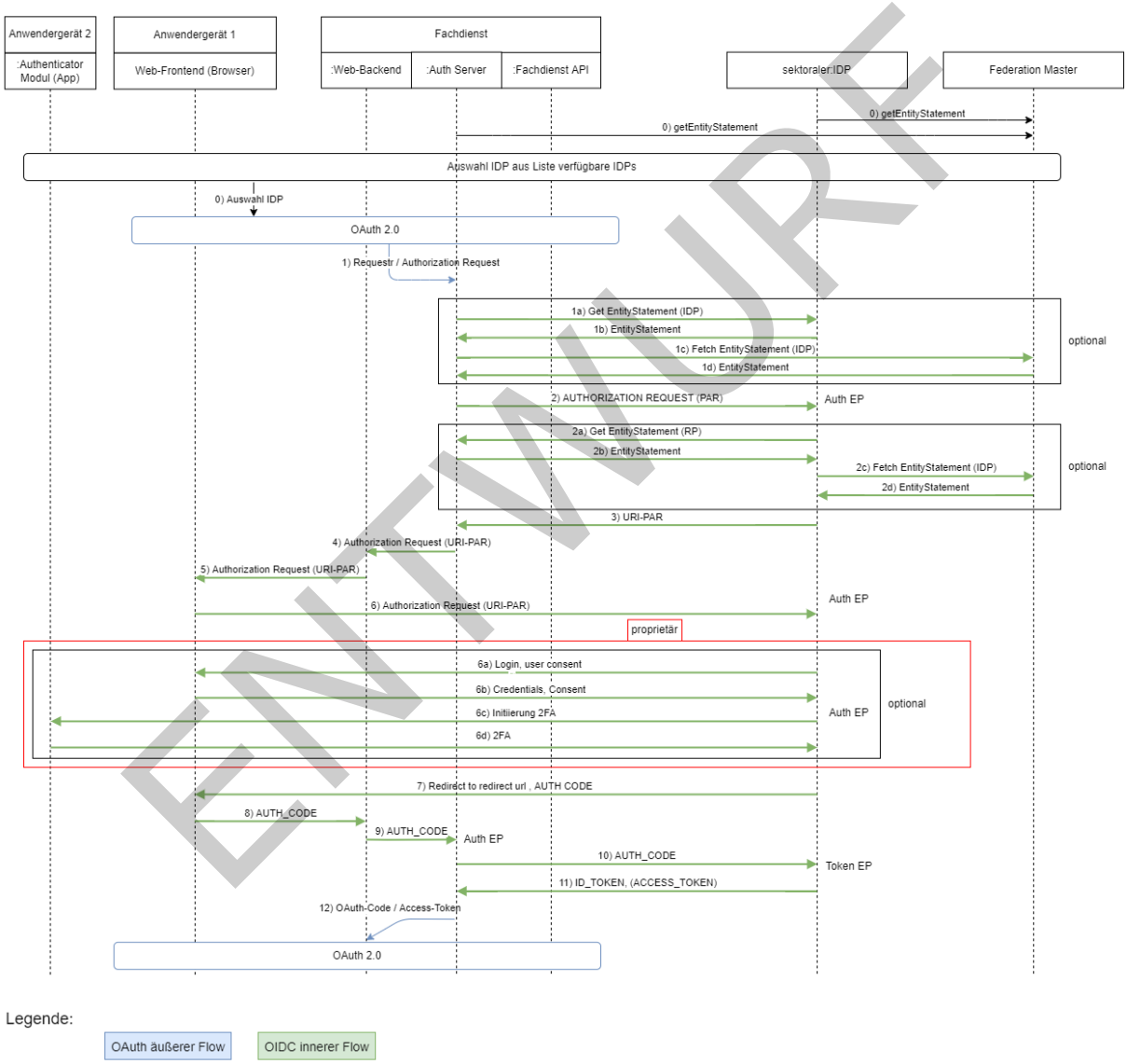

Abbildung 9 : Zwei-Geräte-Flow

7.3.3 Ablaufbeschreibung Zwei-Geräte-Flow

Tabelle 44 : Ablaufbeschreibung Zwei-Geräte-Flow

Schritt	Teilschritt	Gerät	Beschreibung

0		1	• Abruf der Schlüssel des Federation Master
			• Flow zur Auswahl des IDP
			  • Die Auswahl des richtigen IDP ist optional. Ist der IDP bekannt (z. B. durch eine frühere Autorisierung) entfällt der Schritt
			  • Es gibt unterschiedliche Möglichkeiten den Ablauf der IDP-Ermittlung zu gestalten. Spätestens zum Schritt (1a) muss der Ziel-IDP bekannt sein

1		1	Schnittstellendetails analog Web-zu-App Flow (1)
	1-a		Schnittstellendetails analog App-zu-App Flow (1a)
	1-b		Schnittstellendetails analog App-zu-App Flow (1b)
	1-c		Schnittstellendetails analog App-zu-App Flow (1c)
	1-d		Schnittstellendetails analog App-zu-App Flow (1d)
2			Schnittstellendetails analog App-zu-App Flow (2)
	2-a		Schnittstellendetails analog App-zu-App Flow (2a)
	2-b		Schnittstellendetails analog App-zu-App Flow (2b)
	2-c		Schnittstellendetails analog App-zu-App Flow (2c)
	2-d		Schnittstellendetails analog App-zu-App Flow (2d)
3			Schnittstellendetails analog App-zu-App Flow (3)

4			Der Autorisierungsserver antwortet dem Web-Backend mit Request-URI und Client ID zur Weiterleitung über das Anwendungsfrontend an die Adresse des Authenticator des IDP.
5		1	Das Web-Backend leitet den Redirect an das Anwendungsfrontend weiter.
6		1	Das Anwendungsfrontend öffnet die Web-Anwendung des IDP für den Authentifikationsprozess.
	6a	1	Das Web-Frontend des IDP erfragt die Zugangsinformationen und ggf. Consent-Freigabe für die anfragende Anwendung beim Nutzer (1. Faktor, z. B. user/password)
	6b		Der Nutzer übermittelt seine Credentials an den IDP.
	6c	2	Der IDP kann das Authenticator-Modul des IDP (z. B. 2FA) mit in den Prozess einbinden. Dazu sendet der IDP entweder eine push-Nachricht an die Authenticator-App oder fordert den Nutzer zum Start der Authenticator-App auf.
	6d		Der Nutzer tätigt die notwendigen Aktivitäten zur Authentifizierung über das Authenticator-Modul des IDP.
7		1	Der Authorization-Endpunkt des IDP antwortet dem Aufruf des Anwendungsfrontend (Schritt 6) mit dem `AUTHORIZATION_CODE` und einem Redirect zum Fachdienst.
8		1	Die Anwendungsfrontend leitet den `AUTHORIZATION_CODE`(IDP) an sein Web-Backend weiter.
9			Das Web-Backend leitet den `AUTHORIZATION_CODE`(IDP) an den Autorisierungsserver (redirected uri)
10			Schnittstellendetails analog App-zu-App Flow (10)
11			Schnittstellendetails analog App-zu-App Flow (11)
12		1	Schnittstellendetails analog Web-zu-App Flow (11)

7.3.4 Detailinformationen zum Zwei-Geräte-Flow

(1) Authorization Request von Web-Backend zum Authentication-Endpunkt (Auth ES) des Autorisierungsservers des Fachdienstes

- Web-Anwendung → Request analog Web-zu-App Flow (1).

(1-a) Falls der Autorisierungsserver des Fachdienstes das Entity Statement des IDP noch nicht kennt, lädt er dies herunter

Request analog zu App-zu-App Flow (1a).

(1-b) Der IDP sendet sein Entity Statement zurück

Response analog zu App-zu-App Flow (1b).

signed_jwks

Die Werte sind analog zu App-zu-App Flow (1-signed_jwks).

(1-c) Der Autorisierungsserver des Fachdienstes ruft das Entity Statement zum IDP beim Federation Master ab

Request analog zu App-zu-App Flow (1c).

(1-d) Der Federation Master sendet sein Entity Statement über den angefragten sektoralen IDP zurück

Response analog zu App-zu-App Flow (1d).

(2) Der Autorisierungsserver des Fachdienstes sendet ein Pushed Authorization Request an den Authentication-Endpunkt (Auth ES) des sektoralen IDP

HTTP-POST analog zu App-zu-App Flow (2).

private_key_jwt

Das private_key_jwt ist analog zu App-zu-App Flow (2-private_key_jwt).

(2-a) Falls der IDP das Entity Statement des Autorisierungsservers des Fachdienstes noch nicht kennt, lädt er dies herunter

Request analog zu App-zu-App Flow (2a):

(2-b) Der Autorisierungsserver des Fachdienstes sendet sein Entity Statement zurück und der IDP registriert ihn als Client

Response analog zu App-zu-App Flow (2b).

signed_jwks

Die Werte sind analog zu App-zu-App Flow (2b-signed_jwks).

(2-c) Abruf des Entity Statement zum Fachdienst beim Federation Master

Request analog zu App-zu-App Flow (2c).

(2-d) Der Federation Master sendet sein Entity Statement über den Fachdienst zurück

Response analog zu App-zu-App Flow (2d).

(3) Der Authentication-Endpunkt (Auth EP) des sektoralen IDP antwortet dem AS des Fachdienstes mit einer Request URI

Response analog zu App-zu-App Flow (3).

(4) Der Authorization-Server des Fachdienstes antwortet dem Web-Backend mit einem redirect und seiner Request URI

Der Autorisierungsserver antwortet dem Web-Backend mit Request-URI und Client ID zur Weiterleitung über das Anwendungsfrontend an die Adresse des Authenticator des IDP.

(5) Das Web-Backend antwortet dem Frontend mit einem redirect und seiner Request URI

Das Web-Backend leitet den Redirect an das Anwendungsfrontend weiter.

(6) Das Web-Frontend öffnet die URI und damit eine Authentifizierungsseite des IDP

HTTP-GET analog zu App-zu-App Flow (5) - allerdings gibt es in diesem Fall eben kein Authenticator-Modul des sektoralen IDP auf dem Gerät und daher wird unter der Adresse eine Authentifizierungsseite im Browser geöffnet.

(6a-d) Anwender authentifiziert sich nach dem Verfahren des IDP

Der Anwender authentifiziert sich nach dem Verfahren des IDP. Dabei kann als 2. Faktor eine Authenticator-App auf einem 2. Gerät verwendet werden.

Beispielablauf:

6a) IDP Login-Seite im Browser Gerät 1 → Identifikation des Nutzers (möglicherweise/ratsam über ersten Faktor z. B. Name/Passwort)

6b) IDP → Prüfung der Credentials (Optional, wenn 1 Faktor genutzt)

6c) Initiierung des 2. Faktor durch Aufforderung an den Anwender zum Öffnen des Authenticator-Moduls auf einem 2. Gerät oder durch ein push des IDP auf das Gerät mit dem Authenticator-Modul

6d) Authenticator-Modul Gerät 2 → IDP → Abschluss der Authentisierung

Der Nutzer könnte auch einen Code vom IDP gezeigt bekommen im Schritt 6a und tippt/scannt diesem im Authenticator-Modul ein. Auch dies kann eine Kopplung der App zum Prozess beim IDP herstellen.

Varianten gibt es verschiedene aber es muss klar sein zu welcher Session (Request URI) beim IDP diese Authentisierung gehört.

(7) Der Authorization-Endpunkt des sektoralen IDP antwortet dem Web-Frontend (Browser) mit einem Redirect zum Fachdienst

Die Authentifizierungsseite des Authorization-Endpunktes des sektoralen IDP reagiert und sendet dem Web-Frontend einen Redirect zum Fachdienst und den `AUTHORIZATION_CODE`.

Redirect analog zu App-zu-App Flow (7).

(8) Das Web-Frontend (Browser) leitet den `AUTHORIZATION_CODE` an das Web-Backend der Anwendung weiter

Das Anwendungsfrontend gibt die Information mit dem `AUTHORIZATION_CODE` an das Web-Backend der Anwendung weiter.

(9) Das Web-Backend der Anwendung leitet den `AUTHORIZATION_CODE` an den Autorisierungsserver des Fachdienstes

HTTP-POST analog zu App-zu-App Flow (9).

(10) Der Autorisierungsserver reicht den `AUTHORIZATION_CODE`, den `CODE_VERIFIER` und seinen `private_key_jwt` beim Token-Endpunkt des IDP ein

HTTP-POST analog zu App-zu-App Flow (10).

private_key_jwt

Das private_key_jwt ist analog zu App-zu-App Flow (10-private_key_jwt).

(11) Der Autorisierungsserver erhält vom Token-Endpunkt des IDP einen `ID_TOKEN` und `ACCESS_TOKEN` mit den gewünschten `claims`, der mit dem öffentlichen Schlüssel aus der Registrierung verschlüsselt ist

Response analog zu App-zu-App Flow (11).

(12) Einlösen des `ACCESS_TOKEN` und Datenabruf

- Web-Anwendung → weiterer Ablauf analog ab Web-zu-App Flow (11).

8 Anhang C - Empfehlungen zum Aufbau der VAU

Der Schutzbedarf der durch den sektoralen IDP verarbeiteten Daten und der Zugriff auf personenbezogene medizinische Daten der durch ihn ermöglicht wird erfordert einen spezifischen Systemaufbau des Dienstes, durch den ein unberechtigter Zugriff auf diese Daten nicht nur über das Internet sondern auch aus dem Betriebsumfeld des Betreibers (z. B. durch einen oder mehrere Mitarbeiter des Betreibers), technisch ausgeschlossen wird.

Der Systemaufbau des sektoralen IDP ist darüber hinaus dadurch bestimmt, dass die Verfügbarkeit des Dienstes für einzelne Mandanten (Kostenträger) erhalten bleiben muss, auch wenn die Verfügbarkeit für andere Mandanten z. B. durch unerwartet hohe Aktivität eingeschränkt wird.

8.1 Standalone

Der Betreiber des sektoralen IDP stellt pro Instanz eine dedizierte Hardware-Umgebung zur Ausführung des Dienstes bereit. Empfehlungen der gematik zum Aufbau dieser Instanzen sind im Folgenden beschrieben.

Eine Instanz besteht aus:

1. einem Load Balancer, der die Rolle der in [gemSpec_IDP_Sek], Kap. 4.2 festgelegten Eingangspunkte erfüllt:
   - Endpunkt für Pushed Authorization Requests
   - Endpunkt für Authorization Request
   - Token-Endpunkt
   - Endpunkt für Datensynchronisation mit Bestandssystem
2. einer Konfiguration von Servern, die eine Vertrauenswürdige Ausführungsumgebung gemäß [gemSpec_IDP_Sek], Kap. 3.2 bilden – dies umfasst ein HSM für die Attestation der Server und die Handhabung des privaten Schlüssels der Identität der VAU - und
3. einem Datenbanksystem zur Aufnahme sämtlicher persistenter Daten der Anwendung in verschlüsselter Form.

Der Betreiber des sektoralen IDP wird sämtliche Systeme unterbrechungsfrei mit Strom versorgen.

Die folgende Abbildung zeigt den Aufbau einer Instanz der sektoralen IDP im Überblick:

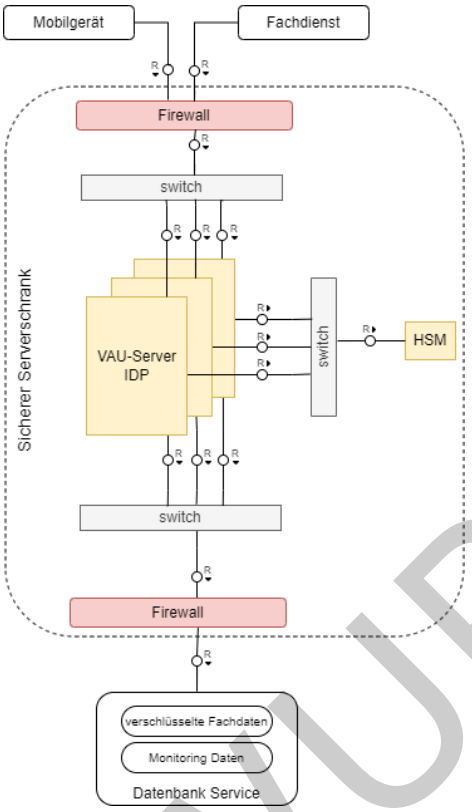

Abbildung 10 : 3.2.6 Umsetzungsempfehlungen für die Vertrauenswürdige Ausführungsumgebung

8.1.1 Load Balancer

Der Anbieter des sektoralen IDP realisiert die Netzanbindung der Server-Systeme des sektoralen IDP über einen Load Balancer mit folgenden Geräte- und Konfigurationseigenschaften:

- Netzanbindung: Der Load Balancer verfügt über öffentlich adressierbaren IP-Adressen. Die Anzahl der jeweils konfigurierten IP-Adressen ist so gewählt, dass die maximale Anzahl gleichzeitiger Client-Verbindungen, die sich aus dem angegebenen Mengengerüst ableitet, ermöglicht wird.
- Abwehr von DoS-Angriffen: Der Load Balancer kann in die Abwehr von DoS-Angriffen eingebunden sein (z. B. für die Abwehr von Syn-Flooding) und als Teil einer umfassenden (vorgelagerten) Infrastruktur zur Abwehr solcher Angriffe an der Limitierung von Netzverkehr mitwirken bzw. zur Angriffserkennung Meldungen über ungültige Aufrufe an die vorgelagerte Abwehr-Infrastruktur senden.
- Protokollierung: Der Load Balancer protokolliert alle für die Überwachung seines betrieblichen Zustands durch den Anbieter erforderlichen Daten.
- Stromversorgung: Der Load Balancer ist mit einer zweifach redundanten Stromversorgung ausgestattet.

8.1.2 Anwendungsserver und zugehörige Infrastruktur

Dem Schutzbedarf der im sektoralen IDP unverschlüsselt verarbeiteten Daten wird durch den Einsatz einer Vertrauenswürdigen Ausführungsumgebung (VAU) gemäß [gemSpec_IDP_Sek], Kap. 3.2 Rechnung getragen. Das in diesem Abschnitt beschriebene Subsystem des sektoralen IDP stellt die vertrauenswürdige Ausführungsumgebung dar.
Der Anbieter des sektoralen IDP stellt sämtliche Anwendungsserver auf denen die fachliche Logik des sektoralen IDP ausgeführt wird (VAU-Server), das HSM zur Attestation dieser Anwendungsserver und zur Bereitstellung und Anwendung des privaten Schlüssels der Dienstidentität des sektoralen IDP sowie sämtliche Komponenten zur Vernetzung der VAU-Server und des HSM in einem oder mehreren Serverschränken (VAU-Serverschränken) bereit. Ein VAU-Serverschrank ist ein Serverschrank der Schutzklasse WK 4, der zusätzlich mit folgenden Sicherheitsvorkehrungen ausgestattet ist:

- einer Abschirmung gegen elektromagnetische Abstrahlung insoweit diese geeignet ist, Daten aus der Verarbeitung innerhalb des Serverschranks von außerhalb des Serverschranks zu extrahieren,
- einem verstärkten Türschloss zur gegenüber Schutzklasse WK 4 verbesserten Abwehr bzw. Verzögerung von Versuchen zum gewaltsamen Eindringen in den Schrank,
- einem Türsensor, der im Falle eines unautorisierten Öffnens der Schranktür die Stromzufuhr aller im Schrank verbauten Systeme sofort unterbricht,
- einem Mechanismus zur Alarmierung bei unautorisierter Öffnung sowie
- einem Mechanismus zur elektronischen Entriegelung aus der Ferne der einen durch die gematik autorisierten Zugang von Mitarbeitern des Anbieter des sektoralen IDP zum Schrank ermöglicht.
  Der Anbieter des sektoralen IDP richtet die VAU-Serverschränke so ein, dass mehrere Gruppen von VAU-Servern verfügbar sind wie im Folgenden beschrieben:
  Jede Gruppe von VAU-Servern besteht aus mindestens 2 physisch separaten VAU-Servern. Die Anzahl der VAU-Server für jede Gruppe wird darüber hinaus durch die Lastanforderungen (für die jeweilige Gruppe) bestimmt. Die verschiedenen Gruppen von VAU-Servern sind jeweils verschiedenen Mandanten des sektoralen IDP zugeordnet:
- Die physische Trennung der VAU-Server dient der Sicherstellung der Verfügbarkeit des sektoralen IDP für die verschiedenen Mandanten im Falle einer (z. B. überlastbedingten) Einschränkung der Verfügbarkeit des sektoralen IDP.

8.1.3 Vernetzung Load-Balancer/VAU-Server

Der Anbieter des sektoralen IDP vernetzt die VAU-Server mit dem Load Balancer wie im Folgenden beschrieben:
Die Zuführung der Netzwerkverbindungen für Zugriffe aus dem Internet zu den VAU-Servern erfolgt über den Load Balancer und von diesem ausgehend über einen Switch und eine Firewall (Eingangs-Switch, bei mehreren VAU-Serverschränken Eingangs-Switches und Firewalls) innerhalb des VAU-Serverschranks. Der Eingangs-Switch, die Firewall und die VAU-Server sind so konfiguriert, dass jede der VAU-Servergruppen ein eigenes Subnetz bildet, dass VAU-Server keine Netzverbindungen untereinander aufbauen können und dass sämtlicher nicht vorgesehener Netzverkehr blockiert wird.

Den VAU-Servern sind die für die Anwendungsfunktionalität erforderlichen mandantenspezifischen URLs zugeordnet. Der Load Balancer ist so konfiguriert, dass er die Verteilung der Requests auf die VAU-Server aufgrund der URLs der Requests vornehmen kann. Der Load Balancer ist weiterhin so konfiguriert, dass er die Verteilung der Requests auf die einzelnen VAU-Server im Round-Robin-Verfahren vornehmen kann.

8.1.4 Vernetzung VAU-Server/HSM

Der Anbieter des sektoralen IDP vernetzt die VAU-Server mit dem HSM im VAU-Serverschrank wie im Folgenden beschrieben:
Alle VAU-Server sind individuell (über ein zweites Netzwerk-Interface der VAU-Server und einen zweiten Switch (HSM-Switch) innerhalb des VAU-Serverschranks) mit dem HSM vernetzt. Diese Vernetzung innerhalb des VAU-Serverschranks darf physisch und logisch nur VAU-Server und das HSM umfassen.
Falls mehr als ein VAU-Serverschrank für eine Instanz des sektoralen IDP erforderlich ist, darf ein einzelnes HSM in nur einem der VAU-Serverschränke von VAU-Servern in den weiteren VAU-Serverschränken mit genutzt werden. In diesem Fall muss die Vernetzung zwischen den Serverschränken als eine Switch-zu-Switch-Verbindung zwischen den dedizierten VAU-Server/HSM Netzen in den einzelnen VAU-Serverschränken ausgeführt sein. Der physische Aufbau der VAU-Serverschränke muss es dabei ausschließen, dass das Verbindungkabel von außerhalb der VAU-Serverschränke manipulierbar ist.
Der HSM-Switch (bei mehreren VAU-Serverschränken die HSM-Switches) und die VAU-Server sind so konfiguriert, dass VAU-Server keine Netzverbindungen untereinander aufbauen können.

8.1.5 Vernetzung VAU-Server/Datenbankserver

Der Anbieter des sektoralen IDP vernetzt die VAU-Server mit der außerhalb der VAU betriebenen Datenbank wie im Folgenden beschrieben:
Alle VAU-Server sind über ein drittes Netzwerk-Interface der VAU-Server und einen dritten Switch (Ausgangs-Switch) und eine Firewall innerhalb des VAU-Serverschranks mit dem Datenbankserver vernetzt.
Der Ausgangs-Switch und die Firewall (bei mehreren VAU-Serverschränken die Ausgangs-Switches und die Firewalls) und die VAU-Server sind so konfiguriert, dass VAU-Server keine Netzverbindungen untereinander aufbauen können und sämtlicher nicht vorgesehene Netzverkehr blockiert wird.

8.1.6 Vernetzung des Management Interface mit dem internen Netz des Anbieters des sektoralen IDP

Der Anbieter des sektoralen IDP stattet alle VAU-Serverschränke mit einem Management Interface mit niedriger Bandbreite (56kbps) aus, dass alle VAU-Server sowie das HSM erreichbar macht und nur die zwingend notwendigen betrieblichen Steuerungsmöglichkeiten zur Abfrage des elementaren Betriebszustands und für einen Start, Neustart sowie das kontrollierte Herunterfahren der Systeme anbietet.

8.1.7 VAU-Server

Die VAU-Server bilden den Kern der Vertrauenswürdigen Ausführungsumgebung. Neben ihrer grundsätzlichen Eignung als Anwendungsserver im Rechenzentrumsbetrieb, müssen VAU-Server zur Vertrauenswürdigkeit der Datenverarbeitung im sektoralen IDP beitragen.
Der Anbieter des sektoralen IDP wird VAU-Server einsetzen, die folgende Sicherheitseigenschaften aufweisen:

- Ein VAU-Server ist frei von Komponenten zur persistenten Speicherung von Daten mit Ausnahme der Firmware (Diskless Server).
- Ein VAU-Server verfügt über einen Boot Loader, der Boot Images über das Netzwerk laden und ihre Signatur gegen ein vorgegebenes, d. h. manipulationssicher konfiguriertes, Zertifikat prüfen kann.
- Ein VAU-Server unterstützt Measured Boot über die gesamte geladene Software sowie über sämtliche sicherheitsrelevanten Plattform-Konfigurationswerte (z. B. mittels eines TPM-Moduls).
- Ein VAU-Server unterstützt Remote Attestation in einer Form, die keine regelmäßige (d. h. bei jedem Systemstart notwendige) Einbindung von Diensten des Herstellers erfordert (z. B. dadurch, dass ein Sealing möglich ist, oder dass eine Attestation unabhängig von Diensten des Herstellers umgesetzt werden kann).
- Ein VAU-Server bietet Hardware-Unterstützung für die Speicherverwaltung (MMU und IOMMU) und die benötigten kryptographischen Primitiven.
  Der Einsatz des Boot Loaders mit Signaturprüfung der Boot Images dient primär dazu, das Laden ungeprüfter Softwarekomponenten zu verhindern, während die Attestation zur Sicherstellung der Integrität der Gesamtheit aus geprüfter Software und Systemkonfiguration dient.

8.1.8 VAU-Server Software Stack

Der Anbieter des sektoralen IDP wird die Software auf den VAU-Servern darauf auslegen, die Sicherheitsziele für die Server zu erreichen, indem der Software Stack (d. h. die Gesamtheit aller auf den Servern geladenen Software) minimalistisch ausgelegt ist (Minimal Trusted Computing Base). Die Software ist gehärtet und bietet einen robusten Mechanismus zur Separation, mittels dessen verschiedene Aspekte der Verarbeitung auf den VAU-Servern gegeneinander isoliert werden.
Der Anbieter des sektoralen IDP nutzt den Separationsmechanismus mindestens dazu, die System Management Funktionen zur Steuerung des Systems durch den Betreiber von der Verarbeitung der schützenswerten Daten zu trennen. Darüber hinaus soll der Anbieter des sektoralen IDP über den Separationsmechanismus eine Partitionierung umsetzen, die potenziell angreifbare Treiber und Protokolle von der Verarbeitung der schützenswerten Daten isoliert sowie die an die einzelnen Hardware-Netzwerkschnittstellen gebundenen Netzwerkfunktionen voneinander und vom Rest der Software trennt. Die Separation soll zudem dazu genutzt werden, die verschiedenen Funktionsmodule zur Ausführung der Fachlogik voneinander zu trennen. Zu beachten ist, dass trotz der Separationsmechanismen die Attestation der gesamten geladenen Software erfolgen muss.
Die Separation der einzelnen fachlichen Verarbeitungsvorgänge (Requests) innerhalb eines Funktionsmoduls voneinander kann der Anbieter des sektoralen IDP auf der Ebene der Anwendungssoftware umsetzen. Die Anwendungssoftware gehört zur Trusted

Computing Base der VAU. Ihre Sicherheits- und insbesondere ihre Separationseigenschaften müssen sicherheitstechnisch bewertbar sein.

8.1.9 Open Source Software Stack

Die Vertrauenswürdigkeit der VAU soll dadurch untermauert werden, dass VAU-Server im Rahmen des Machbaren für die Öffentlichkeit transparente Systeme darstellen. Interessierte Personen oder Organisationen müssen – die notwendige Fachkenntnis vorausgesetzt – anhand öffentlich verfügbarer Informationen in der Lage sein, im Detail nachzuvollziehen, wie die Systeme aufgebaut sind und funktionieren. Diese Nachvollziehbarkeit soll dadurch erreicht werden, dass es dem Anbieter des sektoralen IDP auferlegt wird, eine geeignete Auswahl für die technische Basis der VAU-Server zu treffen, um zu erreichen, dass die Softwarekomponenten der VAU-Server möglichst weitgehend öffentlich im Quellcode offengelegt sind.
Der offengelegte Quellcode ist fortlaufend auf dem Stand des produktiven Systems zu halten. Bei Änderungen an der Software in der Produktionsumgebung ist die öffentliche Dokumentation des Quellcodes unverzüglich zu aktualisieren.
Für alle Teile der Software der VAU, deren Quellcode nicht öffentlich gemacht werden kann, ist der zum jeweiligen Zeitpunkt gegebene binäre Stand dieser Software zu veröffentlichen.

8.1.10 Attestation und Integritätsschutz für VAU-Server

Der Anbieter des sektoralen IDP stattet die VAU-Server mit der Fähigkeit zur Remote Attestation auf der Basis der beim Booten des Systems und beim Laden sämtlicher Software gemessenen Werte und einer Signatur des TPM aus. Die Attestation erfolgt gegenüber dem im VAU-Serverschrank integrierten HSM.
Zur Gewährleistung der Wirksamkeit des durch die Remote Attestation gegenüber dem HSM gegebenen Integritätsschutzes für die Laufzeitumgebung der VAU-Server wird der Anbieter des sektoralen IDP die Software für die VAU-Server im Rahmen des technisch Machbaren so gestalten, dass sich VAU-Server nach vollständigem Abschluss des Boot- und Ladevorgangs die Rechte für ein Nachladen von Software selbst entziehen.

8.1.11 HSM

Der Anbieter des sektoralen IDP integriert ein netzwerkfähiges HSM in den VAU-Serverschrank. Das HSM stellt drei systemspezifische Schnittstellen bereit, nämlich:

1. die Schnittstelle zur Remote Attestation von VAU-Servern,
2. eine Schnittstelle zur Nutzung der kryptographischen Identität der VAU für die Terminierung von TLS Verbindungen sowie
3. eine Schnittstelle zur Nutzung der Signaturfunktion auf Basis des privaten Schlüssels der kryptographischen Identität des sektoralen Identity Providers.

Darüber hinaus bietet das HSM eine Management-Schnittstelle zur Einrichtung des HSM im Rahmen einer Zeremonie und zur Einbringung gültiger Referenzwerte für die Attestation der VAU-Server sowie zur Handhabung des privaten Schlüssels der VAU-Identität.
Die Management-Schnittstelle des HSM wird über das Management Interface des VAU-

Serverschranks über Netz verfügbar gemacht.
Das HSM macht die Schnittstellen 2 und 3 nur erfolgreich attestierten VAU-Servern über eine TLS-Verbindung verfügbar.
Der Anbieter des sektoralen IDP wird Prozessen für die Verwaltung des HSM gemeinsam (Mehraugenprinzip, etc.) mit der gematik etablieren.

8.1.12 Datenbank

Der Anbieter des sektoralen IDP stellt ein Datenbanksystem außerhalb der VAU bereit, in das alle verschlüsselten Identitätsdaten gespeichert werden und dass diese Daten über alle Instanzen des sektoralen IDP synchronisiert.
Die Synchronisation von Änderungen muss unmittelbar erfolgen und innerhalb von 500 ms abgeschlossen sein. Eine Spiegelung der Daten mit noch geringerer Latenz ist aufgrund der Architektur des sektoralen IDP nicht erforderlich.
Das Datenbanksystem stellt für die VAU-Server eine REST-Schnittstelle bereit, über die alle benötigten Datenbanktransaktionen abgebildet werden können.
Der Anbieter des sektoralen IDP wird seine Wahl eines Datenbanksystems sowie dessen wesentliche Eigenschaften hinsichtlich Dimensionierung, Synchronisation der Datenbestände und Integration in seine Systems Management Prozesse im Umsetzungskonzept darlegen.

8.1.13 Repository

Der Anbieter des sektoralen IDP stellt ein Repository außerhalb der VAU bereit, aus dem die VAU-Server ihre Boot Images beziehen können. Die Schnittstelle des Repositories richtet sich nach den Anforderungen des Boot Loaders.
Das Repository muss ausreichend geschützt sein, um Ausfälle des sektoralen IDP durch fehlerhafte Boot Images auszuschließen.
Der Anbieter des sektoralen IDP wird sich mit der gematik über einen geeigneten Prozess zur kontrollierten Einbringung signierter Boot Images verständigen. Dort gibt es bereits etablierte Prozesse auch zu Remote Sitzungen für Schlüsselzeremonien und HSM Interaktionen. Der gematik fällt in diesem Prozess die Rolle des Signers der Boot Images zu.

8.2 Containerlösung

Offener Punkt: Eine Umsetzungsempfehlung für eine (containerbasierte) Cloud-Lösung muss noch erarbeitet werden. Die Erarbeitung sollte durch einen gemeinsamen Arbeitskreis von Experten der Krankenkassen, der Anbieter und der gematik unter Führung der gematik erfolgen.
Bei der Erarbeitung von Cloud-Lösungen sollten mindestens diese Punkte Berücksichtigung finden:

- *Sicherheitsanalyse*
- *Betreiberausschluss*

- *Schutzmaßnahmen*
- *OpenSource-Lösungen*
- *Technische Möglichkeiten der (public) Cloud-Anbieter*

2587

2588