

Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation Identity Provider-Dienst

Version: 1.7.1
Revision: 1146631
Stand: 28.02.2025
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	30.06.2020		initiale Erstellung des Dokuments	gematik
1.1.0	12.10.2020		Einarbeitung Scope-Themen aus R4.0.1	gematik
1.1.1	13.11.2020		Einarbeitung P22.4	gematik
1.2.0	19.02.2021		Einarbeitung P22.5	gematik
1.3.0	14.06.2021		Einarbeitung IDP 2.2.0 (inkl. entsprechender Anteile aus gemF_Tokenverschlüsselung & gemF_Biometrie) und der Änderungsliste IDP_Maintenance_21.1	gematik
1.4.0	17.12.2021		Einarbeitung IDP 2.3.0 (inkl. entsprechender Anteile aus gemF_sektorale_IDP) und der Änderungsliste IDP_CR_Q4	gematik
1.5.0	01.09.2023		Einarbeitung IDP_Maintenance_23.4	gematik
1.5.1	05.09.2023	7.4, 7.6	Anpassung der Beispiele	gematik
1.6.0	15.03.2024		Einarbeitung IDP_24_1	gematik
1.6.1	18.07.2024		Einarbeitung Änderungen aus der gemF_eRp_DiGA	gematik
1.7.0	06.09.2024		Einarbeitung IDP_24_9	gematik
1.7.1	28.02.2025		Einarbeitung IDP_25_1	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes	6
1.1 Zielsetzung.....	6
1.2 Zielgruppe.....	6
1.3 Geltungsbereich.....	6
1.4 Abgrenzungen.....	7
1.5 Methodik.....	7
2 Systemüberblick	8
2.1 Allgemeiner Überblick.....	8
2.2 Detaillierter Überblick.....	9
3 Systemkontext	12
3.1 Akteure und Rollen.....	13
3.2 Begriffsdefinition.....	14
3.3 Verfahrensbeschreibung.....	16
3.4 Abweichende Verfahrensbeschreibung für Primärsysteme.....	20
3.5 Registrierung Anwendungsfrontend und Fachdienst.....	20
3.6 Anwendungsfrontend vorbereitende Maßnahmen.....	20
3.7 Anfrage von Token.....	20
3.8 Aufgaben des Authorization-Endpunktes.....	21
3.8.1 Erstellung des AUTHORIZATION_CODE.....	21
3.9 Einreichen des AUTHORIZATION_CODE.....	21
3.10 Aufgabe des Token-Endpunktes.....	21
3.11 Einreichen des ACCESS_TOKEN beim Fachdienst.....	22
3.12 Aufgabe des Fachdienstes.....	22
4 Zerlegung des Produkttyps	23
4.1.1 Allgemeine Sicherheitsanforderungen.....	23
4.1.2 Sicherheit der Netzübergänge.....	24
4.2 Fehlermeldungen.....	24
4.3 Schnittstellenbeschreibung des IDP-Dienstes.....	25
4.4 Identifikation des Clientsystems.....	26
4.5 Unterstützung der Anmeldung mittels sektoraler Identity Provider.....	27
5 Funktionsmerkmale	28
5.1 Authorization Server Metadata (Discovery Document).....	28
5.1.1 Aufbau des Discovery Documents.....	28

5.1.2 Erneuerung des Discovery Documents.....	29
5.1.3 Schutz des Discovery Document.....	30
5.1.4 Auswahlliste der sektoraler Identity Provider für den Nutzer.....	30
5.2 Authorization-Endpunkt.....	31
5.2.1 Authorization-Endpunkt Eingangsdaten.....	32
5.2.2 Authorization-Endpunkt Ausgangsdaten.....	35
5.3 Token-Endpunkt.....	36
5.3.1 Token-Endpunkt Eingangsdaten.....	37
5.3.2 Token-Endpunkt Ausgangsdaten.....	37
5.4 Pairing-Endpunkt.....	41
5.4.1 Zielsetzung.....	41
5.4.2 Technisches Konzept.....	48
5.4.2.1 <i>Bewertung von Gerätetypen</i>	48
5.4.2.1.1 Spezifikation.....	48
5.4.2.2 <i>Erweiterung des IDP-Dienstes um einen Pairing-Endpunkt</i>	50
5.4.2.2.1 Spezifikation.....	50
5.4.2.3 <i>Daten und Kommunikationsstrukturen</i>	51
5.4.2.4 <i>Nomenklatur Schlüsselmaterial</i>	51
5.4.2.5 <i>Registrierung von alternativen Authentisierungsmitteln</i>	53
5.4.2.5.1 Erläuterungen zum Registrierungsprozess.....	54
5.4.2.5.2 Spezifikation.....	56
5.4.2.6 <i>Verwendung von alternativen Authentisierungsmitteln</i>	59
5.4.2.6.1 Erweiterung des Authorization-Endpunkts.....	59
5.4.2.6.2 Erläuterungen zur Verwendung von alternativen Authentisierungsmitteln.....	60
5.4.2.6.3 Spezifikation.....	62
5.4.2.7 <i>Inspektion und De-Registrierung der am Pairing-Endpunkt gespeicherten Daten</i>	64
5.4.2.7.1 Inspektion am Pairing-Endpunkt.....	64
5.4.2.7.2 Deregistrierung von alternativen Authentisierungsmitteln.....	65
5.4.2.7.3 Spezifikation.....	66
5.5 Authentisierung über andere IDP.....	67
5.5.1 Authentifizierung über andere IDP - Eingangsdaten.....	69
5.5.2 Authentifizierung über andere IDP - Ausgangsdaten.....	70
6 Anhang A - Verzeichnisse.....	73
6.1 Abkürzungen.....	73
6.2 Glossar.....	74
6.3 Abbildungsverzeichnis.....	77
6.4 Tabellenverzeichnis.....	77
6.5 Referenzierte Dokumente.....	78
6.5.1 Dokumente der gematik.....	78
6.5.2 Weitere Dokumente.....	79
7 Anhang B - Beispiele der Objekte und Aufrufe.....	81
7.1 Authorization Request.....	81

7.2 Authorization Response	82
7.3 Authentication Request	83
7.3.1 Authentication Request (SSO).....	85
7.4 Authentication Response	85
7.4.1 Authentication Response (SSO Flow).....	88
7.5 Token Request	88
7.6 Token Response	89
7.7 Aufbau des Discovery Document	92
8 Anhang C - Datenformate und Kommunikationsprotokolle im Zusammenhang mit alternativen Authentisierungsverfahren	94
8.1 Datentypen	94
8.1.1 Übergeordnete Anforderungen.....	94
8.1.1.1 Kodierung.....	94
8.1.1.2 Versionierung.....	94
8.1.1.3 Namen und Claims.....	94
8.1.2 Datentyp "Device_Type".....	95
8.1.3 Datentyp "Device_Information".....	95
8.1.4 Datentyp "Pairing_Data".....	96
8.1.5 Datentyp "Signed_Pairing_Data".....	98
8.1.6 Datentyp "Registration_Data".....	99
8.1.7 Datentyp "Encrypted_Registration_Data".....	99
8.1.8 Datentyp "Authentication_Data".....	100
8.1.9 Datentyp "Signed_Authentication_Data".....	101
8.1.10 Datentyp "Encrypted_Signed_Authentication_Data".....	101
8.1.11 Datentyp "Pairing_Entry".....	102
8.1.12 Datentyp "Pairing_Entries".....	103
8.2 Ausgestaltung der Kommunikation mit dem IDP-Dienst	103
8.2.1 Registrierung von alternativen Authentisierungsmitteln.....	103
8.2.1.1 Request des Authenticator-Moduls bei Registrierung.....	103
8.2.1.2 Response des Pairing-Endpunkts bei Registrierung.....	104
8.2.2 Verwendung von alternativen Authentisierungsmitteln.....	105
8.2.2.1 Request des Authenticator-Moduls bei Verwendung von alternativen Authentisierungsmitteln.....	105
8.2.2.2 Response des Authorization-Endpunkts bei Verwendung von alternativen Authentisierungsmitteln.....	105
8.2.3 Inspektion von Pairing-Daten am Pairing-Endpunkt.....	106
8.2.3.1 Request des Authenticator-Moduls zur Inspektion.....	106
8.2.3.2 Response des Pairing-Endpunkts bei Inspektion.....	106
8.2.4 Deregistrierung von Pairing-Daten am Pairing-Endpunkt.....	107
8.2.4.1 Request des Authenticator-Moduls zur Deregistrierung.....	107
8.2.4.2 Response des Pairing-Endpunkts bei Deregistrierung.....	108
8.3 Vergleichsoperationen	108
8.3.1 Registrierung.....	108

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 Authentifizierung ist die Voraussetzung, damit ein Nutzer über das Anwendungsfrontend Zugang zu Fachdaten eines Fachdienstes erlangen kann. Der IDP-Dienst verwaltet und steuert den Smartcard-basierten Authentifizierungsprozess für verschiedene Anwendungen und stellt diesen nach erfolgter Authentisierung ein ID_TOKEN mit den Identitätsdaten des Nutzers bereit.

Darüber hinaus agiert der IDP-Dienst auch als OAuth2 Authorization Server für das E-Rezept und stellt in dieser Funktion über die reine Authentisierung hinaus mittels ACCESS_TOKEN auch die Authorisierung der Zugriffe eines Anwendungsfrontend auf den E-Rezept-Fachdienst sicher.

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. gemPTV_ATV_Festlegungen, 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 Schutzrechtshabern 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. Außerhalb des TI-Netzes können TLS-Zertifikate von den Mitgliedern des CA/Browser Forum als übliche Anbieter solcher Zertifikate bezogen werden. Diese sind über Standardmechanismen prüfbar und erfordern keine besondere Behandlung der PKI. Wenn notwendig werden weitere Schutzmechanismen wie Zertifikat-Pinning, Certification Authority Authorization (CAA) Records oder ein Monitoring mittels Certificate Transparency (CT) vorgesehen.

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 Punkte 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 für diesen Nutzer angebotenen Daten erhalten. Der IDP-Dienst stellt durch gesicherte JSON Web Token (JWT) attestierte Identitäten bereit.

Dabei erfüllt der IDP-Dienst zwei unterschiedliche Rollen.

1. Authorization Server für das E-Rezept

Gegen Vorlage eines ACCESS_TOKEN, des für die Fachanwendung zuständigen Authorization Server, erhalten Anwendungsfrontends – entsprechend der im Token attestierten Parameter – Zugriff auf die Inhalte des Fachdienstes. Der IDP-Dienst erfüllt diese Funktion für den E-Rezept-Fachdienst und stellt dabei den verschiedenen Anwendungsfrontends (Primärsysteme in Praxen und Apotheken sowie die E-Rezept App) zeitlich begrenzte ACCESS_TOKEN für den Zugang aus. In dieser Funktion bindet der IDP-Dienst für die Authentisierung von Versicherten auch weitere Identity Provider an, welche im Rahmen der TI-Föderation verschiedene Verfahren für die Anmeldung des Nutzers bieten. Die Versicherten können auf diesem Weg mit demselben Authentisierungsverfahren auf verschiedene Anwendungen der TI und ihrer Krankenkasse zugreifen.

2. Identity Provider

Der IDP-Dienst übernimmt darüber hinaus für verschiedene Fachdienst die Aufgabe der smartcard-basierten Authentisierung des Nutzers. Der IDP-Dienst fasst eine eindeutige IdNummer sowie weitere für den Fachdienst notwendige Attribute in signierten JSON Web Token (ID_TOKEN) zusammen. Fachdienste müssen so keine Überprüfung des Nutzers selbst implementieren, sondern können sich darauf verlassen, dass der Besitzer des bei ihnen vorgetragenen ID_TOKEN bereits authentisiert wurde. Des Weiteren stellt der IDP-Dienst sicher, dass die vom Nutzer vorgetragenen Attribute (aus dem Signaturzertifikat einer Smartcard) gültig sind.

Der IDP-Dienst prüft in dieser Funktion, ob das in einem Challenge-Response Prozess vorgetragene X.509-nonQES-Signatur-Zertifikat der verwendeten Prozessor-Chipkarte (eGK, HBA oder SMC-B) zeitlich gültig und ob seine Integrität sichergestellt ist.

Der IDP-Dienst stellt dann ID_TOKEN für den anfragenden Fachdienst aus, welche auf den Inhalten der gültigen AUT-Zertifikate (d.h. C.CH.AUT, C.HP.AUT oder C.HCI.AUT) basieren.

2.2 Detaillierter Überblick

Diese Beschreibung deckt bisher nur die Identity Provider Rolle des IDP-Dienstes ab. Eine genauere Beschreibung und ggf. eine Grafik für die Rolle als Authorization Server innerhalb der Föderation wird noch ergänzt.

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](#)] oder ggf. einem SSO_TOKEN basierend auf [[RFC7519](#)] aus. Gewählt wird für alle Prozesse 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](#)]).

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.

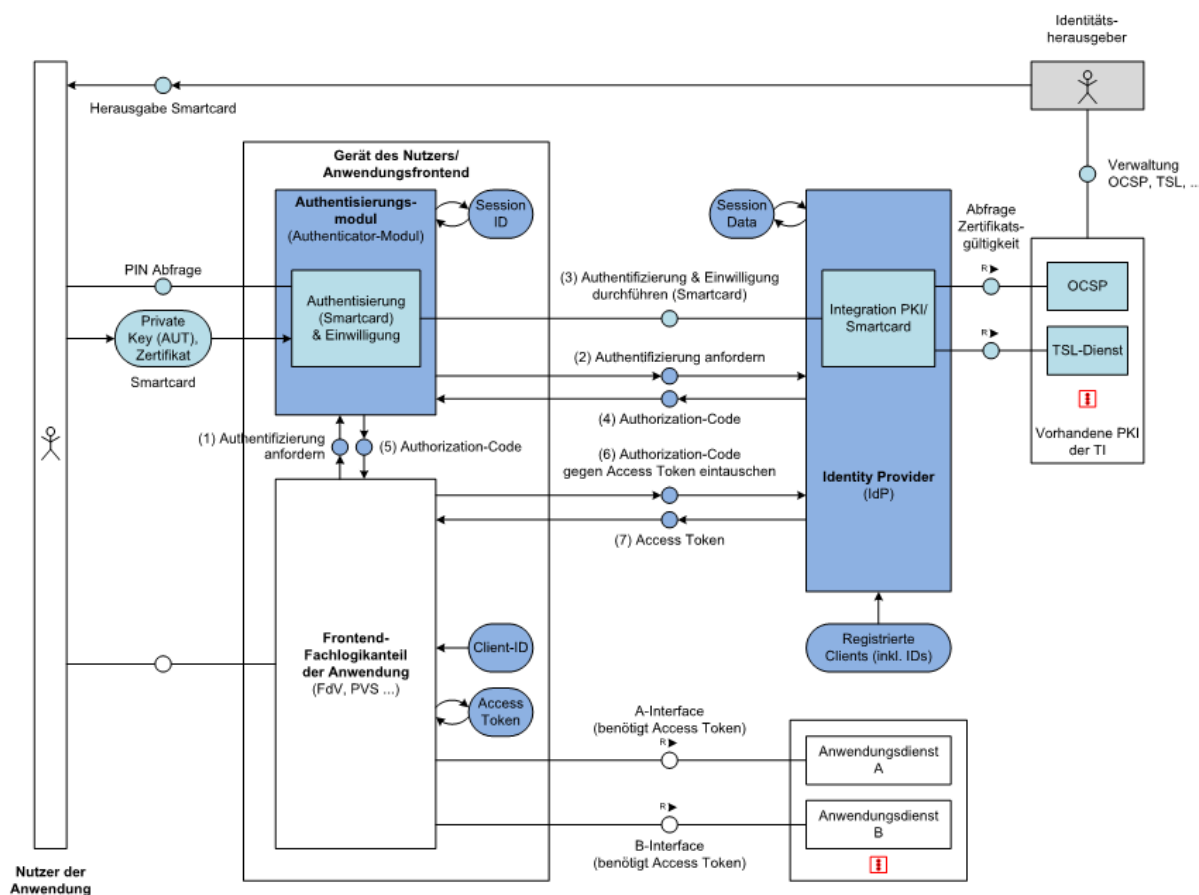


Abbildung 1: Übersichtsschaubild OAuth2.0 IDP-Dienst

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 Authorization-Endpunkt bedient.

Die Teilschritte (6) und (7) werden durch den Token-Endpunkt [RFC6749 # section-3.2] bedient.

Der hier gezeigte 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

Diese Beschreibung deckt bisher nur die Identity Provider Rolle des IDP-Dienstes ab. Eine genauere Beschreibung und ggf. eine Grafik für die Rolle als Authorization Server innerhalb der Föderation wird noch ergänzt.

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 SSO Token kommt nur beim E-Rezept FdV zum Einsatz und erfüllt hier die Funktion eines Refresh-Token.

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

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

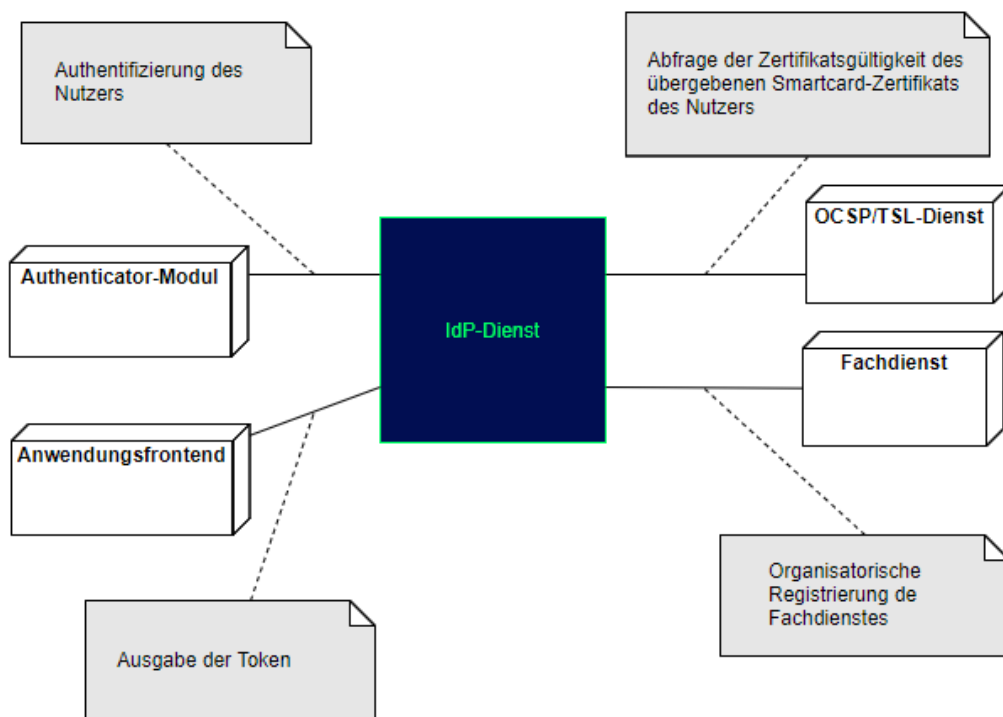


Abbildung 2: Systemkontext aus Sicht des IDP-Dienstes

3.1 Akteure und Rollen

Diese Beschreibung deckt bisher vornehmlich die Rolle als Authorization Server ab. Eine genauere Beschreibung für die Rolle als Identity Provider wird noch ergänzt.

Im Systemkontext des IDP-Dienstes interagieren verschiedene Akteure (Nutzer und aktive Komponenten) in unterschiedlichen OAuth2-Rollen gemäß [[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 Schnittstellen 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 extern genutzten Schlüssel und Adressen des IDP-Dienstes

	PUK	private Key	Adressen
Authorizati on- Endpunkt (AUTH)	PuK_IDP_SIG - für die Signaturprüfung desCHALLENGE_TOKEN, des AUTHORIZATION_CODE und des SSO_TOKEN - kodiert in einem FD.SIG-Zertifikat PuK_IDP_ENC - für die Verschlüsselung der signierten Challenge durch das Authenticator-Modul PuK_IDP_SIG_Sek - Wird für die Authentisierung gegenüber einem sektoralen IDP dort registriert PuK_IDP_TLS - Wird für die Authentisierung	PrK_IDP_SIG - zum Signieren des CHALLENGE_TOKEN des AUTHORIZATION_CODE und des SSO_TOKEN PrK_IDP_ENC - zum Entschlüsseln der signierten Challenge PrK_IDP_SIG_Se	URI_AUTH (authorization_end point) URI_AUTH_SSO (sso_endpoint) URI_AUTH_3RD (third_party_endpoi nt)

	innerhalb der Föderation registriert	<p>k</p> <ul style="list-style-type: none"> - Für die Authentisierung gegenüber einem sektoralen IDP <p>PrK_IDP_TLS</p> <ul style="list-style-type: none"> - Wird für die TLS Authentisierung innerhalb der Föderation verwendet 	
Discovery-Endpunkt (DISC)	<p>PuK_DISC_SIG</p> <ul style="list-style-type: none"> - für die Signaturprüfung des Discovery Document - kodiert in einem FD.SIG-Zertifikat 	<p>PrK_DISC_SIG</p> <ul style="list-style-type: none"> - zum Signieren des Discovery Document 	URI_DISC
Token-Endpunkt (TOKEN)	<p>PuK_IDP_SIG</p> <ul style="list-style-type: none"> - für die Signaturprüfung des ID_TOKEN und des ACCESS_TOKEN - kodiert in einem FD.SIG-Zertifikat <p>PuK_IDP_ENC</p> <ul style="list-style-type: none"> - für die Verschlüsselung des KEY_VERIFIER durch das Anwendungsfrontend 	<p>PrK_IDP_SIG</p> <ul style="list-style-type: none"> - zum Signieren des ID_TOKEN und des ACCESS_TOKEN <p>PrK_IDP_ENC</p> <ul style="list-style-type: none"> - für die Entschlüsselung des ACCESS_TOKEN für den Pairing-Vorgang 	URI_TOKEN (token_endpoint)
Pairing-Endpunkt (PAIR)	<p>PuK_IDP_ENC</p> <ul style="list-style-type: none"> - für die Verschlüsselung des ACCESS_TOKEN für den Pairing-Vorgang 	<p>PrK_IDP_ENC</p> <ul style="list-style-type: none"> - für die Entschlüsselung des ACCESS_TOKEN für den Pairing-Vorgang 	URI_PAIR

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 Document `URI_DISC` stellt somit den zentralen Anlaufpunkt dar, anhand dessen alle weiteren „statischen“ Dienste (Endpunkte des IDP-Dienstes) adressiert werden können.

Hinweis: Bei allen extern genutzten Schlüsseln handelt es sich um ECC-Schlüsselpaare der Kurve `brainpoolP256r1`. Für `IDP_ENC` ist im Gegensatz zu den anderen beiden Schlüsseln keine Bestätigung als Zertifikat vorgesehen. Die maximale Einsatzdauer des Schlüsselpaars liegt analog zum `IDP_SIG` und `DISC_SIG` bei maximal 5 Jahren.

3.3 Verfahrensbeschreibung

Diese Beschreibung deckt bisher vornehmlich die Rolle als Authorization Server ab. Eine genauere Beschreibung dazu, wie Requests von Anwendungsfrontends (e-Rezept) oder Fachdiensten (Funktion IDP) kommend verarbeitet werden und dann entweder das `ACCESS_TOKEN` oder das `ID_TOKEN` verwendet wird, folgt.

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.

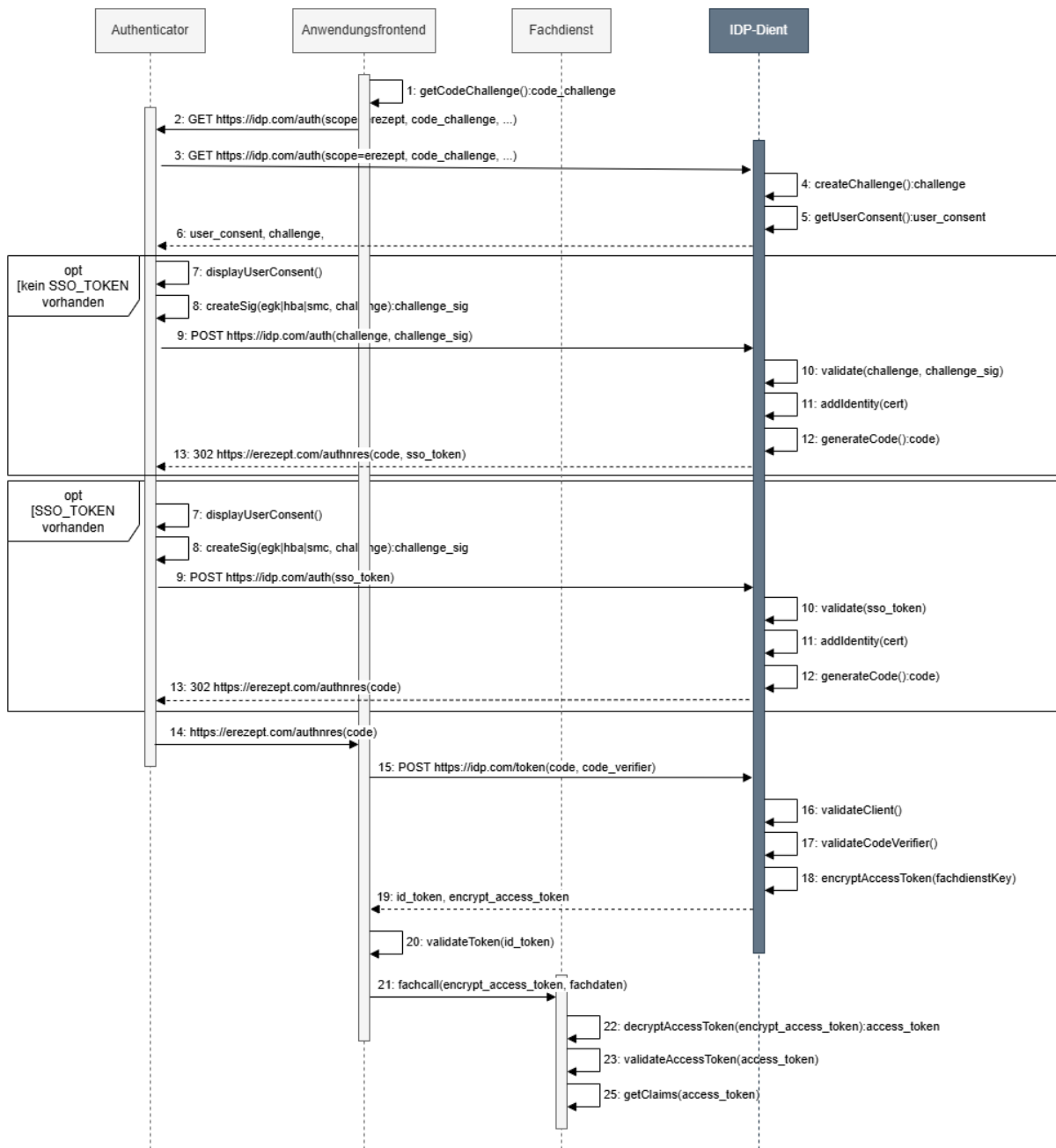


Abbildung 3: Datenfluss-Diagramm IDP-Dienst

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

1. Das Anwendungsfrontend erzeugt sich einen CODE_VERIFIER [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 seinen Authorization Request inklusive der generierten Werte CODE_CHALLENGE, State und Nonce gemäß [RFC8252 # Anhang B] an das Authenticator-Modul.

3. Das Authenticator-Modul überträgt den Authorization Request weiter an den Authorization-Endpunkt des IDP-Dienstes. Der Authorization-Endpunkt prüft die Inhalte des Authorization Request wie etwa die ClientID.
4. Der Authorization-Endpunkt stellt alle Informationen zusammen und erzeugt die CHALLENGE.
5. Der Authorization-Endpunkt stellt den mit dem entsprechenden Fachdienst vereinbarten Consent (Zustimmung des Nutzers zur Verarbeitung der angezeigten Daten) zusammen.
6. Der Authorization-Endpunkt überträgt CHALLENGE_TOKEN und Consent-Abfrage USER_CONSENT zum Authenticator-Modul.
7. 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.
8. Das Authenticator-Modul verwendet die PIN um die CHALLENGE_TOKEN von der Smartcard signieren zu lassen. Falls bereits ein SSO_TOKEN beim Authenticator-Modul existiert, entfällt dieser Schritt.
9. Das Authenticator-Modul überträgt das signierte CHALLENGE_TOKEN mit dem Smartcard-Zertifikat, verschlüsselt mittels PuK_IDP_ENC, an den IDP-Dienst (Antwort Schritt 7). Falls ein SSO_TOKEN beim Authenticator-Modul existiert, wird dieser Token zusammen mit einem unveränderten CHALLENGE_TOKEN zum IDP-Dienst transportiert.
10. Der Authorization-Endpunkt entschlüsselt und validiert die signierte Challenge. 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_IDP_SIG durch.
11. Der Authorization-Endpunkt erstellt den AUTHORIZATION_CODE.
12. Der Authorization-Endpunkt überträgt den AUTHORIZATION_CODE und den SSO_TOKEN an das Authenticator-Modul (Antwort Schritt 3). Falls 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_IDP_SIG signiert. Der Authorization-Endpunkt verschlüsselt das SSO_TOKEN für sich und den AUTHORIZATION_CODE für den Token-Endpunkt. Das zur Verschlüsselung verwendete Schlüsselmaterial muss die Vorgaben der [gemSpec_Krypt] beachten.
13. Das Authenticator-Modul überträgt den AUTHORIZATION_CODE an das Anwendungsfrontend (Antwort Schritt 2).
14. Das Anwendungsfrontend erzeugt sich einen AES256-"Token-Key", verknüpft ihn mit dem CODE_VERIFIER zum KEY_VERIFIER und sendet diesen unter Nutzung des öffentlichen Schlüssels PUK_IDP_ENC verschlüsselt zusammen mit dem AUTHORIZATION_CODE zum Token-Endpunkt des IDP-Dienstes.
15. Der Token-Endpunkt entschlüsselt den AUTHORIZATION_CODE und validiert ihn anhand des öffentlichen Schlüssels PUK_IDP_SIG des Authorization-Endpunktes.
16. Der Token-Endpunkt entschlüsselt und validiert den KEY_VERIFIER, entnimmt aus diesem den CODE_VERIFIER und gleicht diesen mit der CODE_CHALLENGE aus dem AUTHORIZATION_CODE ab.

17. Der Token-Endpunkt erzeugt die erforderlichen Token, signiert sie mit seinem privaten Schlüssel PrK_IDP_SIG und verschlüsselt sie mit dem „Token-Key“ des Anwendungsfrontends, welchen er dem KEY_VERIFIER entnimmt.
18. Der Token-Endpunkt überträgt die Token an das Anwendungsfrontend (Antwort Schritt 16).
19. Das Anwendungsfrontend entschlüsselt die Token mit seinem „Token-Key“ und prüft die Token-Signatur anhand des öffentlichen Schlüssels PUK_IDP_SIG des Token-Endpunktes.
20. Das Anwendungsfrontend reicht das gültige ACCESS_TOKEN auf Anwendungsebene verschlüsselt beim Fachdienst ein.
21. Der Fachdienst entschlüsselt das ACCESS_TOKEN entsprechend dem für diese Anwendung vorgesehenen Verfahren.
22. Der Fachdienst validiert das ACCESS_TOKEN anhand des öffentlichen Schlüssels PUK_TOKEN des Token-Endpunktes.
23. 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 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 und ACCESS_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. Das SSO_TOKEN erfüllt hier für das E-Rezept-FdV die Funktion eines Refresh-Token. 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-Endpunktes 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

A_20737 - Ermöglichung einer organisatorischen Registrierung für Anwendungsfrontends und Fachdienste

Der Anbieter des IdP-Dienstes MUSS eine organisatorische Registrierung von Anwendungsfrontends und Fachdiensten ermöglichen. [<=]

Um ein Anwendungsfrontend nutzen zu können, muss dieses beim IDP-Dienst registriert sein. Die Registrierung des Anwendungsfrontends ist im Dokument [gemSpec_IDP_Frontend] beschrieben.

Anbieter von Fachdiensten müssen ebenfalls die Registrierung ihres Fachdienstes über einen organisatorischen Prozess beim IDP-Dienst durchführen.

Ergänzung: Diese Registrierung erfolgt einmalig für die Anwendung bzw. den Dienst und muss bei Updates nicht 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 einen CODE_VERIFIER (Zufallswert) gemäß [[RFC7636 # section-4.1](#)] und hierüber einen Hash, die CODE_CHALLENGE, gemäß [[RFC7636 # section-4.2](#)] mit dem Algorithmus S256 gemäß [[RFC7636 # section-4.2](#)] erzeugen.

3.7 Anfrage von Token

Die folgende Anfrage an den Authorization-Endpunkt umfasst die Schritte 1-3 aus dem Gesamtablauf des Kapitels 3.3. Der Request wird hierbei entweder vom Anwendungsfrontend (Funktion Authorization Server) oder dem Fachdienst (Funktion Identity Provider) generiert. Die Adressierung des IDP-Dienstes ist als Parameter in einer Konfigurationsdatei oder direkt im Quellcode hinterlegt.

Die Anfrage wird dann über das Authenticator-Modul an den Authorization-Endpunkt des IDP-Dienstes geleitet.

Inhalt der Anfrage ist:

- die `redirect_uri` sowie der *Scope* (Attributumfang) der Anfrage,
- die `client_id` als Identifikation des Anfragenden Systems,
- die eigene Hersteller-ID, Programm Kürzel und Versionsnummer,
- der über den eigenen 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,
- der nonce-Parameter [openid-connect-core#Authentication Request](#)] kann zusätzlich genutzt werden um den Erhalt des korrekten ID_TOKEN zu verifizieren.

Details siehe [7.1- Authorization Request].

3.8 Aufgaben des Authorization-Endpunktes

Der Authorization-Endpunkt nimmt die Anfrage an und überprüft ob `client_id` und `scope` bekannt und in dieser Kombination zulässig sind.

3.8.1 Erstellung des AUTHORIZATION_CODE

Im Fall einer Authentisierung mittels Smartcard sendet der Authorization-Endpunkt eine CHALLENGE an das Authenticator-Modul und prüft anschließend die Signatur der CHALLENGE und das mitgelieferte Zertifikat der Smartcard des Nutzers mittels OCSP/TSL der PKI der Telematikinfrastruktur, bevor er die benötigten Attribute ausliest.

Wenn der IDP-Dienst als Authorization Server für das E-Rezept agiert, kann er alternativ einen eigenen Request an den durch den Nutzer gewählten sektoralen Identity Provider formulieren und bekommt von diesem nach erfolgter Authentisierung des Nutzers ein ID_TOKEN ausgestellt, welchem er die Attribute entnimmt.

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.

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 Anfrage des Anwendungsfrontends bzw. Fachdienstes 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-Wert 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 / ACCESS_TOKEN an das anfragende System 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

Wenn der IDP_Dienst als Authorization Server agiert, nimmt der Fachdienst das zwischen Frontend und Fachdienst anwendungsspezifisch verschlüsselte ACCESS_TOKEN entgegen. Danach überprüft er die Integrität und die Übereinstimmung mit den benötigten Claims. Enthält das ACCESS_TOKEN mehr oder weniger Attribute, als im Scope vereinbart, oder sind diese fehlerhaft 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.

Wenn der IDP-Dienst als reiner Identity Provider agiert, so verwendet der Fachdienst den ID_TOKEN zur Identifikation des gegenüber dem IDP-Dienst authentisierten Nutzer.

Hierbei können auch ACCESS_TOKEN eines eigenen Anwendungsspezifischen Authorization Server zum Einsatz kommen.

4 Zerlegung des Produkttyps

A_20687-01 - Bereitstellung der öffentlichen Schlüsselteile

Der Authorization Server MUSS zu allen verwendeten privaten Schlüsseln PrK_IDP_SIG, PrK_IDP_ENC und PrK_DISC_SIG das öffentliche Pendant PuK_IDP_SIG, PuK_IDP_ENC und PuK_DISC_SIG zum Download bereitstellen. Dies ermöglicht die Prüfung der von den einzelnen Schnittstellen vorgenommenen Signaturen ebenso wie die zielgerichtete Verschlüsselung des Payload für den bestimmten Empfänger. [≤]

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_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. [≤]

A_21302 - Interner Datenaustausch der Komponenten des IdP-Dienstes

Der Anbieter des IdP-Dienstes MUSS für den internen Datenaustausch einen Mechanismus zur Sicherung der Datenintegrität, der Authentizität und zur Vertraulichkeit der Daten zur Verfügung stellen. [≤]

4.1.2 Sicherheit der Netzübergänge

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. [≤]

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_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_20585-01 - IdP-Dienst-Anbieter - Richtlinien für den Paketfilter zum Internet

Der Anbieter des IdP-Dienstes MUSS beim Paketfilter die Weiterleitung von IP-Paketen an der Schnittstelle zum Internet auf das HTTPS- Protokoll beschränken. [≤]

A_20586-01 - IdP-Dienst - Verhalten bei Vollaustlastung

Der Anbieter des IdP-Dienstes MUSS den Paketfilter des IdP-Dienstes so konfigurieren, dass bei Vollaustlastung 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_20682-01 - 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. [≤]

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.

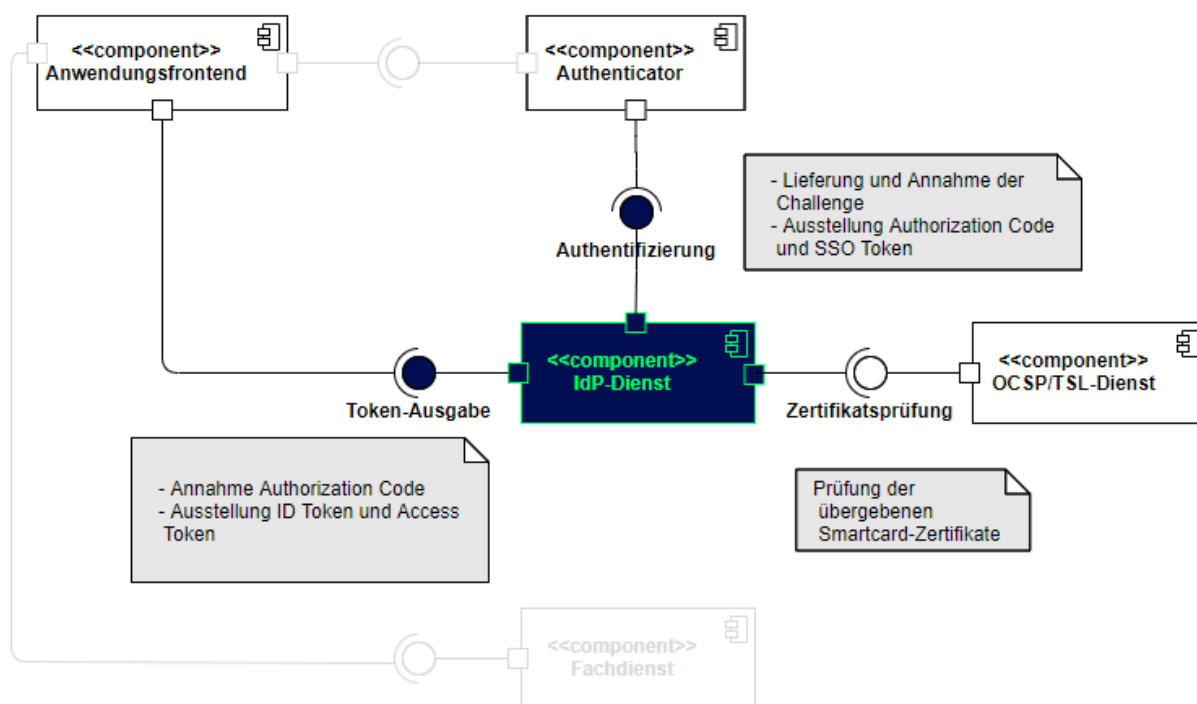


Abbildung 4: Schnittstellen des IDP-Dienstes

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 Anwendungsfondend 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.

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

A_20588-01 - 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 403 beantworten, damit in der Betriebsüberwachung des IdP-Dienstes die Nutzung unzulässiger Clientsysteme erkannt werden kann. [<=]

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_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_22511 - Ausschluss von Authenticator Modul-Versionen am zentralen Identity Provider (Rohdatenerfassung v.02)

Der zentrale Identity Provider MUSS den Verbindungsversuch, der nach A_20590 ausgeschlossenen Kommunikation in den Rohdaten protokollieren. [<=]

A_20742-01 - Vergabe der "client_id" durch den Anbieter des IdP-Dienstes

Der Anbieter des IDP-Dienstes MUSS bei der organisatorischen Registrierung des Anwendungsfrontends diesem eine eindeutige `client_id` zur Nutzung des IDP-Dienstes zuweisen. [≤]

A_21472 - SSO_TOKEN nur für Mobile Endgeräte und nicht für Primärsysteme

Der IDP-Dienst MUSS bei der organisatorischen Registrierung eines Clients anhand seiner `client_id` - beispielsweise des Anwendungsfrontends oder eines Primärsystems - festlegen, ob im Authorization Code Flow bei der Ausgabe eines AUTHORIZATION_CODE auch ein SSO_TOKEN vom IDP-Dienst ausgegeben werden muss. [≤]

4.5 Unterstützung der Anmeldung mittels sektoraler Identity Provider

Im Rahmen der Zugänglichmachung des E-Rezepts für Versicherte ohne NFC-fähige eGK sollen ab dem 01.01.2022 Kassen-eigene Authentifizierungssysteme an das E-Rezept angebunden werden. Diese werden im Folgenden als sektorale Identity Provider bezeichnet, um sie vom zentralen IDP-Dienst abzugrenzen, welcher die Authentisierung der Nutzer direkt oder indirekt über die eGK realisiert und die bereitgestellten Attribute von dieser ableitet.

Zur Einbindung eines sektoralen Identity Provider wird der zentrale IDP-Dienst als Authorization Server zwischen dem Anwendungsfrontend, den sektoralen Identity Providern und dem E-Rezept Fachdienst eingesetzt.

Die Realisierung der ersten Stufe der sektoralen Identity Provider auf dem Vertrauensniveau "substanziell" ist auf den 31.12.2023 befristet. Dies wird mit der Etablierung einer Föderation von Identity Providern abgelöst. Diese erlauben die Authentisierung von Versicherten. Der IDP-Dienst wird in seiner Rolle als Authorization Server des E-Rezept Fachdienstes in dieser Föderation registriert und leitet Authentisierungsanfragen an diese weiter. Parallel dazu werden die zuvor etablierten Prozesse zur Anbindung der ersten Stufe der sektoralen Identity Provider bis zum Ende ihrer Befristung unterstützt. Die hierfür relevanten Anforderungen werden in diesem Dokument durch den Zusatz - "Befristet -" im Titel der Anforderung kenntlich gemacht.

Die für die Funktion des IDP-Dienstes innerhalb der Föderation notwendigen Anforderungen aus `gemSpec_IDP_FD` finden sich in seinem Produkttypsteckbrief.

Weitere Aspekte finden sich in den Kapiteln der jeweiligen Endpunkte.

Hinweis: Der sektorale Identity Provider wird bei Zulassung vermerkt und der zentrale IDP-Dienst geht dann im Auftrag der gematik auf diesen zwecks Registrierung zu.

5 Funktionsmerkmale

5.1 Authorization Server Metadata (Discovery Document)

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.[<=]

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 vertrauenswürdiger 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_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.[<=]

Hinweis 1: Das Discovery Document innerhalb der TI adressiert hierbei die URI der 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 2: 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_20689-01 - Internes Discovery Document - Prüfung der angebotenen URL

Der IdP-Dienst MUSS alle von ihm im internen Discovery Document angebotenen URL ständig auf bloße Erreichbarkeit prüfen.[<=]

A_20690-01 - Externes Discovery Document - Prüfung der angebotenen URL

Der IdP-Dienst MUSS alle von ihm im externen Discovery Document angebotenen URL ständig auf bloße Erreichbarkeit prüfen.[<=]

5.1.1 Aufbau des Discovery Documents

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_20458-02 - Inhalte des Discovery Document

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

- "issuer" (hier ist der IdP-Dienst erreichbar)
- "jwks_uri" (für den Abruf von „PUK_IDP_ENC“ sowie des öffentlichen Schlüssels und des Zertifikats von „PUK_IDP_SIG“ entsprechend TAB_IDP_DIENST_0003 [RFC7517] - identifiziert anhand der „kid“-Parameter (puk_idp_enc / puk_idp_sig)
- "uri_disc" (URI, unter welcher das Discovery Document bereitgestellt wird)
- "authorization_endpoint" (URI des Dienstes und des öffentlichen Verschlüsselungsschlüssels des Authorization-Endpunktes gemäß [RFC6749])
- "sso_endpoint" (URI des Authorization-Endpunktes für Requests mit SSO-Token)
- "auth_pair_endpoint" (URI des Authorization-Endpunktes für Requests mit Pairing-Daten)
- "token_endpoint" (URI des Token-Endpunktes gemäß [RFC6749])
- "uri_puk_idp_enc" und „uri_puk_idp_sig“ (URI der JWK Objekte für die zwei Schlüssel und des Zertifikates).

[<=]

Hinweis: Der genaue Aufbau entspricht [gemSpec_IDP_Dienst#Kapitel 7.7 Aufbau des Discovery Document].

A_23680 - Liste der föderierten IDPs im Discovery Document des IDP-Dienst

Der IDP-Dienst MUSS im Discovery Document unter dem Schlüssel "fed_idp_list_uri" die URL zur Liste der aktuell in der Föderation zur Auswahl stehenden sektoralen Identity Provider publizieren.[<=]

A_23898 - Veröffentlichung des Federation Authorization Endpoint

Der IDP-Dienst MUSS die URL des angebotenen Federation Authorization Endpoint im Discovery Document unter dem Schlüssel "federation_authorization_endpoint" publizieren.[<=]

5.1.2 Erneuerung des Discovery Documents

A_20691-01 - 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_SIG signieren und am mit der gematik vereinbarten Downloadpunkt URI_DISC bereitstellen.[<=]

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.

5.1.3 Schutz des Discovery Document

A_19874-05 - 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_OID # Abschnitt 3.5.4] 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>". [\leq]

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

A_19877-04 - 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_OID # Abschnitt 3.5.4] signiert und TLS-gesichert im Internet zum Download bereitstellen.

Die URL des Downloadpunktes lautet: "<https://idp.app.ti-dienste.de/well-known/openid-configuration>"

[\leq]

Hinweis: Die für die Rolle des IDP-Dienstes vorgesehene professionOID ist in [gemSpec_OID] beschrieben.

A_20591-01 - Festlegungen zur Signatur der Discovery Documents

Der IdP-Dienst MUSS die Signatur der Discovery Documents dabei durch die Verwendung einer JSON Web Signature (JWS) [RFC7515 # section-3 - Compact Serialization] und des Schüssels PrK_DISC_SIG gewährleisten. Als Algorithmus ist dementsprechend "BP256R1" zu wählen.

Der IdP-Dienst MUSS bei der Signaturerstellung das Signaturzertifikat des PUK_DISC_SIG im x5c Claim einbetten. [\leq]

5.1.4 Auswahlliste der sektoraler Identity Provider für den Nutzer

A_23681-02 - Bereitstellung einer Liste der sektoralen Identity Provider innerhalb der Föderation

Der IDP-Dienst MUSS, entsprechend [gemSpec_IDP_FD#AF_10116], unter einer dedizierten URL eine Liste, der in der TI-Föderation registrierten sektoralen Identity Providern bereitstellen. Diese Liste muss der IDP-Dienst unter der URL, welche im Entity Statement des Federation Master unter dem claim idp_list_endpoint angegeben ist, laden (siehe [A_10100*]).

Der IDP-Dienst MUSS diese Daten verwenden, um für jeden sektoralen IDP dieser Liste einen Eintrag für die eigene bereitzustellende Liste mit folgenden Daten zu erstellen:

Datum	Erläuterung	Anmerkung
idp_name	Name der Krankenkasse, zur Anzeige in der E-Rezept App (max. 128 Zeichen)	Entspricht dem Wert in idp_entity.organization_name im Listeneintrag der Liste des Federation Master
idp_iss	URL des claim iss des sektoralen IDP der Föderation (maximal 2000 VSCAHR)	Entspricht dem Wert in idp_entity.iss im Listeneintrag der Liste des Federation Master
idp_sek_2	Boolean True für einen IDP innerhalb der	

	Föderation False für einen IDP im befristet zulässigen Funktionsumfang	
idp_pkv	Boolean False für gesetzliche Krankenversicherungen True für private Krankenversicherungen.	Entspricht dem Wert in idp_entity.pkv im Listeneintrag der Liste des Federation Master
idp_logo	Logo zur Anzeige in der Auswahlliste	Entspricht dem Wert in idp_entity.logo_uri im Listeneintrag der Liste des Federation Master

[<=]

A_23902 - Regelmäßige Aktualisierung der Liste der sektoralen Identity Provider der Föderation

Der IDP-Dienst SOLL die Liste der sektoralen Identity Provider der Föderation täglich aktualisieren.

[<=]

A_23683 - Festlegungen zur Signatur der Liste der sektoralen Identity Providern innerhalb der Föderation

Der IDP-Dienst MUSS die Signatur der Liste der sektoralen Identity Providern innerhalb der Föderation durch die Verwendung einer JSON Web Signature (JWS) [RFC7515 # section-3 - Compact Serialization] und des Schüssels PrK_DISC_SIG gewährleisten. Als Algorithmus ist dementsprechend "BP256R1" zu wählen. Der IDP-Dienst MUSS bei der Signaturerstellung das Signaturzertifikat des PUK_DISC_SIG im x5c-Claim einbetten. [<=]

5.2 Authorization-Endpunkt**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. [<=]

Vorbedingung ist, dass sich das Authenticator-Modul bereits das Discovery Document heruntergeladen und dieses erfolgreich ausgewertet hat.

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. [<=]

A_21315-01 - Bereitstellung einer URI zum Einreichen von SSO_TOKEN

Der IDP-Dienst MUSS für den Authorization-Endpunkt eine zusätzliche Adresse mit der URI-Bezeichnung "sso_endpoint" anbieten, über den im Authorization Code Flow ([openid-connect-core-1_0 # CodeFlowAuth](#)) ein Client einen SS0_TOKEN einliefern kann, um einen AUTHORIZATION_CODE zu erhalten. [<=]

Hinweis: Der angebotene SSO-Endpunkt bzw. die angebotene URI namens sso_endpoint ist nur für die konkrete Übergabe des SS0_TOKEN nutzbar. Alle vorhergehenden Requests, inklusive des Authentication Requests, sind an den Authorization Server zu richten.

5.2.1 Authorization-Endpunkt 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: Der Aufbau der Anfrage entspricht [gemSpec_IDP_Dienst#Kapitel 7.1 Authorization Request].

A_20376 - Verwendung des Attributes "state"

Der Authorization-Endpunkt MUSS den vom Anwendungsfondent initiierten "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_20440-01 - Schematische Prüfung des Consent

Der IDP-Dienst MUSS die bei der organisatorischen Registrierung der App hinterlegten redirect_uri mit der redirect_uri aus dem Claim des CHALLENGE_TOKEN prüfen. Stimmen diese nicht überein, werden keine Token ausgestellt und die weitere Verarbeitung mit einem Fehler Response abgebrochen (vgl. https://openid.net/specs/openid-connect-core-1_0.html#AuthError). [<=]

Hinweis: Nach [[openid-connect-core-1_0.html # AuthRequest](#)] ist es zulässig, dass ein Client mehrere redirect_uri bei der Registrierung hinterlegt. Der muss laut der OIDC-Spezifikation prüfen, ob die im Request gelieferte redirect_uri mit exakt einer der hinterlegten redirect_uri übereinstimmt. Die Prüfung muss über eine Simple String Comparison nach [[RFC3986 # section-6.2.1](#)] erfolgen.

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_20699-03 - Annahme von CHALLENGE_TOKEN: Authentication_Data-Struktur

Der Authorization-Endpunkt MUSS:

- entweder das mit dem Zertifikat der Smartcard des Nutzers signierte und durch das Authenticator-Modul mittels PuK_IDP_ENC verschlüsselte CHALLENGE_TOKEN
- oder – im Fall der Verwendung von alternativen Authentisierungsmitteln – die mit dem privaten Schlüssel des Endgeräts signierte und durch das Authenticator-Modul mittels PuK_IDP_ENC verschlüsselte Authentication_Data-Struktur

annehmen. Er MUSS in beiden Fällen anhand des im Header vorhandenen exp-Claim die Token anhand der Systemzeit auf zeitliche Gültigkeit prüfen. Sind diese nicht mehr gültig, MUSS der Authentifizierungsvorgang abgebrochen werden. Im Fall der Gültigkeit MÜSSEN diese mit dem PrK.IDP.ENC entschlüsselt werden. [<=]

Hinweis 1: Der Aufbau der Anfrage entspricht [gemSpec_IDP_Dienst#Kapitel 7.3 Authentication Request].

Hinweis 2: Als Verschlüsselungsalgorithmus ist ECDH-ES (Elliptic Curve Diffie-Hellman Ephemeral Static key agreement) vorgesehen.

Hinweis 3: Die Prüfung auf zeitliche Gültigkeit des Challenge-Token nach Entschlüsselung ist hiervon unbenommen. Die geschilderte Vorabprüfung gestattet dem IDP-Dienst, bei Empfang von zeitlich ungültigen Token auf eine Entschlüsselung zu verzichten.

A_20951-01 - Validierung der Signatur und des Zertifikats des CHALLENGE_TOKEN

Der Authorization-Endpunkt MUSS die Signatur des vom Authenticator-Modul übertragenen, signierten CHALLENGE_TOKEN anhand des mitgelieferten Authentifizierungszertifikats überprüfen. Die Überprüfung MUSS neben der Signatur auch das Authentifizierungszertifikat anhand von OCSP umfassen. [**<=**]

Hinweis: Der genaue Aufbau des vom Authenticator-Modul übertragenen, signierten CHALLENGE_TOKEN findet sich in [gemSpec_IDP_Dienst#Kapitel 7.3 Authentication Request].

A_22328 - IDP-Dienst: Überprüfung der korrekten keyUsage des AUT-Zertifikats

Der IDP-Dienst MUSS die Werte der Attribute "keyUsage" und "ExtendedKeyUsage" des AUT-Zertifikats des vom Authenticator-Moduls gelieferten Response der Challenge prüfen und eine weitere Verarbeitung verweigern, wenn nicht die folgenden vorgesehenen Werte - neben möglichen weiteren Werten - enthalten sind:

intendedKeyUsage = digitalSignature;

intendedExtendedKeyUsage = id-kp-clientAuth.

Die Nutzung des Attributs "ExtendedKeyUsage" ist optional, weshalb der IDP-Dienst die Überprüfung des Wertes des Attributs "ExtendedKeyUsage" nur durchführen muss, wenn das Attribut auch vorhanden ist. Ein generelles Fehlen des Attributs "ExtendedKeyUsage" führt nicht zur Ablehnung. [**<=**]

A_20946-01 - Annahme eines "SSO_TOKEN"

Der Authorization-Endpunkt MUSS einen vom Authenticator-Modul übertragenen SSO_TOKEN annehmen, sofern anhand der übertragenen client_id festgestellt wird, dass im Registrierungsprozess für diese client_id die Verwendung eines SSO_TOKEN hinterlegt wurde. Ansonsten MUSS der IDP-Dienst die Verarbeitung mit einer Fehlermeldung abbrechen.

[**<=**]

A_20313-01 - Inhalte des Claims

Der IDP-Dienst MUSS ID_TOKEN und ACCESS_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 für ACCESS_TOKEN und ID_TOKEN gemäß der spezifizierten Maximalwerte. [**<=**]

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_20948-01 - 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_IDP_SIG 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_20950-01 - Positive Validierung des "SSO_TOKEN"

Der Authorization-Endpunkt MUSS bei der positiven Validierung des vom Authenticator-Moduls eingereichten SSO_TOKEN einen AUTHORIZATION_CODE 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_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_20692-01 - Maximale Gültigkeitsdauer eines "SSO_TOKEN"

Der Authorization Endpunkt DARF die zeitliche Gültigkeit eines SSO_TOKEN NICHT länger als 86400 Sekunden (24 Stunden) einstellen.

Der Parameter auth_time beinhaltet den Zeitpunkt der letzten Authentisierung gegenüber dem IDP-Dienst mittels Challenge/Response. [≤]

A_20314-01 - Maximale Gültigkeitsdauer des "AUTHORIZATION_CODE" und des "CHALLENGE_TOKEN"

Der Authorization Endpunkt DARF die zeitliche Gültigkeit des CHALLENGE_TOKEN NICHT länger als 180 Sekunden (Challenge-Response) und die des AUTHORIZATION_CODE NICHT länger als 60 Sekunden einstellen. [≤]

A_20318 - Keine Token für widerrufenen 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. [≤]

A_22290 - Zwischenspeichern (Caching) von Smartcard-basierten OCSP-Antworten

Der Produkttyp IDP MUSS OCSP-Antworten zu Gültigkeitsanfragen für Smartcard-basierte Zertifikate zwischenspeichern und bei einer erneuten Prüfung desselben Zertifikates anstelle einer neuen OCSP-Anfrage diese zwischengespeicherte Information verwenden.

Die Informationen MÜSSEN jeweils für einen konfigurierbaren Zeitraum (von maximal 60 Minuten) gespeichert werden.

Voreingestellt sind 30 Minuten.

Der TUC_PKI_006 und der eingebaute Cache des OCSP-Responder sind hierbei zu berücksichtigen. [≤]

5.2.2 Authorization-Endpunkt Ausgangsdaten

A_20521-03 - Inhalt des CHALLENGE_TOKEN an das Authenticator-Modul

Der IDP-Dienst MUSS die ihm vorliegenden Informationen (z.B. CODE_CHALLENGE, SCOPE und alle Informationen über Anwendungsfrontend und Authenticator-Modul) mit seinem privaten Schlüssel PrK_IDP_SIG und der technischen Rolle "oid_idpd" gemäß [gemSpec_OID # Abschnitt 3.5.4] signieren und als JWT ergänzt um die USER_CONSENT-

Anfrage an das Authenticator-Modul senden. Als Algorithmus ist dementsprechend "BP256R1" zu wählen. [<=]

Konnten alle Prüfungen des eingereichten Consent erfolgreich abgeschlossen werden, erstellt der Authorization-Endpunkt ein AUTHORIZATION_CODE ergänzt durch ein SSO_TOKEN.

Der Token-Endpunkt gibt bei Vorlage eines gültigen AUTHORIZATION_CODE einen ACCESS_TOKEN und einen ID_TOKEN heraus. Ein Authorization_Code ist gültig, wenn er

- authentisch,
- zeitlich gültig
- noch nicht verwendet wurde

und von einem Client verwendet wird, der im Besitz des CODE_VERIFIERS zu der im AUTHORIZATION_CODE eingebetteten CODE_CHALLENGE ist.

Hinweis: Der genaue Aufbau der Antwort und des CHALLENGE_TOKEN entspricht [gemSpec_IDP_Dienst#Kapitel 7.2 Authorization Response].

A_20377 - Verwendung des Attributes "state"

Der Authorization-Endpunkt MUSS den state-Parameter [[RFC6749 # section-10.12](#)] des Anwendungsfrontends in allen darauf basierenden Responses verwenden. [<=]

A_20694-01 - Zusammenstellung des "SSO_TOKEN"

Der vom Authorization-Endpunkt ausgestellte SSO_TOKEN MUSS so zusammengestellt sein, dass aus den darin enthaltenen Claims ein neuer AUTHORIZATION_CODE erstellt werden kann, mit Hilfe dessen der Token-Endpunkt einen passenden ACCESS_TOKEN erstellen kann. [<=]

A_20695-01 - Signieren des "SSO_TOKEN"

Der Authorization-Endpunkt MUSS den SSO_TOKEN mit seinem eigenen privaten Schlüssel PrK_IDP_SIG signieren. Als Algorithmus ist dementsprechend "BP256R1" zu wählen. [<=]

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_21317 - Verschlüsselung des "AUTHORIZATION_CODE"

Der Authorization-Endpunkt des IdP-Dienstes MUSS den AUTHORIZATION_CODE für den Token-Endpunkt mit eigenem Schlüsselmaterial verschlüsseln, welches den Anforderungen aus [gemSpec_Krypt] genügt. [<=]

A_20319-01 - 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_OID # Abschnitt 3.5.4] signieren, damit das Authenticator-Modul sicher gewährleisten kann, dass der eingehende AUTHORIZATION_CODE tatsächlich vom IDP-Dienst stammt [[RFC7519 # section-7.1](#)]. [<=]

A_21330 - Ablaufzeitpunkt von "AUTHORIZATION_CODE" und "SSO_TOKEN"

Der Authorization-Endpunkt des IDP-Dienstes MUSS im "exp"-Claim des jeweiligen JWE Headers den Ablaufzeitpunkt des AUTHORIZATION_CODE bzw. SSO_TOKEN liefern. [<=]

Hinweis: Der Aufbau der Header entspricht [gemSpec_IDP_Dienst#Kapitel 7.4 Authentication Response].

A_20693-01 - Senden des "AUTHORIZATION_CODE" und "SSO_TOKEN" an die "REDIRECT_URI"

Der IDP-Dienst MUSS den AUTHORIZATION_CODE und die registrierte REDIRECT_URI an das Authenticator-Modul senden. Der IDP-Dienst MUSS zusätzlich einen SSO_TOKEN mitliefern, wenn die client_id im Registrierungsprozess des Anwendungsfrontends für die Nutzung eines SSO_TOKEN freigeschaltet wurde. [<=]

A_20320-01 - 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: Der genaue Aufbau der Antwort entspricht [gemSpec_IDP_Dienst#Kapitel 7.4 Authentication Response].

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.

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. [<=]

Hinweis: Die Gültigkeitsdauer des AUTHORIZATION_CODE wird im Claim des angesprochenen Fachdienstes definiert.

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. [<=]

Hinweis: Die Gültigkeitsdauer des ACCESS_TOKEN wird im Claim des angesprochenen Fachdienstes definiert.

5.3.1 Token-Endpunkt Eingangsdaten

A_20321-01 - Annahme und Prüfung von "AUTHORIZATION_CODE" und "KEY_VERIFIER"

Der Token-Endpunkt MUSS den vom Anwendungsfrontend übertragenen AUTHORIZATION_CODE und den KEY_VERIFIER des Anwendungsfrontend annehmen. Der AUTHORIZATION_CODE ist dabei mittels eines durch den IDP-Dienst für Authorization-Endpunkt und Token-Endpunkt definierten Verfahren zu entschlüsseln. [<=]

Hinweis 1: Der Aufbau der Anfrage entspricht [gemSpec_IDP_Dienst#Kapitel 7.5 Token Request].

Hinweis 2: Als Verschlüsselungsalgorithmus für den KEY_VERIFIER ist ECDH-ES (Elliptic Curve Diffie-Hellman Ephemeral Static key agreement) vorgesehen.

A_21318 - Prüfung des "AUTHORIZATION_CODE"

Der Token-Endpoint MUSS die Signatur des AUTHORIZATION_CODE unter Verwendung des Schlüssels PuK_IDP_SIG prüfen.[<=]

A_21319 - Prüfung des "CODE_VERIFIER"

Der Token-Endpoint MUSS den CODE_VERIFIER aus dem mittels PuK_IDP_ENC verschlüsselten KEY_VERIFIER extrahieren und die Überprüfung gegen die CODE_CHALLENGE mit S256 (Algorithmus nach [[RFC7636 # section-4.2](#)]) durchführen. [<=]

Hinweis: Der Aufbau des KEY_VERIFIER entspricht [gemSpec_IDP_Dienst#Kapitel 7.5 Token Request].

A_21320 - Entschlüsseln des "Token-Key"

Der Token-Endpoint MUSS den "Token-Key" aus dem mittels PuK_IDP_ENC verschlüsselten KEY_VERIFIER extrahieren.[<=]

Hinweis: Der Aufbau des KEY_VERIFIER entspricht [gemSpec_IDP_Dienst#Kapitel 7.5 Token Request].

A_21309 - Prüfung der Verwendung des Authorization Codes

Der Anbieter des IDP-Dienstes KANN gegen [[RFC6749 # section-10.5](#)] verstoßen, indem der IDP-Dienst nicht die einmalige Verwendung eines AUTHORIZATION_CODE überprüft. Die kurze Gültigkeit und die Verschlüsselung der ausgestellten Token wird als ausreichend sicher betrachtet.[<=]

A_20323 - TOKEN-Ausgabe Protokollierung in allen Fällen

Der Token-Endpoint MUSS die Herausgabe der TOKEN im Positiv- wie auch im Negativfall protokollieren.[<=]

5.3.2 Token-Endpoint Ausgangsdaten

A_20464 - Token-Endpoint (Datensparsamkeit)

Der Token-Endpoint DARF andere Informationen, als die im Claim geforderten, NICHT herausgeben.[<=]

A_20315-01 - "AUTHORIZATION_CODE" nach Gültigkeitsende nicht mehr verwenden

Der Token-Endpoint DARF außerhalb der Gültigkeitsdauer eingehenden AUTHORIZATION_CODE NICHT in ID_TOKEN oder ACCESS_TOKEN eintauschen.[<=]

A_20524-07 - Befüllen der Claims "given_name", "family_name", "organizationName", "professionOID", "idNummer", "organizationIK", "acr" und "amr"

Der Token-Endpoint MUSS benötigte Attribute in Claims für das auszustellende ACCESS_TOKEN und das ID_TOKEN ausschließlich aus dem ihm mit dem signierten CHALLENGE_TOKEN eingereichten Authentifizierungszertifikat der Smartcard (eGK, HBA oder SMC-B) beziehen. Der Claim amr MUSS entsprechend des ursprünglich zur Authentisierung verwendeten Authentisierungsmittels belegt werden.

Der Token-Endpoint MUSS das Attribut given_name und family_name der juristischen und natürlichen Personen sowie die Attribute organizationName, professionOID und idNummer sowie organizationIK entsprechend dem Datenformat der Informationsquelle (Zertifikat) wie folgt befüllen:

Tabelle 4: TAB_IDP_DIENST_0005 Befüllung der Attribute "given_name", "family_name", "organizationName", "professionOID", "idNummer" und "organizationIK"

Attribute	HBA Inhaber (Leistungserbringer) Quell-Zertifikat: C.HP.AUT	SMC-B Inhaber (Leistungserbringer institution, PKV) Quell-Zertifikat: C.HCI.AUT	SM-B Inhaber (Kostenträger) Quell-Zertifikat: C.HCI.AUT	SM-B Inhaber (NCPeH) Quell-Zertifikat: C.HCI.AUT	eGK Inhaber (Versicherte) Quell-Zertifikat: C.CH.AUT
given_name (Zertifikatsfeld)	Vorname (givenName)	Vorname des Verantwortlichen /Inhabers (givenName)	nicht belegt	nicht belegt	Vorname (givenName)
family_name (Zertifikatsfeld)	Nachname (surname)	Nachname des Verantwortlichen /Inhabers (surname)	nicht belegt	nicht belegt	Nachname (surname)
organizationName (Zertifikatsfeld)	leer (organizationName)	Organisationsbezeichnung (commonName)	Organisationsbezeichnung (commonName)	Ländercode des EU-Mitgliedstaates (commonName)	Herausgeber (organizationName)
professionOID (Zertifikatsfeld)	professionOID (Admission/professionOID)	professionOID (Admission/professionOID)	professionOID (Admission/professionOID)	professionOID (Admission/professionOID)	professionOID (Admission/professionOID)
idNummer (Zertifikatsfeld)	Telematik-ID (Admission/registrationNumber)	Telematik-ID (Admission/registrationNumber)	8-<8-stellige eindeutige Betriebsnummer (BBNR) des GKV-SV oder eine alternative ID> (Admission/registrationNumber)	Telematik-ID (Admission/registrationNumber)	unveränderlicher Anteil der KVNR (organizationalUnitName)
organizationIK (Zertifikatsfeld)	leer	leer	leer	leer	Anteil der IK-Nummer (organizationalUnitName)

Tabelle 5: TAB_IDP_DIENST_0006 Befüllung der Attribute "acr" und "amr"

Attribut	HBA Inhaber (Leistungserbringer)	SMC-B / SM-B Inhaber (Leistungserbringerin institution und Kostenträger)	eGK Inhaber (Versicherte)	Versicherte (alternative Authentisierungsmittel)
amr	[„mfa“, „sc“, „pin“]	[„mfa“, „sc“, „pin“]	[„mfa“, „sc“, „pin“]	Gemäß des übertragenen Werts

				des Authenticator-Moduls in der Datenstruktur "Signed_Authentication_Data"
acr	gematik-ehealth-loa-high	gematik-ehealth-loa-high	gematik-ehealth-loa-high	gematik-ehealth-loa-high

[<=]

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.

Hinweis: Der Aufbau von ACCESS_TOKEN und ID_TOKEN entspricht [gemSpec_IDP_Dienst#Kapitel 7.6 Token Response].

A_22271-02 - Befüllen der Claims "display_name", "organizationName", "professionOID", "idNummer", "organizationIK", "acr" und "amr" nach Bestätigung durch einen sektoralen Identity Provider

Der Token-Endpunkt MUSS benötigte Attribute in Claims für das auszustellende ACCESS_TOKEN und das ID_TOKEN ausschließlich aus den entsprechenden Claims des ID_TOKEN des sektoralen Identity Provider beziehen. Der Claim amr MUSS entsprechend des ursprünglich zur Authentisierung verwendeten Authentisierungsmittels belegt werden.

Tabelle 6: TAB_IDP_DIENST_0007 Befüllung der Attribute nach Bestätigung durch einen sektoralen Identity Provider

Attribute	Versicherte
display_name	Vollständiger Name des Versicherten, entspricht dem claim urn:telematik:claims:display_name aus dem vom sektoralen IDP ausgestellten ID-Token
given_name	Vorname des Versicherten, entspricht dem claim urn:telematik:claims:given_name aus dem vom sektoralen IDP ausgestellten ID-Token
family_name	Familienname des Versicherten, entspricht dem claim urn:telematik:claims:family_name aus dem vom sektoralen IDP ausgestellten ID-Token
organizationName	Herausgeber-ID (Institutionskennzeichen), entspricht dem claim urn:telematik:claims:organization aus dem vom sektoralen IDP ausgestellten ID-Token
professionOID	ProfessionOID, entspricht dem claim urn:telematik:claims:profession aus dem vom sektoralen IDP ausgestellten ID-Token. Der Wert ist immer 1.2.276.0.76.4.49 .
idNummer	KVNR, entspricht dem claim urn:telematik:claims:id aus

	dem vom sektoralen IDP ausgestellten ID-Token
organizationIK	Herausgeber-ID (Institutionskennzeichen), entspricht dem claim urn:telematik:claims:organization aus dem vom sektoralen IDP ausgestellten ID-Token
amr	"mfa"
acr	"gematik-ehealth-loa-high"

[<=]

Hinweis 1: Das Vertrauensniveau "gematik-ehealth-loa-high" ist aktuell das einzige LOA das der e-Rezept Fachdienst kennt. Für die zukünftigen föderierten Identity Provider wird es weiter abgestufte Vertrauensniveaus geben.

Hinweis 2: Der Aufbau von ACCESS_TOKEN und ID_TOKEN entspricht [gemSpec_IDP_Dienst#Kapitel 7.6 Token Response].

Hinweis 3: Die Claims given_name und family_name werden befüllt, wenn die Authentisierung nicht durch einen sektoralen IDP der Föderation erfolgte.

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_22326 - IDP-Dienst: Setzen des korrekten Issuer Claims

Der IDP-Dienst MUSS je nach Ursprung (aus der TI oder aus dem Internet) des Aufrufs des Authentication Request den Issuer Claim iss in seinen ausgestellten Token analog zum Issuer Claim issuer des jeweiligen Discovery Document - intern (TI) oder extern (Internet) - setzen.[<=]

A_20327-02 - Signatur des "ID_TOKEN" und "ACCESS_TOKEN"

Der Token-Endpunkt MUSS alle erstellten ID_TOKEN und ACCESS_TOKEN, um deren Integrität sicherzustellen und eine eindeutige Erklärung über deren Herkunft abzugeben, mit seinem privaten Schlüssel PrK_IDP_SIG signieren. [[RFC7523 # section-3](#) Spiegelpunkt 9] ist zu gewährleisten. Als Algorithmus ist dementsprechend "BP256R1" zu wählen.[<=]

Hinweis: Zum Aufbau des Signaturheader siehe [gemSpec_IDP_Dienst#Kapitel 7.6 Token Response].

A_21321 - Verschlüsselung von "ACCESS_TOKEN" und "ID_TOKEN"

Der Token-Endpunkt MUSS ACCESS_TOKEN und ID_TOKEN nach der Signatur mittels JWE [[RFC7516](#)] unter Nutzung des "Token-Key" des Anwendungsfrontend verschlüsseln.[<=]

Hinweis: Zum Aufbau des Verschlüsselungs-Header siehe [gemSpec_IDP_Dienst#Kapitel 7.6 Token Response].

A_20329-01 - Sichere Übertragung von "ID_TOKEN" und "ACCESS_TOKEN"

Der Token-Endpunkt MUSS ID_TOKEN und ACCESS_TOKEN beim Transport mit Transport Layer Security (TLS) gemäß [gemSpec_Krypt] schützen.

[<=]

Hinweis: Der Aufbau von ACCESS_TOKEN und ID_TOKEN entspricht [gemSpec_IDP_Dienst#Kapitel 7.6 Token Response].

5.4 Pairing-Endpunkt

5.4.1 Zielsetzung

Fachdienste der TI erfordern an Frontends der Versicherten in der Regel eine Authentisierung mithilfe der eGK. Während die Authentisierung auf Basis der eGK bei Verwendung von Mobilgeräten gegenüber dem IDP-Dienst im Dokument [gemSpec_IDP_Frontend] dargelegt ist, zielen die in diesem Dokument dargestellten Erweiterungen der IDP-Spezifikationen darauf ab, die Verwendung von alternativen Authentisierungsmitteln zu ermöglichen, die es Nutzern gestatten, sich mithilfe eines Mobilgeräts und vom Nutzer lokal selbst gesetzten Authentisierungsmitteln (wie etwa biometrische oder wissensbasierte Faktoren) ohne weitere Verwendung der eGK gegenüber einem Fachdienst zu authentisieren.

Unter alternativen Authentisierungsmitteln werden hierbei solche verstanden, die aus einem bereits etablierten Authentifizierungsmittel – wie etwa der eGK – abgeleitet werden und dieses in äquivalenter Form ersetzen. Unter einer „Ableitung“ wird hierbei eine zeitbeschränkte, unabhängig nachvollziehbare Beziehung zwischen einem vom Nutzer gesetzten Mittel und einem bereits etablierten Mittel verstanden. Alternative Authentisierungsmittel sind funktional dann anwendbar, wenn das ursprünglich etablierte Authentisierungsmittel anwendbar ist; ihr Lebenszyklus endet spätestens mit Ablauf des Authentisierungsmittels, das zur Ableitung verwendet wurde.

Die zugrundeliegende Konzeption basiert auf der Verwendung von kryptographischen Verfahren in Kombination mit einem geeigneten Endgerät; eine Verwendung von anderen Authentisierungsmitteln gegenüber dem IDP, wie etwa dort hinterlegte Passwörter, ist nicht angestrebt. Grundlage für die kryptographischen Verfahren ist die Erzeugung eines asymmetrischen Schlüsselpaars auf dem Endgerät des Nutzers. Der öffentliche Schlüssel wird mit anderen Metadaten im Rahmen eines Registrierungsprozesses durch den Nutzer mithilfe seiner eGK gegenüber dem IDP-Dienst authentifiziert und am IDP-Dienst zur zukünftigen Authentifizierung des Nutzers registriert. Der zugehörige private Schlüssel wird vom Nutzer in äquivalenter Weise zu dem in [gemSpec_IDP_Dienst] und [gemSpec_IDP_Frontend] beschriebenen Challenge/Response-Verfahren zur Authentisierung angewendet. Bedingung für die Anwendung ist die erfolgreiche Authentifizierung des Nutzers durch lokal gesetzte Faktoren.

Die Kombination von öffentlichem Schlüssel und Metadaten wird im Folgenden als "Pairing" bzw. "Pairing-Daten" benannt. Eine Authentisierung auf Basis des zu einem solchen öffentlichen Schlüssel gehörenden privaten Schlüssels wird "Pairing-basierte Authentisierung" genannt. Das Hinterlegen von Pairing-Daten am IDP-Dienst wird als "Registrierung" bezeichnet. Eine Authentifizierung auf Basis der Pairing-Daten wird als "Pairing-basierte Authentifizierung" bezeichnet. Das Gerät in Kombination mit privatem Schlüssel ist das alternative Authentisierungsmittel. Ein Pairing ist "aktiv", wenn es zur Authentifizierung verwendet werden kann. Voraussetzung für die Aktivierung ist die erfolgreiche Registrierung. Unter "Inspektion" durch den Nutzer wird die Möglichkeit zum Abrufen der zu diesem Zweck auf seinem Gerät sowie der am IDP-Dienst gespeicherten Daten verstanden. Der Ausschluss eines alternativen Authentisierungsmittels oder Pairings zum Zweck der Authentifizierung wird "Deaktivierung" genannt. Eine Deaktivierung ist Folge einer Löschung des alternativen Authentisierungsmittels, eines vom Benutzer vollzogenen Deregistrierungsprozesses, des Endes der Gültigkeit des zur Etablierung verwendeten Authentisierungsmittels oder eine auf Basis einer Einstufung der Eignung des verwendeten Geräts getroffenen Entscheidung des IDP-Dienstes. Authentisierungsmethoden des Nutzers gegenüber seinem Gerät bzw. einem

auf dem Gerät vorhandenen lokalen Nutzeraccount (z. B. durch biometrische Faktoren oder wissensbasierte Faktoren), werden im Folgenden "lokale Authentisierungsmittel", ihre Anwendung "lokale Authentisierung", ihre Prüfung durch das Gerät "lokale Authentifizierung" genannt.

Der Begriff "isolierte Ausführungsumgebung" wird als abstrakter technik-neutraler Oberbegriff für die verschiedenen üblichen Realisierungsformen von auf Geräten verwendeten Schlüsselspeichern verwendet. Sowohl die Terminologie als auch die tatsächlichen Ausprägungen sind leider uneinheitlich. Typische herstellerspezifische Bezeichnungen oder Ausprägungen sind "Trusted Execution Environment", "Secure-Enclave", "StrongBox" oder "Secure-Element". Im Kontext der Einleitung wird hierunter ein geeignet realisierter Schlüsselspeicher verstanden, in dem Schlüssel erzeugt, gespeichert und gelöscht werden, und die Anwendung des Schlüssels (z. B. zum Zweck der Signaturbildung) auf Daten an bestimmte Bedingungen (wie etwa eine Benutzerauthentifizierung) geknüpft ist und der Zugriff auf die Schlüssel als Datenobjekte ausreichend verhindert wird.

Die erfolgreiche Authentifizierung des Nutzers bei Verwendung von alternativen Mitteln soll an die in der folgenden Tabelle genannten Faktoren gebunden sein. Die dort genannten Anforderungen adressieren die Abwehr von missbräuchlicher Verwendung des alternativen Authentisierungsmittels bei Wegfall, Modifikation, vermuteter oder erkannter Kompromittierung einer dieser Faktoren durch Deaktivierung.

Tabelle 7: TAB_IDP_DIENST_0008 Faktoren und abgeleitete Anforderungen an das System

Motivation	Faktoren	Anforderungen an das System
Die Verwendung von alternativen Authentisierungsmitteln ist an den Nutzer als Entität, seine Absicht diese zu nutzen und an das von ihm initial zur Registrierung verwendeten Authentisierungsmittel (hier: eGK) gebunden.	<ul style="list-style-type: none"> • der Nutzer als Entität • die Absicht des Nutzers zur Verwendung von alternativen Authentisierungsmitteln • die Bindung an das ursprüngliche Authentisierungsmittel 	<ul style="list-style-type: none"> • Die Etablierung eines alternativen Authentisierungsmittels erfolgt ausschließlich auf Initiative des Nutzers. • Die Etablierung eines alternativen Authentisierungsmittels ist an die eGK als bereits etabliertes Authentisierungsmittel gebunden. • Die Verwendung eines alternativen Authentisierungsmittels durch den Nutzer zur Authentifizierung gegenüber Fachdiensten ist immer optional. • Die Verwendung eines alternativen Authentisierungsmittels ist an lokale Authentisierungsmittel des Nutzers gebunden, das heißt an wissensbasierte oder biometrische Faktoren, die

		<p>seinem lokalen Nutzeraccount zugeordnet sind.</p> <ul style="list-style-type: none"> • Das System ermöglicht dem Nutzer den Abruf einer Übersicht aller registrierten alternativen Authentifizierungsmittel von jedem Gerät aus • Der Nutzer kann ein alternatives Authentifizierungsmittel jederzeit von jedem Gerät aus deaktivieren. • Der Nutzer kann ein für ein Gerät angelegtes alternatives Authentifizierungsmittel jederzeit über ein Gerät durch lokale Löschung von Schlüsseln deaktivieren.
<p>Die Verwendung von alternativen Authentifizierungsmitteln ist an einen legitimen Besitzer des verwendeten Geräts, seine Präsenz (am Gerät) und die hierbei verwendeten Authentifizierungsmittel zur lokalen Authentisierung (wie Passwörter, PINs oder biometrische Faktoren) zur Nutzung eines lokalen Nutzeraccounts gebunden.</p>	<ul style="list-style-type: none"> • der Besitz des Geräts • die Bindung an einen definierten, lokalen Nutzeraccount • die Präsenz des Nutzers zum Zeitpunkt der Authentisierung 	<ul style="list-style-type: none"> • Die Erzeugung des Schlüsselpaars, die Registrierung des Schlüssels, seine Verwendung zur Authentifizierung und Deaktivierung erfolgt ausschließlich innerhalb eines Nutzeraccounts, der mit lokalen Authentifizierungsmitteln geschützt ist. • Die Anwendung des alternativen Authentifizierungsmittels erfordert die gesonderte Authentifizierung des Nutzers am Gerät durch lokale Authentifizierungsmittel • Eine Deaktivierung des alternativen Authentifizierungsmittels erfolgt durch lokale Löschung des privaten Schlüssels bei • Löschung des Benutzeraccounts innerhalb dessen der Schlüssel erzeugt wurde oder Reset des Geräts,

		<ul style="list-style-type: none"> • bei Entfernung von lokalen Authentisierungsmitteln oder Rückfall auf schwache Formen der Authentisierung (z. B. "wischen") oder • bei Änderung von lokalen Authentisierungsmitteln (z. B. Re-Enrolment von biometrischen Faktoren). • Bei Verlust des Geräts ist eine Deaktivierung des alternativen Authentisierungsmittels von anderen Geräten aus möglich.
<p>Die Anwendung des alternativen Authentisierungsmittels ist an Gerät, Betriebssystem und Authenticator-Modul und an die Eignung der verwendeten kryptographischen Verfahren gebunden. Hierin eingeschlossen ist die Authentifizierung des Nutzers vor Anwendung des privaten Schlüssels.</p>	<ul style="list-style-type: none"> • die Kontrolle des Nutzers über den zur Authentisierung eingesetzten privaten Schlüssel 	<ul style="list-style-type: none"> • gesonderte Authentifizierung des Nutzers vor Anwendung des alternativen Authentisierungsmittels durch lokale Mittel • Anwendung des alternativen Authentisierungsmittels ausschließlich durch am Gerät authentifizierte Nutzer in einer isolierten Ausführungsumgebung • Nicht-Exportierbarkeit des alternativen Authentisierungsmittels • Nicht-Rekonstruierbarkeit des alternativen Authentisierungsmittels aus öffentlichen Daten • Das System bietet die Möglichkeit zum Ausschluss der alternativen Authentisierungsmittels bei erkannter Kompromittierung der verwendeten kryptographischen Verfahren.
<p>Die Möglichkeit zur Anwendung des alternativen Authentisierungsmittels ist an das Authenticator-Modul als Anwendung (potentiell in einer bestimmten Version und ggf. Folgeversionen)</p>	<ul style="list-style-type: none"> • das Authenticator-Modul als Anwendung zur Verwendung des alternativen Authentisierungsmittels 	<ul style="list-style-type: none"> • Die Anwendung des alternativen Authentisierungsmittels erfolgt ausschließlich bei Verwendung des Authenticator-Moduls, das die Erzeugung des Schlüsselpaars auf Kommando des Nutzers

<p>zum Zweck der Authentifizierung am IDP-Dienst gebunden.</p>		<p>gesteuert hat.</p> <ul style="list-style-type: none"> • Eine Deaktivierung des alternativen Authentisierungsmittels erfolgt durch lokale Löschung des privaten Schlüssels bei Deinstallation des Authenticator-Moduls oder Rückfall auf eine frühere Version.
<p>Der Erfolg der Authentifizierung ist an die Kombination von Betriebssystem und Hardware im Hinblick auf Ausstattung und sicherheitstechnischer Einstufung gebunden.</p>	<ul style="list-style-type: none"> • die Kombination von Betriebssystem und Hardware 	<ul style="list-style-type: none"> • Das System gibt dem Betreiber des IDP-Dienstes die Möglichkeit, die vom Benutzer verwendete Geräte in geeignete, bedingt geeignete und ungeeignete Geräte einzustufen. • Bei Verwendung von ungeeigneten Geräten durch den Nutzer wird die Registrierung abgelehnt. • Bei Verwendung eines ungeeigneten Geräts zum Zeitpunkt der Authentifizierung ist die Authentifizierung nicht erfolgreich. • Das System setzt eine zeitliche Beschränkung der Verwendung des Pairings zum Zweck der Authentifizierung bei der Verwendung von bedingt geeigneten Geräten durch.
<p>Die Verwendung des alternativen Authentisierungsmittels ist an die Version des Betriebssystems gebunden, unter dessen Steuerung es erzeugt wurde.</p>	<ul style="list-style-type: none"> • das Betriebssystem des verwendeten Geräts 	<ul style="list-style-type: none"> • Eine Deaktivierung des alternativen Authentisierungsmittels erfolgt durch lokale Löschung des privaten Schlüssels bei Deinstallation von Updates des Betriebssystems oder bei Reset des Gerätes.
<p>Die Verwendung des alternativen Authentisierungsmittels soll an die verwendete Gerätehardware gebunden sein.</p>	<ul style="list-style-type: none"> • die verwendete Gerätehardware 	<ul style="list-style-type: none"> • Der Nutzer kann ein alternatives Authentisierungsmittel jederzeit von jedem Gerät aus deaktivieren (z. B. bei Hardwaredefekten).

Eine Durchsetzung der in der Tabelle genannten und weiterer abgeleiteter Anforderungen ist nicht allein durch das Authenticator-Modul als Softwareeinheit auf dem Gerät des Nutzers und dem IDP-Dienst zu realisieren. Dem Authenticator-Modul obliegt die Prüfung auf das Vorliegen einer – gemessen an Einsatzzweck und Stand der Technik – ausreichend sicheren Einsatzumgebung auf dem mobilen Endgerät, insbesondere auf die Realisierung eines in Interaktion mit Gerät und Betriebssystem durchgesetzten, durchgängigen Schutz der kryptographischen Schlüssel im gesamten Lebenszyklus von

- Erzeugung des Schlüsselpaars,
- Speicherung des privaten Schlüssels,
- Verwendung des privaten Schlüssels zur Authentifizierung,
- Löschung oder endgültige Nicht-Verwendbarkeit des privaten Schlüssels.

Anforderungen an die Implementierung des Authenticator-Moduls zielen darauf ab, dem Betriebssystem und Gerät geeignete Parameter zu setzen, die eine Kompromittierung des kryptographischen Schlüsselmaterials unabhängig von einer Kompromittierung des Authenticator-Moduls oder des Betriebssystems zu verhindern. Komplementär wird der IDP-Dienst durch die Implementierung eines weiteren Endpunkts (dem "Pairing-Endpunkt") erweitert, der dem Nutzer die Möglichkeit zur:

- Registrierung seines Gerätes und des alternativen Authentifizierungsmittels,
- Inspektion der von ihm registrierten Daten und Geräte,
- Deregistrierung eines von ihm registrierten Gerätes und des alternativen Authentifizierungsmittels

gibt. Der Authorization-Endpunkt des IDP-Dienstes wird um die:

- Authentifizierung des Nutzers auf Basis von registrierten alternativen Authentifizierungsmittels

erweitert. Registrierung und Authentifizierung schließen die Bewertung der Gerätehardware auf Basis der vorliegenden Informationen durch den IDP-Dienst ein. Die eingeschlossene Bewertung ist motiviert durch die vorab beschränkten Möglichkeiten des Authenticator-Moduls zur Bewertung des Gerätes. Sie bietet dem Betreiber des IDP-Dienstes die Möglichkeit zur direkten Intervention bei Bekanntwerden von Sicherheitsmängeln einzelner Gerätetypen und deren gezieltem Ausschluss bei der Registrierung oder zum Zweck der Authentisierung.

5.4.2 Technisches Konzept

5.4.2.1 Bewertung von Gerätetypen

Die in den folgenden Abschnitten beschriebenen Erweiterungen des IDP-Dienstes nutzen zur Bewertung eines vom Nutzer verwendeten Gerätetyps zum Zeitpunkt der Registrierung und Authentifizierung zwei Listen (in Kombination im Folgenden Block/Allow-Liste genannt):

- Die Block-Liste enthält Gerätetypen, die nicht für die Authentisierung geeignet sind; ihre Verwendung zum Zweck der Authentisierung ist ausgeschlossen.
- Die Allow-Liste enthält Gerätetypen, die ohne Einschränkung zum aktuellen Zeitpunkt als zum Zweck der Authentisierung uneingeschränkt geeignet angesehen werden.
- Gerätetypen, die sich weder auf der Block- noch auf der Allow-Liste befinden, erhalten eine Einstufung als bedingt geeignet zum Zweck der Authentisierung. Eine bedingte

Eignung setzt eine Einschränkung an die Verwendungszeit des Pairings, die bei Überschreiten eine Deaktivierung des Pairings bewirkt.

Gerätetypen auf diesen Listen werden anhand der Kombination von:

- Herstellername,
- Produktname,
- Modell,
- Betriebssystem und
- Version des Betriebssystems

beschrieben. Die anhand der Listen vorliegende Bewertung ist nicht als statisch anzunehmen, die Bewertung der Eignung erfolgt kontinuierlich auf Basis des Standes der Technik und der allgemeinen Sicherheitspolitik innerhalb der Telematikinfrastruktur. Die Pflege der Datenbasis obliegt dem Anbieter des IDP-Dienstes nach Vorgaben der gematik.

5.4.2.1.1 Spezifikation

A_21404 - Bewertung von Typen mobiler Endgeräte durch den IDP-Dienst

Der IDP-Dienst MUSS zu vorgelegten Informationen über einen Gerätetyp innerhalb der Registrierung oder der Authentifizierung die folgende Einstufung vornehmen können: Ein Gerätetyp ist zur Verwendung als Faktor innerhalb eines alternativen Authentisierungsmittels entweder

- ohne Einschränkung geeignet oder
- mit Einschränkung geeignet oder
- nicht geeignet.

Ein Gerätetyp wird durch die folgenden Informationen beschrieben: Hersteller, Produkt, Modell, Betriebssystem, Version des Betriebssystems. Die Zuordnung wird in den folgenden Anforderungen als "Block/Allow"-Liste beschrieben. Gerätetypen, die ohne Einschränkung geeignet sind, befinden sich auf der "Allow"-Liste; Gerätetypen, die nicht geeignet sind, befinden sich auf der "Block"-Liste.【<=】

Hinweis: Unter "vorgelegten Informationen" werden hierbei diejenigen verstanden, die vom Authenticator-Modul im Zuge der Registrierung oder der Authentifizierung über die Datenstruktur "Device_Information" an den IDP-Dienst übermittelt werden.

A_21405 - Management der Gerätetypen und ihrer Einstufung

Der Anbieter des IDP-Dienstes MUSS dem CERT der gematik das Management der Datenbasis der Gerätetypen und die Zuordnung in eine der oben genannten Kategorien gestatten:

- Das CERT der gematik MUSS für die Anpassung der Block/Allow-Liste des IDP-Dienstes einen Service Request im TI-ITSM-System stellen.
- Der IDP-Dienst muss für die Anpassung der Block/Allow-Liste des IDP-Dienstes einen Service im TI-ITSM bereitstellen und die Datenbasis entsprechend aktualisieren.

Die Daten enthalten in expliziter Form ausschließlich Gerätetypen, die

- ohne Einschränkung geeignet sind,
- nicht geeignet sind.

Gerätetypen, die keine ausdrückliche Nennung erfahren, sind vom IDP-Dienst in den Protokollabläufen als bedingt geeignet einzustufen.【<=】

A_21406 - Anforderungen an Transaktionalität der Freischaltung der Einstufung von Gerätetypen

Innerhalb eines Service-Requests vom CERT der gematik wird der vollständige, ab Freischaltung zur Einstufung zu verwendende Datenbestand der Block/Allow-Liste übermittelt. Der IdP-Dienst MUSS genau diese Daten innerhalb eines festgelegten Zeitraums zum Zweck der Einstufung wirksam werden lassen. Die alte Datenbasis wird hierbei vollständig ersetzt. [≤]

A_21407 - Vorhalten von Backup-Daten zur Einstufung von Gerätetypen

Der IdP-Dienst MUSS die Transaktionen der Datenbasis zur Einstufung der Gerätetypen (Block/Allow-Liste) der letzten 6 Monate speichern und in der Lage sein, bei Bedarf auf eine vorherige Version des Bestandes zurückfallen zu können. [≤]

A_21408 - Exportierbarkeit der Datenbasis zur Einstufung von Gerätetypen

Der Anbieter des IdP-Dienstes MUSS die aktuell verwendete Datenbasis zur Einstufung der Gerätetypen (Block/Allow-Liste) oder die innerhalb der letzten 6 Monate verwendete Version der Datenbasis der gematik zur Verfügung stellen.

[≤]

A_21409 - Anforderung an die Revisionsicherheit des Managements der Datenbasis zur Einstufung von Gerätetypen

Der Anbieter des IDP-Dienstes MUSS die erfolgten Änderungen an der Datenbasis der Block/Allow-Liste revisionsicher protokollieren. Das System MUSS jederzeit Auskunft darüber geben können:

- Was geschehen ist (z. B. Import oder Export der Datenbasis).
- Wer den Import oder Export ausgelöst hat (Basis ist die Protokollierung von Authentifizierungsinformationen).
- Wann dies geschehen ist (Basis sind Zeitstempel der Protokollierung).
- Wie dies geschehen ist (auf welche Weise die Datenbasis übertragen wurden, Quell-IP-Adressen, Systemprotokolle, Art und Weise der Authentisierung am System).

[≤]

5.4.2.2 Erweiterung des IDP-Dienstes um einen Pairing-Endpunkt

Der IDP-Dienst wird um einen weiteren Endpunkt erweitert, der die folgenden Funktionalitäten anbietet:

- Registrierung von alternativen Authentisierungsmitteln,
- Inspektion der registrierten alternativen Authentisierungsmittel,
- Deregistrierung von alternativen Authentisierungsmitteln.

Der Endpunkt wird "Pairing-Endpunkt" genannt. Der Pairing-Endpunkt wird unter einer dedizierten, im Discovery Document publizierten URI des IDP-Dienstes in gleichartiger Weise wie in [2.2- Detaillierter Überblick] beschrieben bereitgestellt. Die Voraussetzung für die Nutzung der am Pairing-Endpunkt oben genannten angebotenen Dienste ist die Vorlage eines gültigen ACCESS_TOKEN wie in [gemSpec_IDP_Dienst] beschrieben.

Im Fall der Registrierungsfunktion muss die Authentifizierung auf Basis der eGK oder eines auf Basis einer eGK-Authentifizierung ausgestellten SSO_TOKEN unter Verwendung eines geeigneten Gerätes erfolgen. Im Fall der Inspektion oder der Deregistrierung sind pauschal alle am IDP-Dienst zugelassenen Authentisierungsmethoden gestattet. Die Verwendung von bereits registrierten alternativen Authentisierungsmitteln und ggf. anderen Geräten ist hier ausdrücklich zugelassen. Alleinige Bedingung ist die Rückführbarkeit auf ein und dieselbe Identität des Nutzers.

5.4.2.2.1 Spezifikation

A_21410 - Registrierung des Pairing-Endpunkts und des Authenticator-Moduls am IdP-Dienst

Der Pairing-Endpunkt MUSS beim IDP-Dienst im Sinne von Abschnitt 4 aus [gemSpec_IDP_FD] registriert sein. Der registrierte Scope MUSS „openid pairing“ sein. Der beantragte Claim MUSS der folgende sein:

- idNummer.

Weitere Body-Claims DÜRFEN NICHT angefordert werden. Bestehende Header-Claims, die zur Prüfung der Gültigkeit von ACCESS_TOKEN verwendet werden, sind hiervon unberührt. Andere Fachdienste DÜRFEN mit dem genannten Scope NICHT registrierbar sein. [≤]

A_21429-02 - Erweiterung des Authorization-Endpunkts: Realisierung verschiedener Authentifizierungsmethoden

Der Pairing-Endpunkt MUSS über eine dedizierte URL bereitgestellt werden. Die URL MUSS im externen Discovery Document aufgenommen und über das Attribut "uri_pair" im externen Discovery Document hinterlegt werden.

Im externen Discovery Document MUSS ebenfalls die Adresse des Endpunkts für die alternative Authentisierung (mit einem registrierten Pairing) über das Attribut "auth_pair_endpoint" hinterlegt werden. [≤]

5.4.2.3 Daten und Kommunikationsstrukturen

Die zur Kommunikation zwischen Authenticator-Modul und Pairing-Endpunkt bzw. Authorization-Endpunkt in den Anwendungsfällen:

- Registrierung
- Authentifizierung
- Inspektion
- Deregistrierung

verwendeten Datenobjekte basieren wie in [gemSpec_IDP_Dienst] und [gemSpec_IDP_Frontend] auf JSON-Datenstrukturen mit bzw. als JSON Web Signature (JWS)-Objekte und JSON Web Encryption (JWE)-Objekte für Datenobjekte mit kryptographischem Schutz. Das Interaktionsmuster folgt den REST-Paradigmen. Datenobjekte, die Identitätsinformationen des Nutzers beinhalten, werden auf Anwendungsebene verschlüsselt.

5.4.2.4 Nomenklatur Schlüsselmaterial

Tabelle 8: TAB_IDP_DIENST_0009 Nomenklatur Schlüsselmaterial

Bezeichnung und Beschreibung		Lebenszyklus				
Name	Beschreibung	Verwendungszweck	Erzeugung	Speicherung	Verwendung	Löschung/Ende der Verwendung
PrK_SE_AUT	privater (alternativer) Authentisierungsschlüssel	Schlüsselpaar zur Authentisierung und Authentifizierung	auf dem mobilen Endgerät	auf dem mobilen Endgerät	durch lokal am Gerät authentifizierten Nutzer	Nutzerkommando oder in Reaktion auf definierte Systemereignisse: Auf

		am IDP-Dienst				Initiative des Nutzers oder durch das Gerät bei Änderungen von App, vom Nutzer registrierten Credentials, Rollback von Betriebssystem-Updates und weiteren Ereignissen
PuK_SE_AUT	öffentlicher (alternativer) Authentifizierungsschlüssel			auf dem mobilen Endgerät, IDP-Dienst	durch den IDP-Dienst im Rahmen der Registrierung und Authentifizierung	Erreichen zeitlicher Bedingungen, Ablauf der zur Registrierung verwendeten eGK, Verwendung von ungeeigneten Geräten
PrK.CH.AUT	privater Authentifizierungsschlüssel der eGK	Signatur der Pairing-Daten	eGK	eGK	durch die eGK auf Initiative des Nutzers	-
PuK.CH.AUT	öffentlicher Authentifizierungsschlüssel der eGK	Authentifizierung der Pairing-Daten	eGK	auf dem mobilen Endgerät, persistent am IDP-Dienst	durch den IDP-Dienst im Rahmen der Registrierung und Authentifizierung des Nutzers	Bei Deaktivierung des Pairings erfolgt keine weitere Verwendung des PuK.CH.AUT durch den IDP-Dienst.
C.CH.AUT	Authentifizierungszertifikat der eGK	Authentifizierung der Pairing-Daten	Kartenherausgeber	auf dem mobilen Endgerät, transient am IDP-Dienst	IDP-Dienst im Rahmen der Registrierung und Authentifizierung	Bei Deaktivierung des Pairings, Sperrung oder Ablauf des Zertifikats erfolgt keine weitere Verwendung des C.CH.AUT.
Key-Identifizier	lokaler auf dem Gerät erzeugter Identifikator des Schlüsselpaars PrK_SE_AUT,	Referenzierung des Schlüssels am IDP	Auf dem mobilen Endgerät	mobiles Endgerät, IDP-Dienst	Authentikator-Modul, IDP-Dienst zur Referenzierung	Auf Initiative des Nutzers im Rahmen der Deregistrierung

	PuK_SE_AUT gegenüber dem IDP-Dienst				ung und De-Referenzierung von Pairing-Daten	ung
Biometrische Referenzmerkmale oder andere durch den Nutzer gesetzte wissensbasierte Faktoren	auf dem Gerät erhobene biometrische Referenzmerkmale oder Templates, bzw. Referenzmerkmale zu anderen wissensbasierten Faktoren	lokale Authentifizierung des Nutzers vor Anwendung des PrK_SE_AUT	Nutzer	mobiles Endgerät	Nutzer	Auf Initiative des Nutzers durch das Gerät
idNummer	Telematik-ID (vgl. [gemSpec_IDP_Dienst], Tabelle 5, TAB_IDP_DIENST_0005)	Referenzierung und Dereferenzierung von Daten	Telematikinfrastruktur	über C.CH.AUT auf dem Endgerät, IDP-Dienst zur Referenzierung oder Dereferenzierung von Daten	Identifikation von Pairing-Daten	auf Initiative des Nutzers im Rahmen der Registrierung

5.4.2.5 Registrierung von alternativen Authentisierungsmitteln

Die Registrierung von alternativen Authentisierungsmitteln basiert auf folgenden zwei Voraussetzungen:

1. Authentifizierung des Nutzers am IDP-Dienst: Der Nutzer authentifiziert sich über des in [gemSpec_IDP_Dienst] und [gemSpec_IDP_Frontend] beschriebenen Challenge-Response-Verfahrens unter Nutzung von eGK oder vorliegendem, auf Basis der eGK bezogenem SSO_TOKEN. Hierin eingeschlossen ist die Autorisierung des Authenticator-Moduls durch den Nutzer zur Einrichtung eines alternativen Authentifizierungsmittels. Im Erfolgsfall mündet der Prozess in einem ACCESS_TOKEN für den Pairing-Endpunkt.
2. Lokale Initialisierung des Geräts des Nutzers: Das Authenticator-Modul prüft die Systemumgebung auf das Vorliegen von Voraussetzungen zur Erzeugung und Verwaltung von kryptographischen Schlüsseln, die durch die vorliegende Systemumgebung ausreichend gegen missbräuchliche Verwendung geschützt sind. Im Erfolgsfall erzeugt das Authenticator-Modul über die Betriebssystem-APIs ein Schlüsselpaar. Der öffentliche Schlüssel und andere Metadaten werden vom Nutzer mit Hilfe der eGK durch eine Signatur authentifiziert.

Unter einem ausreichend vorhandenen Schutz wird hierbei eine Systemumgebung verstanden, die den Anforderungen der Kapitel [gemSpec_IDP_Frontend#Kapitel "Parameter für die Schlüsselerzeugung und Auswahl eines Schlüsselspeichers"] und [gemSpec_IDP_Frontend#Kapitel "Erzeugung des Schlüssels und weiterer Metadaten"] genügt.

Die hier dargestellte Reihenfolge ist nicht bindend. Bedingt durch die zweimalige Leistung einer Signatur durch die eGK ist es dem Anbieter des Authenticator-Moduls überlassen, den konkreten Ablauf unter ergonomischen oder Usability-Gesichtspunkten geeignet zu gestalten und den Benutzer im Rahmen der Benutzerführung über die Wirkung der durchgeführten Aktionen aufzuklären. Der Prozess mündet (siehe 3. in der folgenden

Abbildung) in die Registrierung durch die Übertragung der Pairing-Daten und des ACCESS_TOKEN an den Pairing-Endpunkt. Der Pairing-Endpunkt validiert das übertragene ACCESS_TOKEN, die übertragenen Geräteinformationen und die übertragenen Pairing-Daten. Im Erfolgsfall werden die übertragenen Pairing-Daten am Pairing-Endpunkt gespeichert.

Der Prozess wird durch das folgende Sequenzdiagramm illustriert:

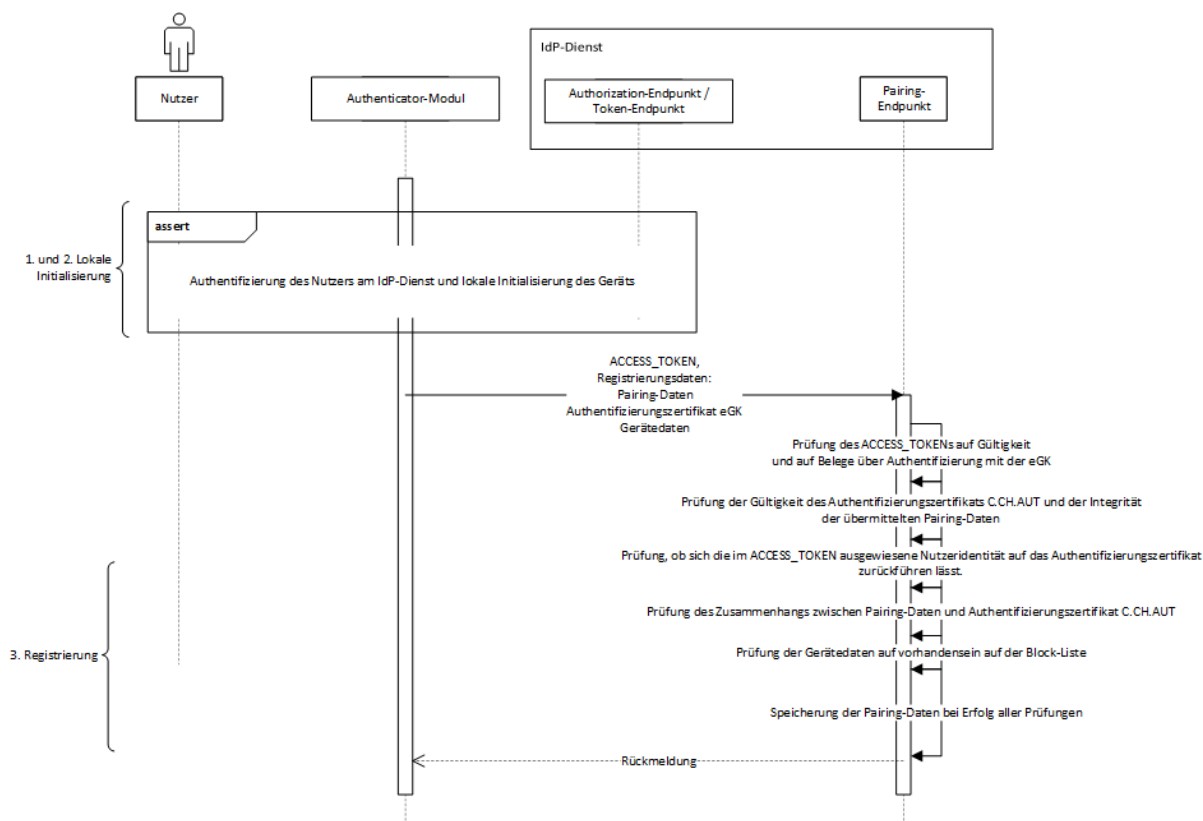


Abbildung 5: Ablauf Registrierungsprozess

Hinweis: Die IDP-Dienst-interne Kommunikation zum Abruf der Pairing-Daten und der Bewertung der Gerätedaten ist in der Abbildung nicht dargestellt. Das Diagramm illustriert die Serviceaktivitäten, nicht die physische Verteilung.

5.4.2.5.1 Erläuterungen zum Registrierungsprozess

5.4.2.5.1.1 Authentifizierung des Nutzers am IDP-Dienst

Der Nutzer authentifiziert sich am IDP-Dienst über das Authenticator-Modul mit Hilfe der eGK oder eines bereits vorhandenen SSO_TOKEN. Das hierbei ausgestellte ACCESS_TOKEN umfasst als Claim die aus dem C.CH.AUT abgeleitete idNummer des Nutzers.

5.4.2.5.1.2 Lokale Initialisierung des Geräts des Nutzers

Dem Authenticator-Modul obliegt die Prüfung auf Vorliegen von geeigneten Voraussetzungen auf dem Endgerät. Die folgenden Aspekte sind hierbei eingeschlossen:

- Verfügbarkeit von geeigneten kryptographischen Algorithmen,
- Vorhandensein eines geeigneten, vom Gerät und Betriebssystem realisierten Schlüsselspeichers,
- Möglichkeit zur Erzeugung eines Schlüsselpaars,

- Möglichkeit zur Zuordnung von lokalen Authentisierungsmitteln zur Anwendung von erzeugten Schlüsseln,
- Durchsetzbarkeit von Anforderungen an die Löschung der erzeugten Schlüssel, z. B. bei Änderung von biometrischen Referenzmerkmalen oder Deinstallation des Accounts.

Das Authenticator-Modul erzeugt das Schlüsselpaar PrK_SE_AUT/PuK_SE_AUT und einen gegenüber dem IDP-Dienst verwendeten Key-Identifizier. Der Nutzer signiert mithilfe des privaten Authentisierungsschlüssels PrK.CH.AUT der eGK die folgenden Daten:

- den öffentlichen Schlüssel PuK_SE_AUT und die zur Anwendung des Schlüssels zu verwendenden Algorithmen,
- die folgenden Daten aus dem Authentifizierungszertifikat C.CH.AUT der eGK:
- der öffentliche Schlüssel und die zur Anwendung des Schlüssels zu verwendenden Algorithmen
- die Seriennummer
- das Gültigkeitsende
- die Informationen zum Aussteller
- den vom Hersteller vergebenen Namen des vom Nutzer verwendeten Geräts,
- ein vom Authenticator-Modul vergebenen Key-Identifizier des Schlüsselpaars PrK_SE_AUT/PuK_SE_AUT zur Identifikation des Schlüssels gegenüber dem IDP-Dienst.

Die Gesamtheit dieser Daten sind die Pairing-Daten.

Zum Einleiten der Registrierung werden die folgenden Daten verschlüsselt an den Pairing-Endpunkt gesandt:

- das bezogene ACCESS_TOKEN,
- die produzierten Pairing-Daten,
- das verwendete Authentisierungszertifikat C.CH.AUT der eGK,
- einen vom Benutzer vergebenen Namen für sein Gerät,
- zum Zeitpunkt der Authentisierung erhobene Geräteinformationen bestehend aus:
- Herstellername,
- den vom Hersteller vergebenen Namen des vom Nutzer verwendeten Geräts,
- Modell,
- Betriebssystem,
- Version des Betriebssystems.

Die Verschlüsselung basiert auf Schlüsseln, die über das Discovery Document des IDP-Dienstes publiziert werden. Im Erfolgsfall werden der Key-Identifizier, das Zertifikat C.CH.AUT und der vom Nutzer vergebene Name des Geräts lokal gespeichert.

Die Aufnahme des vom Hersteller vergebenen Namen des Geräts, dem Produktnamen, in die signierten Pairing-Daten dient dem Nutzer in Kombination mit dem von ihm vergebenen Namen des Geräts der Nachvollziehbarkeit der Zuordnung des Pairings zu einem Gerät bei einer Geräte-übergreifenden Verwaltung seiner Pairing-Daten. Es wird hierbei angenommen, dass der Produktnamen über den gesamten Lebenszyklus des Geräts hinweg konstant bleibt (insbesondere über Betriebssystem-Updates hinaus).

5.4.2.5.1.3 Registrierungsfunktion des Pairing-Endpunkts

Der Pairing-Endpunkt prüft:

- das ACCESS_TOKENS auf Gültigkeit,
- ob anhand der Claims des ACCESS_TOKENS die Authentifizierung auf Basis der eGK nachzuvollziehen ist,
- ob die übermittelten Geräteinformationen sich nicht auf der Block-Liste befinden,
- das übermittelte Authentifizierungszertifikats auf Gültigkeit,
- die Integrität der übermittelten Pairing-Daten mithilfe des öffentlichen Schlüssels aus dem Authentifizierungszertifikat C.CH.AUT,
- ob die im ACCESS_TOKEN vorhandene idNummer identisch mit der des übermittelten Zertifikats C.CH.AUT ist,
- ob die in den Pairing-Daten vorhandenen Angaben zu Seriennummer, Gültigkeitsende und Aussteller identisch zu denen des übermittelten Zertifikats C.CH.AUT sind und ob der in den Pairing-Daten vorhandene öffentliche Schlüssel einschließlich der Angaben zu den Algorithmen identisch zu dem in dem Zertifikat C.CH.AUT ist.

Bei Fehlschlag einer dieser Prüfungen wird der Registrierungsprozess abgebrochen. Bei Erfolg aller dieser Prüfungen werden die folgenden Daten am Pairing-Endpunkt für die weitere Verwendung im Zuge der Authentifizierung hinterlegt:

- die übertragenen Pairing-Daten einschließlich der Signaturdaten,
- den Zeitpunkt der Anlage,
- den vom Nutzer vergebenen Namen für sein Gerät,
- die idNummer und den übertragenen Key-Identifizier.

Die Kombination von idNummer und Key-Identifizier dient zur Referenzierung bzw. Dereferenzierung der Kombination von Pairing-Daten, Zeitpunkt der Anlage und den vom Nutzer vergebenen Namen. Das Authentifizierungszertifikat C.CH.AUT wird hierbei nicht gespeichert. Seine Speicherung obliegt dem Authenticator-Modul für die zukünftige Verwendung des mit dem Prozess etablierten alternativen Authentisierungsmittels.

5.4.2.5.2 Spezifikation

A_21415 - Registrierungsfunktion des IdP-Dienstes: Datenformate und Syntax zur Kommunikation mit dem Authenticator-Modul

Die Registrierungsfunktion des IdP-Dienstes MUSS die folgenden Datentypen zur Kommunikation mit dem Authenticator-Modul verwenden:

- Device_Type
- Device_Information
- Pairing_Data
- Signed_Pairing_Data
- Registration_Data
- Encrypted_Registration_Data.

[<=]

Hinweis: Festlegung zur Ausgestaltung der Datenformate und Kommunikationsprotokolle finden sich im Anhang C dieses Dokuments.

Der Authorization-Endpunkt und der Pairing-Endpunkt müssen das zusätzliche Attribut Security-Patchlevel ("security_patchlevel") in der Authentication_Data/Device_Information/Device_Type-Struktur (Version 1.1) annehmen können ohne es zur Bewertung des Gerätetyps auf Basis der Block/Allow-Liste zu nutzen. Des Weiteren müssen der Authorization-Endpunkt und der Pairing-Endpunkt auch

zukünftig die vorherige Version der Authentication_Data/Device_Information/Device_Type-Struktur (Version 1.0) ohne das Attribut Security-Patchlevel ("security_patchlevel") annehmen.

A_21411 - Registrierungsfunktion des IdP-Dienstes: Fehlermeldungen

Der Pairing-Endpunkt MUSS im Zusammenhang mit der Registrierungsfunktion die folgenden Fehlermeldungen produzieren:

Tabelle 9: Fehlermeldungen im Zusammenhang mit der Registrierungsfunktion

Nummer	Fehlermeldungstext
REG.1	Der Zugriff auf den Dienst kann nicht gewährt werden.
REG.2	Das verwendete Gerät ist nicht für die Authentisierung geeignet.
REG.3	Der erzeugte Schlüssel konnte aufgrund eines internen Fehlers nicht registriert werden.
REG.4	Der erzeugte Schlüssel konnte aufgrund eines bestehenden Eintrags nicht registriert werden.

[<=]

Hinweis: Festlegung zur Ausgestaltung der Datenformate und Kommunikationsprotokolle finden sich im Anhang C dieses Dokuments.

A_21412 - Registrierungsfunktion des IdP-Dienstes: Registrierung

Der Pairing-Endpunkt MUSS zulassen, dass mehrere Geräte pro Nutzer registriert werden.

[<=]

A_21413 - Registrierungsfunktion des IdP-Dienstes: Prüfung des Scope

Der Pairing-Endpunkt MUSS die Nutzung der Registrierungsfunktion ausschließlich auf Basis eines vom IDP-Dienst für das Authenticator-Modul erstellten ACCESS_TOKEN mit dem Scope "openid pairing" ermöglichen. [<=]

A_21425 - Registrierungsfunktion des IdP-Dienstes: Validierung des übermittelten ACCESS_TOKENS

Der Pairing-Endpunkt MUSS die verschlüsselten Registrierungsdaten und den übermittelten ACCESS_TOKEN annehmen. Die Prüfung des ACCESS_TOKEN erfolgt wie in [gemSpec_IDP_FD] beschrieben. Der Pairing-Endpunkt MUSS das bezogene ACCESS_TOKEN mit dem PrK.IDP.ENC entschlüsseln. Das zur Entschlüsselung des ACCESS_TOKEN zu verwendende Verfahren ist ECDH-ES. [<=]

A_21419 - Registrierungsfunktion des IdP-Dienstes: Anforderung an die Authentifizierung des Nutzers

Der Pairing-Endpunkt des IDP-Dienstes MUSS sicherstellen, dass die Registrierung ausschließlich auf Basis von ACCESS_TOKEN durchgeführt wird, die belegen, dass sich der Nutzer auf Basis der eGK authentifiziert hat. Dieser Beleg MUSS aus den Werten für die Claims amr und acr abgeleitet werden. Diese MÜSSEN wie folgt belegt sein: acr="gematik-ehealth-loa-high", amr=["mfa","sc","pin"]. [<=]

A_21420 - Registrierungsfunktion des IdP-Dienstes: Entschlüsselung der Registrierungsdaten

Der Pairing-Endpunkt MUSS die übermittelten Registrierungsdaten mit dem PrK.IDP.ENC entschlüsseln.

[<=]

A_21421 - Registrierungsfunktion des IdP-Dienstes: Validierung der signierten Pairing-Daten

Der Pairing-Endpunkt MUSS die in den übermittelten Registrierungsdaten enthaltenen Pairing-Daten extrahieren und MUSS die Signatur des Nutzers zu den Pairing-Daten prüfen. Die Prüfung des Authentifizierungszertifikats MUSS hierbei auf Basis der aktuellen Systemzeit des Pairing-Endpunkts als Referenzzeit und gemäß [gemSpec_PKI], Abschnitt 8.3.1.1 „TUC_PKI_018“ erfolgen. Der IdP-Dienst MUSS zur Validierung alle Algorithmen unterstützen, die im Zusammenhang mit den Authentisierungsmitteln verwendet werden, die zur Authentisierung am IdP-Dienst zugelassen sind (siehe [gemSpec_IDP_Frontend], Abschnitt 3). Hierbei MÜSSEN die Vorgaben aus Abschnitt 9.3.4 [gemSpec_IDP_Frontend] beachtet werden. Andere Algorithmen DÜRFEN NICHT unterstützt werden. Kann die Authentizität der Pairing-Daten nicht belegt werden oder ist das übermittelte Authentifizierungszertifikat zum aktuellen Zeitpunkt ungültig oder gesperrt, MUSS der Pairing-Endpunkt die Anfrage mit der Fehlermeldung REG.1 beenden. Unter "Authentifizierungszertifikat" wird ein Zertifikat des Typs C.CH.AUT verstanden. [<=]

A_21422 - Registrierungsfunktion des IdP-Dienstes: Validierung des Zusammenhangs von ACCESS_TOKEN und Pairing-Daten

Der Pairing-Endpunkt MUSS überprüfen, ob sich das übermittelte ACCESS_TOKEN und sich die im Authentifizierungszertifikat eingebetteten Identitätsinformationen auf ein und dieselbe Entität zurückführen lassen: Der Pairing-Endpunkt MUSS dafür prüfen, ob der im ACCESS_TOKEN eingebettete Claim idNummer identisch mit der im Authentifizierungszertifikat vorhandenen idNummer ist. Sind diese nicht identisch, MUSS der Pairing-Endpunkt die Anfrage mit der Fehlermeldung REG.1 beenden. Unter Authentifizierungszertifikat wird hierbei ein Zertifikat vom Typ C.CH.AUT verstanden. [<=]

A_21470 - Registrierungsfunktion des IdP-Dienstes: Prüfung des Zusammenhangs zwischen Pairing-Daten und übermittelten Authentifizierungszertifikat

Der Pairing-Endpunkt des IdP-Dienstes MUSS überprüfen, ob sich die in den Pairing-Daten befindlichen Daten „issuer“, „serialnumber“, „auth_cert_subject_public_key_info“, „not_after“ identisch zu den Zertifikatsfeldern „issuer“, „serialNumber“, „subjectPublicKeyInfo“, „notAfter“ des übermittelten Authentifizierungszertifikats sind. Schlägt einer dieser Vergleiche fehl, MUSS der Pairing-Endpunkt die Anfrage mit der Fehlermeldung REG.1 beenden. Unter "Authentifizierungszertifikat" wird ein Zertifikat des Typs C.CH.AUT verstanden. [<=]

A_21423 - Registrierungsfunktion des IdP-Dienstes: Bewertung des Gerätetyps

Der Pairing-Endpunkt des IdP-Dienstes MUSS die vom Authenticator-Modul übertragenen Geräteinformationen gegen die Block/Allow-Liste prüfen. Bei Zuordenbarkeit der Geräteinformationen zu einem Gerätetyp auf der Block-Liste MUSS der Pairing-Endpunkt den Registrierungsvorgang mit der Fehlermeldung REG.2 beenden. [<=]

A_21424 - Registrierungsfunktion des IdP-Dienstes: Speicherung der Pairing-Daten

Der Pairing-Endpunkt MUSS die folgenden Daten und den Zeitpunkt der Speicherung in seiner Pairing-Datenbank speichern:

- den Namen des Geräts (aus den in den Registrierungsdaten übertragenen Geräteinformationen im Datentyp "Device_Information")
- die signierten Pairing-Daten
- den aktuellen Zeitpunkt.

Die logische Kombination dieser Daten MUSS über die idNummer referenzierbar sein und sie MUSS über die Kombination aus (idNummer, keyIdentifier) eindeutig referenzierbar sein. Unter „keyIdentifier“ wird hierbei das Feld „key_Identifier“ aus den Pairing-Daten verstanden. Eventuelle Kollisionen mit bestehenden Pairing-Daten MÜSSEN mit der

Fehlermeldung REG.4 begegnet werden. Der Pairing-Endpunkt DARF die übermittelten Daten in diesem Fall NICHT speichern.[<=]

A_21426 - Registrierungsfunktion des IdP-Dienstes: Nicht-Speicherung des übermittelten Authentifizierungszertifikats

Der Pairing-Endpunkt DARF das im Zuge der Registrierung in den Registrierungsdaten übertragene Authentifizierungszertifikat NICHT persistent speichern. Unter "Authentifizierungszertifikat" wird ein Zertifikat des Typs C.CH.AUT verstanden.[<=]

A_21427 - Registrierungsfunktion des IdP-Dienstes: Rückmeldung an den Nutzer

Der Pairing-Endpunkt MUSS dem Authenticator-Modul eine eindeutige Rückmeldung über den Erfolg oder Misserfolg der Anlage des Pairings übermitteln.[<=]

5.4.2.6 Verwendung von alternativen Authentisierungsmitteln

5.4.2.6.1 Erweiterung des Authorization-Endpunkts

Der Authorization-Endpunkt verwendet den im Zuge der Registrierung den in den Pairing-Daten hinterlegten öffentlichen Schlüssel zur Authentifizierung. Die erfolgreiche Authentifizierung ist abhängig von:

- der Gültigkeit des Authentisierungsmittels, das zur Registrierung des alternativen Authentisierungsmittels verwendet wurde
- der Integrität der Pairing-Daten und deren Bezug zum Authentisierungsmittel, welches bei der Registrierung verwendet wurde
- der Eignung des Geräts
- weiteren zeitlichen Bedingungen an die Verwendungszeit des auf dem Gerät des Nutzers erzeugten Authentifizierungsschlüssels zum Zweck der Authentifizierung in Abhängigkeit der festgestellten Eignung des Geräts
- der Prüfung der Signatur der Authentifizierungsdaten mithilfe des in den gespeicherten Pairing-Daten vorhandenen auf dem Gerät des Nutzers erzeugten Authentifizierungsschlüssels.

Bei Fehlschlag einer dieser Prüfungen gilt der Benutzer als nicht authentifiziert. Die Bewertung des Geräts erfolgt hierbei auf Basis von Informationen, die vom Authenticator-Modul erhoben werden.

Die Verwendung von alternativen Authentisierungsmitteln gestaltet sich in zwei Schritten:

1. Ein Anwendungsfrontend initiiert über das Authenticator-Modul einen Authorization Request an den Authorization-Endpunkt des IDP-Dienstes. Der Authorization-Endpunkt produziert ein Challenge-Token und überträgt dieses an das Authenticator-Modul. Der Nutzer signiert über das Authenticator-Modul eine Datenstruktur, die die folgenden Inhalte umfasst:
 - das empfangene Challenge-Token
 - das lokal gespeicherte Authentifizierungszertifikat
 - aktuelle Geräteinformationen und
 - Angaben zur vom Nutzer verwendeten Methode zur Freischaltung des Authentisierungsschlüssels.
2. Der Authorization-Endpunkt prüft die empfangenen Daten und authentifiziert den Nutzer im Erfolgsfall.

Der Ablauf ist in dem folgenden Sequenzdiagramm dargestellt:

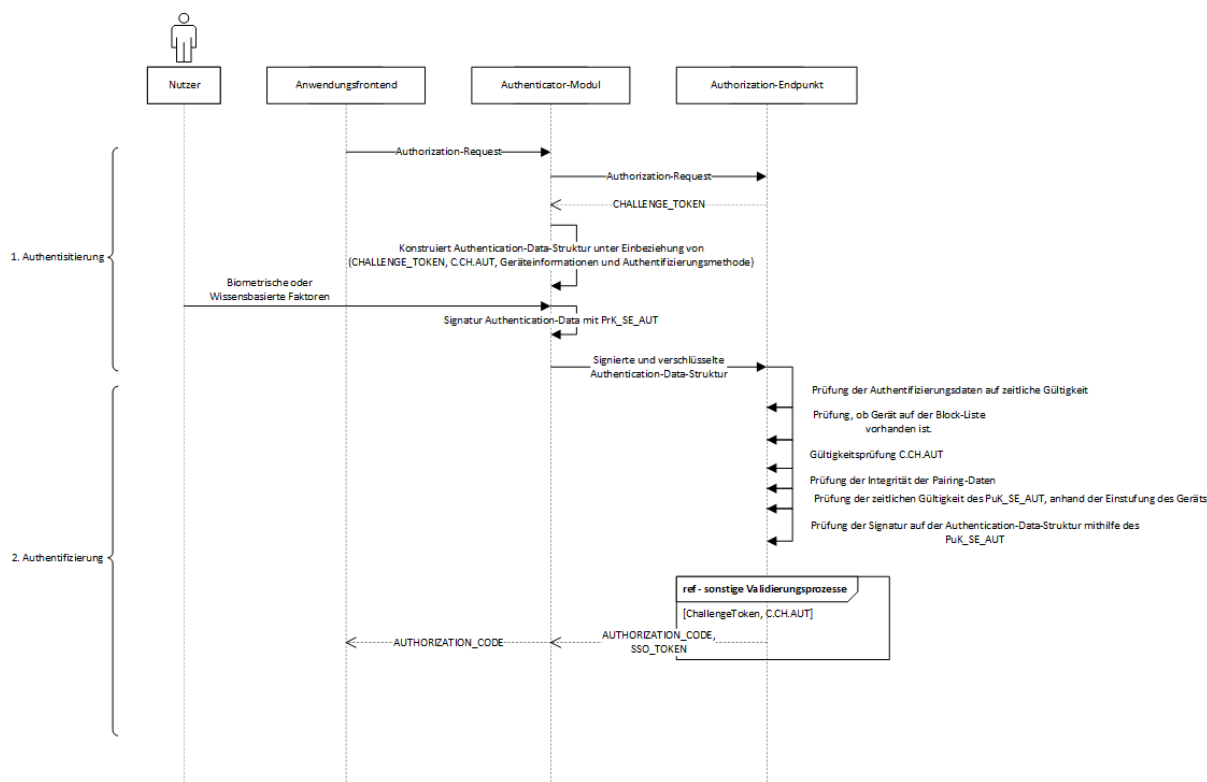


Abbildung 6: Ablauf Verwendung alternativer Authentisierungsmittel

Hinweis: Die IDP-Dienst-interne Kommunikation zum Abruf der Pairing-Daten und der Bewertung der Gerätedaten ist in der Abbildung nicht dargestellt. Das Diagramm illustriert die Serviceaktivitäten, nicht die physische Verteilung.

5.4.2.6.2 Erläuterungen zur Verwendung von alternativen Authentisierungsmitteln

5.4.2.6.2.1 Authentisierung auf Basis von alternativen Authentisierungsmitteln

Um eine Authentifizierung mit alternativen Authentisierungsmitteln zu ermöglichen, wird das in [gemSpec_IDP_Dienst], unter [3.3- Verfahrensbeschreibung] beschriebene Challenge-Response-Verfahren zur Nutzerauthentifizierung um die Möglichkeit zur Validierung von signierten Challenge-Token erweitert, die nicht auf Basis von Schlüsseln der eGK signiert wurden, sondern durch den PrK_SE_AUT eines Gerätes. Die Initiierung des Prozesses erfolgt wie in [gemSpec_IDP_Dienst] beschrieben über das Anwendungsfondend und einen Authorization Request an den Authorization-Endpunkt. Die Antwort des Authenticator-Moduls auf die Challenge des Authorization-Endpunkts gestaltet sich wie folgt: Das Authenticator-Modul produziert eine Datenstruktur, die die folgenden Informationen enthält:

- das vom IDP-Dienst bezogene Challenge-Token
- das lokal gespeicherte, zur Anlage des Pairings verwendete Authentifizierungszertifikat der eGK
- den lokal gespeicherten Key-Identifizierer des Schlüsselpaars PrK_SE_AUT/PuK_SE_AUT
- aktuell vom Authenticator-Modul erhobene Geräteinformationen und
- Informationen über die vom Nutzer verwendeten, lokalen Authentisierungsmittel.

Die Gesamtheit dieser Daten wird im Folgenden "Authentisierungsdaten" genannt. Der Nutzer signiert diese Daten mit dem PrK_SE_AUT und das Authenticator-Modul überträgt

diese als Response auf die bezogene Challenge an den Authorization-Endpunkt. Die übertragenen Daten werden auf Basis von öffentlichen Schlüsseln, die über das Discovery Document des IDP-Dienstes publiziert werden, verschlüsselt. Den verschlüsselten Daten werden Angaben zur zeitlichen Gültigkeit beigefügt, die denen des bezogenen Challenge-Token entsprechen.

5.4.2.6.2.2 Authentifizierungsvorgang

Die Validierung der empfangenen Authentifizierungsdaten gestaltet sich wie folgt:

Der Authorization-Endpunkt prüft die empfangenen Authentifizierungsdaten vor Entschlüsselung auf zeitliche Gültigkeit. Er vollzieht nach Entschlüsselung die folgenden Schritte:

1. Prüfung, ob die übertragenen Geräteinformationen auf der Block-Liste hinterlegt sind. Ist dies der Fall, wird der Authentifizierungsvorgang abgebrochen.
2. Prüfung, ob das übermittelte Authentifizierungszertifikat C.CH.AUT gültig ist. Ist dieses gesperrt oder abgelaufen, wird der Authentifizierungsvorgang abgebrochen.
3. Der gespeicherte Pairing-Eintrag wird anhand der Kombination von idNummer aus dem Authentifizierungszertifikat C.CH.AUT und dem übertragenen Key-Identifizierer dereferenziert. Existiert kein solcher, wird der Authentifizierungsvorgang abgebrochen.
4. Die ursprünglich im Rahmen des Registrierungsprozesses vom Nutzer geleistete Signatur der Pairing-Daten wird vom Authorization-Endpunkt auf mathematische Integrität überprüft. Erweisen sich die hinterlegten Daten als verfälscht, wird der Authentifizierungsvorgang abgebrochen.
5. Der in den Pairing-Daten vorhandene öffentliche Schlüssel PuK_SE_AUT wird zur Authentifizierung der übermittelten Authentifizierungsdaten angewendet, wenn
 - a. die übermittelten Gerätedaten entweder auf ein Gerät auf der Allow-Liste verweisen oder
 - b. andernfalls der Zeitpunkt der Anlage des Pairings nicht mehr als 6 Monate zurückliegt.
6. Der Authorization-Endpunkt prüft die mathematische Integrität der übermittelten Authentifizierungsdaten mithilfe des öffentlichen Schlüssels PuK_CH_AUT aus den hinterlegten Pairing-Daten.

Bei Fehlschlag einer Prüfung wird der Prozess abgebrochen. Bei erfolgreichem Durchlaufen aller Prüfungen gilt der Nutzer als authentifiziert. Alle weiteren Prozesse des Authorization-Endpunkts wie in [5.2- Authorization-Endpunkt] beschrieben werden durchlaufen und münden in der Produktion eines Authorization-Code und eines SSO_TOKEN. Die vom Authenticator-Modul übertragenen Informationen über die Art der vom Nutzer verwendeten lokalen Authentifizierungsmethode zur Anwendung des PrK_SE_AUT werden als Claim in SSO_TOKEN und AUTHORIZATION_CODE verankert.

5.4.2.6.3 Spezifikation

A_21449 - Erweiterung des Authorization-Endpunkts: Datenmodell und Syntax zur Kommunikation mit dem Authenticator-Modul

Der Authorization-Endpunkt des IdP-Dienstes MUSS bei der Verwendung von alternativen Authentifizierungsmitteln die folgenden Datentypen zur Kommunikation mit dem Authenticator-Modul verwenden:

- Authentication_Data
- Signed_Authentication_Data
- Encrypted_Signed_Authentication_Data.

[<=]

A_21428 - Erweiterung des Authorization-Endpunkts: Fehlermeldungen

Der Authorization-Endpunkt MUSS bei Authentifizierung von Nutzern auf Basis von alternativen Authentisierungsmitteln die folgenden Fehlermeldungen produzieren:

Tabelle 10: Fehlermeldungen des Authorization-Endpunkts bei Authentifizierung auf Basis von alternativen Authentisierungsmitteln

Nummer	Fehlermeldungstext
VAL.1	Die Authentifizierung mit einem alternativen Authentisierungsmittel konnte nicht erfolgreich durchgeführt werden.

[<=]

Hinweis: Anforderungen an die initiale Verarbeitung der im Zuge der Authentifizierung empfangenen Datenstruktur "Authentication_Data" stellt die Anforderung A_20699-03.

A_21432 - Erweiterung des Authorization-Endpunkts: Prüfung der vorliegenden Gerätedaten

Der Authorization-Endpunkt MUSS die in der Authentication_Data/Device_Information-Struktur dargelegten Informationen zum Gerätetyp auf Basis der Block/Allow-Liste bewerten. Sofern das Gerät als nicht geeignet im Sinne der Anforderung A_21404 eingestuft ist, MUSS der Authentifizierungsvorgang mit der Fehlermeldung „access_denied“ und der „error_description“ VAL.1 genannt abgebrochen werden. Die Zuordnung zwischen gelieferten Informationen zum Gerätetyp und der Datenbasis der Block/Allow-Liste MUSS auf Basis des Vergleichs der in den einzelnen Claims enthaltenen Strings erfolgen.[<=]

A_21433 - Erweiterung des Authorization-Endpunkts: Validierung des Authentifizierungszertifikats

Der Authorization-Endpunkt MUSS die Gültigkeit des in der Signed_Authentication_Data-Struktur gelieferten Authentifizierungszertifikats überprüfen. Die Zertifikatsprüfung des Authentifizierungszertifikats muss hierbei auf Basis der Systemzeit des Pairing-Endpunkts als Referenzzeit erfolgen. Die Prüfung des Zertifikats MUSS gemäß [gemSpec_PKI], Abschnitt 8.3.1.1 „TUC_PKI_018“ erfolgen. Bei ungültigem oder gesperrtem Zertifikat MUSS der Authorization-Endpunkt die Verarbeitung abbrechen. Die Fehlermeldungen MUSS „access_denied“ mit „error_description“ VAL.1 sein. Unter "Authentifizierungszertifikat" wird ein Zertifikat des Typs C.CH.AUT verstanden.[<=]

A_21434 - Erweiterung des Authorization-Endpunkts: Auslesen der gespeicherten Pairing-Daten

Der Authorization-Endpunkt MUSS die idNummer aus dem übermittelten Authentifizierungszertifikat entnehmen (vergleiche A_20524 [gemSpec_IDP_Dienst] und [gemSpec_PKI], Abschnitt 4.2 zur Erläuterung). Der Authorization-Endpunkt MUSS anhand des in den Signed_Authentication_Data übermittelten Key-Identifiers und der extrahierten idNummer die Pairing-Daten aus der Pairing-Datenbank beziehen. Falls kein solcher Datensatz existiert, MUSS der Authorization-Endpunkt den Prozess mit einer Fehlermeldung abbrechen. Die Fehlermeldungen MUSS „access_denied“ mit „error_description“ VAL.1 sein. Unter "Authentifizierungszertifikat" wird ein Zertifikat des Typs C.CH.AUT verstanden.[<=]

A_21435 - Erweiterung des Authorization-Endpunkts: Validierung der mathematischen Integrität der gespeicherten Pairing-Daten

Der Authorization-Endpunkt MUSS die Integrität der hinterlegten Pairing-Daten mithilfe des öffentlichen Schlüssels des Authentifizierungszertifikats überprüfen. Erweisen sich die (mathematische) Integrität der Daten nicht als gewährleistet, MUSS die Verarbeitung mit

einer Fehlermeldung abgebrochen werden. Die Fehlermeldungen MUSS „access_denied“ mit „error_description“ VAL.1 sein. Unter "Authentifizierungszertifikat" wird ein Zertifikat des Typs C.CH.AUT verstanden.

[<=]

A_21437 - Erweiterung des Authorization-Endpunkts: Bewertung der Gültigkeit des öffentlichen Schlüssels in den Pairing-Daten

Der Authorization-Endpunkt MUSS die Eignung des in den Pairing-Daten gespeicherten öffentlichen Schlüssels PuK_SE_AUT zur Prüfung der Integrität der Signed_Authentication_Data-Struktur wie folgt bewerten:

- Falls der Typ des Geräts des Endnutzers eine Einstufung als uneingeschränkt geeignet im Sinne von A_21404 besitzt, ist der öffentliche Schlüssel für die Validierung der Signed_Authentication_Data-Struktur zu verwenden.
- Falls andernfalls der Typ des Geräts des Endnutzers eine Einstufung als eingeschränkt geeignet im Sinne von A_21404 besitzt, ist der öffentliche Schlüssel für die Validierung der Signed_Authentication_Data-Struktur geeignet, wenn der Anlagezeitpunkt des Pairings nicht mehr als 6 Monate zurückliegt (d.h. sofern die aktuelle Zeit < pairing_entry.creationTime+6 Monate ist). Hierbei DARF auf den vollen Tag aufgerundet werden. Ist dies nicht der Fall, so MUSS die Verarbeitung mit einer Fehlermeldung abgebrochen werden. Die Fehlermeldungen MUSS „access_denied“ mit „error_description“ VAL.1 sein.

[<=]

Hinweis: Die Prüfung auf ein ungeeignetes Gerät durch Abfrage der Block/Allow-Liste erfolgt in Anforderung A_21432.

A_21438 - Erweiterung des Authorization-Endpunkts: Validierung der mathematischen Integrität der signierten Authentication_Data-Struktur

Der Authorization-Endpunkt MUSS die Integrität der übermittelten Signed_Authentication_Data mithilfe des öffentlichen Schlüssels aus dem Pairing-Datensatz überprüfen. Sofern sich die Daten als nicht (mathematisch) integer erweisen MUSS der Authorization-Endpunkt den Prozess mit einer Fehlermeldung abbrechen. Die Fehlermeldungen MUSS „access_denied“ mit „error_description“ VAL.1 sein.

[<=]

A_21439 - Erweiterung des Authorization-Endpunkts: Zu unterstützende Algorithmen zur Prüfung der mathematischen Integrität der signierten Authentication_Data-Struktur

Der Authorization-Endpunkt MUSS im Zuge der Validierung der mathematischen Integrität der Signed_Authentication_Data-Struktur die folgenden Algorithmen unterstützen: ECDSA mit Kurve P-256 und SHA-256.

[<=]

A_21440 - Erweiterung des Authorization-Endpunkts: Produktion des Authorization Code und eines SSO_TOKEN

Der Authorization-Endpunkt MUSS nach erfolgreicher Prüfung der Signed_Authentication_Data einen Authorization Code und einen SSO_TOKEN produzieren. Basis für die Claims liefert das übertragene Authentifizierungszertifikat und die Claims aus der Datenstruktur Authentication_Data. Unter "Authentifizierungszertifikat" wird ein Zertifikat des Typs C.CH.AUT verstanden. In den produzierten SSO_TOKEN und AUTHORIZATION_CODE MUSS der amr-Claim identisch zu dem gleichnamigen Claim in der Authentication_Data-Struktur gesetzt werden.[<=]

5.4.2.7 Inspektion und De-Registrierung der am Pairing-Endpunkt gespeicherten Daten

5.4.2.7.1 Inspektion am Pairing-Endpunkt

Die Inspektionsfunktion gestattet dem Nutzer unter Verwendung des Authenticator-Moduls die von ihm am Pairing-Endpunkt gespeicherten Daten in Augenschein zu nehmen. Bedingung für die Nutzung der Inspektionsfunktion ist die Authentisierung des Nutzers am IDP-Dienst (unter Verwendung von eGK, SSO_TOKEN oder Pairing) belegt durch die Vorlage eines gültigen ACCESS_TOKEN. Die Prüfung des ACCESS_TOKEN beinhaltet keine weiteren Anforderungen, die über die in [gemSpec_IDP_FD#Kapitel 6] hinausgehen. Die Authentifizierung kann hierbei auf Basis jedes vom IDP-Dienst zur Authentifizierung akzeptierten Mittels erfolgen. Das ACCESS_TOKEN enthält die ID-Nummer des Nutzers. Anhand der idNummer lassen sich alle am Pairing-Endpunkt hinterlegten Pairing-Einträge dereferenzieren. Diese werden dem Nutzer zur Ansicht übermittelt. Sie umfassen:

- die signierten Pairing-Daten,
- den Zeitpunkt der Anlage des Pairings und
- den vom Nutzer zugewiesenen Namen des Geräts.

Der Ablauf ist im folgenden Sequenzdiagramm dargestellt:

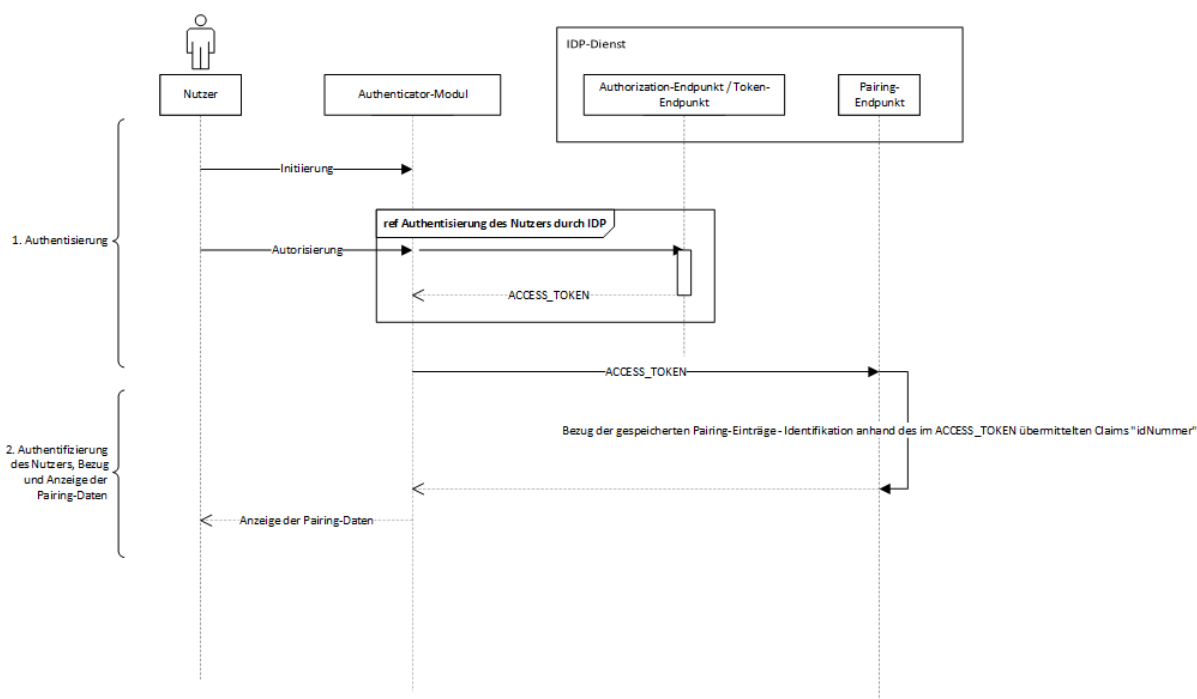


Abbildung 7: Inspektion der Pairing-Daten

5.4.2.7.2 Deregistrierung von alternativen Authentisierungsmitteln

Die Deregistrierungsfunktion gestattet dem Nutzer, von ihm gespeicherte Pairing zu deaktivieren, das heißt den öffentlichen Schlüssel dauerhaft von der Verwendung als Mittel zur Authentifizierung auszuschließen. Bedingung für die Nutzung der Deregistrierungsfunktion ist das Vorliegen eines gültigen ACCESS_TOKEN und die Darstellung der Daten analog wie in [5.4.2.7.1- Inspektion am Pairing-Endpunkt] beschrieben.

Der Nutzer erhält die Option zur Auswahl eines zu deaktivierenden Pairing. Das Authenticator-Modul überträgt den Key-Identifizier des gewählten Pairing-Datensatzes und den bereits bezogenen ACCESS_TOKEN zum Pairing-Endpunkt. Der Pairing-Endpunkt referenziert den zu deaktivierenden Pairing-Eintrag anhand der idNummer im ACCESS_TOKEN und dem übermittelten Key-Identifizier. Der Anbieter des IDP-Dienstes ist verpflichtet, den identifizierten Pairing-Eintrag binnen einer definierten Frist von der Verwendung zur Authentifizierung dauerhaft auszuschließen. Die Konkretisierung der Fristsetzung findet sich in den folgenden Anforderungen.

Motivation: Die Realisierung der Funktion soll dem Nutzer eine umfassende Möglichkeit geben, alternative Authentisierungsmittel voraussetzungslos zu deaktivieren (z. B. in Situationen, in denen ein Verlust oder Defekt eines Gerätes eine lokale Löschung von Authentisierungsmitteln ausschließt).

Der Ablauf ist in folgendem Sequenzdiagramm dargestellt:

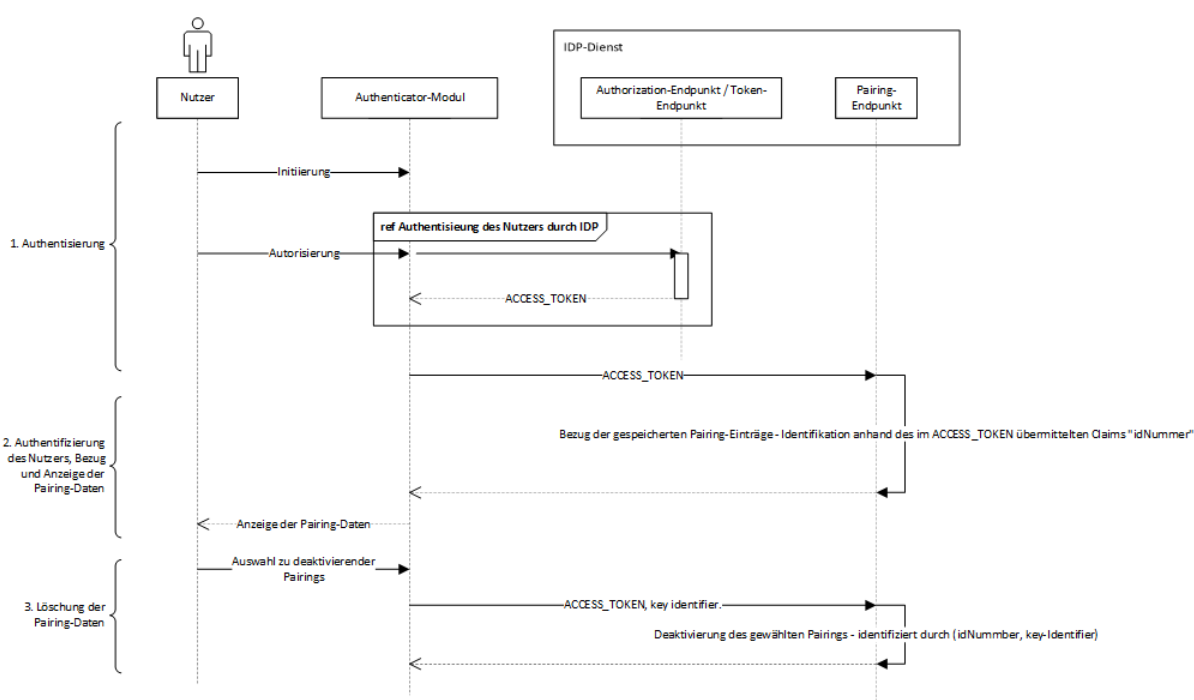


Abbildung 8: Deregistrierung eines Pairings

5.4.2.7.3 Spezifikation

A_21450 - Inspektions- und Deregistrierungsfunktion des IdP-Dienstes: Datenmodell und Syntax zur Kommunikation mit dem Authenticator-Modul

Die Inspektions- und Deregistrierungsfunktion des IdP-Dienstes MUSS die folgenden Datentypen zur Kommunikation mit dem Authenticator-Modul verwenden:

- Pairing_Entry
- Pairing_Entries.

[<=]

Hinweis: Die Ausgestaltung der Datenformate und Kommunikationsprotokolle ist im Anhang C dieses Dokuments dargelegt.

A_21441 - Inspektions- und Deregistrierungsfunktion des IDP-Dienstes: Fehlermeldungen

Der Pairing-Endpunkt des IdP-Dienstes MUSS im Zusammenhang mit der Inspektions- und Deregistrierungsfunktion die folgenden Fehlermeldungen produzieren:

Tabelle 11: Fehlermeldungen Inspektions- und Deregistrierungsfunktion

Nummer	Fehlermeldungstext
AC.1	Der Zugriff auf den Dienst kann nicht gewährt werden.
AC.2	Die Registrierungsdaten konnten nicht bezogen werden.
AC.3	Der Auftrag zur Deaktivierung des Pairings konnte nicht angenommen werden.

[<=]

Hinweis: Die Ausgestaltung der Fehlermeldungen ist im Anhang C dieses Dokuments dargelegt.

A_21445 - Inspektions- und Deregistrierungsfunktion des IdP-Dienstes: Validierung und Verarbeitung des "ACCESS_TOKEN"

Der Pairing-Endpunkt MUSS das bezogene ACCESS_TOKEN mit dem privaten Schlüssel PrK.IDP.ENC entschlüsseln. Das zur Entschlüsselung des ACCESS_TOKEN zu verwendende Verfahren ist ECDH-ES. Die Prüfung des ACCESS_TOKEN erfolgt wie in [gemSpec_IDP_FD] beschrieben. Sofern das ACCESS_TOKEN ungültig ist, MUSS dem Authenticator-Modul die Fehlermeldung AC.1 übermittelt werden.[<=]

A_21452 - Inspektions- und Deregistrierungsfunktion des IdP-Dienstes: Rückgabe der Pairing-Daten

Der Pairing-Endpunkt MUSS die idNummer des Benutzers aus den Claims des ACCESS_TOKEN extrahieren und die Pairing-Datenbank anhand der idNummer nach den bestehenden Pairings des Benutzers abfragen. Die bezogenen Pairing-Daten MÜSSEN dem Authenticator-Modul übermittelt werden. Die Antwort MUSS eine Liste von Pairing_Entry-Daten sein. Sofern die Daten nicht bezogen werden konnten, MUSS dem Authenticator-Modul die Fehlermeldung AC.2 übermittelt werden.[<=]

A_21447 - Inspektions- und Deregistrierungsfunktion des IdP-Dienstes: Annahme des Kommandos zur Deaktivierung des Pairings

Der Pairing-Endpunkt MUSS das Kommando zum Deaktivieren eines Pairing-Eintrags annehmen. Das Kommando umfasst dabei das ACCESS_TOKEN und den Key-Identifizierer des zu deaktivierenden Pairing-Eintrages. Der Pairing-Eintrag MUSS anhand von Key-Identifizierer und der im ACCESS_TOKEN enthaltenen idNummer identifiziert und deaktiviert werden.[<=]

A_21448 - Inspektions- und Deregistrierungsfunktion des IdP-Dienstes: Deaktivierung des identifizierten Pairing-Datensatzes

Der Pairing-Endpunkt MUSS den zur Deaktivierung angefragten Pairing-Datensatz anhand der Kombination aus dem übermittelten Key-Identifizierer und der aus dem ACCESS_TOKEN bezogenen idNummer identifizieren. Das verwendete Pairing MUSS dauerhaft deaktiviert werden. Die zur Authentifizierung verwendeten Authentifizierungsschlüssel dürfen nicht mehr verwendet werden. Sofern der Auftrag zur Deaktivierung nicht angenommen werden konnte, MUSS der Pairing-Endpunkt dem Authenticator-Modul die Fehlermeldung AC.3 übermitteln.[<=]

A_21454 - Performance - IdP-Dienst - Deregistrierung eines Pairings

Der Anbieter des IdP-Dienstes MUSS sicherstellen, dass ein in der Pairing-Datenbank gespeichertes Pairing auf Anfrage des Nutzers innerhalb einer Stunde deaktiviert wird. Dies gilt auch für eine standortübergreifende Realisierung der einzelnen Endpunkte des IdP-Dienstes.[<=]

5.5 Authentisierung über andere IDP

Der IDP-Dienst muss in seiner Rolle als Authorization Server des E-Rezept Fachdienstes einen Third-Party-Authorization Endpoint bereitstellen, an welchem Authorization Requests eingereicht werden, bei denen die eigentliche Authentisierung anschließend unter Nutzung eines sektoralen Identity Provider durchgeführt werden soll. Nach erfolgter Authentisierung erzeugt der IDP-Dienst aus den Daten des abgerufenen ID_TOKEN die eigenen ACCESS_TOKEN für das Anwendungsfrontend und den Fachdienst.

A_23897 - Bereitstellung eines Federation Authorization Endpoint

Der IDP-Dienst MUSS einen Federation Authorization Endpoint unter einer dedizierten URL bereitstellen und dort Authorization Requests des Anwendungsfrontend annehmen sowie, als Antwort auf einen eigenen Authentication Request weitergeleitete, Authorization Codes eines sektoralen Identity Provider verarbeiten.

[<=]

A_23686-01 - Sitzungsmanagement des IDP-Dienstes zu einem sektoralen Identity Provider der Föderation

Der IDP-Dienst MUSS für Anfragen zur Authentisierung an einem sektoralen Identity Provider der Föderation die folgenden Daten zur späteren Korrelation der Anfragen in einer Sitzung speichern:

Quelle	Parameter	Erläuterung
E-Rezept-FdV/Daten aus dem Authorization Request des E-Rezept-FdV	client_id	Dies ist die client-id des Anwendungsfrontend.
	nonce_erp	Dies ist die Nonce für das ID-Token, das zum Anwendungsfrontend gesendet werden soll.
	state_erp	state-Parameter des Anwendungsfrontends
	code_challenge	code_challenge des Anwendungsfrontends
	redirect_uri	redirect_uri aus dem initialen GET Request des Anwendungsfrontends
IDP-Dienst/Daten des Authorization Request des IDP-Dienstes an den sektoralen Identity Provider	nonce_idp	Nonce für die Verifikation des ID-Token des sektoralen Identity Provider
	code_verifier	Code-Verifier des IDP-Dienstes zur Abfrage des ID-Token
	idp_iss	Identifizier für den angefragten sektoralen Identity Provider
	state_idp	state-Parameter des IDP-Dienstes

		zur Abfrage des ID-Token. Über diesen Wert wird die Sitzung anschließend referenziert.
--	--	---

Der IDP-Dienst MUSS diese Daten über den von ihm generierten state-Parameter referenzieren und in den späteren Anwendungsfällen dereferenzieren können. [<=]

5.5.1 Authentifizierung über andere IDP - Eingangsdaten

Der Request geht als Antwort in Form eines HTTP-302-redirect auf den Authorisierungs-Request zurück an das Authenticator-Modul (vergleiche Schritt 2 in [\[gemSpec_IDP_Sek#2.2\]](#)). Die "target_url" des redirect ist dabei die URL des Authenticator-Moduls des sektoralen Identity Provider, welches durch den Parameter "kk_app_id" gewählt wurde.

A_25222 - Liste der redirect_uris im Entity Statement

Der IDP-Dienst MUSS in seiner Rolle als Authorization Server des E-Rezept-Fachdienstes in seinem Entity Statement im claim redirect_uris die redirect_uris aller Clients listen, welche eine Authentisierung mit GesundheitsID unterstützen. [<=]

A_23687-01 - Pushed Authorization-Request des IDP-Dienstes an sektorale Identity Provider

Wenn der Authorization Request einen Parameter idp_iss mit dem Identifikator eines sektoralen Identity Provider enthält, MUSS der IDP-Dienst selbst als Client einen Pushed Authorization Request (PAR) gemäß [\[gemSpec_IDP_FD#AF_10117\]](#) direkt an den zugehörigen sektoralen Identity Provider stellen. Hierbei sind die folgenden Parameter zu verwenden:

Parameter	Wert	Anmerkung
client_id	"https://idp.app.ti-dienste.de"	Identifiziert den IDP-Dienst als Relying Party innerhalb der Föderation
state	dynamisch und zufällig zu erzeugen	Dieser State wird verwendet, um die später erhaltene Antwort zuzuordnen.
redirect_uri	Übernahme der redirect_uri aus dem initialen GET Request des Anwendungsfrontends	Die Adresse des Authorization Endpunkts zur Annahme der Antwort vom sektoralen IDP ist die redirect_uri aus der Anfrage des Clients. Die zu der E-Rezept-App gehörige uri ist z.B. https://das-e-rezept-fuer-deutschland.de/extauth .
code_challenge	mit der Methode "S256" erzeugter Hash des Code-Verifiers	

code_challenge_method	"S256"	
response_type	code	
nonce	dynamisch und zufällig zu erzeugen	Nonce für das vom sektoralen IDP-Dienst zu erzeugende ID-Token
scope	"openid urn:telematik:display_name urn:telematik:versicherter"	Notwendige Scopes für den Zugriff für die Autorisierung von Nutzern am E-Rezept Fachdienst
acr_values	"gematik-ehealth-loa-high"	Angefordertes Niveau der Nutzerauthentisierung

[<=]

A_23688 - Authorization Request nach erfolgreichem Pushed Authorization-Request

Wenn der Pushed Authorization Request (PAR) korrekt mit einer request_uri beantwortet wurde MUSS der IDP-Dienst einen Authorization Request in Form eines HTTP-302-redirect auf den Autorisierungs-Request zurückgeben. Die target_url des redirect ist dabei die URL des Authorization-Endpunkt des sektoralen IDP. Der Request wird dann vom Anwendungsfreond an das Authenticator Modul des sektoralen IDP weitergeleitet werden über welches der Nutzer die Authentisierung durchführt. Hierbei sind die folgenden Parameter zu verwenden:

Parameter	Wert	Anmerkung
client_id	"https://idp.app.ti-dienste.de"	Identifer des IDP-Dienstes als Relying Party innerhalb der Föderation
request_uri	urn:Fachdienst007:bwc4JK-ESC0w8acc191e-Y1LTC2	URI zur späteren Identifikation des Request durch den sektoralen IDP.

[<=]

A_23690-01 - Annahme von AUTHORIZATION_CODE

Der IDP-Dienst MUSS am Authorization-Endpunkt eingehende AUTHORIZATION_CODE annehmen und anhand des state-Parameters die in A_23686* definierten Sitzungsdaten zur weiteren Bearbeitung rekonstruieren.[<=]

5.5.2 Authentifizierung über andere IDP - Ausgangsdaten

Hinweis: Aufgrund der Speicherung und Referenzierung der Sitzungsdaten durch den state-Parameter, reduziert sich die Prüfung auf Gleichheit zwischen übermittelten und gespeicherten "state" auf die Prüfung des Vorhandenseins einer unter dem übermittelten "state" gespeicherten Sitzung.

A_23691-02 - Token Request des IDP-Dienstes an Identity Provider der Föderation

Wenn der zu einem eingereichten AUTHORIZATION_CODE_IDP korrelierende Authorization Request einen Parameter idp_iss mit dem Identifikator eines sektoralen Identity Provider enthielt, MUSS der IDP-Dienst einen Token Request gemäß gemSpec_IDP_FD#AF_10118 an den sektoralen Identity Provider stellen und sich dabei als 'Confidential Client' authentisieren.

Es sind die folgenden Inhalte zu verwenden:

Parameter	Wert	Anmerkung
grant_type	authorization_code	
code	AUTHORIZATION_CODE_IDP	Authorization_Code des sektoralen Identity Provider
code_verifier	<code_verifier des IDP-Dienstes>	aus den Sitzungsdaten
client_id	"https://idp.app.ti-dienste.de"	
redirect_uri	redirect_uri aus dem initialen GET Request des Anwendungsfrontends	Die Adresse des Authorization Endpunkt zur Annahme der Antwort vom sektoralen IDP ist die redirect_uri aus der Anfrage des Clients. Die zu der E-Rezept-App gehörige uri ist z.B. https://das-e-rezept-fuer-deutschland.de/ext/auth .
client_assertion_type	"urn:ietf:params:oauth:client-assertion-type:self_signed_tls_client_auth"	Authentisierung des confidential client entsprechend https://openid.net/specs/openid-connect-core-1_0.html#ClientAuthentication durch ein TLS Clientzertifikat

Der Timeout, mit dem ein Request zum sektoralen Identity Provider abgebrochen werden darf, wird einheitlich für alle Abfragen (und analog zur Festlegung im Kontext der OCSP Requests) auf den in A_21340-* definierten Wert festgelegt. **[<=]**

Hinweis: Aufgrund der Speicherung und Referenzierung der Sitzungsdaten durch den state-Parameter, reduziert sich die Prüfung auf Gleichheit zwischen übermittelten und gespeicherten state auf die Prüfung des Vorhandenseins einer unter dem übermittelten state gespeicherten Sitzung.

A_23692 - Authentisierung des IDP-Dienstes gegenüber Identity Providern der Föderation

Der IDP-Dienst MUSS sich gegenüber dem Token-Endpunkt des sektoralen Identity Provider entsprechend RFC8705-section 2.2

<https://www.rfc-editor.org/rfc/rfc8705.html#name-self-signed-certificate-mut> als 'confidential client' mit einem TLS Clientzertifikat authentisieren. [**<=**]

A_22269 - Produktion eines Authorization Code nach Bestätigung des sektoralen Identity Provider

Nach erfolgreicher Prüfung des ID-TOKEN MUSS der IDP-Dienst auf Basis der Sitzungsdaten und der Inhalte der Claims einen Authorization Code wie in [gemSpec_IDP#5.2.2] beschrieben produzieren und diesen zusammen mit dem gespeicherten 'State' und einemSSO_TOKEN an das Anwendungsfrontend übermitteln. [**<=**]

A_22270 - Löschung der Sitzungsdaten zum sektoralen Identity Provider

Der IDP-Dienst MUSS nach erfolgter Ausstellung des Authorization Code oder Abbruch des Prozesses die unter dem vom ihm erzeugten 'State' gespeicherten Sitzungsdaten zur Anfrage an einen sektoralen Identity Provider löschen. [**<=**]

Der restliche Flow gestaltet sich wie in den Dokumenten [gemSpec_IDP_Dienst], [gemSpec_IDP_Frontend] und [gemSpec_IDP_FD] beschrieben. Dieses schließt die Produktion eines geeigneten ACCESS_TOKEN durch den IDP-Dienst mit ein.

6 Anhang A - Verzeichnisse

6.1 Abkürzungen

Kürzel	Erläuterung
AVS	Apothekenverwaltungssystem
CAA	Certification Authority Authorization
CT	Certificate Transparency
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.
Alternatives Authentisierungsmittel	Mit einem bereits etablierten Authentisierungsmittel verknüpft Mittel zur Authentisierung gegenüber Fachdiensten.
Authentifizierung des Nutzers am Gerät oder lokale Authentifizierung	Authentifizierungsmittel des Nutzers zur Nutzung eines Kontos auf einem Mobilgerät.
Authentifizierungszertifikat	Unter einem Authentifizierungszertifikat werden Kontext der Registrierung und Verwendung von alternativen Authentisierungsmittel Zertifikate vom Typ C.CH.AUT der eGK verstanden.
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.
Autorisierte Anwendung eines Schlüssels	Anwendung eines kryptographischen Schlüssels auf Daten durch einen berechtigten Nutzer.
Betriebssystem (oder Plattform)	Der Name des Betriebssystems eines Geräts.
Besitz (eines Geräts)	Verwendungshoheit eines Nutzers über ein Mobilgerät.
Bewertung (eines	Sicherheitsbezogene Einstufung eines Gerätetyps mit Bezug

Gerätetyps)	auf seine Eignung als Authentisierungsmittel
Block/Allow-Liste(n)	Registrierung von Gerätetypen und Zuordnung zu einer Bewertung.
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.
Gerät	Alle Arten von mobilen Endgeräten.
Geräteinformationen	Die Kombination der Informationen zum Typ eines Geräts und dem Namen eines Geräts.
Hersteller	Der Name des Herstellers eines Geräts.
Inspektion	Einsicht des Nutzers in die für die Anwendung von alternativen Authentisierungsmitteln gespeicherten Daten auf seinem Gerät oder innerhalb des IDP-Dienstes.
ID Token	Ein auf JSON basiertes und nach [RFC7519] (JWT) genormtes Identitäts-Token, mit dem ein Client (Anwendungsfrontend) die Identität eines Nutzers überprüfen kann.
Key-Identifizier	Einem kryptographischen Schlüssel oder einem Schlüsselpaar zugeordnete Zeichenkette zur Identifikation des Schlüssels.

Löschung oder Invalidierung	Unter Löschung eines Schlüssels sollen pauschal alle Operationen verstanden werden, die einen kryptographischen Schlüssel dauerhaft einer Anwendung entziehen.
Modell	Der vom Hersteller vergebene Name des Geräts.
Name (eines Geräts)	Ein vom Nutzer vergebener Name eines Geräts.
Open Authorization 2.0	Ein Protokoll zur Autorisierung für Web-, Desktop und Mobile Anwendungen. Dabei wird es einem Endbenutzer (Resource Owner) ermöglicht, einer Anwendung (Client) den Zugriff auf Daten oder Dienste (Resources) zu ermöglichen, die von einem Dritten (Resource Server) bereitgestellt werden.
OpenID Connect	OpenID Connect (OIDC) ist eine Authentifizierungsschicht, die auf dem Autorisierungsframework OAuth 2.0 basiert. Es ermöglicht Clients, die Identität des Nutzers anhand der Authentifizierung durch einen Autorisierungsserver zu überprüfen (siehe [openid-connect-core 1.0]).
Pairing	Prüfbare Verbindung von kryptographischem Schlüsselmaterial zu einer innerhalb der Telematikinfrastruktur registrierten kryptographischen Identität.
Produkt-Name	Der vom Hersteller für den Endkunden vergebene Name eines Geräts.
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. Das SSO Token kommt nur beim E-Rezept FdV zum Einsatz und erfüllt hier die Funktion eines Refresh-Token.

Token-Endpunkt	Ein Endpunkt des Authorization Servers, welcher für die Ausstellung von Token (ID_TOKEN und ACCESS_TOKEN) zuständig ist.
Typ (eines Geräts)	Eine existente Kombination von Gerät und Betriebssystem beschrieben durch Namen des Herstellers, Namen des Produkts, Betriebssystem und Version des Betriebssystems.
Version (eines Betriebssystems)	Die Bezeichnung der Version des Betriebssystems eines Geräts.

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

6.3 Abbildungsverzeichnis

Abbildung 1: Übersichtsschaubild OAuth2.0 IDP-Dienst.....10

Abbildung 2: Systemkontext aus Sicht des IDP-Dienstes.....12

Abbildung 3: Datenfluss-Diagramm IDP-Dienst.....17

Abbildung 4: Schnittstellen des IDP-Dienstes.....25

Abbildung 5: Ablauf Registrierungsprozess.....54

Abbildung 6: Ablauf Verwendung alternativer Authentisierungsmittel.....60

Abbildung 7: Inspektion der Pairing-Daten.....65

Abbildung 8: Deregistrierung eines Pairings.....66

6.4 Tabellenverzeichnis

Tabelle 1: TAB_IDP_DIENST_0001 Akteure und OAuth2-Rollen.....13

Tabelle 2: TAB_IDP_DIENST_0002 Kurzbezeichnung der Schnittstellen des IDP-Dienstes. 14

Tabelle 3: TAB_IDP_DIENST_0003 Bezeichnungen der extern genutzten Schlüssel und Adressen des IDP-Dienstes..... 14

Tabelle 4: TAB_IDP_DIENST_0005 Befüllung der Attribute "given_name", "family_name", "organizationName", "professionOID", "idNummer" und "organizationIK".....38

Tabelle 5: TAB_IDP_DIENST_0006 Befüllung der Attribute "acr" und "amr".....39

Tabelle 6: TAB_IDP_DIENST_0007 Befüllung der Attribute nach Bestätigung durch einen sektoralen Identity Provider.....40

Tabelle 7: TAB_IDP_DIENST_0008 Faktoren und abgeleitete Anforderungen an das System43

Tabelle 8: TAB_IDP_DIENST_0009 Nomenklatur Schlüsselmaterial.....51

Tabelle 9: Fehlermeldungen im Zusammenhang mit der Registrierungsfunktion.....57

Tabelle 10: Fehlermeldungen des Authorization-Endpunkts bei Authentifizierung auf Basis von alternativen Authentisierungsmitteln.....62

Tabelle 11: Fehlermeldungen Inspektions- und Deregistrierungsfunktion.....66

Tabelle 12: Kodierung von Daten.....94

Tabelle 13: Schema Datentyp "Device_Type".....95

Tabelle 14: Schema Datentyp "Device_Information".....95

Tabelle 15: Schema Datentyp "Pairing_Data".....96

Tabelle 16: Schema Datentyp "Signed_Pairing_Data".....98

Tabelle 17: Schema Datentyp "Registration_Data".....99

Tabelle 18: Schema Datentyp "Encrypted_Registration_Data".....99

Tabelle 19: Schema Datentyp "Authentication_Data".....100

Tabelle 20: Schema Datentyp "Signed_Authentication_Data".....101

Tabelle 21: Schema Datentyp "Encrypted_Signed_Authentication_Data".....101

Tabelle 22: Schema Datentyp "Pairing_Entry".....102

Tabelle 23: Schema Datentyp "Pairing_Entries".....103

Tabelle 24: Registrierungsprozess/Anfrage Authenticator-Modul.....103

Tabelle 25: Registrierungsprozess/Antwort Pairing-Endpunkt.....104

Tabelle 26: Authentifizierung/Anfrage Authenticator-Modul.....105

Tabelle 27: Authentifizierung/Antwort Authorization-Endpunkt.....105

Tabelle 28: Inspektion/Anfrage Authenticator-Modul.....106

Tabelle 29: Inspektion/Antwort Pairing-Endpunkt.....106

Tabelle 30: Deregistrierung/Anfrage Authenticator-Modul.....107

Tabelle 31: Deregistrierung/Antwort Pairing-Endpunkt.....108

Tabelle 32: Vergleichsoperationen im Rahmen der Registrierung.....108

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.

[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
[RFC4648]	The Base16, Base32, and Base64 Data Encodings https://tools.ietf.org/html/rfc4648
[RFC5480]	Elliptic Curve Cryptography Subject Public Key Information https://tools.ietf.org/html/rfc5480
[RFC5280]	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile https://tools.ietf.org/html/rfc5280
[RFC7516]	JSON Web Encryption (JWE) https://tools.ietf.org/html/rfc7516
[RFC8176]	Authentication Method Reference Values https://tools.ietf.org/html/rfc8176
[openid-connect-modrna]	OpenID Connect MODRNa Authentication Profile 1.0 https://openid.net/specs/openid-connect-modrna-authentication-1_0.html
[RFC4648]	The Base16, Base32, and Base64 Data Encodings https://tools.ietf.org/html/rfc4648
[RFC5480]	Elliptic Curve Cryptography Subject Public Key Information https://tools.ietf.org/html/rfc5480
[RFC5280]	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile https://tools.ietf.org/html/rfc5280

7 Anhang B - Beispiele der Objekte und Aufrufe

Folgendes Kapitel verdeutlicht beispielhaft den Aufbau der einzelnen Objekte und Aufrufe, welche im Rahmen der Beantragung von ACCESS_TOKEN und ID_TOKEN beim IDP-Dienst generiert und ausgetauscht werden.

Die einzelnen Aufrufe werden den Schritten aus Kapitel [3.3 Verfahrensbeschreibung] zugewiesen und in entsprechenden Unterkapiteln dargestellt.

Diese Beispiele ersetzen die vorher an verschiedenen Stellen des Dokumentes verteilten Darstellungen.

Die gematik wird weitere Beispielsätze auf ihrer Webseite in jeweils aktueller Form bereitstellen.

7.1 Authorization Request

2. Das Anwendungsfrontend überträgt seinen Authorization Request inklusive der generierten Werte CODE_CHALLENGE, State und Nonce gemäß [RFC8252#Anhang B] an das Authenticator-Modul.

3. Das Authenticator-Modul überträgt den Authorization Request weiter an den Authorization-Endpunkt des IDP-Dienstes.

/sign_response?

&client_id=eRezeptApp

Die Client-ID des Primärsystems wird beim Registrieren des Primärsystems beim IDP festgelegt.

&state=AcYxMQ5MZMpRh6WOBjs8

Dieser Parameter wird vom Client zufällig generiert, um CSRF zu verhindern. Indem der Server mit diesem Wert antwortet, werden Redirects legitimiert.

&redirect_uri=http%3A%2F%2Fredirect.gematik.de%2Ferezept

Die URL wird vom Primärsystem beim Registrierungsprozess im IDP hinterlegt und leitet die Antwort des Servers an diese Adresse um.

&code_challenge=SU8xsVcUypYGUI2g-mzs7rvR2IMtQ9vyj_9Hxs0WcII

Der Hashwert des Code-Verifiers wird zum IDP als Code-Challenge gesendet. Teil von PKCE.

&code_challenge_method=S256

Das Primärsystem generiert einen Code-Verifier und erzeugt darüber einen Hash im Verfahren SHA-256, hier abgekürzt als S256. Teil von PKCE.

&response_type=code

Referenziert den erwarteten Response-Type des Flows. Muss immer 'code' lauten. Damit wird angezeigt, dass es sich hierbei um einen Authorization Code Flow handelt. Für eine nähere Erläuterung siehe OpenID-Spezifikation.

&nonce=nN4LkW1moAwg1tofYZtf

String zur Verhinderung von CSRF-Attacken. Dieser Wert ist optional. Wenn er mitgegeben wird, muss der gleiche Wert im abschließend ausgegebenen ID_TOKEN wiederauftauchen.

scope=openid+e-rezept

Der Scope entspricht dem zwischen E-Rezept-Fachdienst und IDP festgelegten Wert. Mit diesem antwortet der E-Rezept-Fachdienst bei fehlendem ACCESS_TOKEN und http-Statuscode 401.

&idp_iss=kk_idp_4711

Nur für Requests an den 3rd Party Authorization_Endpoint des IDP-Dienstes durch das E-Rezept-FdV. Eindeutiger interner Identifier des anzufragenden sektoralen Identity Provider, welcher für die Authentisierung des Nutzers gewählt wurde.

7.2 Authorization Response

7. Der Authorization-Endpoint überträgt CHALLENGE und Consent-Abfrage USER_CONSENT zum Authenticator-Modul.

200

Cache-Control=no-store,

Pragma=no-cache,

Version=0.1-SNAPSHOT,

Content-Type=application/json,

Transfer-Encoding=chunked,

Date=<Zeitpunkt der Antwort. Beispiel Fri, 05 Feb 2021 12:46:20 GMT>

Keep-Alive=timeout=60,

Connection=keep-alive

Response-Body:

```
{
  "challenge": "eyJhbGciOiJC...",
  "user_consent": {
    "requested_scopes": {
      "e-rezept": "Zugriff auf die E-Rezept-Funktionalität.",
      "openid": "Zugriff auf den ID_TOKEN."
    },
    "requested_claims": {
      "organizationName": "Zustimmung zur Verarbeitung der Organisationszugehörigkeit",
      "professionOID": "Zustimmung zur Verarbeitung der Rolle",
      "idNummer": "Zustimmung zur Verarbeitung der ID (z.B. Krankenversichertennummer, Telematik-ID)",
    }
  }
}
```

```
"given_name": "Zustimmung zur Verarbeitung des Vornamens",
"family_name": "Zustimmung zur Verarbeitung des Nachnamens",
"display_name": "Zustimmung zur Verarbeitung des Anzeigenamens"
}
}
}
```

CHALLENGE_TOKEN:

```
{
  "alg": "BP256R1",
  "typ": "JWT",
  "kid": "puk_idp_sig"
}
{
  "iss": "http://url.des.idp",
  "response_type": "code",
  "snc": "[server-nonce. Wird verwendet, um noise hinzuzufügen.
Beispiel: \"yvcVQO09TFsFfaj312RfYoi1oII0BHtIDIRpzD8Gu0\"]",
  "code_challenge_method": "S256",
  "token_type": "challenge",
  "nonce": "nN4LkW1moAwg1tofYZtf",
  "client_id": "eRezeptApp",
  "scope": "openid e-rezept",
  "state": "AcYxMQ5MZMpRh6WOBjs8",
  "redirect_uri": "http://redirect.gematik.de/erezept",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244172]",
  "iat": "[Zeitpunkt der Ausstellung des Token. Beispiel: 1618243992]",
  "code_challenge": "SU8xsVcUypYGUI2g-mzs7rvR2IMtQ9vyj_9Hxs0WcII",
  "jti": "[A unique identifier for the token, which can be used to prevent reuse of the
token. Value is a case-sensitive string. Beispiel: \"07925bdc7c5d9990\"]"
}
```

7.3 Authentication Request

10. Das Authenticator-Modul überträgt signierte CHALLENGE mit dem Smartcard-Zertifikat an den IDP-Dienst (Antwort Schritt 7).

`https://<FQDN Server>/<AUTHORIZATION_ENDPOINT>`

Multiparts:

`signed_challenge=<signierter und anschließend verschlüsselter Challenge Token>`

Challenge Response (Encryption Header):

```
{
  "alg": "ECDH-ES",
  "enc": "A256GCM",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244172]",
  "cty": "NJWT",
  "epk": {
    "kty": "EC",
    "x": "LgkJSQwrz1bCoFjSLhay9O7TLaQImYW7jeOF6XmpQX4",
    "y": "dTC6ri-f1QqpJp7M4LLg0lw4FzrzNc29nrrzjPwEWWc",
    "crv": "BP-256"
  }
}
{
  "njwt": "[Ein verschachtelt enthaltenes JWT] - Die verschlüsselte Challenge Response."
}
```

Challenge Response (Decrypted):

```
{
  "typ": "JWT",
  "cty": "NJWT",
  "alg": "BP256R1",
  "x5c": [
    "[Enthält das verwendete Signer-Zertifikat als Base64 ASN.1 DER-Encoding. Hier
    kommt ausnahmsweise NICHT URL-safes Base64-Encoding zum Einsatz!]"
  ]
}
{
  "njwt": "[Ein verschachtelt enthaltenes JWT] - Dieses Token ist die vom Server in der
  vorigen Nachricht übergebene Challenge. Sie muss exakt wie empfangen auch wieder
  übertragen werden. "
}
```

7.3.1 Authentication Request (SSO)

10. Falls ein SSO_TOKEN beim Authenticator-Modul existiert, wird dieser Token zusammen mit einem unveränderten CHALLENGE_TOKEN zum IDP-Dienst transportiert.

https://<FQDN Server>/<SSO_ENDPOINT>

Multiparts:

sso_token=<SSO Token des IDP>
unsigned_challenge = < unveränderter Challenge_Token>

7.4 Authentication Response

14. Der Authorization-Endpoint überträgt den AUTHORIZATION_CODE und den SSO_TOKEN an das Authenticator-Modul (Antwort Schritt 3).

302

Cache-Control=no-store,

Pragma=no-cache,

Location=http://redirect.gematik.de/erezept?

code=<Authorization Code in Base64-URL-Safe Encoding. Wird unten detaillierter aufgeführt> Der Authorization-Code. Er berechtigt zur Abholung einesACCESS_TOKEN. Er ist vom IDP für den IDP verschlüsselt und dementsprechend vom Client nicht weiter zu verarbeiten.

&state=AcYxMQ5MZMpRh6WOBjs8 Der state der Session. Sollte dem zufällig generierten state-Wert aus der initialen Anfrage entsprechen.

&sstoken=<SSO Token in Base64-URL-Safe Encoding. Wird unten detaillierter aufgeführt> Der SSO Token. Mit diesem kann der Client sich wiederholt einloggen, ohne erneut den Besitz der Karte durch Unterschreiben einer Challenge beweisen zu müssen. Er ist vom IDP für den IDP verschlüsselt und dementsprechend vom Client nicht weiter zu verarbeiten. - Nicht im Fall der Anmeldung über ein Primärsystem.

Content-Length=0,

Date=<Zeitpunkt der Antwort. Beispiel Fri, 05 Feb 2021 12:46:20 GMT>

Keep-Alive=timeout=60, Connection=keep-alive

Response-Body:

Die Inhalte von Authorization Code und SSO_TOKEN sind IDP-spezifisch und nicht normativ vorgegeben. Sie dienen hier nur dem Verständnis.

Authorization Code (Encryption Header):

```
{  "alg": "dir",
  "enc": "A256GCM",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244053]",
  "cty": "NJWT"
}
{
  "njwt": "[Ein verschachtelt enthaltenes JWT] - Der Authorization Code"
}
```

Authorization Code (Decrypted):

```
{
  "alg": "BP256R1",
  "typ": "JWT",
  "kid": "puk_idp_sig"
}
{
  "organizationName": "AOK Plus",
  "professionOID": "1.2.276.0.76.4.49",
  "idNummer": "X114428530",
  "iss": "http://url.des.idp",
  "response_type": "code",
  "snc": "[server-nonce. Wird verwendet, um noise hinzuzufügen.
  Beispiel: \"Ay6WUqtAUcV2p9WZYHPo\"]",
  "code_challenge_method": "S256",
  "given_name": "Juna",
  "display_name": "Juna Fuchs",
  "token_type": "code",
  "nonce": "nN4LkW1moAwg1tofYZtf",
  "client_id": "eRezeptApp",
  "scope": "openid e-rezept",
  "auth_time": "[Timestamp der Authentisierung. Beispiel: 1618243993]",
  "redirect_uri": "http://redirect.gematik.de/erezept",
  "state": "AcYxMQ5MZMpRh6WOBjs8",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244053]",
  "family_name": "Fuchs",
  "iat": "[Zeitpunkt der Ausstellung des Token. Beispiel: 1618243993]",
  "code_challenge": "SU8xsVcUypYGUI2g-mzs7rvR2IMtQ9vyj_9Hxs0WcII",
  "jti": "[A unique identifier for the token, which can be used to prevent reuse of the
  token. Value is a case-sensitive string. Beispiel: \"6e8a61e316472f3b\"]"
}
```

SSO Token (Encryption Header):

```
{
  "alg": "dir",
  "enc": "A256GCM",
  "exp": "[Gültigkeit des Token. Beispiel: 1618287193]",
  "cty": "NJWT"
}
```

```
{
  "njwt": "[Ein verschachtelt enthaltenes JWT] - Das SSO Token"
}
```

SSO Token (Decrypted):

```
{
  "alg": "BP256R1",
  "typ": "JWT",
  "kid": "puk_idp_sig"
}
{
  "organizationName": "AOK Plus",
  "professionOID": "1.2.276.0.76.4.49",
  "auth_time": "[Timestamp der Authentisierung. Beispiel: 1618243993]",
  "idNummer": "X114428530",
  "iss": "http://url.des.idp",
  "cnf": {
    "x5c": [
      "[Enthält das verwendete Signer-Zertifikat als Base64 ASN.1 DER-Encoding. Hier
      kommt ausnahmsweise NICHT URL-safes Base64-Encoding zum Einsatz!]"
    ],
    "kid": "844508318621525",
    "kty": "EC",
    "crv": "BP-256",
    "x": "dTXa6yPKCjlr9MbVFxeaLEu82xSCsRrfwclrLpFqBCs",
    "y": "AJGsJ1cCyGEpCH0ss8JvD4OAHJS8IMm1_rM59jliS-10"  },
    "given_name": "Juna",
    "display_name": "Juna Fuchs",
    "exp": "[Gültigkeit des Token. Beispiel: 1618287193]",
    "iat": "[Zeitpunkt der Ausstellung des Token. Beispiel: 1618243993]",
    "family_name": "Fuchs"
  }
}
```

7.4.1 Authentication Response (SSO Flow)

Falls 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.

302

Cache-Control=no-store,
Pragma=no-cache,
Location=https://<FQDN Server>/<TOKEN_ENDPOINT>
 ?code=<Authorization Code des IDP>
 &state=<OAuth 2.0 state value. Constant over complete flow. Value is a case-sensitive string. Beispiel: 'mLE7xX1ysbZQ4Of5YiZ8'>,
Content-Length=0,
Date=<Zeitpunkt der Antwort. Beispiel Fri, 05 Feb 2021 12:46:20 GMT>
Keep-Alive=timeout=60,
Connection=keep-alive

7.5 Token Request

16. Das Anwendungsfrontend sendet CODE_VERIFIER und AUTHORIZATION_CODE zum Token-Endpunkt des IDP-Dienstes.

https://<FQDN Server>/<TOKEN_ENDPOINT>
Multipart:
client_id=eRezeptApp
&code=<Authorization Code des IDP>
&grant_type=authorization_code
&key_verifier=<verschlüsselter KEY_VERIFIER>
&redirect_uri=http%3A%2F%2Fredirect.gematik.de%2Ferezept

Key_verifier (Encryption Header):

```
{  
  "alg": "ECDH-ES",  
  "enc": "A256GCM",  
  "cty": "JSON",  
  "epk": {  
    "kty": "EC",  
    "x": "jqrGQIZqCGQgK7OtJj0gfWPWSbStracf_PreBrA05Lc",  
    "y": "V4bbOGUgr-AV6NFLnPJNYkyPLcR_1QkGIR6w4bK1wdl",  
    "crv": "BP-256"  } }  
}
```

Key_verifier (Body)

```
{  
  "token_key": "T0hHOHNKOTFaREcxTmN0dVRKSURraTZxNEpheGxaUEs",  
  "code_verifier": "W91A37hQ8oeDRVpknYgpYthjl4LqYy95A87ISy9zpUM"  
}
```

```
}
```

7.6 Token Response

20. Der Token-Endpoint überträgt die Token an das Anwendungsfrontend (Antwort Schritt 16).

200

Cache-Control=no-store,

Pragma=no-cache,

Version=0.1-SNAPSHOT,

Content-Type=application/json,

Transfer-Encoding=chunked,

Date=<Zeitpunkt der Antwort. Beispiel Fri, 05 Feb 2021 12:46:20 GMT>

Keep-Alive=timeout=60,

Connection=keep-alive

Response-Body:

```
{
  "expires_in": 300,
  "token_type": "Bearer",
  "id_token": <Mit dem Token_Key verschlüsseltes ID_TOKEN.>
  "access_token": <Mit dem Token_Key verschlüsseltes ACCESS_TOKEN.>}

```

Access Token (Encryption Header):

```
{ "alg": "dir",
  "enc": "A256GCM",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244294]",
  "cty": "NJWT"
}
{
  "njwt": "[Ein verschachtelt enthaltenes JWT] - Das ACCESS_TOKEN"
}

```

Access Token (Decrypted):

```
{
  "alg": "BP256R1",
  "typ": "at+JWT",
  "kid": "puk_idp_sig"
}

```

```
}
{
  "sub": "[subject. Base64(sha256(audClaim + idNummerClaim + serverSubjectSalt)).
Beispiel: \"ez4D403gBzH1IhnYoxA4aUU-7spqPbWUyUeLPoA79CM\"]",
  "professionOID": "1.2.276.0.76.4.49",
  "organizationName": "AOK Plus",
  "idNummer": "X114428530",
  "amr": [
    "mfa",
    "sc",
    "pin" ],
  "iss": "http://url.des.idp",
  "given_name": "Juna",
  "display_name": "Juna Fuchs",
  "client_id": "eRezeptApp",
  "acr": "gematik-ehealth-loa-high",
  "aud": "https://erp-test.zentral.erp.splitdns.ti-dienste.de/",
  "azp": "eRezeptApp",
  "scope": "openid e-rezept",
  "auth_time": "[Timestamp der Authentisierung. Beispiel: 1618243993]",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244294]",
  "family_name": "Fuchs",
  "iat": "[Zeitpunkt der Ausstellung des Token. Beispiel: 1618243994]",
  "jti": "[A unique identifier for the token, which can be used to prevent reuse of the
token. Value is a case-sensitive string. Beispiel: \"c0bf3cebe428e3c9\"]"
}
```

ID Token (Encryption Header):

```
{
  "alg": "dir",
  "enc": "A256GCM",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244294]",
  "cty": "NJWT"
}
{
  "njwt": "[Ein verschachtelt enthaltenes JWT] - Das ID Token"
}
```

ID Token (Decrypted):

```

{
  "alg": "BP256R1",
  "typ": "JWT",
  "kid": "puk_idp_sig"
}
{
  "at_hash": "[Erste 16 Bytes des Hash des Authentication Token
Base64(subarray(Sha256(authentication_token), 0, 16)). Beispiel: \"5AZmDxrYImUa6-
kjMNAL3g\"]",
  "sub": "[subject. Base64(sha256(audClaim + idNummerClaim + serverSubjectSalt)).
Beispiel: \"ez4D403gBzH1IhnYoxA4aUU-7spqPbWUyUeLPoA79CM\"]",
  "organizationName": "AOK Plus",
  "professionOID": "1.2.276.0.76.4.49",
  "idNummer": "X114428530",
  "amr": [
    "mfa",
    "sc",
    "pin" ],
  "iss": "http://url.des.idp",
  "given_name": "Juna",
  "display_name": "Juna Fuchs",
  "nonce": "nN4LkW1moAwg1tofYZtf",
  "aud": "eRezeptApp",
  "acr": "gematik-ehealth-loa-high",
  "azp": "eRezeptApp",
  "auth_time": "[Timestamp der Authentisierung. Beispiel: 1618243993]",
  "scope": "openid e-rezept",
  "exp": "[Gültigkeit des Token. Beispiel: 1618244294]",
  "iat": "[Zeitpunkt der Ausstellung des Token. Beispiel: 1618243994]",
  "family_name": "Fuchs",
  "jti": "[A unique identifier for the token, which can be used to prevent reuse of the
token. Value is a case-sensitive string. Beispiel: \"c1c760ca67fe1306\"]"
}

```

7.7 Aufbau des Discovery Document

```

{
  "alg": "BP256R1",

```



```
"kid": "puk_disc_sig",
"x5c": [
  "[Enthält das verwendete Signer-Zertifikat als Base64 ASN.1 DER-Encoding. Hier
  kommt ausnahmsweise NICHT URL-safes Base64-Encoding zum Einsatz!]"
]
}
{
  "authorization_endpoint": "[URL des Authorization Endpunkts.]",
  "federation_authorization_endpoint": "[URL des Authorization Endpunkt für Anfragen an
  IDP der Föderation - dieser ist nur im Internet verfügbar
  ]",
  "auth_pair_endpoint": "[URL des Pairing-Authorization-Endpunkts - dieser ist nur im
  Internet verfügbar]"
  "third_party_authorization_endpoint" : "[URL des third_party-Authorization-Endpunkts -
  dieser ist nur im Internet verfügbar]"
  "sso_endpoint": "[URL des SSO-Authorization Endpunkts.]",
  "uri_pair": "[URL des Pairing-Endpunkts.]",
  "token_endpoint": "[URL des Authorization-Endpunkts.]",
  "uri_disc": "[URL des Discovery Document.]",
  "issuer": "http://url.des.idp",
  "jwks_uri": "[URL einer JWKS-Struktur mit allen vom Server verwendeten Schlüsseln]",
  "exp": "[Gültigkeit des Token. Beispiel: 1618330390]",
  "iat": "[Zeitpunkt der Ausstellung des Token. Beispiel: 1618243990]",
  "uri_puk_idp_enc": "http://url.des.idp/idpEnc/jwk.json",
  "uri_puk_idp_sig": "http://url.des.idp/idpSig/jwk.json",
  "subject_types_supported": [
    "pairwise"
  ],
  "id_token_signing_alg_values_supported": [
    "BP256R1"
  ],
  "response_types_supported": [
    "code"
  ],
  "scopes_supported": [
    "openid",
    "e-rezept"
    "pairing"
  ],
}
```

```
"response_modes_supported": [  
  "query"  
],  
"grant_types_supported": [  
  "authorization_code"  
],  
"acr_values_supported": [  
  "gematik-ehealth-loa-high"  
],  
"token_endpoint_auth_methods_supported": [  
  "none"  
],  
"code_challenge_methods_supported": [  
  "S256"  
]  
}
```

8 Anhang C - Datenformate und Kommunikationsprotokolle im Zusammenhang mit alternativen Authentisierungsverfahren

8.1 Datentypen

Die Verwendung der hier genannten Datentypen und ihre Ausgestaltung sind verpflichtend. Gleiches gilt für die dargestellte Detaillierung der Kommunikationsprotokolle.

8.1.1 Übergeordnete Anforderungen

Die folgenden Festlegungen gelten übergeordnet für alle verwendeten Datentypen.

8.1.1.1 Kodierung

Tabelle 12: Kodierung von Daten

Datentyp	Kodierung	Repräsentation in JSON
String	UTF-8	JSON/String, vergleiche [RFC7519] , Abschnitt 7 "Strings"
Binärdaten	base64url, ohne Padding	JSON/String, vergleiche [RFC4648] , Abschnitt 5, "Base 64 Encoding with URL and Filename Safe Alphabet" , [RFC7515] , Anhang C, "Notes on Implementing base64url Encoding without Padding"
Zeitangabe n	NumericDate, vergleiche [RFC7519] , Abschnitt 2 "Terminology"	JSON/Number, vergleiche [RFC7519] , Abschnitt 6 "Numbers"

8.1.1.2 Versionierung

Die in den folgenden Abschnitten genannten Datentypen enthalten eine Name-/Wert-Kombination zur Anzeige der Version der Datenobjekte. Diese ist "<Name des Datenobjektes>_version". Der Wert ist konstant mit "1.0" zu belegen.

8.1.1.3 Namen und Claims

Namen von JSON-Elementen sind ausnahmslos in sog. "Snake-Case" dargestellt. Namen von Datenobjekten in Großschreibung.

8.1.2 Datentyp "Device_Type"

Der Datentyp "Device_Type" wird zur Übertragung von Informationen über einen Gerätetyp vom Authenticator-Modul zum IDP-Dienst verwendet. Der Datensatz wird vom Authenticator-Modul produziert und vom Pairing-Endpunkt und dem Authorization-Endpunkt verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 13: Schema Datentyp "Device_Type"

Datenformat: JSON			
Name	Verpflichtend	Type	Hinweise
device_type_data_version	ja	JSON/String, Konstant "1.1"	-
manufacturer	ja	JSON/String	Name des Herstellers eines Geräts
product	ja	JSON/String	Produktname des Geräts gegenüber dem Endkunden
model	ja	JSON/String	Name des Modells
os	ja	JSON/String	Betriebssystem
os_version	ja	JSON/String	Version des Betriebssystems
security_patchlevel	nein	JSON/String	Format "YYYY-MM-DD"

8.1.3 Datentyp "Device_Information"

Der Datentyp "Device_Information" wird zur Übertragung von Informationen über ein Gerät verwendet. Der Datensatz wird vom Authenticator-Modul produziert und vom Pairing-Endpunkt verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 14: Schema Datentyp "Device_Information"

Datenformat: JSON			
Name	Verpflichtend	Type	Hinweise
device_information_data_version	ja	JSON/String, Konstant "1.0"	-
name	ja	JSON/String	vom Benutzer

			vergebener Name für das Gerät
device_type	ja	JSON/Object Device_Type	siehe [8.1.2- Datentyp "Device_Type"]

8.1.4 Datentyp "Pairing_Data"

Der Datentyp "Pairing_Data" wird zur Bindung des PuK_SE_AUT an Metadaten verwendet. Der Datentyp wird vom Authenticator-Modul produziert und von Pairing-Endpoint und Authorization-Endpoint verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 15: Schema Datentyp "Pairing_Data"

Datenformat: JSON			
Name	Verpflichtend	Type	Hinweise
pairing_data_version	ja	JSON/String, Konstant "1.0"	Version der Pairing-Daten-Instanz
se_subject_public_key_info	ja	JSON/String	base64url-codierte, DER-codierte ASN.1-Repräsentation des öffentlichen Schlüssels aus dem Secure-Elements und des verwendeten Algorithmus in Form einer „SubjectPublicKeyInfo“-Struktur, gemäß [RFC5480], Abschnitt 2 "Subject Public Key Information Fields" . Der öffentliche Schlüssel muss ohne die Verwendung von Punktkompression abgebildet werden (vergleiche [RFC5480], Abschnitt 2.2 "Subject Public Key").
key_identifier	ja	JSON/String	32 Byte langer Key-Identifizier des Schlüsselpaars PrK_SE_AUT/PuK_SE_AUT in base64url-Kodierung
product	ja	JSON/String	Produktname des Gerätetyps gegenüber

			dem Endkunden
serialnumber	ja	JSON/String	Ergebnis der Konvertierung der Seriennummer des C.CH.AUT von Oktettstring in den binären Zertifikatsdaten nach Integer (Funktion "OS2I" gemäß [gemSpec_COS], A_13587) und anschließender Repräsentation als String
issuer	ja	JSON/String	base64url-kodierte, DER-kodierte RDNSequence (vergleiche RFC5280 , Abschnitt 4.1.2.4 "Issuer") des Issuer-Felds aus dem Authentifizierungszertifikat C.CH.AUT
not_after	ja	JSON/String	Das Ende des Gültigkeitszeitraums des Authentifizierungszertifikats C.CH.AUT. Repräsentiert als NumericDate (vergleiche RFC7519 , Abschnitt 2 "Terminology")
auth_cert_subject_public_key_info	ja	JSON/String	base64url-codierte, DER-codierte ASN.1-Repräsentation des öffentlichen Schlüssels aus dem Authentifizierungszertifikat und des zu verwendenden Algorithmus in Form einer „SubjectPublicKeyInfo“-Struktur: <ul style="list-style-type: none"> Für elliptische Kurven RFC5480, Abschnitt 2 "Subject Public Key Information Fields" Der öffentliche Schlüssel muss ohne die Verwendung von Punktkompression abgebildet werden (vergleiche

			<p>[RFC5480], Abschnitt 2.2 "Subject Public Key").</p> <ul style="list-style-type: none"> Im Fall von RSA-Signaturen gemäß [RFC5280], Abschnitt 4.1.2.7 "Subject Public Key Info" und [RFC5280], Abschnitt 4.1 "Basic Certificate Fields".
--	--	--	---

8.1.5 Datentyp "Signed_Pairing_Data"

Der Datentyp "Signed_Pairing_Data" repräsentiert den vom Nutzer mithilfe der eGK signierte Instanz des Datentyps "Pairing_Data". Der Datentyp wird vom Authenticator-Modul produziert und von Pairing-Endpunkt und Authorization-Endpunkt verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 16: Schema Datentyp "Signed_Pairing_Data"

Datenformat: JWS, kompakte Serialisierung gemäß [RFC7515]			
Header			
Name	Verpflichtend	Type	Hinweise
alg	ja	JSON/String, Konstant "BP256R1"	-
typ	ja	JSON/String, Konstant "JWT"	-
Payload			
Pairing-Daten-Instanz	ja	JSON/Object - Pairing-Data	siehe [8.1.4-Datentyp "Pairing_Data"]

8.1.6 Datentyp "Registration_Data"

Der Datentyp "Registration_Data" bündelt die zur Verifikation und Anlage eines Pairing benötigten Daten. Der Datentyp wird vom Authenticator-Modul produziert und vom Pairing-Endpunkt verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 17: Schema Datentyp "Registration_Data"

Datenformat: JSON

Name	Verpflichtend	Type	Hinweise
registration_data_version	ja	JSON/String, Konstant "1.0"	-
signed_pairing_data	ja	JSON/String	Repräsentation der Signed_Pairing_Data-Instanz in kompakter Serialisierung, vergleiche [RFC7515] .
auth_cert	ja	JSON/String	DER-Kodierung des X.509 Authentifizierungszertifikat der eGK in base64url-Kodierung.
device_information	ja	JSON/Object "Device_Information"	siehe [8.1.3- Datentyp "Device_Information"]

8.1.7 Datentyp "Encrypted_Registration_Data"

Der Datentyp "Encrypted_Registration_Data" repräsentiert die verschlüsselte Form einer "Registration_Data"-Instanz. Der Datentyp wird vom Authenticator-Modul produziert und vom Pairing-Endpunkt verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 18: Schema Datentyp "Encrypted_Registration_Data"

Datenformat: JWE, Compact Serialization gemäß [RFC7516]			
Header			
Name	Verpflichtend	Typ	Hinweise
alg	ja	JSON/String, Konstant "ECDH-ES"	vergleiche [RFC7516 JSON Web Encryption (JWE)]
enc	ja	JSON/String, Konstant "A256GCM"	
typ	ja	JSON/String, Konstant "JWT"	
cty	ja	JSON/String, Konstant "JSON"	
epk	ja	JSON/Object,	
Encrypted Key, Initialization Vector, Ciphertext und Authentication Tag wie in [RFC7516, "JSON Web Encryption (JWE)"]			

8.1.8 Datentyp "Authentication_Data"

Der Datentyp "Authentication_Data" repräsentiert die zur Authentifizierung des Nutzers verwendeten Daten. Der Datentyp wird vom Authenticator-Modul produziert und vom Authorization-Endpunkt verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 19: Schema Datentyp "Authentication_Data"

Datenformat: JSON			
Name	Verpflichtend	Type/Wert	Hinweise
authentication_data_version	ja	JSON/String, Konstant "1.0"	-
challenge_token	ja	JSON/String	Repräsentation des ChallengeToken in kompakter Serialisierung, vergleiche [RFC7515]
auth_cert	ja	JSON/String	DER-Kodierung des X.509 Authentifizierungszertifikat der eGK in base64url-Kodierung
key_identifizier	ja	JSON/String	32 bytes in base64url-Codierung
device_information	ja	JSON/Object "Device_Information"	Siehe [8.1.3- Datentyp "Device_Information"]
amr	ja	JSON/Array von JSON/Strings	Angaben zu der vom Nutzer verwendeten Authentisierungsmethode zur Anwendung des Schlüssels PrK_SE_AUT als Array von JSON/String-Werten, vergleiche die Darstellungen in [RFC8176]

8.1.9 Datentyp "Signed_Authentication_Data"

Der Datentyp "Signed_Authentication_Data" repräsentiert die vom Nutzer mithilfe des PrK_SE_AUT signierte Instanz einer "Authentication_Data"-Struktur. Der Datentyp wird

vom Authenticator-Modul produziert und vom Authorization-Endpoint verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 20: Schema Datentyp "Signed_Authentication_Data"

Datenformat: JWS, kompakte Serialisierung gemäß [RFC7515]			
Header			
Name	Verpflichtend	Typ/Wert	Hinweise
alg	ja	JSON/String, Konstant "ES256"	-
typ	ja	JSON/String, Konstant "JWT"	-
Payload			
Authentication-Data-Instanz	ja	JSON/Object Authentication_Data	Siehe [8.1.3- Datentyp "Device_Information"]

8.1.10 Datentyp "Encrypted_Signed_Authentication_Data"

Der Datentyp "Encrypted_Signed_Authentication_Data" repräsentiert eine verschlüsselte "Signed_Authentication_Data"-Instanz. Der Datentyp wird vom Authenticator-Modul produziert und vom Authorization-Endpoint verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 21: Schema Datentyp "Encrypted_Signed_Authentication_Data"

Datenformat: JWE, Compact Serialization gemäß [RFC7516]			
Header			
Name	Verpflichtend	Typ/Wert	Hinweise
alg	ja	JSON/String, Konstant "ECDH-ES"	siehe [RFC7516]
enc	ja	JSON/String, Konstant "A256GCM"	
typ	ja	JSON/String, Konstant "JWT"	
cty	ja	JSON/String, Konstant "JWT"	

epk	ja	JSON/Object,	
exp	ja	JSON/Number	Zeitpunkt des Ablaufs der Gültigkeit der vorhandenen Claims, repräsentiert als NumericDate (vergleiche [RFC7519] , Abschnitt 2 "Terminology")
Encrypted Key, Initialization Vector, Ciphertext und Authentication Tag wie in [RFC7516]			

8.1.11 Datentyp "Pairing_Entry"

Der Datentyp "Pairing_Entry" repräsentiert die am Pairing-Endpoint gespeicherten Daten. Der Datentyp wird vom Pairing-Endpoint produziert und vom Authenticator-Modul verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 22: Schema Datentyp "Pairing_Entry"

Datenformat: JSON			
Name	Verpflichtend	Type/Wert	Hinweise
pairing_entry_data_version	ja	JSON/String, Konstant "1.0"	-
name	ja	JSON/String,	vom Nutzer vergebener Name des Geräts
creation_time	ja	JSON/Number	Zeitpunkt der Anlage des Pairing als Numeric-Date
signed_pairing_data	ja	JSON/String	Repräsentation der Signed_Pairing_Data-Instanz in kompakter Serialisierung, vergleiche [RFC7515]

8.1.12 Datentyp "Pairing_Entries"

Der Datentyp "Pairing_Entries" repräsentiert eine Liste von "Pairing_Entry"-Instanzen. Der Datentyp wird vom Pairing-Endpoint produziert und vom Authenticator-Modul verwendet. Der Datentyp umfasst die Elemente des folgenden Schemas:

Tabelle 23: Schema Datentyp "Pairing_Entries"

Datenformat: JSON			
Name	Verpflichten	Typ/Wert	Hinweise

	d		
pairing_entries	ja	JSON/Array von Pairing_Entry-Instanzen	siehe [8.1.11- Datentyp "Pairing_Entry"] Falls keine Pairing-Einträge des Nutzers existieren, ist dieser Wert mit einem leeren Array zu belegen.

8.2 Ausgestaltung der Kommunikation mit dem IDP-Dienst

8.2.1 Registrierung von alternativen Authentisierungsmitteln

8.2.1.1 Request des Authenticator-Moduls bei Registrierung

Tabelle 24: Registrierungsprozess/Anfrage Authenticator-Modul

Request		
HTTP-Method	URL	Hinweise
Post	<URL des Pairing-Endpunkts aus dem Discovery Document>	vergleiche Attribut "uri_pair" in A_21429-02.
Header		
Feld	Wert	Hinweise
Content-Type	application/x-www-form-urlencoded; charset=UTF-8	-
Authorization	"Bearer" <ACCESS_TOKEN>	vergleiche [RFC6750] , Abschnitt 2.1 "Authorization Request Header Field"
Accept	application/json; charset=UTF-8	-
User-Agent	<User-Agent>	siehe [gemSpec_IDP_Dienst], Abschnitt 4.4
Body		
Form-Key	Wert	Hinweise
encrypted_registration_data	serialisierte Encrypted_Registration_Data-	vergleiche [RFC7516] , Abschnitt 3.1 "JWE"

	Instanz	Compact Serialization Overview
--	---------	--

8.2.1.2 Response des Pairing-Endpunkts bei Registrierung

Tabelle 25: Registrierungsprozess/Antwort Pairing-Endpunkt

Response			
HTTP-Response Status-Code	Body	Bedeutung (Vgl. A_21411)	Hinweise
200	-	Das Pairing wurde erfolgreich angelegt.	-
403	siehe Hinweise	REG.1: Der Zugriff auf den Dienst kann nicht gewährt werden. Das ACCESS_TOKEN konnte nicht erfolgreich validiert werden.	Im Hinblick auf die syntaktische Ausgestaltung der Fehlermeldungen gelten die Anforderungen aus [4.2... Fehlermeldungen]
400		REG.2: Das verwendete Gerät ist nicht für die Authentifizierung geeignet.	
500		REG.3 Der erzeugte Schlüssel konnte aufgrund eines internen Fehlers nicht registriert werden.	
409		REG.4 Der erzeugte Schlüssel konnte aufgrund eines bestehenden Eintrags nicht registriert werden.	

8.2.2 Verwendung von alternativen Authentisierungsmitteln

8.2.2.1 Request des Authenticator-Moduls bei Verwendung von alternativen Authentisierungsmitteln

Tabelle 26: Authentifizierung/Anfrage Authenticator-Modul

Request		
HTTP-Method	URL	Hinweise
Post	<URL des Authorization-Endpunkts für alternative Authentisierung>	vergleiche Attribut "auth_pair_endpoint" in A_21429-02.
Header		

Feld	Wert	Hinweise
Content-Type	application/x-www-form-urlencoded; charset=UTF-8	-
User-Agent	<User-Agent>	siehe [4.4-Identifikation des Clientsystems]
Body		
Form-Key	Wert	Hinweise
encrypted_signed_authentication_data	serialisierte Encrypted_Registration_Data-Instanz	vergleiche [RFC7516] , Abschnitt 3.1 "JWE Compact Serialization Overview"

8.2.2.2 Response des Authorization-Endpunkts bei Verwendung von alternativen Authentisierungsmitteln

Tabelle 27: Authentifizierung/Antwort Authorization-Endpunkt

Response			
http-response Status-Code	Body	Bedeutung (Vgl. A_21411)	Hinweise
200	-	Der Nutzer ist authentifiziert.	-
400	siehe Hinweise	VAL.1: Die Authentifizierung konnte nicht erfolgreich durchgeführt werden.	Im Hinblick auf die syntaktische Ausgestaltung der Fehlermeldungen gelten die Anforderungen aus [4.2-Fehlermeldungen].

8.2.3 Inspektion von Pairing-Daten am Pairing-Endpunkt

8.2.3.1 Request des Authenticator-Moduls zur Inspektion

Tabelle 28: Inspektion/Anfrage Authenticator-Modul

Request		
HTTP-Method	URL	Hinweise

GET	<URL des Pairing-Endpunkts aus dem Discovery Document>	vergleiche Attribut "uri_pair" in A_21429-02
Header		
Feld	Wert	Hinweise
Accept	application/json; charset=UTF-8	-
Authorization	"Bearer" <ACCESS_TOKEN>	vergleiche [RFC6750] , Abschnitt 2.1 "Authorization Request Header Field"
User-Agent	<User Agent>	siehe [4.4- Identifikation des Clientsystems]
Body		
-		

8.2.3.2 Response des Pairing-Endpunkts bei Inspektion

Tabelle 29: Inspektion/Antwort Pairing-Endpunkt

Response			
http-response Status-Code	Body	Bedeutung	Hinweise
200	Content-Type=application/JSON Pairing-Entries-Instanz.	Die Pairing-Daten wurden erfolgreich abgerufen.	Sofern keine Pairing-Daten vorliegen muss der Pairing-Endpunkt ein leeres Array in Form einer Pairing_Entries-Instanz zurückgeben.
403	siehe Hinweise	Siehe AC.1 in A_21441	Im Hinblick auf die syntaktische Ausgestaltung der Fehlermeldungen gelten die Anforderungen aus [4.2- Fehlermeldungen] . Die Ausgestaltung der http-response-Codes im Fall von technischen Fehlern (AC.2) muss entsprechend der Art des Fehlers ausgestaltet werden.
4XX/5XX		Siehe AC.2 in A_21441	

8.2.4 Deregistrierung von Pairing-Daten am Pairing-Endpunkt

8.2.4.1 Request des Authenticator-Moduls zur Deregistrierung

Tabelle 30: Deregistrierung/Anfrage Authenticator-Modul

Request		
HTTP-Method	URL	Hinweise
DELETE	<URL des Pairing-Endpunkts aus dem Discovery Document>/base64url(key_identifier)	vergleiche Attribut "uri_pair" in A_21429-02
Header		
Feld	Wert	Hinweise
Accept	application/json; charset=UTF-8	
Authorization	"Bearer" <ACCESS_TOKEN>	vergleiche [RFC6750], Abschnitt 2.1 "Authorization Request Header Field"
User-Agent	<User Agent>	siehe [4.4- Identifikation des Clientsystems]
Body		
-		

8.2.4.2 Response des Pairing-Endpunkts bei Deregistrierung

Tabelle 31: Deregistrierung/Antwort Pairing-Endpunkt

Response			
http-response Status-Code	Body	Bedeutung	Hinweise
204	-	Das durch die Kombination von key_identifier und idNummer identifizierte Pairing wurde deaktiviert.	-
403	siehe Hinweise	siehe AC.1 in A_21441	Im Hinblick auf die syntaktische Ausgestaltung der Fehlermeldungen gelten die Anforderungen aus
4XX/5XX		siehe AC.3 in A_21441	

			[4.2- Fehlermeldungen]. Die Ausgestaltung der http-response-Codes im Fall von technischen Fehlern (AC.3) muss entsprechend der Art des Fehlers ausgestaltet werden.
--	--	--	--

8.3 Vergleichsoperationen

Die folgenden Vergleichsoperationen sind innerhalb des Ablaufs notwendig:

8.3.1 Registrierung

Tabelle 32: Vergleichsoperationen im Rahmen der Registrierung

Datenobjekt	zu Datenobjekt	Vergleichsoperation
Signed_Pairing_Data/issuer	Registration_Data/auth_cert/issuer	Byte-weise
Signed_Pairing_Data/serialnumber	Registration_Data/auth_cert/serial_number	Gleiche Integer-Werte
Signed_Pairing_Data/not_after	Registration_Data/auth_cert/not_after	Gleichheit als Zahl bezogen auf die Repräsentation als Numeric-Date
Signed_Pairing_Data/auth_cert_subject_public_key_info	Registration_Data/auth_cert/SubjectPublicKeyInfo	Byte-weise