

1 *Beim vorliegenden Dokument handelt es sich um einen Entwurf der gematik in Vorbereitung auf zukünftige*
2 *normative Festlegungen als Grundlage entsprechender Zulassungs- und Bestätigungsverfahren. Die gematik*
3 *veröffentlicht diesen Entwurf mit dem Ziel, dass sich Interessierte bereits frühzeitig einen Überblick über die*
4 *mögliche Weiterentwicklung der Telematikinfrastruktur verschaffen können. Die gematik übernimmt keine*
5 *Gewähr für die Aktualität, Richtigkeit und Vollständigkeit dieses Entwurfes und behält sich das Recht vor,*
6 *ohne vorherige Ankündigung Änderungen oder Ergänzungen vorzunehmen oder von den Regelungen*
7 *insgesamt bzw. teilweise Abstand zu nehmen.*

11 Elektronische Gesundheitskarte und Telematikinfrastruktur

20 Spezifikation

21 Identity Provider-Dienst

22
23
24
25
26
27
28
29

Version: [1.12.0 CC](#)
Revision: [294966305861](#)
Stand: [09.12.11.2020](#)
Status: [zur Abstimmung](#) freigegeben
Klassifizierung: öffentlich [Entwurf](#)
Referenzierung: gemSpec_IDP_Dienst

30

Dokumentinformationen

31 Änderungen zur Vorversion

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

34

35 Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	30.06.20		initiale Erstellung des Dokuments	gematik
1.1.0	12. 11 10.20		Einarbeitung Scope-Themen aus R4.0.1	gematik
1.1.1	13.11.20		Einarbeitung P22.4	gematik
1.2.0 CC	09.12.20		Einarbeitung P22.5	gematik

36

Inhaltsverzeichnis

37	1 Einordnung des Dokumentes	6
38	1.1 Zielsetzung	6
39	1.2 Zielgruppe	6
40	1.3 Geltungsbereich	6
41	1.4 Abgrenzungen	6
42	1.5 Methodik	7
43	2 Systemüberblick	8
44	2.1 Allgemeiner Überblick	8
45	2.2 Detaillierter Überblick	9
46	3 Systemkontext	12
47	3.1 Akteure und Rollen	12
48	3.2 Begriffsdefinition	14
49	3.3 Verfahrensbeschreibung	15
50	3.4 Abweichende Verfahrensbeschreibung für Primärsysteme	18
51	3.5 Registrierung Anwendungsfrontend und Fachdienst	19
52	3.6 Anwendungsfrontend vorbereitende Maßnahmen	19
53	3.7 Anfrage eines ACCESS_TOKEN	19
54	3.8 Aufgaben des Authorization Endpunktes	20
55	3.8.1 Unzureichende Attribute für das Claim	20
56	3.8.2 Erstellung des AUTHORIZATION_CODE	20
57	3.9 Einreichen des AUTHORIZATION_CODE	20
58	3.10 Aufgabe des Token Endpunktes	20
59	3.11 Einreichen des "ACCESS_TOKEN" beim Fachdienst	20
60	3.12 Aufgabe des Fachdienstes	21
61	4 Zerlegung des Produkttyps	22
62	4.1.1 Allgemeine Sicherheitsanforderungen	22
63	4.1.2 Sicherheit der Netzübergänge	23
64	4.2 Fehlermeldungen	23
65	4.3 Schnittstellenbeschreibung des IdP Dienstes	24
66	4.4 Identifikation des Clientsystems	26
67	5 Funktionsmerkmale	27
68	5.1 Authorization Server Metadata (Discovery Document)	27
69	5.1.1 Aufbau des Discovery Documents	28

70	5.1.2 Erneuerung des Discovery Documents	29
71	5.1.3 Schutz des Discovery Documents	29
72	5.2 Authorization-Endpunkt	30
73	5.2.1 Authorization-Server-Eingangsdaten	30
74	5.2.2 Authorization-Endpunkt-Ausgangsdaten	34
75	5.3 Token-Endpunkt	37
76	5.3.1 Token-Endpunkt-Eingangsdaten	37
77	5.3.2 Token-Endpunkt-Ausgangsdaten	37
78	6 Anhang A – Verzeichnisse	41
79	6.1 Abkürzungen	41
80	6.2 Glossar	42
81	6.3 Abbildungsverzeichnis	43
82	6.4 Tabellenverzeichnis	44
83	6.5 Referenzierte Dokumente	44
84	6.5.1 Dokumente der gematik	44
85	6.5.2 Weitere Dokumente	45
86	1 Einordnung des Dokumentes	6
87	1.1 Zielsetzung	6
88	1.2 Zielgruppe	6
89	1.3 Geltungsbereich	6
90	1.4 Abgrenzungen	6
91	1.5 Methodik	7
92	2 Systemüberblick	8
93	2.1 Allgemeiner Überblick	8
94	2.2 Detaillierter Überblick	9
95	3 Systemkontext	12
96	3.1 Akteure und Rollen	12
97	3.2 Begriffsdefinition	14
98	3.3 Verfahrensbeschreibung	15
99	3.4 Abweichende Verfahrensbeschreibung für Primärsysteme	18
100	3.5 Registrierung Anwendungsfrontend und Fachdienst	19
101	3.6 Anwendungsfrontend vorbereitende Maßnahmen	19
102	3.7 Anfrage eines ACCESS TOKEN	19
103	3.8 Aufgaben des Authorization-Endpunktes	20
104	3.8.1 Unzureichende Attribute für das Claim	20
105	3.8.2 Erstellung des AUTHORIZATION CODE	20
106	3.9 Einreichen des AUTHORIZATION CODE	20
107	3.10 Aufgabe des Token-Endpunktes	20

108	3.11 Einreichen des "ACCESS_TOKEN" beim Fachdienst.....	20
109	3.12 Aufgabe des Fachdienstes	21
110	4 Zerlegung des Produkttyps	22
111	4.1.1 Allgemeine Sicherheitsanforderungen	22
112	4.1.2 Sicherheit der Netzübergänge.....	23
113	4.2 Fehlermeldungen.....	23
114	4.3 Schnittstellenbeschreibung des IdP-Dienstes.....	24
115	4.4 Identifikation des Clientsystems	26
116	5 Funktionsmerkmale	27
117	5.1 Authorization Server Metadata (Discovery Document)	27
118	5.1.1 Aufbau des Discovery Documents.....	28
119	5.1.2 Erneuerung des Discovery Documents	29
120	5.1.3 Schutz des Discovery Documents	29
121	5.2 Authorization-Endpunkt	30
122	5.2.1 Authorization Server Eingangsdaten	30
123	5.2.2 Authorization-Endpunkt Ausgangsdaten	34
124	5.3 Token-Endpunkt	37
125	5.3.1 Token-Endpunkt Eingangsdaten	37
126	5.3.2 Token-Endpunkt Ausgangsdaten	37
127	6 Anhang A – Verzeichnisse	41
128	6.1 Abkürzungen	41
129	6.2 Glossar	42
130	6.3 Abbildungsverzeichnis.....	43
131	6.4 Tabellenverzeichnis.....	44
132	6.5 Referenzierte Dokumente	44
133	6.5.1 Dokumente der gematik.....	44
134	6.5.2 Weitere Dokumente.....	45
135		
136		

137

1 Einordnung des Dokumentes

1.1 Zielsetzung

139 Die vorliegende Spezifikation definiert die Anforderungen zu Herstellung, Test und
140 Betrieb des Produkttyps Identity Provider (IdP)-Dienst. Der IdP-Dienst basiert auf den
141 Standards OpenID Connect (OIDC), Open Authorization 2.0 (OAuth 2) und JSON Web
142 Token (JWT). Die hier beschriebenen Schnittstellen werden vom Authenticator-Modul und
143 vom Anwendungsfrontend für eine Authentifizierung eines Nutzers anhand einer
144 Smartcard genutzt. Diese Authentifizierung ist die Voraussetzung, damit ein
145 Anwendungsfrontend Zugang zu Fachdaten eines Fachdienstes erlangen kann. Der IdP-
146 Dienst verwaltet und steuert den Authentifizierungsprozess für das E-Rezept und
147 perspektivisch auch für weitere Anwendungen.

1.2 Zielgruppe

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

1.3 Geltungsbereich

152 Dieses Dokument enthält normative Festlegungen zur Telematikinfrastruktur (TI) des
153 deutschen Gesundheitswesens. Der Gültigkeitszeitraum der vorliegenden Version und
154 deren Anwendung in Zulassungs- oder Abnahmeverfahren wird durch die gematik GmbH
155 in gesonderten Dokumenten (z.B. Dokumentenlandkarte, Produkttypsteckbrief,
156 Leistungsbeschreibung) festgelegt und bekannt gegeben.

157

Schutzrechts-/Patentrechtshinweis

159 *Die nachfolgende Spezifikation ist von der gematik allein unter technischen Gesichtspunkten erstellt
160 worden. Im Einzelfall kann nicht ausgeschlossen werden, dass die Implementierung der Spezifikation
161 in technische Schutzrechte Dritter eingreift. Es ist allein Sache des Anbieters oder Herstellers, durch
162 geeignete Maßnahmen dafür Sorge zu tragen, dass von ihm aufgrund der Spezifikation angebotene
163 Produkte und/oder Leistungen nicht gegen Schutzrechte Dritter verstoßen und sich ggf. die
164 erforderlichen Erlaubnisse/Lizenzen von den betroffenen Schutzrechtsinhabern einzuholen. Die
165 gematik GmbH übernimmt insofern keinerlei Gewährleistungen.*

166

1.4 Abgrenzungen

168

169 Nicht Bestandteil des vorliegenden Dokumentes sind die Verfahrensschritte zur Erstellung
170 des notwendigen Schlüsselmaterials. Es wird angenommen, dass Fachdienste ihre
171 innerhalb der TI zu verwendenden Zertifikate für die Transport Layer Security (TLS)-

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

174 Als Umsetzungsleitlinie ist [OpenID Connect Core 1.0] heranzuziehen. Die TI-weit
175 übergreifenden Festlegungen – insbesondere aus Dokumenten wie beispielsweise
176 [gemSpec_Krypt] bezüglich Algorithmen und Schlüsselstärken sowie [gemSpec_PKI]
177 bezüglich zu verwendender Zertifikatstypen und deren Attributausprägungen – haben
178 Bestand, sind weiterhin bindend und werden nicht in diesem Dokument beschrieben.

179

180 **1.5 Methodik**

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

185

186 Sie werden im Dokument wie folgt dargestellt:

187 **<AFO-ID> - <Titel der Afo>**

188 Text / Beschreibung

189 [**<=**]

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

192

193 **Hinweise auf offene Punkte**

Offene Punkten werden im Dokument in dieser Darstellung ausgewiesen.

194

195

2 Systemüberblick

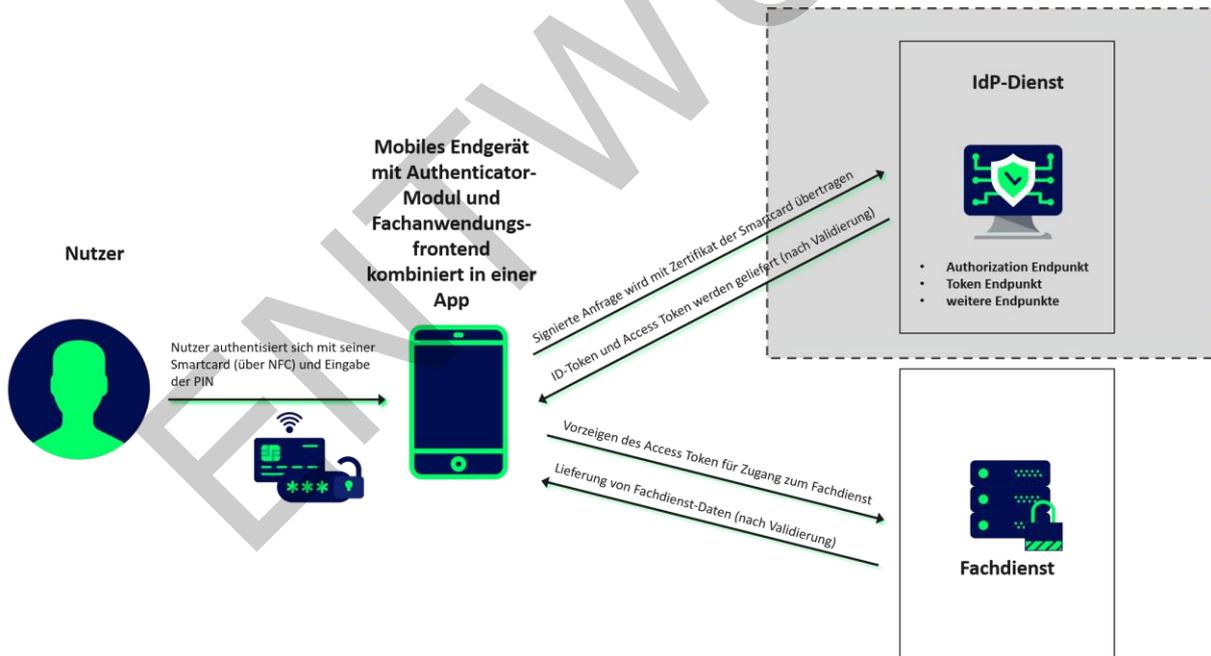
196

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

203

2.1 Allgemeiner Überblick

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



211

212

213

214

Abbildung 1: Systemüberblick (vereinfacht)

215

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

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

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

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

231

232 2.2 Detaillierter Überblick

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

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

- 245 • Discovery-Endpunkt ("OAuth 2.0 Authorization Server Metadata" [[RFC8414](#)])
- 246 • Redirection-Endpunkt (Teil des "The OAuth 2.0 Authorization Framework" [[RFC6749 # section-3.1.2](#)])
- 247
- 248 • Authorization-Endpunkt (Teil des "The OAuth 2.0 Authorization Framework" [[RFC6749](#)])
- 249
- 250 • Token-Endpunkt ([[RFC6749 # section-3.2](#)])
- 251 Teildienst 1 "ID_TOKEN" ([[openid-connect-core 1.0 # IDToken](#)])
- 252 Teildienst 2 "ACCESS_TOKEN" ([[RFC6749 # section-1.4](#) & [RFC6749 # section-5](#)])
- 253 Teildienst 3 "SSO_TOKEN" ([[RFC7519](#)]).

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

256 Teildienste wie das Authenticator-Modul und das Anwendungsfrontend befinden sich in
257 dem mit "Gerät des Nutzers" bezeichneten Bereich.

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

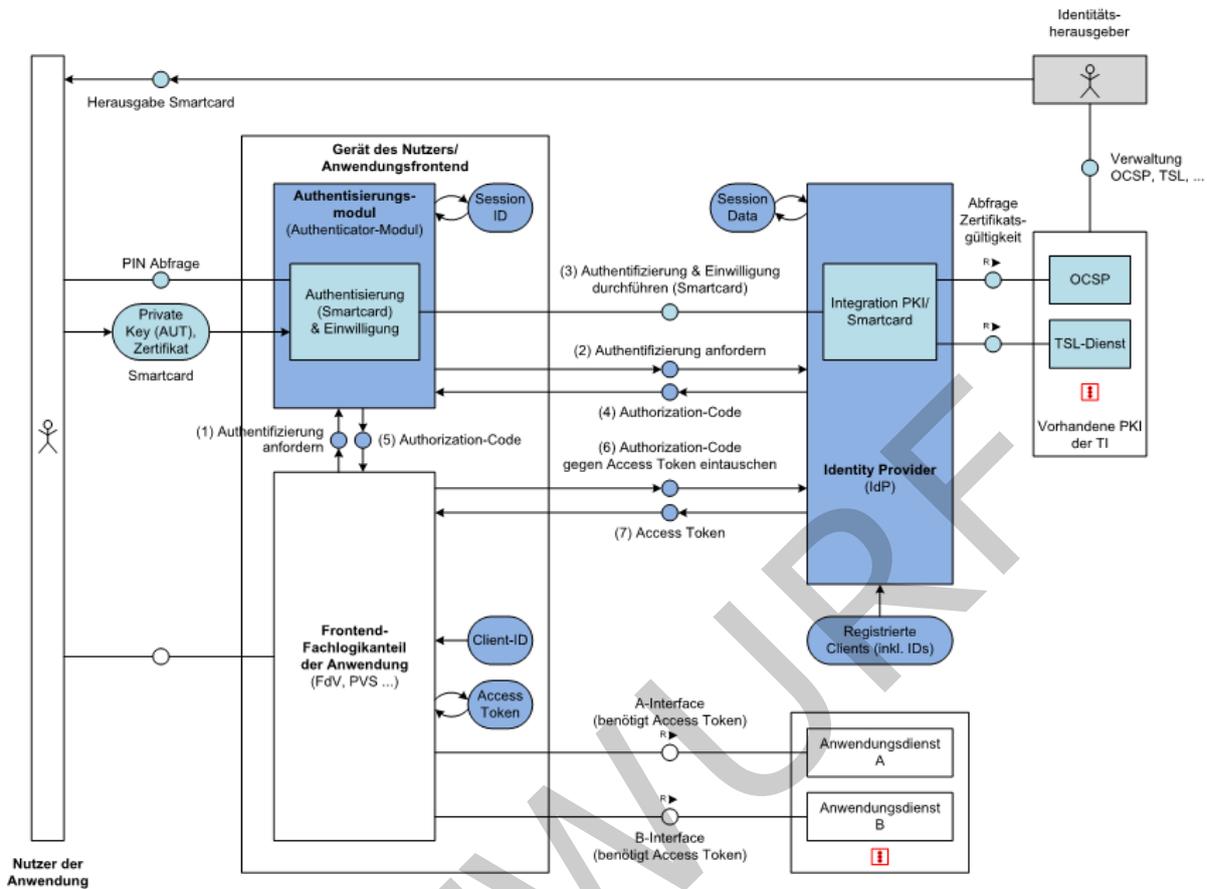


Abbildung 2: Übersichtsschaubild OAuth2.0 Smartcard-IdP-Dienst

260

261

262

263 Erläuterungen zur obigen Abbildung:

264 Die Teilschritte (1) und (5) werden bei mobilen Endgeräten (FdV) via Redirection-
 265 Endpunkt [RFC6749 # section-3.1.2] realisiert.

266 Die Teilschritte (1) und (5) können bei Primärsystemen (PVS, AVS, KVS) abweichend von
 267 [RFC6749 # section-9] behandelt werden.

268 Die Teilschritte (2) und (4) werden durch den Authorization-Endpoint gemäß [RFC6749
 269 # section-3.1] bedient. Der Teilschritt (3) Challenge-Response wird durch den
 270 Authorizatio-Endpoint bedient.

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

273 Der hier gezeigte Smartcard-IdP-Dienst stellt eine Basisleistung innerhalb der TI dar und
 274 soll die sichere Identifikation der Akteure anhand der ihnen bereitgestellten
 275 Identifikationsmittel (Smartcards) ermöglichen. Der Standard lässt hierbei die
 276 Einbringung weiterer Identity Provider für unterschiedlichste Identifikationsverfahren zu,
 277 ohne dass Fachdienste hierfür eine Änderung der Zugangsmechanismen realisieren
 278 müssen.

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

281 Weitere zu beachtende Standards sind die folgenden:

282 Request for Comments JWT (JSON Web Token) [RFC7519], JWS (JSON Web Signature)
 283 [RFC7515], JWE (JSON Web Encryption) [RFC7516], JWK (JSON Web Key) [

284 [RFC7517](#)], JWA (JSON Web Algorithm) [[RFC7518](#)] und WebFinger [[RFC7033](#)] sowie
285 OAuth 2.0 Bearer [[RFC6750](#)], OAuth 2.0 Assertion [[RFC7521](#)], OAuth 2.0 JWT Profile [[RFC7523](#)],
286 OAuth 2.0 Responses [[RFC6749](#)].

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

289

ENTWURF

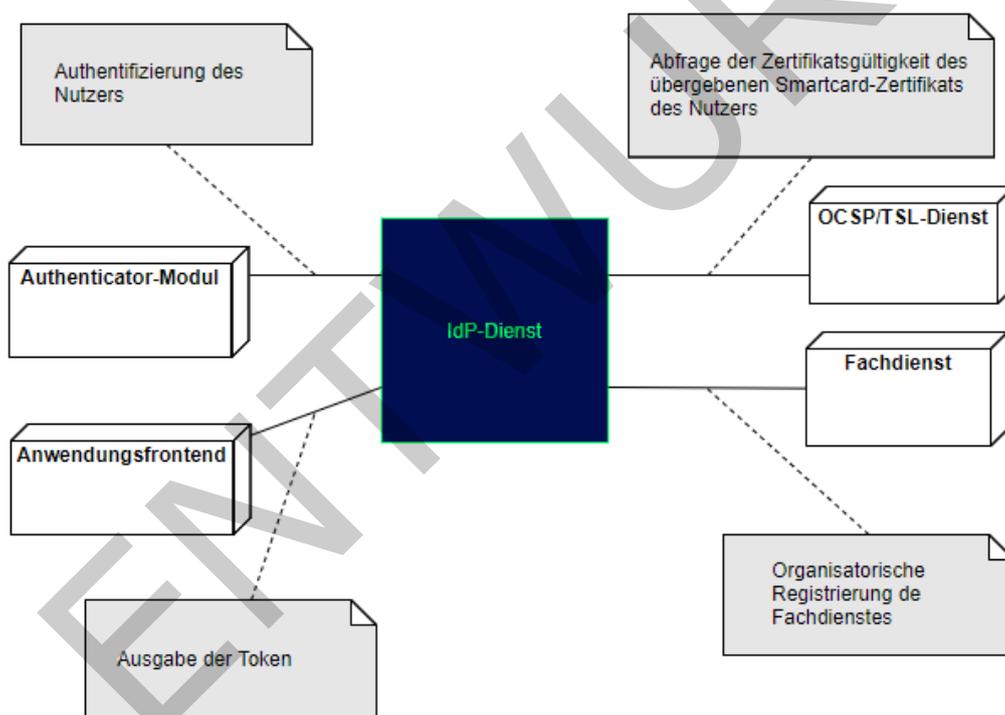
290

3 Systemkontext

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

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

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



303

304

305

Abbildung 3: Systemkontext aus Sicht des IdP-Dienstes

3.1 Akteure und Rollen

307 Im Systemkontext des IdP-Dienstes interagieren verschiedene Akteure (Nutzer und
 308 aktive Komponenten) in unterschiedlichen OAuth2-Rollen gemäß [[RFC6749 # section-
 309 1.1](#)].

310

311 **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

312

313 **Nutzer (Rolle: Resource Owner)**

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

316 Der Resource Owner verfügt über die folgenden Komponenten:

- 317 • Endgerät des Nutzers
- 318 • Authenticator-Modul
- 319 • Anwendungsfrontend

320

321 **Fachdienst (Rolle: Resource Server)**

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

327

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

330 Der Client greift mit dem Authenticator-Modul und dem Anwendungsfrontend (OIDC
331 Relying Party bzw. OAuth2 Client) auf Fachdienste (Resource Server) und ihre
332 geschützten Fachdaten (Protected Resource) zu. Das Anwendungsfrontend kann auf
333 einem Server als Webanwendung (Primärsystem als Terminalserver), auf einem Desktop-
334 PC oder einem mobilen Gerät (z.B. Smartphone) ausgeführt werden.

335

336 **IdP-Dienst (Rolle: Authorization Server)**

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

340

341 **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

342

343 Weitere Akteure im Kontext IdP-Dienst sind:

344 **Fachdaten (Rolle: Protected Resource)**345 Die geschützten Fachdaten, welche vom Fachdienst (Resource Server) angeboten
346 werden.

347

348 **3.2 Begriffsdefinition**349 Die folgende Tabelle enthält die Abkürzungen (für die privaten Schlüssel PrK und für
350 öffentliche Schlüssel PUK) der verschiedenen Endpunkte des IdP-Dienstes und deren
351 Verwendung.

352

353 **Tabelle 3: TAB_IDP_DIENST_0003 Bezeichnungen der Schlüssel und deren URI**

	PUK	URI PUK	private Key	URI Dienst
Authorizatio n Server				URI_DD
Authorizatio n-Endpunkt (AUTH)	PUK_AUTH - für die Signaturprüfung des "AUTHORIZATION_COD E" und des "SSO_TOKEN"	PUK_URI_AUT H	PrK_AUTH - zum Signieren des "AUTHORIZATION_COD E" und des "SSO_TOKEN"	URI_AUTH
Discovery- Endpunkt (DISC)	PUK_DISC - für die Signaturprüfung des Discovery Document	PUK_URI_DISC	PrK_DISC - zum Signieren des Discovery Document	URI_DISC

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

354
 355 *Hinweis:* Werden alle Teildienste auf einem Server gemeinsam betrieben, so können
 356 diese dasselbe Schlüsselmaterial verwenden. Werden Teildienste auf unterschiedlichen
 357 physischen oder logischen Servern betrieben, so sind die Endpunkte mit eigenem
 358 Schlüsselmaterial auszustatten.

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

362

363 3.3 Verfahrensbeschreibung

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

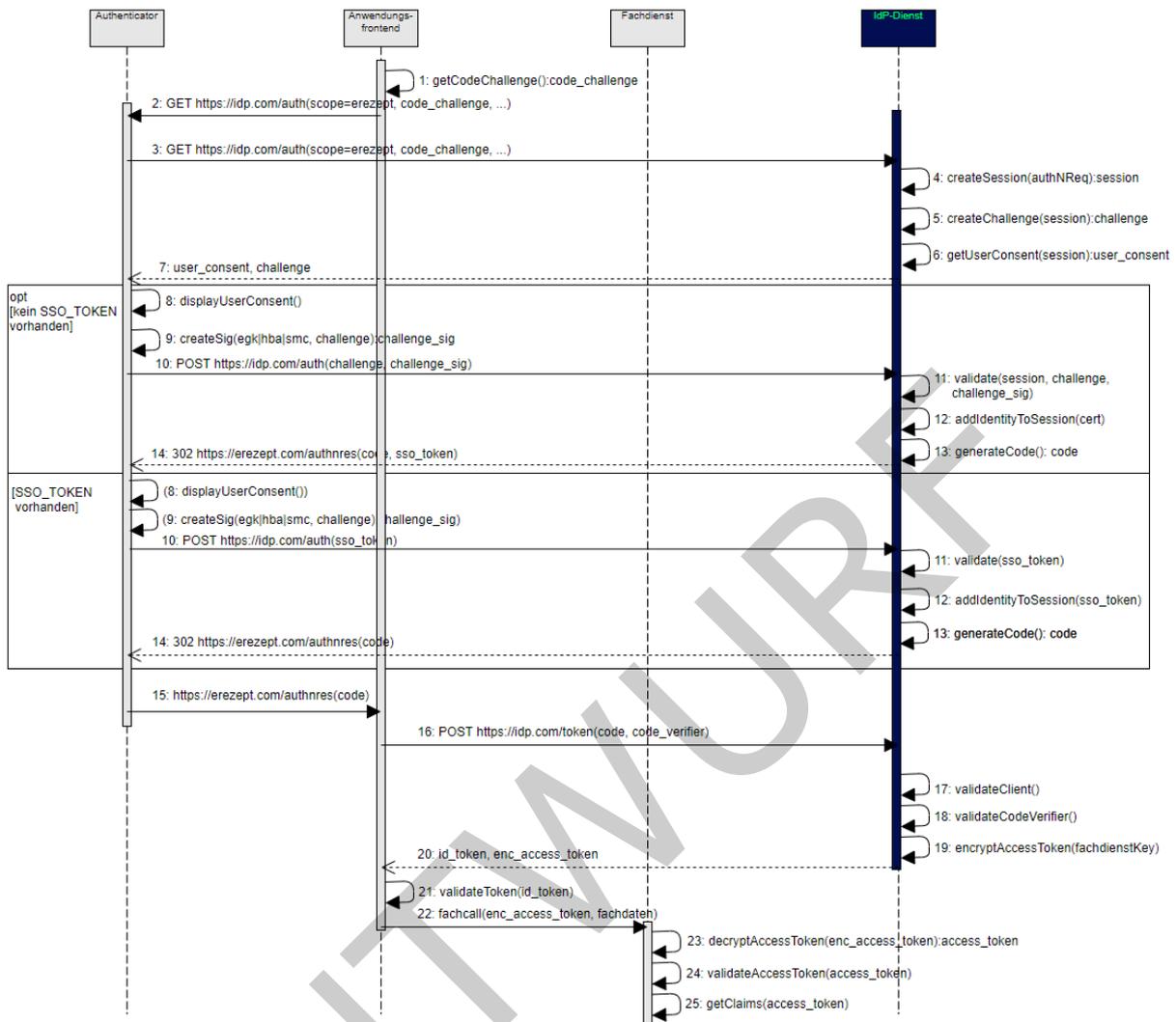


Abbildung 4: Datenfluss-Diagramm IdP-Dienst

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

1. Das Anwendungsfrontend erzeugt sich einen "CODE_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 die "CODE_CHALLENGE" gemäß [RFC8252 # Anhang B] an das Authenticator-Modul.
3. Das Authenticator-Modul überträgt die "CODE_CHALLENGE" mit der genutzten "code_challenge_method" S256 weiter an den Authorization-Endpunkt des IdP-Dienst.
4. Der Authorization-Endpunkt legt eine "SESSION_ID" an und speichert alle Informationen zum Vorgang in der "CHALLENGE".
5. Der Authorization-Endpunkt stellt alle Informationen zusammen und erzeugt die "CHALLENGE".

- 387 6. Der Authorization-Endpunkt stellt den mit dem entsprechenden Fachdienstes
388 vereinbarten Consent (Zustimmung des Nutzers zur Verarbeitung der angezeigten
389 Daten) zusammen.
- 390 7. Der Authorization-Endpunkt überträgt "CHALLENGE" und Consent-Abfrage
391 "USER_CONSENT" zum Authenticator-Modul.
- 392 8. Das Authenticator-Modul fordert den Nutzer zu Consent-Freigabe auf mittels
393 Smartcard und PIN-Eingabe. Falls bereits ein "SSO_TOKEN" beim Authenticator-
394 Modul existiert, entfällt dieser Schritt.
- 395 9. Das Authenticator-Modul verwendet die PIN um die "CHALLENGE" von der
396 Smartcard signieren zu lassen. Falls bereits ein "SSO_TOKEN" beim Authenticator-
397 Modul existiert, entfällt dieser Schritt.
- 398 10. Das Authenticator-Modul überträgt "CHALLENGE" mit dem Smartcard-Zertifikat an
399 den IdP-Dienst (Antwort Schritt 7). Falls ein "SSO_TOKEN" beim Authenticator-
400 Modul existiert, wird diese Token anstatt der "Challenge" zum IdP-Dienst
401 transportiert.
- 402 11. Der Authorization-Endpunkt validiert die "SESSION_ID", "CHALLENGE" und
403 "SIGNATUR". Die Signatur wird anhand des im "x5c"-Header mitgelieferten
404 Authentifizierungszertifikats der Smartcard validiert. Falls ein "SSO_TOKEN"
405 angenommen wurde, wird dieses validiert. Entschlüsselt wird das "SSO_TOKEN"
406 vom Authorization-Endpunkt mit seinem Schlüsselmaterial, welches er zur
407 Verschlüsselung genutzt hat. Die Überprüfung der Signatur des "SSO_TOKEN" führt
408 der Authorization-Endpunkt anhand seines öffentlichen Schlüssels "PUK_AUTH"
409 durch.
- 410 12. Der Authorization-Endpunkt verknüpft die "SESSION_ID" mit der Identität aus der
411 Signatur. Falls ein "SSO_TOKEN" angenommen wurde, verknüpft der Authorization-
412 Endpunkt die "SESSION_ID" mit der Identität aus dem "SSO_TOKEN".
- 413 13. Der Authorization-Endpunkt erstellt den "AUTHORIZATION_CODE".
- 414 14. Der Authorization-Endpunkt überträgt den "AUTHORIZATION_CODE" und den
415 "SSO_TOKEN" an das Authenticator-Modul (Antwort Schritt 3). Falls das das
416 Authenticator-Modul ein vorhandenes "SSO_TOKEN" an den Authorization-Endpunkt
417 zur Erlangung eines "AUTHORIZATION_CODE" geschickt hat, wird kein neues
418 "SSO_TOKEN" vom Authorization -Endpunkt erstellt und verschickt. Der
419 "AUTHORIZATION_CODE" und das "SSO_TOKEN" werden vom Authorization-Endpunkt
420 mit seinem privaten Schlüssel "PrK_AUTH" signiert. Der Authorization-Endpunkt
421 verschlüsselt das "SSO_TOKEN" für sich. Das zur Verschlüsselung verwendete
422 Schlüsselmaterial muss die Vorgaben der [gemSpec_Krypt] beachten.
- 423 15. Das Authenticator-Modul überträgt den "AUTHORIZATION_CODE" an das
424 Anwendungsfondend (Antwort Schritt 2).
- 425 16. Das Anwendungsfondend sendet "CODE_VERFIER" und "AUTHORIZATION_CODE"
426 zum Token-Endpunkt des IDP-Dienstes.
- 427 17. Der Token-Endpunkt validiert den Client ("SESSION_ID" in der signierten
428 "CHALLENGE"). Der Token-Endpunkt validiert den "AUTHORIZATION_CODE" anhand
429 des öffentlichen Schlüssels "PUK_AUTH" des Authorization-Endpunktes.
- 430 18. Der Token-Endpunkt validiert den "CODE_VERFIER" und gleicht diesen mit der
431 "CODE_CHALLENGE" ab.

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

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

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

467 3.4 Abweichende Verfahrensbeschreibung für Primärsysteme

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

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

479 3.5 Registrierung Anwendungsfrontend und Fachdienst

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

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

486 **A_20737 - Ermöglichung einer organisatorischen Registrierung für** 487 **Anwendungsfrontends und Fachdienste**

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

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

496 3.6 Anwendungsfrontend vorbereitende Maßnahmen

497 Das Anwendungsfrontend muss ein "CODE_VERIFIER" (Zufallswert) gemäß [[RFC7636 #](#)
498 [section-4.1](#)] und hierüber einen Hash, die "CODE_CHALLENGE", gemäß [[RFC7636 #](#)
499 [section-4.2](#)] mit dem Algorithmus S256 gemäß [[RFC7636 # section-4.2](#)] erzeugen.

500 3.7 Anfrage eines ACCESS_TOKEN

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

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

507 Inhalt der Anfrage ist:

- 508 • die "REDIRECT_URI" sowie Bezeichnung des aufzurufenden Fachdienstes,
- 509 • die eigene Hersteller-ID, Programm Kürzel und Versionsnummer,
- 510 • der über das eigene "CODE_VERIFIER" [[RFC7636 # section-4.1](#)] gebildete HASH
511 "code_challenge" [[RFC7636 # section-4.2](#)] mit Angabe des Algorithmus
512 "code_challenge_method" [[RFC7636 # section-4.3](#)],

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

515

516 **3.8 Aufgaben des Authorization-Endpunktes**

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

521 **3.8.1 Unzureichende Attribute für das Claim**

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

525 **3.8.2 Erstellung des AUTHORIZATION_CODE**

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

531 **3.9 Einreichen des AUTHORIZATION_CODE**

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

534 **3.10 Aufgabe des Token-Endpunktes**

535 Der Token-Endpunkt des IdP-Dienstes nimmt die Daten des Anwendungsfrontends
536 entgegen und prüft neben deren Integrität, ob der eingereichte "CODE_VERIFIER" bei
537 Nutzung des Hash-Verfahrens S256 (nach [[RFC7636 # section-4.2](#)]) zum bitgleichen
538 Hash-Wert führt. Stimmt der Hash-Werte aus dem initialen Aufruf des Authenticator-
539 Moduls - die "CODE_CHALLENGE" - mit dem gebildeten Hash-Wert überein, ist
540 sichergestellt, dass Aufrufer und Initiator identisch sind. Der Token-Endpunkt gibt
541 daraufhin das "ID_TOKEN" und das "ACCESS_TOKEN" an das Anwendungsfrontend heraus.

542 **3.11 Einreichen des "ACCESS_TOKEN" beim Fachdienst**

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

545 **3.12 Aufgabe des Fachdienstes**

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

553 Bei positiver Validierung gewährt der Fachdienst Zugriff auf seine Fachdaten.

554

ENTWURF

555

4 Zerlegung des Produkttyps

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

564 Der IdP-Dienst stellt die zentralisierte Identitätsprüfung der auf die Fachdienste
565 zugreifenden Nutzer bereit. Als weitere Teile der Gesamtlösung sind neben dem IdP-
566 Dienst die Clients (Anwendungsfrontend/Primärsystem) und die Fachdienste zu nennen,
567 auf denen Fachdaten für den Zugriff durch die Nutzer (z. B. Versicherte oder Bediener
568 eines AVS, PVS oder KVS) bereitgestellt werden. Ein IdP-Dienst bietet Fachdiensten seine
569 Dienste an, auf welche Millionen Nutzer zeitgleich zugreifen. Eine wesentliche Ergänzung
570 des IdP-Dienstes ist das Authenticator-Modul, welches auf den dezentralen Komponenten
571 in den Praxen, Kliniken, Apotheken und bei den Versicherten betrieben wird.
572

573

574 **A_20687 - Bereitstellung der PUK**

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

580 **A_20732 - Aufnahme der öffentlichen Schlüssel in das Discovery Document**

581 Der Authorization Server MUSS zu jedem privaten Schlüssel dessen öffentlichen Teil mit
582 einer eigenen absoluten URI in das Discovery Document aufnehmen. [<=]

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

586 **A_20686 - Erweiterte Nutzung von Schlüsseln**

587 Der Authorization Server MUSS die einzelnen Schnittstellen (AUTH, DISC, TOKEN) mit
588 getrennten Interfaces bedienen. [<=]

589

590 **4.1.1 Allgemeine Sicherheitsanforderungen**

591 **A_20582-01 - IdP-Dienst - Berücksichtigung OWASP-Top-10-Risiken**

592 Der IdP-Dienst MUSS Maßnahmen zum Schutz sowohl vor den zum Zulassungszeitpunkt
593 aktuellen OWASP-Top-10-Risiken umsetzen, als auch nach den zum Zulassungszeitpunkt
594 aktuellen OWASP-Top-10-Risiken. [<=]

595 4.1.2 Sicherheit der Netzübergänge

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

600 **A_20583 - IdP-Dienst – Sicherung zum Transportnetz Internet durch Paketfilter**

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

606 **A_20584 - IdP-Dienst – Platzierung des Paketfilters Internet**

607 Der Anbieter des IdP-Dienstes DARF den Paketfilter des IdP-Dienstes zum Schutz in
608 Richtung Transportnetz Internet NICHT physisch auf dem vorgeschalteten TLS-
609 terminierenden Load Balancer implementieren. [<=]

610 **A_20585 - IdP-Dienst – Richtlinien für den Paketfilter zum Internet**

611 Der Paketfilter des IdP-Dienstes MUSS die Weiterleitung von IP-Paketen an der
612 Schnittstelle zum Internet auf das HTTPS- Protokoll beschränken. [<=]

613 **A_20586-01 - IdP-Dienst – Verhalten bei Vollauslastung**

614 Der Anbieter des IdP-Dienstes MUSS den Paketfilter des IdP-Dienstes so konfigurieren,
615 dass bei Vollauslastung der Systemressourcen im IdP-Dienst keine weiteren
616 Verbindungen angenommen werden. [<=]

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

620

621 **A_20587 - IdP-Dienst – Richtlinien zum TLS-Verbindungsaufbau**

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

630 4.2 Fehlermeldungen

631

632 **A_20680 - Format der Fehlermeldungen**

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

635 **A_20681 - Nutzung von eindeutigen Error-Codes bei der Erstellung von Fehlermeldungen**

636
637 Der IdP-Dienst MUSS Fehler durch eine eindeutige Nummer erkennbar machen und der
638 gematik eine Liste der Error-Codes zur Verfügung stellen, damit die Ursachenklärung
639 vereinfacht möglich wird. [<=]

640 **A_20682 - Verwendung eines einheitlichen Schemas für die Aufbereitung von**
 641 **Fehlermeldungen**

642 Der IdP-Dienst MUSS alle ausgeworfenen Fehlermeldungen zur Weiterverarbeitung in
 643 einem einheitlichen Schema aufbereiten und bereitstellen. Zeitstempel MÜSSEN auf der
 644 UTC basieren.

645 Beispiel: Ist eine Signatur nicht vorhanden oder defekt, bricht der Authorization-
 646 Endpunkt die Bearbeitung mit dem registrierten Fehlercode und einer für den Nutzer
 647 verständlichen Fehlermeldung ab.

648 **Tabelle 4: TAB_IDP_DIENST_0004 Schema der Fehlermeldungen**

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

649 [**<=**]

650 **A_20683 - Formulierung der Fehlermeldungen**

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

654 **A_20684 - Nutzung einer eindeutigen Beschreibung beim Aufbau von**
 655 **Fehlermeldungen**

656 Der IdP-Dienst MUSS jedem Fehler eine eindeutige eigene Beschreibung zukommen
 657 lassen, sodass eine Fehlermeldung nicht für unterschiedliche Fehlerursachen zur
 658 Anwendung kommt. [**<=**]

659 **A_20685 - Ausgabe der Fehlermeldungen in umgekehrter Reihenfolge des**
 660 **Auftretens**

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

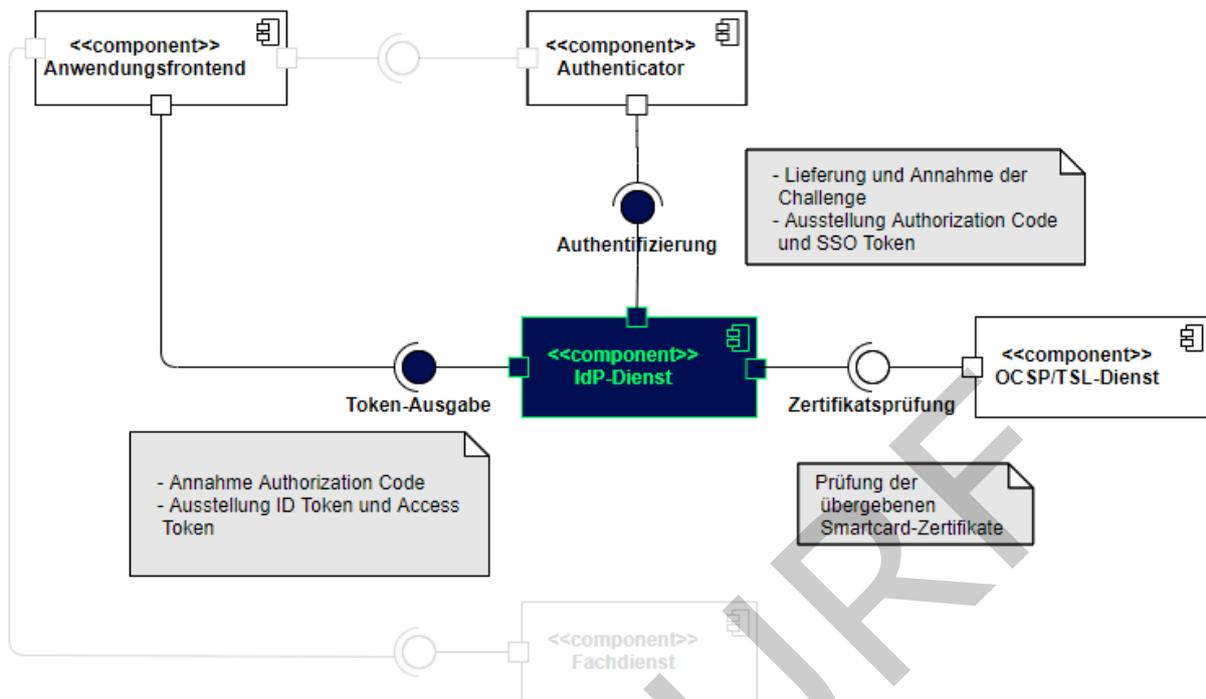
663

664 **4.3 Schnittstellenbeschreibung des IdP-Dienstes**

665 Der IdP-Dienst bietet zahlreiche Schnittstellen gegenüber unterschiedlichen Akteuren
 666 inner- und außerhalb der TI an, weswegen es notwendig ist, die einzelnen Schnittstellen
 667 so zu beschreiben, dass andere Akteure deren Funktionsweise leichter verstehen
 668 können. Nachfolgende Abbildung skizziert die Schnittstellen des IdP-Dienstes.

669 Komponenten und Schnittstellen, welche nicht direkt vom IdP-Dienst genutzt werden,
 670 sind in der Abbildung grau hinterlegt.

671



672

673

Abbildung 5: Schnittstellen des IdP-Dienstes

674

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

684 Bei der ersten Kontaktaufnahme erzeugt der Authorization Server die
 685 "SUBJECT_SESSION", welche im weiteren Verlauf als Zeitpunkt der letzten
 686 Authentisierung gegen die eGK oder den HBA gewertet wird. Basierend darauf dürfen
 687 weitere "ACCESS_TOKEN" und "SSO_TOKEN" für andere Anwendungsfrontends und
 688 Fachdienste ausgegeben werden, wenn das jeweils vorliegende Claim durch die dem
 689 Authorization Server vorliegenden Informationen bedient werden kann. Ist der Zeitpunkt
 690 der letzten Authentisierung zu lange her oder wird das Authenticator-Modul zum ersten
 691 Mal gestartet, muss eine Authentisierung erfolgen.

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

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

697

698

699 **4.4 Identifikation des Clientsystems**

700 Der IdP-Dienst verwaltet und steuert den Authentisierungsprozess für das E-Rezept und
701 perspektivisch auch weitere Anwendungen. Damit kommt ihm eine Relevanz in der
702 Gesundheitsversorgung zu, die sich zum einen in einer hohen Verfügbarkeit und zum
703 anderen in einem hohen Angriffspotential widerspiegelt. Zur Unterstützung der
704 betrieblichen Überwachung des IdP-Dienstes wird die Nutzung der im Feld befindlichen
705 Clientsysteme protokolliert. Dabei ist der Zugriff auf die Schnittstellen des IdP-Dienstes
706 nur durch Primärsysteme der Leistungserbringer, sein eigenes Authenticator-Modul und
707 zugelassene E-Rezept-FdVs zulässig. Der E-Rezept-Fachdienst erkennt die Clientsysteme
708 anhand des User-Agent-Header eingehender HTTP-Requests und protokolliert diesen
709 Wert.
710

711 **A_20588 - IdP-Dienst - Erkennung Clientsystem User-Agent**

712 Der IdP-Dienst MUSS das vom aufrufenden Nutzer verwendete Clientsystem
713 (Authenticator-Modul, E-Rezept-FdV oder Primärsystem) anhand des im HTTP-Request
714 enthaltenen Header-Feld "User-Agent" gemäß [RFC7231] erkennen und in den Einträgen
715 zur Performance-Rohdatenerfassung gemäß [gemSpec_Perf] protokollieren. Der IdP-
716 Dienst MUSS bei fehlendem User-Agent-Header den Request mit dem HTTP-Status-Code
717 400 beantworten, damit in der Betriebsüberwachung des IdP-Dienstes die Nutzung
718 unzulässiger Clientsysteme erkannt werden kann. [≤]

719 **A_20589 - IdP-Dienst – Ausschluss bestimmter Clientsystem-Versionnummern
720 von der Kommunikation**

721 Der IdP-Dienst MUSS die aus dem Internet vom Clientsystem mitgeteilte
722 Versionsnummer aus dem HTTP-Header User-Agent, erkennen und festgelegte
723 Versionsnummern über ein Blacklisting von einer Kommunikation mit dem IdP-Dienst
724 ausschließen können. Der IdP-Dienst MUSS in diesen Fällen eine entsprechende
725 Fehlermeldung an das Clientsystem geben. [≤]

726 **A_20590 - IdP-Dienst – Ausschluss von Clientsystem-Versionen**

727 Der Anbieter des IdP-Dienstes MUSS ausschließlich auf Anweisung der gematik
728 Clientsysteme mit bestimmten Versionsnummern von einer Kommunikation mit dem IdP-
729 Dienst ausschließen. [≤]

730 **A_20742 - Vergabe der "client_id" durch den IdP-Dienst**

731 Der IdP-Dienst MUSS bei der organisatorischen Registrierung des Anwendungsfrentends
732 diesem eine eindeutige "client_id" zur Nutzung des IdP-Dienstes zuweisen.
733 [≤]

734

5 Funktionsmerkmale

735 5.1 Authorization Server Metadata (Discovery Document)

736 Der Authorization Server dient dazu bestehende Identitäten zu prüfen und das
 737 Prüfungsergebnis in einer einheitlichen Form abgestimmt und durch zusätzliche
 738 Mechanismen gesichert bereitzustellen. Basis dieser Dienstleistung ist ein
 739 vertrauenswürdigen Verzeichnis, aus welchem hervorgeht, an welchen Schnittstellen
 740 dieser Dienst oder seine Teildienste erreichbar sind, wie diese Schnittstellen abgesichert
 741 sind und woher man die zur Etablierung der gewünschten Sicherheit erforderlichen
 742 Materialien beziehen kann. Gemäß dem verwendeten Standard OpenID Connect mit
 743 OAuth 2.0 kommen JSON Web Token (JWT), JSON Web Encryption (JWE), JSON Web
 744 Signature (JWS) und JSON Web Key (JWK) zum Einsatz.

745 Um nutzenden Anwendungen eine einheitliche Bezugsquelle für die Adressierung von
 746 Schnittstellen zu schaffen, werden die für alle Akteure grundlegenden Schnittstellen im
 747 sogenannten Discovery Document zusammengefasst und dort unter der "URI_DISC"
 748 gemäß [[RFC8414 "OAuth 2.0 Authorization Server Metadata"](#)] veröffentlicht.

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

752 ~~A_20895—Anbieter IdP-Dienst—Resource Records FQDN idp~~

753 ~~Der Anbieter des IdP-Dienstes MUSS im Namensraum der TI und in den Nameservern~~
 754 ~~Internet die Resource Records gemäß nachstehender Tabelle verwalten.~~

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

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

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

758
 759 <https://idp.zentral.idp.splitdns.ti-dienste.de/openid-configuration> - Schnittstelle
 760 Discovery-Endpunkt
 761 <https://idp.zentral.idp.splitdns.ti-dienste.de/authorization> - Schnittstelle Authorization-
 762 Endpunkt
 763 <https://idp.zentral.idp.splitdns.ti-dienste.de/token> - Schnittstelle Token-Endpunkt
 764 [**<=**]

765 **A_20457 - Verwendung eindeutiger URI**

766 Der IdP-Dienst MUSS alle verwendeten Adressen in Form von URL gemäß [[RFC1738](#)]
 767 angeben und in einem Discovery Document gemäß [[RFC8414 # section-2](#)] innerhalb der
 768 TI und im Internet veröffentlichen. [**<=**]

769 **A_20688 - Discovery Document interne und externe Adressierung**

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

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

776 Hinweis: Es gibt je ein internes und externes (public) "Discovery Document". Diese
 777 unterscheiden sich in den darin angebotenen URI, welche gleichlautend im Host-Anteil
 778 auf unterschiedliche Domänen bzw. Top-Level-Domain (TLD) verweisen.

779 **A_20689 - Internes Discovery Document - Prüfung vor Veröffentlichung**

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

783 **A_20690 - Externes Discovery Document - Prüfung vor Veröffentlichung**

784 Der IdP-Dienst MUSS alle von ihm angebotenen URI betreiben und URI anderer Dienste,
 785 insbesondere Fachdienste, vor deren Veröffentlichung im externen Discovery Document
 786 auf bloße Erreichbarkeit prüfen. [**<=**]

787

788

789 **5.1.1 Aufbau des Discovery Documents**

790 Der Authorization Server muss das Discovery Document gemäß [[RFC8414](#)] bereitstellen.

791 **A_20439 - Das Discovery Document enthält statische Adressen**

792 Der Discovery-Endpunkt MUSS sowohl im internen, als auch im externen Discovery
 793 Document die Akteure mit ihrer URI veröffentlichen.
 794 [**<=**]

795 A_20458 - Inhalte des Discovery Documents

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

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

806 [**<=**]

807 *Hinweis:* Ein Beispiel eines Discovery Documents kann unter folgendem Link gefunden
808 werden: [HEART II # rfc.section.3.1.5](#)

809

810 5.1.2 Erneuerung des Discovery Documents

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

815 A_20691 - Das Discovery Document ist maximal 24 Stunden alt

816 Der Discovery-Endpoint MUSS das Discovery Document regelmäßig alle 24 Stunden oder
817 nach durchgeführten Änderungen umgehend neu erstellen, mit dem
818 "PrK_DISC" signieren und und am mit der gematik vereinbarten Downloadpunkt
819 "URI_DISC" bereitstellen.

820 [**<=**]

821 A_20592 - Aktualisierung der Discovery Documents

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

825

826 5.1.3 Schutz des Discovery Documents

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

**829 ~~A_19874-04A_19874-03~~ - Bereitstellung des internen Discovery Documents
830 innerhalb der TI**

831 Der IdP-Dienst MUSS das interne Discovery Document mit einem Zertifikat des Typs
832 FD.SIG und der technischen Rolle „oid_idpd“ gemäß [gemSpec_Krypt # Abschnitt 3.7]
833 signiert, an einem spezifischen Downloadpunkt TLS-gesichert innerhalb der TI
834 bereitstellen.

835 Die URL des Downloadpunktes lautet: "[https://idp.zentral.idp.splitdns.ti-](https://idp.zentral.idp.splitdns.ti-dienste.de/-well-known/openid-configuration)
836 [dienste.de/-well-known/openid-configuration](https://idp.zentral.idp.splitdns.ti-dienste.de/-well-known/openid-configuration)". <=[**<=**]

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

839 **A_19877-02A_19877-01 - Bereitstellung des externen Discovery**
840 **Documents im Internet**

841 Der IdP-Dienst MUSS das externe Discovery Document mit einem Zertifikat des
842 Typs FD.SIG und der technischen Rolle „oid_idpd“ gemäß [gemSpec_Krypt #
843 Abschnitt 3.7] signiert und TLS-gesichert im Internet zum Download bereitstellen.
844 Die URL des Downloadpunktes lautet: "[https://idp.zentral.idp.splitdns.ti-](https://idp.zentral.idp.splitdns.ti-dienste.de/.well-known/openid-configuration)
845 [dienste.de/.well-known/openid-configuration](https://idp.zentral.idp.splitdns.ti-dienste.de/.well-known/openid-configuration)" [`<=`]

846 *Hinweis:* Die für die Rolle des IdP-Dienstes vorgesehene professionOID ist in
847 [gemSpec_OID] beschrieben. ~~Der externe Downloadpunkt des Discovery Document ist~~
848 ~~der folgende: [idp.zentral.idp.splitdns.ti-](https://idp.zentral.idp.splitdns.ti-dienste.de/.well-known/openid-configuration)~~
849 ~~[dienste.de/.well-known/openid-configuration](https://idp.zentral.idp.splitdns.ti-dienste.de/.well-known/openid-configuration)~~

850 **A_20591 - Festlegungen zur Signatur der Discovery Documents**

851 Der IdP-Dienst MUSS die Signatur der Discovery Documents als base64-kodierte CMS-
852 Signatur gemäß [RFC5652] realisieren und die Festlegungen aus gemSpec_Krypt#5.6.2
853 beachten.

854 Der IdP-Dienst MUSS bei der Signaturerstellung das Signaturzertifikat als Attribut *signing*
855 *certificate reference* gemäß [CADES # Kapitel 5.7.3 „Signing Certificate Reference

856 Attributes"] einbetten. [`<=`]

857

858 **5.2 Authorization-Endpunkt**

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

862 **A_20434 - Einhaltung der Standards bei der Realisierung des Authorization-**
863 **Endpunkts**

864 Der IdP-Dienst MUSS die Schnittstelle „Authorization-Endpunkt“ gemäß [RFC6749 "The
865 OAuth 2.0 Authorization Framework"] und [RFC8252 „OAuth 2.0 for Native Apps“] und
866 weiteren darin festgelegten Standards implementieren. [`<=`]

867 **A_19863 - Schutz vor überalterter Software (Apple)**

868 Der Anbieter IdP-Dienst MUSS dafür Sorge tragen, dass die im Apple App Store
869 veröffentlichte Software bei Änderungen automatisiert aktualisiert wird, sodass jederzeit
870 die dauerhafte Verwendung fehlerhafter Software ausgeschlossen werden kann. [`<=`]

871 **A_19865 - Schutz vor überalterter Software (Android)**

872 Der Anbieter des IdP-Dienstes MUSS dafür Sorge tragen, dass die im Google Play Store
873 veröffentlichte Software bei Änderungen automatisiert aktualisiert wird, sodass jederzeit
874 die dauerhafte Verwendung fehlerhafter Software ausgeschlossen werden kann. [`<=`]

875 **5.2.1 Authorization Server Eingangsdaten**

876

877 **A_20698 - Annahme des Authorization Request**

878 Der Authorization-Endpunkt MUSS die im Authorization Request des Authenticator-
879 Moduls mitgelieferten "CODE_CHALLENGE" und den "SCOPE" annehmen. [`<=`]

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

```
882 GET /auth?response_type=code&scope=openid%20e-  
883 rezept&state=af0ifjlsldkj&client_id=ZXJlemVwdClhcHA&redirect_uri=https%3A%2F  
884 %2Fapp.e-  
885 rezept.com%2Fauthnres&code_challenge_method=S256&code_challenge=S41HgHxhXL1  
886 CIpfGvivWYpb09b_QKzva-9ImuZbt0Is
```

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

892 **A_20376 - Verwendung des Attributes "state"**

893 Der Authorization-Endpunkt MUSS den vom Anwendungsfrontend initiierten "state"-
894 Parameter gemäß [[RFC6749 # section-10.12](#)] bei einer Redirection an den Client in
895 seiner Antwort verwenden. [\leq]

896 **A_20731 - Verwendung des Attributes "auth_time"**

897 Der Authorization-Endpunkt MUSS den Parameter "auth_time" mit dem Zeitpunkt der
898 letzten Authentisierung gegen das zugelassene Authentifizierungsmittel (z.B. Auslösen
899 der Signatur durch Smartcard in freigeschaltetem Zustand) setzen. [\leq]

900

901 **A_20440 - Schematische Prüfung des Consent**

902 Der IdP-Dienst MUSS den eingereichten Consent auf dessen Übereinstimmung mit dem
903 vorliegenden Claim (mit dem Fachdienst abgestimmte Key/Value-Paare im Payload des
904 Token) zum beantragten Token abgleichen, insbesondere die "redirect_uri" aus dem
905 Registrierungszusammenhang. [\leq]

906 **A_20379 - Abbruch bei schematischer Inkonsistenz im Consent**

907 Der Authorization-Endpunkt MUSS die Bearbeitung mit dem registrierten Fehlercode und
908 einer für den Nutzer verständlichen Fehlermeldung abbrechen, wenn das Schema des
909 Consent und das des vorliegenden Claims nicht übereinstimmen. [\leq]

910 **A_20310 - Verarbeitung des Consent**

911 Der Authorization-Endpunkt MUSS die Ausstellung des vereinbarten
912 "AUTHORIZATION_CODE" mit den im Claim vorliegenden Parametern veranlassen, wenn
913 der Authorization-Endpunkt den eingereichten Consent allen vorgesehenen Prüfungen
914 unterzogen hat und dabei keine Fehler aufgetreten sind. [\leq]

915 **A_20459 - Das Attribut AUTH_TIME muss in allen Token unverändert bleiben**

916 Der Authorization-Endpunkt DARF den Zeitpunkt der letzten Authentisierung im Attribut
917 "auth_time" NICHT verändern. [\leq]

918 **A_20699 - Annahme der signierten "CHALLENGE"**

919 Der Authorization-Endpunkt MUSS die "CHALLENGE", signiert mit dem Zertifikat der
920 Smartcard des Nutzers, übertragen durch das Authenticator-Modul annehmen. [\leq]

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

```
923 GET /auth?response_type=code&scope=openid%20e-  
924 rezept&state=af0ifjlsldkj&client_id=ZXJlemVwdClhcHA&redirect_uri=https%3A%2F  
925 %2Fapp.erezept.com%2Fauthnres&code_challenge_method=S256&code_challenge=S41  
926 HgHxhXL1CIpfGvivWYpb09b_QKzva-9ImuZbt0Is
```

```
927 HTTP/1.1  
928 Host: idp.com
```

- 929 X-E-Rezept-App: 1.0
930 Accept: application/json
931 User-Agent: E-Rezept-App/1.0
- 932 **A_20951 - Validierung der Signatur und des Zertifikats der "CHALLENGE"**
933 Der Authorization-Endpoint MUSS die Signatur der vom Authenticator-Modul übertragenen, signierten
934 "CHALLENGE" anhand des mitgelieferten Authentifizierungs-Zertifikats überprüfen. Die Überprüfung
935 MUSS neben der Signatur auch das Authentifizierungszertifikat anhand von OCSP umfassen. [`<=`]
- 936 **A_20946 - Annahme eines "SSO_TOKEN"**
937 Der Authorization-Endpoint MUSS einen vom Authenticator-Modul übertragenen
938 "SSO_TOKEN" annehmen. [`<=`]
- 939 **A_20947 - Entschlüsselung des "SSO_TOKEN"**
940 Der Authorization-Endpoint MUSS den angenommenen "SSO_TOKEN" mit seinem eigenen
941 Schlüsselmaterial, welches zur Verschlüsselung genutzt wurde, entschlüsseln. [`<=`]
- 942 **A_20948 - Validierung des "SSO_TOKEN"**
943 Der Authorization-Endpoint MUSS den angenommenen und entschlüsselten "SSO_TOKEN"
944 validieren. Die Validierung MUSS die Überprüfung der Signatur anhand seines
945 öffentlichen Schlüssels PUK_AUTH und die Überprüfung der zeitlichen Gültigkeit des
946 "SSO_TOKEN" anhand des Attributs "auth_time" umfassen. [`<=`]
- 947 **A_20949 - Anforderung einer Authentisierung bei negativer Validierung des**
948 **"SSO_TOKEN"**
949 Der Authorization-Endpoint MUSS eine neue Authentisierung vom Authenticator-Modul
950 anfordern, wenn die Validierung des vom Authenticator-Moduls eingereichten
951 "SSO_TOKEN" fehlschlägt. [`<=`]
- 952 **A_20950 - Positive Validierung des "SSO_TOKEN"**
953 Der Authorization-Endpoint MUSS bei der positiven Validierung des vom Authenticator-
954 Moduls eingereichten "SSO_TOKEN" einen "ACCESS_TOKEN" für den angefragten Fachdienst
955 ausstellen. [`<=`]
- 956 Hinweis: Der Authorization-Endpoint muss damit die im "SSO_TOKEN" gelieferten Claims
957 überprüfen und einen "AUTHORIZATION_CODE" für den angefragten Fachdienst
958 ausstellen.
- 959 **A_20522 - Erstellen einer "SESSION_ID"**
960 Der Authorization-Endpoint MUSS eine neue "SESSION_ID" anlegen, sobald ein
961 Authorization Request eingeht. [`<=`]
- 962 **A_20523 - Zusammenstellung der Claims zum "user_consent"**
963 Der Authorization-Endpoint MUSS die für den vorgetragenen "SCOPE" vom einfordernden
964 Fachdienst erwarteten Claims zur "USER_CONSENT"-Anfrage zusammenstellen. [`<=`]
- 965
- 966 **A_20460 - Der Authorization-Endpoint bestätigt ausschließlich**
967 **Zertifikatsinformationen**
968 Der Authorization-Endpoint MUSS bei der Annahme des Zertifikates durch ein Challenge-
969 Response-Verfahren prüfen, ob der Nutzer auch die zum Zertifikat (Besitz) gehörige PIN
970 (Wissen) kennt, um sicherzustellen, dass der Nutzer berechtigt ist, die vorgetragene
971 Identität (Zertifikat) zu nutzen. [`<=`]
- 972 **A_20521-01 - Inhalt der Challenge an das Authenticator-Modul**
973 Der IdP-Dienst MUSS die ihm vorliegenden Session-Informationen (z.B. "SESSION_ID",
974 "CODE_CHALLENGE", "SCOPE" und alle Informationen über Anwendungsfrontend und
975 Authenticator-Modul) mit seinem privaten Schlüssel "PrK_AUTH" signieren und als JWT
976 ergänzt um die "USER_CONSENT"-Anfrage an das Authenticator-Modul senden. [`<=`]

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

```
979 Challenge JWT:
980 challenge_headers = {
981   "typ": "JOSE+JSON",
982   "iat": 1591714252326,
983   "exp": 1591714552326,
984   "jti": "c3a8f9c8-aa62-11ea-ac15-6b7a3355d0f6",
985   "snc": "sLlxlkskAyuzdDOWe8nZeeQVFBWgscNkRcpgHmKidFc"
986 }
987 challenge_payload = {
988   "response_type": "code",
989   "scope": "openid e-rezept",
990   "client_id": "ZXJlemVwdClhcHA",
991   "state": "af0ifjsldkj",
992   "redirect_uri": "https://app.e-rezept.com/authnres",
993   "code_challenge_method": "S256",
994   "code_challenge": "S41HgHxhXLlCIpfGvivWYpbO9b_QKzva-9ImuZbt0Is"
995 }
```

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

```
1000 HTTP/1.1 200 OK
1001 Content-Type: application/json
1002 Cache-Control: no-store
1003 Pragma: no-cache
1004
1005 {
1006   "challenge":
1007   "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpUURSlNPTiIsImhhdCI6MTU5MTcxNDI1MjMy.....
1008   ",
1009   "user_consent": {
1010     "client_name": "E-Rezept App",
1011     "url": "https://e-rezept.com/",
1012     "requested_scope": {
1013       "openid": "Der Zugriff auf den ID Token"
1014       "e-rezept": "Zugriff auf die E-Rezept Funktionalität."
1015     },
1016     "show_once": true,
1017     "amr": ["JWT-Challenge-Response"]
1018     // ggf. mehr Informationen, welche dem Nutzer angezeigt werden sollen,
1019     wie die Auflistung der mit der Zustimmung weitergegebenen Daten
1020   }
1021 }
```

1022

1023 **A_20604 - Signatur der Challenge**

1024 Der IdP-Dienst MUSS die Challenge für die Authentisierung mit einem Zertifikat des Typs
1025 FD.SIG und der technischen Rolle „oid_idpd“ gemäß [gemSpec_Krypt # Abschnitt 3.7]
1026 signieren. [\leq]

1027 **A_20313 - Inhalte des Claims**

1028 Der IdP-Dienst MUSS "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN" für
1029 unterschiedliche Fachdienste gemäß den mit dem jeweiligen Fachdienst abgestimmten
1030 Claims bereitstellen. Sind Inhalte des Claims teilweise oder das gesamte Claim für einen

1031 registrierten Fachdienst nicht gesetzt, befüllt der IdP-Dienst die einzelnen Parameter der
1032 Gültigkeitsdauer ("SUBJECT_SESSION", "AUTHORIZATION_CODE", "ACCESS_TOKEN",
1033 "SSO_TOKEN" und "ID_TOKEN") gemäß der spezifizierten Maximalwerte. [<=]

1034 **A_20692 - Maximale Gültigkeitsdauer einer "SUBJECT_SESSION"**

1035 Der Authorization Server DARF die zeitliche Gültigkeit einer "SUBJECT_SESSION" NICHT
1036 länger als 86400 Sekunden (24 Stunden) einstellen.

1037 Der Parameter "auth_time" beinhaltet den Zeitpunkt der letzten Authentisierung. [<=]

1038 **A_20314 - Maximale Gültigkeitsdauer des "AUTHORIZATION_CODE"**

1039 Der Authorization Server DARF die zeitliche Gültigkeit des "AUTHORIZATION_CODE" NICHT
1040 länger als 180 Sekunden (Challenge-Response) und nach dessen Übergabe an das
1041 Anwendungsfrontend nicht länger als 60 Sekunden einstellen. [<=]

1042 **A_20315 - "AUTHORIZATION_CODE" nach Gültigkeitsende nicht mehr
1043 verwenden**

1044 Der Authorization Server DARF außerhalb der Gültigkeitsdauer eingehenden
1045 "AUTHORIZATION_CODE" NICHT in "ID_TOKEN", "ACCESS_TOKEN" oder "SSO_TOKEN"
1046 eintauschen. [<=]

1047 Die Gültigkeitsdauer des "AUTHORIZATION_CODE" wird im Claim des angesprochenen
1048 Fachdienstes definiert.

1049

1050 **A_20462 - Maximale Gültigkeitsdauer des "ID_TOKEN"**

1051 Der Token-Endpunkt DARF "ID_TOKEN" mit einer Gültigkeitsdauer von mehr als 86400
1052 Sekunden (24 Stunden) NICHT ausstellen. [<=]

1053 **A_20463 - Maximale Gültigkeitsdauer des "ACCESS_TOKEN"**

1054 Der Token-Endpunkt DARF "ACCESS_TOKEN" mit einer Gültigkeitsdauer von mehr als 300
1055 Sekunden (5 Minuten) NICHT ausstellen. [<=]

1056 Die Gültigkeitsdauer des "ACCESS_TOKEN" wird im Claim des angesprochenen
1057 Fachdienstes definiert.

1058

1059 **A_20464 - Token-Endpunkt (Datensparsamkeit)**

1060 Der Token-Endpunkt DARF andere Informationen, als die im Claim geforderten,
1061 NICHT herausgeben. [<=]

1062 **A_20318 - Keine Token für widerrufen Entitäten**

1063 Der Authorization-Endpunkt DARF für nicht existente Entitäten NICHT
1064 einen "AUTHORIZATION_CODE", einen "ID_TOKEN", einen "ACCESS_TOKEN" oder
1065 einen "SSO_TOKEN" auszustellen. [<=]

1066 **A_20465 - Zertifikatsprüfung gegen OCSP-Responder**

1067 Der Authorization-Endpunkt MUSS das Zertifikat des Antragstellers immer gegen den
1068 zugehörigen OCSP-Responder innerhalb der TI auf Gültigkeit prüfen. [<=]

1069

1070 5.2.2 Authorization-Endpunkt Ausgangsdaten

1071 Konnten alle Prüfungen des eingereichten Consent erfolgreich abgeschlossen werden,
1072 erstellt der Authorization-Endpunkt ein "ID_TOKEN", "ACCESS_TOKEN", ergänzt durch ein
1073 "SSO_TOKEN". Die Übertragung der ~~Token-erfolgt~~ Tokenerfolgt jedoch nicht direkt über
1074 das Authenticator-Modul, sondern in Form eines "AUTHORIZATION_CODE". Die Token

1075 werden am Token-Endpunkt zum Download bereitgestellt, wo das jeweilige
1076 Anwendungsfrontend diese gegen gleichzeitige Vorlage von "authorization_code" und
1077 des eigenen "code_verifier", auf welchem der bereits vorliegende Hash-Wert beruht,
1078 erhält.

1079 **A_20377 - Verwendung des Attributes "state"**

1080 Der Authroization-Endpunkt MUSS den "state"-Parameter [[RFC6749 # section-10.12](#)]
1081 des Anwendungsfrontends in allen darauf basierenden Responses verwenden.[<=]

1082 **A_20694 - Zusammenstellung des "SSO_TOKEN"**

1083 Der Authorization-Endpunkt MUSS den "SSO_TOKEN" so zusammenstellen, dass alle
1084 Informationen, welche für die Ausstellung eines neuen "ACCESS_TOKEN" benötigt werden,
1085 im Token vorhanden sind.[<=]

1086 **A_20695 - Signieren des "SSO_TOKEN"**

1087 Der Authorization-Endpunkt MUSS den "SSO_TOKEN" mit seinem eigenen privaten
1088 Schlüssel signieren "PrK_AUTH".[<=]

1089 **A_20696 - Verschlüsselung des "SSO_TOKEN"**

1090 Der Authorization-Endpunkt verschlüsselt den "SSO_TOKEN" für sich selbst mit eigenem
1091 Schlüsselmaterial, welches die gemSpec_Krypt beachtet.[<=]

1092 **A_20697 - Zusammenstellung des "AUTHORIZATION_CODE"**

1093 Der Authorization-Endpunkt erzeugt den "AUTHORIZATION_CODE" anhand der vom
1094 Authenticator-Modul übergebenen Daten im "CHALLENGE".[<=]

1095 **A_20319 - Signatur des "AUTHORIZATION_CODE"**

1096 Der IdP-Dienst MUSS den "AUTHORIZATION_CODE" für die Authentisierung mit einem
1097 Zertifikat des Typs FD.SIG und der technischen Rolle „oid_idpd“ gemäß [gemSpec_Krypt
1098 # Abschnitt 3.7] signieren damit das Authenticator-Modul sicher gewährleisten kann,
1099 dass der eingehende "AUTHORIZATION_CODE" tatsächlich vom IdP-Dienst stammt [[RFC7519 # section-7.1](#)].[<=]
1100

1101 **A_20693 - Senden des "AUTHORIZATION_CODE" und "SSO_TOKEN" an die 1102 "REDIRECT_URI"**

1103 Der IdP-Dienst MUSS den "AUTHORIZATION_CODE" und den "SSO_TOKEN" an das
1104 Authenticator-Modul über eine Redirection an die registrierte "REDIRECT_URI" des
1105 Anwendungsfrontends senden.[<=]

1106 **A_20320 - Sichere Übertragung des "AUTHORIZATION_CODE"**

1107 Der Authorization-Endpunkt MUSS den Transport des "AUTHORIZATION_CODE" über
1108 unsichere Netze (z.B. Internet) durch Verwendung von Transport Layer Security (TLS)
1109 gemäß den Vorgaben der [gemSpec_Krypt] sichern [[RFC7523 # section-7](#)].[<=]

1110 *Hinweis:* Nachfolgend wird beispielhaft ein "AUTHORIZATION_CODE" in Form eines JSON
1111 Web Token (JWT) dargestellt:

1112 Authorization Code:

```
1113 code_header = {  
1114   "typ": "JOSE",  
1115   "jti": "18017c1c-aa7b-11ea-ac15-6b7a3355d0f6",  
1116   "iat": 1591714352326,  
1117   "exp": 1591714652326,  
1118   "msg_type": "code"  
1119 }  
1120 code_payload = {  
1121   "response_type": "code",  
1122   "scope": "openid e-ezept",  
1123   "client_id": client_id,
```


1180
1181
1182

1183 **5.3 Token-Endpunkt**

1184 Am Token-Endpunkt nimmt der Authorization Server den "AUTHORIZATION_CODE",
1185 welchen er selbst am Authorization-Endpunkt ausgegeben hat, entgegen. Da beide vom
1186 Authorization Server selbst erstellt wurden, ist deren Prüfung auf Integrität keine
1187 besondere Herausforderung. Allerdings muss der Token-Endpunkt beim Einreichen eines
1188 "AUTHORIZATION_CODE" das dabei mit übertragene "CODE_VERIFIER" verarbeiten, um
1189 mittels Vergleich der Hash-Werte die Übereinstimmung des den "AUTHORIZATION_CODE"
1190 einreichenden mit dem ursprünglich authentisierten Client sicherzustellen. Das
1191 verwendete Hash-Verfahren ist im Authorization Request anzugeben.

1192 **5.3.1 Token-Endpunkt Eingangsdaten**

1193 **A_20321 - Annahme und Prüfung von "AUTHORIZATION_CODE" und** 1194 **"CODE_VERIFIER"**

1195 Der Token-Endpunkt MUSS den vom Anwendungsfrontend übertragenen
1196 "AUTHORIZATION_CODE" nach Überprüfung des zeitgleich
1197 eingereichten "CODE_VERIFIER" entwerten und das "ID_TOKEN" und "ACCESS_TOKEN"
1198 gesichert herausgeben. Der Token-Endpunkt MUSS die Überprüfung des
1199 "CODER_VERIFIER" gegen die "CODE_CHALLENGE" mit S256 (Algorithmus nach [[RFC7636](#)
1200 # section-4.2]) durchführen.[<=]

1201 **A_20474 - "AUTHORIZATION_CODE" einmalige Verwendung**

1202 Der Authorization Server MUSS sicherstellen, dass auf einen wiederholt eingereichten
1203 "AUTHORIZATION_CODE" keine weiteren Token herausgegeben werden.[<=]

1204 **A_20323 - TOKEN-Ausgabe Protokollierung in allen Fällen**

1205 Der Token-Endpunkt MUSS die Herausgabe der "TOKEN" im Positiv- wie auch im
1206 Negativfall protokollieren.[<=]

1207 **5.3.2 Token-Endpunkt Ausgangsdaten**

1208

1209 Alle vom IdP-Dienst herausgegebenen Informationen müssen mit dem privateKey des
1210 jeweiligen Teildienstes signiert sein, da die mit TLS abgesicherte Verbindung nicht in
1211 allen Anwendungsszenarien die Integrität der übertragenen Daten gewährleistet.

1212

1213 **A_20524 - Befüllen der Claims "given_name", "family_name",** 1214 **"organizationName", "professionOID" und "idNummer"**

1215 Der Token-Endpunkt MUSS benötigte Attribute in Claims für das auszustellende
1216 "ACCESS_TOKEN" und das "ID_TOKEN" ausschließlich aus dem ihm mit der "challenge"
1217 eingereichten Authentifizierungs-Zertifikat der Smartcard (eGK, HBA oder SMC-B)
1218 beziehen.

1219 Der Token-Endpunkt MUSS das Attribut "given_name" und "family_name" der
1220 juristischen und natürlichen Personen sowie die Attribute "organizationName",

1221 "professionOID" und "idNummer" entsprechend des Datenformates der
 1222 Informationsquelle (Zertifikat) wie folgt befüllen:
 1223

1224 **Tabelle 5: TAB_IDP_DIENST_0005 Befüllung der Attribute "given_name",**
 1225 **"family_name", "organizationName", "professionOID" und "idNummer"**

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

1226 [**<=**]

1227 Hinweis: Nachfolgend wird ein beispielhafter Payload eines "ACCESS_TOKEN" dargestellt.

1228 {

1229 "iss": "https://idp1.telematik.de/jwt",
 1230 "sub": "RabcUSuuWKKZEEHmrcNm_kUDOW13uaGU5Zk8OoBwiNk",
 1231 "professionOID": "1.2.276.0.76.4.50",
 1232 "nbf": 1585336956,
 1233 "exp": 1585337256,
 1234 "iat": 1585336956,
 1235 "given_name": "der Vorname",
 1236 "family_name": "der Nachname",
 1237 "organizationName": "Institutions- oder Organisations-Bezeichnung",
 1238 "idNummer": "3-15.1.1.123456789",

```
1239     "jti": "<IDP>_01234567890123456789",
1240     "aud": "https://erp.telematikzentral.erp.ti-dienste.de/login"
1241 }

```

1242 **A_20952 - Claim "aud" im Token setzen**

1243 Der IdP-Dienst MUSS den Claim "aud" im "ACCESS_TOKEN" entsprechend des angefragten
1244 Scopes des Authenticator-Moduls mit der URL des Fachdienstes füllen.[<=]

1245 Hinweis:

1246 Für den E-Rezept-Fachdienst wird beispielsweise der folgende Wert genutzt: "aud" :
1247 "erp.zentral.erp.ti-dienste.de".

1248 **A_20327 - Signatur des "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN"**

1249 Der Token-Endpoint MUSS alle erstellten "ID_TOKEN", "ACCESS_TOKEN" und
1250 "SSO_TOKEN" mit seinem privateKey "PRK_TOKEN" gemäß signieren, um dessen Integrität
1251 sicherzustellen und eine eindeutige Erklärung über dessen Herkunft abzugeben. [
1252 [RFC7523 # section-3](#) Spiegelpunkt 9 und [openid-heart-oauth2-](#)
1253 [1_0.html#rfc.section.3.2.1](#)] sind zu gewährleisten.[<=]

1254 **A_20328 - Verschlüsselung des "ACCESS_TOKEN"**

1255 Der Token-Endpoint MUSS das "ACCESS_TOKEN" mit dem öffentlichen Schlüssel des
1256 Fachdienstes "PUK_FD" verschlüsseln, um das "ACCESS_TOKEN" vor Kenntnisnahme
1257 durch Dritte, z.B. auch auf dem Endgerät des Nutzers, zu schützen [[RFC6750 # section-](#)
1258 [5.2](#)].
1259 [<=]

1260 **A_20329 - Sichere Übertragung von "ID_TOKEN", "ACCESS_TOKEN" und** 1261 **"SSO_TOKEN"**

1262 Der Token-Endpoint MUSS "ID_TOKEN", "ACCESS_TOKEN" und "SSO_TOKEN" beim
1263 Transport mit Transport Layer Security (TLS) gemäß [gemSpec_Krypt] schützen.
1264 [<=]

1265 **A_20330 - Ausgabe der Token**

1266 Der Token-Endpoint MUSS für den Versand der "ID_TOKEN" und "ACCESS_TOKEN" an das
1267 Anwendungsfrontend, die vom Authenticator-Modul im Consent der mit dem Vorgang
1268 verbundenen "SUBJECT_SESSION" gemeldete URI verwenden. Eine URI-Umleitung MUSS
1269 ausgeschlossen werden [[RFC6749 # section-10.6](#)]. [<=]

1270 *Hinweis:* Nachfolgend wird beispielhaft ein "ID_TOKEN" und ein "ACCESS_TOKEN" in Form
1271 eines JSON Web Token (JWT) dargestellt:

1272 ID Token:

```
1273 idt_headers = {
1274     # "typ":"JOSE+JSON"
1275 }
1276 idt_payload = {
1277     "iss": "https://idp.com/oidc",
1278     "sub": "RabcUSuuWKKZEEHmrcNm_kUDOW13uaGU5Zk8OoBwiNk",
1279     "aud": [client_id],
1280     "iat": 1591714452326,
1281     "exp": 1591714752326,
1282     "at_hash": at_hash
1283 }
1284

```

1285 *Hinweis:* Der ID Token wird vom Token-Endpoint signiert (mit "PrK_TOKEN")

1286

1287 Access Token:

```
1288 at_headers = {
1289   "typ":"at+JWT"
1290 }
1291 at_payload = {
1292   "iss": "https://idp.com/oidc",
1293   "sub": "RabcUSuuWKKZEEHmrcNm_kUDOW13uaGU5Zk8OoBwiNk",
1294   "aud": "https://rs.e-rezept.com/,erp.zentral.erp.ti-dienste.de",
1295   "client_id": client_id,
1296   "scope": "openid e-rezept",
1297   "iat": 1591714452326,
1298   "exp": 1591714752326,
1299   "jti": "d8557394-ab37-11ea-ac15-6b7a3355d0f6",
1300   "professionOID":"<die Profession OID>",
1301   "idNummer":"<KV-Nummer>",
1302   "name":"<Name des Versicherten>"
1303 }
```

1304 *Hinweis:* Der Access Token wird vom Token-Endpunkt signiert (mit "PrK_TOKEN") und für
1305 den Fachdienst verschlüsselt (mit "PUK_FD"). Nachfolgend wird beispielhaft die Antwort
1306 des Token-Endpunkt als dargestellt. Der "ID_TOKEN" und der "ACCESS_TOKEN" werden nur
1307
1308 angedeutet.

```
1309
1310 HTTP/1.1 200 OK
1311 Content-Type: application/json
1312 Cache-Control: no-store
1313 Pragma: no-cache
1314
1315 {"token_type": "Bearer",
1316  "expires_in": 300,
1317  "id_token": "...",
1318  "access_token": "...",
1319 }
1320
```

1321

6 Anhang A – Verzeichnisse

1322

6.1 Abkürzungen

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

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

1323 **6.2 Glossar**

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

Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.
ID Token	Ein auf JSON basiertes und nach [RFC7519] (JWT) genormtes Identitäts-Token, mit dem ein Client (Anwendungsfrontend) die Identität eines Nutzers überprüfen kann.
Open Authorization 2.0	Ein Protokoll zur Autorisierung für Web-, Desktop und Mobile Anwendungen. Dabei wird es einem Endbenutzer (Resource Owner) ermöglicht, einer Anwendung (Client) den Zugriff auf Daten oder Dienste (Resources) zu ermöglichen, die von einem Dritten (Resource Server) bereitgestellt werden.
OpenID Connect	OpenID Connect (OIDC) ist eine Authentifizierungsschicht, die auf dem Autorisierungsframework OAuth 2.0 basiert. Es ermöglicht Clients, die Identität des Nutzers anhand der Authentifizierung durch einen Autorisierungsserver zu überprüfen (siehe [openid-connect-core 1.0]).
JSON Web Token	Ein auf JSON basiertes und nach [RFC7519] (JWT) genormtes Access-Token. Das JWT ermöglicht den Austausch von verifizierbaren Claims innerhalb seines Payloads.
Resource Owner	OAuth2-Rolle (siehe [RFC6749 # section-1.1]): Eine Entität (Nutzer), die einem Dritten den Zugriff auf ihre geschützten Ressourcen gewähren kann. Diese Ressourcen werden durch den Resource Server bereitgestellt. Ist der Resource Owner eine Person, wird dieser als Nutzer bezeichnet.
Resource Server	OAuth2 Rolle (siehe [RFC6749 # section-1.1]): Der Server (Dienst), auf dem die geschützten Ressourcen (Protected Resources) liegen. Er ist in der Lage, auf Basis von Access Tokens darauf Zugriff zu gewähren. Ein solcher Token repräsentiert die delegierte Autorisierung des Resource Owners.
SSO Token	Gegen Vorlage eines gültigen SSO Token ist keine erneute Nutzerauthentisierung für die Ausstellung eines Access Tokens am IdP-Dienst nötig.
Token-Endpunkt	Ein Endpunkt des Authorization Servers, welcher für die Ausstellung von Token ("ID_TOKEN" und "ACCESS_TOKEN") zuständig ist.

1324 Das Glossar wird als eigenständiges Dokument (vgl. [gemGlossar]) zur Verfügung
 1325 gestellt.

1326 6.3 Abbildungsverzeichnis

1327 [Abbildung 1: Systemüberblick \(vereinfacht\)](#)..... 8

1328	Abbildung 2: Übersichtsschaubild OAuth2.0 Smartcard-IdP-Dienst.....	10
1329	Abbildung 3: Systemkontext aus Sicht des IdP-Dienstes.....	12
1330	Abbildung 4: Datenfluss-Diagramm IdP-Dienst.....	16
1331	Abbildung 5: Schnittstellen des IdP-Dienstes	25
1332	Abbildung 1: Systemüberblick (vereinfacht).....	8
1333	Abbildung 2: Übersichtsschaubild OAuth2.0 Smartcard-IdP-Dienst.....	10
1334	Abbildung 3: Systemkontext aus Sicht des IdP-Dienstes.....	12
1335	Abbildung 4: Datenfluss-Diagramm IdP-Dienst.....	16
1336	Abbildung 5: Schnittstellen des IdP-Dienstes	25
1337		

1338 6.4 Tabellenverzeichnis

1339	Tabelle 1: TAB_IDP_DIENST_0001 Akteure und OAuth2-Rollen.....	13
1340	Tabelle 2: TAB_IDP_DIENST_0002 Kurzbezeichnung der Schnittstellen des IdP-Dienstes	
1341	14
1342	Tabelle 3: TAB_IDP_DIENST_0003 Bezeichnungen der Schlüssel und deren URI	14
1343	Tabelle 4: TAB_IDP_DIENST_0004 Schema der Fehlermeldungen	24
1344	Tabelle 5: TAB_IDP_DIENST_0005 Befüllung der Attribute "given_name", "family_name",	
1345	"organizationName", "professionOID" und "idNummer"	38
1346	Tabelle 1: TAB IDP DIENST 0001 Akteure und OAuth2-Rollen.....	13
1347	Tabelle 2: TAB IDP DIENST 0002 Kurzbezeichnung der Schnittstellen des IdP-Dienstes	
1348	14
1349	Tabelle 3: TAB IDP DIENST 0003 Bezeichnungen der Schlüssel und deren URI	14
1350	Tabelle 4: TAB IDP DIENST 0004 Schema der Fehlermeldungen	24
1351	Tabelle 5: TAB IDP DIENST 0005 Befüllung der Attribute "given_name", "family_name",	
1352	"organizationName", "professionOID" und "idNummer"	38
1353		

1354 6.5 Referenzierte Dokumente

1355 6.5.1 Dokumente der gematik

1356 Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument
 1357 referenzierten Dokumente der gematik zur Telematikinfrastruktur. Der mit der
 1358 vorliegenden Version korrelierende Entwicklungsstand dieser Konzepte und
 1359 Spezifikationen wird pro Release in einer Dokumentenlandkarte definiert; Version und
 1360 Stand der referenzierten Dokumente sind daher in der nachfolgenden Tabelle nicht
 1361 aufgeführt. Deren zu diesem Dokument jeweils gültige Versionsnummern sind in der
 1362 aktuellen, von der gematik veröffentlichten Dokumentenlandkarte enthalten, in der die
 1363 vorliegende Version aufgeführt wird.

1364

[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

1365

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

1366