Beim vorliegenden Dokument handelt es sich um einen Entwurf der gematik in Vorbereitung auf zukünftige normative Festlegungen als Grundlage entsprechender Zulassungs- und Bestätigungsverfahren. Sowohl einzelne Aspekte, welche als offene Punkte im Dokument kenntlich gemacht worden sind, als auch ergänzende Festlegungen zu einigen Details befinden sich noch in Diskussion. Die gematik reicht diesen Entwurf mit dem Ziel in die Kommentierung, die Umsetzung des Systemdesigns der Telematikinfrastruktur möglichst detailliert zu ergänzen und ein Verständnis der weiteren Dokumente des Release 4.0 zu ermöglichen. Es sollen explizit bereits frühzeitig Fragen und Anmerkungen aufgenommen und diskutiert werden können.

Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation
Identity Provider - Dienst

Version:	1.0.0 CC
Revision:	230722
Stand:	30.04.2020
Status:	zur Abstimmung freigegeben
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemSpec_IDP_Dienst

Dokumentinformationen

Änderungen zur Vorversion

Es handelt sich um die Erstversion des Dokumentes.

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0 CC	30.04.20		initiale Erstellung des Dokuments	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die Lösung zielt zunächst auf die Einführung eines IdP als neuen Dienst der TI und ermöglicht die Entwicklung von Anwendungen basierend auf OpenID connect. Der IdP-Dienst basiert auf der verwendung der bereits wird von der gematik zunächst für alle Nutzer der Fachanwendung Rezept zur Verfügung gestellt und bietet dieser Anwendung damit unabhängig vom Aufbau weiterer OpenID connect basierter Identity Providern die Möglichkeit alle Teilnehmer der TI anhand ihrer bereits etablierten Identitätsmerkmale zu Identifizieren und zu Authentisieren. Der IdP-Dienst ist als eine erste möglichst breite Ausbaustufe hin zu einer Lösung mit verteilten Identity Providern anzusehen.

Für kommende Releases ist daher vorgesehen ein flexibleres Identity Management vorzusehen. Dieses wird es Identitätsherausgebern (z.B. Krankenkassen, LEO, …) ermöglichen als Anbieter eigene IdPs mit flexibler Identitätenverwaltung in die TI einzubringen die den IDP-Dienst für die von ihnen verwalteten Identitäten ersetzen. Die Anbieter können dabei alternative Authentisierungslösungen anbieten, sofern diese sicher genug sind.

Der Standard OpenID connect und darauf beruhende Produkte im Markt bieten zusätzliche Funktionen an, die perspektivisch interessant sind. Dies betrifft z.B. die Integration SAML2-basierter Anwendungen und den Austausch von Identitäten mit externen (föderierten) IdPs für ein sektorenübergreifendes oder EU-weites Identity Management.

1.2 Zielgruppe

1.3 Geltungsbereich

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

Schutzrechts-/Patentrechtshinweis

Die nachfolgende Spezifikation ist von der gematik allein unter technischen Gesichtspunkten erstellt worden. Im Einzelfall kann nicht ausgeschlossen werden, dass die Implementierung der Spezifikation in technische Schutzrechte Dritter eingreift. Es ist allein Sache des Anbieters oder Herstellers, durch geeignete Maßnahmen dafür Sorge zu tragen, dass von ihm aufgrund der Spezifikation angebotene Produkte und/oder Leistungen nicht gegen Schutzrechte Dritter verstoßen und sich ggf. die

erforderlichen Erlaubnisse/Lizenzen von den betroffenen Schutzrechtsinhabern einzuholen. Die gematik GmbH übernimmt insofern keinerlei Gewährleistungen.

1.4 Abgrenzungen

Spezifiziert werden in dem Dokument die von dem Produkttyp bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt.

Der in diesem Dokument beschriebene IdP-Dienst bietet die Basis für eine IdP-gestützte Identifikationslösung, um unterschiedlichen Nutzergruppen einen ihrer Rolle entsprechenden Zugriff auf Dienste der Provider-Zone der TI zu ermöglichen. Es werden die Schnittstellen beschrieben, die dieser zu bieten hat und anhand welcher Rahmenbedingungen diese umzusetzen sind. Grundsätzlich sind alle angebotenen Leistungen anhand der im Abschnitt 4.1- Übergreifende Festlegungen aufgeführten RFC (Request for Comments) und natürlich den daraus hervorgehenden BCP (Best Current Practice) umzusetzen. Als Umsetzungsleitlinie sind [OpenID Connect Core 1.0] mit der Erweiterung [HEART I] & [HEART II] (siehe6.5.2- Weitere Dokumente) heranzuziehen.

Teile der IdP-gestützten Identifikation der Nutzer sind ebenfalls die Dokumente [gemSpec_IDP_FD], welches die Dienste-Anbindung von Fachdiensten an den IdP-Dienst beschreibt, sowie [gemSpec_IDP_Frontend] für Endgeräte der Nutzer. Nutzer sind Leistungserbringer inklusive ihrer Institutionen ebenso wie Versicherte und nicht zuletzt Fachdienste, die ihre gesicherten Ressourcen auf diesem Wege bereitstellen.

Der mit diesen Dokumenten in Gänze beschriebene IdP-Dienst stellt eine Basisleistung der TI dar, um die durch unterschiedliche TSP herausgegebenen Identitäten zentralisiert auf deren Gültigkeit und Nutzungsberechtigung prüfen zu können und anhand des Prüfungsresultates entsprechende "`ID_TOKEN`" auszustellen bzw. die Ausstellung zu untersagen.

Die vollständige Anforderungslage für den Produkttyp ergibt sich aus weiteren Konzept- und Spezifikationsdokumenten; diese sind in dem Produkttypsteckbrief des Produkttyps IdP-Dienst verzeichnet.

Nicht Bestandteil des vorliegenden Dokumentes sind die Verfahrensschritte zur Erstellung des notwendigen Schlüsselmaterials. Es wird angenommen, dass Fachdienste ihre innerhalb der TI zu verwendenden Zertifikate für die TLS-Sicherung, über zentrale Plattformdienste der TI beziehen und diese dort auch geprüft werden können.

Ebenso wird angenommen, dass verwendete Hard- und Software geeignet ist, um eigenes asymmetrisches Schlüsselmaterial für die Ver- und Entschlüsselung sowie Signatur gemäß den verwendeten Standards erzeugen und dieses verwalten kann. Es werden keine Vorgaben dahingehend gemacht, wie die einzelnen Akteure zu dem für sie notwendigen Schlüsselmaterial gelangen. Jedoch sind normative Vorgaben bezüglich Regelmäßigkeit und Lebenszyklen (Quantität) sowie Stärke und Algorithmen (Qualität) des Schlüsselmaterials enthalten.

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.

1.5.1 Hinweis auf offene Punkte

Offene Punkten werden im Dokument in dieser Darstellung ausgewiesen.

2 Systemüberblick

Im aktuellen Stand des Dokumentes fehlen Festlegungen zur Integration von Primärsystemen in der Rolle der Authenticator Applikation

Im aktuellen Stand des Dokumentes fehlen, noch in Diskussion befindliche, Festlegungen zur Erweiterung des Integritätsschutz des Discovery Documents sowie weiterer verwendeter Schlüssel. Die Standards sehen Verschlüsselungen vor aber lassen die Methoden des Schlüsselaustausch offen und die gematik ist noch in Abstimmung zu praktikablen Methoden.

Im aktuellen Stand des Dokumentes fehlen an einigen Punkten noch Verweise auf die zugrundeliegenden Operationen der Standards OpenID Connect und OAuth2.

Die vorliegende Spezifikation definiert die Anforderungen zu Herstellung, Test und Betrieb des Produkttyps IdP-Dienst, welcher selbst Teil der IdP-gestützten Nutzerauthentifizierung zur Verwendung von openID Connect mit OAuth 2.0 ist.

Dabei wird insoweit vom Standard abgewichen das der Resource Owner (Fachdienst) die Identifikation nicht durch einen redirect auf den Authorization Server (IdP-Dienst) überträgt, sondern der Aufruf der Protected Resource erst statt findet, nachdem die Identifikation erfolgt ist. Dies vereint die im Protokollfluss des Standard in Abbildung 1 [RFC6749 # section-1.2] dargestellten "Resource Owner" und "Authorization Server" zu einem Dienst.

Der IdP-Dienst führt also nicht nur die Identifikation des Nutzers durch, sondern stattet diesen auch selbst mit deinem ID_TOKEN gemäß [RFC6749 # section-1.4] aus. Gewählt wird aus Sicherheitsaspekten der "Authorization Code Grant" gemäß [RFC6749 # section-4.1] wobei daran erinnert sei, dass die Anfrage nicht per redirect organisiert wird, sondern der Authenticator die Rolle des User-Agent übernimmt. Die Verwendung eines (mobilen) Webbrowsers ist aufgrund der Nutzung von SmartCards als Authentisierungsmedium nicht umsetzbar.

Der IdP-Dienst teilt sich auf in mehrere Teildienste, welche auf unterschiedlicher Hardware betrieben werden können und im Falle des Authenticators sind. Die Teildienste, welche vom IdP-Dienst als zentrale Plattformleistung innerhalb der TI erbracht werden, besitzen dort eine statische IP-Adressierung und somit statische URI. Diese statisch adressierten Teildienste umfassen:

- Discovery-Endpunkt ("OAuth 2.0 Authorization Server Metadata" [RFC8414])
- Authorization-Endpunkt (Teil des "The OAuth 2.0 Authorization Framework" [RFC6749])
- Token-Endpunkt [RFC6749 # section-3.2]
  Teildienst 1 ID_TOKEN [RFC6749 # section-1.4 & RFC6749 # section-5]
  Teildienst 2 REFRESH_TOKEN [RFC6749 # section-1.5 & RFC6749 # section-6]
- Token Introspection-Endpunkt [RFC7662 # section-2]
- Revocation-Endpunkt [RFC7009 # section-2]
- User Info-Endpunkt [OpenID Connect Core v1.0]

Der einzig nicht zentral betriebene Teildienst des IdP-Dienstes ist der sogenannte Authenticator, welcher auf jedem einzelnen Endgerät der Nutzer für nur dieses zuständig ist und dort für mehrere Anwendungsfrontends zeitgleich als Authenticator seinen Dienst tun kann.

Die dynamisch adressierten Teildienste des IdP-Dienstes umfassen:

- Authenticator Applikation (der in [RFC6749 # section-4.1] beschriebene User-Agent)

Im folgenden Schaubild sind die vom IdP-Dienst bereitgestellten Teildienste mit blau hinterlegt.

Teildienste wie Authenticator und Anwendungsfrontend befinden sich in dem mit "Gerät des Nutzers" bezeichneten Bereich.

Fachdienste sind nicht näher bestimmt und befinden sich im Block unterhalb des Identity Providers.

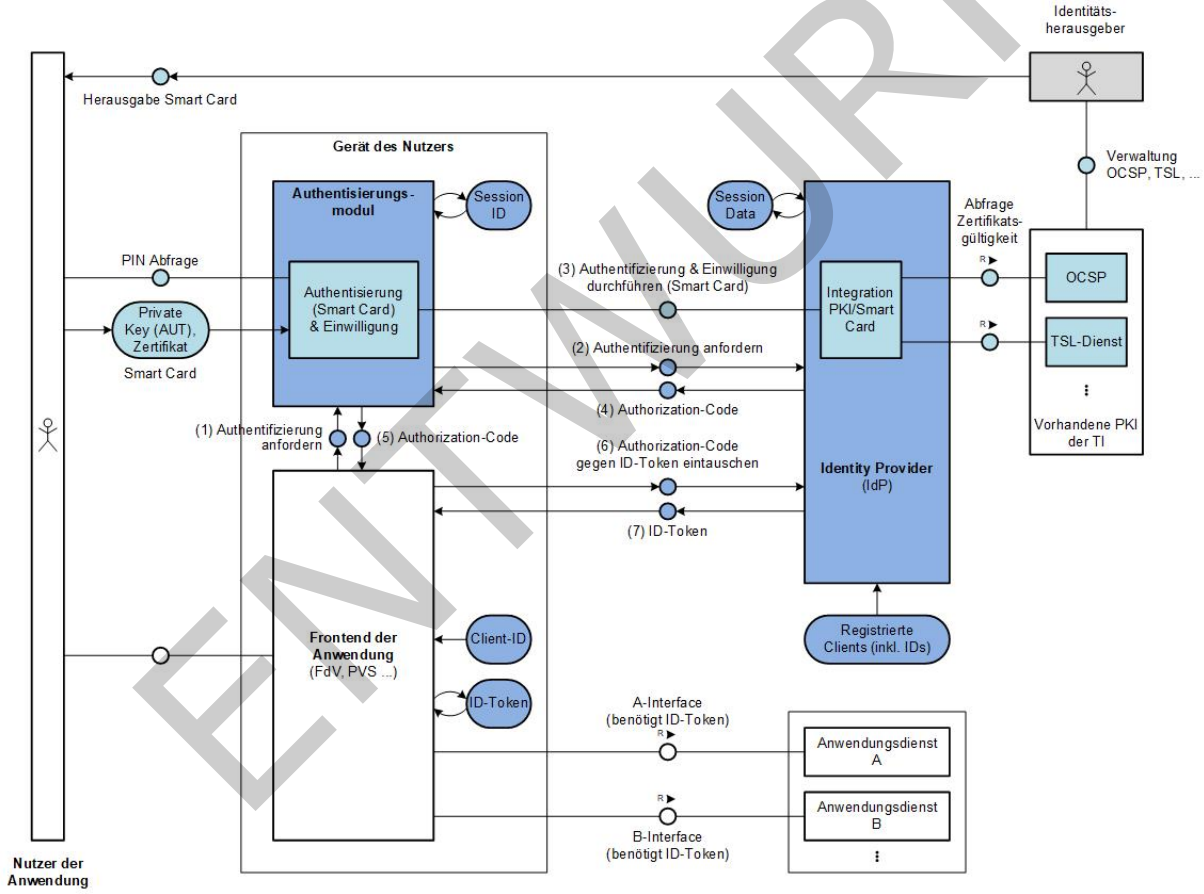

Abbildung 1 Übersichtsschaubild OAuth2.0 Smartcard-IdP-Dienst

Die gematik versucht alle technisch möglichen Maßnahmen umzusetzen, um die bekannten [RFC6749 # section-10], aber auch unbekannten sicherheitskritischen Aspekte zu betrachten, um einen maximalen Schutz für die in höchstem Maße schützenswerten Informationen der Versicherten sowie der Leistungserbringer und ihrer Institutionen zu gewährleisten. Der hier gezeigte Smartcard-IdP-Dienst stellt eine Basisleistung innerhalb der TI dar und soll die sichere Identifikation der Akteure anhand der ihnen bereitgestellten Identifikationsmittel (Smartcards) ermöglichen. Der Standard

lässt hierbei die Einbringung weiterer Identity Provider für unterschiedlichste Identifikationsverfahren zu ohne, dass Fachdienste hierfür eine grundlegende Änderung der Zugangsmechanismen erfahren müssen.

Aus diesem Grund werden von den in den jeweiligen Standards angeführten Optionen immer diejenigen gewählt, die das Maß an Sicherheit verbessern können. Wird angeboten, dass etwas vor dem Transport verschlüsselt werden kann, ist diese Maßnahme zwingend umzusetzen. Wird an anderer Stelle eine Umsetzungsempfehlung ausgesprochen, um das Maß der Sicherheit zu erhöhen, so wird diese als zwingende Maßnahme vorgesehen.

Generell findet die Umsetzung der in den Standards angebotenen optionalen Möglichkeiten gemäß [OpenID Connect] unterstützt durch [OpenID Connect Core v1.0], [OpenID Connect Discovery v1.0], [OpenID Connect Registration v1.0] statt. Um den höchsten Anforderungen an Datenschutz und Datensicherheit gerecht zu werden, kommen weitere Sicherheitsaspekte zum Tragen, welche in "HEART I" [OpenID Heart - OpenID Connect v1.0] und "HEART II" [OpenID Heart - OAuth 2 v1.0] beschrieben sind bzw. gefordert werden.

Diese selbst basieren wiederum auf zahlreichen anderen Standards, wovon hier nur die erste Ebene genannt sein soll:
Request for Comments JWT (JSON Web Token) [RFC7519], JWS (JSON Web Signature) [RFC7515], JWE (JSON Web Encryption) [RFC7516], JWK (JSON Web Key) [RFC7517], JWA (JSON Web Algorithm) [RFC7518] und WebFinger
sowie OAuth 2.0 Bearer, OAuth 2.0 Assertion, OAuth 2.0 JWT Profile, OAuth 2.0 Responses.

Die Gesamtliste der referenzierten Standards finden sie im Abschnitt 6.5.2- Weitere Dokumente.

3 Systemkontext

Anwendungsfrontend der Versicherten (Resource Owner) wollen auf Fachdienste (Relying Party [https://openid.net/specs/openid-connect-token-bound-authentication-1_0-03.html]) zugreifen. Der Authenticator ist eine Anwendung auf dem Endgerät des Nutzers, welche sich eindeutig gegenüber dem IdP-Dienst registriert und zukünftig von diesem bei Berechtigungsfreigaben, ähnlich einer Zweifaktor-Authentifizierung, in die Prozesse mit eingebunden wird. Der notwendige Authentisierungsprozess wird vom IdP-Dienst (Resource Server) angeboten, kann jedoch vom oben genannten Anwendungsfrontend nicht direkt, sondern nur über den Authenticator angesprochen werden. Anwendungsfrontend und IdP-Dienst treten daher mit allen anderen Akteuren in Kontakt, Fachdienste und Authenticator jedoch nur mit IdP-Dienst und Anwendungsfrontend. Eine Verbindung zwischen Authenticator und Fachdienst findet niemals statt.

3.1 Verfahrensbeschreibung

Hinweis: Alle Akteure (Nutzer, Fachdienste ebenso wie der IdP-Dienst selbst) sind beim Authorization Server mit all ihren tokenbasierten Schnittstellen registriert und haben das Doscovery Document zur Kenntnis genommen und die für sie relevanten URI und PUK der Gegenstellen im Zugriff. Daher kennen alle Akteure auch die URI der Gegenstellen. Alle Vorgänge werden jeweils mit dem publicKey der endgültigen Gegenstelle verschlüsselt.Der "ACCESS_CODE" wird für das Anwendungsfrontend mit dessen "PUK_FRONT" und "ID_TOKEN" für den jeweiligen Fachdienst mit dessen "PUK_FD" verschlüsselt. Alle Instanzen adressieren sich gegenseitig über deren registrierte URI. Ist einer Instanz die URI der Gegenstelle nicht bekannt, liegt entweder ein Fehler in der Registrierung vor und diese ist zu wiederholen oder das Discovery Document des IdP-Dienstes wurde nicht regelkonform ausgewertet.

3.2 Registrierung Authenticator und Anwendungsfrontend

Um ein Anwendungsfrontend nutzen zu können, muss dieses mit einem Authenticator verbunden und beide (Authenticator und Anwendungsfrontend) müssen beim IdP-Dienst registriert sein.

Die Registrierung von Authenticators und Anwendungsfrontend ist im Dokument [gemSpec_IDP_Frontend] beschrieben. Beim IdP-Dienst ist eine eindeutige Verknüpfung des registrierten Authenticators mit dem auf dem Anwendungsfrontend ausgewählten Profil gespeichert.

Startet der Nutzer sein Endgerät, auf welchem der Authenticator installiert ist und das Gerät ist online, verbindet sich der Authenticator automatisch mit dem Authorization Server und meldet dort die URI, unter welcher er aktuell erreichbar ist und gegebenenfalls einen neuen öffentlichen Schlüssel "PUK_MOD".

Der IdP-Dienst muss die vom Endgerät des Nutzers aus mit dem Authorization Server interagierende Authenticator Applikation stellen und in den jeweiligen Stores für die Betriebssysteme Android (Google Play Store) und Apple (Apple App Store) registrieren und zum kostenlosen Download anbieten.

3.3 Anwendungsfrontend vorbereitende Maßnahmen

Das Anwendungsfrontend muss vor dem Aufruf des Authenticators ein Schlüsselpaar bestehend aus öffentlichem und privaten Schlüssel gemäß [gemSpec_Krypt] erzeugen. Außerdem muss das Anwendungsfrontend ein SECRET (Zufallswert mit mindestens 128 Bit Güte) und hierüber einen Hash mit einem Algorithmus gemäß [gemSpec_Krypt] erzeugen.

3.4 Beschaffung des ID_TOKEN

Der Nutzer ruft sein Anwendungsfrontend auf, wählt sein im Endgerät gespeichertes Profil aus, und gibt seine Zugangsdaten in das Anwendungsfrontend ein. Danach baut das Anwendungsfrontend eine Verbindung zu der im Programm hinterlegten URI des IdP-Dienstes auf, um sich von dort das aktuelle Discovery Document herunterzuladen.

Anhand des Discovery Documents erfährt das Anwendungsfrontend die URI aller Dienste (Authorization-, Token- und Token-Revocation Endpunkte sowie diejenigen der angebotenen Fachdienste) und wo diese jeweils ihre öffentlichen Schlüssel hinterlegt haben. Über den Authorization Server erfährt es den zuletzt durch den Authenticator dort hinterlegten öffentlichen Schlüssel "PUK_MOD". Da die Verknüpfung des Anwendungsfrontend mit seinem Authenticator bereits beim IdP-Dienst registriert ist, kann es seine Anfrage für ein "ID_TOKEN" über den IdP-Dienst an seinen Authenticator schicken.

Inhalt der mit dem "PUK_MOD" verschlüsselten Anfrage ist:

- die eigene URI sowie Bezeichnung des aufzurufenden Fachdienstes,
- die eigene Programmbezeichnung mit Versionsnummer,
- der über das eigene SECRET gebildete HASH mit Angabe des Algorithmus,
- der eigene öffentliche Teil des Schlüssels "PUK_FRONT".

Der Authorization-Endpunkt leitet die verschlüsselte Token-Anfrage an die zuletzt eingetragene URI des Authenticators weiter.

Ist der Authenticator offline (nicht erreichbar), wird der Nutzer darauf hingewiesen, dass der Authenticator gestartet werden muss, um dort den Request freizugeben. Wird der Authenticator gestartet und ist online, kann man darauf den Request bestätigen.

3.5 Bereitstellung des Authenticators

Der Betreiber des IdP-Dienstes stellt den Authenticator über die für das jeweilige Betriebssystem üblichen Verteilungsmechanismen bereit.

3.6 Aufgaben des Authenticators

Der Authenticator entschlüsselt die "ID_TOKEN"-Anfrage des mit ihm verknüpften Anwendungsfrontend und prüft diese.

3.6.1 Bei noch bestehender Subject Session

Besteht eine aktive Subject Session, versucht der Authenticator am Authorization-Endpunkt, das "ID_TOKEN" für das Anwendungsfrontend zu beantragen. Die Beantragung erfolgt ebenfalls verschlüsselt mit dem "PUK_AUTH", welchen der Authenticator auf Basis des Discovery Document ermittelt hat.

3.6.2 Bei fehlender oder ungültiger Subject Session

Bei einer fehlenden oder abgelaufenen Subject Session erfordert der Authorization-Endpunkt, dass der Authenticator sich erneut authentisiert. Der Nutzer wird aufgefordert, seine Smartcard (eHBA oder eGK) mit dem Authenticator zu verbinden, woraufhin dieser die Registrierung durchführt und eine neue Subject Session etabliert wird.

3.7 Aufgaben des Authorization-Endpunktes

Der Authorization-Endpunkt nimmt die Anfrage und entschlüsselt diese mit seinem privaten Schlüssel "PRK_AUTH". Nach der Signatur- und Integritätsprüfung prüft der Authorization-Endpunkt, ob die mit der ID_TOKEN-Anfrage eingereichten mit den im Claim des Fachdienstes geforderten Attributen bedient werden können.

3.7.1 Unzureichende Attribute für das Claim

Kann das Claim nicht voll bedient werden, fordert der Authorization-Endpunkt zur erneuten Authentisierung und Freigabe der erforderlichen Attribute auf.

3.7.2 Erstellung des ACCESS_CODE

Sind alle im Claim geforderten Attribute vorhanden und deren aktuelle Gültigkeit geprüft, erstellt der Authorization-Endpunkt einen ACCESS_CODE und verschlüsselt diesen für das Anwendungsfrontend. Der Authorization Server veranlasst die Erstellung des "ID_TOKEN" und – wenn dies im Claim des Fachdienstes vorgesehen ist – zusätzlich des REFRESH_TOKEN.

Den ACCESS_CODE sendet der Authorization-Endpunkt an die mit der Antragsstellung mitgeteilte URI des Anwendungsfrontend "URI_FRONT" mit dessen öffentlichen Schlüssel verschlüsselt zurück.

3.8 Einreichen des ACCESS_CODE

Das Anwendungsfrontend verschlüsselt den ACCESS_CODE zusammen mit dem von ihm im <<Verweisziel>> erzeugten SECRET, mit dem öffentlichen Schlüssel des Token-Endpunktes "PUK_TOKEN" und reicht diesen dann beim Token-Endpunkt ein.

3.9 Aufgabe des Token-Endpunktes

Der Token-Endpunkt nimmt die verschlüsselten Daten des Anwendungsfrontend entgegen und prüft neben deren Integrität, ob das eingereichte SECRET bei Nutzung des identischen Hash-Verfahrens zum bitgleichen Hash-Wert führt. Stimmen die beiden Hash-Werte aus dem initialen Aufruf (siehe <<Verweisziel>> und dem nun gebildeten Hash-Wert überein, ist sichergestellt, dass Aufrufer und Initiator identisch sind. Der Token-Endpunkt gibt daraufhin das"`ID_TOKEN`" mit dem öffentlichen Schlüssel des Fachdienstes "`PUK_FD`" verschlüsselt an das Anwendungsfrontend heraus.

3.10 Einreichen des "ID_TOKEN" beim Fachdienst

Um schlussendlich Zugriff auf den Fachdienst zu bekommen, reicht das Anwendungsfrontend das verschlüsselte "`ID_TOKEN`" beim Fachdienst ein.

3.11 Aufgabe des Fachdienstes

Der Fachdienst nimmt das verschlüsselte "`ID_TOKEN`" entgegen und entschlüsselt es mit seinem privaten Schlüssel "PRK_FD". Danach überprüft es die Integrität und die Übereinstimmung mit dem eigenen Claim. Enthält das "`ID_TOKEN`" mehr oder weniger Attribute, als im Claim vereinbart, oder sind diese fehlerhaft oder nicht befüllt, stimmt die Integrität oder Signatur des "`ID_TOKEN`" nicht oder ist das "`ID_TOKEN`" zeitlich nicht mehr gültig, bricht der Fachdienst die Kommunikation mit einer Fehlermeldung ab.

Anderenfalls gewährt der Fachdienst dem Anwendungsfrontend Zugriff gemäß eigener Spezifikation.

3.12 Akteure und Rollen

3.13 Akteure

Resource Owner (Nutzer)

Der Resource Owner ist der Nutzer, dem die gesicherten Ressourcen (Protected Resource) auf dem gesicherten Server (Protected Server) zugeordnet sind**.** Im Falle der TI ist der Resource Owner der Versicherte, Arzt oder die Leistungserbringer-Institution.

~~Eine Entität (Fachdienstanbieter), die einem Dritten den Zugriff auf ihre geschützten Ressourcen (Fachdienste) gewähren kann. Diese Ressourcen werden durch den Resource Server bereitgestellt. Im Falle der Telematikinfrastruktur sind die darin angebotenen Dienste zwar in der Hoheit der Fachdiensteanbieter, werden jedoch allen berechtigten angeboten.~~

Der Nutzer (Client im herkömmlichen Sinne) ist der Resource Owner, also derjenige, welcher auf die durch die Protected Resource bereitgestellten Informationen Zugriff bekommt.

Resource Server [rfc7165#section-5.2] (Technische Einrichtung zum Bereitstellen der Protected Resource)

Der Resource Server wird im Kontext der Telematik Infrastruktur durch den Fachdienst dargestellt, welcher die geschützten Informationen (Protected Resource) dem Nutzer (Resource Owner) auf der sicheren Umgebung (Protected Server) bereitstellt.

~~Der Fachdienstanbieter, auf dem die geschützten Ressourcen (Protected Resources) liegen [PP1] [TJ2]. Er ist in der Lage, auf Basis von Access Tokens darauf Zugriff zu gewähren. Ein solcher Token repräsentiert die delegierte Autorisierung des Resource Owners.~~

Client

~~Der Client wird durch das Anwendungsfrontend (Relying Party) des Nutzers dargestellt, die auf geschützte Ressourcen (Fachdienste) des Resource Owners zugreifen möchte. Das Anwendungsfrontend kann auf einem Server als Webanwendung, auf einem Desktop PC oder mobilen Gerät (z.B. Smartphone) ausgeführt werden.~~

Der Client ist derjenige, der den IdP-Dienst bzw. die von ihm bereitgestellten Teildienste (Endpunkte) zur Bereitstellung seiner eigenen schützenswerten Daten auf dem Resource Server verwendet.

4 Zerlegung des Produkttyps

Detailiertere Festlegungen analog zu gemSpec_FD_eRp#5.6.4 Sicherheit der Netzübergänge sind noch offen.

Festlegungen für eine Ergänzung des Integritätsschutz des Discovery Dokumentes mittels TI-PKI sind noch in Diskussion

4.1 Übergreifende Festlegungen

Rollenausschluss

Der Anbieter des IdP-Dienstes muss sicherstellen, dass durch Vertreterregelungen oder Krankheitsausfälle keine Verletzung der vorgesehenen Rollentrennung erfolgen kann. Insbesondere ist es den Mitarbeitern des IdP-Dienstes untersagt gleichfalls Entwickler eines Anwendungsfrontend zu sein oder zukünftig zu werden. Ebenso ist es Mitarbeitern und Führungskräften des Anbieters für den IdP-Dienst untersagt für andere Anbieter von Diensten für die Telematik Infrastruktur tätig zu sein oder zu werden.

Zugriff durch Mitarbeiter

Der Anbieter IdP-Dienst stellt Identitätsnachweise aus, mit deren Hilfe es möglich ist, auf vertrauliche insbesondere persönliche Informationen zuzugreifen. Der Anbieter des IdP-Dienstes muss darum alles in seiner Macht stehende unternehmen um die Zuverlässigkeit, Integrität, Vertrauenswürdigkeit seiner Mitarbeiter sicher zu stellen. Gegebenenfalls soll auch eine Sicherheitsüberprüfung gemäß SÜG durchzuführen. Die in den Fachdiensten durch die "`ID_TOKEN`" erreichbaren Daten sind in höchstem Maße schützenswert und unterliegen strengster Geheimhaltung.

Dokumentationspflicht

Der Anbieter des IdP-Dienstes hat eine besondere Dokumentationspflicht. So muss jeglicher Zugang bzw. Zugriff auf Systeme die im direkten Zusammenhang mit dem IdP-Dienst stehen dokumentieren. Hiermit sind insbesondere administrative Zugriffe auf die Systeme gemeint, welche ausschließlich im mindestens "Vier-Augen-Prinzip" zu erfolgen haben.

Service Lokalisierung

Die Lokalisierung des vom IdP-Dienst angebotenen Service erfolgt ausschließlich über die Bekanntmachung im Fachportal der gematik GmbH.

Verwendete Standards

Soweit nicht explizit etwas anderes beschrieben ist, verhalten sich alle Schnittstellen des IdP-Dienstes und der daran angeschlossenen ortsveränderliche Komponenten (Endgerät des Nutzers mit Authenticator und Anwendungsfrontend) beschrieben in [gemSpec_IDP_Frontend] sowie Fachdienste als Teil der zentralen TI Providerzone beschrieben in [gemSpec_IDP_FD] nach den unten aufgelisteten Standards. Verschärft durch die für Applikationen im Gesundheitswesen vorgesehene Erweiterung HEART, in welcher eine erweiterte Schärfung der Vorgaben aus den Standards beschrieben ist.

Die verwendeten Standards sind:

- RFC3986 (URI)
- RFC7009 (JSON REVOCATION)
- RFC7165 (JOSE)
- RFC7231 (HTTP)
- RFC7515 (JWS JSON SIGNATURE)
- RFC7516 (JWE JSON ENCRYPTION)
- RFC7517 (JWK JSON KEY)
- RFC7518 (JWE JSON ALGORITHM)
- RFC7519 (JWT JSON WEB TOKEN)
- RFC7520 (JOSE Protection)
- RFC7521 (Assertion Authorization)
- RFC7522 (Assertion SAML 2.0)
- RFC7523 (JSON TOKEN Profile)
- RFC6749 (Oauth2)
- RFC7591 (OAuth2 Dynamic Client Registration)
- RFC6750 (Oauth2 Bearer)
- RFC7636 (Oauth Proof Key for Public Client)
- RFC7662 (OAuth TOKEN INTROSPECTION)
- RFC8417 (Security Event Token)

A_19881 - Gültigkeitsdauer von JSON-Schlüsselmaterial
Das von Servern zu verwendende Schlüsselmaterial ist maximal 48 Stunden alt und KANN von der Komponenten-PKI ausgegeben werden.
Server MÜSSEN zwei Schlüsselgenerationen gleichzeitig bedienen, wobei die Erneuerung jeweils alle 24 Stunden erfolgen MUSS. Der ältere Schlüssel ist somit <24 Stunden und der jüngere Schlüssel <48 Stunden gültig.**[<=]**

A_19870 - Verwendung eindeutiger URI [RFC7800#section-1]
Der Anbieter IdP-Dienst MUSS alle verwendeten Adressen in Form von URI gemäß [RFC3986] angeben und in einem Discovery Document innerhalb der TI und im Internet veröffentlichen. **[<=]**

A_19871 - Bekanntgabe des Downloadpunktes im Fachportal der gematik
Der Downloadpunkt des Discovery DocumentMUSS im Fachportal der gematik veröffentlicht sein. **[<=]**

A_19872 - Discovery Document interne und externe Adressierung
Das Discovery Document MUSS sowohl innerhalb der TI als auch im Internet mit voneinander abweichender Adressierung veröffentlicht werden.
Das Discovery Document innerhalb der TI adressiert hierbei die URI der Fachdienste und Schnittstellen des IdP-Dienstes innerhalb der Telematikinfrastruktur. Das im Internet bereitgestellte Discovery Document stellt die URI der angebotenen Fachdienste im Internet mit dort auflösbaren Adressen oder den statischen IP-Adressen bereit. **[<=]**

A_19873 - Inhalte des Discovery Documents
Achtung: Es gibt je ein internes und externes (public) "Discovery Document". Diese unterscheiden sich in den darin angebotenen URI welche gleichlautend im Host-Anteil auf unterschiedliche Domänen bzw. TLD (Top-Level-Domain) verweisen.

Das Discovery Document gemäß [RFC8414 # section-2] MUSS mindestens folgende Attribute als URI beinhalten:
iss (Hier ist der IdP-Dienst erreichbar)
jwks_uri (für Abruf der/des PUK des Authorization Server [RFC7517])
URI_DD (URI, unter welcher das Discovery Document bereitgestellt ist)
URI_AUTH & PUK_URI_AUTH URI des Dienstes und öffentlichen Schlüssels des Authorization-Endpunktes gemäß [RFC6749]
URI_TOKEN & PUK_URI_TOKEN URI des Dienstes und öffentlichen Schlüssels des Token-Endpunktes gemäß [RFC6749]
URI_INT & PUK_URI_INT URI des Dienstes und öffentlichen Schlüssels des Introspection-Endpunktes gemäß [RFC7662]
URI_REV & PUK_URI_REV URI des Dienstes und öffentlichen Schlüssels des Revocation-Endpunktes gemäß [RFC7009]
URI_INFO & PUK_URI_INFO URI des Dienstes und öffentlichen Schlüssels des Userinfo-Endpunktes

Hinweis: Die vom Authorization Server angebotenen SchnittstellenSOLLEN auf unterschiedlichen physischen Schnittstellen erreichbar sein. In jedem Fall MUSS der IdP-Dienst die URI im Discovery Document für jede einzelne Schnittstelle eintragen, um ein Hardware-Splitting (3-Tier-Lösung) zu ermöglichen.**[<=]**

Hinweis:

"URI_MOD": URI der Authenticator APP; wird dynamisch registriert und kann daher nicht im Discovery Document stehen. Diese ist eineindeutig mit der "SUBJECT_SESSION" verbunden.

"URI_FRONT": Ist die URI des Anwendungsfrontend. Sie wird dynamisch registriert und kann daher nicht im Discovery Document stehen.

Die Anwendungs-Session ist eindeutig mit dem Anwendungsfrontend des Nutzers über dessen "URI_FRONT" verbunden und kann durch die "SUBJECT_SESSION" identifiziert werden.

A_19874 - Bereitstellung Internes Discovery Documents innerhalb der TI
Der IdP-Dienst MUSS das internal Discovery Document, immer nach Änderungen und nach Schlüsselwechseln mit seinem privaten Schlüssel "PRK_DD" gemäß [ML-7887 - Kryptographische Algorithmen für XML-Dokumente] signiert, an einem spezifischen Downloadpunkt TLS-gesichert innerhalb der TI bereitstellen.**[<=]**

A_19875 - Absicherung des Internen Discovery Document innerhalb der TI mit TLS
Um eine TI-interne Prüfung des TLS-Zertifikates zu ermöglichen MUSS hier ein Zertifikat der Komponenten-PKI vom Type "C.FD.TLS-S" gemäß [ML-7034 - C.FD.TLS-S Server-Authentisierung (ehemals C.SF.SSL-S)] mit der technischen Rolle "oid_idpd" verwendet werden. **[<=]**

A_19876 - Internes Discovery Document - Prüfung vor Veröffentlichung
Der IdP-Dienst **MUSS** alle von ihm im angebotenen URI und URI anderer Dienste insbesondere Fachdienste vor deren Veröffentlichung im internal Discovery Document auf bloße Erreichbarkeit prüfen. **[<=]**

A_19877 - Bereitstellung Externes Discovery Document im Internet

Der IdP-Dienst stellt das public Discovery Document immer nach Änderungen und nach einem Schlüsselwechsel mit einer spezifischen URI TLS-gesichert im Internet zum Download bereit. Das public Discovery Document MUSS mit seinem privaten Schlüssel des Discovery-Endpunktes "PRK_DD" gemäß [ML-7887 - Kryptographische Algorithmen für XML-Dokumente] signiert sein.
[<=]

A_19878 - Absicherung des Externen Discovery Document im Internet mit TLS
Der IdP-Dienst MUSS, für die HTTPS-Schnittstellen im Internet, Extended Validation TLS-Zertifikate eines Herausgebers gemäß [CAB-Forum] verwenden. **[<=]**

A_19879 - Externes Discovery Document - Prüfung vor Veröffentlichung
Der IdP-Dienst MUSS alle von ihm angebotenen URI betreiben und URI anderer Dienste insbesondere Fachdienste vor deren Veröffentlichung im public Discovery Document auf bloße Erreichbarkeit prüfen.
[<=]

A_19880 - Bereitstellung der PUK
(Zuweisung IdP-Dienst, Fachdienste, Frontend)
Zwecks Überprüfung der vom Authorization Server vorgenommenen Signaturen MÜSSEN alle vom Authorization Server verwendeten PUK's zum Download bereitstehen. Ergänzend zum Standard MUSS der Downloadpunkt der PUK's im Discovery Document (DD) in Form von URI's eingetragen sein.
[<=]

A_19895 - Erweiterte Nutzung von Schlüsseln
(Zuweisung IdP-Dienst)
Der Authorization Server MUSS die einzelnen Schnittstellen (AUTH, DISC, TOKEN, INT, REV, INFO) mit getrennten Interfaces bedienen. Die Verwendung des identischen Schlüsselmaterials für mehrere Schnittstellen ist nicht zulässig **[<=]**

4.2 Fehlermeldungen

Entstehen während eines Verarbeitungsprozesses Fehler, muss der jeweilige Dienst-teil diese sofort protokollieren und melden. Es ist nicht ausreichend, aufgetretene Fehler nur sporadisch zu melden, da bei auftretenden Fehlfunktionen bei täglich bis zu 4 Millionen eRezepten in Spitzenzeiten mehrere hunderttausend pro Stunde anfallen. Es ist unwahrscheinlich, aber theoretisch möglich, dass alle Leistungserbringer quasi zeitgleich eRezepte einstellen, wodurch innerhalb von nur einer Sekunde über 160.000 eRezepte eingestellt würden. Ebenso ist davon auszugehen, dass die Leistungserbringer täglich in großer Zahl relativ zeitgleich ihre Tätigkeit aufnehmen und so binnen weniger Minuten mehrere tausend "SUBJECT_SESSION" zu etablieren sind.

Aus diesem Grund muss ein flexibles, mit einem Minimum an übertragenen Daten agierendes Frühwarnsystem Schwächen im Gesamtkonstrukt erkennbar machen und mit Fehlermeldungen in Echtzeit reagieren.

A_19896 - Format der Fehlermeldungen
(Zuweisung IdP-Dienst)
Der IdP-Dienst MUSS für die verschiedenen Teilfunktionen geeignete Fehlermeldungen erzeugen und diese ohne zeitlichen Verzug in Form eines UDP-Multicast an die von der gematik bestimmten Ziele übermitteln. **[<=]**

A_19899 - Fehlermeldungen sind nutzerfreundlich und basieren einheitlich auf UTC
Es folgt ein Beispiel einer möglichen Fehlermeldung, welche vom IdP-Dienst für alle durch ihn bereitgestellten Funktionen in ähnlicher Weise und gleicher Qualität umgesetzt werden MUSS.
Ist die Signatur nicht vorhanden, defekt oder stimmt diese mit der vom Authenticator registrierten URI nicht überein, bricht der Authorization-Endpunkt die Bearbeitung mit dem registrierten Fehlercode und einer für den Nutzer verständlichen Fehlermeldung ab.

Fehlermeldungscode	Fehlermeldungstext
IDPD_1001.1: 1583844803	*Signature Consent: fehlende Signatur. [10.03.2020 13:53:23]*
IDPD_1001.2: 1583844803	*Signature Consent: falsche URI. [10.03.2020 13:53:23]*
IDPD_1001.3: 1583844803	*Signature Consent: falscher Algorithmus. [10.03.2020 13:53:23]*

[<=]

Schnittstellenbeschreibung des IdP-Dienstes

Der IdP-Dienst bietet zahlreiche Schnittstellen gegenüber unterschiedlichen Akteuren inner- und außerhalb der TI anbieten muss, ist es notwendig die einzelnen Schnittstellen zu beschreiben.

Es folgt eine Auflistung der durch den IdP-Dienst anzubietenden Schnittstellen sowie **vorerst** eine kurze Beschreibung der auf dieser Schnittstelle zu verarbeiten Daten. Die Auflistung der Schnittstellen erfolgt im Verlauf ihrer zeitlichen Nutzung am Authorization Server.

Die erste tokenbezogene Anfrage an den Authorization Server geht am Authorization-Endpunkt ein.

Hier reicht der Authenticator den `"CONSENT"` ein, mit welchem das `"ID_TOKEN"` erstellt werden soll und erhält den `"ACCESS_CODE"` zurück. Bei der ersten Kontaktaufnahme erzeugt der Authorization Server die `"SUBJECT_SESSION"`, welche im weiteren Verlauf als Zeitpunkt der letzten Authentisierung gegen die eGK oder den eHBA gewertet wird. Basierend darauf dürfen weitere `"ID_TOKEN"` und `"REFRESH_TOKEN"` für andere Anwendungsfrontend und Fachdienste ausgegeben werden, wenn das jeweils vorliegende Claim durch die dem Authorization Server vorliegenden Informationen bedient werden kann und die PIN-Eingabe der letzten Authentifizierung zeitlich noch brauchbar ist. Ist der Zeitpunkt der letzten Authentisierung zu lange her oder wird der Authenticator zum ersten Mal gestartet, muss eine Authentisierung erfolgen.

Der Vorgang der Authentifizierung gegen die eGK oder den eHBA ist nicht Bestandteil dieser Spezifikation, sondern wird in einem gesonderten Dokument beschrieben, da sich die daraus hervorgehenden Anforderungen denen den mobilen Teil auf dem Endgerät des Nutzers richten.

4.3 Begriffsdefinition

Die folgende Tabelle enthält die Abkürzungen (für die private Schlüssel PrK und für öffentliche Schlüssel PUK) der verschiedenen Akteure und deren Verwendung, wobei

diese Informationen für den Authorization Server nicht angegeben sind, da hier ausschließlich die URI des Downloadpunktes des Discovery Document hinterlegt wäre.

Speziell die Verschlüsselung der einzelnen Artefakte befindet sich noch in Diskussion. Einige der aktuell vorgesehenen Verschlüsselungen können womöglich entfallen und damit auch eine Reihe der jetzt benannten Schlüssel.

	PUK	URI PUK	privateKey	URI Dienst
Authenticator (AUTH_MOD)	PUK_MOD	PUK_URI_MOD	PRK_MOD	URI_MOD
Anwendungsfrontend (FRONT)	PUK_FRONT	PUK_URI_FRONT	PrK_FRONT	URI_FRONT
Authorization Server				URI_DD
Authorization-Endpunkt (AUTH)	PUK_AUTH	PUK_URI_AUTH	PrK_AUTH	URI_AUTH
Discovery-Endpunkt (DISC)	PUK_DISC	PUK_URI_DISK	RrK_DISC	URI_DISC
Token-Endpunkt (TOKEN)	PUK_TOKEN	PUK_URI_TOKEN	PrK_TOKEN	URI_TOKEN
Introspection-Endpunkt (INT)	PUK_INT	PUK_URI_INT	PrK_INT	URI_INT
Revocation-Endpunkt (REV)	PUK_REV	PUK_URI_REV	PrK_REV	URI_REV
Userinfo-Endpunkt (INFO)	PUK_INFO	PUK_URI_INFO	PrK_INFO	URI_INFO

Die URI-DD stellt somit den zentralen Anlaufpunkt dar, anhand dessen alle weiteren „statischen" Dienste und Akteure adressiert werden können. Alle dynamisch registrierten Akteure (Endgerät des Nutzers, Authenticator und Frontend des Versicherten) werden bei der ersten Registrierung am Authorization-Endpunkt gespeichert. Im späteren Verlauf sind deren URI immer Bestandteil der Redirection des signierten Tokens bzw. Access Codes, wodurch eine Adressierung über einen Hilfsdienst nicht notwendig ist.

4.4 Registrierung von Endgerät und Anwendungsfrontend

A_19882 - Endgeräte und Anwendungsfrontend Registrierung
(Zuweisung IdP-Dienst, Frontend)
Alle Endgeräte Anwendungen und Endgeräte Anwendungsfrontends des Nutzers MÜSSEN sich beim Authorization Server am Authorization-Endpunkt gemäß [openid-heart-oauth2-1_0 # rfc.section.3.1.3 und RFC7591] registrieren.
[<=]

A_19883 - Anwendungsfrontend gibt sich zu erkennen (Anforderung der gematik)
(Zuweisung IdP-Dienst, Frontend)
Das Anwendungsfrontend, welches über den Authenticator am Authorization-Endpunkt eine Beantragung eines Tokens anstößt, MUSS dabei seine interne Programmkennung und die aktuell installierte Version in Form einer nachvollziehbaren Versionsnummer (Version.Major.Minor.Build) mitteilen.
[<=]

Beispiel:

```
"<Programmbezeichnung> 1.3.15.125634"
```

A_19884 - Voraussetzung der dynamischen Registrierung (Service Discovery)
(Zuweisung IdP-Dienst, Fachdienste, Frontend)
Bei der Registrierung von Endgeräten und Anwendungsfrontends MUSS das Discovery Documents (DD) [RFC8414] eingelesen und ausgewertet und danach die darin befindlichen URI bzw. benötigten PUK‘s der Dienste verwendet werden. **[<=]**

A_19892 - Dynamische Registrierung (Authenticator)
(IdP-Dienst und Frontend)
Bei der dynamischen Registrierung MUSS der Authenticator neben seiner absoluten URI "URI_MOD" auch seinen öffentlichen Schlüssel "PUK_MOD" sowie die Versionsnummer und Bezeichnung des anzumelden Authenticators einreichen.
[<=]

A_19893 - Dynamische Registrierung (Anwendungsfrontend)
(IdP-Dienst und Frontend)
Bei der dynamischen Registrierung MUSS das Anwendungsfrontend neben seiner absoluten URI "URI_FRONT" auch seinen öffentlichen Schlüssel"`PUK_FRONT`" sowie die Versionsnummer und Bezeichnung des anzumelden Anwendungsfrontend einreichen.
[<=]

A_19894 - Dynamische Registrierung (Absicherung durch TLS)
Die Übertragungen bei der dynamischen Registrierung MÜSSEN mit dem öffentlichen Schlüssel des Authorization-Endpunktes "`PUK_AUTH`" verschlüsselt und zusätzlich serverseitig TLS-gesichert erfolgen. Der "`PUK_AUTH`" ist über die entsprechende URI "`URI_PUK_AUTH`" aus dem Discovery Document zu beziehen.
[<=]

A_19854 - Zwischenspeichern des Discovery Documents
(Zuweisung Fachdienste, Frontend)
Der Authenticator und das Anwendungsfrontend sowie Fachdienste SOLLEN das Discovery Document zwischenspeichern. Die URI der Downloadpunkte der PUK "URI_PUK_FD" der angebotenen Fachdienste ändern sich generell nicht. Die Inhalte der Discovery Documents ändern sich nur bei der Registrierung neuer Fachdienste oder bei Änderunegn derer URI's.
[<=]

A_19855 - Inhalt der Fachdienste im Discovery Documents
(Zuweisung IdP-Dienst, Fachdienste, Frontend)
Fachdienste MÜSSEN im Discovery Document die URI "`URI_FD`" eintragen, unter welcher der von ihnen angebotene Service erreichbar ist. Außerdem MUSS im Discovery Document die URI eingetragen sein, unter welcher sich Akteure auf sicherem Wege TLS-gesichert den öffentlichen Schlüssel "`URI_PUB_FD`" des Dienstes beschaffen können. Zudem MUSS angegeben werden, welche Algorithmen vom Dienst unterstützt werden.

Das Discovery Document enthät weitere Informationen welche den IdP-Dienst betreffen.
[<=]

A_19954 - Einbindung des Primärsystems
Das Primärsystem (PVS, AVS und KVS) MUSS eine gesonderte Schnittstelle implementieren, welche die Funktionalität des Authenticators übernimmt und die eigentlich mit der Smartcard durchgeführte Challenge-Response-Verfahren bei mindestens gleichwertiger Qualität abbildet.
[<=]

A_19955 - Primärsysteme Herkunft des Schlüsselmaterials (TLS-Sicherung)
~~Primärsysteme **MÜSSEN** das für die TLS-Sicherung benötigte Schlüsselmaterial "C.FD.TLS-S" gemäß [gemSpec_PKI] von der Komponenten-PKI beziehen.~~
[<=]

A_19956 - Primärsysteme Herkunft des Schlüsselmaterials (JWT, JWE, JWS)
Primärsysteme MÜSSEN das für die JSON-Verschlüsselung und Signatur benötigte Schlüsselmaterial "PRK_MOD" und "PUK_MOD" gemäß [gemSpec_PKI] selbst erzeugen. Algorithmen und Hashverfahren sind gemäß [gemSpec_Krypt] anzuwenden.
[<=]

4.5 Zähler, Zeitstempel und Performance

Aufgaben und Inhalte der Zähler, Zeitstempel und Performance-Informationen beinhalten niemals schützenswerte Informationen des Nutzers oder der angesprochenen Fachdienste. Bei diesen Daten geht es um die reine Überwachung der Dienstqualität und somit Einhaltung der vereinbarten Service-Level (SLA).

Der IdP-Dienst stellt Teildienste zur Verfügung, mit deren Hilfe es Dienstanbietern ermöglicht wird, Nutzer der TI zu identifizieren und somit die Zugriffsrechte auf die ihnen zugänglichen Informationen zu organisieren. Es ist daher immanent wichtig, dass eine durchweg akzeptable Dienstgüte sowohl in der geforderten Quantität (Anzahl parallel möglicher Zugriffe) als auch Qualität (Integrität, Vertraulichkeit und Verfügbarkeit) bereitsteht. Daher muss der IdP-Dienst viele Vorgänge protokollieren und über den Status in kurzen zeitlichen Abständen automatisiert berichten. Hierbei kann es nicht zielführend sein, alle Protokolldaten auszuliefern und diese ohne vorherige Aggregation auszuliefern.

Vorgaben zur Performance einzelner Schnittstellen befinden sich noch in Abstimmung. Daraus resultierend können auch noch keine finalen Festlegungen für das Reporting in diesem Dokument sowie in gemSpec_Perf getroffen werden.
Der Funktionsblock App-Check befindet sich noch in Diskussion.

A_19856 - Granularität der Zeitstempel
(Zuweisung IdP-Dienst, Fachdienste)
Der Authorization-Server MUSS alle Zeitstempel mindestens mit der Granularität Sekunden nach 01.01.1970 00:00 Uhr UTC erfassen.
[<=]

A_19857 - Abruf der Performance-Protokollierung zu jeder Zeit
(Zuweisung IdP-Dienst)
Der Authorization Server MUSS die Zähler-, Zeitstempel- und Performance- Daten

jederzeit auf Abruf bereit stellen.
[<=]

A_19858 - Regelmäßigkeit
(Zuweisung IdP-Dienst)
Der Authorization Server MUSS die gesammelten Performance Werte, im fünfminütigen Rhythmus, in Form einer signierten UDP-Multicast POST Nachricht an die von der gematik maximal vier bestimmten Empfänger, senden.
[<=]

A_19859 - Bereitstellungsart
(Zuweisung IdP-Dienst)
Das Datenformat der fünfminütigen Statusmeldung bestimmt der folgende Basisdatensatz, aus welchem mit dem RRD-Tool (Round-Robin-Datenbank) kurz- mittel- und langfrist-Prognosen und damit verbundene Graphen erzeugt werden können. Durch die im Basisdatensatz mitgeführten Zähler lässt sich zudem ableiten, wie sich der IdP-Dienst bei zunehmender Last voraussichtlich verhalten wird. Alle *Zeitstempel* MÜSSEN in Sekunden nach 01.01.1970 00:00:00 Uhr UTC angegeben werden.
[<=]

Beispieldatensatz:

1583841876:0.0121:***1583841876***:0.1101:***1583841872***:***1583841843***:***1583841821***:***158384161***\ :891:890:871:331:810:0.55:0.60:0.48:1442351

Bedeutung der Parameter in ihrer Reihenfolge im Beispieldatensatz:

Parameter	Beschreibung
"last_consent"	Zeitpunkt des zuletzt eingereichten Consent
"time_consent"	Durchschnittliche Prüfungszeit inkl. Signatur- und Schemaprüfung
"last_access"	Zeitpunkt der Herausgabe des letzten "ACCESS_CODE"
"time_access"	Durchschnittliche Haltedauer "ACCESS_CODE" zu Token-Ausgabe
"last_token"	Zeitpunkt der letzten "ID_TOKEN"-Ausgabe mit "ACCESS_CODE"
"last_refresh"	Zeitpunkt der letzten "REFRESH_TOKEN"-Annahme
"last_intro"	Zeitpunkt der letzten Introspection mit Token-ID und Ergebnis
"last_userinfo"	Zeitpunkt der letzten herausgegebenen Userinfo mit Token-ID
"count:c"	Zähler der Anträge am Authorization-Endpunkt
"count:a"	Zähler herausgegebener "ACCESS_CODE"
"count:t"	Zähler herausgegebener Token
"count:i"	Zähler durchgeführter Token Introspections
"count:u"	Zähler herausgegebener Userinformationen
"server:avg"	Server-Auslastung load average: 0.55, 0.60, 0.48
"server:con"	Anzahl offener "SUBJECT_SESSION"

Bisher mit OPS auf der Tonspur so in etwa abgestimmt. Genauer Wert ist noch zu suchen und abzustimmen.

A_19501 - Funktionsblock App-Check für die Betriebsdatenerfassung
Jede Komponente mit einer Kommunikationsschnittstelle zu einer mobilen Anwendung (App) MUSS den Funktionsblock "App-Check" implementieren.
Der Funktionsblock "App-Check" MUSS die Identifikatoren der sich verbindenden App [Hersteller-ID, Versions-ID und Build-ID] erfassen und in einem konfigurierbaren Intervall an die Betriebsdatenerfassung übermitteln.
Voreingestellt ist 5 Minuten.**[<=]**

A_19503 - Erheben von Betriebsdaten von Apps (Anzahl der Verbindungen)
Jeder Komponente (App) MUSS vor der fachlichen Kommunikation an den Funktionsblock "App-Check" des Kommunikationspartners seine Identifikatoren [Hersteller-ID, Versions-ID und Build-ID] übermitteln.**[<=]**

A_19504 - Erheben von Betriebsdaten von Apps (Erfolgsermittlung)
Nach jedem Anwendungsfall und vor Beendigung der Kommunikation MUSS jede Komponente (App) an den Funktionsblock "App-Check" des Kommunikationspartners seine Identifikatoren [Hersteller-ID, Versions-ID und Build-ID] und Erfolg oder Misserfolg des Anwendungsfalles übermitteln.
Für den Fall des Misserfolges MUSS eine Fehlermeldung mit dem Namen der fehlgeschlagenen Operation erfolgen. **[<=]**

A_19502 - Ausschluss von Apps an der Kommunikation durch Funktionseinheit App-Check
Die Funktionseinheit App-Check MUSS Apps, welche die die Sicherheitsvorgaben nicht erfüllen, effektiv von der Kommunikation ausschließen.
[<=]

5 Funktionsmerkmale

5.1 Authorization Server Metadata (Discovery Document)

Der Authorization Server ist eine künstliche Metaebene um bestehende Identitäten zu prüfen und das Prüfungsergebnis in einer einheitlichen Form abgestimmt und durch zusätzliche Mechanismen gesichert bereitzustellen. Basis dieser Dienstleistung ist ein vertrauenswürdiges Verzeichnis, aus welchem hervorgeht, an welchen Schnittstellen dieser Dienst oder seine Teildienste erreichbar ist, wie diese Schnittstellen abgesichert sind und woher man die zur etablierung der gewünschten Sicherheit erforderlichen Materialien beziehen kann. Gemäß dem Verwendeten Standard OpenIDConnect mit OAuth 2.0 kommen JSON Web Token (JWT), JSON Web Encryption (JWE), JSON Web Signature (JWS) und JSON Web Key (JWK) zum Einsatz. Das Adressieren der Fachdienste und serverbasierten Systeme ist einfach, da diese erstmal im Discovery Document eingetragen allen Interessierten zugänglich sind.

Damit auch bisher nicht im Discovery Document eingetragene Akteure adressierbar werden, müssen diese dem Authorization Server bekannt gemacht werden. Bisher unbekannte Entitäten können zur Laufzeit anhand einer dynamischen Registrierung nachträglich bekannt gemacht werden und bekommen so die Möglichkeit auf bereits bestehende Fachdienste zuzugreifen.

Damit ein vorher nicht registriertes Endgerät oder dessen Anwendungsfrontend in das Verzeichnis adressierbarer Entitäten aufgenommen werden können, müssen sich diese beim Authorization Server anmelden bzw. bei der ersten Kontaktaufnahme registrieren. Im Verlauf der Registrierung wird ein sogenannter Authenticator installiert, dessen Aufgabe es ist die Kommunikation mit bestimmten Schnittstellen des IdP-Dienstes zu übernehmen und den Datenaustausch an die formalen Erfordernisse anzupassen.

Um kein Henne-Ei-Problem zu erschaffen, werden die für alle Akteure grundlegenden Schnittstellen im sogenannten Discovery Document zusammengefasst uund gemäß [RFC8414 "OAuth 2.0 Authorization Server Metadata"] veröffentlicht.

Alle Akteure, welche den IdP-Dienst nutzen wollen, sind angehalten dieses Discovery Document zu lokalisieren, herunterzuladen, zu prüfen und den Inhalt in den geplanten Betrieb einzubeziehen.

Im aktuellen Stand des Dokumentes fehlen, noch in Diskussion befindliche, Festlegungen zur Erweiterung des Integritätsschutz des Discovery Documents sowie Ergänzungen zum Transport von Schlüsselmaterial über das Discovery Document. Die Standards sehen Verschlüsselungen vor aber lassen die Methoden des Schlüsselaustausch offen und die gematik ist noch in Abstimmung zu praktikablen Methoden.

5.1.1 Aufbau des Discovery Documents

Aufbau gemäß RFC erweitert durch gematik-spezifische Erweiterungen um HEART I & II zu verwirklichen.

5.1.2 Erneuerung des Discovery Documents

Intervall der Erneuerung und Gründe für eine Änderung

5.1.3 Schutz des Discovery Documents

Schutz des Discovery Documents auf Dateiebene und während des Transportes durch unsichere Netzwerke

5.2 Authorization-Endpunkt

Vorbedingung ist, dass der Authenticator bereits eine "SUBJECT_SESSION" mit dem Authorization Server etabliert, sich das Discovery Document heruntergeladen und dieses erfolgreich ausgewertet hat.

A_19860 - Der Authorization-Endpunkt Standards
(Zuweisung IdP-Dienst)
Der IdP-Dienst MUSS die Schnittstelle „Authorization-Endpunkt" gemäß [RFC6749 The OAuth 2.0 Authorization Framework] und [RFC8252 „OAuth 2.0 for Native Apps"] und weiteren darin verwiesenen Standards implementieren.
[<=]

A_19861 - Authorization-Endpunkt Authenticator APP
(Zuweisung IdP-Dienst, Frontend)
Der Anbieter des IdP-Dienstes MUSS den Authenticator über den für das jeweilige Betriebssystem üblichen Software-Verteilungspunkt selbst bereitstellen oder bereitstellen lassen.
[<=]

A_19862 - Authenticator im Apple App Store
(Zuweisung IdP-Dienst, Frontend)
Der Anbieter des IdP-Dienstes MUSS den Authenticator für das Betriebssystem Apple im dafür vorgesehenen Apple App Store für den Nutzer kostenfrei bereitstellen.
[<=]

A_19863 - Schutz vor überalterter Software (Apple)
Der Anbieter IdP-Dienst MUSS dafür Sorge tragen, dass die im Apple App Store veröffentlichte Software bei Änderungen automatisiert aktualisiert wird, sodass jederzeit die dauerhafte Verwendung fehlerhafter Software ausgeschlossen werden kann.
[<=]

A_19864 - Authenticator im Google Play Store
(Zuweisung IdP-Dienst, Frontend)
Der Anbieter des IdP-Dienstes MUSS den Authenticator für das Betriebssystem Google Android im dafür vorgesehenen Google Play Store für den Nutzer kostenfrei bereitstellen.
[<=]

A_19865 - Schutz vor überalterter Software (Android)
Der Anbieter des IdP-Dienstes MUSS dafür Sorge tragen, dass die im Google Play Store veröffentlichte Software bei Änderungen automatisiert aktualisiert wird, sodass jederzeit die dauerhafte Verwendung fehlerhafter Software ausgeschlossen werden kann.
[<=]

5.2.1 Authorization Server Eingangsdaten

A_19853 - Protokollierung der Consent-Bestätigung
(Zuweisung IdP-Dienst, Frontend)
Am Authorization-Endpunkt MUSS der IdP-Dienst den vom Nutzer am Authenticator bestätigten "consent" protokollieren. Die Bestätigung des Consent wird im Zähler vermerkt und der Zeitstempel für die letzte Consent-Abstimmung im Parameter "last_consent" (siehe Zähler, Zeitstempel und Performance) protokolliert.
[<=]

A_19850 - Enschlüsseln der Eingangsdaten am Authorization-Endpunkt
(Zuweisung IdP-Dienst)
Der Authorization-Endpunkt entschlüsselt die Daten mit dem zum aktuell im Discovery Document veröffentlichten öffentlichen Schlüssel "PUK_AUTH" gehörenden privaten Schlüssel "PRK_AUTH". Ist die Entschlüsselung mit diesem Schlüssel nicht möglich, SOLL der Authorization-Endpunkt versuchen, die Entschlüsselung mit privaten Schlüssel der vorhergehenden Generation vorzunehmen.
[<=]

A_19851 - Aufbewahrung alter Schlüssel
(Zuweisung IdP-Dienst, Fachdienste)
Der IdP-Dienst MUSS die in den letzten 72 Stunden verwendeten Schlüssel aufbewahren, um eine nachträgliche Entschlüsselung von Daten zu ermöglichen.
[<=]

A_19852 - Verwendung des Attributes "auth_time"
(Zuweisung Fachdienste und IdP-Dienst)
Der Authorization-Endpunkt MUSS den Parameter "auth_time" mit dem Zeitpunkt der letzten Authentisierung gegen das zugelassene Authentifizierungsmittel (z.B. Auslösen der Signatur durch Smartcard mit PIN-Eingabe) setzen.
[<=]

A_19848 - Verwendung des Attributes "Bearer"
(Zuweisung Alle)
Der Token-Endpunkt MUSS Token so ausstellen, dass diese die eineindeutige Kennung des Inhabers "bearer" enthalten. [rfc6750].
[<=]

Anhand dieser kann zu jeder Zeit die Quelle des Tokens auf der Nutzerseite adressiert werden

A_19849 - ACCESS_CODE und ID_- oder REFRESH_TOKEN nur für gültige Zertifikate
Der Authorization-Endpunkt MUSS ausschließen, "ACCESS_CODE" oder Token für existente und nicht widerrufene Entitäten auszustellen. Das vorgetragene Zertifikat des Antragstellers MUSS vom Authorization-Endpunkt gegen den OCSP-Responder innerhalb der TI auf Gültigkeit geprüft werden.
[<=]

A_19835 - Entschlüsselung des Consent
Der Authorization-Endpunkt MUSS den verschlüsselten Consent mit seinem eigenen privaten Schlüssel "PRK_AUTH" entschlüsseln.
[<=]

A_19836 - Signaturprüfung des Consent

Der Authorization-Endpunkt MUSS die Signatur des entschlüsselten "consent" gegen den PUK des Authenticators "PUK_AUTH" prüfen.
[<=]

A_19837 - Schematische Prüfung des Consent
Der IdP-Dienst MUSS den eingereichten Consent auf dessen Übereinstimmung mit dem vorliegenden Schema (Claim) zum beantragten Token abgleichen, insbesondere die "redirect_uri" aus dem Registrierungszusammenhang. Stimmen das Schema des Consent und das des vorliegenden Claims nicht überein, MUSS der Authorization-Endpunkt die Bearbeitung mit dem registrierten Fehlercode und einer für den Nutzer verständlichen Fehlermeldung abbrechen. Insbesondere MUSS der Authorization-Endpunkt prüfen, ob die zugrundeliegende "SUBJECT_SESSION" bereits widerrufen ist.
[<=]

A_19838 - Verarbeitung des Consent
Hat der Authorization-Endpunkt den eingereichten Consent allen vorgesehenen Prüfungen unterzogen und sind dabei keine Fehler aufgetreten, MUSS der Authorization-Endpunkt die Ausstellung des oder der im Claim vereinbarten "ID_TOKEN" und ggf. "REFRESH_TOKEN" mit den im Claim vorliegenden Parametern veranlassen. Der Zeitpunkt der letzten Authentisierung MUSS im Parameter "auth_time" festgehalten und DARF bis zur erneuten Authentisierung NICHT verändert werden.**[<=]**

A_19839 - Der Authorization-Endpunkt bestätigt ausschließlich Zertifikatsinformationen
Um sicher zu stellen, dass der Nutzer berechtigt ist, die vorgetragene Identität (Zertifikat) zu nutzen MUSS der Authorization-Endpunkt bei der Annahme des Zertifikates durch ein Challenge Response Verfahren prüfen, ob der Nutzer auch die zum Zertifikat gehörige PIN kennt.
Die durch eines der folgenden Zertifikate nachgewiesenen Informationen dürfen vom Userinfo-Endpunkt preisgegeben werden, soweit diese dem Claim des anfragenden Fachdienstes entsprechen:
- C.CH.AUT [OID:1.2.276.0.76.4.70] bei eGK (elektronische Gesundheitskarte)
- C.HP.AUT [OID:1.2.276.0.76.4.75] bei eHBA (elektronischer Heilberufsausweis)
- C.HCI.AUT [OID:1.2.276.0.76.4.77] bei SMC-B (Secure Module Card - B)

[<=]

A_19840 - Inhalte des Claims
(Zuweisung IdP-Dienst, Fachdienste)
Der IdP-Dienst MUSS "ID_TOKEN" und "REFRESH_TOKEN" für unterschiedliche Fachdienste gemäß den mit dem jeweiligen Fachdienst abgestimmten Claims bereitstellen. Sind Inhalte des Claims teilweise oder das gesamte Claim für einen registrierten Fachdienst nicht gesetzt, befüllt der IdP-Dienst die einzelnen Parameter gemäß der folgenden Maximalwerte:
[<=]

A_19841 - Maximale Gültigkeitsdauer einer "SUBJECT_SESSION"
Die Gültigkeitsdauer einer "SUBJECT_SESSION" DARF NICHT länger als 86400 Sekunden (24 Stunden) betragen.
Der Parameter "auth_time" beinhaltet den Zeitpunkt der letzten Authentisierung.
[<=]

Es liegt in der Verantwortung und im Ermessen des Betreibers des Fachdienstes, Anforderungen zu definieren, wie lange die letzte Authentisierung nachgenutzt werden darf.

A_19842 - Maximale Gültigkeitsdauer des "ACCESS_CODE"

Diese maximale Gültigkeitsdauer des "ACCESS_CODE" DARF NICHT länger als 180 Sekunden (3 Minuten) nach der Übergabe an den Authenticator und maximal 60 Sekunden (1 Minute) nach der Übergabe an das Anwendungsfrontend betragen. Binnen dieser Zeit MUSS der "ACCESS_CODE" gegen ein "ID_TOKEN" eingetauscht sein.
[<=]

Die Gültigkeitsdauer des "ACCESS_CODE" wird im Claim des angesprochenen Fachdienstes definiert.

A_19843 - Maximale Gültigkeitsdauer des "REFRESH_TOKEN"
Die maximale Gültigkeitsdauer von "REFRESH_TOKEN" DARF NICHT länger als 14400 Sekunden (4 Stunden) betragen.
[<=]

Die Gültigkeitsdauer des "REFRESH_TOKEN" wird im Claim des angesprochenen Fachdienstes definiert.

A_19844 - Maximale Gültigkeitsdauer des "ID_TOKEN"
Die maximale Gültigkeitsdauer des "ID_TOKEN" DARF NICHT länger als 900 Sekunden (15 Minuten) betragen.

[<=]

Die Gültigkeitsdauer des "REFRESH_TOKEN" wird im Claim des angesprochenen Fachdienstes definiert.

A_19845 - Informationen im Claim
Fachdienste MÜSSEN die im Claim angeforderten Informationen über den Nutzer bei ihrer Registrierung angeben. Andere Informationen, als die im Claim geforderten, DARF der Token-Endpunkt NICHT herausgeben.
Alle Token MÜSSEN die eineindeutige Kennung des Inhabers "bearer" z.B. Telematik-ID enthalten. **[<=]**

A_19977 - Keine Token für widerrufene Entitäten
Der Authorization-Endpunkt MUSS ausschließen, "ACCESS_CODE", "ID_TOKEN" oder "REFRESH_TOKEN" für nicht existente oder widerrufene Entitäten auszustellen.
[<=]

A_19978 - Zertifikatsprüfung gegen OCSP-Responder
Das Zertifikat des Antragstellers MUSS immer gegen den OCSP-Responder innerhalb der TI auf Gültigkeit geprüft werden.
[<=]

Anhand der eindeutigen Kennung (Telematik-ID bzw. KVNR) kann zu jeder Zeit die Quelle des Tokens auf der Nutzerseite adressiert werden.

5.2.2 Authorization-Endpunkt Ausgangsdaten

Konnten alle Prüfungen des eingereichten Consent erfolgreich abgeschlossen werden, erstellt der Authorization-Endpunkt ein "ID_TOKEN" ggf. ergänzt durch ein damit verbundenes "REFRESH_TOKEN". Die Übertragung des "ID_TOKEN" erfolgt jedoch nicht direkt über den Authenticator, sondern in Form eines "ACCESS_CODE". Dieser

"ACCESS_CODE" wird vom Authorization-Endpunkt so ausgestellt (verschlüsselt), dass dieser nur vom Anwendungsfrontend verwendet werden kann. Das/die Token werden am Token-Endpunkt zum Download bereitgestellt, wo das jeweilige Anwendungsfrontend diese gegen gleichzeitige Vorlage von "ACCESS_CODE" und des eigenen "SECRET" (auf welchem der bereits vorliegende Hash-Wert beruht) erhält.

A_19846 - Signatur des "ACCESS_CODE"
(Zuweisung IdP-Dienst, Frontend)
Der Authorization Server MUSS den "ACCESS_CODE" mit dem "PRK_AUTH" signieren, damit der Authenticator sicher gewährleisten kann, dass der eingehende "ACCESS_CODE" tatsächlich von diesem stammt [RFC7519 # section-7.1].
[<=]

A_19847 - Verschlüsselung des "ACCESS_CODE"
(Zuweisung IdP-Dienst, Frontend)
Der Authorization Entpunkt MUSS den "ACCESS_CODE" vor dem Übertragen mit dem öffentlichen Schlüssel des Anwendungsfrontend "PUK_FRONT" verschlüsseln [RFC7519 # section-7.1 und RFC7523 # section-7].
[<=]

A_19832 - Sichere Übertragung des "ACCESS_CODE"
(Zuweisung IdP-Dienst, Fachdienste)
Der Authorization-Endpunkt MUSS den Transport des Authorization Code über unsichere Netze (z.B. Internet) durch Verwendung von Transport Layer Security (TLS) gemäß den Vorgaben der [gemSpec_Krypt] sichern[RFC7523 # section-7].
[<=]

5.3 Redirection Endpunkt

Der Authorization Server muss einen Redirection Endpunkt gemäß [RFC6749 # section-3.1.2] bereitstellen, damit das Anwendungsfrontend die Adressierung zum Authenticator über den IdP_Dienst auflösen kann. Ohne Redirection Endpunkt ist es ein schwieriges oder gegebenenfalls unmögliches Unterfangen die aktuelle IP-Adresse und somit URI des Authenticators zu ermitteln, da keine namensbasierte Adressierung erfolgen kann. Es muss daher einen Dienst geben, der vom Anwendungsfrontend aus leicht erreichbar ist und der zudem die Auflösung des Adressierungshindernisses beseitigt.

Der IdP_Dienst muss zu diesem Zweck eine Datenbank betreiben, anhand welcher es einem Anwendungsfrontend möglich wird den mit ihm verknüpften Authenticator zu ermitteln. Bei der Registrierung des Authenticators wird dieser durch den Authorization Server mit einem eineindeutigen Identifikationsmerkmal versehen und in der Datenbank abgespeichert. Ber der Registrierung eines Anwendungsfrontend wird der Nutzer aufgefordert, die eindeutige Verknüpfung zwischen dem Anwendungsfrontend und dem Authenticator zu bestätigen. Der Authorization Server gibt bei der Registrierung eines Anwendungsfrontend eine URL oder einen QR-Code bekannt, welche mit dem betroffenen Authenticator aufzurufen ist. Ruft der Authenticator diese URL durch Eingabe von Hand, nutzen einer entsprechenden Verlinkung oder einscannen des QR-Code auf wird vom IdP_Dienst die Rückfrage gestellt, ob die Verknüpfung zwischen Authenticator und diesem Anwendungsfrontend erfolgen soll. Ist diese Rückfrage bestätigt, speichert der IdP_Dienst diese n:m Beziehung in der Datenbank ab und setzt sie mit einem Zeitstempel versehen auf aktiv. Der Zeitstempel dient hier der Forensik, woraus sich ergibt, dass der Eintrag nicht mehr gelöscht wird. Im Falle einer späteren Löschung findet die Deaktivierung des Eintrags mit erneutem Zeitstempel statt. Auf diese Weise

kann eine einmal etablierte Beziehung, auch lange nachdem diese wieder entfernt wurde, erkannt werden.

Ruft nun das Anwendungsfrontend den Redirection Endpunkt auf, wobei es seinen eigenen Identifier überträgt, kann der Authorization Server den oder die für dieses Anwendungsfrontend registrierten Authenticator ermitteln und den HTTP/1.1 GET oder POST Request an den Authenticator weiterleiten.

Hinweis:

Es ist erforderlich, dass der Authenticator mittels HTTP/1.1 POST-Request erreichbar ist und die Anrufe nicht an einer Firewall oder einem Application Layer Gateway (ALG) geblockt werden. Ist dies der Fall, muss der Authenticator erst eine Verbindung zum Authorization Server etablieren, damit diese Sicherheitsfunktion kurzeitig abgestellt wird. Der IdP_Dienst muss in diesem Fall auch den von der Firewall oder dem ALG übermittelten Port in die Redirection mit einbeziehen.

A_20106 - Bereitstellung Redirection Endpunkt
Der Authorization Server MUSS einen Redriection Endpunkt gemäß [RFC6749 # section-3.1.2] bereitstellen.
[<=]

A_20110 - Absicherung Redirection Endpunkt
Der Redirection Endpunkt MUSS HTTP/1.1 GET oder POST Requests ohne Fehlermeldung ablehnen, wenn der im HTTP/1.1-Header angegebene Identifier in der Datenbank nicht vorkommt. Anwendungsfrontend und Authentictaor MÜSSEN vor der Nutzung des Redirection Endpunkt beim Authorization Server registriert sein.
[<=]

5.3.1 Eingangsdaten Redirection Endpunkt

Dieser Abschnitt muss noch ausformuliert werden

A_20107 - Redirection Endpunkt Auswertung HTTP-Header
Der Redirection Endpunkt MUSS anhand der im Aufruf (HTTP/1.1 GET oder POST Request) in den Headerinformationen ermitteln von welchem Anwendungsfrontend der Aufruf stammt und in der Datenbank ermitteln, an welchen Authenticator die Anfrage weiterzuleiten ist.
[<=]

5.3.2 Ausgangsdaten Redirection Endpunkt

Dieser Abschnitt muss noch ausformuliert werden

A_20108 - Redirection Endpunkt Ergänzung mit Portangaben
Der Redirection Endpunkt MUSS die Weiterleitung eines HTTP Request an einen Authenticator durch dessen priviligierten Port ergänzen.
[<=]

5.4 Token-Endpunkt

Am Token-Endpunkt nimmt der Authorization Server einerseits "ACCESS_CODE" und andererseits "REFRESH_TOKEN", welcher er selbst am Authorization-Endpunkt oder am

Token-Endpunkt ausgegeben hat, entgegen. Da beide vom Authorization Server selbst erstellt wurden, ist deren Prüfung auf Integrität keine besondere Herausforderung. Allerdings muss der Token-Endpunkt beim Einreichen eines "ACCESS_CODE" das dabei mit übertragenen "SECRET" verarbeiten, um daraus mit dem identischen Hash-Verfahren auf den identischen Hash-Wert zu kommen, wie dieser beim Beantragen des Tokens eingereicht wurde. Auf diese Weise kann der Token-Endpunkt sicherstellen, dass diejenige Entität dieselbe ist, welchen den ursprünglichen Token Beantragungsprozess angestoßen hat. Das verwendete Hash-Verfahren ist im Authorization Request anzugeben.

5.4.1 Token-Endpunkt Eingangsdaten

A_19825 - Annahme und Prüfung von "ACCESS_CODE" und "SECRET"
(Zuweisung IdP-Dienst)
Der Token-Endpunkt MUSS den vom Anwendungsfrontend übertragenen "ACCESS_CODE" nach Überprüfung des zeitgleich eingereichten "SECRET" entwerten und das mit dem ursprünglichen "CONSENT" ausgestattete "ID_TOKEN" gesichert herausgeben.
[<=]

A_19826 - "ACCESS_CODE" einmalige Verwendung
Der Authorization Server MUSS beide bestehenden "SUBJECT_SESSION" terminieren, wenn ein "ACCESS_CODE" wiederholt eingereicht wird.
[<=]

Hinweis: Da es nicht sicher ist, ob in einem solchen Fall der erste oder zweite Vortragende den "ACCESS_CODE" rechtmäßig verwendet, müssen beide mit dem "ACCESS_CODE" in verbindung stehenden "SUBJECT_SESSION" eliminiert werden.

A_19827 - "ID_TOKEN" Protokollierung in allen Fällen
Der Token-Endpunkt MUSS die Herausgabe eines "ID_TOKEN" im Positiv- wie auch im Negativfall protokollieren.
[<=]

A_19828 - Annahme und Prüfung des "REFRESH_TOKEN"
(Zuweisung IdP-Dienst)
Der Token-Endpunkt MUSS die von ihm selbst ausgegebenen "REFRESH_TOKEN" annehmen und mit seinem "PRK_TOKEN" entschlüsseln. Treten bei der Entschlüsselung Bitfehler auf, MÜSSEN "REFRESH_TOKEN" und "SUBJECT_SESSION" sofort terminiert werden.
[<=]

Die erfolgreiche Entschlüsselung ist ein Garant dafür, dass das "REFRESH_TOKEN" seit dessen Herausgabe unverändert ist.

A_19829 - "REFRESH_TOKEN" Protokollierung nur im Negativfall
Der Token-Endpunkt MUSS nur im Negativfall, also im Falle der Ablehnung des "REFRESH_TOKEN", dessen Ablehnung protokollieren.
[<=]

5.4.2 Token-Endpunkt Ausgangsdaten

Da perspektivisch eine Vielzahl unterschiedlicher Fachdienste mit den vom Authorization Server bereitgestellten "ID_TOKEN" erreichbar gemacht werden, müssen alle Schnittstellen und Protokolle möglichst sicher umgesetzt werden. Hierzu gehört, dass

eine mögliche „KANN"- oder „SOLL"-Signatur gemäß HEART-Erweiterung [openid-heart-1_0 Stand 2017-05-31 und openid-heart-oauth2] verpflichtend wird. Ebenso müssen alle Token und alle mit der Beantragung der Token in Verbindung zu bringenden Informationsaustausche zusätzlich zum TLS mit den öffentlichen Schlüsseln der Empfänger verschlüsselt werden.

Anwendungen des Nutzers und Authenticator reichen hierzu bei der dynamischen Registrierung ihre privaten Schlüssel "PUK_MOD" und "PUK_FRONT" ein, wodurch diese für alle folgenden Prozessschritte von Beginn an allen Beteiligten zur Signaturprüfung und Verschlüsselung bereitgestellt werden können.

Alle vom Dienstanbieter verbreiteten Informationen müssen vor dem Verschlüsseln (mit dem PUK des Empfängers) mit dem privateKey des Dienstes signiert sein, da die mit einer TLS-Sicherung verbundene Herausgeber Identifizierung nicht in allen Anwendungsszenarien gegeben ist.

Der Empfänger der Daten muss also auch ohne TLS in der Lage sein, die Herkunft und Integrität der Daten prüfen zu können. Daher werden Verschlüsselung und Signatur immer als MUSS-Anforderung spezifiziert.

Die Absicherung des Transportweges erfolgt ausschließlich serverseitig, da gespeichertem Schlüsselmaterial im Endgerät des Nutzers, ebenso wie dem Schlüsselmaterial des Anwendungsfrontend nicht vertraut werden kann.

A_19820 - Erfolgreiche Antwort auf "REFRESH_TOKEN"
(Zuweisung IdP-Dienst)
Erfüllt das Token alle Voraussetzung gemäß [openid-connect-core], beinhaltet es alle im Claim geforderten Attribute und konnte es erfolgreich getestet werden, MUSS der Token-Endpunkt den Anwendungsfrontend das "ID_TOKEN" und ggf. ein "REFRESH_TOKEN" ausstellen.
[<=]

A_19821 - Signatur des "ID_TOKEN" und "REFRESH_TOKEN"
(Zuweisung IdP-Dienst, Fachdienste)
Der Token-Endpunkt MUSS alle erstellten "ID_TOKEN" und "REFRESH_TOKEN" mit seinem privateKey gemäß [gemSpec_Krypt#6] signieren, um dessen Integrität sicherzustellen und eine eineindeutige Erklärung über dessen Herkunft abzugeben. [RFC7523#Abschnitt 3 Spiegelpunkt 9 und openid-heart-oauth2-1_0.html#rfc.section.3.2.1] ist zu gewährleisten.
[<=]

A_19822 - Verschlüsselung des "ID_TOKEN"
(Zuweisung IdP-Dienst, Fachdienste)
Der Token-Endpunkt MUSS das "ID_TOKEN" mit dem öffentlichen Schlüssel des Fachdienstes "PUK_FD" verschlüsseln, für welchen dieses bestimmt ist, um das "ID_TOKEN" vor Kenntnisnahme durch Dritte, z.B. auch auf dem Endgerät des Nutzers, zu schützen [RFC6750 # section-5.2 und openid-heart-oauth2-1_0.html#rfc.section.3.2.1].
[<=]

A_19816 - Verschlüsselung des "REFRESH_TOKEN"
(Zuweisung IdP-Dienst)
Der Token-Endpunkt MUSS das "REFRESH_TOKEN" mit seinem eigenen öffentlichen Schlüssel "PUK_TOKEN" verschlüsseln [RFC6750 # section-5.2 und openid-heart-oauth2-1_0.html#rfc.section.3.2.1].
[<=]

A_19817 - Signatur des "REFRESH_TOKEN"
Der Token-Endpunkt MUSS das "REFRESH_TOKEN" mit seinem privaten Schlüssel "PRK_TOKEN" signieren [RFC6750 # section-5.2 und openid-heart-oauth2-1_0.html#rfc.section.3.2.1].
[<=]

A_19818 - "REFRESH_TOKEN" mathematische und zeitliche Gültigkeit
Der Token-Endpunkt MUSS die Signatur des "REFRESH_TOKEN" auf zeitliche Gültigkeit und mathematische Korrektheit überprüfen.
[<=]

A_19819 - "REFRESH_TOKEN" Integritätsprüfung
Der Token-Endpunkt MUSS anhand der Signatur die Integrität des "REFRESH_TOKEN" sicherstellen.
[<=]

A_19809 - Sichere Übertragung von "ID_TOKEN" und "REFRESH_TOKEN"
(Zuweisung IdP-Dienst)
Der Token-Endpunkt MUSS "ID_TOKEN" und "REFRESH_TOKEN" beim Transport mit Transport Layer Security (TLS) gemäß [BCP195] und [gemSpec_Krypt] schützen.**[<=]**

A_19810 - "ID_TOKEN" richten sich an dasselbe Anwendungsfrontend
(Zuweisung IdP-Dienst)
Der Token-Endpunkt MUSS "ID_TOKEN" eines Vorgangs für genau ihren Fachdienst mit dessen "PUK_FD" einheitlich verschlüsseln.
[<=]

A_19976 - "REFRESH_TOKEN" sind immer an den Token-Endpunkt gerichtet
Der Token-Endpunkt MUSS "REFRESH_TOKEN" mit seinem eigenen öffentlichen Schlüssel "PUK_TOKEN" verschlüsseln. **[<=]**

A_19811 - Adressierung der Token beim Versand
(Zuweisung IdP-Dienst)
Der Token-Endpunkt MUSS für den Versand der "ID_TOKEN"und "REFRESH_TOKEN" an das Anwendungsfrontend, die vom Authenticator im Consent der mit dem Vorgang verbundenen "SUBJECT_SESSION" gemeldete URI verwenden. Eine URI-Umleitung DARF NICHT akzeptiert werden.
[<=]

5.5 Token Introspection-Endpunkt

Der Token Introspection-Endpunkt bietet den Fachdiensten die Möglichkeit, vom Token-Endpunkt herausgegebene "ID_TOKEN", auf deren Bindung zu einer bestimmten Entität zu überprüfen. Der Token Introspection-Endpunkt liefert an dieser Schnittstelle gegen Vorlage des "ID_TOKEN" die von ihm mit dem Token im Consent bestätigten Daten des verwendeten Claim. Hier besteht für einen Fachdienst die Möglichkeit, für das ihm vorgelegte Token dessen aktuellen Gültigkeitsstatus zu erfragen. Auf diese Weise ist es möglich, Token auch während ihrer eigentlich noch andauernden Gültigkeit zu widerrufen.

Die Token Introspection soll von Fachdiensten mindestens einmal während der Gültigkeitsdauer des "ID_TOKEN" erfolgen.

A_19812 - Token Introspection Timeout
Der Token Introspection-Endpunkt MUSS auch unter maximaler Belastung eine Token Introspection ordnungsgemäß beantworten, sodass das Timeout von 3 Sekunden beim

Fachdienst nicht erreicht wird.
[<=]

5.5.1 Token Introspection-Endpunkt Eingangsdaten

A_19806 - Prüfung von "ID_TOKEN" am Introspection-Endpunkt
(Zuweisung IdP-Dienst, Fachdienste)
Um dem Fachdienst die Gültigkeitsprüfung des "ID_TOKEN" zu ermöglichen, MUSS der Introspection-Endpunkt gegen Vorlage des "ID_TOKEN" Auskunft über die mit dem "ID_TOKEN" bestätigten Meta-Informationen erteilen. Das zu prüfende "ID_TOKEN" MUSS mit dem String "id_token" gekennzeichnet sein.
[<=]

A_19807 - Nur Fachdienste führen Token Introspection durch
(Zuweisung IdP-Dienst, Fachdienste)
Der Token Introspection-Endpunkt DARF Anfragen zur Introspection von "ID_TOKEN" NICHT von anderen als Fachdiensten "URI_FD" annehmen [rfc7662#section-4].
[<=]

Hinweis:

Zur Token Introspection berechtigt ist ausschließlich der Fachdienst als Empfänger des "ID_TOKEN". Eine Introspection durch das Anwendungsfrontend ist nicht möglich, da das "ID_TOKEN" mit dem öffentlichen Schlüssel des Fachdienstes "PUK_FD" verschlüsselt ist und somit auch nur der adressierte Fachdienst dieses entschlüsseln kann.

Hinweis: Prüfung von "REFRESH_TOKEN" am Introspection-Endpunkt [rfc7662#section-5]

Eine Prüfung der Gültigkeit durch das Anwendungsfrontend ist nicht möglich, da das "REFRESH_TOKEN" nur gegen ein "ID_TOKEN" eingetauscht werden kann. Das "REFRESH_TOKEN" ist durch den String "refresh_token" im Parameter "token_type" gekennzeichnet und kann nur vom Token-Endpunkt entschlüsselt werden, da es mit dessen "PUK_TOKEN" verschlüsselt wurde.

5.5.2 Token Introspection-Endpunkt Ausgangsdaten

Der Token Introspection-Endpunkt prüft nach dem Einreichen einer Token Introspektion-Anfrage seine Zuständigkeit.

A_19808 - Inhalte der Token Introspection Antwort
(Zuweisung IdP-Dienst, Fachdienste)
Nach einer erfolgreichen Prüfung eines "ID_TOKEN" MUSS der Token Introspection-Endpunkt dem anfragenden Fachdienst eine Token Introspection-Antwort geben [rfc7662#section-2.2].
[<=]

A_19798 - Speichern der Token Introspection Antwort (Token Introspection Response Caching)
(Zuweisung IdP-Dienst, Fachdienste)
Fachdienste SOLLEN die Antwort des Token Introspection-Endpunktes zwischenspeichern [rfc7662#section-2.2].
[<=]

A_19975 - Haltbarkeit der Token Introspection Antwort

Die Zwischenspeicherung DARF NICHT länger als die halbe Gültigkeitsdauer des "ID_TOKEN" betragen [rfc7662#section-2.2].
[<=]

Beispielberechnung:
Math.floor("iat"+("exp"-"iat")/2).

A_19799 - Die Token Introspection Antwort ist signiert
(Zuweisung IdP-Dienst)
Das Ergebnis der Token Introspection MUSS vom Token Introspection-Endpunkt mit dessen "PrK_INT" signiert werden, um die Integrität der Daten sicher zu stellen und deren Herkunft glaubhaft zu belegen [JWT introspection response # section-5].
[<=]

A_19800 - Die Token Introspection Antwort ist verschlüsselt
(Zuweisung IdP-Dienst, Fachdienste)
Der Token Introspection-Endpunkt MUSS die Introspection-Antwort vor dem Versand an den jeweiligen Fachdienst mit dessen öffentlichen Schlüssel "PUK_FD", wie im Discovery Document angeboten, verschlüsseln [JWT introspection response # section-5].
[<=]

A_19796 - Verwendung von Transport Layer Security (TLS) bei Token Introspection
(Zuweisung IdP-Dienst, Fachdienste)
Der Token Introspection-Endpunkt MUSS zusätzlich zu den anderen Schutzmaßnahmen (JWT-Verschlüsselung und JWT-Signatur), Transport Layer Security (TLS) gemäß [BCP195] und [gemSpec_Krypt] nutzen [JWT introspection response # section-8.2].
[<=]

5.6 Token Revocation-Endpunkt

"ID_TOKEN" und "REFRESH_TOKEN" sowie die zugrundeliegende "SUBJECT_SESSION" haben eingeschränkte Lebenszyklen von wenigen Minuten bis zu mehreren Stunden. Da es möglich sein kann, dass Token auf dem Transportweg gestohlen oder unberechtigt kopiert werden, muss es Möglichkeiten geben, die Gültigkeit vor dem natürlichen zeitlichen Ableben zu beenden. So muss es beispielsweise beim Verlust des mobilen Endgerätes die Möglichkeit geben, die mit dem Gerät etablierte "SUBJECT_SESSION" (Sicherheitsbeziehung) zu terminieren. Der IdP-Dienst stellt hierfür einen Token Revocation-Endpunkt bereit, welcher gemäß [RFC7009] reagiert.

5.6.1 Token Revocation-Endpunkt Eingangsdaten

Der Token Revocation-Endpunkt stellt einen sehr wichtigen Dienst bereit, mit dessen Hilfe es möglich ist, bereits ausgestellte noch aktive "ID_TOKEN" und ggf. "REFRESH_TOKEN" zu widerrufen. Wenngleich diese Schnittstelle äußerst selten verwendet wird, ist es umso wichtiger, dass sie in diesen Situationen einwandfrei reagiert und die erwarteten Maßnahmen standardkonform umsetzt.

Der Token Revocation-Endpunkt wird daher strikt nach den Vorgaben des aktuellen Standards (derzeit [RFC7009]) umgesetzt. Die Token Revocation ist frei übersetzt als Widerrufs-Anfrage zu verstehen, da der Widerruf nicht ungeprüft durchgeführt wird.

A_19793 - Gestaltung des Antrags auf Token-Widerruf (Token Revocation)

(Zuweisung IdP-Dienst, Fachdienste, Frontend)
Der Fachdienst oder das Anwendungsfrontend welches berechtigt ist, den Widerruf eines Token zu beantragen, MUSS diese Anfrage gemäß [RFC7009 # section-2.1] umsetzen.
[<=]

A_19794 - Mindestangaben für den Token Widerruf (Token Revocation minimal information)
(Zuweisung IdP-Dienst, Fachdienste, Frontend)
Der Token Revocation-Endpunkt DARF einen Token-Widerruf NICHT bearbeiten, wenn die Anfrage unvollständig ist. Die Anfrage MUSS das Token im Parameter "`token`" und den Hinweis auf den Token-Typ "`token_type_hint`" mit dem Wert ("`SUBJECT_SESSION`", "`ID_TOKEN`" oder "`REFRESH_TOKEN`") enthalten.[RFC7009 # section-4.1.2]
[<=]

A_19795 - Token-Widerrufsanfragen sind zu signieren
(Zuweisung IdP-Dienst, Fachdienste ggf. Frontend aber womit?)
Der Token Revocation-Endpunkt MUSS bei eingehenden Widerrufsanfragen die Herkunft und Integrität überprüfen. Der Token Revocation-Endpunkt DARF NICHT auf unsignierte Widerrufsanfragen durch Widerruf des Tokens reagieren. [RFC7009 # section-5]
[<=]

A_19792 - Widerruf des "REFRESH_TOKEN" ("refresh_token" Revocation)
(Zuweisung IdP-Dienst)
Der Widerruf des "REFRESH_TOKEN" "`REFRESH_TOKEN`" bezieht sich ausschließlich auf das "`REFRESH_TOKEN`" und führt nicht zum Widerruf des auf Basis des "`REFRESH_TOKEN`" ausgestellten "`ID_TOKEN`" [RFC7009 # section-2].
Der Revocation-Endpunkt MUSS sicherstellen, dass nur das "`REFRESH_TOKEN`" widerrufen wird, wenn der Widerrufsantrag sich auf das "`REFRESH_TOKEN`" bezieht [RFC7009 # section-2].
[<=]

A_19791 - Widerruf des "ID_TOKEN" ("id_token" Revocation)
(Zuweisung IdP-Dienst, Fachdienste)
Der Widerruf des "`ID_TOKEN`" bezieht sich ausschließlich auf das "`ID_TOKEN`" und führt nicht zum Widerruf des möglicherweise zugrundeliegenden "`REFRESH_TOKEN`".
Der Revocation-Endpunkt MUSS sicherstellen, dass nur das "`ID_TOKEN`" widerrufen wird, wenn der Widerrufsantrag sich auf das "`ID_TOKEN`" bezogen hat. [RFC7009 # section-2]
[<=]

A_19788 - Widerruf der "SUBJECT_SESSION" durch Authenticator
(Zuweisung IdP-Dienst, Frontend)
Wird vom Authenticator beim Revocation-Endpunkt ein Antrag auf Widerruf der "`SUBJECT_SESSION`" (Back-Channel Session Revocation) eingereicht, MUSS der Revocation-Endpunkt den Widerruf aller Token, welche auf dieser "`SUBJECT_SESSION`" basieren, durchführen. Der Widerruf MUSS vom Authenticator mit willentlichem "Logoff" durch den Nutzer oder Deinstallation des Authenticators initiiert werden. [RFC8417]
[<=]

A_20018 - Widerruf der "SUBJECT_SESSION" durch Fachdienste
Wird von einem Fachdienst beim Revocation-Endpunkt ein Antrag auf Widerruf der "`SUBJECT_SESSION`" (Back-Channel Session Revocation) eines Anwendungsfrontends eingereicht, MUSS der Revocation-Endpunkt den Widerruf aller Token, welche auf dieser "`SUBJECT_SESSION`" basieren, durchführen. Zudem MUSS der Revocation-Endpunkt den mit der "`SUBJECT_SESSION`" verknüpften Authenticator in seiner Datenbasis als gelöscht

markieren. [RFC8417]
[<=]

A_19789 - Widerruf der "SUBJECT_SESSION" durch Backchannel Revocation
Der Revocation-Endpunkt MUSS alle aktiven "ID_TOKEN" und "REFRESH_TOKEN" mit sofortiger Wirkung widerrufen und dem Token Introspection-Endpunkt die notwenigen Informationen zukommen lassen, wenn eine Backchannel-Revocation beantragt wird, damit widerrufene Token zukünftig als bereits widerrufen identifiziert werden. [RFC8417]
[<=]

A_19790 - Backchannel Revocation Information an Authorization-Endpunkt
Der Revocation-Endpunkt MUSS die "SUBJECT_SESSION" zwischen Authenticator und Authorization-Endpunkt terminieren, sodass ohne erneute Authentifizierung durch das Endgerät des Nutzers kein weiteres Token beantragt werden kann. [RFC8417]
[<=]

A_19786 - Verwendung widerrufener Token
(Zuweisung Fachdienste, Frontend)
Das Anwendungsfrontend, welches ein "ID_TOKEN" oder Refresh- Token widerrufen hat, DARF dieses Token NICHT erneut verwenden und MUSS dessen lokale Löschung sicherstellen.
[<=]

A_19787 - Verwendung terminierter "SUBJECT_SESSION"
(Zuweisung IdP-Dienst)
Hat ein Authenticator die "SUBJECT_SESSION" widerrufen, MUSS der Authorization-Endpunkt die weitere Beantragung von "ID_TOKEN" oder "REFRESH_TOKEN" basierend auf dieser widerrufenen "SUBJECT_SESSION" unterbinden.
[<=]

5.6.2 Token Revocation-Endpunkt Ausgangsdaten

Der Token Revocation-Endpunkt beantwortet die Widerrufsanträge gewissermaßen nicht. Das Ergebnis des Widerrufsantrags stellt sich in der Form dar, dass ein erfolgreicher Widerruf mit dem HTTP-Status-Code 200 beantwortet wird. Ebenso werden fehlerhafte Widerrufsanträge oder Anträge, welche sich auf nicht existente Token bzw. Sessions beziehen, mit dem HTTP-Status-Code 200 bedient.

Abweichend hiervon kann der Token Revocation-Endpunkt eine Fehlermeldung liefern, wenn das zu sperrende Token eine Sperrung nicht vorsieht oder die Berechtigung zum Widerruf nicht vorliegt. In solchen Fällen reagiert der Token Revocation-Endpunkt mit dem HTTP-Status-Code 503, woraus der Authenticator oder das Anwendungsfrontend schließen kann, dass das Token bzw. die "SUBJECT_SESSION" noch existent sind.

A_19784 - Rückmeldung des Status-Code der erfolgreichen Widerrufsumsetzung [RFC7009#section-2.2]
(Zuweisung IdP-Dienst, Frontend)
Der Token Revocation-Endpunkt MUSS den vorgebrachten Sperrantrag mit HTTP-Status-Code 200 beantworten, wenn der Widerruf durchgeführt wurde oder der Widerrufsantrag fehlerhaft oder unvollständig war.
[<=]

A_19783 - Rückmeldung des Status-Code der nicht erfolgten Widerrufsumsetzung
Der Revocation-Endpunkt MUSS eine nicht erfolgte Umsetzung eines Widerrufsantrags mit dem HTTP-Status-Code 503 beantworten.
Die Rückmeldung kann um den Hinweis "`unsupported_token_type`" oder im Falle einer Verzögerung mit "`Retry-After`" ergänzt werden.
[<=]

5.7 Userinfo-Endpunkt

Der Userinfo-Endpunkt ist eine Basis-Schnittstelle des JSON-Web-Token-Standards und bietet Informationen über den Nutzer des Tokens an. Hier kann der angesprochene Fachdienst gegen Vorlage des "`ID_TOKEN`" im get-Request in Erfahrung bringen, welche Daten im Zusammenhang mit dem "`ID_TOKEN`" im "`CONSENT`" bestätigt wurden.

Mögliche Inhalte eines Standard Claims und somit der Rahmen der im "`CONSENT`" zu bestätigten Informationen sind die durch das jeweilige Identifikationsmittel bereitgestellten Informationen.

Bei der elektronischen Gesundheitskarte (eGK) gehen die Informationen die im Consent bestätigt werden können aus [gemSpec_PKI#5.1.3.1 C.CH.AUT und C.CH.AUT_ALT – Authentisierung eGK] hervor.

Beim elektronischen Heilberufsausweis (eHBA) ergeben sich diese Informationen aus der Spezifikation [gemSpec_PKI#5.2.1.1 C.HP.AUT – Authentisierung HBA].

Bei der Verwendung einer Secure Module Card eines Leistungserbringers (SMC-B) ergibt sich der Umfang der Informationen aus [gemSpec_PKI#5.3.4 X.509 Zertifikatsprofile der SMC-B] hier genauer der Profiltyp C.HCI.AUT (gemäßTab_PKI_238).

Andere als die aus den Zertifikaten hervorgehende Informationen über den Nutzer kann und darf der Userinfo-Endpunkt nicht preisgeben, da deren Herkunft nicht nachweisbar ist.

A_19782 - Informationen am Userinfo-Endpunkt
(Zuweisung IdP-Dienst, Fachdienste)
Der Userinfo-Endpunkt gibt nur solche Informationen preis, welche nachweislich aus einem durch einen zugelassenen TSP der Telematikinfrastruktur (TI) herausgegebenen Authentisierungszertifikat hervorgehen. Weitere Informationen, abgesehen von META-Informationen,DARF der Userinfo-Endpunkt NICHT preisgeben.
[<=]

6 Anhang A – Verzeichnisse

6.1 Abkürzungen

Kürzel	Erläuterung

6.2 Glossar

Begriff	Erläuterung
Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.

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

6.3 Abbildungsverzeichnis

6.4 Tabellenverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

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 Telematikinfrastruktur. Der mit der vorliegenden Version korrelierende Entwicklungsstand dieser Konzepte und

Spezifikationen wird pro Release in einer Dokumentenlandkarte definiert; Version und Stand der referenzierten Dokumente sind daher in der nachfolgenden Tabelle nicht aufgeführt. Deren zu diesem Dokument jeweils gültige Versionsnummern sind in der aktuellen, von der gematik veröffentlichten Dokumentenlandkarte enthalten, in der die vorliegende Version aufgeführt wird.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Einführung der Gesundheitskarte – Glossar

6.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
HEART I	openid-heart-openid-connect-1_0 [https://openid.net/specs/openid-heart-openid-connect-1_0-2017-05-31.html] (Stand: 03.10.2016)
HEART II	Health Relationship Trust Profile for OAuth 2.0 [https://openid.net/specs/openid-heart-oauth2-1_0.html] (Stand: 08.07.2018)
OpenID Connect Core	OpenID Connect Core 1.0 [https://openid.net/specs/openid-connect-core-1_0.html] (Stand: 08.11.2014)
connect2id	[https://connect2id.com/]
RFC3986	URI
RFC7009	JSON REVOCATION
RFC7165	JOSE
RFC7231	HTTP
RFC7515	JWS JSON SIGNATURE
RFC7516	JWE JSON ENCRYPTION
RFC7517	JWK JSON KEY

RFC7518	JWE JSON ALGORITHM
RFC7519	JWT JSON WEB TOKEN
RFC7520	JOSE Protection
RFC7521	Assertion Authorization
RFC7522	Assertion SAML 2.0
RFC7523	JSON Token Profile
RFC6749	Oauth2
RFC7591	Dynamic Registration
RFC6750	Oauth2 Bearer
RFC7636	Oauth Proof Key for Public Client
RFC7662	OAuth Token Introspection
RFC8417	Security Event Token
CAB-Forum	Liste vertrauenswürdiger Zertifikatsherausgeber (Root-CAs) für Anwendungen im Internet https://cabforum.org/members/

1498