Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation
Identity Provider-Dienst

Version:	1.~~1.1~~2.0
Revision:	~~308662~~328333
Stand:	~~18.12.2020~~19.02.2021
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	gemSpec_IDP_Dienst

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	30.06.20		initiale Erstellung des Dokuments	gematik
1.1.0	12.10.20		Einarbeitung Scope-Themen aus R4.0.1	gematik
1.~~2.0~~1.1	~~18.12~~13.11.20		Einarbeitung P22.4	gematik
1.2.0	19.02.21		Einarbeitung P22.5	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die Anforderungen zu Herstellung, Test und Betrieb des Produkttyps Identity Provider (IdP)-Dienst. Der IdP-Dienst 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 vom Anwendungsfrontend für eine Authentifizierung eines Nutzers anhand einer Smartcard genutzt. Diese Authentifikation ist die Voraussetzung, damit ein Anwendungsfrontend Zugang zu Fachdaten eines Fachdienstes erlangen kann. Der IdP-Dienst verwaltet und steuert den Authentifizierungsprozess für das E-Rezept und perspektivisch auch für weitere Anwendungen.

1.2 Zielgruppe

Das Dokument richtet sich an Hersteller und Anbieter von Identity Providern, welche die Funktionen des IdP-Dienstes der gematik realisieren wollen.

1.3 Geltungsbereich

Dieses Dokument enthält normative Festlegungen zur Telematikinfrastruktur (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. Dokumentenlandkarte, Produkttypsteckbrief, Leistungsbeschreibung) festgelegt 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

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 Transport Layer Security (TLS)-

Sicherung über zentrale Plattformdienste der TI beziehen und diese dort auch geprüft werden können.

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.

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.

Hinweise auf offene Punkte

Offene Punkten werden im Dokument in dieser Darstellung ausgewiesen.

2 Systemüberblick

Im Rahmen der kontinuierlichen Erweiterung der Vorgaben für Identity Provider innerhalb der TI werden die Vorgaben weiter angepasst werden. Dies beinhaltet Festlegungen zur Einführung föderierter Identity Provider, die Unterstützung weiterer Anwendungen und Nutzungsszenarien, Vorgaben für zulässige Authentisierungsverfahren, Schnittstellen für die Inter-App-Kommunikation zu einer getrennten Authenticator-Anwendung sowie die mögliche Einführung weiterer Endpunkte entsprechend [openid-connect-core].

2.1 Allgemeiner Überblick

In der Telematikinfrastruktur werden zahlreiche Fachdienste angeboten. Anwendungsfrontends können über die Authentifizierung des Nutzers am IdP-Dienst Zugriff zu den von den Fachdiensten angebotenen Daten erhalten. Der IdP-Dienst stellt durch gesicherte JSON Web Token (JWT) attestierte Identitäten aus. Gegen Vorlage eines "ACCESS_TOKEN" erhalten Anwendungsfrontends – entsprechend der im Token attestierten professionOID – Zugriff auf die Inhalte der Fachdienste.

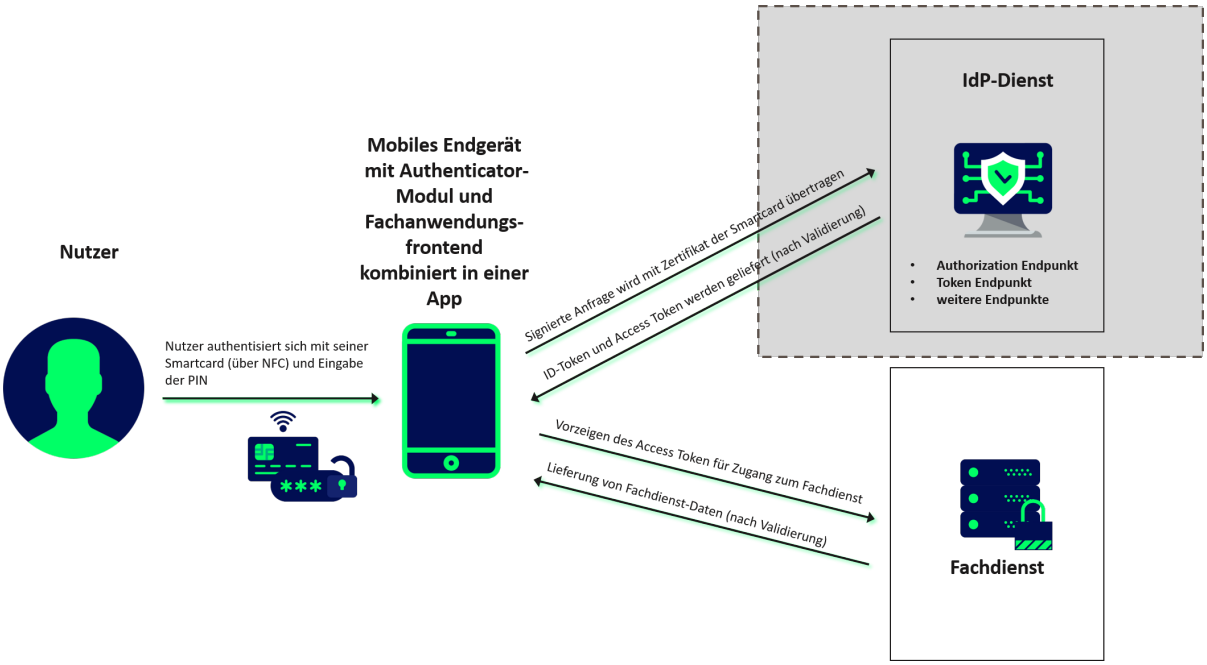

Abbildung 1: Systemüberblick (vereinfacht)

Die obige Abbildung stellt den Systemüberblick dar. Der Authentifizierungsprozess, welcher mit der Ausstellung und Übergabe der Token an das Anwendungsfrontend endet, wird dabei zur besseren Übersicht vereinfacht dargestellt.

Der IdP-Dienst übernimmt für den Fachdienst die Aufgabe der Identifikation des Nutzers. Der IdP-Dienst fasst die professionOID sowie weitere für den Fachdienst notwendige Attribute in signierten JSON Web Token ("ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN") zusammen. Fachdienste müssen keine Überprüfung des Nutzers selbst implementieren, sondern können sich darauf verlassen, dass der Besitzer des bei ihnen vorgetragenen "ACCESS_TOKEN" bereits identifiziert wurde. Des Weiteren stellt der IdP-Dienst sicher, dass die vom Nutzer vorgetragenen Attribute (aus dem Signaturzertifikat) gültig sind.

Der IdP-Dienst prüft, ob das vorgetragene X.509-nonQES-Signatur-Zertifikat der verwendeten Prozessor-Chipkarte (eGK, HBA oder SMC-B) für die vorgesehene Laufzeit des Tokens zeitlich gültig und ob dessen Integrität sichergestellt ist.

Der IdP-Dienst stellt nur solche "ACCESS_TOKEN" aus, welche auf gültigen AUT-Zertifikaten (d.h. C.CH.AUT, C.HP.AUT oder C.HCI.AUT) basieren.

2.2 Detaillierter Überblick

Der IdP-Dienst führt die Identifikation des Nutzers durch und stattet diesen mit einem "ID_TOKEN" gemäß [openid-connect-core 1.0 # IDToken], einem "ACCESS_TOKEN" gemäß [RFC6749 # section-1.4] und einem "SSO_TOKEN" basierend auf [RFC7519] aus. Gewählt wird aus Sicherheitsaspekten der "Authorization Code Grant" gemäß [RFC6749 # section-4.1]. Die Verwendung von PKCE (Proof Key for Code Exchange by OAuth Public Clients) gemäß [RFC7636] wird gefordert.

Der IdP-Dienst teilt sich in mehrere Teildienste auf. Einzelne Teildienste werden zentral und bei Bedarf auf unterschiedlicher Hardware verteilt betrieben. Das Authenticator-Modul wird grundsätzlich auf dezentraler Hardware zusammen mit dem Primärsystem oder auf dem mobilen Endgerät des Nutzers betrieben.Der IdP-Dienst stellt unterschiedliche Endpunkte bereit, welche eine statische IP-Adressierung und somit statische URI besitzen. Diese statisch adressierten Endpunkte umfassen:

- Discovery-Endpunkt ("OAuth 2.0 Authorization Server Metadata" [RFC8414])
- Redirection-Endpunkt (Teil des "The OAuth 2.0 Authorization Framework" [RFC6749 # section-3.1.2])
- Authorization-Endpunkt (Teil des "The OAuth 2.0 Authorization Framework" [RFC6749])
- Token-Endpunkt ([RFC6749 # section-3.2])
  Teildienst 1 "ID_TOKEN" ([openid-connect-core 1.0 # IDToken])
  Teildienst 2 "ACCESS_TOKEN" ([RFC6749 # section-1.4 & RFC6749 # section-5])
  Teildienst 3 "SSO_TOKEN" ([RFC7519]).

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

Teildienste wie das Authenticator-Modul und das 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.

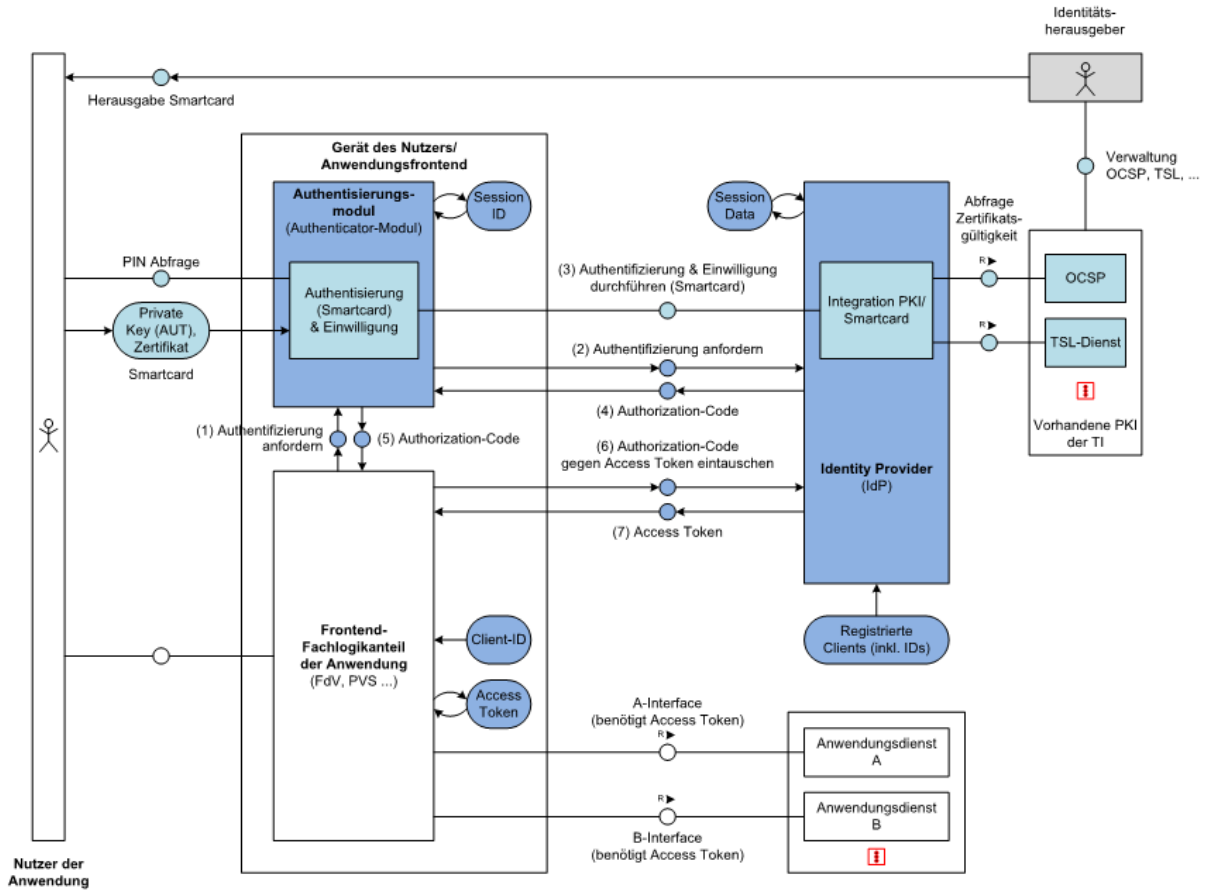

Abbildung 2: Übersichtsschaubild OAuth2.0 Smartcard-IdP-Dienst

Erläuterungen zur obigen Abbildung:
Die Teilschritte (1) und (5) werden bei mobilen Endgeräten (FdV) via Redirection-Endpunkt [RFC6749 # section-3.1.2] realisiert.
Die Teilschritte (1) und (5) können bei Primärsystemen (PVS, AVS, KVS) abweichend von [RFC6749 # section-9] behandelt werden.
Die Teilschritte (2) und (4) werden durch den Authorization-Endpunkt gemäß [RFC6749 # section-3.1] bedient. Der Teilschritt (3) Challenge-Response wird durch den Authorizatio-Endpunkt bedient.
Die Teilschritte (6) und (7) werden durch den Token-Endpunkt [RFC6749 # section-3.2] bedient.

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 Änderung der Zugangsmechanismen realisieren müssen.

Die Umsetzung basiert grundsätzlich auf [OpenID Connect Core v1.0] und [OpenID Connect Discovery v1.0].

Weitere zu beachtende Standards sind die folgenden:
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 [RFC7033] sowie OAuth 2.0 Bearer [RFC6750], OAuth 2.0 Assertion [RFC7521], OAuth 2.0 JWT Profile [RFC7523], OAuth 2.0 Responses [RFC6749].

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

3 Systemkontext

Die untere Abbildung beschreibt den Systemkontext aus Sicht des IdP-Dienstes. Das Authenticator-Modul liefert die Daten zur Authentifizierung des Nutzers an den IdP-Dienst. Bei positiver Validierung – gegen den OCSP/TSL-Dienst der Public Key Infrastructure (PKI) der gematik – liefert der IdP-Dienst einen "AUTHORIZATION_CODE" zurück. Der IdP-Dienst liefert ebenso einen "SSO_TOKEN", wodurch das Authenticator-Modul einen weiteren "AUTHORIZATION_CODE" ohne erneute Nutzerauthentifizierung erhalten kann.

Das Anwendungsfrontend registriert sich innerhalb eines organisatorischen Prozesses am IdP-Dienst. Das Anwendungsfrontend erlangt gegen Vorlage des "AUTHORIZATION_CODE" einen "ID_TOKEN" und einen "ACCESS_TOKEN". Das Anwendungsfrontend erhält gegen Vorlage des "ACCESS_TOKEN" Zugang zu den Fachdaten des Fachdienstes.

Der Fachdienst registriert sich am IdP-Dienst in Form eines organisatorischen Prozesses.

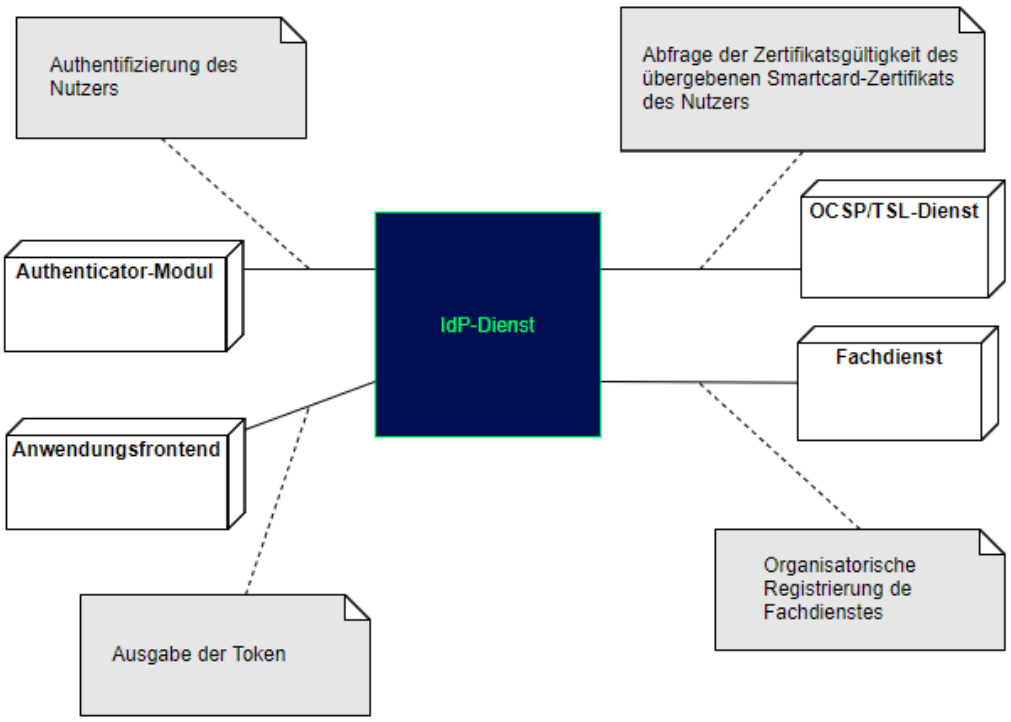

Abbildung 3: Systemkontext aus Sicht des IdP-Dienstes

3.1 Akteure und Rollen

Im Systemkontext des IdP-Dienstes interagieren verschiedene Akteure (Nutzer und aktive Komponenten) in unterschiedlichen OAuth2-Rollen gemäß ~~[RFC6749 # section-1.1~~[RFC6749 # section-1.1].

Tabelle 1: TAB_IDP_DIENST_0001 Akteure und OAuth2-Rollen

Akteur	OAuth2-Rolle
Nutzer	Resource Owner
Fachdienst	Resource Server
Anwendungsfrontend	Teil des Clients
Authenticator-Modul	Teil des Clients
IdP-Dienst	Authorization Server
Fachdaten	Protected Resource

Nutzer (Rolle: Resource Owner)

Der Resource Owner ist der Nutzer, welcher auf die beim Fachdienst (Resource Server) für ihn bereitgestellten Daten (Protected Resource) zugreift.

Der Resource Owner verfügt über die folgenden Komponenten:

- Endgerät des Nutzers
- Authenticator-Modul
- Anwendungsfrontend

Fachdienst (Rolle: Resource Server)

Der Resource Server ist der Fachdienst, der dem Nutzer (Resource Owner) Zugriff auf seine Fachdaten (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 Owners.

Anwendungsfrontend/Authenticator-Modul kombiniert in einer Applikation (Rolle: Client)

Der Client greift mit dem Authenticator-Modul und dem Anwendungsfrontend (OIDC Relying Party bzw. OAuth2 Client) 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) ausgeführt werden.

IdP-Dienst (Rolle: Authorization Server)

Der Authorization Server authentifiziert den Resource Owner (Nutzer) und stellt "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN" für den vom Resource Owner erlaubten Anwendungsbereich (SCOPE) aus, welche dieser wiederum beim Fachdienst einreicht.

Tabelle 2: TAB_IDP_DIENST_0002 Kurzbezeichnung der Schnisttstellen des IdP-Dienstes

Kurzzeichen	Schnittstelle
AUTH	Authorization-Endpunkt
TOKEN	Token-Endpunkt
REDIR	Redirection-Endpunkt
DD	Discovery Document-Endpunkt

Weitere Akteure im Kontext IdP-Dienst sind:

Fachdaten (Rolle: Protected Resource)

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

3.2 Begriffsdefinition

Die folgende Tabelle enthält die Abkürzungen (für die privaten Schlüssel PrK und für öffentliche Schlüssel PUK) der verschiedenen Endpunkte des IdP-Dienstes und deren Verwendung.

Tabelle 3: TAB_IDP_DIENST_0003 Bezeichnungen der Schlüssel und deren URI

	PUK	URI PUK	private Key	URI Dienst
Authorization Server				URI_DD
Authorization-Endpunkt (AUTH)	PUK_AUTH - für die Signaturprüfung des "AUTHORIZATION_CODE" und des "SSO_TOKEN"	PUK_URI_AUTH	PrK_AUTH - zum Signieren des "AUTHORIZATION_CODE" und des "SSO_TOKEN"	URI_AUTH
Discovery-Endpunkt (DISC)	PUK_DISC - für die Signaturprüfung des Discovery Document	PUK_URI_DISC	PrK_DISC - zum Signieren des Discovery Document	URI_DISC

Token-Endpunkt (TOKEN)	PUK_TOKEN - für die Signaturprüfung des "ID_TOKEN" und des "ACCESS_TOKEN"	PUK_URI_TOKEN	PrK_TOKEN - zum Signieren des "ID_TOKEN" und des "ACCESS_TOKEN"	URI_TOKEN
Fachdienst (FD)	PUK_FD - für die Verschlüsselung des "ACCESS_TOKEN"	PUK_URI_FD	PrK_FD - für die Entschlüsselung des "ACCESS_TOKEN"	URI_FD

Hinweis: Werden alle Teildienste auf einem Server gemeinsam betrieben, so können diese dasselbe Schlüsselmaterial verwenden. Werden Teildienste auf unterschiedlichen physischen oder logischen Servern betrieben, so sind die Endpunkte mit eigenem Schlüsselmaterial auszustatten.

Die URL des Discovery Documents "URI_DD" stellt somit den zentralen Anlaufpunkt dar, anhand dessen alle weiteren „statischen" Dienste (Endpunkte des IdP-Dienstes und der Fachdienste) adressiert werden können.

3.3 Verfahrensbeschreibung

Vorbereitende Maßnahmen: Das Anwendungsfrontend und der Fachdienst haben sich im Zuge eines organisatorischen Prozesses beim IdP-Dienst registriert. Das Anwendungsfrontend und das Authenticator-Modul haben das Discovery Dokument eingelesen und kennen damit die Uniform Resource Identifier (URI) und die öffentlichen Schlüssel der vom IdP-Dienst angebotenen Endpunkte. Der Fachdienst hat bei der Registrierung am IdP-Dienst seinen öffentlichen Schlüssel hinterlegt.

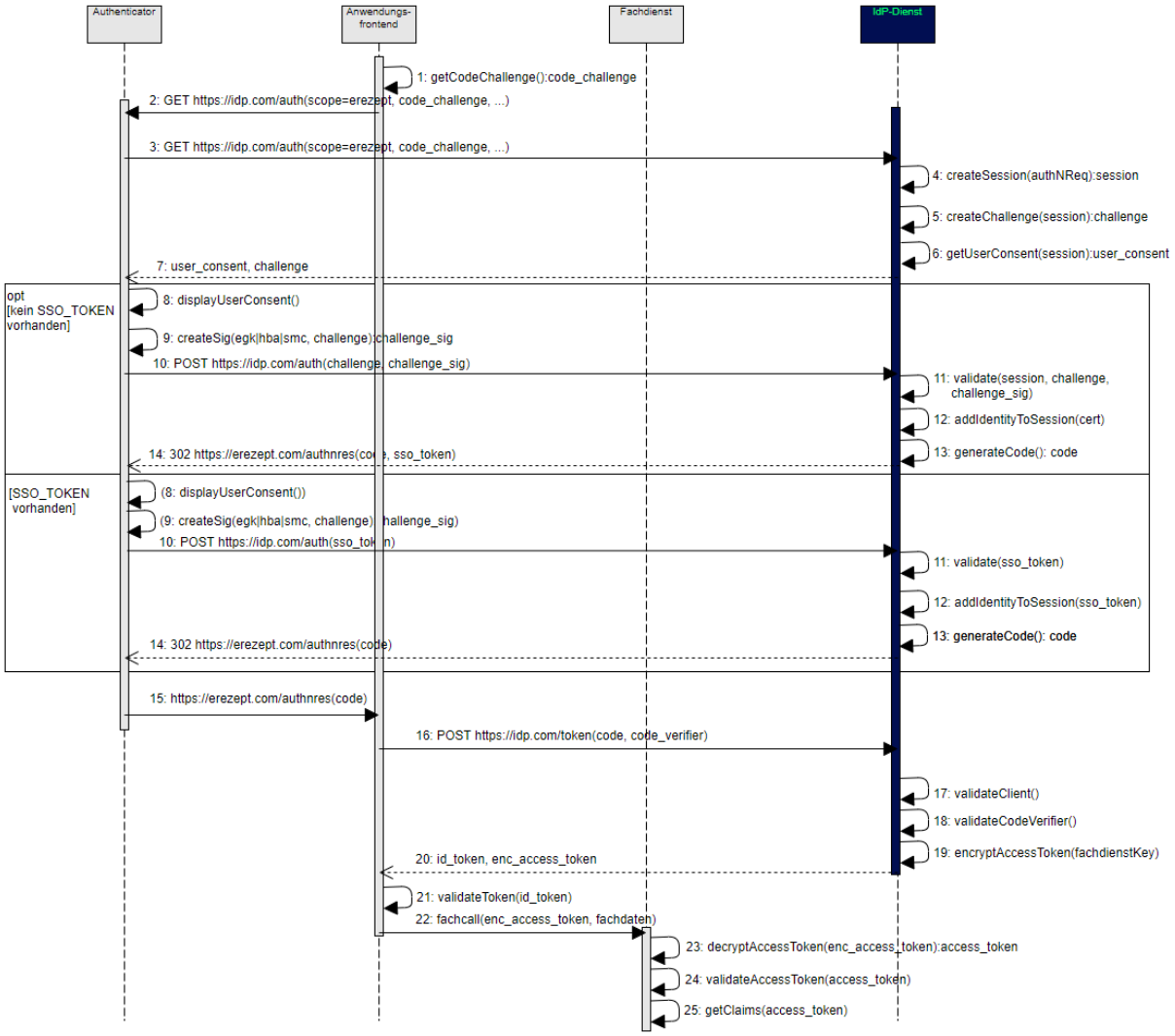

Abbildung 4: Datenfluss-Diagramm IdP-Dienst

Die Prozessschritte, welche notwendig sind, damit ein mobiles Anwendungsfrontend einen Token erhält sind:

1. Das Anwendungsfrontend erzeugt sich einen "CODE_VERFIER" [RFC7636 # section-4.1] und bildet darüber den Hash "CODE_CHALLENGE" mit dem Hash-Algorithmus S256 gemäß [RFC 7636 # section-4.2].
2. Das Anwendungsfrontend überträgt die "CODE_CHALLENGE" gemäß [RFC8252 # Anhang B] an das Authenticator-Modul.
3. Das Authenticator-Modul überträgt die "CODE_CHALLENGE" mit der genutzten "code_challenge_method" S256 weiter an den Authorization-Endpunkt des IdP-Dienst.
4. Der Authorization-Endpunkt legt eine "SESSION_ID" an und speichert alle Informationen zum Vorgang in der "CHALLENGE".
5. Der Authorization-Endpunkt stellt alle Informationen zusammen und erzeugt die "CHALLENGE".

6. Der Authorization-Endpunkt stellt den mit dem entsprechenden Fachdienstes vereinbarten Consent (Zustimmung des Nutzers zur Verarbeitung der angezeigten Daten) zusammen.
7. Der Authorization-Endpunkt überträgt "CHALLENGE" und Consent-Abfrage "USER_CONSENT" zum Authenticator-Modul.
8. Das Authenticator-Modul fordert den Nutzer zu Consent-Freigabe auf mittels Smartcard und PIN-Eingabe. Falls bereits ein "SSO_TOKEN" beim Authenticator-Modul existiert, entfällt dieser Schritt.
9. Das Authenticator-Modul verwendet die PIN um die "CHALLENGE" von der Smartcard signieren zu lassen. Falls bereits ein "SSO_TOKEN" beim Authenticator-Modul existiert, entfällt dieser Schritt.
10. Das Authenticator-Modul überträgt "CHALLENGE" mit dem Smartcard-Zertifikat an den IdP-Dienst (Antwort Schritt 7). Falls ein "SSO_TOKEN" beim Authenticator-Modul existiert, wird diese Token anstatt der "Challenge" zum IdP-Dienst transportiert.
11. Der Authorization-Endpunkt validiert die "SESSION_ID", "CHALLENGE" und "SIGNATUR". Die Signatur wird anhand des im "x5c"-Header mitgelieferten Authentifizierungszertifikats der Smartcard validiert. Falls ein "SSO_TOKEN" angenommen wurde, wird dieses validiert. Entschlüsselt wird das "SSO_TOKEN" vom Authorization-Endpunkt mit seinem Schlüsselmaterial, welches er zur Verschlüsselung genutzt hat. Die Überprüfung der Signatur des "SSO_TOKEN" führt der Authorization-Endpunkt anhand seines öffentlichen Schlüssels "PUK_AUTH" durch.
12. Der Authorization-Endpunkt verknüpft die "SESSION_ID" mit der Identität aus der Signatur. Falls ein "SSO_TOKEN" angenommen wurde, verknüpft der Authorization-Endpunkt die "SESSION_ID" mit der Identität aus dem "SSO_TOKEN".
13. Der Authorization-Endpunkt erstellt den "AUTHORIZATION_CODE".
14. Der Authorization-Endpunkt überträgt den "AUTHORIZATION_CODE" und den "SSO_TOKEN" an das Authenticator-Modul (Antwort Schritt 3). Falls das das Authenticator-Modul ein vorhandenes "SSO_TOKEN" an den Authorization-Endpunkt zur Erlangung eines "AUTHORIZATION_CODE" geschickt hat, wird kein neues "SSO_TOKEN" vom Authorization -Endpunkt erstellt und verschickt. Der "AUTHORIZATION_CODE" und das "SSO_TOKEN" werden vom Authorization-Endpunkt mit seinem privaten Schlüssel "PrK_AUTH" signiert. Der Authorization-Endpunkt verschlüsselt das "SSO_TOKEN" für sich. Das zur Verschlüsselung verwendete Schlüsselmaterial muss die Vorgaben der [gemSpec_Krypt] beachten.
15. Das Authenticator-Modul überträgt den "AUTHORIZATION_CODE" an das Anwendungsfrontend (Antwort Schritt 2).
16. Das Anwendungsfrontend sendet "CODE_VERFIER" und "AUTHORIZATION_CODE" zum Token-Endpunkt des IDP-Dienstes.
17. Der Token-Endpunkt validiert den Client ("SESSION_ID" in der signierten "CHALLENGE"). Der Token-Endpunkt validiert den "AUTHORIZATION_CODE" anhand des öffentlichen Schlüssels "PUK_AUTH" des Authorization-Endpunktes.
18. Der Token-Endpunkt validiert den "CODE_VERFIER" und gleicht diesen mit der "CODE_CHALLENGE" ab.

19. Der Token-Endpunkt erzeugt die erforderlichen Token, signiert die Token mit seinem privaten Schlüssel "PrK_TOKEN" und verschlüsselt das"ACCESS_TOKEN" mit dem öffentlichen Schlüssel "PUK_FD" des angeforderten Fachdienstes.
20. Der Token-Endpunkt überträgt die Token an das Anwendungsfrontend (Antwort Schritt 16).
21. Das Anwendungsfrontend prüft die Token-Signatur anhand des öffentlichen Schlüssels "PUK_TOKEN" des Token-Endpunktes.
22. Das Anwendungsfrontend reicht das gültige "ACCESS_TOKEN" beim Fachdienst ein.
23. Der Fachdienst entschlüsselt das "ACCESS_TOKEN" mit seinem privaten Schlüssel "PrK_FD".
24. Der Fachdienst validiert das "ACCESS_TOKEN" anhand des öffentlichen Schlüssels "PUK_TOKEN" des Token-Endpunktes.
25. Der Fachdienst zieht die Claims (d. h. die Key/Value-Paare im Payload eines Tokens) aus dem "ACCESS_TOKEN" und gibt bei positiver Validierung den Zugriff auf die Fachdaten frei.

Hinweis: Verwendet der Nutzer ein Primärsystem, führt der Konnektor in in Schritt 9 die Funktion "externalAuthenticate" für eine Signatur mit der SMC-B durch. Setzt der Nutzer ein mobiles Endgerät ein, ruft das Authenticator-Modul die Signaturfunktion eines HBA oder einer eGK für eine nonQES-Signatur der Smartcard auf. Die erforderlichen Token in Schritt 19 sind "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN". Das Authenticator-Modul kann mit dem "SSO_TOKEN" einen neuen "AUTHORIZATION_CODE" beim IdP-Dienst ohne erneute Nutzer-Authentifizierung anfordern und damit ein neues "ACCESS_TOKEN" vom IdP-Dienst erhalten. Im"SSO_TOKEN" hinterlegt der IdP-Dienst die für ihn selbst bestimmten Informationen zum gesamten Vorgang, sodass er keine schützenswerten Informationen zentral speichern muss. Das "SSO_TOKEN" beinhaltet alle Daten, die beim IdP-Dienst benötigt werden, um auf die Vorgangshistorie zurückzugreifen und ggf. neue "ACCESS_TOKEN" herauszugeben. Die Informationen im "SSO_TOKEN" sind mit dem öffentlichen Schlüssel des Authorization-Servers für diesen selbst verschlüsselt und können ausschließlich mit dem privaten Schlüssel des Authorization Servers wieder entschlüsselt werden.

Im Schaubild Datenflussdiagramm IdP-Dienst oben ist der Datenfluss zwischen Anwendungsfrontend, Authenticator-Modul, IdP-Dienst und Fachdienst dargestellt. Der Datenfluss weicht im Falle von Primärsystemen hiervon ab, wenngleich Primärsysteme ebenfalls Nutzer-Endgeräte sind. Die Abweichung des Datenflusses wird im nächsten Kapitel erläutert.

3.4 Abweichende Verfahrensbeschreibung für Primärsysteme

Da bei Primärsystemen der Zugriff auf das Authenticator-Modul nicht in allen Fällen in Form eines Links innerhalb des Systems erfolgen kann, muss von der Vorgehensweise für mobile Endgeräte des Nutzers abgewichen werden. Das Primärsystem hat nicht die Möglichkeit, die Anfrage zum freizugebenden Consent anzuzeigen und nicht zur Eingabe der PIN aufzufordern. Zum Betrieb des Primärsystems ist es notwendig, dass sich die SMC-B im freigeschalteten Modus befindet. Damit muss die Freischaltung der SMC-B genutzt werden, um die Consent-Freigabe dauerhaft zu bestätigen und die vom IdP-Dienst in Schritt 6 geforderte Challenge ohne PIN-Eingabe zu realisieren.

Für die Signatur der Challenge wird die Funktion "externalAuthenticate" des Konnektors verwendet, welcher diesen Funktionsaufruf nur von den Primärsystemen entgegennimmt, an welchen ein Mitarbeiter der Praxis aktiv eingeloggt ist.

3.5 Registrierung Anwendungsfrontend und Fachdienst

Um ein Anwendungsfrontend nutzen zu können, muss dieses gemeinsam mit einem Authenticator-Modul in einer Applikation kombiniert und am IdP-Dienst registriert sein. Die Registrierung des Anwendungsfrontends ist im Dokument [gemSpec_IDP_Frontend] beschrieben.

Anbieter von Fachdiensten müssen Ihre Fachdienste über einen organisatorischen Prozess am IdP-Dienst durchführen.

A_20737 - Ermöglichung einer organisatorischen Registrierung für Anwendungsfrontends und Fachdienste
Der Anbieter des IdP-Dienstes MUSS eine organisatorische Registrierung von Anwendungsfrontends und Fachdiensten ermöglichen.**[<=]**

Ergänzung: Diese Registrierung erfolgt einmalig für die Anwendung bzw. den Dienst und muss nicht bei Updates wiederholt werden. Die Registrierung des Fachdienstes beinhaltet dabei auch die Abstimmung der Claims und die Gültigkeitsdauer der erstellten Token (siehe [gemSpec_IDP_FD#Kapitel 4]), wobei der Fachdienst seinen Bedarf an den gewünschten Attributen erklärt. Anpassungen an den Claims bedürfen einer erneuten Abstimmung und Registrierung.

3.6 Anwendungsfrontend vorbereitende Maßnahmen

Das Anwendungsfrontend muss ein "CODE_VERIFIER" (Zufallswert) gemäß [RFC7636 # section-4.1] und hierüber einen Hash, die "CODE_CHALLENGE", gemäß [RFC7636 # section-4.2] mit dem Algorithmus S256gemäß [RFC7636 # section-4.2] erzeugen.

3.7 Anfrage eines ACCESS_TOKEN

Die folgende Anfrage an den Authorization-Endpunkt umfasst die Schritte 1-3 aus dem Gesamtablauf des Kapitels 3.2. Der Nutzer ruft sein Anwendungsfrontend auf. Die Addressierung des IdP-Dienstes ist im Anwendungsfrontend als Parameter in einer Konfigurationsdatei oder direkt im Quellcode hinterlegt.

Das Anwendungsfrontend liefert seine Anfrage auf ein "ACCESS_TOKEN" über das Authenticator-Modul an den Authorization-Endpunkt.

Inhalt der Anfrage ist:

- die "REDIRECT_URI" sowie Bezeichnung des aufzurufenden Fachdienstes,
- die eigene Hersteller-ID, Programmkürzel und Versionsnummer,
- der über das eigene "CODE_VERIFIER" [RFC7636 # section-4.1] gebildete HASH "code_challenge" [RFC7636 # section-4.2] mit Angabe des Algorithmus "code_challenge_method" [RFC7636 # section-4.3],

- der "STATE"-Parameter [RFC8252 # section-8.9] wird genutzt, um CSRF (Cross-Site-Request-Forgery) zu verhindern.

3.8 Aufgaben des Authorization-Endpunktes

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

3.8.1 Unzureichende Attribute für das Claim

Kann das Claim nicht voll bedient werden, gibt der Authorization-Endpunkt eine Fehlermeldung gemäß [RFC6749 # section-5.2] und fordert den Nutzer zur erneuten Authentisierung und Freigabe der erforderlichen Attribute auf.

3.8.2 Erstellung des AUTHORIZATION_CODE

Sind alle im Claim geforderten Attribute vorhanden und die Gültigkeit der Attribute geprüft, erstellt der Authorization-Endpunkt einen "AUTHORIZATION_CODE" und sendet diesen an das Anwendungsfrontend. Der Authorization-Endpunkt prüft die Signatur der "CHALLENGE" und das mitgelieferte Zertifikat der Smartcard des Nutzers gegen den OCSP/TSL-Dienst der PKI der gematik.

3.9 Einreichen des AUTHORIZATION_CODE

Das Anwendungsfrontend reicht den "AUTHORIZATION_CODE" zusammen mit dem "CODE_VERIFIER" beim Token-Endpunkt ein.

3.10 Aufgabe des Token-Endpunktes

Der Token-Endpunkt des IdP-Dienstes nimmt die Daten des Anwendungsfrontends entgegen und prüft neben deren Integrität, ob der eingereichte "CODE_VERIFIER" bei Nutzung des Hash-Verfahrens S256 (nach [RFC7636 # section-4.2]) zum bitgleichen Hash-Wert führt. Stimmt der Hash-Werte aus dem initialen Aufruf des Authenticator-Moduls - 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" und das "ACCESS_TOKEN" an das Anwendungsfrontend heraus.

3.11 Einreichen des "ACCESS_TOKEN" beim Fachdienst

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

3.12 Aufgabe des Fachdienstes

Der Fachdienst nimmt das "ACCESS_TOKEN" entgegen. Der Fachdienst muss das "ACCESS_TOKEN" mit seinem privaten Schlüssel "PrK_FD" entschlüsseln. Danach überprüft er die Integrität und die Übereinstimmung mit dem eigenen Claim. Enthält das "ACCESS_TOKEN" mehr oder weniger Attribute, als im Claim vereinbart, oder sind diese fehlerhaft oder nicht befüllt, stimmt die Integrität oder Signatur des "ACCESS_TOKEN" nicht oder ist das "ACCESS_TOKEN" zeitlich nicht mehr gültig, bricht der Fachdienst die Kommunikation mit einer dem Abbruchgrund entsprechenden Fehlermeldung ab.

Bei positiver Validierung gewährt der Fachdienst Zugriff auf seine Fachdaten.

4 Zerlegung des Produkttyps

Der Produkttyp besteht aus einer zentralen Komponente (IdP-Dienst). Diese wird bei der Durchführung des Authentifizierungsprozesses vom Authenticator-Modul unterstützt. Das Authenticator-Modul übernimmt die Ausführung der Nutzerauthentisierung. Bei Verwendung eines stationären Endgerätes mit installiertem Primärsystem, realisiert das Primärsystem die Funktionalität des Authenticator-Moduls. Das Anwendungsfrontend ist ebenso als Teil des Primärsystems realisiert. Bei Verwendung eines mobilen Endgeräts, ist dort sowohl das Authenticator-Modul, als auch das Anwendungsfrontend gemeinsam in einer Applikation installiert.

Der IdP-Dienst stellt die zentralisierte Identitätsprüfung der auf die Fachdienste zugreifenden Nutzer bereit. Als weitere Teile der Gesamtlösung sind neben dem IdP-Dienst die Clients (Anwendungsfrontend/Primärsystem) 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 IdP-Dienst bietet Fachdiensten seine Dienste an, auf welche Millionen Nutzer zeitgleich zugreifen. Eine wesentliche Ergänzung des IdP-Dienstes ist das Authenticator-Modul, welches auf den dezentralen Komponenten in den Praxen, Kliniken, Apotheken und bei den Versicherten betrieben wird.

A_20687 - Bereitstellung der PUK
Der Authorization Server MUSS zu allen verwendeten privaten Schlüsseln "`PrK_AUTH`", "`PrK_TOKEN`" und "`PrK_DISC`" das öffentliche Pendant "`PUK_AUTH`", "`PUK_TOKEN`" und "`PUK_DISC`" zum Download bereitstellen. Dies ermöglicht die Prüfung der von den einzelnen Schnittstellen vorgenommenen Signaturen ebenso wie die zielgerichtete Verschlüsselung des Payloads für den bestimmten Empfänger.**[<=]**

A_20732 - Aufnahme der öffentlichen Schlüssel in das Discovery Document
Der Authorization Server MUSS zu jedem privaten Schlüssel dessen öffentlichen Teil mit einer eigenen absoluten URI in das Discovery Document aufnehmen.**[<=]**

Hinweis: Die Bereitstellung von öffentlichem Schlüsselmaterial bezieht sich auf die Schlüssel zum Signieren und ggf. Verschlüsseln der JSON Web Token. Hiermit sind nicht die öffentlichen Schlüssel der TLS-Verschlüsselung gemeint.

A_20686 - Erweiterte Nutzung von Schlüsseln
Der Authorization Server MUSS die einzelnen Schnittstellen (AUTH, DISC, TOKEN) mit getrennten Interfaces bedienen.**[<=]**

4.1.1 Allgemeine Sicherheitsanforderungen

A_20582-01 - IdP-Dienst - Berücksichtigung OWASP-Top-10-Risiken
Der IdP-Dienst MUSS Maßnahmen zum Schutz sowohl vor den zum Zulassungszeitpunkt aktuellen OWASP-Top-10-Risiken umsetzen, als auch nach den zum Zulassungszeitpunkt aktuellen OWASP-Top-10-Risiken.**[<=]**

4.1.2 Sicherheit der Netzübergänge

Der IdP-Dienst wird für Versicherte über das Internet erreichbar gemacht und für Leistungserbringer über das Netz der TI. Die folgenden Anforderungen beschreiben die für diese Netzübergänge erforderlichen Sicherheitsmechanismen. Für den Netzübergang aus dem Internet als Transportnetz zum IdP-Dienst ist ein Paketfilter erforderlich.

A_20583 - IdP-Dienst – Sicherung zum Transportnetz Internet durch Paketfilter
Der Anbieter des IdP-Dienstes MUSS dafür sorgen, dass das Transportnetz Internet durch einen Paketfilter (ACL) gesichert wird und ausschließlich die erforderlichen Protokolle weiterleitet. Der Anbieter des IdP-Dienstes MUSS dafür sorgen, dass der Paketfilter des IdP-Dienstes frei konfigurierbar auf der Grundlage von Informationen aus OSI-Layer 3 und 4 ist, das heißt Quell- und Zieladresse, IP-Protokoll sowie Quell- und Zielport.**[<=]**

A_20584 - IdP-Dienst – Platzierung des Paketfilters Internet
Der Anbieter des IdP-Dienstes DARF den Paketfilter des IdP-Dienstes zum Schutz in Richtung Transportnetz Internet NICHT physisch auf dem vorgeschalteten TLS-terminierenden Load Balancer implementieren.**[<=]**

A_20585 - IdP-Dienst – Richtlinien für den Paketfilter zum Internet
Der Paketfilter des IdP-Dienstes MUSS die Weiterleitung von IP-Paketen an der Schnittstelle zum Internet auf das HTTPS- Protokoll beschränken.**[<=]**

A_20586-01 - IdP-Dienst – Verhalten bei Vollauslastung
Der Anbieter des IdP-Dienstes MUSS den Paketfilter des IdP-Dienstes so konfigurieren, dass bei Vollauslastung der Systemressourcen im IdP-Dienst keine weiteren Verbindungen angenommen werden.**[<=]**

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

A_20587 - IdP-Dienst – Richtlinien zum TLS-Verbindungsaufbau
Der Anbieter des IdP-Dienstes MUSS dafür sorgen, dass der Eingangspunkt des IdP-Dienstes sich beim TLS-Verbindungsaufbau über das Transportnetz gegenüber dem Client mit einem Extended Validation TLS-Zertifikat eines Herausgebers gemäß [CAB-Forum] authentisiert. Der Anbieter MUSS dafür sorgen, dass das Zertifikat sich an die jeweilige Schnittstelle des Eingangspunkts für Primärsysteme, Authenticator-Module und Frontends der Versicherten des IdP-Dienstes bindet, damit Clientsysteme beim TLS-Verbindungsaufbau eine vereinfachte Zertifikatsprüfung mit TLS-Standardbibliotheken durchführen können.**[<=]**

4.2 Fehlermeldungen

A_20680 - Format der Fehlermeldungen
Der IdP-Dienst MUSS für die verschiedenen Teilfunktionen geeignete Fehlermeldungen erzeugen und diese an den jeweiligen Aufrufer übergeben.**[<=]**

A_20681 - Nutzung von eindeutigen Error-Codes bei der Erstellung von Fehlermeldungen
Der IdP-Dienst MUSS Fehler durch eine eindeutige Nummer erkennbar machen und der gematik eine Liste der Error-Codes zur Verfügung stellen, damit die Ursachenklärung vereinfacht möglich wird.**[<=]**

A_20682 - Verwendung eines einheitlichen Schemas für die Aufbereitung von Fehlermeldungen
Der IdP-Dienst MUSS alle ausgeworfenen Fehlermeldungen zur Weiterverarbeitung in einem einheitlichen Schema aufbereiten und bereitstellen. Zeitstempel MÜSSEN auf der UTC basieren.
Beispiel: Ist eine Signatur nicht vorhanden oder defekt, bricht der Authorization-Endpunkt die Bearbeitung mit dem registrierten Fehlercode und einer für den Nutzer verständlichen Fehlermeldung ab.

Tabelle 4: TAB_IDP_DIENST_0004 Schema der Fehlermeldungen

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]*

[<=]

A_20683 - Formulierung der Fehlermeldungen
Der IdP-Dienst MUSS Fehlermeldungen, welche dem Nutzer angezeigt werden, in der Art ausformulieren, dass es dem Nutzer möglich ist, eigenes Fehlverhalten anhand der Fehlermeldung abzustellen.**[<=]**

A_20684 - Nutzung einer eindeutigen Beschreibung beim Aufbau von Fehlermeldungen
Der IdP-Dienst MUSS jedem Fehler eine eindeutige eigene Beschreibung zukommen lassen, sodass eine Fehlermeldung nicht für unterschiedliche Fehlerursachen zur Anwendung kommt.**[<=]**

A_20685 - Ausgabe der Fehlermeldungen in umgekehrter Reihenfolge des Auftretens
Der IdP-Dienst MUSS aufeinander aufbauende Fehlermeldungen in der umgekehrten Reihenfolge ihres Auftretens "Traceback (most recent call last)" ausgeben.**[<=]**

4.3 Schnittstellenbeschreibung des IdP-Dienstes

Der IdP-Dienst bietet zahlreiche Schnittstellen gegenüber unterschiedlichen Akteuren inner- und außerhalb der TI an, weswegen es notwendig ist, die einzelnen Schnittstellen so zu beschreiben, dass andere Akteure deren Funktionsweise leichter verstehen können. Nachfolgende Abbildung skizziert die Schnittstellen des IdP-Dienstes. Komponenten und Schnittstellen, welche nicht direkt vom IdP-Dienst genutzt werden, sind in der Abbildung grau hinterlegt.

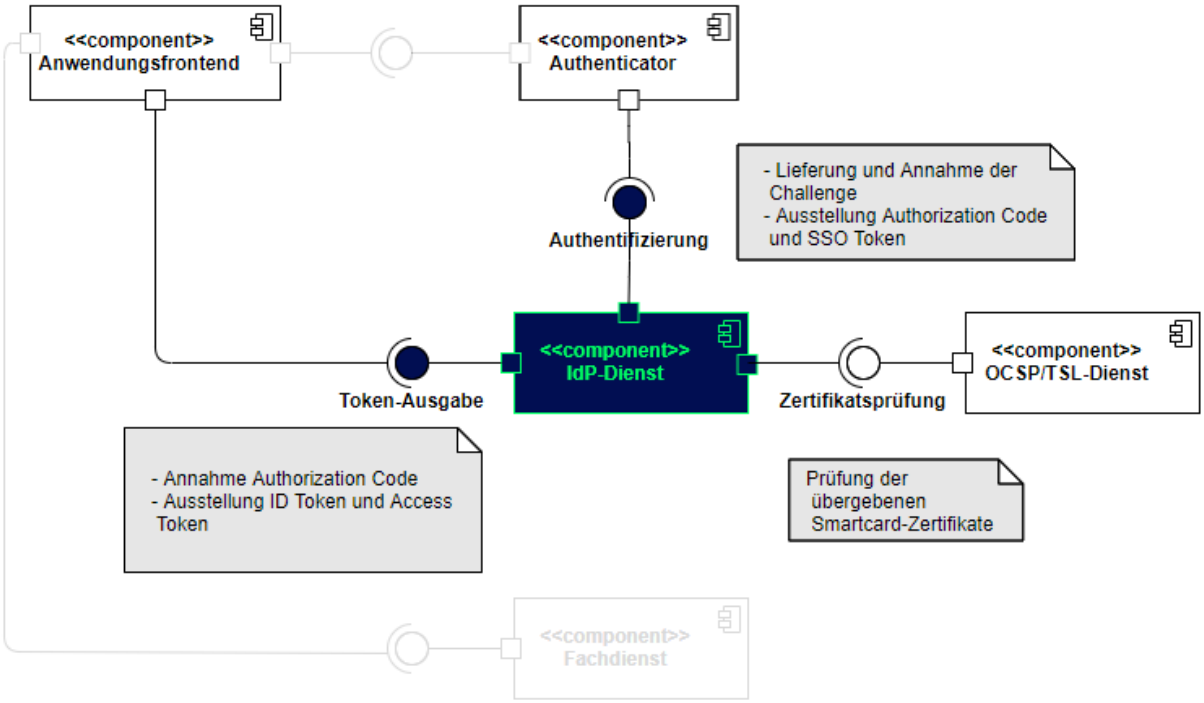

Abbildung 5: Schnittstellen des IdP-Dienstes

Die erste tokenbezogene Anfrage an den Authorization Server des IdP-Dienstes geht am Authorization-Endpunkt [RFC6749 # section-3.1] ein. Das Authenticator-Modul reicht dort am Endpunkt den "CONSENT" mit der "CHALLENGE" ein, mit welchem die "TOKEN" erstellt werden sollen, und erhält den "AUTHORIZATION_CODE" zurück, falls die Prüfung der signierten "CHALLENGE" und die Prüfung des übergebenen Smartcard-Zertifikats am OCSP/TSL-Dienst positiv ausfallen. Das Anwendungsfrontend reicht den "AUTHORIZATION_CODE" am Token-Endpunkt [RFC6749 # section-3.2] des IdP-Dienstes ein. Der IdP-Dienst überprüft den "AUTHORIZATION_CODE" und stellt bei positiver Validierung einen "ID_TOKEN" und einen "ACCESS_TOKEN" aus.

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 HBA gewertet wird. Basierend darauf dürfen weitere "ACCESS_TOKEN" und "SSO_TOKEN" für andere Anwendungsfrontends und Fachdienste ausgegeben werden, wenn das jeweils vorliegende Claim durch die dem Authorization Server vorliegenden Informationen bedient werden kann. Ist der Zeitpunkt der letzten Authentisierung zu lange her oder wird das Authenticator-Modul zum ersten Mal gestartet, muss eine Authentisierung erfolgen.

Hinweise für die Implementierung der Authentifizierung für Primärsystemen werden in [gemILF_PS_eRp] beschrieben.

Der Vorgang der Authentifizierung gegen die eGK oder den HBA ist nicht Bestandteil dieser Spezifikation, sondern ist im gesonderten Dokument [gemSpec_IDP_Frontend] beschrieben.

4.4 Identifikation des Clientsystems

Der IdP-Dienst verwaltet und steuert den Authentisierungsprozess für das E-Rezept und perspektivisch auch weitere Anwendungen. Damit kommt ihm eine Relevanz in der Gesundheitsversorgung zu, die sich zum einen in einer hohen Verfügbarkeit und zum anderen in einem hohen Angriffspotential widerspiegelt. Zur Unterstützung der betrieblichen Überwachung des IdP-Dienstes wird die Nutzung der im Feld befindlichen Clientsysteme protokolliert. Dabei ist der Zugriff auf die Schnittstellen des IdP-Dienstes nur durch Primärsysteme der Leistungserbringer, sein eigenes Authenticator-Modul und zugelassene E-Rezept-FdVs zulässig. Der E-Rezept-Fachdienst erkennt die Clientsysteme anhand des User-Agent-Header eingehender HTTP-Requests und protokolliert diesen Wert.

A_20588 - IdP-Dienst - Erkennung Clientsystem User-Agent
Der IdP-Dienst MUSS das vom aufrufenden Nutzer verwendete Clientsystem (Authenticator-Modul, E-Rezept-FdV oder Primärsystem) anhand des im HTTP-Request enthaltenen Header-Feld "User-Agent" gemäß [RFC7231] erkennen und in den Einträgen zur Performance-Rohdatenerfassung gemäß [gemSpec_Perf] protokollieren. Der IdP-Dienst MUSS bei fehlendem User-Agent-Header den Request mit dem HTTP-Status-Code 400 beantworten, damit in der Betriebsüberwachung des IdP-Dienstes die Nutzung unzulässiger Clientsysteme erkannt werden kann.**[**<=**]**

A_20589 - IdP-Dienst – Ausschluss bestimmter Clientsystem-Versionsnummern von der Kommunikation
Der IdP-Dienst MUSS die aus dem Internet vom Clientsystem mitgeteilte Versionsnummer aus dem HTTP-Header User-Agent, erkennen und festgelegte Versionsnummern über ein Blacklisting von einer Kommunikation mit dem IdP-Dienst ausschließen können. Der IdP-Dienst MUSS in diesen Fällen eine entsprechende Fehlermeldung an das Clientsystem geben.**[**<=**]**

A_20590 - IdP-Dienst – Ausschluss von Clientsystem-Versionen
Der Anbieter des IdP-Dienstes MUSS ausschließlich auf Anweisung der gematik Clientsysteme mit bestimmten Versionsnummern von einer Kommunikation mit dem IdP-Dienst ausschließen.**[**<=**]**

A_20742 - Vergabe der "client_id" durch den IdP-Dienst
Der IdP-Dienst MUSS bei der organisatorischen Registrierung des Anwendungsfrontends diesem eine eindeutige "`client_id`" zur Nutzung des IdP-Dienstes zuweisen.
[<=**]**

5 Funktionsmerkmale

5.1 Authorization Server Metadata (Discovery Document)

Der Authorization Server dient dazu 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 sind, wie diese Schnittstellen abgesichert sind und woher man die zur Etablierung der gewünschten Sicherheit erforderlichen Materialien beziehen kann. Gemäß dem verwendeten Standard OpenID Connect mit OAuth 2.0 kommen JSON Web Token (JWT), JSON Web Encryption (JWE), JSON Web Signature (JWS) und JSON Web Key (JWK) zum Einsatz.

Um nutzenden Anwendungen eine einheitliche Bezugsquelle für die Adressierung von Schnittstellen zu schaffen, werden die für alle Akteure grundlegenden Schnittstellen im sogenannten Discovery Document zusammengefasst und dort unter der `"URI_DISC"` 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.

~~**A_20895 - Anbieter IdP-Dienst - Resource Records FQDN idp**~~
~~Der Anbieter des IdP-Dienstes MUSS im Namensraum der TI und in den Nameservern Internet die Ressource Records gemäß nachstehender Tabelle verwalten.~~

~~Resource Record Bezeichner~~	~~Resource Record Type~~	~~Beschreibung~~
~~idp.ti-dienste.de~~	~~A Record~~	~~A Resource Records zur Namensauflösung von FQDN des IdP-Dienstes in IP-Adressen im Namensraum der TI~~
~~idp.idp.ti-dienste.de~~	~~A Record~~	~~A Resource Records zur Namensauflösung von FQDN des IdP-Dienstes in IP-Adressen im Namensraum Internet~~
~~idp.ti-dienste.de~~	~~AAAA Record~~	~~AAAA Resource Records zur Namensauflösung von FQDN des IdP-Dienstes in IP-Adressen im Namensraum Internet~~
~~_idp._tcp.ti-dienste.de~~	~~TXT~~	~~TXT Resource Records zur Ermittlung der Aufruf-Schnittstelle des IdP-Dienstes. Der für die Adressierung benötigte Resource Record MUSS bereitgestellt werden. Das in den Klammern angegebene Kürzel MUSS verwendet werden.~~ ~~• IdP-Dienst-Schnittstelle (idp)~~

		~~• OCSP-Status-Proxy (ocspf)~~ ~~Das key/value-Paar des TXT-Records hat folgende Struktur (die spitzen Klammern dienen der Abgrenzung eines Wertes): "idp=<Schnittstelle IdP>"~~

~~[<=]~~A_21219 - Anbieter IDP-Dienst - Schnittstellenadressierung
Der Anbieter des IDP-Dienst MUSS die im Internet angebotene Schnittstelle des IDP-Dienst unter der folgenden URL zur Verfügung stellen:

https://idp.zentral.idp.splitdns.ti-dienste.de/openid-configuration - Schnittstelle Discovery-Endpunkt
https://idp.zentral.idp.splitdns.ti-dienste.de/authorization - Schnittstelle Authorization-Endpunkt
https://idp.zentral.idp.splitdns.ti-dienste.de/token - Schnittstelle Token-Endpunkt
[<=]

A_20457 - Verwendung eindeutiger URI
Der IdP-Dienst MUSS alle verwendeten Adressen in Form von URL gemäß [RFC1738] angeben und in einem Discovery Document gemäß [RFC8414 # section-2] innerhalb der TI und im Internet veröffentlichen.**[<=]**

A_20688 - Discovery Document interne und externe Adressierung
Der Discovery-Endpunkt MUSS die Discovery Documents für interne und externe Adressierung sowohl innerhalb der TI als auch im Internet veröffentlichen.**[<=]**

Das Discovery Document innerhalb der TI adressiert hierbei die URI der Fachdienste und Schnittstellen des IdP-Dienstes innerhalb der TI. Das im Internet bereitgestellte Discovery Document stellt die URI der angebotenen Fachdienste im Internet mit dort auflösbaren Adressen bereit.

Hinweis: 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. Top-Level-Domain (TLD) verweisen.

A_20689 - Internes Discovery Document - Prüfung vor Veröffentlichung
Der IdP-Dienst MUSS alle von ihm im internen Discovery Document angebotenen URL und URL anderer Dienste, insbesondere Fachdienste, vor deren Veröffentlichung im internen Discovery Document auf bloße Erreichbarkeit prüfen.**[<=]**

A_20690 - 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 externen Discovery Document auf bloße Erreichbarkeit prüfen.**[<=]**

5.1.1 Aufbau des Discovery Documents

Der Authorization Server muss das Discovery Document gemäß [RFC8414] bereitstellen.

A_20439 - Das Discovery Document enthält statische Adressen
Der Discovery-Endpunkt MUSS sowohl im internen, als auch im externen Discovery Document die Akteure mit ihrer URI veröffentlichen.
[<=]

A_20458 - Inhalte des Discovery Documents
Der Discovery-Endpunkt MUSS sowohl im internen, als auch im externen Discovery Document gemäß [RFC8414 # section-2] mindestens die folgenden Attribute als URI angeben:

- "iss" (hier ist der IdP-Dienst erreichbar)
- "jwks_uri" (für den Abruf der/des PUK des Authorization Server [RFC7517])
- "URI_DISC" (URI, unter welcher das Discovery Document bereitgestellt ist)
- "URI_AUTH" &"PUK_URI_AUTH" URI des Dienstes und des öffentlichen Schlüssels des Authorization-Endpunktes gemäß [RFC6749]
- "URI_TOKEN" &"PUK_URI_TOKEN" URI des Dienstes und des öffentlichen Schlüssels des Token-Endpunktes gemäß [RFC6749]

[<=]

Hinweis: Ein Beispiel eines Discovery Documents kann unter folgendem Link gefunden werden: HEART II # rfc.section.3.1.5

5.1.2 Erneuerung des Discovery Documents

Der Authorization Server muss das Discovery Document mit den Metainformationen zu den Teildiensten mindestens einmal täglich und immer nach Änderungen mit dem "PrK_DISC" signieren und am mit der gematik vereinbarten Downloadpunkt "URI_DISC" bereitstellen.

A_20691 - Das Discovery Document ist maximal 24 Stunden alt
Der Discovery-Endpunkt MUSS das Discovery Document regelmäßig alle 24 Stunden oder nach durchgeführten Änderungen umgehend neu erstellen, mit dem "PrK_DISC" signieren und und am mit der gematik vereinbarten Downloadpunkt "URI_DISC"bereitstellen.
[<=]

A_20592 - Aktualisierung der Discovery Documents
Der IdP-Dienst MUSS das interne und externe Discovery Document bei Änderungen mit dem "PrK_DISC" neu signieren und am mit der gematik vereinbarten Downloadpunkt"URI_DISC" bereitstellen**[<=]**

5.1.3 Schutz des Discovery Documents

Der Authorization Server schützt die Integrität des Discovery Document auf Dateiebene durch eine Signatur und während des Transportes zusätzlich mittels TLS.

A_19874-04~~A_19874-03~~ - Bereitstellung des internen Discovery Documents innerhalb der TI
Der IdP-Dienst MUSS das interne Discovery Document mit einem Zertifikat des Typs FD.SIG und der technischen Rolle „oid_idpd" gemäß [gemSpec_Krypt # Abschnitt 3.7] signiert, an einem spezifischen Downloadpunkt TLS-gesichert innerhalb der TI bereitstellen.
Die URL des Downloadpunktes lautet: "**https://idp.zentral.idp.splitdns.ti-dienste.de~~/.well-known~~/openid-configuration**". <=**[**<=**]**

~~Hinweis: Das Discovery Document kann unter dem relativen Pfad .well-known/openid-configuration eingelesen werden.~~

A_19877-02~~A_19877-01~~ - Bereitstellung des externen Discovery Documents im Internet
Der IdP-Dienst MUSS das externe Discovery Document mit einem Zertifikat des Typs FD.SIG und der technischen Rolle „oid_idpd" gemäß [gemSpec_Krypt # Abschnitt 3.7] signiert und TLS-gesichert im Internet zum Download bereitstellen.
Die URL des Downloadpunktes lautet: "**https://idp.zentral.idp.splitdns.ti-dienste.de~~/.well-known~~/openid-configuration**" **[**<=**]**

Hinweis: Die für die Rolle des IdP-Dienstes vorgesehene professionOID ist in [gemSpec_OID] beschrieben. ~~Der externe Downloadpunkt des Discovery Document ist der folgende: idp.ti-dienste.de/.well-known/openid-configuration~~

A_20591 - Festlegungen zur Signatur der Discovery Documents
Der IdP-Dienst MUSS die Signatur der Discovery Documents als base64-kodierte CMS-Signatur gemäß [RFC5652] realisieren und die Festlegungen aus gemSpec_Krypt#5.6.2 beachten.
Der IdP-Dienst MUSS bei der Signaturerstellung das Signaturzertifikat als Attribut *signing certificate reference* gemäß [CAdES # Kapitel 5.7.3 „Signing Certificate Reference Attributes"] einbetten.**[**<=**]**

5.2 Authorization-Endpunkt

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

A_20434 - Einhaltung der Standards bei der Realisierung des Authorization-Endpunkts
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 festgelegten Standards implementieren.**[**<=**]**

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_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_20698 - Annahme des Authorization Request
Der Authorization-Endpunkt MUSS die im Authorization Request des Authenticator-Moduls mitgelieferten "CODE_CHALLENGE" und den "SCOPE" annehmen.**[<=]**

Hinweis: Nachfolgend wird beispielhaft der Authorization Request als HTTP GET-Request dargestellt, welcher vom Authenticator-Modul initiiert wird:

```
GET /auth?response_type=code&scope=openid%20e-
rezept&state=af0ifjsldkj&client_id=ZXJlemVwdC1hcHA&redirect_uri=https%3A%2F
%2Fapp.e-
rezept.com%2Fauthnres&code_challenge_method=S256&code_challenge=S41HgHxhXL1
CIpfGvivWYpbO9b_QKzva-9ImuZbt0Is

HTTP/1.1
Host: idp.com
X-E-Rezept-App: 1.0
Accept: application/json
User-Agent: E-Rezept-App/1.0
```

A_20376 - Verwendung des Attributes "state"
Der Authorization-Endpunkt MUSS den vom Anwendungsfrontend initierten "state"-Parameter gemäß [RFC6749 # section-10.12] bei einer Redirection an den Client in seiner Antwort verwenden.**[<=]**

A_20731 - Verwendung des Attributes "auth_time"
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 in freigeschaltetem Zustand) setzen.**[<=]**

A_20440 - Schematische Prüfung des Consent
Der IdP-Dienst MUSS den eingereichten Consent auf dessen Übereinstimmung mit dem vorliegenden Claim (mit dem Fachdienst abgestimmte Key/Value-Paare im Payload des Token) zum beantragten Token abgleichen, insbesondere die "redirect_uri" aus dem Registrierungszusammenhang.**[<=]**

A_20379 - Abbruch bei schematischer Inkonsistenz im Consent
Der Authorization-Endpunkt MUSS die Bearbeitung mit dem registrierten Fehlercode und einer für den Nutzer verständlichen Fehlermeldung abbrechen, wenn das Schema des Consent und das des vorliegenden Claims nicht übereinstimmen.**[<=]**

A_20310 - Verarbeitung des Consent
Der Authorization-Endpunkt MUSS die Ausstellung des vereinbarten "AUTHORIZATION_CODE" mit den im Claim vorliegenden Parametern veranlassen, wenn der Authorization-Endpunkt den eingereichten Consent allen vorgesehenen Prüfungen unterzogen hat und dabei keine Fehler aufgetreten sind. **[<=]**

A_20459 - Das Attribut AUTH_TIME muss in allen Token unverändert bleiben
Der Authorization-Endpunkt DARF den Zeitpunkt der letzten Authentisierung im Attribut "auth_time" NICHT verändern.**[<=]**

A_20699 - Annahme der signierten "CHALLENGE"
Der Authorization-Endpunkt MUSS die "CHALLENGE", signiert mit dem Zertifikat der Smartcard des Nutzers, übertragen durch das Authenticator-Modul annehmen.**[<=]**

Hinweis: Der folgende Aufruf skizziert einen beispielhaften HTTP GET-Request an den Authorization-Endpunkt, welcher vom Authenticator-Modul initiiert wird:

```
GET /auth?response_type=code&scope=openid%20e-
rezept&state=af0ifjsldkj&client_id=ZXJlemVwdC1hcHA&redirect_uri=https%3A%2F
%2Fapp.erezept.com%2Fauthnres&code_challenge_method=S256&code_challenge=S41
HgHxhXL1CIpfGvivWYpbO9b_QKzva-9ImuZbt0Is

HTTP/1.1
Host: idp.com
X-E-Rezept-App: 1.0
Accept: application/json
User-Agent: E-Rezept-App/1.0
```

A_20951 - Validierung der Signatur und des Zertifikats der "CHALLENGE"
Der Authorization-Endpunkt MUSS die Signatur der vom Authenticator-Modul übertragenen, signierten "CHALLENGE" anhand des mitgelieferten Authentifizierungs-Zertifikats überprüfen. Die Überprüfung MUSS neben der Signatur auch das Authentifizierungszertifikat anhand von OCSP umfassen. **[<=]**

A_20946 - Annahme eines "SSO_TOKEN"
Der Authorization-Endpunkt MUSS einen vom Autenticator-Modul übertragenen "SSO_TOKEN" annehmen.**[<=]**

A_20947 - Entschlüsselung des "SSO_TOKEN"
Der Authorization-Endpunkt MUSS den angenommenen "SSO_TOKEN" mit seinem eigenen Schlüsselmaterial, welches zur Verschlüsselung genutzt wurde, entschlüsseln.**[<=]**

A_20948 - Validierung des "SSO_TOKEN"
Der Authorization-Endpunkt MUSS den angenommenen und entschlüsselten "SSO_TOKEN" validieren. Die Validierung MUSS die Überprüfung der Signatur anhand seines öffentlichen Schlüssels PUK_AUTH und die Überprüfung der zeitlichen Gültigkeit des "SSO_TOKEN" anhand des Attributs "auth_time" umfassen.**[<=]**

A_20949 - Anforderung einer Authentisierung bei negativer Validierung des "SSO_TOKEN"
Der Authorization-Endpunkt MUSS eine neue Authentisierung vom Authenticator-Modul anfordern, wenn die Validierung des vom Authenticator-Moduls eingereichten "SSO_TOKEN" fehlschlägt.**[<=]**

A_20950 - Positive Validierung des "SSO_TOKEN"
Der Authorization-Endpunkt MUSS bei der positiven Validierung des vom Authenticator-Moduls eingereichten "SSO_TOKEN" einen "ACCESS_TOKEN" für den angefragten Fachdienst ausstellen.**[<=]**

Hinweis: Der Authorization-Endpunkt muss damit die im "SSO_TOKEN" gelieferten Claims überprüfen und einen "AUTHORIZATION_CODE" für den angefragten Fachdienst ausstellen.

A_20522 - Erstellen einer "SESSION_ID"
Der Authorization-Endpunkt MUSS eine neue "SESSION_ID" anlegen, sobald ein Authorization Request eingeht.**[<=]**

A_20523 - Zusammenstellung der Claims zum "user_consent"
Der Authorization-Endpunkt MUSS die für den vorgetragenen "SCOPE" vom einfordernden Fachdienst erwarteten Claims zur "USER_CONSENT"-Anfrage zusammenstellen.**[<=]**

A_20460 - Der Authorization-Endpunkt bestätigt ausschließlich Zertifikatsinformationen
Der Authorization-Endpunkt MUSS bei der Annahme des Zertifikates durch ein Challenge-Response-Verfahren prüfen, ob der Nutzer auch die zum Zertifikat (Besitz) gehörige PIN (Wissen) kennt, um sicherzustellen, dass der Nutzer berechtigt ist, die vorgetragene Identität (Zertifikat) zu nutzen.**[<=]**

A_20521-01 - Inhalt der Challenge an das Authenticator-Modul
Der IdP-Dienst MUSS die ihm vorliegenden Session-Informationen (z.B. "SESSION_ID", "CODE_CHALLENGE", "SCOPE" und alle Informationen über Anwendungsfrontend und Authenticator-Modul) mit seinem privaten Schlüssel "PrK_AUTH" signieren und als JWT ergänzt um die "USER_CONSENT"-Anfrage an das Authenticator-Modul senden.**[<=]**

Hinweis: Nachfolgend wird beispielhaft ein "CHALLENGE_TOKEN" in Form eines JSON Web Token (JWT) dargestellt:

```
Challenge JWT:
challenge_headers = {
    "typ":"JOSE+JSON",
    "iat":1591714252326,
    "exp":1591714552326,
    "jti":"c3a8f9c8-aa62-11ea-ac15-6b7a3355d0f6",
    "snc":"sLlxlkskAyuzdDOwe8nZeeQVFBWgscNkRcpgHmKidFc"
}
challenge_payload = {
    "response_type":"code",
    "scope":"openid e-rezept",
    "client_id":"ZXJlemVwdC1hcHA",
"state":"af0ifjsldkj",
    "redirect_uri":"https://app.e-rezept.com/authnres",
    "code_challenge_method":"S256",
    "code_challenge":"S41HgHxhXL1CIpfGvivWYpbO9b_QKzva-9ImuZbt0Is"
}
```

Der Authorization-Endpunkt hat den "CHALLENGE_TOKEN" mit seinem privaten Schlüssel "PRK_AUTH" signiert. Der folgende Aufruf skizziert beispielhaft die Antwort des Authorization-Endpunktes, welche vom Authenticator-Modul angenommen wird. Der "CHALLENGE_TOKEN" wird dabei nur angedeutet:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "challenge":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpPU0UrSlNPTiIsImlhdCI6MTU5MTcxNDI1MjMy......
",
  "user_consent": {
    "client_name": "E-Rezept App",
    "url": "https://e-rezept.com/",
    "requested_scope": {
      "openid": "Der Zugriff auf den ID Token"
      "e-rezept": "Zugriff auf die E-Rezept Funktionalität."
    },
"show_once": true,
"amr": ["JWT-Challenge-Response"]
    // ggf. mehr Informationen, welche dem Nutzer angezeigt werden sollen,
wie die Auflistung der mit der Zustimmung weitergegebenen Daten
```

```
    }
}
```

A_20604 - Signatur der Challenge
Der IdP-Dienst MUSS die Challenge für die Authentisierung mit einem Zertifikat des Typs FD.SIG und der technischen Rolle „oid_idpd" gemäß [gemSpec_Krypt # Abschnitt 3.7] signieren. **[<=]**

A_20313 - Inhalte des Claims
Der IdP-Dienst MUSS `"ID_TOKEN"`, `"ACCESS_TOKEN"` und `"SSO_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 der Gültigkeitsdauer (`"SUBJECT_SESSION"`, `"AUTHORIZATION_CODE"`, `"ACCESS_TOKEN"`, `"SSO_TOKEN"` und `"ID_TOKEN"`) gemäß der spezifizierten Maximalwerte.**[<=]**

A_20692 - Maximale Gültigkeitsdauer einer "SUBJECT_SESSION"
Der Authorization Server DARF die zeitliche Gültigkeit einer`"SUBJECT_SESSION"` NICHT länger als 86400 Sekunden (24 Stunden) einstellen.
Der Parameter `"auth_time"` beinhaltet den Zeitpunkt der letzten Authentisierung.**[<=]**

A_20314 - Maximale Gültigkeitsdauer des "AUTHORIZATION_CODE"
Der Authorization Server DARF die zeitliche Gültigkeit des `"AUTHORIZATION_CODE"` NICHT länger als 180 Sekunden (Challenge-Response) und nach dessen Übergabe an das Anwendungsfrontend nicht länger als 60 Sekunden einstellen.**[<=]**

A_20315 - "AUTHORIZATION_CODE" nach Gültigkeitsende nicht mehr verwenden
Der Authorization Server DARF außerhalb der Gültigkeitsdauer eingehenden `"AUTHORIZATION_CODE"` NICHT in `"ID_TOKEN"`, `"ACCESS_TOKEN"` oder `"SSO_TOKEN"` eintauschen.**[<=]**

Die Gültigkeitsdauer des `"AUTHORIZATION_CODE"` wird im Claim des angesprochenen Fachdienstes definiert.

A_20462 - Maximale Gültigkeitsdauer des "ID_TOKEN"
Der Token-Endpunkt DARF `"ID_TOKEN"` mit einer Gültigkeitsdauer von mehr als 86400 Sekunden (24 Stunden) NICHT ausstellen.**[<=]**

A_20463 - Maximale Gültigkeitsdauer des "ACCESS_TOKEN"
Der Token-Endpunkt DARF `"ACCESS_TOKEN"` mit einer Gültigkeitsdauer von mehr als 300 Sekunden (5 Minuten) NICHT ausstellen.**[<=]**

Die Gültigkeitsdauer des `"ACCESS_TOKEN"` wird im Claim des angesprochenen Fachdienstes definiert.

A_20464 - Token-Endpunkt (Datensparsamkeit)
Der Token-Endpunkt DARF andere Informationen, als die im Claim geforderten, NICHT herausgeben.**[<=]**

A_20318 - Keine Token für widerrufene Entitäten
Der Authorization-Endpunkt DARF für nicht existente Entitäten NICHT einen `"AUTHORIZATION_CODE"`, einen `"ID_TOKEN"`, einen `"ACCESS_TOKEN"` oder einen `"SSO_TOKEN"` auszustellen. **[<=]**

A_20465 - Zertifikatsprüfung gegen OCSP-Responder
Der Authorization-Endpunkt MUSS das Zertifikat des Antragstellers immer gegen den zugehörigen OCSP-Responder innerhalb der TI auf Gültigkeit prüfen.**[<=]**

5.2.2 Authorization-Endpunkt Ausgangsdaten

Konnten alle Prüfungen des eingereichten Consent erfolgreich abgeschlossen werden, erstellt der Authorization-Endpunkt ein "ID_TOKEN", "ACCESS_TOKEN", ergänzt durch ein "SSO_TOKEN". Die Übertragung der Token erfolgt jedoch nicht direkt über das Authenticator-Modul, sondern in Form eines "AUTHORIZATION_CODE". Die Token werden am Token-Endpunkt zum Download bereitgestellt, wo das jeweilige Anwendungsfrontend diese gegen gleichzeitige Vorlage von "authorization_code" und des eigenen "code_verifier", auf welchem der bereits vorliegende Hash-Wert beruht, erhält.

A_20377 - Verwendung des Attributes "state"
Der Authroization-Endpunkt MUSS den "state"-Parameter [RFC6749 # section-10.12] des Anwendungsfrontends in allen darauf basierenden Responses verwenden.**[<=]**

A_20694 - Zusammenstellung des "SSO_TOKEN"
Der Authorization-Endpunkt MUSS den "SSO_TOKEN" so zusammenstellen, dass alle Informationen, welche für die Ausstellung eines neuen "ACCESS_TOKEN" benötigt werden, im Token vorhanden sind.**[<=]**

A_20695 - Signieren des "SSO_TOKEN"
Der Authorization-Endpunkt MUSS den "SSO_TOKEN" mit seinem eigenen privaten Schlüssel signieren "PrK_AUTH".**[<=]**

A_20696 - Verschlüsselung des "SSO_TOKEN"
Der Authorization-Endpunkt verschlüsselt den "SSO_TOKEN" für sich selbst mit eigenem Schlüsselmaterial, welches die gemSpec_Krypt beachtet.**[<=]**

A_20697 - Zusammenstellung des "AUTHORIZATION_CODE"
Der Authorization-Endpunkt erzeugt den "AUTHORIZATION_CODE" anhand der vom Authenticator-Modul übergebenen Daten im "CHALLENGE".**[<=]**

A_20319 - Signatur des "AUTHORIZATION_CODE"
Der IdP-Dienst MUSS den "AUTHORIZATION_CODE" für die Authentisierung mit einem Zertifikat des Typs FD.SIG und der technischen Rolle „oid_idpd" gemäß [gemSpec_Krypt # Abschnitt 3.7] signieren damit das Authenticator-Modul sicher gewährleisten kann, dass der eingehende"AUTHORIZATION_CODE" tatsächlich vom IdP-Dienst stammt [RFC7519 # section-7.1].**[<=]**

A_20693 - Senden des "AUTHORIZATION_CODE" und "SSO_TOKEN" an die "REDIRECT_URI"
Der IdP-Dienst MUSS den "AUTHORIZATION_CODE" und den "SSO_TOKEN" an das Authenticator-Modul über eine Redirection an die registrierte "REDIRECT_URI" des Anwendungsfrontends senden.**[<=]**

A_20320 - Sichere Übertragung des "AUTHORIZATION_CODE"
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].**[<=]**

Hinweis: Nachfolgend wird beispielhaft ein "AUTHORIZATION_CODE" in Form eines JSON Web Token (JWT) dargestellt:

```
Authorization Code:
code_header = {
    "typ": "JOSE",
    "jti": "18017c1c-aa7b-11ea-ac15-6b7a3355d0f6",
    "iat": 1591714352326,
    "exp": 1591714652326,
    "msg_type": "code"
}
code_payload = {
    "response_type": "code",
    "scope": "openid e-ezept",
    "client_id": client_id,
"state":"af0ifjsldkj",
    "redirect_uri": "https://app.e-rezept.com/authnres",
    "code_challenge_method": "S256",
    "code_challenge": "S41HgHxhXL1CIpfGvivWYpbO9b_QKzva-9ImuZbt0Is",
    "claims": {
      "sub": "RabcUSuuWKKZEEHmrcNm_kUDOW13uaGU5Zk8OoBwiNk",
  // die ausgelesenen Werte aus dem Smart Card Zertifikat
      "professionOID": "<die Profession OID>",
  "idNummer":"<KV-Nummer>",
  "name": "<Name des Versicherten>"
    }
}
```

Hinweis: Nachfolgend wird beispielhaft ein "SSO_TOKEN" in Form eines JSON Web Token (JWT) dargestellt (Der IdP entscheidet selbst über den Inhalt , da nur er diesen auch lesen kann. Entscheidend ist, dass die Informationen zur Ausstellung eines neuen "AUTHORIZATION_CODE" - die Nutzer Informationen aus dem Smartcard-Zertifikat des Nutzers - enthalten sind):

```
{ // protected header
     "alg": "dir",
     "enc": "A256GCM",
     "kid": <key_identifier of the ecryption key>,
     "jti": "c3a8f9c8-aa62-11ea-ac15-6b7a3355d0f6",
     "iat": <issuance time>,
     "exp": <expiration time of the token>,
     "typ": "JOSE",
     "msg_typ": "SSO-Token"
   }
   .
   { // body
     "iss": "https://idp.com/oidc",
     "aud": "https://idp.com/oidc",
     "cnf": {
       "jwk": {
         "kty": "EC",
         "use": "sig",
     "x": "l8tFrhx-34tV3hRICRDY9zCkDlpBhF42UQUfWVAWBFs",
     "y": "9VE4jf_Ok_o64zbTTlcuNJajHmt6v9TDVrU0CdvGRDA",
     "crv": "P-256"
       }
     },
     "claims": {
"professionOID": "<die Profession OID>",
"idNummer":"<KV-Nummer>",
"name": "<Name des Versicherten>"
```

```
      }
    }
```

Hinweis: Nachfolgend wird beispielhaft die Antwort des Authorization-Endpunkt als Redirection dargestellt. Der "AUTHORIZATION_CODE" und der "SSO_TOKEN" werden nur angedeutet":

```
HTTP/1.1 302 Found
Location: https://app.e-rezept.com/authnres?code=eyJhbGciOiJkaXIiL
CJlbmMiOiJBMjU2R0NNIiwiZXhwIjoxNT....&ssotoken=eyJhbGciOiJkaXIiLCJlbmMiOiJB
MjU2R0NNIiwiZXhwIjoxNTkxNzE0NjU....&state=af0ifjsldkj
```

5.3 Token-Endpunkt

Am Token-Endpunkt nimmt der Authorization Server **den** "AUTHORIZATION_CODE", welchen er selbst am Authorization-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 "AUTHORIZATION_CODE" das dabei mit übertragene "CODE_VERIFIER" verarbeiten, um mittels Vergleich der Hash-Werte die Übereinstimmung des den "AUTHORIZATION_CODE" einreichenden mit dem ursprünglich authentisierten Client sicherzustellen. Das verwendete Hash-Verfahren ist im Authorization Request anzugeben.

5.3.1 Token-Endpunkt Eingangsdaten

A_20321 - Annahme und Prüfung von "AUTHORIZATION_CODE" und "CODE_VERIFIER"
Der Token-Endpunkt MUSS den vom Anwendungsfrontend übertragenen "AUTHORIZATION_CODE" nach Überprüfung des zeitgleich eingereichten "CODE_VERIFIER" entwerten und das "ID_TOKEN" und "ACCESS_TOKEN" gesichert herausgeben. Der Token-Endpunkt MUSS die Überprüfung des "CODER_VERIFIER" gegen die "CODE_CHALLENGE" mit S256 (Algorithmus nach [RFC7636 # section-4.2]) durchführen.**[**<=**]**

A_20474 - "AUTHORIZATION_CODE" einmalige Verwendung
Der Authorization Server MUSS sicherstellen, dass auf einen wiederholt eingereichten "AUTHORIZATION_CODE" keine weiteren Token herausgegeben werden.**[**<=**]**

A_20323 - TOKEN-Ausgabe Protokollierung in allen Fällen
Der Token-Endpunkt MUSS die Herausgabe der "TOKEN" im Positiv- wie auch im Negativfall protokollieren.**[**<=**]**

5.3.2 Token-Endpunkt Ausgangsdaten

Alle vom IdP-Dienst herausgegebenen Informationen müssen mit dem privateKey des jeweiligen Teildienstes signiert sein, da die mit TLS abgesicherte Verbindung nicht in allen Anwendungsszenarien die Integrität der übertragenen Daten gewährleistet.

A_20524 - Befüllen der Claims "given_name", "family_name", "organizationName", "professionOID" und "idNummer"
Der Token-Endpunkt MUSS benötigte Attribute in Claims für das auszustellende "ACCESS_TOKEN" und das "ID_TOKEN" ausschließlich aus dem ihm mit der "challenge" eingereichten Authentifizierungs-Zertifikat der Smartcard (eGK, HBA oder SMC-B) beziehen.
Der Token-Endpunkt MUSS das Attribut "given_name" und "family_name" der juristischen und natürlichen Personen sowie die Attribute "organizationName", "professionOID" und "idNummer" entsprechend des Datenformates der Informationsquelle (Zertifikat) wie folgt befüllen:

Tabelle 5: TAB_IDP_DIENST_0005 Befüllung der Attribute "given_name", "family_name", "organizationName", "professionOID" und "idNummer"

Attribute	Leistungserbringer (HBA) Quell-Zertifikat: C.HP.AUT	Leistungserbringerinstitution (SMC-B) Quell-Zertifikat: C.HCI.AUT	Versicherte (eGK) Quell-Zertifikat: C.CH.AUT
Attribute "given_name" (Zertifikatsfeld)	Vorname (surname)	Vorname des Verantwortlichen/ Inhabers (surname)	Vorname (surname)
Attribute "family_name" (Zertifikatsfeld)	Nachname (givenName)	Nachname des Verantwortlichen/ Inhabers (givenName)	Nachname (givenName)
Attribute "organizationName" (Zertifikatsfeld)	leer (organizationName)	Organisationsbezeichnung (organizationName)	Herausgeber (organizationName)
Attribute "professionOID" (Zertifikatsfeld)	professionOID (Admission/professionOID)	professionOID (Admission/professionOID)	professionOID (Admission/professionOID)
Identifier "idNummer" (Zertifikatsfeld)	Telematik-ID (Admission/ registrationNumber)	Telematik-ID (Admission/ registrationNumber)	unveränderlicher Anteil der KVNR (organizationalUnitName)

[<=]

Hinweis: Nachfolgend wird ein beispielhafter Payload eines "ACCESS_TOKEN" dargestellt.

```
{
      "iss": "https://idp1.telematik.de/jwt",
      "sub": "RabcUSuuWKKZEEHmrcNm_kUDOW13uaGU5Zk8OoBwiNk",
```

```
    "professionOID": "1.2.276.0.76.4.50",
    "nbf": 1585336956,
    "exp": 1585337256,
    "iat": 1585336956,
    "given_name": "der Vorname",
    "family_name": "der Nachname",
    "organizationName": "Institutions- oder Organisations-Bezeichnung",
    "idNummer": "3-15.1.1.123456789",
    "jti": "<IDP>_01234567890123456789",
    "aud": "erp.zentral.erp.ti-dienste.de"
}
```

A_20952 - Claim "aud" im Token setzen
Der IdP-Dienst MUSS den Claim "aud" im "ACCESS_TOKEN" entsprechend des angefragten Scopes des Authenticator-Moduls mit der URL des Fachdienstes füllen.**[<=]**

Hinweis:

Für den E-Rezept-Fachdienst wird beispielsweise der folgende Wert genutzt: "aud" : "erp.zentral.erp.ti-dienste.de".

A_20327 - Signatur des "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN"
Der Token-Endpunkt MUSS alle erstellten "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN" mit seinem privateKey "PRK_TOKEN" gemäß signieren, um dessen Integrität sicherzustellen und eine eineindeutige Erklärung über dessen Herkunft abzugeben. [RFC7523 # section-3 Spiegelpunkt 9 und openid-heart-oauth2-1_0.html#rfc.section.3.2.1] sind zu gewährleisten.**[<=]**

A_20328 - Verschlüsselung des "ACCESS_TOKEN"
Der Token-Endpunkt MUSS das "ACCESS_TOKEN" mit dem öffentlichen Schlüssel des Fachdienstes "PUK_FD" verschlüsseln, um das "ACCESS_TOKEN" vor Kenntnisnahme durch Dritte, z.B. auch auf dem Endgerät des Nutzers, zu schützen [RFC6750 # section-5.2].
[<=]

A_20329 - Sichere Übertragung von "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN"
Der Token-Endpunkt MUSS "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN" beim Transport mit Transport Layer Security (TLS) gemäß [gemSpec_Krypt] schützen.
[<=]

A_20330 - Ausgabe der Token
Der Token-Endpunkt MUSS für den Versand der "ID_TOKEN" und "ACCESS_TOKEN" an das Anwendungsfrontend, die vom Authenticator-Modul im Consent der mit dem Vorgang verbundenen "SUBJECT_SESSION" gemeldete URI verwenden. Eine URI-Umleitung MUSS ausgeschlossen werden [RFC6749 # section-10.6].**[<=]**

Hinweis: Nachfolgend wird beispielhaft ein "ID_TOKEN" und ein "ACCESS_TOKEN" in Form eines JSON Web Token (JWT) dargestellt:

ID Token:

```
idt_headers = {
    # "typ":"JOSE+JSON"
}
idt_payload = {
 "iss": "https://idp.com/oidc",
 "sub": "RabcUSuuWKKZEEHmrcNm_kUDOW13uaGU5Zk8OoBwiNk",
```

```
 "aud": [client_id],
 "iat": 1591714452326,
 "exp": 1591714752326,
 "at_hash": at_hash
}
```

Hinweis: Der ID Token wird vom Token-Endpunkt signiert (mit "PrK_TOKEN")

Access Token:

```
at_headers = {
    "typ":"at+JWT"
}
at_payload = {
    "iss": "https://idp.com/oidc",
    "sub": "RabcUSuuWKKZEEHmrcNm_kUDOW13uaGU5Zk8OoBwiNk",
    "aud": "erp.zentral.erp.ti-dienste.de",
    "client_id": client_id,
    "scope": "openid e-rezept",
    "iat": 1591714452326,
    "exp": 1591714752326,
    "jti": "d8557394-ab37-11ea-ac15-6b7a3355d0f6",
    "professionOID":"<die Profession OID>",
"idNummer":"<KV-Nummer>",
"name":"<Name des Versicherten>"
}
```

Hinweis: Der Access Token wird vom Token-Endpunkt signiert (mit "PrK_TOKEN") und für den Fachdienst verschlüsselt (mit "PUK_FD"). Nachfolgend wird beispielhaft die Antwort des Token-~~Endpunkts~~Endpunkt als dargestellt. Der "ID_TOKEN" und der "ACCESS_TOKEN" werden nur angedeutet.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{"token_type": "Bearer",
 "expires_in": 300,
 "id_token": "...",
 "access_token": "...",
}
```

6 Anhang A – Verzeichnisse

6.1 Abkürzungen

Kürzel	Erläuterung
AVS	Apothekenverwaltungssystem
DLL	Dynamic Link Library
eGK	Elektronische Gesundheitskarte
HBA	Heilberufsausweis
IdP	Identity Provider
JSON	JavaScript Object Notation
JWE	JSON Web Encryption
JWS	JSON Web Signature
JWT	JSON Web Token
NFC	Near Field Communication (Kommunikation im Nahfeld einer Antenne)
OAuth 2.0	Open Authorization 2.0
OCSP	Online Certificate Status Protocol
OIDC	OpenID Connect
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PVS	Praxisverwaltungssystem
QES	Qualifizierte Elektronische Signatur
SMC-B	Security Module Card Typ B, Institutionenkarte
TI	Telematikinfrastruktur
TLD	Top Level Domain

TLS	Transport Layer Security
TSL	Trust-service Status List
URI	Uniform Resource Identifier

6.2 Glossar

Begriff	Erläuterung
Access Token	Ein Access Token (nach [RFC6749 # section-1.4]) wird vom Client (Anwendungsfrontend) benötigt, um auf geschützte Daten eines Resource Servers zuzugreifen. Die Representation kann als JSON Web Token erfolgen.
Authorization Server	OAuth2 Rolle (siehe [RFC6749 # section-1.1]): Der Authorization Server ist Teil des IdP-Dienstes. Der Server authentifiziert den Resource Owner (Nutzer) und stellt Access Tokens für den vom Resource Owner erlaubten Anwendungsbereich (Scope) für einen Resource Server bzw. eine auf einem Resource Server existierende Protected Resource aus.
Claim	Ein Key/Value-Paar im Payload eines JSON Web Token.
Client	OAuth2 Rolle (siehe [RFC6749 # section-1.1]): Eine Anwendung (Relying Party), die auf geschützte Ressourcen des Resource Owners zugreifen möchte, die vom Resource Server bereitgestellt werden. Der Client kann auf einem Server (Webanwendung), Desktop-PC, mobilen Gerät etc. ausgeführt werden.
Consent	Zustimmung des Nutzers zur Verarbeitung der angezeigten Daten. Der Consent umfasst die Attribute, welche vom IdP-Dienst 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.
Discovery Document	Ein OpenID Connect Metadatendokument (siehe [openid-connect-discovery 1.0]), das den Großteil der Informationen enthält, die für eine App zum Durchführen einer Anmeldung erforderlich sind. Hierzu gehören Informationen wie z.B. die zu verwendenden URLs und der Speicherort der öffentlichen Signaturschlüssel des Dienstes.

Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.
ID Token	Ein auf JSON basiertes und nach [RFC7519] (JWT) genormtes Identitäts-Token, mit dem ein Client (Anwendungsfrontend) die Identität eines Nutzers überprüfen kann.
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]).
JSON Web Token	Ein auf JSON basiertes und nach [RFC7519] (JWT) genormtes Access-Token. Das JWT ermöglicht den Austausch von verifizierbaren Claims innerhalb seines Payloads.
Resource Owner	OAuth2-Rolle (siehe [RFC6749 # 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 [RFC6749 # 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 Owners.
SSO Token	Gegen Vorlage eines gültigen SSO Token ist keine erneute Nutzerauthentisierung für die Ausstellung eines Access Tokens am IdP-Dienst nötig.
Token-Endpunkt	Ein Endpunkt des Authorization Servers, welcher für die Ausstellung von Token ("`ID_TOKEN`" und "`ACCESS_TOKEN`") zuständig ist.

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 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
[gemILF_PS_eRp]	gematik: Spezifikation Implementierungsleitfaden Primärsysteme - E-Rezept
[gemSpec_IDP_Frontend]	gematik: Spezifikation Identity Provider-Frontend
[gemSpec_IDP_FD]	gematik: Spezifikation Identity Provider-Fachdienst
[gemSpec_Krypt]	gematik: Übergreifende Spezifikation: Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur
[gemSpec_OID]	gematik: Übergreifende Spezifikation: Festlegung von OIDs
[gemSpec_PKI]	gematik: Übergreifende Spezifikation: PKI
[gemSpec_Perf]	gematik: Übergreifende Spezifikation: Performance und Mengengerüst TI-Plattform

6.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[openid-connect-core]	OpenID Connect Core 1.0 (November 2014) https://openid.net/specs/openid-connect-core-1_0.html
[openid-connect-discovery]	OpenID Connect Discovery 1.0 (November 2014) https://openid.net/specs/openid-connect-discovery-1_0.html
[RFC6749]	The OAuth 2.0 Authorization Framework (Oktober 2012) https://tools.ietf.org/html/rfc6749
[RFC6750]	The OAuth 2.0 Authorization Framework: Bearer Token Usage (Oktober 2012) https://tools.ietf.org/html/rfc6750
[RFC7033]	Webfinger (September 2013) https://tools.ietf.org/html/rfc7033
[RFC7231]	Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content (Juni 2014) https://tools.ietf.org/html/rfc7231

[RFC7515]	JSON Web Signature (Mai 2015) https://tools.ietf.org/html/rfc7515
[RFC7516]	JSON Web Encryption (Mai 2015) https://tools.ietf.org/html/rfc7516
[RFC7519]	JSON Web Token (Mai 2015) https://tools.ietf.org/html/rfc7519
[RFC7523]	JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants (Mai 2015) https://tools.ietf.org/html/rfc7523
[RFC7636]	Proof Key for Code Exchange by OAuth Public Clients (September 2015) https://tools.ietf.org/html/rfc7636
[RFC8252]	OAuth 2.0 for Native Apps (Oktober 2017) https://tools.ietf.org/html/rfc8252