

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

20

21
22
23
24
25
26
27
28

29
30

Telematikinfrastruktur 2.0

Feature: Zero Trust

Version:	1.0.1_CC
Revision:	925986
Stand:	07.06.2024
Status:	zur Abstimmung freigegeben
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemF_Zero-Trust

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_CC	15.04.2024		initiale Erstellung	gematik
1.0.1_CC	07.06.2024		ZT für LE-Zugang hinzugefügt	gematik

Inhaltsverzeichnis

39	1 Einordnung des Dokuments	6
40	1.1 Zielsetzung	6
41	1.2 Zielgruppe	6
42	1.3 Abgrenzungen	6
43	1.4 Methodik	7
44	1.4.1 Anforderungen.....	7
45	2 Features und Epics.....	8
46	2.1 Clientregistrierung	8
47	2.1.1 Wiedererkennung bekannter Clients.....	8
48	2.1.2 Device Security Rating.....	8
49	2.2 Policy Enforcement.....	8
50	2.2.1 Zugriffsschutz.....	8
51	2.2.2 http Proxy.....	9
52	2.3 Decide from Policies	9
53	2.3.1 Maschinenlesbare Zugriffsregeln.....	9
54	2.3.2 Ein reproduzierbares Ja/Nein/Vielleicht.....	9
55	2.3.3 Policies nach Betroffenheit.....	9
56	2.4 Policy Information und Administration.....	9
57	2.4.1 Policy Verwaltung.....	9
58	2.4.2 Monitoring	10
59	2.5 Client Authorization.....	10
60	2.5.1 Autorisierung auf Basis von Policy Entscheidungen.....	10
61	2.5.2 Client Authentication	10
62	3 Einordnung in die TI 2.0	11
63	4 Technisches Konzept	14
64	4.1 Zero Trust Cluster.....	14
65	4.2 Policy Enforcement Point (PEP).....	14
66	4.3 Policy Decision Point (PDP)	15
67	4.4 Trust Client.....	15
68	4.5 Policy Information und -Administration	16
69	4.5.1 Policy Information Point (PIP)	16
70	4.5.2 Policy Administration Point (PAP).....	16
71	4.6 Clientregistrierung	17
72	4.7 Monitoring	20
73	4.7.1 Security Information and Event Management (SIEM):	20
74	4.7.2 Shared Signals	20
75	4.7.3 Telemetrie, Monitoring und Logging	21
76	4.8 Zusammenspiel mit IdP	21

77	4.9 Fachdienst-Backend	22
78	5 Spezifikation	23
79	5.1 Übergreifende Anforderungen für Datenschutz und Sicherheit.....	23
80	5.1.1 Sicherheits- und Datenschutzanforderungen an Logging und Monitoring	24
81	5.1.2 Sicherheits- und Datenschutz-Anforderungen an das Security Monitoring	25
82	5.1.3 Sicherheits- und Datenschutz-Anforderungen an die Verarbeitung von Daten	
83	mit dem Schutzbedarf "sehr hoch"	27
84	5.1.4 Sicherheits- und Datenschutz Anforderungen an dem Trust Client	28
85	5.2 Anforderungen an Clientsysteme.....	28
86	5.2.1 Hersteller und Herausgeber	28
87	5.2.2 Verbindungsaufbau	29
88	5.2.3 Clientregistrierung	30
89	5.2.4 Nutzerauthentifizierung.....	31
90	5.2.5 Session Management.....	31
91	5.3 Zero Trust Cluster.....	32
92	5.4 Anforderungen an Policy Enforcement Points	33
93	5.4.1 PEP Client Registry	33
94	5.4.1.1 Sicherheits- und Datenschutz-Anforderungen an dem PEP Client	
95	Registration.....	35
96	5.4.2 PEP Relying Party.....	35
97	5.4.3 PEP Authorization Server.....	36
98	5.4.3.1 Service Discovery.....	37
99	5.4.3.2 Ablauf der SM-B Authentifizierung mit DPoP.....	39
100	5.4.4 PEP http Proxy.....	43
101	5.4.5 Sicherheits- und Datenschutz-Anforderungen an dem PEP.....	44
102	5.4.6 Konfiguration	44
103	5.5 Anforderungen an den Policy Decision Point	45
104	5.6 Anforderungen an den PIP und PAP Service	48
105	5.7 Anforderungen an den Betrieb der Zero Trust Komponenten.....	50
106	5.7.1 Anforderungen für nahtlose Aktualisierungen	50
107	5.7.2 Anforderungen für Steuerung durch Feature-Flags	51
108	5.7.3 Anforderungen zur Überwachung des Betriebsstatus	51
109	5.7.4 Betriebliche Schnittstellendefinition der Zero Trust-Komponenten	52
110	5.8 Anforderungen an den Test der Zero-Trust Komponenten	53
111	5.8.1 Testartefakte.....	53
112	5.8.2 Testtreiberschnittstelle und Testunterstützung	54
113	5.8.3 Bereitstellung der Testkomponenten und Testartefakte.....	55
114	5.8.4 Testumgebungen und Quality Gates.....	55
115	6 Dokumentenhaushalt.....	56
116	6.1 Neue Dokumente	56
117	6.2 Übersicht betroffener Dokumente	56
118	6.3 Übersicht Produkt- und Anbietertypen	56
119	7 Beispiele und Referenzimplementierungen	57
120	8 Anhang A – Verzeichnisse	58

121	8.1 Abkürzungen	58
122	8.2 Abbildungsverzeichnis.....	58
123	8.3 Tabellenverzeichnis	59
124	8.4 Referenzierte Dokumente.....	59
125	8.4.1 Dokumente der gematik.....	59
126	8.4.2 Weitere Referenzen	60
127		
128		

129

1 Einordnung des Dokuments

130 Dieses Dokument stellt eine übergreifende Spezifikation dar, ohne einen ersten
131 konkreten Bezug zu einem Produkttypen oder zu Schnittstellen herzustellen.
132 Anforderungen dieses Dokuments werden Produkttypen, Schnittstellen, Komponenten
133 oder Diensten von konkreten Use Cases bzw. von Fachanwendungen zugewiesen.

134 Die in diesem Dokument beschriebenen Konzepte, Abläufe und Informationsmodelle
135 dienen der Umsetzung der Paradigmen des Zero Trust in der "Telematikinfrastruktur
136 2.0".

137 Das Zero-Trust-Modell ist ein Sicherheitskonzept, das auf dem Prinzip strenger
138 Zugriffskontrollen und dem grundsätzlichen Misstrauen (kein implizites Vertrauen)
139 gegenüber jedem Kommunikationsteilnehmer beruht, selbst denen, die sich bereits
140 innerhalb eines Netzwerkperimeters befinden. Es handelt sich um ein
141 Sicherheitsrahmenwerk, das erfordert, dass alle Benutzer und deren Clients (Gerät und
142 App), sowohl innerhalb als auch außerhalb der Netzwerkperimeter, authentifiziert,
143 autorisiert und kontinuierlich auf ihre Sicherheitskonfiguration und Sicherheitsnachweise
144 überprüft werden, bevor ihnen Zugriff auf Anwendungen und Daten gewährt oder dieser
145 aufrechterhalten wird. Motiviert durch den „Assume Breach“-Ansatz basiert dieses
146 Architekturdesign-Paradigma im Kern auf dem Prinzip der minimalen Rechte aller
147 Entitäten in der Gesamtinfrastruktur.

1.1 Zielsetzung

149 Ziel des Dokuments ist die Sammlung der technischen, betrieblichen und testrelevanten
150 Anforderungen an Clients, Backendservices, Produkttypen, Komponenten und Dienste,
151 die sich untereinander über das Internet vernetzen, im Gegensatz zur bestehenden TI als
152 geschlossenes VPN. Dieses Pattern wird bisweilen auch als TI 2.0 bezeichnet.

1.2 Zielgruppe

154 Dieses Dokument richtet sich an Architekten und Entwickler von Komponenten, Diensten,
155 Produkttypen, Schnittstellen und Clients für den Datenaustausch im deutschen
156 Gesundheitswesen.

1.3 Abgrenzungen

158 Diesem Dokument ist kein Produkt- oder Anbietertyp zuzuordnen. Anforderungen in
159 diesem Dokument finden Anwendung in Produkt- und Anbietertypen von konkreten
160 Fachanwendungen bzw. Use Cases.

161 1.4 Methodik

162 1.4.1 Anforderungen

163 Anforderungen als Ausdruck normativer Festlegungen werden durch eine eindeutige ID
164 sowie die dem [RFC2119] entsprechenden, in Großbuchstaben geschriebenen deutschen
165 Schlüsselworten MUSS, DARF NICHT, SOLL, SOLL NICHT, KANN gekennzeichnet.

166 Da in dem Beispielsatz „Eine leere Liste DARF NICHT ein Element besitzen.“ die Phrase
167 „DARF NICHT“ semantisch irreführend wäre (wenn nicht ein, dann vielleicht zwei?), wird
168 in diesem Dokument stattdessen „Eine leere Liste DARF KEIN Element besitzen.“
169 verwendet. Die Schlüsselworte werden außerdem um Pronomen in Großbuchstaben
170 ergänzt, wenn dies den Sprachfluss verbessert oder die Semantik verdeutlicht.

171 Anforderungen werden im Dokument wie folgt dargestellt:

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

173 Text / Beschreibung

174 [**<=**]

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

177

2 Features und Epics

178 Der folgende Abschnitt gibt einen groben Überblick über die Features und Epics, die sich
179 in Anwendungen wiederfinden, wenn sie nach dem Paradigma des Zero Trust umgesetzt
180 werden. Diese Epics sind als Enabler zu verstehen, um Fachanwendungen einen sicheren
181 Verbindungsaufbau zwischen Clientsystemen und Backenddiensten zu ermöglichen. Es
182 werden keine User Stories formuliert, da für den Verbindungsaufbau keine
183 Nutzerinteraktion angedacht ist.

184 Im Rahmen der Nutzeridentifikation (Authentifizierung) findet eine Verifikation
185 ausgegebener Authentisierungsmerkmale statt, deren Nutzerinteraktion als Teil der
186 Spezifikation des Identity Managements beschrieben sind.

187 2.1 Clientregistrierung

188 Gemäß des Zero Trust Ansatzes ist jeder Schnittstellenaufruf potentiell gefährlich, soweit
189 nicht anders festgestellt. Dazu zählt auch das Vertrauen in bekannte bzw. Misstrauen in
190 unbekannte Geräte bzw. Clients. Um Geräte bzw. Clients wiedererkennbar zu machen,
191 soll eine Registrierung dieser erfolgen. Sind in der Registrierung zusätzliche
192 Sicherheitsmerkmale über das Gerät und den Aufrufkontext feststellbar, stärken diese
193 das Vertrauen in nachfolgenden Aufrufen fachlicher Schnittstellen.

194 2.1.1 Wiedererkennung bekannter Clients

195 Die Wiedererkennung bekannter Geräte und Clients und deren Bindung an
196 identifizierbare Nutzer des Gesundheitswesens muss über eine Registrierung erfolgen.
197 Die Identifikation des Nutzers erfolgt dabei über ein unterstütztes Identifikationsmerkmal
198 (SmartCard oder digitale Identität) und einen selbstgewählten, vom System
199 unterstützten zweiten Faktor (E-Mail, SMS, etc.).

200 2.1.2 Device Security Rating

201 Zum Einschluss bzw. Ausschluss bestimmter Eigenschaften von Geräten und Clients,
202 sollen selbige einer automatischen Sicherheitsprüfung unterzogen werden können (DSR -
203 Device Security Rating), soweit es die gegebenen Plattformmechanismen erlauben.

204 2.2 Policy Enforcement

205 Für den Zugriff auf personenbezogene und medizinische Daten und zur Sicherstellung der
206 Integrität, Verfügbarkeit, Vertraulichkeit und Authentizität transportierter Daten gelten
207 Regeln. Diese fachlichen, technischen und organisatorischen Regeln gelten bei jedem
208 Zugriff auf Daten, die über eine Schnittstelle zugreifbar gemacht werden.

209 2.2.1 Zugriffsschutz

210 Das Policy Enforcement soll als eine Art Gatekeeper bzw. Türsteher den Zugriff auf
211 Schnittstellen von Backendservices durch beliebige Clients durchsetzen. Grundlage ist

212 das Vertrauen in eine Policy-Entscheidung durch eine Komponente zur Auswertung eines
213 Regelwerks.

214 **2.2.2 http Proxy**

215 Nur und ausschließlich wenn ein Zugriffsversuch als legitim bewertet wird, soll dieser
216 Zugriff gewährt werden, im Sinne des oben genannten Gatekeepers bzw. Türstehers
217 muss dann auch ein Zugriffsversuch gewährt und an eine fachliche Schnittstelle
218 weitergereicht werden.

219 **2.3 Decide from Policies**

220 Die Menge an Regeln für die Gewähr eines Zugriffs auf Daten oder Schnittstellen speist
221 sich aus gesetzlichen Forderungen bzw. Verboten, Vertragskonstrukten,
222 Sicherheitsmechanismen, Architekturentscheidungen und Informationen aus der
223 "Umgebung" des Betriebs von Clients und Backendservices.

224 **2.3.1 Maschinenlesbare Zugriffsregeln**

225 Die Menge (potentiell) geltender Regeln zur Absicherung des Zugriffs auf Daten und
226 Dienste formt ein Set von Policies. Um im Fall eines Zugriffsversuchs schnell entscheiden
227 zu können, sollen diese Regeln maschinenlesbar definiert sein.

228 **2.3.2 Ein reproduzierbares Ja/Nein/Vielleicht**

229 Die Auswertung eines komplexen Regelwerks liefert bei identischen Eingangsparametern
230 reproduzierbar das identische Ergebnis.

231 **2.3.3 Policies nach Betroffenheit**

232 Regeln beziehen sich auf verschiedene Aspekte einer Zugriffsentscheidung. Es gelten
233 fachliche Regeln, Regeln zur Benutzung von Clients bzw. Geräten und ebenso technische
234 Regeln sowie solche, die Betriebsumgebung von Backenddiensten betreffend.

235 **2.4 Policy Information und Administration**

236 Regeln können sich ändern und Regeln beeinflussen Regeln.

237 **2.4.1 Policy Verwaltung**

238 Eine Policyentscheidung kann Eingangsinformation für andere Policies sein, ebenso kann
239 das Ändern von Rahmenbedingungen oder eine Anomalieerkennung zur Beeinflussung
240 von Policies führen. Aus diesem Grund führen Beobachtungen über Policyentscheidungen
241 zu Informationen über das Gesamtsystem, die als Eingangsdaten für nachfolgende
242 Policyentscheidungen herangezogen werden. Daneben ist es erforderlich, Anpassungen
243 am Regelwerk dem System über authentizitäts- und integritätsgeschützte Wege bekannt
244 zu machen.

245 **2.4.2 Monitoring**

246 Durch ein Monitoring von Betriebsparametern und Telemetriedaten wird die
247 Durchsetzung von Policies sowie die Auswirkung möglicher Policyänderungen
248 transparent.

249 **2.5 Client Authorization**

250 Menschen benutzen Clients (Kombination aus Gerät und App). Jeder Zugriff auf Daten
251 oder Schnittstellen wird auf eine menschliche Interaktion (Authentisierung)
252 zurückgeführt. Nach Stand der Technik erfolgt die sichere Authentifizierung meist über 2
253 Faktoren. Zur Wiedererkennung und sicheren Identifikation werden Menschen und Clients
254 Authentifizierungsmerkmale ausgestellt. Die sichere Identifikation und Authentifizierung
255 ist eine wichtige Eingangsgröße für Zugriffsentscheidungen (s. o.).

256 **2.5.1 Autorisierung auf Basis von Policy Entscheidungen**

257 Die Autorisierung von Zugriffen auf Daten oder Schnittstellen wird bei positiver
258 Entscheidung durch ein Set von Policies gewährt. Die Zugriffsentscheidung und -gewähr
259 bettet sich in eine Verkettung von Informationen und von Aufrufen verschiedener
260 Schnittstellen ein, die dem fachlichen Aufruf einer Schnittstelle bzw. Abruf von Daten
261 voranstehen. Stand der Technik dieses Flows mehrerer Aufrufe und der dabei
262 transportierten Informationen ist der OAuth2-Standard, vgl. [RFC6749 et al.].

263 **2.5.2 Client Authentication**

264 Menschen und Clients werden anhand sicherer Merkmale authentifiziert, die Identifikation
265 ist nachrangig bzw. in nachgelagerten fachlichen Anwendungsfällen bzw. in fachlichen
266 Zugriffsregeln relevant.

267 Kann ein Mensch oder Client nicht sicher authentifiziert werden oder wird der
268 Authentifizierung zeitlich oder anderweitig nicht vertraut oder passen die Umgebungs-
269 bzw. die den Aufruf begleitenden Parameter nicht zum Vertrauen in die Authentifizierung,
270 wird eine erneute Authentifizierung als erforderlich angesehen ("Step-Up-
271 Authentication").

3 Einordnung in die TI 2.0

Die TI 1.0 bildet eine Infrastruktur, deren Sicherheit auf der sicheren Zugangskontrolle zu einem geschlossenen zentralen Netzwerk mit Diensten beruht. In der TI 2.0 werden die Dienste direkt im Internet angeboten und bedürfen daher einem Schutz vor unberechtigtem Zugriff pro Dienst. Dieser Schutz wird nach dem Zero Trust Paradigma durch den Policy Enforcement Point und den Policy Decision Point durchgesetzt.

Diese übergreifende Spezifikation richtet Anforderungen an Akteure, die sich über das Internet miteinander vernetzen. Diese Akteure seien im Folgenden einerseits Clients (Software: Aufrufende einer Schnittstelle, Anfragende an einen Datenabruf oder -zugriff, wird auf einem bestimmten Gerät ausgeführt), häufig bedient durch einen Menschen, und Backendservices (Software: bereitstellende Schnittstelle, Datenbereitstellung etc.) auf der anderen Seite.

Zur Absicherung dieser Clients und Backendservices werden zum einen Anforderungen erhoben und wird eine Empfehlung gegeben, diese Anforderungen in konkreten Softwarekomponenten innerhalb dieser Akteure umzusetzen. Die Empfehlung zur Separierung der Zero Trust Mechanismen in unterschiedliche Komponenten folgt der Zero Trust NIST Referenzarchitektur, welche im Feinkonzept [gemKPT_Zero_Trust] vorgeschlagen und für passend befunden wurde.

Figure 4-1 General ZTA Reference Architecture

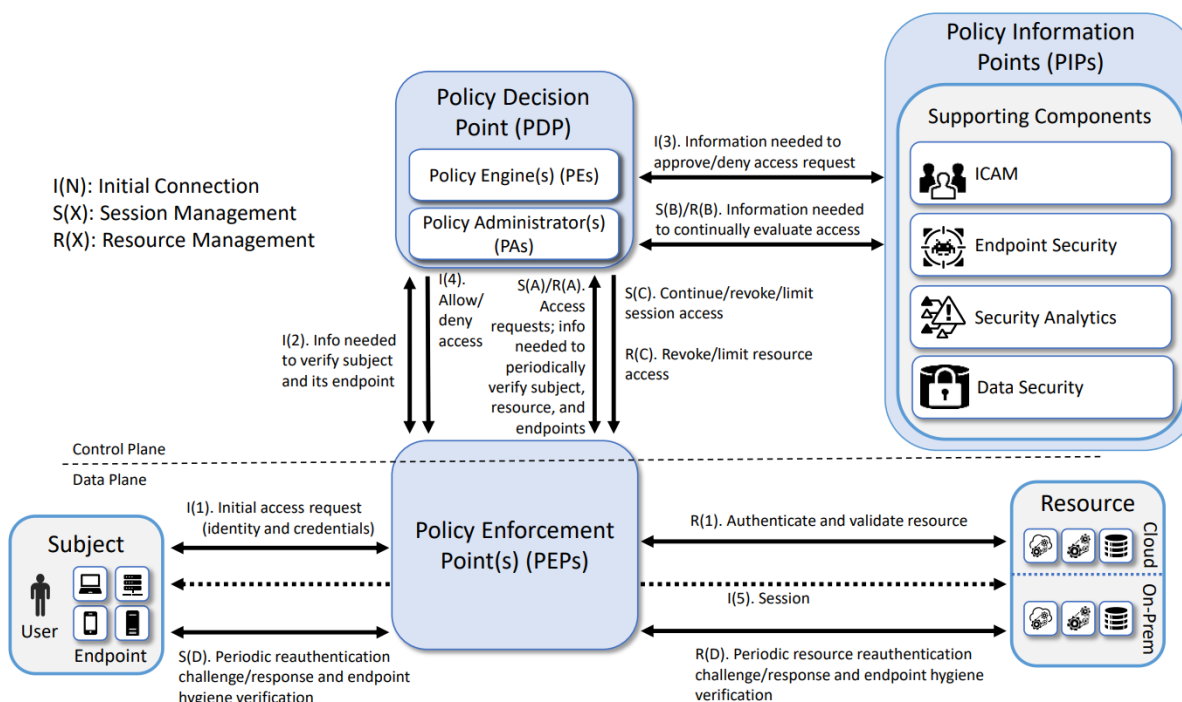
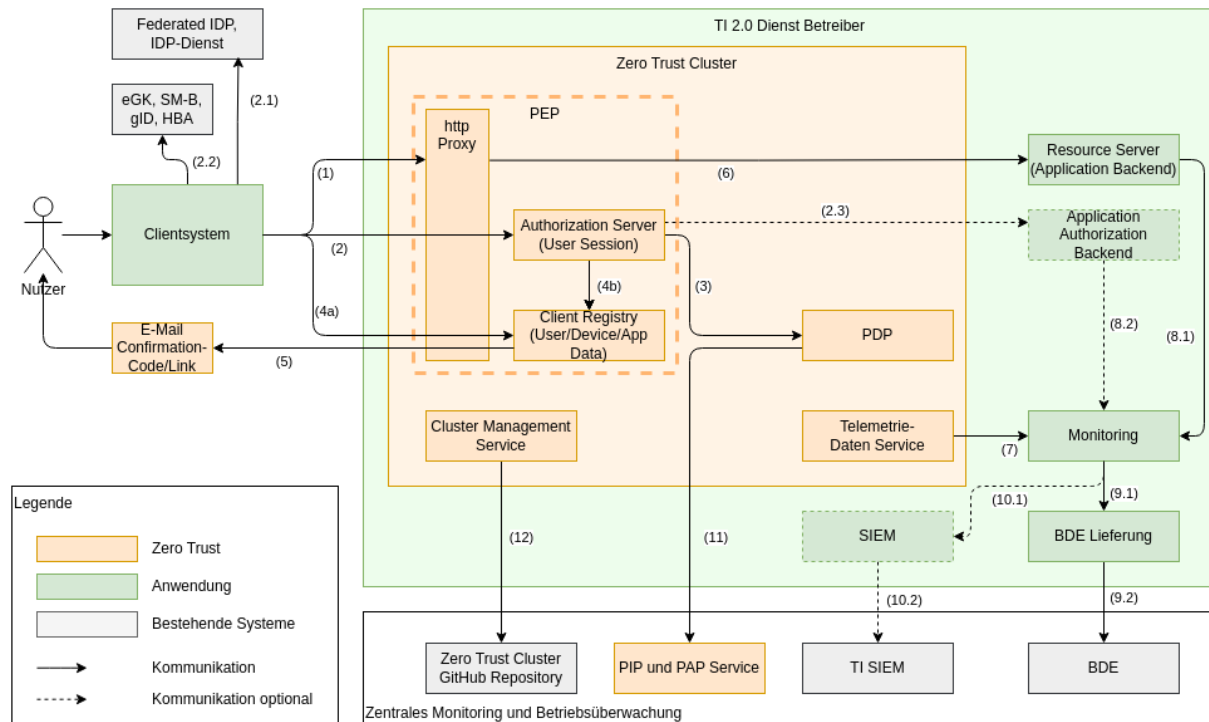


Abbildung 1: NIST Zero Trust Referenzarchitektur

Im Architekturkonzept der TI 1.0 werden konkrete Umgebungsannahmen zu Consumer Zonen, Secure Consumer Zonen, Plattformzonen, Personal Zonen usw. getroffen, in denen kein (Personal Zone) bzw. ein gewisses Sicherheitsniveau (überall sonst) axiomatisch angenommen wird. Das Zero Trust Konzept löst sich von der Aufteilung in verschiedene Zonen, insbesondere, da weniger (teilweise gar keine mehr) TI-Plattform-Produkttypen zwischen den Datenaustauschen unter Clients mit Backendservices

involviert werden. Im Folgenden ist eine Produkttypzerlegung für die Umsetzung der NIST-Referenzarchitektur einer generischen Fachanwendung dargestellt.

In diesem Pattern greift ein Nutzer über ein Clientsystem auf Daten eines TI 2.0 Dienstes zu. Das folgende Bild zeigt eine Übersicht der beteiligten Komponenten in der Vernetzung zwischen einem Clientsystem (links grün) und einem Backendservice (rechts grün: Ressource Server).



Die obige Abbildung zeigt die Einbettung von Zero Trust bezogenen, logischen Komponenten (orange) in die Aufrufkette zwischen einem Client und einem Ressource Server (grün). Dargestellt sind zusätzlich heute bereits vorhandene und genutzte Komponenten und Dienste, die für die Nutzerauthentifizierung (z. B. eGK und IDP) bzw. die Betriebsüberwachung (z. B. mittels Betriebsdatenerfassung - kurz BDE) in Anwendungsfällen der TI 2.0 weitergenutzt werden können (grau). In diese Abbildung sind diverse Architekturentscheidungen eingearbeitet, die im Kapitel 4 und 5 erläutert bzw. spezifiziert werden.

Kurzbeschreibung der Komponenten und Schnittstellen

(1) und (6): Der http Proxy erlaubt den Zugriff auf Daten des Resource Servers, wenn ein gültiges Access token im Authorization Header enthalten ist.

(2), (2.1) und (2.2): sowie optional (2.3): Um ein Access token vom Authorization Server zu erhalten, ist eine Authentifizierung des Nutzers erforderlich.

(3): Der Authorization Server stellt nur ein Access token aus, wenn der PDP seine Erlaubnis gegeben hat.

(4a) und (4b): Der Authorization Server fordert bei mobilen Apps zusätzlich zur Nutzer-Authentifizierung eine Client Registrierung mit Geräte/App-Attestierung ein, bevor ein Access token ausgestellt wird. Bei stationären Clientsystemen (z. B. Primärsystem) erfolgt die Geräteregistrierung implizit bei der SM-B Authentisierung.

- 325 (5): Die Client Registrierung erfordert eine Bestätigung des Nutzers durch einen zweiten
326 Faktor.
- 327 (7): Die Telemetrie-Daten der Komponenten des ZT Clusters werden an das Monitoring
328 System des Betreibers übergeben.
- 329 (8.1) und (8.2): Der Resource Server und der optionale Application Authorization Server
330 werden ebenfalls vom Monitoring überwacht.
- 331 (j9.1) und (9.2): Aus den Monitoring Daten werden die Daten der Betriebs-Daten-
332 Erfassung gebildet und versendet.
- 333 (10.1) und (10.2): Wenn der Betreiber ein SIEM einsetzt, werden aus dem Monitoring
334 SIEM Daten ermittelt und optional an das zentrale SIEM der gematik gesendet.
- 335 (11): Der PDP fragt regelmäßig den PIP und PAP Service nach neuen Policies und Daten
336 ab.
- 337 (12): Der Cluster Management Service überwacht die Clusterkonfiguration und setzt
338 durch, dass die im GitHub Repository gespeicherte Konfiguration ausgeführt wird.

4 Technisches Konzept

Im Kapitel zuvor wurden zwei Abbildungen vorgestellt, welche technischen Zero Trust-Komponenten (orange) an der Umsetzung fachlicher Anwendungsfälle von Clients, Komponenten und Backendservices von Fachanwendungen (grün) beteiligt sind. Im Folgenden werden diese technischen Komponenten genauer beschrieben und eingeordnet, welche Rolle sie in einer Architektur nach dem Zero Trust-Paradigma einnehmen.

Zero Trust in der TI zeichnet sich über folgende Eigenschaften aus:

- Registrierung des Clients (Gerät und App) zu einer Identität
- Attestation der Client-Eigenschaften
- Bereitstellung einer von Maschinen interpretierbaren Policy durch die gematik
- Einheitliches Durchsetzen der Policy durch die Fachdienste
- Sicherstellung des Sicherheitszustands der gesamten TI, Anbieter übergreifend
- Telemetrie und Monitoring

4.1 Zero Trust Cluster

Der Zero Trust Cluster (ZT Cluster) besteht aus Policy Enforcement Point (PEP), Policy Decision Point (PDP) sowie betriebsunterstützenden Komponenten (Cluster Management Service und Telemetrie Daten Service). Der PEP ist aufgeteilt in http Proxy, Authorization Server und Client Registry. Jeder TI 2.0 Dienst hat einen ZT Cluster zum Schutz des Dienstes vor unberechtigtem Zugriff. Der ZT Cluster wird in der Verantwortung des TI 2.0 Dienst-Betreibers betrieben.

4.2 Policy Enforcement Point (PEP)

Ein Policy Enforcement Point (PEP) ist eine Schlüsselkomponente im Zero Trust-Paradigma, der darauf abzielt, dass Sicherheitsmodell von einem vertrauensbasierten auf ein verifizierungsbasiertes umzustellen. Der PEP dient dazu, den Zugriff auf Ressourcen, basierend auf vordefinierten Richtlinien, zu kontrollieren und durchzusetzen. Im Kontext der TI 2.0 übernimmt der PEP folgende Funktionen:

- Der PEP agiert als http Proxy, der den Datenverkehr zwischen Clientanwendungen und den zu schützenden Ressourcen kontrolliert. Dadurch kann der PEP den gesamten Datenverkehr überwachen und filtern, um sicherzustellen, dass er den festgelegten Sicherheitsrichtlinien entspricht.
- Der PEP ist als vertrauenswürdige Relying Party im föderierten Identitätsmanagement registriert. Dadurch kann der PEP Identitätsinformationen von Benutzern sicher und vertrauenswürdig beziehen und bei Bedarf eine (erneute) Nutzer-Authentifizierung an die IDPs delegieren. Dadurch stellt der PEP sicher, dass nur authentifizierte Benutzer Zugriff auf die geschützten Ressourcen erhalten.
- Der PEP fungiert als OAuth2 Authorization Server und verwaltet die Autorisierung von Benutzeranfragen auf geschützte Ressourcen. Zudem überwacht der PEP die

Benutzersessions, um sicherzustellen, dass sie gültig sind und den Sicherheitsrichtlinien entsprechen.

- Der PEP ermöglicht die dynamische Registrierung von Clients, die auf geschützte Ressourcen zugreifen möchten. Dies umfasst auch die Offband-Bestätigung, bei der zusätzliche Sicherheitsmechanismen (Verifikation via E-Mail oder SMS) verwendet werden, um die Identität und Integrität (plattformabhängig) der registrierten Clients zu überprüfen.

Insgesamt agiert der PEP als Kontrollpunkt in der Zero Trust Architektur, der sicherstellt, dass nur autorisierte Benutzer und Geräte Zugriff auf die Ressourcen eines Dienstes erhalten und dass dabei die definierten Sicherheitspolicies eingehalten werden. Die Entscheidung zwischen verschiedenen Policies auf Basis der vom Client übergebenen Signale, Sicherheitsnachweise und Token trifft der Policy Decision Point.

In der TI 2.0 ist der PEP Teil der Betriebsumgebung eines Fachdienstes einer Fachanwendung.

4.3 Policy Decision Point (PDP)

Ein Policy Decision Point (PDP) ist die wesentliche Komponente im Zero Trust-Paradigma, die Zugriffsentscheidungen auf Plattformebene trifft, indem sie Richtlinien (Policies) interpretiert und anhand dieser Richtlinien Zugriffsanfragen bewertet. Folgende Funktionen eines PDP sind dabei zentral:

- Der PDP analysiert und interpretiert die Sicherheitsrichtlinien, die im Rahmen des Zero Trust-Modells definiert sind. Diese Policies können Kriterien enthalten wie Benutzeridentität, Gerätetyp, Standort, Zeitpunkt der Anfrage und andere Kontextinformationen ("Signale"), die relevant für die Zugriffsentscheidung sind.
- Basierend auf der Interpretation der Policies trifft der PDP Entscheidungen darüber, ob eine Zugriffsanfrage auf eine bestimmte Ressource genehmigt oder abgelehnt wird. Diese Entscheidungen erfolgen auf Plattformebene, was bedeutet, dass der PDP die Zugriffsanfragen im Kontext der gesamten Plattform oder des Netzwerks bewertet, und nicht isoliert betrachtet. Die Zugriffsentscheidung resultiert dann in der Ausstellung eines Access Tokens, das für den konkret angefragten Zugriff verwendet wird (siehe Policy Enforcement).
- Der PDP verwendet dabei die Informationen, die ihm übermittelt werden, um die Zugriffsentscheidung zu treffen. Dazu gehören nicht nur die Policies selbst, sondern auch Echtzeitinformationen über den Zustand von Benutzeridentitäten, Geräten und andere Kontextinformationen, die für die Bewertung der Zugriffsanfrage relevant sind.

Durch die Analyse von Policies und die Bewertung von Zugriffsanfragen auf Plattformebene trägt der PDP dazu bei, sicherzustellen, dass nur autorisierte Benutzer und Geräte Zugriff auf geschützte Ressourcen erhalten.

4.4 Trust Client

Im Kontext von Zero Trust stellt der "Trust Client" eine logische Komponente innerhalb einer Clientanwendung (Primärsystem (PS), Frontend des Versicherten (FdV) etc.) dar, die im Rahmen des Zero Trust-Paradigmas als vertrauenswürdig eingestuft wird. Dies steht im Gegensatz zu der traditionellen Annahme, dass ein, eine Schnittstelle

421 aufrufendes, Clientsystem automatisch als vertrauenswürdig betrachtet wird. Das
422 Vertrauen in Clientanwendungen erwächst beispielsweise durch regelmäßige
423 Softwareupdates, authentische und verschlüsselte Kommunikation und die Verwendung
424 von Zertifikaten, die die Einhaltung dieser Maßnahmen beweisen.

425 Ein Trust Client im Zero Trust-Modell wird nicht mehr blind als vertrauenswürdig
426 angesehen, sondern muss genauso wie alle Komponenten im Netzwerk kontinuierlich
427 authentifiziert und autorisiert werden. Selbst wenn ein Endpunkt als Trust Client
428 eingestuft ist, bedeutet dies nicht, dass er ungehinderten Zugriff auf alle Ressourcen im
429 Netzwerk hat. Stattdessen werden Zugriffsentscheidungen basierend auf aktuellen
430 Richtlinien, Kontextinformationen, Bedrohungsinformationen und insbesondere in
431 Kenntnis des diesen Client benutzenden Benutzers getroffen (s. u. Zusammenspiel mit
432 Identity Provider (IdP)).

433 **4.5 Policy Information und -Administration**

434 Im Zero Trust-Paradigma spielen der Policy Information Point (PIP) und der Policy
435 Administration Point (PAP) wichtige Rollen bei der Verwaltung und Durchsetzung von
436 Sicherheitsrichtlinien bzw. Policies. Zusammen ermöglichen der PIP und der PAP eine
437 zentrale Verwaltung und Bereitstellung von Policies im Zero Trust-Netzwerk.

438 Der PAP stellt Policies bereit und der PIP stellt die Daten für die Policies bereit, sodass
439 sich aus beiden ein Regelwerk ergibt, das der PDP anwendet, um zu entscheiden, ob eine
440 Kommunikationsanfrage zulässig ist.

441 **4.5.1 Policy Information Point (PIP)**

442 Der PIP ist für die Bereitstellung von Informationen über Sicherheitsrichtlinien
443 zuständig. Er dient als zentraler Informationsdienst, der anderen Systemen und
444 Komponenten im Zero Trust-Netzwerk Zugriff auf aktuelle Sicherheitsrichtlinien
445 ermöglicht. Der PIP kann Attribute wie Benutzerrollen, Zugriffsrechte, Gerätezustände
446 und andere Kontextinformationen bereitstellen, die von anderen Komponenten für die
447 Zugriffsentscheidung benötigt werden. Der PIP kann Daten aus verschiedenen Quellen
448 beziehen, einschließlich einer zentralen Richtliniendatenbank, externen
449 Identitätsanbietern, Sicherheitsinformationen von Geräten und anderen Quellen.

450

451 **4.5.2 Policy Administration Point (PAP)**

452 Der PAP ist für die Verwaltung und Konfiguration von Sicherheitsrichtlinien
453 verantwortlich. Er bietet eine Schnittstelle oder eine Konsole, über die Richtlinien in
454 hoheitlicher Verantwortung definiert, geändert und gelöscht werden können. Policy-
455 Administratoren können im PAP Zugriffsregeln, Autorisierungsniveaus,
456 Bedrohungsabwehrmaßnahmen und andere Sicherheitsrichtlinien festlegen. Der PAP
457 ermöglicht es Policy-Administratoren, Richtlinien - basierend auf verschiedenen Kriterien
458 wie Benutzerrollen, Gruppenzugehörigkeit, Standorten und Geräteattributen - zu
459 differenzieren. Änderungen an den Sicherheitsrichtlinien, die im PAP vorgenommen
460 werden, werden an den PIP weitergegeben, damit andere Komponenten im Zero Trust-
461 Netzwerk auf die aktualisierten Richtlinien zugreifen können. Das Vertrauen in
462 bereitgestellte und angepasste Policies wird über Signaturen für die Sicherstellung von
463 Integrität und Authentizität jeder Policy sichergestellt.

4.6 Clientregistrierung

Alle Clients, die mit Diensten der TI2.0 kommunizieren, sind zur Laufzeit bekannt. Mit einer Attestierung in Abhängigkeit der verfügbaren Mechanismen der Laufzeitumgebung (Geräte-Features, Betriebssystem) kann ein Vertrauen und eine Wiedererkennung von Clients und Geräten aufgebaut werden.

Die folgenden statischen Eigenschaften werden im Rahmen der Bereitstellung von Clientanwendungen erfasst und sind unabhängig von der Nutzung durch einen konkreten Benutzer.

Tabelle 1: Statische Eigenschaften Clientsysteme auf Hersteller-/Herausgeber-/Anbieterebene

Client Eigenschaft	Beschreibung
Produkt-Id	Eindeutige ID der Client-Software,, vergeben durch gematik über einen ITSM Prozess
Produkt-Name	Produkt-Name, vergeben durch den Hersteller
Hersteller-Id	Kennung des Herstellers aus TI-ITSM
Hersteller-Name	Name des Herstellers aus TI-ITSM
Produkt-Plattform	<p>Zunächst werden zwei Plattformen gesondert behandelt: Android und Apple (iOS, macOS etc). Diese beide Plattformen bieten Mechanismen für die Attestation der Client-Instanzen und der Umgebung, in welcher diese Clients ausgeführt werden.</p> <p>Alle andere Clients werden zunächst als generische Software-Produkte eingestuft.</p>
Produkt-Plattform-Id	<p>Plattformspezifisch eindeutige Kennung der Client-Software</p> <p>Android: Package-Name und Signer-Zertifikat Fingerprint Apple: Bundle-ID und Apple-ID Software: Registriert durch gematik , analog zu ClientIds in E-Rezept</p>
Attestation-Methode	<p>Die Methode, nach welcher die Client-Software und die Ablaufumgebung attestiert werden kann.</p> <p>Zunächst werden Android und Apple unterstützt, weil diese Plattformen entsprechende Mechanismen zur Remote-Attestation anbieten.</p> <p>Perspektivisch ist es geplant, weitere Plattformen zur Attestation der Clients einzuführen, z. B. über TPM 2.0 auf Windows und Linux. Nicht attestierbare Software-Clients müssen zunächst einen weiteren Faktor verwenden, z. B. SM-B.</p>

Bei der Registrierung werden die statischen Eigenschaften eines Client-Systems für jede Client-Instanz mit einem vom Client-Nutzer signierten Softwarestatement bekannt gemacht. Durch die Registrierung bekommen die Clients eine kryptographische Identität und werden Server-seitig an den Nutzer gebunden. Der Nutzer muss bei der Registrierung eine TI-Identität vorweisen, z. B. GesundheitsID oder SM-B (über IDP-Dienst der gematik).

Zur Laufzeit werden die Client-Eigenschaften durch Client-Instanz-Eigenschaften ergänzt. Sie sind spezifisch für eine konkrete Installation auf einem bestimmten Gerät eines Benutzers. Sie sind insofern dynamisch, als dass sich der Patchlevel des Betriebssystems oder sich die Version der Clientinstanz durch Updates verändern kann.

Tabelle 2 : Eigenschaften Clientsysteme auf Instanzebene (pro Installation)

Client-Instanz-Eigenschaften	Beschreibung
Produkt-Version	Aktuelle Version der Client-Software
Client-Nutzer (Owner)	Informationen über den Client-Nutzer
Client-Eigenschaften (Posture)	Aktuelle Eigenschaften der Ablaufumgebung des Clients, insbesondere: <ul style="list-style-type: none">• Betriebssystem• Betriebssystem-Version
Client-Attestation	Falls die Plattform die Attestation des Clients ermöglicht, wird hier plattformspezifische Attestation angegeben. <ul style="list-style-type: none">• Android: Key ID Attestation• Apple: DCAppAttest• Software: keine Attestation, nur Nutzer-Bindung

Die Auskünfte bzw. Attestation von Clientsoftware und Geräten werden von einer Backendschnittstelle für die Clientregistrierung geprüft. Zusätzlich wird über diese Schnittstelle eine Offband-Verifikation des Benutzers durchgeführt, beispielsweise über Bestätigungscode oder -link via E-Mail. Ist die Clientregistrierung erfolgreich, wird der Client-Instanz(!) ein Nachweis über die attestierten Client- und Client-Instanz-Eigenschaften ausgestellt. Dieser Nachweis ist in folgenden Aufrufen von Schnittstellen der TI Teil der Zugangsprüfung.

Die Geräteattribute werden von den Plattformen der Endgeräte geliefert. Ihre Erhebung erfolgt im TrustClient des Endgeräts mittels plattformspezifischer Attestierungs- und Erhebungsmechanismen (siehe [Apple Platform Security Guide] und [Android Platform Security Model]). Die Attribute sind daher für die jeweilige Plattform und ihr Sicherheitsmodell spezifisch. Die für die Zugriffsentscheidung verwendeten Attribute werden daher im Folgenden für iOS-Geräte separat von denen für Android-Geräte aufgeführt.

Android**Tabelle 3: Verwendete Device Claims für Android-Geräte**

Attribute	Beschreibung
aktuelle Android Version	aktuell auf dem Gerät laufende Android Version bzw. API Level / SDK Version
Android Version bei Veröffentlichung	Android Version (API level) mit welchem das Gerät veröffentlicht / CTS durchlief
Patchlevel	Verschiedene Patch-Level-Angaben für OS & Co
FDE / FBE	Gibt an, ob Geräteverschlüsselung unterstützt wird und ob diese aktiviert ist.
System PIN / Password / Pattern gesetzt	Gibt an, ob ein PIN/Pattern/Passwort für den Sperrbildschirm gesetzt ist.
System PIN / Password / Pattern Qualität	Über den Device Policy Manger kann abgefragt werden, ob aktuell bestimmte Passwort-Komplexitätslevel erfüllt werden.
VerifiedBoot verfügbar	Gibt an, ob VerifiedBoot auf dem Gerät zur Verfügung steht (siehe [VerifiedBoot]).
Mainline Patchlevel	Gibt an, wann der letzte Mainline Patch installiert wurde.
Gerätehersteller / -modell	Gibt Informationen zu Hersteller, Model usw. zurück.
Biometric Class	Gibt Informationen zur Güte der vorhandenen biometrischen Sensoren zurück.

iOS**Tabelle 4: Verwendete Device Claims für iOS-Geräte**

Attribute	Description
System Name	Name des Betriebssystems auf dem Gerät
System Version	Version des Betriebssystems auf dem Gerät
Model	Art des Geräts, z. B. "iPhone" oder "iPod touch"

Attribute	Description
identifizierForVendor	Eindeutige Kennung des Geräts für den App-Anbieter
App Version	Version der App auf dem Gerät

4.7 Monitoring

Das Monitoring im Kontext von Zero Trust ist ein entscheidender Aspekt, um die Sicherheit des Netzwerks und der Ressourcen kontinuierlich zu überwachen und potenzielle Bedrohungen oder Anomalien zu identifizieren. Dieses bedient sich auch eines Security Information and Event Management(SIEM) und Shared Signals, die zukünftige Policyentscheidungen beeinflussen, in dem Erkenntnisse des Monitorings über den Policy Information Point den Policy Decision Points der verschiedenen Fachanwendungen verfügbar gemacht werden.

Insgesamt ermöglicht das Monitoring im Kontext von Zero Trust eine kontinuierliche Überwachung der Sicherheitslage, indem es aktuelle Sicherheitsrichtlinien berücksichtigt, potenzielle Bedrohungen identifiziert und auf Shared Signals zurückgreift, um umfassende Sicherheitseinblicke zu erhalten.

4.7.1 Security Information and Event Management (SIEM):

SIEM-Systeme spielen eine zentrale Rolle im Monitoring im Zero Trust-Paradigma. Sie sammeln Daten aus verschiedenen Quellen wie Protokollen, Ereignissen und Alarmen von Sicherheitskomponenten im Netzwerk. Durch die Analyse dieser Daten in Echtzeit können SIEM-Systeme potenzielle Sicherheitsvorfälle erkennen und Anomalien identifizieren. SIEM-Systeme können auf die vom PIP bereitgestellten Sicherheitsrichtlinien zugreifen und sicherstellen, dass die Überwachung entsprechend den aktuellen Richtlinien erfolgt. Anbieter von Betriebsleistungen mittels Produkttypen der TI (1.0) erhalten durch eine Anbieterzulassung die Auflage, Anforderungen an ein [ISMS] zu erfüllen. Hierfür können sie bspw. SIEM-Systeme oder Intrusion Detection Systeme (IDS) verwenden. In der Weiterentwicklung zur TI 2.0 wird dieses Konzept fortgeführt, und finden die so gesammelten Informationen über den Sicherheitszustand eines Systems wieder Eingang in Zugriffsentscheidungen eines Policy Decision Points.

4.7.2 Shared Signals

Shared Signals sind Hinweise oder Indikatoren für Sicherheitsvorfälle, die von verschiedenen Systemen und Quellen im Netzwerk gemeinsam genutzt werden. Diese Signale können von verschiedenen Sicherheitskomponenten wie Firewalls, Endpunktschutzsystemen, Intrusion Detection Systems (IDS) und anderen generiert werden.

SIEM-Systeme aggregieren und korrelieren diese Signale, um umfassende Einblicke in die Sicherheitslage des Netzwerks zu erhalten und potenzielle Bedrohungen zu identifizieren. Durch die Integration von Shared Signals in das Monitoring kann eine umfassende und ganzheitliche Sicherheitsüberwachung gewährleistet werden, die potenzielle Angriffe frühzeitig erkennt und darauf reagiert.

4.7.3 Telemetrie, Monitoring und Logging

Betriebliche Daten zum Zwecke des Monitorings (Telemetrie) werden von den Zero Trust Komponenten erhoben und für die eingesetzten Zero Trust Komponenten übergreifend in einer gesicherten Monitoring-Komponente erfasst. Bei der Nachbereitung der Telemetriedaten werden personenbezogene oder-beziehbare Daten anonymisiert, um diese bereinigten Daten dem Betreiber regelhaft zugänglich zu machen. Das bereinigte Monitoring-Log kann von dem Betreiber für sein eigenes betriebliches Monitoring und als Quelle für sein SIEM-System verwendet werden. Das bereinigte Monitoring-Log wird unter anderem zur Generierung von Rohdatenlieferungen und Bestandsdaten für die gematik benutzt.

4.8 Zusammenspiel mit IdP

Im Zero Trust-Paradigma arbeiten der PEP und der IdP zusammen, um den Zugriff auf Ressourcen - basierend auf den definierten Sicherheitsrichtlinien und der Benutzeridentität - zu kontrollieren. Das Stichwort "Step-up-Authentifizierung" bezieht sich auf eine Sicherheitsmaßnahme, bei der der Benutzer zusätzliche Authentifizierungsschritte durchlaufen muss, um auf sensible Ressourcen zuzugreifen. Diese Maßnahme wird wie folgt realisiert:

1. **Zugriffsanfrage des Benutzers:** Ein Benutzer möchte auf eine geschützte Ressource zugreifen und sendet eine Zugriffsanfrage an den PEP.
2. **Überprüfung durch den PEP:** Der PEP empfängt die Zugriffsanfrage und überprüft die http Header-Daten. Dies kann bedeuten, dass der PEP feststellt, dass die Zugriffsanfrage eine höhere Sicherheitsstufe erfordert als die Standardauthentifizierungsmethode des Benutzers. Oder der Authentifizierung wird nicht mehr vertraut, da sie zu weit in der Vergangenheit liegt.
3. **Weiterleitung an den IDP:** Falls eine Step-up-Authentifizierung erforderlich ist, löst der PEP die Zugriffsanfrage an den IDP aus, der für die Authentifizierung des Benutzers zuständig ist.
4. **Step-up-Authentifizierung:** Der IDP erkennt die Anforderung für eine Step-up-Authentifizierung und fordert den Benutzer auf, zusätzliche Authentifizierungsschritte durchzuführen. Dies könnte beispielsweise die Eingabe eines Einmalpassworts, die Verwendung von Biometrie oder die Bestätigung über ein zweites Gerät sein.
5. **Authentifizierungsbestätigung:** Nach erfolgreicher Durchführung der Step-up-Authentifizierung bestätigt der IDP die Identität des Benutzers gegenüber dem PEP.
6. **Zugriffsgewährung durch den PEP:** Der PEP erhält die Authentifizierungsbestätigung vom IDP und gewährt dem Benutzer basierend auf den Sicherheitsrichtlinien Zugriff auf die angeforderte Ressource.

Das Zusammenspiel zwischen PEP und IDP ermöglicht es, den Zugriff auf sensible Ressourcen - basierend auf der aktuellen Sicherheitslage und der Identität des Benutzers - zu steuern. Die Step-up-Authentifizierung stellt sicher, dass zusätzliche Sicherheitsmaßnahmen - wenn erforderlich - ergriffen werden, um die Integrität und Vertraulichkeit der geschützten Daten zu gewährleisten.

585 **4.9 Fachdienst-Backend**

586 Das Fachdienst-Backend stellt das Ziel jedes Zugriffswunschs eines Nutzers über sein
587 Clientsystem dar. Es stellt fachliche Schnittstellen zur Nutzung durch Clientsysteme dar,
588 die über die Mechanismen des Zero Trust abgesichert werden.

589

5 Spezifikation

590 Dieses Kapitel beschreibt die technische Umsetzung der beschriebenen Konzepte an die
591 oben eingeführten Komponenten des Zero Trust (Zero Trust-Komponenten) als
592 generische Produkt- und Anbietertypen. Diese Anforderungen finden Anwendung in den
593 Steckbriefen von konkreten Produkt- und Anbietertypen der jeweiligen Fachanwendung
594 und erhalten erst in der dortigen Zuordnung ein konkretes Prüfverfahren.

595 5.1 Übergreifende Anforderungen für Datenschutz und Sicherheit

596 **A_25400 - Zero Trust-Komponenten - Umsetzung Sicherer** 597 **Softwareentwicklungsprozess**

598 Der Hersteller einer Zero Trust-Komponente MUSS einen sicheren
599 Softwareentwicklungsprozess umsetzen (siehe [gemSpec_DS_Hersteller#Kapitel 2.2
600 Sicherer Softwareentwicklungsprozess]).[<=]

601 **A_25401 - Zero Trust-Komponenten - Darstellung der Voraussetzungen für** 602 **sicheren Betrieb des Produkts im Betriebshandbuch**

603 Der Hersteller einer Zero Trust-Komponente MUSS für sein Produkt im dazugehörigen
604 Betriebshandbuch leicht ersichtlich darstellen, welche Voraussetzungen vom Betreiber
605 und der Betriebsumgebung erfüllt werden müssen, damit ein sicherer Betrieb des
606 Produktes gewährleistet werden kann. [<=]

607 **A_25402 - Zero Trust-Komponenten - Schutz der transportierten Daten**

608 Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass die Vertraulichkeit und
609 Integrität der transportierten Daten gewährleistet ist. [<=]

610 **A_25403 - Zero Trust-Komponenten - Schutzmaßnahmen gegen die OWASP Top** 611 **10 Risiken**

612 Alle Zero-Trust-Komponenten MÜSSEN technische Maßnahmen zum Schutz vor den
613 Risiken in der aktuellen Version der [OWASP-Top-10-Risiken] umsetzen. [<=]

614 **A_25404 - Zero Trust-Komponenten - Angriffe erkennen**

615 Alle Zero Trust-Komponenten MÜSSEN Maßnahmen zur Erkennung, Kategorisierung und
616 Protokollierung bzw. Meldung von Angriffen umsetzen. [<=]

617 *Hinweis: Für die Kategorisieren von Angriffen ist eine Kategorisierung nach "CAPEC:*
618 *OWASP Related Patterns"[CAPEC OWASP] zu verwenden.*

619 **A_25405 - Zero Trust-Komponenten - Angriffen entgegenwirken**

620 Alle Zero Trust-Komponenten MÜSSEN Maßnahmen zur Schadensreduzierung und -
621 verhinderung von Angriffen umsetzen.[<=]

622 **A_25406 - Zero Trust-Komponenten - Eingabe Validierung von Operationen**

623 Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass alle Daten und Parameter, die
624 über eine API kommuniziert werden, sicherheitstechnisch validiert werden.[<=]

625 *Hinweis: Eine Eingabe-Validierung von Fachdienst APIs erfolgt im Fachdienst und nicht in*
626 *den Zero Trust-Komponenten.*

627 **A_25407 - Zero Trust-Komponenten - Sicherheitstechnische Validierung von** 628 **Policy und Konfigurationen**

629 Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass alle Daten und Parameter, die
630 von einer Konfigurationsdatei oder Policy gelesen werden, sicherheitstechnisch validiert
631 werden.[<=]

A_25408 - Zero Trust-Komponenten - Verbot unbefugter Profilbildung

Der Betreiber einer Zero Trust-Komponente DARF anfallende Zero Trust-Verbindungsdaten (Client Eigenschaften, Client-IP-Adresse etc.) NICHT für eine unbefugte Profilbildung der verbundenen Clients bzw. ihrer Nutzer verwenden. [≤]

A_25409 - Zero Trust-Komponenten - Privacy by Design

Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass bei Konfigurationsmöglichkeiten die datenschutzfreundlichere Option vorausgewählt ist. [≤]

A_25410 - Zero Trust-Komponenten - Verbot von Werbe- und Usability-Tracking

Alle Zero Trust-Komponenten DÜRFEN im Produktivbetrieb ein Werbe- und Usability-Tracking NICHT verwenden. [≤]

A_25411 - Zero Trust-Komponenten - Verbot vom dynamischen Inhalt

Alle Zero Trust-Komponenten DÜRFEN dynamischen Inhalt von Drittanbietern NICHT herunterladen und verwenden. [≤]

A_25412 - Zero Trust-Komponenten - Zusätzliche Verschlüsselung bei der Persistierung

Unabhängig davon, ob die Daten schon verschlüsselt vorliegen, MÜSSEN alle Zero Trust-Komponenten die Daten der Komponente bei der Persistierung verschlüsseln. [≤]

A_25413 - Zero Trust-Komponenten - Ordnungsgemäße IT-Administration

Der Betreiber einer Zero Trust-Komponente MUSS die Maßnahmen für erhöhten Schutzbedarf des BSI-Bausteins „OPS.1.1.2 Ordnungsgemäße IT-Administration“ [BSI-Grundschutz] während des gesamten Betriebs der Komponente umsetzen. [≤]

A_25718 - Zero Trust-Komponenten - Bereitstellung Security-KPIs

Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass die für die Komponente relevante Security-KPIs in A_25484 automatisch für den Betreiber bereitgestellt werden. [≤]

Hinweis: Die Anforderung ist besonders wichtig, falls die Zero Trust-Komponente in einer VAU betrieben wird. Die Bereitstellung der Daten soll in das betriebliche Rohdaten-Log erfolgen.

5.1.1 Sicherheits- und Datenschutzanforderungen an Logging und Monitoring

Hinweis: Die Anforderungen dieses Abschnitts könnten sich noch ändern, falls sich bei der Umsetzung des Zero Trust herausstellt, dass weitere Protokollierungen auf Seiten des Betreibers notwendig werden.

A_25744 - Zero Trust-Komponenten - Datenschutzkonformes Logging und Monitoring

Die Zero Trust-Komponenten MÜSSEN die für den Betrieb des Zero Trust erforderlichen Logging- und Monitoring-Informationen in solcher Art und Weise erheben und verarbeiten, dass mit technischen Mitteln ausgeschlossen ist, dass dem Betreiber der Zero Trust-Komponenten vertrauliche oder zur unautorisierten Profilbildung geeignete Daten zur Kenntnis gelangen. [≤]

A_25745 - Zero Trust-Komponenten - Keine medizinischen Informationen in Logging und Monitoring

Die Zero Trust-Komponenten MÜSSEN sicherstellen, dass in für den Betrieb erstellten Protokollen keine personenbezogenen medizinischen Informationen enthalten sind (u. a.

678 medizinische Daten von Versicherten oder Informationen, aus denen sich ableiten lässt,
679 bei welchen Leistungserbringerinstitutionen ein Versicherter in Behandlung ist).[<=]

680 **A_25746 - Zero Trust-Komponenten - Keine sicherheitsrelevanten Daten in**
681 **Logging und Monitoring**

682 Die Zero Trust-Komponenten MÜSSEN sicherstellen, dass in für den Betrieb erstellten
683 Protokollen des Betreibers keine sicherheitsrelevanten Daten enthalten sind.[<=]

684 *Hinweis: Sicherheitsrelevante Daten sind zum Beispiel, Kryptoschlüssel, Access/Refresh*
685 *Token usw.*

686 **A_25747 - Zero Trust-Komponenten - Löschfristen Protokolle**

687 Der Betreiber einer Zero Trust-Komponente MUSS sicherstellen, dass die

- 688 • zum Zwecke der Fehleranalyse erhobenen Protokolle nach Behebung des Fehlers
689 unverzüglich gelöscht werden,
- 690 • zum Zwecke des Security Monitorings erhobenen Protokolle nach 6 Monaten
691 gelöscht werden.

692 [\leq]

693 **5.1.2 Sicherheits- und Datenschutz-Anforderungen an das Security**
694 **Monitoring**

695 Das SIEM, Plattform-Monitoring und Shared Signals bilden das Security Monitoring von
696 Zero Trust ab. Die folgenden Anforderungen beschreiben die Fähigkeiten des Security
697 Monitorings und welche Ereignisse erkannt werden sollen.

698 **A_25419 - Security Monitoring - Erkennungsfähigkeit**

699 Der Betreiber des TI 2.0 Dienstes MUSS sicherstellen, dass das Monitoring System die
700 folgenden Merkmale der Kommunikation erkennen kann:

- 701 • Geolokation - Land und Ort
- 702 • Impossible Travel
- 703 • Zugriffe über TOR Netzwerke
- 704 • Zugriff von VPN-Provider
- 705 • Zugriffe über Proxies
- 706 • Zugriffe über Botnetze
- 707 • Traffic Volumen Anomalien
- 708 • Network-Protokoll Anomalien

709 [\leq]

710 *Hinweis: Impossible Travel ist eine Methode zur Anomalieerkennung in der*
711 *Cybersicherheit, die potenzielle Kompromittierungen identifiziert, indem sie*
712 *Benutzeranmeldeaktivitäten analysiert und mit geografischen Standorten korreliert.*
713 *Dabei werden Fälle markiert, in denen auf das Benutzerkonto innerhalb eines verdächtig*
714 *kurzen Zeitraums aus zwei verschiedenen Ländern zugegriffen wird.*

715 Der Fachdienst kann eine Missbrauchserkennung implementieren. Dabei werden mögliche
716 Angriffe und Anomalien innerhalb der Anwendungslogik erkannt (z. B.
717 falsche/manipulierte Metadaten für Dokumente in der elektronischen Patientenakte
718 (ePA)) und an das SIEM System gemeldet.

A_25421 - Security Monitoring - Empfang von Missbrauchserkennung auf Fachdienstebene

Falls der Fachdienst eine Missbrauchserkennung durchführt, MUSS der Betreiber des Security Monitorings sicherstellen, dass das Security Monitoring solche Missbrauchssignale von dem Fachdienst empfangen und verarbeitet werden kann. [<=]

A_25420 - Security Monitoring - Kommunikationsmerkmale signalisieren

Der Betreiber des Monitoring Systems MUSS sicherstellen, dass bei Erkennung folgender Kommunikationsmerkmale die erforderlichen Informationen automatisiert an den PEP gesendet werden:

- Impossible Travel
- Zugriffe über TOR Netzwerke
- Zugriffe über Proxies
- Zugriffe über Botnetze
- Zugriff von VPN-Provider
- Traffic Volumen Anomalien
- Network-Protokoll Anomalien
- Missbrauchssignale von dem Fachdienst (falls implementiert)

[<=]

Hinweis: Mit dem Signal erhält der PEP die Information, dass sich eine Eigenschaft der aktuellen Session geändert hat. Der PEP sperrt automatisch das aktuelle Access Token, sodass der Client ein neues Access Token beim Authorization Server abfragen muss. Die Abfrage des neuen Access Token beinhaltet immer eine Entscheidung durch den PDP.

A_25484 - Security Monitoring - Security KPIs

Der Betreiber des Monitoring Systems MUSS einmal täglich als Teil der Bestandsdatenlieferung die folgende Sicherheits-KPIs automatisiert über die von der gematik angebotene Schnittstelle an das TI SIEM System übermitteln:

- Anzahl versuchter Zugriffe von nicht registrierten Geräten (hier muss die KPIs zwischen Fachdienst APIs und Clientregistrierung APIs unterscheiden)
- Anzahl von Netzwerk-Protokoll-Anomalien
- Anzahl von Zugriffen von Botnetzen
- Anzahl von Zugriffen aus jedem Land gezählt plus weitere Zugriffe, die separat in Versicherte und LE ausgewiesen werden
- Anzahl fehlerhafter Gerätefreischaltungen plus weitere breakdown in Versicherte und LE
- Anzahl von Impossible travel Zugriffen (inkl. Land- und Ortsdaten) plus weitere breakdown in Versicherte und LE
- Anzahl von Zugriffen über TOR Netzwerke plus weitere breakdown in Versicherte und LE
- Anzahl von Zugriffen über Proxies plus weitere breakdown in Versicherte und LE
- Anzahl von Zugriffen über VPNs plus weitere breakdown in Versicherte und LE
- Traffic Volumes plus weitere breakdown in Versicherte und LE
- Anzahl erkannte Angriffe in Kategorie (siehe A_25404) plus weitere breakdown in Versicherte und LE

- Anzahl fehlerhafte Authorization Codes vom IDP.

[<=]

Hinweis: Security KPIs beinhalten anonyme Daten und sind nicht auf individuelle Nutzer zurückzuführen.

Hinweis: Netzwerk-Protokoll-Anomalien sind z.B. ungewöhnliche Netzwerk-Aktivitäten, Netzwerk-Protokoll-Aktivitäten oder die Manipulation von Netzwerk-Paketen.

A_25485 - Security Monitoring - Sicherheitsmeldung bei Aktualisierung von PIP-Daten oder PDP-Policies

Der Betreiber des Security Monitoring MUSS sicherstellen, dass bei der Aktualisierung der PIP-Daten oder PDP-Policies eine Sicherheitsmeldung automatisiert über die von der gematik angebotene Schnittstelle an das TI SIEM System übermittelt wird.[<=]

A_25606 - Security Monitoring - Fehlermeldung bei Aktualisierung von PIP-Daten oder PDP-Policies

Der Betreiber des Security Monitoring MUSS sicherstellen, dass beim folgenden Fehler während der Aktualisierung der PIP-Daten oder PDP-Policies eine Fehlermeldung automatisiert über die von der gematik angebotene Schnittstelle an das TI SIEM System übermittelt wird:

- Policy Download Fehler ((http Fehlercode: 400, 404)
- Fehler bei der Integritätsprüfung der Policysignatur

[<=]

5.1.3 Sicherheits- und Datenschutz-Anforderungen an die Verarbeitung von Daten mit dem Schutzbedarf "sehr hoch"

Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Betreiber keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, gibt es Sonderanforderungen, um die Daten während der Verarbeitung zu schützen.

A_25608 - PEP und PDP - Verarbeitung von Daten mit Schutzbedarf "sehr hoch"

Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Betreiber keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Betreiber den ZT Cluster in einer VAU umsetzen. [<=]

A_25763 - Zero Trust-Komponenten - Private Schlüssel der Komponenten-Identitäten in einem HSM

Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Betreiber keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Betreiber die privaten Schlüssel der Identitäten aller Zero Trust-Komponenten in einem HSM speichern. [<=]

A_25764 - Zero Trust-Komponenten - Sicherer Betrieb und Nutzung eines HSMs

Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Betreiber keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Betreiber beim Einsatz eines HSMs sicherstellen, dass die auf dem HSM verarbeiteten privaten Schlüssel, Konfigurationen und eingesetzte Software nicht unautorisiert ausgelesen, unautorisiert verändert, unautorisiert ersetzt oder in anderer Weise unautorisiert benutzt werden können.[<=]

A_25765 - Zero Trust-Komponenten - Einsatz zertifizierter HSM

Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Betreiber keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf,

807 MUSS der Betreiber beim Einsatz eines HSMs sicherstellen, dass dessen Eignung durch
808 eine erfolgreiche Evaluierung nachgewiesen wurde. Als Evaluierungsschemata kommen
809 dabei Common Criteria, ITSEC oder Federal Information Processing Standard (FIPS) in
810 Frage.

811 Die Prüftiefe MUSS mindestens:

- 812 1. FIPS 140-2 Level 3 oder
- 813 2. FIPS 140-3 Level 3 oder
- 814 3. Common Criteria EAL 4+ (mit AVA_VAN.5)

815 entsprechen. [≤]

816 **A_26065 - Nur zugelassene Images in Produktion**

817 Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der
818 Betreiber keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf,
819 MUSS der Betreiber mit technischem Mittel sicherstellen, dass nur von gematik signierte
820 und für den Einsatz in der PU vorgesehene Produktimages in der PU laufen können. [≤]

821 *Hinweis: Ein Beispiel dafür wäre eine Art Plattform Attestierung, die eine Manipulation*
822 *von Images erkennen und ausschließen kann.*

823 **5.1.4 Sicherheits- und Datenschutz Anforderungen an dem Trust** 824 **Client**

825 **A_25802 - Trust Client - Einhaltung der BSI-Prüfvorschrift**

826 Der Trust Client MUSS die Prüfaspekte aus BSI [TR-03161] erfüllen, sofern sie für das
827 Trust Client anwendbar sind. [≤]

828 *Hinweis: Nicht anwendbar können zum Beispiel sein: O.Paid .. Die Anwendbarkeit ist*
829 *zwischen Hersteller des Trust Clients und dem Gutachter zu klären. Der Gutachter gibt*
830 *sein Votum über die Erfüllung der BSI [TR-03161] in Form der Bewertung der Erfüllung*
831 *der A_25802 ab, wobei die A_25802 als „umgesetzt“ bewertet werden kann, wenn die*
832 *anwendbaren Prüfaspekte der BSI [TR-03161] aus Sicht des Gutachters erfüllt sind.*

833 **5.2 Anforderungen an Clientsysteme**

834 Ein Clientsystem ist eine Softwarekomponente in der Verwendung eines Benutzers zum
835 Ausführen fachlicher Anwendungsfälle z.B. als Primärsystem (PVS, AVS, LIS, KIS etc.)
836 oder als Frontend des Versicherten (ePA-App, E-Rezept-App, TI-Messenger etc.). Dieses
837 Clientsystem wird dem Benutzer durch einen Hersteller bzw. Herausgeber zur Verfügung
838 gestellt.

839 **5.2.1 Hersteller und Herausgeber**

840 **A_25335 - Hersteller Clientsystem - Hinweise und Maßnahmen sicherer Betrieb**

841 Der Hersteller bzw. Herausgeber eines Clientsystems MUSS den Benutzer über
842 Maßnahmen zum sicheren Betrieb seines Clientsystems vor der Inbetriebnahme
843 informieren und während des Betriebs stets zum Abruf durch den Benutzer
844 bereithalten. [≤]

845 **A_25336 - Hersteller Clientsystem - Regelmäßige Updates**

846 Der Hersteller bzw. Herausgeber eines Clientsystems MUSS, solange das Produkt nicht
847 abgekündigt ist, dem Benutzer regelmäßig (z. B. quartalsweise) Updates für das

848 Clientsystem bereitstellen, um das Clientsystem dauerhaft auf dem Stand der Technik zu
849 halten und Sicherheitslücken zu schließen. [≤]

850 **A_25337 - Hersteller Clientsystem - Registrierung für product_id**

851 Der Hersteller bzw. Herausgeber eines Clientsystems MUSS sich über einen
852 organisatorischen Prozess bei der gematik für die Nutzung von Diensten, für welche
853 Token abgerufen werden sollen, registrieren. Der Hersteller bzw. Herausgeber eines
854 Clientsystems bekommt dabei eine "product_id" zugewiesen, die in jeder Instanz des
855 Clientsystems verwendet werden MUSS. [≤]

856 **A_25427 - Hersteller Clientsystem Android - Google Cloud Projekt**

857 Der Hersteller bzw. Herausgeber eines Clientsystems für eine Android-basierte
858 Betriebsumgebung MUSS ein Google Cloud Projekt führen oder eine alternative
859 Plattformattestierung verwenden, um Nachweise über die Geräteintegrität einer jeden
860 Clientsysteminstallation beziehen zu können. [≤]

861 **5.2.2 Verbindungsaufbau**

862 **A_25338 - Clientsystem - User-Agent**

863 Das Clientsystem MUSS in allen http-Requests an Dienste der TI den http-Header user-
864 agent gemäß [RFC7231] mit <product_id>/<product_version> mit

- 865 • <product_id> gemäß Registrierung bei der gematik mit Länge maximal 20
866 Zeichen, Zeichenvorrat [0-9a-zA-Z_-]
- 867 • <product_version> gemäß Produktidentifikation mit Länge 1-20
868 Zeichen, Zeichenvorrat [0-9a-zA-Z_-].]

869 des Clientsystems befüllen. [≤]

870 **A_25339 - Clientsystem - Exponential Backoff**

871 Das Clientsystem SOLL bei Server-seitigen Fehlermeldungen, die auf eine Überlastung
872 des Zielsystems schließen lassen (z. B. http-status 5xx, 429 - too many requests etc.),
873 erneute Verbindungsversuche nach dem Prinzip des Exponential Backoffs [ExpBack]
874 durchführen. [≤]

875 **A_25340 - Clientsystem - Zertifikatsprüfung**

876 Das Clientsystem MUSS alle Zertifikate, die es aktiv verwendet (bspw. TLS-
877 Verbindungsaufbau), auf Integrität und Authentizität prüfen. Falls die Prüfung kein
878 positives Ergebnis ("gültig") liefert, so MUSS es die von dem Zertifikat und den darin
879 enthaltenen Attributen (bspw. öffentliche Schlüssel) abhängenden Arbeitsabläufe
880 ablehnen.

881 Das Clientsystem MUSS alle öffentlichen Schlüssel, die es verwenden will, auf eine positiv
882 verlaufene Zertifikatsprüfung zurückführen können. [≤]

883 **A_25341 - Clientsystem - TLS- oder JWT-Clientauthentisierung**

884 Das Clientsystem MUSS sich in allen http-Requests an der Außenschnittstelle des PEP
885 entweder

- 886 • mittels eines gültigen TLS-Clientzertifikats oder
- 887 • mittels private JWT und DPoP-Token

888 als legitimer Client authentisieren. Verfügt das Clientsystem über kein gültiges
889 Clientauthentisierungsmittel, MUSS es den Bezug über die Clientregistrierung
890 starten. [≤]

891 **5.2.3 Clientregistrierung**

Offener Punkt: Details in diesem Kapitel werden im Rahmen der Implementierung zwischen gematik und dem Zero-Trust-Hersteller festgelegt. Die Client Registrierung wird dienstübergreifend ermöglichen, dass im Falle von Big Apps der Nutzer nur einmalig aktiv werden muss, um sein Gerät mittels 2. Faktor zu bestätigen.

892 **A_25432 - Clientsystem - Ablauf Clientregistrierung**

893 Das Clientsystem MUSS, sofern es an Schnittstellen der Telematikinfrastuktur wegen
894 einer ungültigen/fehlenden Geräte/App-Registrierung abgewiesen wird, eine
895 Registrierung am Service der Client Registry durchführen, in dem es

- 896
- den/die Benutzer:in mittels OpenID Connect authentifiziert,
 - 897 • kryptografische Client-Credentials lokal generiert,
 - 898 • die generierten Credentials sowie die Clientintegrität attestiert und
 - 899 • eine zusätzliche Benutzerbestätigung mittels One-Time-Passwort über einen
900 zweiten Kommunikationsweg (z. B. E-Mailbestätigung) startet

901 [\leq]

902 **A_25766 - Clientsystem - Client Credentials in TI Qualität**

903 Das Clientsystem MUSS die Client-Credentials im Form von kryptografischen Schlüsseln
904 gemäß der Festlegungen in [gemSpec_Krypt] (Verfahren, Algorithmen, Schlüssellängen
905 etc.) unterstützen. [\leq]

906 **A_25769 - Clientsystem - Client Credentials sicher generieren und schützen**

907 Das Clientsystem auf mobilem Gerät mit Apple- oder Android-basierter
908 Betriebsumgebung MUSS die Generierung der Client-Credentials in Hardware-Modulen
909 (alternativ Android TEE) derart generieren und speichern, dass ein Kopieren und
910 Exportieren der Schlüssel durch die Hardware (alternativ Android TEE) verhindert
911 wird. [\leq]

912 **A_25770 - Clientsystem - Client Credentials Rotation**

913 Das Clientsystem MUSS seine Client-Credentials regelmäßig rotieren (erneuern und neu
914 registrieren), wobei die Häufigkeit der Rotation durch die gematik nach einer Auswertung
915 der initialen Benutzererfahrung festgelegt wird. [\leq]

916 *Hinweis: Perspektivisch werden weitere Attestierungsmechanismen für Clientsysteme*
917 *aufgenommen, z. B. FIDO2, TPM2.*

918 **A_25767 - Clientsystem - Clientkey in JWT oder Zertifikat**

919 Das Clientsystem MUSS wahlweise Private Key JWT [RFC7521] und [RFC7523] sowie
920 DPoP [RFC9449] oder mutual TLS [RFC8705] zur Authentifizierung unterstützen. [\leq]

921 **A_25434 - Clientsystem - Clientregistrierung mit bestätigten**922 **Umgebungseigenschaften Android**

923 Das Clientsystem für eine Android-basierte Betriebsumgebung MUSS seine Client-
924 Credentials, App-Integrität und -Authentizität sowie OS-/FW und HW-Eigenschaften über
925 Key and ID Attestation gegenüber PEP Client Registrierung bestätigen.. [\leq]

926 **A_25768 - Clientsystem - Clientregistrierung mit bestätigten**927 **Umgebungseigenschaften Apple**

928 Das Clientsystem für eine Apple-basierte Betriebsumgebung (iOS, macOS) MUSS die
929 Client-Credentials, App Integrität und Authentizität über DCAAppAttest gegenüber PEP
930 Client Registrierung bestätigen. Eigenschaften der Laufzeitumgebung MÜSSEN durch das
931 Clientsystem über einen geprüften Prozess bestätigt werden. [\leq]

A_25758 - Clientsystem - Erfassung Kontaktinformation für Offband-Verifikation

Das Clientsystem MUSS vom Benutzer eine strukturell valide Kontaktinformation (E-Mailadresse, Telefonnummer) abfragen und für eine Offband-Verifikation (Trust on First Use) an Endpunkt der Client Registry übertragen. [≤]

A_25732 - Clientsystem - Unterstützung des Nutzers bei der Registrierung

Das Clientsystem MUSS den Nutzer bei der Clientregistrierung und -Verwaltung geeignet unterstützen (z. B. mittels Guide, Tutorial o. ä.). [≤]

A_25733 - Clientsystem - Clientverwaltung und manuelle De-Registrierung

Das Clientsystem MUSS dem Nutzer eine Übersicht aller beim Clientregistrierungsdienst registrierten Clients darstellen und die Möglichkeit zur De-Registrierung einzelner Clients anbieten. [≤]

A_25734 - Clientsystem - Zugriffsprotokoll Clientregistrierung

Das Clientsystem MUSS dem Nutzer einen Einblick in das Zugriffsprotokoll der Schnittstellen des Clientregistrierungsdienstes für genutzte Clients dieses Nutzers geben. [≤]

A_25735 - Clientsystem - Aktivierung Push-Benachrichtigung

Das Clientsystem MUSS dem Nutzer die Möglichkeit geben, Push-Benachrichtigungen für Aktivitäten über registrierte Clients und Neuregistrierungen für diesen Nutzer zu aktivieren. [≤]

5.2.4 Nutzerauthentifizierung**A_25761 - Clientsystem - Nutzerauthentifizierung mittels etablierter Standards**

Das Clientsystem MUSS die Mechanismen OAuth2, OpenID Connect und OpenID Federation (Auswahl des zuständigen sektoralen IDP) unterstützen. [≤]

Hinweis: Perspektivisch sollen Clientsysteme auch OpenID for Verifiable Credentials (OIDC4VC) unterstützen.

A_25762 - Clientsystem - Nutzerauthentifizierung - Unterstützung etablierter Identitäten und Dienste

Das Clientsystem MUSS zur Authentifizierung des Nutzers mindestens eines der folgenden Verfahren unterstützen:

- Authentifizierung des Nutzers gegenüber einem Sektoralen IDP der IDP Föderation gemäß [gemSpec_IDP_Sek] (GesundheitsID)
- Authentifizierung des Nutzers gegenüber dem zentralen IDP-Dienst der TI gemäß [gemSpec_IDP_Dienst] (SmartCardIDP für kartengebundene Identitäten).
- Authentifizierung des Nutzers mittels SM-B signiertem Client Assertion JWT und DPoP gemäß [RFC7523] und [RFC9449].

[≤]

5.2.5 Session Management**A_25781 - Clientsysteme - OAuth2 Autorisierung**

Das Clientsystem MUSS die Rolle eines OAuth2 Clients [RFC6749] übernehmen und eine Autorisierung vom Authorisation Server einholen. Dabei MUSS PKCE Flow [RFC7636] verwendet werden.

[≤]

A_25782 - Clientsystem - OAuth2 Session Management

Das Clientsystem MUSS

- die vom Authorisation Server ausgestellten Access- und Refresh-Token gemäß [RFC6749#1.5] sowie die DPoP Schlüssel gemäß [RFC9449] bis zur nächsten Aufforderung zur Autorisierung oder Authentifizierung als User-Session sicher aufbewahren,
- regelmäßig neue Access-Token über Refresh-Token erneuern und
- eine Refresh-Token-Rotation gemäß [RFC6749#10.4] unterstützen.

[<=]

A_25783 - Clientsystem - Anweisungen aus http Response Status Codes und Header folgen

Das Clientsystem MUSS die http Response Status Codes und http Header entsprechend der Vorgaben der Fachdienste und Zero Trust APIs auswerten und den Anweisungen daraus folgen und insbesondere

- eine Step-Up- oder erneute Authentifizierung des Nutzers,
- eine Re-Autorisierung und erneute Attestation der Client-Instanz,
- eine Anzeige der Warnungen aufgrund der Policy Entscheidungen und
- ein Nonce-Replay gemäß [RFC8555#6.5.1]

umsetzen.[<=]

5.3 Zero Trust Cluster

Die Software des Zero Trust Clusters wird im Auftrag der gematik entwickelt und als signierte Docker Container in einer gematik Container Registry sowie mit Kubernetes Manifest Dateien in einem gematik GitHub Repository bereitgestellt, sodass die Betreiber von TI 2.0 Diensten ihren spezifischen ZT Cluster darauf aufbauend als Kubernetes Cluster konfigurieren können. Die ZT Cluster-Konfiguration muss in ein GitHub Repository der gematik als git submodule verlinkt werden. Der Cluster Management Service des Zero Trust Clusters setzt durch, dass nur die im gematik GitHub Repository verlinkte ZT Cluster Konfiguration ausgeführt werden kann. Dadurch ist es möglich, dass der Betreiber seine Cluster-Konfiguration selbständig anpassen und die gematik den korrekten Einsatz des ZT Clusters prüfen kann.

A_26106 - ZT Cluster, Verwendung der gematik Docker Container

Der Betreiber eines TI 2.0 Dienstes MUSS für seinen ZT Cluster die von der gematik bereitgestellten signierten Docker Container für PEP, PDP, Cluster Management Service und Telemetrie-Daten Service verwenden.[<=]

A_26105 - ZT Cluster, Durchsetzung der Konfiguration

Der Betreiber eines TI 2.0 Dienstes MUSS für seinen ZT Cluster die ihm zugewiesene Konfiguration aus dem GitHub Repository der gematik verwenden.[<=]

Hinweis: Für die Anpassung und Inbetriebnahme von geänderten ZT Cluster Konfigurationen ist ein Continuous Delivery Prozess mit Quality Gates vorgesehen, der sich noch in der Entwicklung befindet.

1015 5.4 Anforderungen an Policy Enforcement Points

1016 Der Policy Enforcement Point (PEP) stellt die zentrale Sicherheitskomponente einer Zero
1017 Trust Architektur dar, da in dieser alle Zugriffsentscheidungen durchgesetzt (engl.:
1018 enforce) werden. Gemäß oben vorgeschlagener Architekturzerlegung, empfiehlt es sich,
1019 den PEP in mehreren Teilkomponenten zu realisieren.

1020 Mit der Registrierung an der Komponente Client Registry werden von Nutzern verwendete
1021 Clients (Gerät/App Kombinationen) identifizierbar gemacht und können an ausgegebene
1022 Zugriffstoken gebunden werden. Über einen Authorization Server kann im PEP eine
1023 Benutzersession angelegt und verwaltet werden, über die eine mögliche Veränderung von
1024 Sessionparametern (verwendetes Gerät/App, Zugriffsfrequenz) beobachtet werden kann,
1025 um nach Bedarf zusätzliche Sicherheitsmechanismen aktivieren zu können (Step-Up-
1026 Authentication, Throttling etc.). Ist ein Zugriff zu gewähren, wird der Nutzer-Request
1027 über einen http Proxy an das angefragte Resource Backend weitergeleitet.

1028 5.4.1 PEP Client Registry

1029 **A_25644 - PEP Client Registration - Mobile Attestation**

1030 Die Komponente Client Registry MUSS die Clients/Apps bei der Registrierung über
1031 folgende Mechanismen attestieren:

- 1032 • Android Key ID Attestation
- 1033 • Apple DCAAppAttest.
- 1034 • SM-B signiertes Client Assertion JWT

1035 [\leq]

1036 **A_25645 - PEP Client Registration - Attestation mittels TI-Smartcard**

1037 Die Komponente Client Registry MUSS die Registrierung über eine TI-Smartcard
1038 durchführen (eGK, SMC-B, HBA), falls die Ausführungsumgebung des Clients keine
1039 plattformseitigen Attestation-Mechanismen anbietet. Die Verwendung des zentralen IDP-
1040 Dienstes ist für die Nutzerauthentifizierung zulässig. [\leq]

1041 **A_25648 - PEP Client Registration - Device Session Credentials**

1042 Die Komponente Client Registry MUSS die Client Credentials aufbewahren und
1043 Verifikationsmechanismen dem PEP Authorization Server bereitstellen. [\leq]

1044 **A_25649 - PEP Client Registration - Regelmäßige Wiederholung der Attestation**

1045 Die Komponente Client Registry MUSS die Client Attestierung regelmäßig gemäß
1046 Festlegungen in der Device Policy wiederholen. [\leq]

1047 **A_25650 - PEP Client Registration - TI-Identität in Attestation**

1048 Die Komponente Client Registry MUSS den registrierten Client an eine TI-Identität (KVNR
1049 oder TelematikID, festgestellt z. B. über die Einbindung des zentralen IDP-Dienstes oder
1050 eines Sektoralen IDP) binden. [\leq]

1051 **A_25651 - PEP Client Registration - Offband Nutzer Verification**

1052 Die Komponente Client Registry MUSS einen Offband Prozess (z. B. E-Mail, SMS) für die
1053 Kommunikation mit diesem Nutzer unterstützen (Trust on First Use), wobei der Nutzer
1054 seine E-Mail Adresse bzw. Kontaktinformation eigenverantwortlich vergibt. [\leq]

1055 **A_25652 - PEP Client Registration - Push Gateway**

1056 Die Komponente Client Registry MUSS Push-Notifications über die von App-Anbietern
1057 bereitgestellten Push-Gateways unterstützen, um die Notifications an bestimmte oder alle
1058 registrierte Clients eines Anwenders verschicken zu können. [\leq]

A_25737 - PEP Client Registration - Push Notification

Die Komponente Client Registry MUSS eine Push Benachrichtigung an alle registrierten Clients des Nutzers, für die eine Push Notification aktiviert ist, verschicken, sobald sich Änderungen an der Liste der registrierten Clients dieses Nutzers ergibt. [≤]

Hinweis: Wie Clients ihre Push Konfiguration in den PEP eintragen können, wird in einer Folgeversion des vorliegenden Dokuments festgelegt.

A_25653 - PEP Client Registration - Umsetzung der Device Policy

Die Komponente Client Registry MUSS die zulässigen Clients entsprechend der Konfiguration oder Policy (über PDP-Decision) ermitteln und nur diese gemäß der festgelegten Device Policy registrieren. Geräte, die die geforderten Parameter der Device Policy nicht unterstützen bzw. nicht das geforderte Niveau erreichen, MÜSSEN abgelehnt werden. [≤]

A_25752 - PEP Client Registration - Nutzer über Hintergrund zur Ablehnung der Gerätregistrierung informieren

Falls ein Gerät nicht die geforderten Parameter der Device Policy unterstützt bzw. das geforderte Niveau nicht erreicht, MUSS die Komponente Client Registry den Nutzer nutzerfreundlich darüber informieren, welche Geräteigenschaften zu der Ablehnung geführt haben. [≤]

A_25654 - PEP Client Registration - Minimum Device Policy

Die Komponente Client Registry MUSS Client Mindestanforderungen vor der Registrierung verifizieren und kann hierfür eine Schnittstelle des PDP verwenden. [≤]

A_25738 - PEP Client Registration - Telemetrie Clientregistrierung

Die Komponente Client Registry MUSS in den Telemetriedaten zu jeder versuchten Clientregistrierung folgende Parameter ohne einen Nutzerbezug protokollieren:

- Geräteparameter (Betriebssystem(-version), Patchlevel, Geolocation etc.) gemäß Geräteattestierung
- verwendeter Faktor für Offband-Verifikation (E-Mail, SMS etc.)
- Zeitstempel Registrierung, Zeitpunkt Offband-Bestätigung
- verwendeter Faktor der Nutzerauthentifizierung (SmartCard, Digitale Identität)
- Status/Ergebnis des Registrierungsversuchs

[≤]

A_25754 - PEP Client Registration - Notfall-Recovery-Prozess für Nutzer

Die Komponente Client Registry MUSS dem Nutzer einen Notfall-Recovery-Prozess anbieten, falls der Nutzer sein letztes Gerät verloren und keinen Zugriff mehr auf seine registrierte E-Mail-Adresse/Telefonnummer hat. [≤]

A_26064 - Access Token bei Monitoring-Signalen sperren

Falls das Monitoring System eine Änderung in den Kommunikationsmerkmalen signalisiert, muss der PEP das aktuelle Access Token sperren. [≤]

Hinweis: der Client muss danach ein neues Access Token beim Authorization Server abfragen. Die Abfrage des neuen Access Token beinhaltet immer eine Entscheidung durch den PDP.

5.4.1.1 Sicherheits- und Datenschutz-Anforderungen an dem PEP Client Registration

A_25751 - PEP Client Registration - Anwendungsfälle nur vom registrierten Client

Nach der erfolgreichen Registrierung des ersten Clients (Geräts/App Kombination), MUSS die Komponente Client Registry sicherstellen, dass die folgenden Anwendungsfälle nur von einem registrierten Client durchgeführt werden kann:

- Gerät löschen
- Gerät umbenennen
- E-Mail-Adresse hinzufügen
- E-Mail-Adresse aktualisieren

[<=]

A_25748 - PEP Client Registration - Maximale Anzahl von Geräten

Die Komponente Client Registry MUSS sicherstellen, dass ein Nutzer maximal 256 Geräte registrieren kann. Der Wert muss konfigurierbar sein. [<=]

A_25749 - PEP Client Registration - Nutzer Protokollierung

Die Komponente Client Registry MUSS ein Nutzerprotokoll führen und die folgenden Anwendungsfälle für den Nutzer protokollieren:

- Gerät hinzufügen
- Gerät löschen
- Gerät umbenennen
- E-Mail-Adresse hinzufügen
- E-Mail-Adresse aktualisieren

[<=]

A_25750 - PEP Client Registration - Nutzer über sicherheitsrelevante Ereignisse informieren

Die Komponente Client Registry MUSS sicherstellen, dass der Nutzer bei folgenden Anwendungsfällen informiert wird:

- Gerät hinzufügen
- Gerät löschen
- Gerät umbenennen
- E-Mail-Adresse aktualisieren

[<=]

Hinweis: Der Nutzer kann z.B. durch eine geeignete E-Mail oder App-Notifikation über die sicherheitsrelevanten Ereignisse informiert werden.

5.4.2 PEP Relying Party

A_25655 - PEP - Relying Party

Die Komponente Policy Enforcement Point MUSS in der TI-Föderation als Relying Party registriert sein. [<=]

1139 A_25656 - PEP - Entity Statement

1140 Die Komponente Policy Enforcement Point MUSS die Redirect-URIs aller zulässigen
1141 Clients als erlaubte Redirect-URIs im Entity Statement ausweisen. [≤]

1142 A_25657 - PEP - Authentication über sektoralen IDP

1143 Die Komponente Policy Enforcement Point (bzw. ihre Subkomponente Authorization
1144 Server) MUSS die Nutzer über sektorale IDPs authentifizieren können. [≤]

1145 A_25658 - PEP - Authentication über SmartCard IDP

1146 Die Komponente Policy Enforcement Point (bzw. ihre Subkomponente Authorization
1147 Server) MUSS die Nutzer über den zentralen IDP-Dienst (SmartCard-IDP)
1148 authentifizieren können. [≤]

1149 5.4.3 PEP Authorization Server**1150 A_25760 - PEP Authorization Server - OAuth2 Schnittstellen**

1151 Die Komponente Authorization Server MUSS eine OAuth2 Schnittstelle gemäß [RFC6749]
1152 und [RFC7636] implementieren.

1153 Der Authorization Server MUSS am Token Endpunkt REFRESH_TOKEN entsprechend
1154 [RFC6749] ausstellen können. [≤]

1155 A_25659 - PEP Authorization Server - Check Device Registration

1156 Die Komponente Authorization Server MUSS die Client Instanzen über einen der
1157 folgenden Mechanismen authentifizieren:

- 1158 • Mutual-TLS Client Authentication gemäß [RFC8705]
- 1159 • JSON Web Token Client Authentication gemäß [RFC7523] mit DPOP gemäß
1160 [RFC9449]

1161 und Anfragen, die weder noch eine der genannten Mechanismen verwendet konsequent
1162 ablehnen. [≤]

1163 A_25660 - PEP Authorization Server - Session Management mittels**1164 AccessTokens und Refresh-Tokens**

1165 Die Komponente Authorization Server MUSS ein Session Management mittels OAuth2
1166 und Ausgabe, Verwaltung und Entzug von Access- und Refresh-Token gemäß
1167 [RFC6749#1.5] unterstützen. [≤]

1168 A_25661 - PEP Authorization Server - Umsetzung der Policy Decision

1169 Die Komponente Authorization Server MUSS die Zugriffs-Entscheidung eines PDP mittels
1170 der Ausgabe eines Access- und eines Refresh-Tokens umsetzen bzw. eine
1171 Zugriffsverweigerung mit einem http-Statuscode 403 quittieren. [≤]

1172 A_25662 - PEP Authorization Server - Refresh-Token Rotation

1173 Die Komponente Authorization Server MUSS eine Refresh-Token Rotation gemäß
1174 [RFC6749#10.4] erzwingen und MUSS sicherstellen, dass ein Refresh Token nur einmal
1175 gegen ein Access-Token und ein Refresh-Token getauscht werden kann. [≤]

1176 A_25663 - PEP Authorization Server - Token-Binding an Device-Registration

1177 Die Komponente Authorization Server MUSS auszugebende Access-Token und Refresh-
1178 Token an die, über einen der Mechanismen:

- 1179 • TLS Client Zertifikat Binding gemäß [RFC8705]
- 1180 • OAuth 2.0 Demonstrating Proof of Possession (DPOP) gemäß [RFC9449]

1181 identifizierte Client-Instanz binden, in dem im Token-Binding-Claim die Angabe der
1182 verwendeten Clientidentifikation als "jkt" oder "x5t#S256" eindeutig referenziert
1183 wird. [≤]

A_25664 - PEP Authorization Server - Token Laufzeit gemäß Policy

Die Komponente Authorization Server MUSS die Laufzeit der ausgegebenen Token entsprechend der Festlegungen aus der getroffenen Zugriffsentscheidung des PDP setzen. [\leq]

A_25665 - PEP Authorization Server - Plugin-Schnittstelle Application Authorization Backend

Die Komponente Authorization Server MUSS eine Plug-In Schnittstelle zu einem anwendungsspezifischen Authorization Backend [GITHUB-Authz-Backend] implementieren und dabei die folgenden Signale und Informationen aus der erhaltenen Zugriffsanfrage weiterreichen.

Tabelle 5: PEP Authorization Server - Plugin-Schnittstelle Application Authorization Backend

Operation	Operation Kennung	Input	Output
Benachrichtigung über die Ablehnung des Zugriffs durch PEP	notifyAccessDenied	Trace-Id Subject-Information Client-Information PDP-Decision	-
Anwendungsspezifische Autorisierung	authorizeAccess	Trace-Id Session-Id Authorization-Scopes Authorization-Details Subject-Information Client-Information PDP-Decision	Zugriff erlauben Ja/Nein Zusätzliche Authorization Scopes Zusätzliche Authorization Details Zusätzliche Claims
Benachrichtigung über abgelaufene oder terminierte Sessions	notifySessionTermination	Trace-Id Session-Id	-

[\leq]

5.4.3.1 Service Discovery

Der Authorization Server ermöglicht Clients die Ermittlung der bereitgestellten Endpunkte durch Abfrage des Well-known json Dokuments unter `http GET /.well-known/oauth-authorization-server`. Wenn der FQDN des Authorization Servers nicht bekannt ist, kann das Well-known json Dokuments abgefragt werden, indem ein Endpunkt des Resource Servers ohne gültiges access token aufgerufen wird. Im body der http 401 Response ist das Well-known json Dokument enthalten.

A_26037 - PEP Authorization Server, Well-known

Die Komponente http Proxy MUSS für den Authorization Server ein Well-known json Dokument gemäß [RFC8615] und [RFC8414] unter folgender URL bereitstellen:

<https://<as-fqdn>/.well-known/oauth-authorization-server>

Das Well-known json Dokument MUSS mit dem Schema

1209 [https://raw.githubusercontent.com/gematik/spec-t20r/main/src/schemas/as-well-](https://raw.githubusercontent.com/gematik/spec-t20r/main/src/schemas/as-well-known.yaml)
1210 [known.yaml](https://raw.githubusercontent.com/gematik/spec-t20r/main/src/schemas/as-well-known.yaml) validiert werden können.

1211 Das Attribut `scopes_supported` MUSS mindestens die folgenden Werte enthalten:
1212 - `zero:register`
1213 - `zero:manage`

1214 Weitere Werte für das Attribut `scopes_supported` müssen per Konfiguration ergänzt
1215 werden können.

1216 Es MUSS ein Attribut `nonce_endpoint` enthalten sein, dass die URL zur Abfrage neuer
1217 Nonce-Werte angibt.

1218 Es MUSS ein Attribut `openid_providers_endpoint` enthalten sein, dass die URL zur
1219 Abfrage der unterstützten OpenID Provider enthält. [`<=`]

1220 *Hinweis: Der FQDN des Authorization Servers wird vom Anbieter des TI 2.0 Dienstes*
1221 *vergeben.*

1222 *Die unterstützten OpenID Provider sind in der PU [https://idp.app.ti-](https://idp.app.ti-dienste.de/directory/fed_idp_list)*
1223 *[dienste.de/directory/fed_idp_list](https://idp.app.ti-dienste.de/directory/fed_idp_list) und in der RU [https://idp-ref.app.ti-](https://idp-ref.app.ti-dienste.de/directory/fed_idp_list)*
1224 *[dienste.de/directory/fed_idp_list](https://idp-ref.app.ti-dienste.de/directory/fed_idp_list).*

1225 **A_26090 - PEP Authorization Server, Well-known Erstellung**

1226 Die Komponente http Proxy MUSS für den Authorization Server das Well-known json
1227 Dokument aus den Konfigurationsdaten des Zero Trust Clusters erzeugen.
1228 [`<=`]

1229 Beispiel Well-known json Dokument des Authorization Servers:
1230 {
1231 "issuer": "https://zerobin.zt.dev.ccs.gematik.solutions",
1232 "authorization_endpoint":
1233 "https://zerobin.zt.dev.ccs.gematik.solutions/auth",
1234 "token_endpoint": "https://zerobin.zt.dev.ccs.gematik.solutions/token",
1235 "jwks_uri": "https://zerobin.zt.dev.ccs.gematik.solutions/jwks",
1236 "nonce_endpoint": "https://zerobin.zt.dev.ccs.gematik.solutions/nonce"
1237 "openid_providers_endpoint": "https://idp.app.ti-
1238 dienste.de/directory/fed_idp_list"
1239 "scopes_supported": [
1240 "zero:register",
1241 "zero:manage"
1242],
1243 "response_types_supported": [
1244 "code"
1245],
1246 "response_modes_supported": [
1247 "query"
1248],
1249 "grant_types_supported": [
1250 "authorization_code"
1251],
1252 "token_endpoint_auth_methods_supported": [
1253 "none"
1254],
1255 "token_endpoint_auth_signing_alg_values_supported": [
1256 "ES256"
1257],
1258 "service_documentation": "https://github.com/gematik/zero-lab",
1259 "ui_locales_supported": [
1260 "de",
1261 "en"
1262],
1263 "code_challenge_methods_supported": [

```
1264     "s256"  
1265   ]  
1266 }
```

1267 **5.4.3.2 Ablauf der SM-B Authentifizierung mit DPoP**

1268 Die SM-B Authentifizierung wird für Nutzer und Clients angeboten, die nicht über andere
1269 geeignete Credentials verfügen, um sich als berechtigte TI-Teilnehmer auszuweisen. Die
1270 Authentifizierung erfolgt gemäß OAuth2 Client Authentifizierung [RFC7523] mit SM-B
1271 signiertem Client Assertion JWT. Replay-Attacken werden durch die Aufnahme einer
1272 server-generierten Nonce als JTI Claim in der Client-Assertion verhindert.

1273 **A_26091 - ZT Cluster, SM-B Authentifizierung mit DPoP**

1274 Der Zero Trust Cluster MUSS den Ablauf gemäß Abbildung SM-
1275 *B_Authentisierung_mit_DPoP* sowie gemäß [RFC7523] und [RFC9449] unterstützen. [\leq]

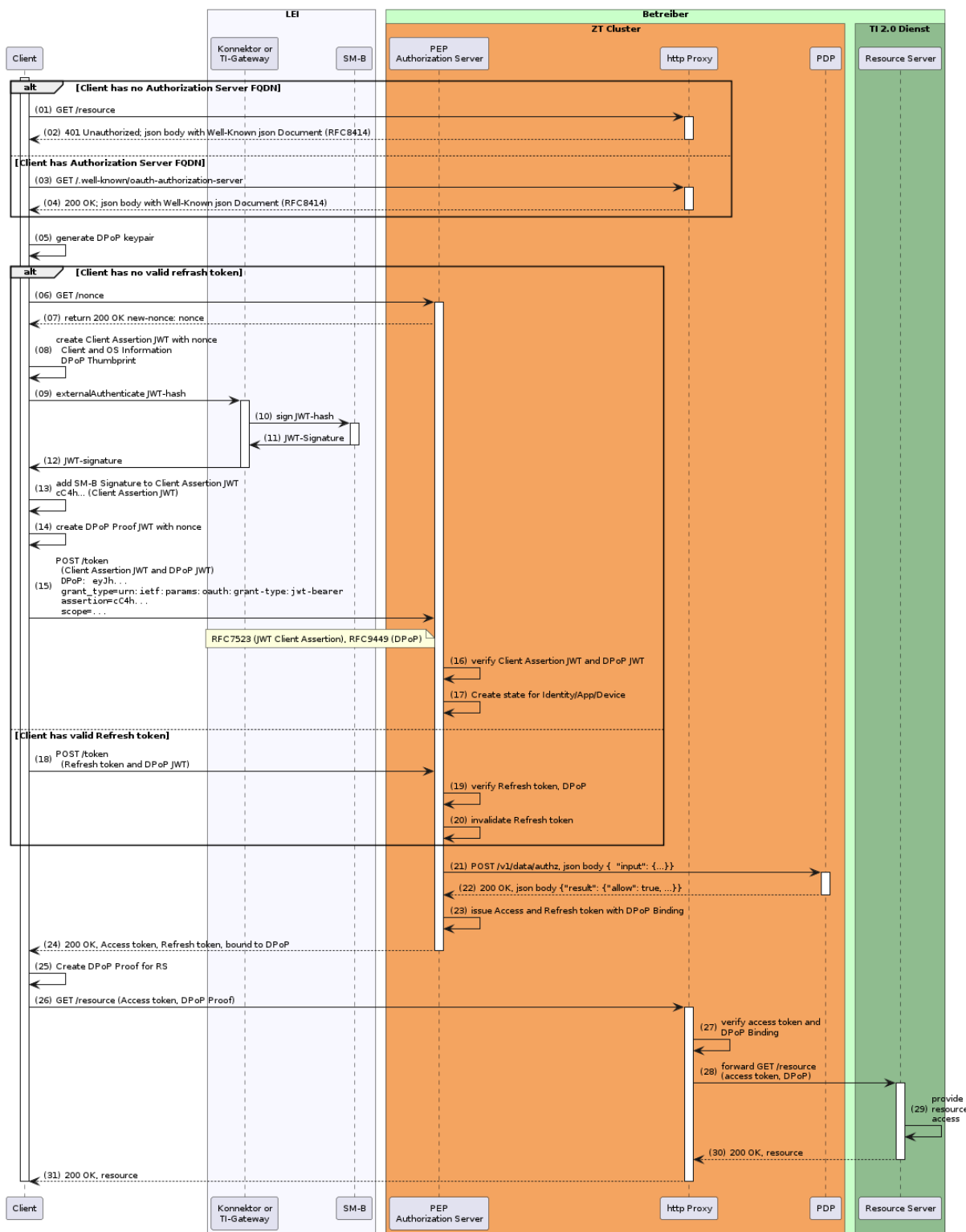


Abbildung 3 SM-B_Authentisierung_mit_DPoP

Schritt (1) bis (4) dienen dazu die Endpunkte des Authorization Servers zu ermitteln. Wenn der FQDN des Authorization Servers noch nicht bekannt ist, dann versucht der Client in Schritt (1) auf Daten des Resource Servers zuzugreifen. Die Anfrage wird vom http Proxy mit http 401 Unauthorized abgelehnt und der Client erhält das Well-known json Dokument mit den Endpunkten des Authorization Servers. Alternativ kann bei

1283 bekanntem FQDN des Authorization Servers das Well-known json Dokument direkt
1284 abgefragt werden (Schritte (3) und (4)).

1285 Im Schritt (05) generiert der Client ein ephemeres key pair für DPoP ([RFC9449]). DPoP
1286 stellt kryptografisch sicher, dass Access token, die für diesen Client vom Authorization
1287 Server ausgestellt wurden, auch nur von diesem Client verwendet werden können.

```
1288 {  
1289     "crv": "P-256",  
1290     "kty": "EC",  
1291     "x": "...",  
1292     "y": "...",  
1293     "d": "..."  
1294 }
```

1295 Zur Abwehr von Replay-Attacken wird in Schritt (06) eine server-generierte Nonce
1296 abgefragt.

```
1297 HEAD /nonce http/1.1  
1298 Host: as.example.com  
1299  
1300 http/1.1 200 OK  
1301 new-nonce: ...Nonce from the AS...
```

1302 Danach erzeugt der Client in Schritt (08) bis (13) das Client Assertion JWT, in dem auch
1303 Informationen über das Gerät, das Betriebssystem und die App (PS) enthalten sind. Das
1304 JWT wird mit der SM-B signiert. Im Attribut "jkt" ist der Hash des öffentlichen
1305 Schlüssels des Client DPoP Keys enthalten.

```
1306 {  
1307     "nonce": "...Nonce from the AS...",  
1308     "iss": "urn:telematik:telematik-id:9-123456789", // Telematik ID from  
1309 X.509 certificate  
1310     "sub": "...Client ID...",  
1311     "aud": "https://as.example.com", // AS URL  
1312     "iat": 1562262611,  
1313     "exp": 1562262616,  
1314     "cnf": {  
1315         "jkt": "...thumbprint of the DPoP key..."  
1316     },  
1317     "urn:telematik:client-self-assessment": {  
1318         "product_id": "PS-000",  
1319         "product_version": "0.5.0",  
1320         "manufacturer_id": "HRST-001",  
1321         "platform": "software"  
1322     },  
1323     "runtime": {  
1324         "os": "Microsoft Windows",  
1325         "os_version": "10.0.19045.4291",  
1326         "os_arch": "x86",  
1327     }  
1328 }
```

1328 Der Client erzeugt in Schritt (14) das DPoP Proof JWT, in dem die nonce enthalten ist.

```
1329 {  
1330     "typ": "dpop+jwt",  
1331     "alg": "ES256",  
1332     "jwk": {  
1333         "kty": "EC",  
1334         "x": "18tFrhx-34tV3hRICRDY9zCkDlpBhF42UQUfWVAWBfS",  
1335         "y": "9VE4jfh_Ok_o64zbTTlCuNJajHmt6v9TDVrU0CdvGRDA",  
1336     }  
1337 }
```

```
1336     "crv": "P-256"
1337   }
1338 }
1339 .
1340 {
1341   "jti": "-BwC3ESc6acc2lTc",
1342   "htm": "POST",
1343   "htu": "https://server.example.com/token",
1344   "nonce": "...nonce from the AS..."
1345   "iat": 1562262616
1346 }
1347 .
1348 ES256 signature of the DPoP JWT
```

1349 In Schritt (15) wird ein POST /token Request gesendet, um Access token und Refresh
1350 token zu erhalten.

```
1351 POST /token HTTP/1.1
1352 Host: as.example.com
1353
1354 grant_type=authorization_code&
1355 code=n0esc3NRze7LTCu7iYzS6a5acc3f0ogp4&
1356 client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-
1357 type%3Ajwt-bearer&
1358 client_assertion=...JWT Assertion signed by SM-C...
```

1359 Der Authorization Server prüft die im Request übergebenen Client Assertion JWT und
1360 DPoP JWT (16) und erzeugt oder aktualisiert den Eintrag für den SM-B Nutzer und seine
1361 App in der Client Registry (17).

1362 Wenn ein gültiges Refresh token vorhanden ist, dann kann der Client anstatt der Schritte
1363 (06) bis (17) den Schritt (18) nutzen, um das Refresh token gegen ein neues Access
1364 token und ein neues Refresh token einzutauschen. Das alte Refresh token wird geprüft
1365 (19) und verliert sofort seine Gültigkeit (20).

1366 In Schritt (21) und (22) wird durch den PDP die Entscheidung getroffen, ob die vom
1367 Client übergebenen Header-Daten hinreichend sind, um dem Authorization Server zu
1368 erlauben, für den Client neue Access und Refresh token auszustellen. Wenn ein Refresh
1369 token verwendet wurde, übergibt der Authorization Server zusätzlich Daten aus dem
1370 Eintrag in der Client Registry an den PDP.

1371 Wenn die Erlaubnis erteilt wurde, stellt der Authorization Server neue Access und refresh
1372 token mit DPoP Binding aus (23) und sendet sie an den Client (24).

```
1373 HTTP/1.1 200 OK
1374 Content-Type: application/json
1375
1376 {
1377   "access_token": "...",
1378   "token_type": "DPoP",
1379   "expires_in": 3600,
1380   "refresh_token": "...",
1381   "scope": "..."
1382 }
1383 Inhalt des Access token
1384 {
1385   "alg": "ES256",
1386   "kid": "...",
1387 }
```

```

1388 .
1389 {
1390   "iss": " https://as.example.com",
1391   "sub": "...Client ID...",
1392   "aud": "...Client ID...",
1393   "exp": 1562266216,
1394   "cnf": {
1395     "jkt": "...thumbprint of the DPoP key..."
1396   }
1397 }
1398 .
1399 ES256 signature of the access token

```

1400 Der Client erzeugt ein neues DPoP Proof JWT für den Zugriff auf Daten des Resource
 1401 Servers (25) und sendet Request, mit Access token und DPoP Proof im Header, an den
 1402 Resource Server.

1403 Der http Proxy prüft das Access token und das DPoP Binding (27) und leitet, wenn diese
 1404 gültig sind, den Request an den Resource Server weiter (28).

1405 Der Resource Server empfängt den Request, führt seine Fachlogik aus (29) und sendet
 1406 eine Antwort an den http Proxy (30), der diese an den Client weiterleitet (31).

1407 5.4.4 PEP http Proxy

1408 Die Komponente http Proxy ist die "letzte" vor das Resource Backend geschaltete Zero
 1409 Trust-Komponente und prüft das Access token im Authorization Header des Requests. Ist
 1410 das Access token gültig, wird der Zugriff gewährt. Zudem wird der Request um
 1411 zusätzliche http-Header angereichert, um ein Tracing zu ermöglichen.

1412 **A_25666 - PEP http Proxy - TLS Terminierung**

1413 Die Komponente http Proxy MUSS den TLS Kanal terminieren und alle http-Header
 1414 validieren.[<=]

1415 **A_25667 - PEP http Proxy - Verifikation Access-Token Binding**

1416 Die Komponente http Proxy MUSS das Access-Token Binding über einen der folgenden
 1417 Mechanismen verifizieren:

1418 • TLS Client Zertifikat Binding gemäß [RFC8705]

1419 • OAuth 2.0 Demonstrating Proof of Possession (DPoP) gemäß [RFC9449],

1420 d. h. der Claim "jkt" oder "x5t#s256" im Access-Token MUSS eindeutig der Angabe im
 1421 TLS-Client-Zertifikat bzw. DPoP-Token entsprechen.[<=]

1422 **A_25668 - PEP http Proxy - Access-Token Validierung**

1423 Die Komponente http Proxy MUSS das übergebene Access-Token validieren.
 1424 Insbesondere MÜSSEN

1425 • die Signatur des Authorization Servers gültig,

1426 • die Angaben zur zeitlichen Gültigkeit (Felder: *iat*, *exp*) valide und

1427 • die Angabe *aud* für das Resource Backend korrekt eingetragen

1428 sein.[<=]

1429

1430 **A_25669 - PEP http Proxy - Zusätzliche http-Header**

1431 Die Komponente http Proxy MUSS die http Requests an das Resource Backend
 1432 weiterleiten und dabei die folgenden zusätzlichen http-Header einsetzen.

1433 **Tabelle 6: PEP http Proxy - Zusätzliche http-Header**

http-Header	Format	Schema
X-ZTA-Subject	Base64-URL kodierte JSON Struktur	subject.yaml
X-ZTA-Scopes	URL-Encoded String	-
X-ZTA-Authorization-Details	Base64-URL kodierte JSON Struktur	offen
X-ZTA-Client	Base64-URL kodierte JSON Struktur	client-instance.yaml
X-ZTA-PDP-Decision	Base64-URL kodierte JSON Struktur	pdp-decision.yaml

1434
1435 Gleichnamige http-Header aus dem ursprünglichen http-Request MÜSSEN entfernt bzw.
1436 überschrieben werden. [≤]

1437 **5.4.5 Sicherheits- und Datenschutz-Anforderungen an dem PEP**1438 **A_25445 - PEP - Zugriffsentscheidung nur über PDP**

1439 Der PEP MUSS sicherstellen, dass Zugriffe auf den Fachdienst nur durch eine positive
1440 Zugriffsentscheidung vom PDP möglich sind. [≤]

1441 **A_25840 - PEP - Sichere interne Kommunikation**

1442 PEP MUSS die ausgehende interne Kommunikation zum PDP, zum Betreiber spezifischer
1443 Dienste und zur Anwendung spezifischer Dienste über die im Cluster vorhandenen
1444 Mechanismen absichern und deren Authentizität verifizieren können. [≤]

1445 **A_25448 - PEP - Nutzerzugriff nur von registrierten Geräten**

1446 Der PEP MUSS sicherstellen, dass der Zugriff eines Nutzers auf den Fachdienst nur von
1447 einem vom Nutzer registrierten Gerät möglich ist. [≤]

1448 **A_25449 - PEP- Nutzeridentität nur von einem zugelassenem IDP**

1449 Der PEP MUSS sicherstellen, dass nur Nutzeridentitäten von einem zugelassenen IDP
1450 akzeptiert werden. [≤]

1451 **A_25447 - PEP - Kommunikation nur mit authentischen PDP**

1452 Der PEP MUSS sicherstellen, dass er mit einem authentischen und korrekt konfigurierten
1453 PDP kommuniziert. [≤]

1454 **A_25486 - PEP - Abbruch durch Anomalie Signale**

1455 Falls das Security Monitoring eine Anomalie beim Zugriff eines Clients signalisiert, MUSS
1456 der PEP das Access-Token des Clients annullieren und damit die aktuelle fachliche
1457 Operation abbrechen. [≤]

1458 **5.4.6 Konfiguration**1459 **A_26038 - PEP, Konfigurations-Parameter**

1460 Der PEP MUSS die folgenden Konfigurations-Parameter unterstützen.

1461 **Tabelle 7 PEP_Konfigurations-Parameter**

Konfigurations-Parameter	Default Wert	Beschreibung
as-fqdn	n/a	FQDN des PEP Authorization Servers
scopes_supported		vom PEP Authorization Server unterstützte scope Werte Minimal müssen die Zero Trust scope Werte unterstützt werden: - zero:register - zero:manage Zusätzliche scope Werte ergeben sich aus dem Bedarf des Dienstes, der durch den Zero Trust Cluster geschützt wird.

1462

1463 [**<=**]1464 *Hinweis: Weitere Konfigurationsparameter werden zusammen mit dem Zero Trust*1465 *Hersteller festgelegt und hier ergänzt.*1466 **5.5 Anforderungen an den Policy Decision Point**

1467 Der PDP implementiert einen [Open Policy Agent] (OPA). Die Policies und die zugehörigen
1468 Daten erhält der PDP per Download vom PIP und PAP Service. Aus den Input-Daten vom
1469 PEP, den Daten vom PIP und den Policies vom PAP ermittelt der PDP eine Entscheidung
1470 und gibt diese zurück an den PEP.

1471 Neben der OPA Instanz, die die Entscheidung für den PEP trifft (aktive Instanz), ob eine
1472 Kommunikation zulässig ist, implementiert der PDP noch eine zweite OPA Instanz, die mit
1473 einem zweiten OPA Bundle vom PIP und PAP Service arbeitet, aber die getroffenen
1474 Entscheidungen nicht an den PEP zurück gibt. Diese Instanz wird Simulations-Instanz
1475 genannt.

1476 **A_25739 - PDP, Open Policy Agent Instanzen**

1477 Der PDP MUSS zwei Open Policy Agent (OPA) Instanzen bereitstellen, wobei eine Instanz
1478 die Entscheidung für den PEP trifft (aktive Instanz), und eine Instanz eine Entscheidung
1479 trifft, diese aber nicht an den PEP sendet (Simulations-Instanz).

1480 Die OPA Instanzen MÜSSEN gemäß Tabelle OPA_Konfiguration konfiguriert werden.

1481 **Tabelle 8: OPA_Konfiguration**

Konfiguration	Aktive OPA Instanz	Simulations-OPA Instanz
services: - name: <service name> url: <PIP und PAP service>	<service name> Innerhalb der PDP Konfiguration verwendeter Service Name. <PIP und PAP service> Download-Endpunkt des PIP und PAP Servicees entsprechend der verwendeten Umgebung Default: Produktions-Instanz: https://pip-pap.ti-dienste.de Referenz-Instanz: https://pip-pap-ref.ti-dienste.de Test-Instanz: https://pip-pap-test.ti-dienste.de	wie aktive PDP Instanz
bundles: authz: service: <service name> resource: <path> persist: true	<path> Der Pfad wird wie in [pip-pap-service.yaml] beschrieben angegeben. Durch das label latest wird die neueste bundle.tar.gz Datei für die aktive PDP Instanz heruntergeladen. Default: /policies/<TI service>/latest	<path> Durch das label latest-sim wird die neueste bundle.tar.gz Datei für die Simulations-Instanz des PDP heruntergeladen. Default: /policies/<TI service>/latest-sim
bundles: authz: polling: min_delay_seconds: <min> max_delay_seconds: <max>	<min> Minimale Zeit bis zum nächsten Polling. Default: 300 <max> Maximale Zeit bis zum nächsten Polling. Default: 320	wie aktive PDP Instanz

Konfiguration	Aktive OPA Instanz	Simulations-OPA Instanz
<pre> bundles: authz: signing: keyid: <PIP_and_PAP_key> scope: read </pre>	<pre> <PIP_and_PAP_key> Die keyid des Schlüssels, mit dem die OPA Bundles signiert sind. </pre>	wie aktive PDP Instanz
<pre> decision_logs: service: <service name> resource: \${DL_REMOTE_URL} reporting: min_delay_seconds: <min> max_delay_seconds: <max> </pre>	<pre> \${DL_REMOTE_URL} Die URL des Remote Servers, zu dem die decision logs gesendet werden. Dieser Parameter wird per environment Variable übergeben, sodass jeder Betreiber des PDP seinen eigenen Server angeben kann, der die decision logs empfängt. <min> Minimale Verzögerung bis zum nächsten Versand. Default: 300 <max> Maximale Verzögerung bis zum nächsten Versand. Default: 360 </pre>	wie aktive PDP Instanz

1482 [**<=**]

1483 *Hinweis: Änderungen an der Konfiguration sind im Einvernehmen mit der gematik*
 1484 *möglich.*

1485 Der OPA aktualisiert seine Policies und Daten nach dem vorgegebenen Polling-Intervall.
 1486 Jede Download Anfrage enthält immer ein `If-None-Match` Header mit dem hash des
 1487 zuletzt heruntergeladenen bundles (aus dem `ETag` Header der Response). Wenn es keine
 1488 neuen Daten zum Download gibt, dann beantwortet der PIP/PAP Service die Anfrage mit
 1489 304 Not Modified.

1490 Die Bundles sind immer signiert. Der OPA prüft die Signatur mit dem zur konfigurierten
 1491 `keyid` passenden Schlüssel. Gültige Schlüssel und deren `keyid` werden über einen
 1492 Continuous Delivery Workflow aus GitHub geladen.

1493 Die decision logs werden an die vom Betreiber des PDP festgelegte URL gesendet.

1494 **A_25772 - PDP - persistente Speicherung von decision logs**

1495 Der Betreiber eines TI 2.0 Dienstes MUSS einen Service bereitstellen, der OPA decision
 1496 logs vom PDP entgegennimmt und diese persistent für 6 Monate speichert. [**<=**]

1497 **A_25450 - PDP - Policy nur vom gematik PIP und PAP Service**

1498 Der PDP MUSS sicherstellen, dass nur Policies und Daten vom gematik PIP und PAP
 1499 Service importiert werden können. [**<=**]

1500 A_25451 - PDP - Integritätsprüfung der Policies

1501 Der PDP MUSS sicherstellen, dass Policies vom gematik PIP und PAP Service nur nach
1502 einer positiven Integritätsprüfung importiert werden können. [≤]

1503 A_25452 - PDP - Tamper-Proof Protokollierung von Administrationsaktivitäten

1504 Der PDP MUSS ein "Tamper-Proof" Audit-Log von allen administrativen Vorgängen
1505 umsetzen. [≤]

1506 A_25774 - PDP - Löschfristen für Auditeinträge des Admin Audit-Logs

1507 Der PDP MUSS sicherstellen, dass die Löschung eines Auditeintrags den gesetzlichen
1508 Vorgaben entspricht und frühestens nach 12 Monaten erfolgt. [≤]

1509 A_25775 - PDP - Kontrolle des Audit-Logs

1510 Der Betreiber des ZT Clusters MUSS das Audit-Log mindestens alle 3 Monate durch zwei
1511 unabhängige Rollen im Vieraugenprinzip kontrollieren. Diese Rollen DÜRFEN NICHT an
1512 der Administration des ZT Clusters teilnehmen. Bei der Kontrolle ist insbesondere auf
1513 ungewöhnliche, nicht nachvollziehbare oder maliziöse Administratoraktivitäten zu
1514 achten. [≤]

1515 A_25453 - PDP - Transparenz der installierte Policies

1516 Der PDP MUSS sicherstellen, dass die gematik zu jeder Zeit feststellen kann, welche
1517 Policies und welche Policy-Versionen im PDP installiert sind. [≤]

1518 A_25490 - PDP - Sicherheitsmeldung bei Änderungen und Aktualisierung

1519 Der PDP MUSS sicherstellen, dass bei Aktualisierung und Änderungen der Policies oder
1520 PIP-Daten eine Sicherheitsmeldung an das Security Monitoring automatisiert übermittelt
1521 wird. [≤]

1522 A_25771 - PDP - Automatisierte Prüfung nach Policy-Aktualisierungen

1523 Der PDP muss alle 5 Minuten prüfen, ob Aktualisierungen der installierten und
1524 verwendeten Policy/PIP Daten vorhanden sind. [≤]

1525 5.6 Anforderungen an den PIP und PAP Service

1526 Der PIP und PAP Service stellt OPA Bundles für die PDP Instanzen der Zero Trust Cluster
1527 der Dienste der TI 2.0 bereit.

1528 A_25670 - PIP und PAP - Bereitstellung Download-Endpunkt

1529 Der PIP und PAP Service MUSS Download-Endpunkte für OPA Bundles gemäß OpenAPI
1530 Spezifikation [pip-pap-service.yaml] Version 1.0.0 in den folgenden Instanzen
1531 bereitstellen:

1532 Produktions-Instanz: <https://pip-pap.ti-dienste.de>

1533 Referenz-Instanz: <https://pip-pap-ref.ti-dienste.de>

1534 Test-Instanz: <https://pip-pap-test.ti-dienste.de>. [≤]

1535 *Hinweis: Die Bereitstellung der Test-Instanz erfolgt nur während einer Testphase und*
1536 *kann eine andere Version der [pip-pap-service.yaml] unterstützen. Für die Entwicklung*
1537 *und Tests anderer Komponenten wird empfohlen, die Referenz-Instanz zu verwenden.*

1538 A_25671 - PIP und PAP - TLS am Download-Endpunkt

1539 Der PIP und PAP Service MUSS sich beim TLS-Verbindungsaufbau am Download-
1540 Endpunkt gegenüber Clients mit einem Extended Validation TLS-Zertifikat eines
1541 Herausgebers gemäß [CAB-Forum] authentisieren. [≤]

1542 A_25672 - PIP und PAP - Kompatibilität mit OPA Bundles

1543 Der PIP und PAP Service MUSS die Policies und Daten als [OPA Bundle]
1544 bereitstellen. [≤]

A_25680 - PIP und PAP - download path

Der PIP und PAP Service MUSS OPA Bundles mit dem filename = bundle.tar.gz unter dem Pfad `/policies/{application}/{label}` bereitstellen, wobei mindestens die label "latest" und "latest-sim" pro application angeboten werden. [≤]

Hinweis: Siehe [pip-pap-service.yaml]. Unter dem label "latest" werden Bundles für den aktiven PDP bereitgestellt. Unter dem label "latest-sim" werden Bundles für die Simulations-PDP Instanz bereitgestellt.

Der PIP und PAP Service bezieht die OPA Bundles aus einem GitHub Repository der gematik. Die Bundles werden in einem GitOps CI Prozess mit Quality Gates entwickelt und für die PDPs der Zero Trust Cluster bereitgestellt.

A_25673 - PIP und PAP - ETags für OPA Bundles

Der PIP und PAP Service MUSS für jedes zum Download bereitgestellte OPA Bundle in der Response ein ETag Header-Element verwenden, das aus dem Hashwert der bundle.tar.gz Datei besteht. [≤]

Hinweis: Durch die Verwendung des Hashwertes der bundle.tar.gz Datei als ETag wird es möglich, den Download-Endpunkt auf mehrere Server zu verteilen. Wichtig ist nur, dass das ETag auf allen Servern gleich ist, damit bereits erhaltene OPA Bundles nicht erneut heruntergeladen werden.

A_25674 - PIP und PAP - OPA Bundle Signaturprüfung

Der PIP und PAP Service MUSS für alle bereitgestellten OPA Bundles prüfen, ob deren Signatur vorhanden und gültig ist. [≤]

Hinweis: In dieser Spezifikation wird der Prozess, wie die OPA Bundles sicher in den PIP und PAP Service gelangen, nicht festgelegt.

A_25464 - PAP und PIP - Tamper-Proof Protokollierung von Administrationsaktivitäten

Der PIP und PAP Service MUSS ein "Tamper-Proof" Audit-Log von allen administrativen Vorgängen umsetzen. [≤]

A_25777 - PAP und PIP - Löschfristen Auditeinträge des Admin Audit-Logs

Der PIP und PAP Service MUSS sicherstellen, dass die Löschung eines Auditeintrags den gesetzlichen Vorgaben entspricht und frühestens nach 12 Monaten erfolgt. [≤]

A_25778 - PAP und PIP - Kontrolle des Audit-Logs

Der Betreiber des PIP und PAP Services MUSS das Audit-Log mindestens alle 3 Monate durch zwei unabhängige Rollen im Vieraugenprinzip kontrollieren. Diese Rollen DÜRFEN NICHT an der Administration des PAPs oder PIPs teilnehmen. Bei der Kontrolle ist insbesondere auf ungewöhnliche, nicht nachvollziehbare oder maliziöse Administratoraktivitäten zu achten. [≤]

A_25465 - PAP und PIP - Änderungen nur durch berechtigte Nutzer

Der PIP und PAP Service MUSS sicherstellen, dass nur berechtigte Nutzer Änderungen von Policies oder PIP-Daten durchführen können. [≤]

A_25466 - PAP und PIP - Sicherheitsmeldung bei Aktualisierung von Policies oder PIP-Daten

Der PIP und PAP Service MUSS sicherstellen, dass bei der Aktualisierung der Policies oder der PIP-Daten eine Sicherheitsmeldung automatisiert über die von der gematik angebotene Schnittstelle an das TI SIEM System übermittelt wird. [≤]

A_25467 - PAP und PIP - Änderungen nur unter 4 Augen

Der PIP und PAP Service MUSS sicherstellen, dass Änderungen in Policies oder PIP-Daten nur im Vieraugenprinzip durchgeführt werden können. [≤]

1592 **A_25791 - PAP und PIP - Zentrale Verwaltung von Feature-Flags**
1593 PIP und PAP Service MUSS eine zentrale Schnittstelle oder ein System zur Verwaltung
1594 der Feature-Flags bereitstellen.[<=]

1595 **5.7 Anforderungen an den Betrieb der Zero Trust Komponenten**

1596 Die Zero Trust-Komponenten PEP und PDP werden als Kubernetes (k8s) Cluster
1597 betrieben. In einem von der gematik vorgegebenen GitHub Repository werden die
1598 Konfigurationsdateien des k8s Clusters bereitgestellt, mit denen die aktuelle Version des
1599 Clusters erstellt und ausgeführt werden kann. Im Cluster ist zusätzlich ein Cluster
1600 Management Service (Continuous Delivery (CD) Komponente) enthalten, der den
1601 Betriebszustand des Clusters überwacht und regelmäßig prüft, ob eine neuere Version
1602 des Clusters im GitHub Repository verfügbar ist, und ggf. das Cluster automatisch
1603 aktualisiert.

1604 **A_25773 - Zero Trust-Cluster - Nutzung des von gematik bereitgestellten Zero Trust Clusters**

1605 Der Betreiber eines Dienstes der TI 2.0 MUSS den von gematik bereitgestellten Zero
1606 Trust-Cluster verwenden, um den Zugang zum TI 2.0 Dienst zu kontrollieren.[<=]

1608 **A_25776 - Zero Trust-Cluster - Keine eigenmächtige Veränderung der Konfiguration**

1609 Der Betreiber eines Dienstes der TI 2.0 DARF NICHT eigenmächtig die Konfiguration des
1610 Zero Trust-Clusters verändern.[<=]

1612 **5.7.1 Anforderungen für nahtlose Aktualisierungen**

1613 Es ist durch geeignete Maßnahmen sicherzustellen, dass ein unterbrechungsfreier
1614 Betrieb, bzw. die durchgängige Verfügbarkeit zu jeder Zeit gewährleistet ist.

1615 **A_25784 - Zero Trust-Komponenten - Download von Aktualisierungen im Hintergrund**

1616 Die Komponente der Zero Trust Architektur MUSS in der Lage sein, Aktualisierungen im
1617 Hintergrund herunterzuladen, ohne den laufenden Betrieb zu beeinträchtigen.[<=]

1619 **A_25785 - Zero Trust-Komponenten - Nahtloser Übergang zu neuen Versionen**

1620 Die Komponente der Zero Trust Architektur MUSS einen Mechanismus bieten, der einen
1621 nahtlosen Übergang zu neuen Versionen oder Patches ermöglicht, ohne die Verfügbarkeit
1622 für Endbenutzer zu unterbrechen.[<=]

1623 **A_26104 - Zero Trust-Komponenten - Protokollieren von Änderungen**

1624 Die Komponente der Zero Trust Architektur MUSS jede Änderung protokollieren,
1625 einschließlich des Zeitpunkts der Änderung und des Administrators, der die Änderung
1626 vorgenommen hat.[<=]

1627 **A_25786 - Zero Trust-Komponenten - Abschluss von Transaktionen vor Aktualisierung**

1628 Die Komponente der Zero Trust Architektur MUSS sicherstellen, dass alle aktuellen
1629 Transaktionen und Anfragen abgeschlossen oder ordnungsgemäß übernommen werden,
1630 bevor ein Update finalisiert wird.[<=]

1632 **A_25787 - Zero Trust-Komponenten - Gewährleistung der Systemintegrität**
1633 **während Aktualisierungen**

1634 Die Komponente der Zero Trust Architektur MUSS während des gesamten
1635 Aktualisierungsprozesses die Systemintegrität und Sicherheitsrichtlinien
1636 aufrechterhalten. [≤]

1637 **A_25788 - Zero Trust-Komponenten - Unterstützung von Rollbacks**

1638 Die Komponente der Zero Trust Architektur MUSS die Fähigkeit besitzen, zu einer
1639 stabilen Vorversion zurückzukehren, sollte eine Aktualisierung fehlerhaft sein oder
1640 abgebrochen werden müssen. [≤]

1641 **A_25789 - Zero Trust-Komponenten - Schnelle Rollback-Durchführung**

1642 Die Komponente der Zero Trust Architektur MUSS Rollbacks schnell und ohne manuelle
1643 Eingriffe durchführen können. [≤]

1644 **5.7.2 Anforderungen für Steuerung durch Feature-Flags**

1645 **A_25790 - Zero Trust-Komponenten - Aktivierung/Deaktivierung von**
1646 **Funktionen in Echtzeit**

1647 Die Komponente der Zero Trust Architektur MUSS Funktionen oder Verhaltensweisen zur
1648 Laufzeit durch Feature-Flags aktivieren oder deaktivieren können, ohne dass ein Neustart
1649 erforderlich ist. [≤]

1650 **A_25792 - Zero Trust-Komponenten - Protokollierung von Feature-Flag-**
1651 **Änderungen**

1652 Die Komponente der Zero Trust Architektur MUSS jede Änderung an Feature-Flags
1653 „Tamper-Proof“ protokollieren, einschließlich des Zeitpunkts der Änderung und des
1654 Administrators, der die Änderung vorgenommen hat. [≤]

1655 *Hinweis: Hier sollte das Protokoll mit dem Tamper-Proof Audit-Log in*
1656 *A_25452 kombiniert werden.*

1657 **A_25793 - Zero Trust-Komponenten - Zugriffskontrolle für Feature-Flag-**
1658 **Verwaltung**

1659 Die Komponente der Zero Trust Architektur MUSS Zugriffskontrollen implementieren, um
1660 sicherzustellen, dass nur autorisierte Benutzer Feature-Flags ändern können. [≤]

1661 **5.7.3 Anforderungen zur Überwachung des Betriebsstatus**

1662 **A_25794 - Zero Trust-Komponenten - Implementierung von Health Checks**

1663 Die Komponente der Zero Trust Architektur MUSS Health Checks implementieren, um
1664 ihren aktuellen Zustand und ihre Verfügbarkeit zu überwachen. [≤]

1665 **A_25797 - Zero Trust-Komponenten - Verfügbarkeit der Health Checks für**
1666 **gematik Monitoring**

1667 Die Komponente der Zero Trust Architektur MUSS die Schnittstellen zu Health Checks
1668 dem gematik Monitoring zur Verfügung stellen. [≤]

1669 **A_25795 - Zero Trust-Komponenten - Automatische Antwort auf Health Check**
1670 **Anfragen**

1671 Die Komponente der Zero Trust Architektur MUSS automatisch auf Health Check
1672 Anfragen antworten können, um ihre Funktionalität und Verfügbarkeit zu
1673 bestätigen. [≤]

A_25796 - Zero Trust-Komponenten - Bereitstellung von Zustandsinformationen

Die Komponente der Zero Trust Architektur MUSS detaillierte Zustandsinformationen als Teil ihrer Health Check Antworten bereitstellen, einschließlich - aber nicht beschränkt auf - Betriebszeit, letzte erfolgreiche Transaktion und eventuelle Fehlerzustände. [≤]

A_25798 - Zero Trust-Komponenten - Regelmäßige Selbstüberprüfung

Die Komponente der Zero Trust Architektur MUSS in der Lage sein, regelmäßige Selbstüberprüfungen durchzuführen, um interne Funktionen und Abhängigkeiten zu verifizieren und sicherzustellen, dass sie korrekt arbeiten. [≤]

A_25799 - Zero Trust-Komponenten - Protokollierung von Health Check Ergebnissen

Die Komponente der Zero Trust Architektur MUSS die Ergebnisse der Health Checks protokollieren, um eine Historie ihrer Betriebszustände und eventuell aufgetretener Probleme zu erhalten. [≤]

A_25800 - Zero Trust-Komponenten - Benachrichtigung bei Fehlern

Die Komponente der Zero Trust Architektur MUSS im Falle eines fehlgeschlagenen Health Checks oder der Erkennung eines kritischen Zustandes automatisch eine Benachrichtigung an ein vordefiniertes Management- oder Monitoring-System senden. [≤]

Der Hersteller der Zero Trust-Komponenten wird im Rahmen seiner Entwicklungs- und Wartungstätigkeit die Aufgaben eines Third (3rd) Level Supports gewährleisten. First- (1st) und Second- (2nd) Level Supporttätigkeiten fallen zukünftig in den Verantwortungsbereich des Dienstes, in welchen die Zero Trust-Komponenten eingebettet werden.

Hinweis: Weitere spezifikatorische Regelungen finden zu einem späteren Zeitpunkt statt.

5.7.4 Betriebliche Schnittstellendefinition der Zero Trust-Komponenten

Die Zero-Trust Komponenten PEP und PDP stellen Endpunkte zur Verfügung, um die grundlegende Funktionalität, eingebettet in einen Service, zu gewährleisten. Jeder Dienst, der die Zero-Trust Komponenten betreibt, stellt damit folgende Endpunkte für einen Nutzer zur Verfügung. Die Tabelle orientiert sich an den Schnittstellendefinition aus [gemKPT_Betr].

Table 1 Tab_gemF_Zero-Trust_Schnittstellendefinition_PEPPDP

A-ID	Schnittstellen::Operation / Anwendungsfall	Beschreibung
A01	-	- vorbelegt für Verfügbarkeitsberechnung -
A02	/.well-known/	Abruf gültiger Autorisierungsserver
A03	GET /nonce/	Nonce abrufen
A04	POST /token <JWT Client Assert>	Autorisierung ohne Refresh Token
A05	POST /token <Refresh Token>	Autorisierung mit Refresh Token

Zusätzlich werden mittels zentralen Komponenten weitere Endpunkte zur Verfügung gestellt (PIP/PAP), welche von den Zero-Trust Komponenten abgefragt werden, um beispielsweise aktualisierte Policy-Informationen abzuholen. Folgende Endpunkte werden nachfolgend definiert.

Table 2 Tab_gemF_Zero-Trust_Schnittstellendefinition_PIPPAP

A-ID	Schnittstellen::Operation / Anwendungsfall	Beschreibung
A01	-	- vorbelegt für Verfügbarkeitsberechnung -
A02	GET /policies/{application}/{label}	Abruf der Policy eines Dienstes

Beim Erfassen der Daten des Funktionsaufrufs GET /policies/{application}/{label} muss der PIP/PAP-Dienst die Werte für {application} und {label} zusätzlich mit erfassen. Die Systematik zur Betriebsdatenerfassung wird zu einem späteren Zeitpunkt konkret festgelegt, orientiert sich aber an den Festlegungen zur Betriebsdatenerfassung Version 2 der [gemSpec_Perf].

5.8 Anforderungen an den Test der Zero-Trust Komponenten

Durch einen Shift-Left-Ansatz werden Testmaßnahmen von Anfang an für alle spezifizierten Komponenten und Dienste in der Produktentwicklung verankert. Hierbei arbeiten alle Beteiligten eng zusammen und machen Ihre Testmaßnahmen transparent.

Es wird angestrebt, so früh wie möglich durch PoCs und Durchstichtests die Umsetzbarkeit der Spezifikationen und die Anwendungsszenarien der Nutzer nachzuweisen.

5.8.1 Testartefakte

A_26081 - Format von Testszenarien

Der Hersteller der Zero Trust Komponenten MUSS alle für den funktionalen Test benötigten Testszenarien und Testfälle nach den BDD-Prinzip erstellen. [\leq]

Hinweis: Eine genaue Definition zum Format der Testszenarien und Testfälle erfolgt später in Abstimmung mit dem Hersteller.

A_26083 - Effiziente Entwicklung von Testartefakten

Der Hersteller der Zero Trust Komponenten MUSS alle für den Test und funktionsfähige Testumgebungen benötigten Testartefakte (z.B. Testsuiten, Testfälle mit Testszenarien, Testdaten, Mocks und Simulatoren) entwickeln. [\leq]

Hinweis: Eine genaue Definition der benötigten Testartefakte erfolgt später in Abstimmung mit dem Hersteller.

A_26089 - Automatisierung von Testartefakten

Der Hersteller der Zero Trust Komponenten MUSS alle Testartefakte so entwickeln und bereitstellen, dass sie, soweit möglich, eine automatisierte Testausführung ermöglichen. [\leq]

A_26084 - Freie Nutzung und Weiterentwicklung von Testartefakten

Der Hersteller der Zero Trust Komponenten MUSS die entwickelten Testartefakte kosten- und lizenzfrei zur Nutzung und Weiterentwicklung zur Verfügung stellen. [\leq]

A_26085 - Support für Testartefakte

Der Hersteller der Zero Trust Komponenten MUSS einen Support für die von ihm entwickelten Testartefakte anbieten. [\leq]

Hinweis: Eine genaue Definition von Art und Umfang des zu leistenden Support erfolgt später in Abstimmung mit dem Hersteller.

5.8.2 Testtreiberschnittstelle und Testunterstützung

Die hier spezifizierten Komponenten und Dienste dienen der Absicherung von Produkttypen einer Fachanwendung. Da sie elementare Sicherheitsfunktionen umsetzen, stehen sie ggf. einer Implementierung und Testung der zu schützenden Fachlichkeit in **nicht**-produktiven Umgebungen im Wege. Daher ist es sinnvoll, in den entsprechenden Zero Trust-Komponenten eine Testtreiber-Schnittstelle bzw. einen Testmodus zu implementieren, der die Umgehung der Sicherheitsmechanismen unter bestimmten Rahmenbedingungen (z. B. Routing in ein Testsystem, wenn Testpolicy aktiv etc.) ermöglicht.

A_26086 - Bereitstellen einer Testtreiberschnittstelle

Der Hersteller der Zero Trust Komponenten MUSS in jeder Komponente eine Testtreiberschnittstelle bereitstellen, die den automatisierten Nachweis aller funktionalen Anforderungen an die jeweilige Komponente in einem Blackbox-Test, d.h. einem Test an den Außenschnittstellen, ermöglicht, die sonst über die normale API der Komponente nicht oder nur mit einem erhöhten Aufwand testbar wäre. [\leq]

Hinweis: Eine genaue Abstimmung zu Art und Umfang der Testtreiberschnittstelle in den jeweiligen Komponenten erfolgt später in Abstimmung mit dem Hersteller.

A_26087 - Keine Testtreiberschnittstelle in produktiv einsetzbaren Komponenten

Der Hersteller der Zero Trust Komponenten MUSS sicherstellen, dass die Testtreiberschnittstelle in produktiv einsetzbaren Komponenten nicht enthalten ist. [\leq]

A_26088 - API und Dokumentation der Testtreiberschnittstelle

Der Hersteller der Zero Trust Komponenten MUSS die API der Testtreiberschnittstelle und ihre Dokumentation frei verfügbar bereitstellen [\leq]

A_26092 - Keine Hardwareabhängigkeiten bei Komponenten in Testumgebungen

Der Hersteller der Zero Trust Komponenten MUSS seine Komponenten in Testumgebungen so entwickeln und bereitstellen, dass Hardwareabhängigkeiten, z.B. zu einem HSM deaktiviert und durch Simulatoren oder Mocks ersetzt werden können. [\leq]

A_26093 - Kenntnis der privaten Schlüsseln in Testumgebungen

Der Hersteller der Zero Trust Komponenten MUSS in Testumgebungen sicherstellen, dass die verwendeten privaten Schlüssel bekannt sind. Dies kann z.B. durch das Einbringen von bekannten Testschlüsseln als privaten Schlüsseln erfolgen, oder durch das Auslesen der privaten Schlüssel aus dem Schlüsselspeicher. [\leq]

A_26094 - Mitwirken bei übergreifenden Testmaßnahmen

Der Hersteller der Zero Trust Komponenten MUSS bei der Vorbereitung und Durchführung von übergreifenden Testmaßnahmen wie z.B. Durchstichtests oder Connectathons mitwirken, z.B. durch Bereitstellung und Konfiguration der benötigten Komponenten und Testumgebungen oder durch Bereitstellen von Support. [\leq]

5.8.3 Bereitstellung der Testkomponenten und Testartefakte

Die hier spezifizierten Komponenten und Dienste haben keinen Selbstzweck, sondern kommen in der Realisierung von Produkttypen einer Fachanwendung zum Einsatz. Sie werden "as Code" bereitgestellt, und in Build-Pipelines der Komponenten und Produkttypen der Fachanwendung eingebettet. Weiterhin soll die Testausführung automatisiert werden. Dadurch wird ein automatisches und regelmäßiges Deployment in Testumgebungen im Sinne eines Continuous Testings ermöglicht.

A_26096 - Bereitstellung von Testkomponenten und Testartefakten

Der Hersteller von Zero Trust Komponenten MUSS Testkomponenten und Testartefakte als signierte Artefakte (wie z.B. ein Container-Image) bereitstellen. [\leq]

Hinweis: Eine genaue Definition zum Format der signierten Artefakte erfolgt später in Abstimmung mit dem Hersteller.

A_26097 - Verwendbarkeit von Testkomponenten und Testartefakten in automatisierten CI/CD-Pipelines

Der Hersteller von Zero Trust Komponenten MUSS Testkomponenten und Testartefakte so entwickeln und bereitstellen, dass sie in automatisierten CI/CD-Pipelines verwendet werden können. [\leq]

A_26098 - Konfigurierbarkeit von Testkomponenten und Testartefakten

Der Hersteller von Zero Trust Komponenten MUSS Testkomponenten und Testartefakte so konfigurierbar entwickeln und bereitstellen, dass diese in allen benötigten Testumgebungen eingesetzt werden können. [\leq]

Hinweis: Eine genaue Definition aller benötigten Testumgebungen erfolgt später in Abstimmung mit dem Hersteller.

5.8.4 Testumgebungen und Quality Gates

Um die Qualität einer Zero Trust Komponente zu messen, muss diese in verschiedenen Testumgebungen (z.B. für die Entwicklung, den Produkttest, den Integrationstest) validiert werden und müssen die für die jeweilige Testumgebung vorgesehenen Quality Gates bestehen. Quality Gates bestehen zum Beispiel im erfolgreichen Durchführen von Testsuiten oder im Erbringen geforderter Nachweise. Dies soll nach Möglichkeit automatisiert erfolgen.

A_26099 - Bestehen von Quality Gates

Der Hersteller von Zero Trust Komponenten MUSS das Quality Gate einer Testumgebung bestehen, wenn für die Testumgebung eines definiert wurde. [\leq]

Hinweis: Eine genaue Definition, welche Testumgebung und welche Quality Gates dazugehören, erfolgt später in Abstimmung mit dem Hersteller.

A_26100 - Labeln von Komponenten

Der Hersteller von Zero Trust Komponenten MUSS seine Komponenten mit dem zur Testumgebung gehörenden Label versehen, wenn er deren Quality Gate erfolgreich durchlaufen hat. Dadurch wird die Qualität der Komponenten im Produkt-Lebenszyklus transparent gemacht und damit auch, für welche Zwecke und in welchen Umgebungen sie eingesetzt werden darf. [\leq]

1829

6 Dokumentenhaushalt

1830 Dieses Dokument hat die nachfolgenden Auswirkungen auf den Dokumenten- und
1831 Anforderungshaushalt der Telematikinfrastruktur.

1832 6.1 Neue Dokumente

1833 Dieses Dokument wird zunächst als "Sicherheitsfeature" eines TI 2.0 Dienstes eingeführt.
1834 Die Anforderungen sind als übergreifende Spezifikation zu betrachten, die erst im
1835 Kontext eines konkreten TI 2.0 Dienstes wirksam werden. Daher erfordert die
1836 Realisierung des Zero Trusts immer eine zusätzliche TI 2.0 dienstspezifische
1837 Spezifikation, wobei die hier formulierten Anforderungen Eingang in die Produkt- und
1838 Anbietertyp-Steckbriefe des entsprechenden TI 2.0 Dienstes finden.

1839 6.2 Übersicht betroffener Dokumente

1840 Aus dieser Spezifikation ergeben sich zunächst keine direkten Änderungsbedarfe an
1841 anderen Dokumenten.

1842 6.3 Übersicht Produkt- und Anbietertypen

1843 Die hier spezifizierten Komponenten und Dienste stellen keine isoliert zulassungsfähigen
1844 Produkttypen dar. Sie liefern einen Anforderungshaushalt für anwendungsspezifische
1845 Komponenten und Dienste, die dann zusammen einen Produkt- bzw. Anbietertyp einer
1846 konkreten Fachanwendung bzw. der TI als Plattform bilden.

1847

7 Beispiele und Referenzimplementierungen

1848 Die gematik stellt API-Spezifikationen und Proof-of-Concept-Implementierungen im
1849 Internet zur freien Verfügung.

1850 Das Projekt <https://dsr.gematik.solutions> demonstriert eine Attestation mobiler
1851 Anwendungen auf gängigen mobilen Betriebssystemplattformen.

1852 Im github-Projekt <https://github.com/gematik/spec-t20r> werden die
1853 Schnittstellenspezifikationen der hier spezifizierten Zero Trust-Komponenten
1854 veröffentlicht.

1855 Die folgenden beiden Projekte <https://github.com/gematik/zero-lab> und
1856 <https://github.com/gematik/zero-lab-apple> demonstrieren die Anwendungsfälle zur
1857 Clientregistrierung auf Apple- und Android-Geräten.

1858 **8 Anhang A – Verzeichnisse**

1859 **8.1 Abkürzungen**

1860 **Tabelle 9: Im Dokument verwendete Abkürzungen**

Kürzel	Erläuterung
BDE	Betriebsdatenerfassung
DSR	Device Security Rating
eGK	elektronische Gesundheitskarte
GesundheitsID	Digitale Identität
IdP	Identity Provider
IDS	Intrusion Detection System
ISMS	Informationssicherheitsmanagementsystem
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
SIEM	Security Information and Event Management
TI	Telematikinfrastruktur
TI-ITSM	IT-Service-Management der TI

1861 **8.2 Abbildungsverzeichnis**

1862 |Abbildung 1: NIST Zero Trust Referenzarchitektur..... 11

1863 |Abbildung 2 : Zero_Trust_Architektur_der_TI_2.0..... 12

1864 |Abbildung 3 SM-B_Authentisierung_mit_DPoP 40

1865 |

8.3 Tabellenverzeichnis

Tabelle 1: Statische Eigenschaften Clientsysteme auf Hersteller-/Herausgeber- /Anbiiterebene	17
Tabelle 2 : Eigenschaften Clientsysteme auf Instanzebene (pro Installation	18
Tabelle 3: Verwendete Device Claims für Android-Geräte	19
Tabelle 4: Verwendete Device Claims für iOS-Geräte	19
Tabelle 5: PEP Authorization Server - Plugin-Schnittstelle Application Authorization Backend.....	37
Tabelle 6: PEP http Proxy - Zusätzliche http-Header	44
Tabelle 7 PEP_Konfigurations-Parameter.....	45
Tabelle 8: OPA_Konfiguration.....	46
Tabelle 9: Im Dokument verwendete Abkürzungen	58
Tabelle 10: Referenzierte Dokumente der gematik	59
Tabelle 11: Weitere Referenzen.....	60

8.4 Referenzierte Dokumente

8.4.1 Dokumente der gematik

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

Tabelle 10: Referenzierte Dokumente der gematik

[Quelle]	Herausgeber: Titel
[gemAPI_ZT]	gematik: OpenAPI Schnittstellenspezifikation Zero Trust https://github.com/gematik/spec-t20r
[gemGlossar]	gematik: Glossar der Telematikinfrastruktur
[gemKPT_Betr]	gematik: Betriebskonzept Online-Produktivbetrieb
[gemKPT_Zero_Trust]	gematik: Feinkonzept Zero Trust https://fachportal.gematik.de/fileadmin/Fachportal/Downloadcenter/gemKPT_Zero_Trust_V1.0.0.pdf
[gemSpec_DS_Hersteller]	gematik: Spezifikation Datenschutz- u. Sicherheitsanforderungen der TI an Hersteller https://gemspec.gematik.de/docs/gemSpec/gemSpec_DS_Hersteller/gemSpec_DS_Hersteller_V1.5.1/

[Quelle]	Herausgeber: Titel
[gemSpec_IDP_Dienst]	gematik: Spezifikation Identity Provider-Dienst https://gemspec.gematik.de/docs/gemSpec/gemSpec_IDP_Dienst/gemSpec_IDP_Dienst_V1.6.0/
[gemSpec_IDP_Sek]	gematik: Spezifikation Sektoraler Identity Provider https://gemspec.gematik.de/docs/gemSpec/gemSpec_IDP_Sek/gemSpec_IDP_Sek_V2.3.0/
[gemSpec_Krypt]	gematik: Übergreifende Spezifikation: Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur https://gemspec.gematik.de/docs/gemSpec/gemSpec_Krypt/gemSpec_Krypt_V2.31.0/
[gemSpec_Perf]	gematik: Übergreifende Spezifikation Performance und Mengengerüst TI-Plattform
[pip-pap-service.yaml]	gematik: OpenAPI Schnittstellenspezifikation für Policy Information Point und Policy Administration Point API https://raw.githubusercontent.com/gematik/spec-t20r/develop/src/openapi/pip-pap-api.yaml

1886 8.4.2 Weitere Referenzen

1887 Tabelle 11: Weitere Referenzen

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[Android Platform Security Model]	The Android Platform Security Model (2023) https://research.google/pubs/the-android-platform-security-model/
[Apple Platform Security Guide]	Einführung in die Sicherheit der Apple-Plattformen https://support.apple.com/de-de/guide/security/seccd5016d31/web
[BSI-Grundschutz]	IT-Grundschutz - Informationssicherheit mit System https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/it-grundschutz_node.html

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[BSI-Prüfvorschrift]	<p>Prüfvorschrift für den Produktgutachter des „ePA-Frontend des Versicherten“ und des „E-Rezept-Frontend des Versicherten.“</p> <p>https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/DigitaleGesellschaft/Pruefvorschrift_Produktgutachter_ePA-Frontend.html</p>
[CAB-Forum]	<p>Certification Authority Browser Forum (CA/Browser Forum)</p> <p>https://cabforum.org/</p>
[CAPEC OWASP]	<p>CAPEC: OWASP Related Patterns</p> <p>CAPEC - CAPEC-659: OWASP Related Patterns (Version 3.9) (mitre.org)</p>
[ExpBackoff]	<p>Exponential Backoff</p> <p>https://en.wikipedia.org/wiki/Exponential_backoff</p>
[GPI-API]	<p>Google Play Integrity API</p> <p>https://developer.android.com/google/play/integrity/standard</p>
[ISMS]	<p>Spezifikation Datenschutz- und Sicherheitsanforderungen der TI an Anbieter (Abschnitt 3.3)</p> <p>https://gemspec.gematik.de/docs/gemSpec/gemSpec_DS_Anbieter/latest/#3.3</p>
[OPA Bundle]	<p>Open Policy Agent, Bundles</p> <p>https://www.openpolicyagent.org/docs/latest/management-bundles/</p>
[Open Policy Agent]	<p>Open Policy Agent</p> <p>https://www.openpolicyagent.org/docs/latest/</p>
[OWASP-Top-10-Risiken]	<p>OWASP Top 10</p> <p>https://owasp.org/www-project-top-ten/</p>
[TR-03161]	<p>BSI TR-03161 Anforderungen an Anwendungen im Gesundheitswesen</p> <p>https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03161/tr-03161.html</p>
[RFC2119]	<p>Key words for use in RFCs to Indicate Requirement Levels</p> <p>https://datatracker.ietf.org/doc/html/rfc2119</p>
[RFC2986]	<p>PKCS #10: Certification Request Syntax Specification</p> <p>https://datatracker.ietf.org/doc/html/rfc2986</p>
[RFC6749]	<p>The OAuth 2.0 Authorization Framework</p> <p>https://datatracker.ietf.org/doc/html/rfc6749</p>

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[RFC7231]	Hypertext Transfer Protocol (http/1.1): Semantics and Content https://datatracker.ietf.org/doc/html/rfc7231
[RFC7521]	Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants https://datatracker.ietf.org/doc/html/rfc7521
[RFC7523]	JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants https://datatracker.ietf.org/doc/html/rfc7523
[RFC7636]	Proof Key for Code Exchange by OAuth Public Clients https://datatracker.ietf.org/doc/html/rfc7636
[RFC8555]	Automatic Certificate Management Environment (ACME) https://datatracker.ietf.org/doc/html/rfc8555#section-6.5.1
[RFC8705]	OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens https://datatracker.ietf.org/doc/html/rfc8705
[RFC9449]	OAuth 2.0 Demonstrating Proof of Possession (DPoP) https://datatracker.ietf.org/doc/html/rfc9449
[Verified Boot]	Verifizierter Start https://source.android.com/docs/security/features/verifiedboot?hl=de

1888

1889