Telematikinfrastruktur 2.0

Spezifikation
Zero Trust

Version:	1.0.0_CC
Revision:	1045511
Stand:	15.11.2024
Status:	in Bearbeitung
Klassifizierung:	
Referenzierung:	gemSpec_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
			Übernahme aus Feature-Dokument [gemF_Zero-Trust] V1.0.1	gematik
1.0.0_CC	15.11.2024		zur Abstimmung freigegeben	gematik

Inhaltsverzeichnis

1 Einordnung des Dokuments 6

1.1 Zielsetzung 6

1.2 Zielgruppe 6

1.3 Abgrenzungen 6

1.4 Methodik 7

1.4.1 Anforderungen 7

2 Features und Epics 8

2.1 Clientregistrierung 8

2.1.1 Wiedererkennung bekannter Clients 8

2.1.2 Device Security Rating 8

2.2 Policy Enforcement 8

2.2.1 Zugriffsschutz 9

2.2.2 http Proxy 9

2.3 Decide from Policies 9

2.3.1 Maschinenlesbare Zugriffsregeln 9

2.3.2 Ein reproduzierbares Ja/Nein/Vielleicht 9

2.3.3 Policies nach Betroffenheit 9

2.4 Policy-Information und -Administration 9

2.4.1 Policy-Verwaltung 10

2.4.2 Monitoring 10

2.5 Client Authorization 10

2.5.1 Autorisierung auf Basis von Policy-Entscheidungen 10

2.5.2 Client Authentication 10

3 Einordnung in die TI 2.0 11

4 Technisches Konzept 14

4.1 Zero Trust-Cluster 14

4.2 Policy Enforcement Point (PEP) 14

4.3 Policy Decision Point (PDP) 15

4.4 Trust Client 16

4.5 Policy-Information und -Administration 16

4.5.1 Policy Information Point (PIP) 16

4.5.2 Policy Administration Point (PAP) 16

4.6 Clientregistrierung 17

4.7 Monitoring 20

4.7.1 Security Information and Event Management (SIEM): 20

4.7.2 Shared Signals 21

4.7.3 Telemetrie, Monitoring und Logging 21

4.8 Zusammenspiel mit Identity Provider 21

4.9 Fachdienst-Backend....22

5 Spezifikation....23

5.1 Übergreifende Anforderungen für Datenschutz und Sicherheit....23
5.1.1 Sicherheits- und Datenschutzanforderungen an Logging und Monitoring....25
5.1.2 Sicherheits- und Datenschutz-Anforderungen an das Security Monitoring....25
5.1.3 Sicherheits- und Datenschutz-Anforderungen an die Verarbeitung von Daten mit dem Schutzbedarf "sehr hoch"....28
5.1.4 Sicherheits- und Datenschutz Anforderungen an dem Trust Client in FdVs....29

5.2 Anforderungen an Clientsysteme und Trust Clients....29
5.2.1 Hersteller und Herausgeber....29
5.2.2 Verbindungsaufbau....30
5.2.3 Clientregistrierung....31
5.2.4 Nutzerauthentifizierung....32
5.2.5 Session Management....33
5.2.6 Liste der HTTP-Statuscodes....33

5.3 Zero Trust-Cluster....36

5.4 Anforderungen an Policy Enforcement Points....37
5.4.1 PEP http Proxy....38
5.4.2 Sicherheits- und Datenschutz-Anforderungen an den PEP....41

5.5 Anforderungen an den Policy Decision Point....41
5.5.1 Policy Engine....41
5.5.2 PDP Client Registry....44
5.5.2.1 Sicherheits- und Datenschutz-Anforderungen an die PDP Client Registry. .45
5.5.3 PDP Relying Party....46
5.5.4 PDP Authorization Server....47
5.5.4.1 Service Discovery....49
5.5.4.2 Ablauf der SM(C)-B Authentifizierung mit DPoP....51
5.5.4.3 Ablauf der Authentifizierung bei Dienst-zu-Dienst Kommunikation....56
5.5.5 PDP Datenbank....57
5.5.6 Sicherheits- und Datenschutzanforderungen an den PDP....58
5.5.7 Konfiguration....58

5.6 Anforderungen an den PIP und PAP Service....59

5.7 Anforderungen an den Betrieb der Zero Trust-Komponenten....60
5.7.1 Anforderungen für nahtlose Aktualisierungen....61
5.7.2 Anforderungen für Steuerung durch Feature-Flags....61
5.7.3 Anforderungen zur Überwachung des Betriebsstatus....61
5.7.4 Leistungs-Anforderungen....62
5.7.5 Betriebliche Schnittstellendefinition der Zero Trust-Komponenten....63

5.8 Anforderungen an Dienste der TI....64

5.9 Anforderungen an den Test der Zero Trust-Komponenten....64

6 Beispiele und Referenzimplementierungen....65

7 Anhang A – Verzeichnisse....66

7.1 Abkürzungen....66

7.2 Glossar....67

7.3 Abbildungsverzeichnis....69

7.4 Tabellenverzeichnis....69

7.5 Referenzierte Dokumente.....70
7.5.1 Dokumente der gematik.....70
7.5.2 Weitere Referenzen.....71

1 Einordnung des Dokuments

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

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

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

1.1 Zielsetzung

Ziel des Dokuments ist die Sammlung der technischen, betrieblichen und testrelevanten Anforderungen an Clients, Komponenten und Dienste, die Zero Trust-Aspekte beinhalten oder nutzen.

Das Ziel des Zero Trust-Ansatzes besteht darin, die IT-Sicherheitslandschaft grundlegend zu transformieren, um den Schutz von Daten, Anwendungen und Systemen vor modernen Bedrohungen und Angreifern zu gewährleisten. Im Gegensatz zu traditionellen Sicherheitsmodellen, die auf dem Konzept eines sicheren Perimeters basieren, setzt Zero Trust auf die Annahme, dass keine Benutzer oder Systeme, unabhängig von ihrem Standort innerhalb oder außerhalb des Netzwerks, von Natur aus vertrauenswürdig sind.

1.2 Zielgruppe

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

1.3 Abgrenzungen

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

1.4 Methodik

1.4.1 Anforderungen

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

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

Anforderungen werden im Dokument wie folgt dargestellt:
<AFO-ID> - <Titel der Afo>
Text / Beschreibung
[<=]

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

2 Features und Epics

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

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

2.1 Clientregistrierung

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

2.1.1 Wiedererkennung bekannter Clients

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

2.1.2 Device Security Rating

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

2.2 Policy Enforcement

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

2.2.1 Zugriffsschutz

Das Policy Enforcement soll als eine Art Gatekeeper bzw. Türsteher den Zugriff auf Schnittstellen von Backendservices durch beliebige Clients durchsetzen. Grundlage ist das Vertrauen in eine Policy-Entscheidung durch eine Komponente zur Auswertung eines Regelwerks.

2.2.2 http Proxy

Der http Proxy stellt sicher, dass nur Requests mit gültigem Access Token sowie bestandenen zusätzlichen Prüfungen an den Ressource Server weitergeleitet werden. Welche Prüfungen zusätzlich erfolgen, wird über Attribute im Access Token gesteuert.

2.3 Decide from Policies

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

2.3.1 Maschinenlesbare Zugriffsregeln

Die Menge (potentiell) geltender Regeln zur Absicherung des Zugriffs auf Daten und Dienste formt ein Set von Policies. Um im Fall eines Zugriffsversuchs schnell entscheiden zu können, sollen diese Regeln maschinenlesbar definiert sein. Die Regeln sollen zusätzlich menschenlesbar sein, um die Entwicklung und Wartung der Regeln zu vereinfachen.

2.3.2 Ein reproduzierbares Ja/Nein/Vielleicht

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

2.3.3 Policies nach Betroffenheit

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

2.4 Policy-Information und -Administration

Die Aufgabe des Policy Information Point (PIP) ist es, relevante Attribute und Informationen zur Entscheidungsfindung (Daten) zu liefern, während der Policy Administration Point (PAP) für die Verwaltung und Bereitstellung der Richtlinien (Policies) verantwortlich ist.

2.4.1 Policy-Verwaltung

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

2.4.2 Monitoring

Durch ein Monitoring von Betriebsparametern und Telemetrie-Daten wird die Durchsetzung von Policies sowie die Auswirkung möglicher Policy-Änderungen transparent.

2.5 Client Authorization

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

Für die Dienst zu Dienst Kommunikation ist es ebenso erforderlich den anfragenden Dienst zu identifizieren, um eine Zugriffsentscheidung treffen zu können.

2.5.1 Autorisierung auf Basis von Policy-Entscheidungen

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

2.5.2 Client Authentication

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

Kann ein Mensch oder Client nicht sicher authentifiziert werden oder wird der Authentifizierung zeitlich oder anderweitig nicht vertraut oder passen die Umgebungs- bzw. die den Aufruf begleitenden Parameter nicht zum Vertrauen in die Authentifizierung, wird eine erneute Authentifizierung als erforderlich angesehen ("Step-Up-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 eines Schutzes 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 der Clients und Backendservices werden Anforderungen erhoben, die in konkreten Softwarekomponenten innerhalb dieser Akteure umzusetzen sind. 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

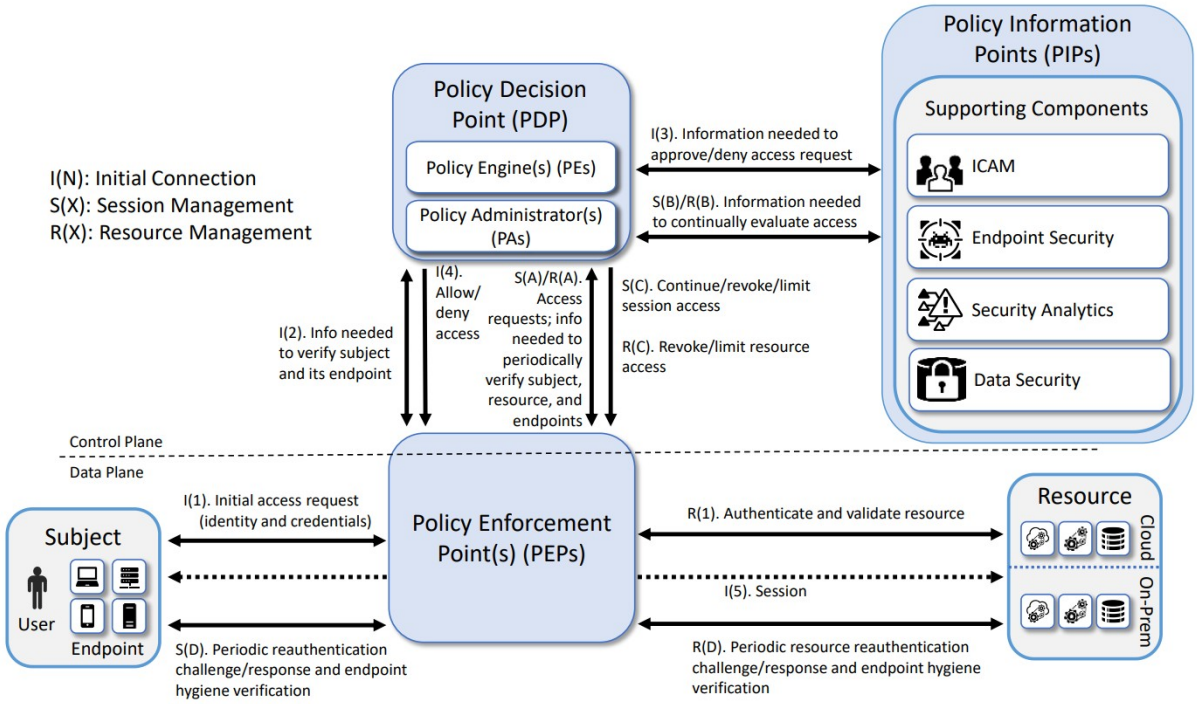

Abbildung 1: NIST Zero Trust-Referenzarchitektur

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 Trust Client auf Daten eines TI 2.0 Dienstes zu. Das folgende Bild zeigt eine Übersicht der beteiligten Komponenten in der Vernetzung zwischen einem Trust Client (links grün) und einem Backendservice (rechts grün: Ressource Server).

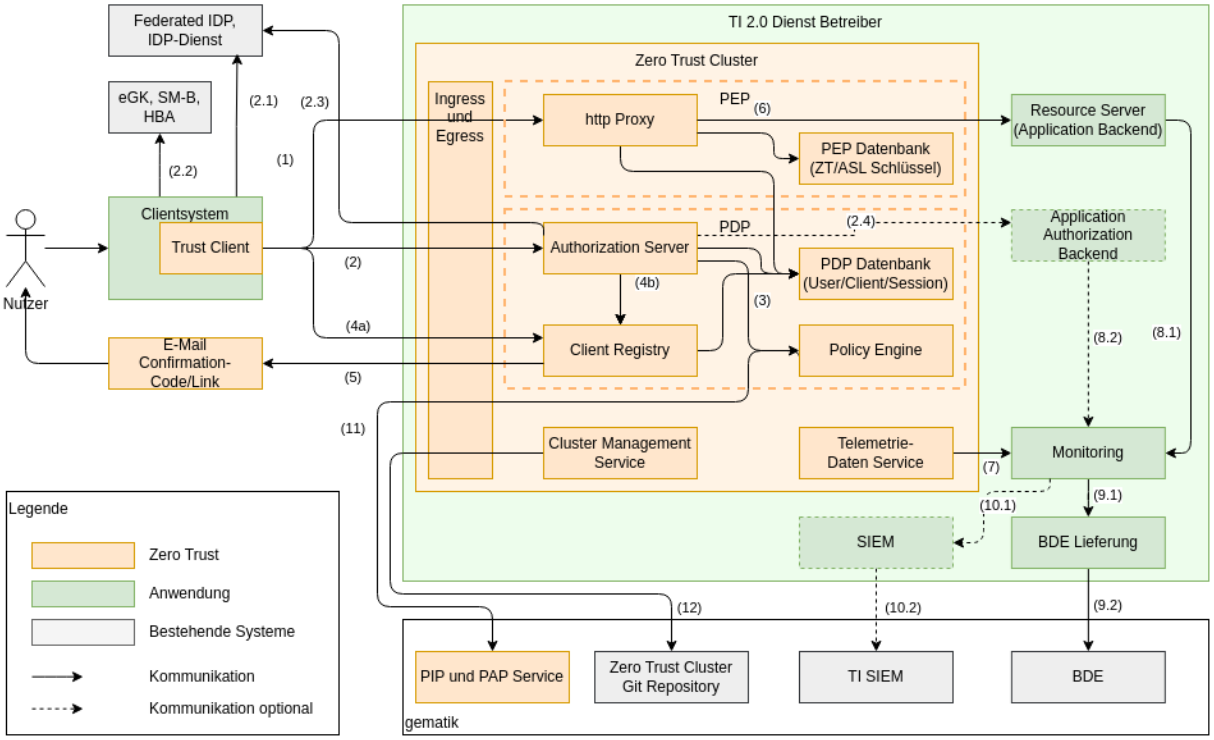

Abbildung 2: Zero Trust-Architektur der TI 2.0

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) bis (2.3) sowie optional (2.4): 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 die Policy Engine durch Anwendung der Policies die 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(C)-B Authentisierung. Der Authorization Server erkennt anhand des verwendeten OAuth Flows, ob die Geräteregistrierung implizit oder mit Geräte/App-Attestierung erfolgt.

(5): Die Client Registrierung erfordert eine Bestätigung des Nutzers durch einen zweiten Faktor.

(7): Die Telemetrie-Daten der Komponenten des ZT Clusters werden an das Monitoring System des Anbieters übergeben.

(8.1) und (8.2): Der Resource Server und der optionale Application Authorization Server werden ebenfalls vom Monitoring überwacht.

(j9.1) und (9.2): Aus den Monitoring Daten werden die Daten der Betriebsdatenerfassung gebildet und versendet.

(10.1) und (10.2): Wenn der Anbieter ein SIEM einsetzt, werden aus dem Monitoring SIEM-Daten ermittelt und optional an das zentrale SIEM der gematik gesendet.

(11): Der PDP fragt regelmäßig den PIP und PAP Service nach neuen Policies und Daten ab.

(12): Der Cluster Management Service überwacht die Clusterkonfiguration und setzt durch, dass die im git 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 besteht aus einem http Proxy und einer Datenbank. Der PDP enthält die Policy Engine, den Authorization Server und die Client Registry sowie eine Datenbank. 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-Anbieters 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 stellt die Außenschnittstelle des Dienstes dar, in den er integriert ist.

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. Am PEP werden

Betriebsdaten erhoben, verarbeitet und dem Telemetrie Daten Service im Zero Trust-Cluster zur Verfügung gestellt.

4.3 Policy Decision Point (PDP)

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

- Der PDP fungiert als OAuth2 Authorization Server und verwaltet die Autorisierung von Benutzeranfragen auf geschützte Ressourcen. Zudem überwacht der Authorization Server die Benutzersessions, um sicherzustellen, dass sie gültig sind und den Sicherheitsrichtlinien entsprechen. Der Authorization Server ist als vertrauenswürdige Relying Party im föderierten Identitätsmanagement registriert. Dadurch kann der Authorization Server Identitätsinformationen von Benutzern sicher und vertrauenswürdig beziehen und bei Bedarf eine (erneute) Nutzer-Authentifizierung an die IDPs delegieren. Dadurch stellt der Authorization Server sicher, dass nur authentifizierte Benutzer Zugriff auf die geschützten Ressourcen erhalten.
- Der PDP ermöglicht mit der Client Registry 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.
- Der PDP analysiert und interpretiert in der Policy Engine die Sicherheitsrichtlinien, die im Rahmen des Zero Trust-Modells definiert sind. Diese Policies können Kriterien wie Benutzeridentität, Gerätetyp, Standort, Zeitpunkt der Anfrage und andere Kontextinformationen ("Signale") enthalten, die relevant für die Zugriffsentscheidung sind. Basierend auf der Interpretation derPolicies trifft die Policy Engine Entscheidungen darüber, ob eine Zugriffsanfrage auf eine bestimmte Ressource genehmigt oder abgelehnt wird. Diese Entscheidungen erfolgen auf Plattformebene, was bedeutet, dass die Policy Engine 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).
- Die Policy Engine verwendet dabei die Informationen, die ihr 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.

Am PDP werden Betriebsdaten erhoben, verarbeitet und dem Telemetrie Daten Service im Zero Trust-Cluster zur Verfügung gestellt.

4.4 Trust Client

Im Kontext von Zero Trust der TI stellt der "Trust Client" eine logische Komponente innerhalb einer Clientanwendung (Primärsystem (PS), Frontend des Versicherten (FdV) etc.) dar.

Ein Trust Client im Zero Trust-Modell wird nicht als vertrauenswürdig angesehen, sondern muss - genauso wie alle Komponenten im Netzwerk - kontinuierlich authentifiziert und autorisiert werden. Die Zugriffsentscheidungen werden basierend auf aktuellen Richtlinien, Kontextinformationen, Bedrohungsinformationen und insbesondere in Kenntnis des diesen Client benutzenden Benutzers getroffen.

Die Aufgaben des Trust Clients sind:

- Erzeugung, sichere Speicherung und Prüfung der kryptographischen App/Geräte Identität
- Erzeugung der App/Gerät-Attestierung und Ermittlung und Übertragung der Eigenschaften der Laufzeitumgebung (Betriebssystem, Betriebssystem Version, etc.)
- Implementierung des OAuth Flows
- Management der Sessions inkl. Verwaltung der Access und Refresh Token.
- Management der Clientregistrierungen

4.5 Policy-Information und -Administration

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

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

4.5.1 Policy Information Point (PIP)

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

4.5.2 Policy Administration Point (PAP)

Der PAP ist für die Verwaltung und Konfiguration von Sicherheitsrichtlinien verantwortlich. Er bietet eine Schnittstelle oder eine Konsole, über die Richtlinien in hoheitlicher Verantwortung definiert, geändert und gelöscht werden können. Policy-Administratoren können im PAP Zugriffsregeln, Autorisierungsniveaus, Bedrohungsabwehrmaßnahmen und andere Sicherheitsrichtlinien festlegen. Der PAP ermöglicht es Policy-Administratoren, Richtlinien - basierend auf verschiedenen Kriterien wie Benutzerrollen, Gruppenzugehörigkeit, Standorten und Geräteattributen - zu differenzieren. Änderungen an den Sicherheitsrichtlinien, die im PAP vorgenommen werden, werden an den PIP weitergegeben, damit andere Komponenten im Zero Trust-Netzwerk auf die aktualisierten Richtlinien zugreifen können. Das Vertrauen in

bereitgestellte und angepasste Policies wird über Signaturen für die Sicherstellung von 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
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	Zunächst werden zwei Plattformen gesondert behandelt: Android und Apple (iOS, macOS etc.). Diese beiden Plattformen bieten Mechanismen für die Attestation der Client-Instanzen und der Umgebung, in welcher diese Clients ausgeführt werden. Alle andere Clients werden zunächst als generische Software-Produkte eingestuft.
Produkt-Plattform-Id	Plattformspezifisch eindeutige Kennung der Client-Software Android: Package-Name und Signer-Zertifikat Fingerprint Apple: Bundle-ID und Apple-ID Software: Registriert durch gematik , analog zu ClientIds in E-Rezept
Attestation-Methode	Die Methode, nach welcher die Client-Software und die Ablaufumgebung attestiert werden kann. Zunächst werden Android und Apple (iOS) unterstützt, weil diese Plattformen entsprechende Mechanismen zur Remote-Attestation anbieten. Windows und Linux verwenden zur Attestation der Clients TPM 2.0. Apple (MacOS) verwendet die Secure Enclave.

Der Nutzer muss sich vor der Clientregistrierung mit einer TI-Identität authentifizieren, z. B. GesundheitsID oder SM(C)-B. Bei der Registrierung werden die statischen Eigenschaften eines Client-Systems für jede Client-Instanz mit einem vom Client-Nutzer signierten Softwarestatement bekannt gemacht. Der Client erzeugt dabei eine kryptographische Identität (DPoP Key [RFC9449]), die Server-seitig an den Nutzer und Client gebunden wird.

Bei der SM(C)-B Authentisierung mit DPoP erfolgt die Clientregistrierung implizit (wenn die SM(C)-B freigeschaltet ist). Das vom Trust Client automatisch erstellte Client Assertion JWT enthält die Clientregistrierungsdaten und wird per SM(C)-B signiert.

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 (aus dem gID id_token oder aus dem SM(C)-B Zertifikat)
Client-Eigenschaften (Posture)	Aktuelle Eigenschaften der Ablaufumgebung des Clients, insbesondere: • Betriebssystem • Betriebssystem-Version
Client-Attestation	Falls die Plattform die Attestation des Clients ermöglicht, wird hier die plattformspezifische Attestation angegeben. • Google-Android: Key and ID Attestation • Apple: DCAppAttest • Software: keine Attestation, nur Nutzer-Bindung

Hinweis: Für andere mobile Betriebssysteme wird die Attestation unterstützt, wenn sie verfügbar ist.

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
utsname.machine	Art und Version des Geräts, z. B. ”iPhone 15”

identifierForVendor	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 Policy-Entscheidungen 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 TI1.0 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 Monitoring-Komponente erfasst. Bei der Nachbereitung der Telemetrie-Daten werden personenbezogene oder -beziehbare Daten anonymisiert, um diese bereinigten Daten dem Anbieter regelhaft zugänglich zu machen. Das bereinigte Monitoring-Log kann von dem Anbieter 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 Identity Provider

Im Zero Trust-Paradigma arbeiten der PDP Authorization Server 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 Authorization Server 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.

4.9 Fachdienst-Backend

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

5 Spezifikation

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

Hinweis: Details in diesem Kapitel werden im Rahmen der Implementierung zwischen gematik und dem Zero Trust-Hersteller festgelegt und in einer Folgeversion veröffentlicht.

Offener Punkt: Details in diesem Kapitel werden im Rahmen der Implementierung zwischen gematik und dem Zero Trust-Hersteller festgelegt und in einer Folgeversion veröffentlicht.

5.1 Übergreifende Anforderungen für Datenschutz und Sicherheit

A_25400 - Zero Trust-Komponenten - Umsetzung Sicherer Softwareentwicklungsprozess
Der Hersteller einer Zero Trust-Komponente MUSS einen sicheren Softwareentwicklungsprozess umsetzen (siehe [gemSpec_DS_Hersteller#Kapitel 2.2 Sicherer Softwareentwicklungsprozess]).**[<=]**

A_25401 - Zero Trust-Komponenten - Darstellung der Voraussetzungen für sicheren Betrieb des Produkts im Betriebshandbuch
Der Hersteller einer Zero Trust-Komponente MUSS für sein Produkt im dazugehörigen Betriebshandbuch leicht ersichtlich darstellen, welche Voraussetzungen vom Anbieter und der Betriebsumgebung erfüllt werden müssen, damit ein sicherer Betrieb des Produktes gewährleistet werden kann. **[<=]**

A_25402 - Zero Trust-Komponenten - Schutz der transportierten Daten
Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass die Vertraulichkeit und Integrität der transportierten Daten gewährleistet ist.
Alle Endpunkte der Zero Trust-Komponenten MÜSSEN TLS gesichert sein.**[<=]**

Hinweis: Es wird empfohlen ein Service Mesh (z. B. Istio oder linkerd) einzusetzen.

A_26517 - Zero Trust-Komponenten - Unterstützung von mTLS
Alle Zero Trust-Komponenten MÜSSEN die Konfiguration und Nutzung von mTLS unterstützen.**[<=]**

A_25403 - Zero Trust-Komponenten - Schutzmaßnahmen gegen die OWASP Top 10 Risiken
Alle Zero Trust-Komponenten MÜSSEN technische Maßnahmen zum Schutz vor den Risiken in der aktuellen Version der [OWASP-Top-10-Risiken] umsetzen.**[<=]**

A_25404 - Zero Trust-Komponenten - Angriffe erkennen
Alle Zero Trust-Komponenten MÜSSEN Maßnahmen zur Erkennung, Kategorisierung und Protokollierung bzw. Meldung von Angriffen umsetzen.**[<=]**

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

A_25405 - Zero Trust-Komponenten - Angriffen entgegenwirken
Alle Zero Trust-Komponenten MÜSSEN Maßnahmen zur Schadensreduzierung und -verhinderung von Angriffen umsetzen.**[<=]**

A_25406 - Zero Trust-Komponenten - Eingabe Validierung von Operationen
Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass alle Daten und Parameter, die über eine API kommuniziert werden, sicherheitstechnisch validiert werden.**[<=]**

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

A_25407 - Zero Trust-Komponenten - Sicherheitstechnische Validierung von Policy und Konfigurationen
Alle Zero Trust-Komponenten MÜSSEN sicherstellen, dass alle Daten und Parameter, die von einer Konfigurationsdatei oder Policy gelesen werden, sicherheitstechnisch validiert werden.**[<=]**

A_25408 - Zero Trust-Komponenten - Verbot Profilbildung
Der Anbieter einer Zero Trust-Komponente DARF Profile - außer zum Zweck des Security Monitorings - NICHT bilden.
[<=]

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 Anbieter einer Zero Trust-Komponente MUSS die Maßnahmen für erhöhten Schutzbedarf aus dem BSI-Bausteins „OPS.1.1.2 Ordnungsgemäße IT-Administration“ [BSI-Grundschutz] während des gesamten Betriebs der Komponente umsetzen. **[<=]**

A_26479 - Zero Trust-Komponenten - Ordnungsgemäße Änderung von Konfigurationen
Alle Zero Trust-Komponenten MÜSSEN durch technische und organisatorische Mittel sicherstellen, dass eine Änderung der Konfiguration der Komponente und die Änderungen von Feature Flags nur unter 4-Augen erfolgen kann. **[<=]**

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 Anbieter 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 Anbieters 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 Anbieter der Zero Trust-Komponenten vertrauliche oder zur Profilbildung geeignete Daten zur Kenntnis gelangen. **[<=]**

Hinweis: Der Telemetrie-Daten Service im ZT Cluster muss die vom PEP und PDP gesammelten Telemetrie-Daten so verändert an das Monitoring System des Anbieters weitergeben, dass eine Profilbildung nicht mehr möglich ist.

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

A_25746 - Zero Trust-Komponenten - Keine sicherheitsrelevanten Daten in Logging und Monitoring
Die Zero Trust-Komponenten MÜSSEN sicherstellen, dass in für den Betrieb erstellten Protokollen des Anbieters keine sicherheitsrelevanten Daten enthalten sind.**[<=]**

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

A_25747 - Zero Trust-Komponenten - Löschfristen Protokolle
Der Anbieter einer Zero Trust-Komponente MUSS sicherstellen, dass die

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

[<=]

5.1.2 Sicherheits- und Datenschutz-Anforderungen an das Security Monitoring

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

A_25419 - Security Monitoring - Erkennungsfähigkeit
Der Anbieter einer Zero Trust-Komponente MUSS sicherstellen, dass das Monitoring System die folgenden Merkmale der Kommunikation erkennen kann:

- Geolokation - Land und Ort
- Impossible Travel

- Zugriffe über TOR Netzwerke
- Zugriff von VPN-Provider
- Zugriffe über Proxies
- Zugriffe über Botnetze
- Traffic-Volumen-Anomalien
- Network-Protokoll Anomalien

[<=]

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

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

A_25421 - Security Monitoring - Empfang von Missbrauchserkennung auf Fachdienstebene
Falls der Fachdienst eine Missbrauchserkennung durchführt, MUSS der Anbieter 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 Anbieter des Monitoring Systems MUSS sicherstellen, dass bei Erkennung folgender Kommunikationsmerkmale die erforderlichen Informationen automatisiert an den PDP 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 Authorization Server die Information, dass sich eine Eigenschaft der aktuellen Session geändert hat. Der Authorization Server 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 die Policy Engine.

A_25484 - Security Monitoring - Security KPIs
Der Anbieter 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 Gerätfreischaltungen 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 Anbieter des Security Monitoring MUSS sicherstellen, dass bei der Aktualisierung der vom PIP und PAP Service heruntergeladenen Daten 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 Anbieter des Security Monitoring MUSS sicherstellen, dass beim folgenden Fehler während der Aktualisierung der vom PIP und PAP Service heruntergeladenen Daten 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 Policy-Signatur

[<=]

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 Anbieter keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, gibt es Sonderanforderungen, um die Daten während der Verarbeitung zu schützen.

Offener Punkt: Der ZT Cluster muss in einer VAU laufen können. Deswegen muss der ZT Cluster die Speicherung und Verwendung der Privatschlüssel von Komponente-

Identitäten in einem HSM unterstützen. Details in diesem Kapitel werden im Rahmen der Implementierung zwischen gematik und dem Zero Trust-Hersteller festgelegt und in einer Folgeversion veröffentlicht.

A_25608 - ZT Cluster - Verarbeitung von Daten mit Schutzbedarf "sehr hoch"
Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Anbieter keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Anbieter 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 Anbieter keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Anbieter 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 Anbieter keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Anbieter 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 Anbieter keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Anbieter beim Einsatz eines HSMs sicherstellen, dass dessen Eignung durch eine erfolgreiche Evaluierung nachgewiesen wurde. Als Evaluierungsschemata kommen dabei Common Criteria oder Federal Information Processing Standard (FIPS) in Frage. Die Prüftiefe MUSS mindestens:

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

entsprechen.**[<=]**

A_26065 - Nur zugelassene Images in Produktion
Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Anbieter keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Anbieter mit technischem Mittel sicherstellen, dass nur von gematik signierte und für den Einsatz in der PU vorgesehene Produktimages in der PU laufen können. **[<=]**

Offener Punkt: Für den Betrieb des ZT Clusters in einer VAU werden von der gematik signierte und für den Einsatz in der PU vorgesehene Produktimages verwendet, aber für die Ausführung in der VAU angepasst. Über die Attestation des VAU-Produktimages wird der Nachweis erbracht, dass das Produktimage der gematik verwendet wurde. Wie der Nachweis genau erfolgt, dass bei Ausführung von ZT Cluster-Komponenten in einer VAU die für den Einsatz in der PU vorgesehene Produktimages verwendet werden, wird in einer Folgeversion des Dokuments festgelegt.

5.1.4 Sicherheits- und Datenschutz Anforderungen an dem Trust Client in FdVs

A_25802 - Trust Client - Einhaltung der BSI-Prüfvorschrift
Der Trust Client eines FdVs MUSS die Prüfaspekte aus BSI [BSI-Prüfvorschrift] erfüllen, sofern sie für Der Trust Client anwendbar sind.**[<=]**

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

5.2 Anforderungen an Clientsysteme und Trust Clients

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

Ein Trust Client ist ein Teil des Clientsystems, der die Kommunikation mit dem PDP und dem PEP eines Dienstes übernimmt.

Mobile iOS und Android Apps werden unterstützt und setzen ebenfalls einen Trust Client um. Anforderungen, die nur für diese Clientsysteme und Trust Clients gelten werden, verwenden die Bezeichnung "mobiler Client" oder "mobiles Clientsystem" und "mobiler Trust Client".

5.2.1 Hersteller und Herausgeber

A_25335 - Hersteller Clientsystem - Hinweise und Maßnahmen sicherer Betrieb
Der Hersteller bzw. Herausgeber eines Clientsystems MUSS den Benutzer über Maßnahmen zum sicheren Betrieb seines Clientsystems vor der Inbetriebnahme informieren und während des Betriebs stets zum Abruf durch den Benutzer bereithalten. **[<=]**

A_25336 - Hersteller Clientsystem - Regelmäßige Updates
Der Hersteller bzw. Herausgeber eines Clientsystems MUSS, solange das Produkt nicht abgekündigt ist, dem Benutzer regelmäßig (z. B. quartalsweise) Updates für das Clientsystem bereitstellen, um das Clientsystem dauerhaft auf dem Stand der Technik zu halten und Sicherheitslücken zu schließen.**[<=]**

A_25337 - Hersteller Clientsystem - Registrierung für product_id
Der Hersteller bzw. Herausgeber eines Clientsystems MUSS sich über einen organisatorischen Prozess bei der gematik für die Nutzung von Diensten, für welche Token abgerufen werden sollen, registrieren. Der Hersteller bzw. Herausgeber eines Clientsystems bekommt dabei eine "product_id" zugewiesen, die in jeder Instanz des Clientsystems verwendet werden MUSS.**[<=]**

A_25427 - Hersteller Clientsystem Android - Google Cloud Projekt
Der Hersteller bzw. Herausgeber eines Clientsystems für eine Android-basierte Betriebsumgebung MUSS ein Google Cloud Projekt führen oder eine gleichwertige alternative Plattformattestierung verwenden, um Nachweise über die Geräteintegrität einer jeden Clientsysteminstallation beziehen zu können.**[<=]**

5.2.2 Verbindungsaufbau

A_25338 - Trust Client - Identifikation mittels product_id
Der Trust Client MUSS bei der SM(C)-B Authentifizierung mit DPoP in der JWT-Assertion unter `"urn:telematik:client-self-assessment"` die Client-Daten gemäß `[client-instance.yaml]` eintragen. Dabei MUSS die von der gematik für das Clientsystem ausgestellte `product_id` nach [A_25337] und eine selbst festgelegte Versionskennzeichnung nach folgendem Schema verwendet werden:

- <product_id> gemäß Registrierung bei der gematik mit Länge maximal 20 Zeichen, Zeichenvorrat [0-9a-zA-Z\-],
- <product_version> gemäß Produktidentifikation mit Länge 1-20 Zeichen, Zeichenvorrat [0-9a-zA-Z\-\.].

[<=]

A_25339 - Trust Client - Exponential Backoff
Der Trust Client SOLL bei Server-seitigen Fehlermeldungen, die auf eine Überlastung des Zielsystems schließen lassen (z. B. http-status 5xx, 429 - too many requests etc.), erneute Verbindungsversuche nach dem Prinzip des Exponential Backoffs [ExpBack] durchführen.**[<=]**

Hinweis: Die Parameter für das Verfahren des Exponential Backoffs werden vom Hersteller des Trust Clients festgelegt.

A_25340 - Trust Client- Zertifikatsprüfung
Der Trust Client MUSS alle Zertifikate, die es aktiv verwendet (bspw. TLS-Verbindungsaufbau), auf Integrität und Authentizität prüfen. Falls ein Zertifikat ungültig ist, so MUSS der Trust Client die von dem Zertifikat und den darin enthaltenen Attributen (bspw. öffentliche Schlüssel) abhängenden Arbeitsabläufe ablehnen.
Der Trust Client MUSS alle öffentlichen Schlüssel, die es verwenden will, auf eine positiv verlaufene Zertifikatsprüfung zurückführen können.**[<=]**

A_26681 - Trust Client - Umsetzen eines ZT/ASL-Kanals
Der Trust Client MUSS einen ZT/ASL-Kanal (Client-Seite) umsetzen können. Die Verwendung des ZT/ASL-Kanals MUSS durch Konfiguration pro TI 2.0 Dienst ein- und ausschaltbar sein. In der Default-Einstellung ist der ZT/ASL-Kanal ausgeschaltet.
Das Access Token und das DPoP Token müssen außerhalb der Verschlüsselung des ZT/ASL-Kanals transportiert werden.
Wenn ein ZT/ASL-Kanal verwendet und ein PoPP-Token im Request transportiert wird, dann MUSS das PoPP-Token durch den ZT/ASL-Kanal geschützt transportiert werden.**[<=]**

Hinweis: Die Anforderungen für den ZT/ASL-Kanal sind in [C_12090_Anlage] zu finden.

5.2.3 Clientregistrierung

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

A_25432 - Trust Client - Ablauf Clientregistrierung

Der mobile Trust Client MUSS, sofern es an Schnittstellen der Telematikinfrastruktur wegen einer ungültigen/fehlenden Geräte/App-Registrierung abgewiesen wird, eine Registrierung am Service der Client Registry durchführen, in dem es

- den/die Benutzer:in mittels OpenID Connect authentifiziert,
- kryptografische Client-Credentials lokal generiert,
- die generierten Credentials sowie die Clientintegrität attestiert und
- eine zusätzliche Benutzerbestätigung mittels One-Time-Passwort (gemäß [TR-03107-1]) über einen zweiten Kommunikationsweg (z. B. E-Mailbestätigung) startet.

[<=]

Hinweis: Für die Clientregistrierung muss das Vertrauensniveau hoch, nicht erreicht werden.

A_25766 - Trust Client - Client Credentials in TI Qualität
Der Trust Client MUSS die Client-Credentials im Form von kryptografischen Schlüsseln gemäß der Festlegungen in [gemSpec_Krypt] (Verfahren, Algorithmen, Schlüssellängen etc.) unterstützen.**[<=]**

A_25769 - Trust Client - Client Credentials sicher generieren und schützen
Das Trust Client auf mobilem Gerät mit Apple- oder Android-basierter Betriebsumgebung MUSS die Generierung der Client-Credentials derart generieren und speichern, dass ein Kopieren und Exportieren der Schlüssel verhindert wird.**[<=]**

Hinweis: Eine Speicherung der Schlüssel in einem Hardware-Modul ist gegenüber einer Software-Lösung (z. B. Android TEE) zu bevorzugen.

A_25770 - Trust Client - Client Credentials Rotation
Der Trust Client MUSS seine Client-Credentials regelmäßig rotieren (erneuern und neu registrieren), wobei die Häufigkeit der Rotation durch die gematik nach einer Auswertung der initialen Benutzererfahrung festgelegt wird.**[<=]**

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

A_25767 - Trust Client - Clientkey in JWT
Das Trust Client MUSS Private Key JWT [RFC7521] und [RFC7523] sowie DPoP [RFC9449] zur Authentifizierung unterstützen.**[<=]**

A_25434 - Trust Client - Clientregistrierung mit bestätigten Umgebungseigenschaften Android
Der Trust Client für eine Google-Android basierte Betriebsumgebung MUSS seine Client-Credentials, App-Integrität und -Authentizität sowie OS-/FW und HW-Eigenschaften über Key and ID Attestation gegenüber PDP Client Registrierung bestätigen..**[<=]**

A_25768 - Trust Client - Clientregistrierung mit bestätigten Umgebungseigenschaften Apple
Der Trust Client für eine Apple-basierte Betriebsumgebung (iOS, macOS) MUSS die Client-Credentials, App Integrität und Authentizität über DCAppAttest gegenüber PDP Client Registry bestätigen. Eigenschaften der Laufzeitumgebung MÜSSEN durch das Clientsystem über einen geprüften Prozess bestätigt werden.**[<=]**

A_25758 - Trust Client - Erfassung Kontaktinformation für Offband-Verifikation
Der mobile Trust Client 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 - Trust Client - Unterstützung des Nutzers bei der Registrierung
Der mobile Trust Client MUSS den Nutzer bei der Clientregistrierung und -Verwaltung geeignet unterstützen (z. B. mittels Guide, Tutorial o. ä.).**[<=]**

A_25733 - Trust Client - Clientverwaltung und manuelle De-Registrierung
Der Trust Client MUSS dem Nutzer eine Übersicht aller beim Clientregistrierungsdienst registrierten Clients darstellen und die Möglichkeit zur De-Registrierung einzelner Clients anbieten.**[<=]**

A_25734 - Trust Client - Zugriffsprotokoll Clientregistrierung
Der Trust Client 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 - Trust Client - Nutzerauthentifizierung mittels etablierter Standards
Der Trust Client MUSS die Mechanismen OAuth2 Client Assertion JWT oder OAuth Authorization Code Flow, OpenID Connect und OpenID Federation (Auswahl des zuständigen sektoralen IDP) unterstützen. **[<=]**

Hinweis: Perspektivisch sollen Trust Clients auch OpenID for Verifiable Credentials (OIDC4VC) unterstützen. OAuth2 Client Assertion JWT wird vor allem für stationäre Clients mit SM(C)-B Authentisierung verwendet. OAuth Authorization Code Flow, OpenID Connect und OpenID Federation wird grundsätzlich von mobilen Clients verwendet, kann aber auch von stationären Clients verwendet werden.

A_25762 - Trust Client - Nutzerauthentifizierung - Unterstützung etablierter Identitäten und Dienste
Der Trust Client 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(C)-B signiertem Client Assertion JWT und DPoP gemäß [RFC7523] und [RFC9449].

[<=]

5.2.5 Session Management

A_25781 - Trust Clients - OAuth2 Autorisierung
Der Trust Client MUSS bei die Rolle eines OAuth2 Clients [RFC6749] übernehmen und eine Autorisierung vom Authorization Server einholen. Dabei MUSS PKCE Flow [RFC7636] verwendet werden.
[<=]

A_25782 - Trust Client - OAuth2 Session Management
Der Trust Client MUSS

- die vom Autorisation 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,

- nach Bedarf abgelaufene Access Token über Refresh Token erneuern und
- eine Refresh Token-Rotation gemäß [RFC6749#10.4] unterstützen.

[<=]

A_25783 - Trust Client - Anweisungen aus http Response Status Codes und Header folgen
Der Trust Client 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 Replay-Nonce gemäß [RFC8555#6.5.1]

umsetzen.**[<=]**

5.2.6 Liste der HTTP-Statuscodes

Der folgende Abschnitt enthält die HTTP-Statuscodes, die Trust Clients von Zero Trust-Komponenten erhalten können, basierend auf den spezifischen Schritten wie Authentifizierung, Clientregistrierung und HTTP-Proxy.

A_27007 - Trust Client - http Statuscodes
Der Trust Client MUSS die http Statuscodes gemäß Tabelle ZT_http_Statuscodes unterstützen. **[<=]**

Tabelle 5 ZT_http_Statuscodes

Endpunkte	http Statuscodes
Authentifizierung mit Client Assertion JWT	200 OK: Authentifizierung erfolgreich. Der Client kann mit der gewünschten Operation fortfahren, z.B. Zugriff auf geschützte Ressourcen. 400 Bad Request: Fehlerhafte oder ungültige JWT-Assertion (z.B. falsches Format oder fehlende Claims). Der Client sollte die JWT-Assertion auf Fehler prüfen (z.B. Format, fehlende Claims) und die Anfrage entsprechend korrigieren oder erneut senden. 401 Unauthorized: Authentifizierung fehlgeschlagen, z.B. aufgrund einer ungültigen Signatur oder eines abgelaufenen JWT. Der Client sollte überprüfen, ob das JWT korrekt signiert und nicht abgelaufen ist. Er kann ein neues JWT generieren und die Anfrage erneut senden. 403 Forbidden: Der Client hat keine Berechtigung, auf die angeforderte Ressource zuzugreifen, obwohl er authentifiziert ist. Der Client sollte keine weiteren Versuche unternehmen und den Nutzer entsprechend informieren.
Authentifizierung mit OIDC und Authorization Code Flow	200 OK: Erfolgreiche Authentifizierung und Autorisierung. Der Client kann mit der gewünschten Operation fortfahren, z.B. Zugriff auf geschützte Ressourcen. 302 Found: Der Client wird zum Autorisierungs-Endpunkt des Identitätsproviders umgeleitet. Der Client sollte der

	Weiterleitungs-URL folgen, um den nächsten Schritt im Autorisierungscode-Austausch abzuschließen. 400 Bad Request: Fehlerhafte Anfrage, z.B. fehlende oder ungültige Parameter (z.B. falscher `redirect_uri`, `client_id`). Der Client sollte die Anfrageparameter überprüfen und sicherstellen, dass alle erforderlichen Parameter korrekt sind. Eine erneute Anfrage kann nötig sein. 401 Unauthorized: Authentifizierung fehlgeschlagen, z.B. ungültiger Code oder Token. Der Client sollte den Autorisierungscode überprüfen und gegebenenfalls einen neuen Code anfordern oder den Flow neu starten. 403 Forbidden: Autorisierung fehlgeschlagen, z.B. fehlende Zugriffsrechte. Der Client hat möglicherweise keine Berechtigung, die angeforderte Ressource zu nutzen. Der Client sollte den Nutzer darüber informieren und keine weiteren Anfragen stellen.
Clientregistrierung	201 Created: Client erfolgreich registriert. Der Client sollte die Registrierungsdaten sicher speichern und mit der weiteren Interaktion fortfahren. 400 Bad Request: Fehlerhafte Anfrage, z.B. ungültige oder unvollständige Clientdaten. Der Client sollte die übermittelten Registrierungsdaten überprüfen und mögliche Fehler beheben, bevor er erneut versucht, sich zu registrieren. 401 Unauthorized: Fehlende oder ungültige Authentifizierung bei der Registrierung. Der Client muss sicherstellen, dass er korrekt authentifiziert ist. Er sollte die Authentifizierungsdaten überprüfen und erneut versuchen, sich zu registrieren. 403 Forbidden: Zugriff verweigert, z.B. wenn der Client nicht berechtigt ist, sich zu registrieren. Der Client sollte prüfen, ob er die Berechtigung zur Registrierung hat. Wenn nicht, sollte er keine weiteren Versuche unternehmen und den Nutzer informieren. 409 Conflict: Konflikt bei der Registrierung, z.B. ein Client mit der gleichen ID existiert bereits. Der Client sollte keine weiteren Registrierungsversuche senden und den Nutzer über die das Serverproblem informieren.
HTTP Proxy	200 OK: Anfrage erfolgreich durch den Proxy weitergeleitet. Der Client kann die angeforderte Ressource wie gewohnt verwenden. 301 Moved Permanently: Permanente Weiterleitung der Anfrage durch den Proxy. Der Client sollte die neue URL speichern und zukünftige Anfragen an diese Adresse senden. 302 Found: Temporäre Weiterleitung durch den Proxy. Der Client sollte der Weiterleitung folgen, um die angeforderte Ressource zu erhalten, aber die ursprüngliche URL für zukünftige Anfragen beibehalten. 400 Bad Request: Ungültige Anfrage an den Proxy. Der Client sollte die Anfrage überprüfen und sicherstellen, dass sie korrekt formatiert und vollständig ist, bevor er sie erneut sendet. 403 Forbidden: Zugriff auf die angeforderte Ressource durch den Proxy verweigert. Der Client sollte keine weiteren Anfragen an diese Ressource senden und den Nutzer über die fehlende Berechtigung informieren. 404 Not Found: Die angeforderte Ressource wurde nicht gefunden.

	502 Bad Gateway: Der Proxy hat eine ungültige Antwort vom Upstream-Server erhalten. Der Client sollte die Anfrage später erneut senden, da der Fehler auf einem Problem des Upstream-Servers beruhen könnte. Gegebenenfalls kann der Nutzer informiert werden. 504 Gateway Timeout: Der Proxy hat auf eine Antwort vom Upstream-Server gewartet, diese aber nicht innerhalb des Timeouts erhalten. Der Client sollte die Anfrage nach einer angemessenen Wartezeit erneut versuchen und den Nutzer darüber informieren, dass der Server nicht rechtzeitig geantwortet hat.
Allgemeine http Statuscodes	429 Too Many Requests: Zu viele Anfragen innerhalb eines bestimmten Zeitraums (Rate Limiting). Der Client sollte die Anzahl der Anfragen reduzieren und eine geeignete Wartezeit (Retry-After Header beachten) einhalten, bevor er die Anfrage erneut sendet. 500 Internal Server Error: Allgemeiner Serverfehler. Der Client sollte den Vorgang möglicherweise zu einem späteren Zeitpunkt wiederholen oder den Nutzer auf ein Problem auf dem Server hinweisen.

A_26662 - ZT Cluster, http Fehlerdetails
Der ZT Cluster SOLL zu den http Statuscodes Details ergänzen, wenn sie für den Trust Client oder das Clientsystem hilfreich sind.**[<=]**

Beispiel für eine sinnvolle Ergänzung der Fehlerdetails:

Bei der Authentifizierung mit Client Assertion JWT fehlt im JWT eine Angabe product_version oder die product_version ist nicht in der Policy Engine bekannt, dann antwortet der Authorization Server mit http Status 400 Bad Request sowie einer Beschreibung, dass die product_version fehlt oder nicht bekannt ist.

Offener Punkt: Das Format der Fehlerdetails und wie Fehlerdetails zu Zugriffsentscheidungen bereitgestellt werden, wird in einer Folgeversion des vorliegenden Dokuments festgelegt.

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 git Repository bereitgestellt, sodass die Anbieter von TI 2.0 Diensten ihren spezifischen ZT Cluster darauf aufbauend als Kubernetes Cluster konfigurieren können. Die ZT Cluster-Konfiguration muss in ein git Repository der gematik als git submodule verlinkt werden. Der Cluster Management Service des Zero Trust-Clusters setzt durch, dass nur die im gematik git Repository verlinkte ZT Cluster-Konfiguration ausgeführt werden kann. Dadurch ist es möglich, dass der Anbieter 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 Anbieter 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 Anbieter eines TI 2.0 Dienstes MUSS für seinen ZT Cluster die ihm zugewiesene Konfiguration aus dem git 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.

A_25840 - ZT Cluster - Sichere interne Kommunikation
Alle Komponenten des ZT Cluster MÜSSEN die ausgehende interne Kommunikation zu anbieterspezifischen Diensten und zu anwendungsspezifischen Diensten über die im Cluster vorhandenen Mechanismen absichern und deren Authentizität verifizieren können.**[<=]**

A_26519 - ZT Cluster, Unterstützung von Service-Mesh Lösungen
Die Komponenten des ZT Cluster MÜSSEN es ermöglichen, dass Service-Mesh Lösungen zur Verwaltung der Komponenten eingesetzt werden können.**[<=]**

A_26521 - ZT Cluster, Unterstützung von Canary Releases
Die Komponenten des ZT Cluster MÜSSEN Canary Releases unterstützen.**[<=]**

A_25666 - ZT Cluster - TLS Terminierung
Die Komponente Ingress MUSS TLS für von außen eingehende Verbindungen terminieren können. Alternativ wird die TLS-Verbindung am PEP http Proxy terminiert.
[<=]

A_26964 - ZT Cluster - OCSP Stapling
Komponenten innerhalb des ZT Clusters (inkl. Ingress), die von außen kommende TLS Verbindung terminieren, MÜSSEN TLS-OCSP-Stapling [RFC6066] verwenden. Die im TLS-Handshake mit gesendete OCSP-Response MUSS max. 50 Minuten alt sein. Sollte vom entsprechenden OCSP-Responder trotz regelmäßigen Versuchens keine OCSP-Response bezogen werden können, so MUSS die jüngste zur Verfügung stehende OCSP-Response verwendet werden und es MUSS regelmäßig weiter probiert werden (bspw. im 5'-Takt).
[<=]

A_26639 - ZT Cluster - Unterstützung Websocket
Die Komponente Ingress MUSS das WebSocket-Protokoll unterstützen.**[<=]**

A_26640 - ZT Cluster - http Protokoll-Versionen
Die Komponente Ingress MUSS das http Protokoll in den Versionen http/1.1, http/2 und SOLL http/3 unterstützen.**[<=]**

A_25652 - ZT Cluster - Push Gateway
Der ZT Cluster MUSS Push-Notifications über die von App-Anbietern bereitgestellten Push-Gateways unterstützen, um die Notifications an bestimmte oder alle registrierte Clients eines Anwenders verschicken zu können.**[<=]**

A_25737 - ZT Cluster - Push Notification
Der ZT Cluster 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.**[<=]**

Offener Punkt: Wie Clients ihre Push Konfiguration in den PDP eintragen können und wie Resource Server und ZT Cluster-Komponenten Push-Notification Events an eine neue ZT Cluster-Komponente übergeben können, damit diese Notification Events entsprechend der Push-Konfiguration versendet werden können, wird in einer Folgeversion des vorliegenden Dokuments festgelegt.

Offener Punkt: In welchem Format die Daten des Telemetrie Daten Service an das Monitoring gesendet werden und wie die Konfiguration für den Endpunkt des Monitoring Systems erfolgt, wird in einer Folgeversion des vorliegenden Dokuments festgelegt.

A_26988 - Telemetrie-Daten Service - Fehlermeldungen
Alle Komponenten des ZT Cluster MÜSSEN ihre Fehlermeldungen so bereitstellen, dass der Telemetrie-Daten Service die Fehlermeldungen sammeln und an einen Monitoring Service weiterleileiten kann.**[<=]**

A_26661 - ZT Cluster - http Statuscodes
Der ZT Cluster MUSS die http Statuscodes gemäß Tabelle ZT_http_Statuscodes unterstützen. **[<=]**

5.4 Anforderungen an Policy Enforcement Points

Der Policy Enforcement Point (PEP) stellt die zentrale Sicherheitskomponente einer Zero Trust-Architektur dar, da in dieser alle Zugriffsentscheidungen durchgesetzt (engl.: enforce) werden.

5.4.1 PEP http Proxy

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

A_26666 - PEP http Proxy - TLS Terminierung
Die Komponente PEP http Proxy MUSS TLS für eingehende Verbindungen von außerhalb Kubernetes terminieren können. Alternativ wird die TLS-Verbindung an der Komponente Ingress terminiert.**[<=]**

A_26195 - PEP http Proxy - Unterstützung Websocket
Die Komponente http Proxy MUSS das WebSocket-Protokoll unterstützen.**[<=]**

A_26641 - PEP http Proxy - http Protokoll-Versionen
Die Komponente http Proxy MUSS das http Protokoll in den Versionen http/1.1, http/2 und SOLL http/3 unterstützen.**[<=]**

A_25667 - PEP http Proxy - Verifikation Access Token Binding
Die Komponente http Proxy MUSS das Access Token Binding über den Mechanismus OAuth 2.0 Demonstrating Proof of Possession (DPoP) gemäß [RFC9449] verifizieren; d. h. der Claim "`jkt`" im Access Token MUSS eindeutig der Angabe im DPoP-Token entsprechen.**[<=]**

A_25668 - PEP http Proxy - Access Token Validierung
Die Komponente http Proxy MUSS das übergebene Access Token validieren. Insbesondere MÜSSEN

- die Signatur des Authorization Servers gültig,
- die Angaben zur zeitlichen Gültigkeit (Felder: `iat`, `exp`) valide,
- die Angabe `aud` für das Resource Backend korrekt eingetragen und
- die Angabe `scope` und `aud` passend zur Request url

sein.
Die Signatur des Access Tokens ist gültig, wenn sie mathematisch gültig ist und die Signatur vom Authorization Server im gleichen ZT Cluster erstellt wurde (default

Einstellung) oder von einem Authorization Server, der in der Konfiguration des http Proxy angegeben und im Entity Statement des Federation Master aufgeführt ist. Es MUSS möglich sein, mehrere Authorization Server in die Konfiguration des http Proxy einzutragen.
Wenn das Access Tokens ungültig ist, dann MUSS der Request mit http Code `401 Unauthorized` beantwortet werden.**[<=]**

Hinweis: Jeder Authorization Server, der zur Föderation des Federation Masters gehört, veröffentlicht ein eigenes Entity Statement, in dem die Schlüssel enthalten sind, mit denen die Signatur der Access Token geprüft werden kann.
Hinweis: Einige TI 2.0 Dienste benötigen ein PoPP-Token. Diese werden im Request Header PoPP übertragen und vom http Proxy geprüft.

A_26477 - PEP http Proxy - PoPP-Token Validierung
Die Komponente http-Proxy MUSS, wenn im Request der Header PoPP vorhanden ist (Request Header `PoPP`), das PoPP-Token validieren.
Insbesondere MUSS die Signatur des PoPP Servers gültig sein. Die Signatur des PoPP-Tokens ist gültig, wenn sie mathematisch gültig ist und die Signatur vom PoPP Server erstellt wurde. Der http Proxy MUSS ein vorhandenes und noch gültiges JWKS des PoPP Servers verwenden oder das JWKS des PoPP Servers herunterladen, um anhand der im JWKS enthaltenen Schlüssel die Signatur des PoPP-Tokens zu prüfen.
Der claim `actorId` des PoPP Token MUSS mit dem Attribut `identifier` aus den zum Access Token gehörenden Nutzer-Daten übereinstimmen.
Vor Ablauf der Gültigkeit MUSS der http Proxy ein neues JWKS des PoPP-Servers herunterladen,.
Wenn die Signatur des PoPP-Tokens ungültig ist oder eine der anderen Prüfungen nicht erfolgreich war, dann MUSS der Request mit http Code `403 Forbidden` beantwortet werden.
[<=]

Hinweis: Wenn ein ZT/ASL-Kanal am http Proxy terminiert wird, dann wird das PoPP Token durch den ZT/ASL-Kanal geschützt transportiert.

A_26493 - PEP http Proxy - Umgang mit JWKS des Popp Servers
Die Komponente http-Proxy MUSS, falls nach Ablauf der Gültigkeitsdauer für heruntergeladene JWKS des Popp Servers, kein neues JWKS heruntergeladen werden kann,

- das bestehende JWKS des Popp Servers für eine weitere Gültigkeitsdauer nutzen und
- einen Fehler für das Monitoring-System des Anbieters generieren.

[<=]

A_26480 - PEP http Proxy - Umsetzen eines ZT/ASL-Kanals
Die Komponente http-Proxy MUSS einen ZT/ASL-Kanal (Server-Seite) umsetzen können. Die Verwendung des ZT/ASL-Kanals MUSS durch Konfiguration ein- und ausschaltbar sein. In der Default-Einstellung ist der ZT/ASL-Kanal ausgeschaltet.**[<=]**

A_26946 - PEP http Proxy - Verwaltung der ZT/ASL-Kanal Schlüssel in der PEP Datenbank
Die Komponente http Proxy MUSS die ZT/ASL-Kanal Schlüssel in der PEP Datenbank verwalten, damit für die Terminierung des ZT/ASL-Kanals mehrere http Proxy Instanzen genutzt werden können.**[<=]**

A_26947 - PEP Datenbank - Zugriff nur für den PEP http Proxy
Die Komponente PEP Datenbank MUSS Zugriffe ausschließlich für den PEP http Proxy gewähren.**[<=]**

Hinweis: Die Anforderungen für den ZT/ASL-Kanal sind in [C_12090_Anlage] zu finden.

A_26492 - PEP http Proxy - Weiterleitung von Client-Daten
Die Komponente http Proxy MUSS so konfiguriert werden können, dass pro Endpunkt des Resource Servers die Weiterleitung der Client-Daten durch den http Proxy ein- und ausgeschaltet werden kann. Die default-Einstellung ist keine Weiterleitung der Client-Daten. Wenn die Weiterleitung der Client-Daten eingeschaltet ist, dann fragt der http Proxy die Client-Daten anhand der `cid` aus dem Access Token von der Client-Registry ab und fügt sie als zusätzlichen Header in den Request ein.
[<=]

A_25669 - PEP http Proxy - Zusätzliche http-Header
Die Komponente http Proxy MUSS die http Requests mit allen http-Headern an das Resource Backend weiterleiten und dabei die folgenden zusätzlichen http-Header einsetzen.

Tabelle 6: PEP http Proxy - Zusätzliche http-Header

http-Header	Format	Schema
ZTA-User-Info	Base64-URL kodierte JSON Struktur des User-Info Inhalts	[user-info.yaml]
ZTA-PoPP-Token-Content	Base64-URL kodierte JSON Struktur des Access Token Inhalts Optional: Wird nur gesetzt, wenn im Request ein PoPP Header enthalten ist.	PoPP Token Payload
ZTA-Client-Data	Base64-URL kodierte JSON Struktur der Client-Daten Optional: Wird nur gesetzt, wenn die Konfiguration des http Proxy für die URL des Requests die Weiterleitung der Client-Daten festlegt.	[client-instance.yaml]

Gleichnamige http-Header aus dem ursprünglichen http-Request MÜSSEN entfernt bzw. überschrieben werden.**[<=]**

Hinweis: Die Schema-Dateien sind in [GitHub ZT Schemas] festgelegt.

A_26560 - PEP http Proxy - Weiterleitungskonfiguration
Der PEP http-Proxy MUSS es ermöglichen, dass Regeln für die Weiterleitung von Request URLs an die URLs von Resource Servern konfiguriert werden können.
[<=]

A_26561 - PEP http Proxy - Caching
Der PEP http Proxy MUSS so konfiguriert werden können, dass Caching von Inhalten der Response möglich ist.**[<=]**

A_26589 - PEP http Proxy - Nutzer-Daten
Die Komponente http Proxy MUSS für jeden Request mit gültigem Access Token die Nutzer-Daten anhand des Parameters `jti` des Access Token aus der PDP Datenbank abfragen und als neuen http Header `ZTA-User-Info` in den Request eintragen, bevor der Request an den Resource Server weitergeleitet wird.**[<=]**

A_26590 - PEP http Proxy - Client-Daten
Wenn die Request-Weiterleitung mit Client-Daten konfiguriert wurde und der Request ein gültiges Access Token hat, dann MUSS die Komponente http Proxy die Client-Daten anhand des Parameters `sub` des Access Token aus der PDP Datenbank abfragen und als

neuen http Header `ZTA-Client-Data` in den Request eintragen, bevor der Request an den Resource Server weitergeleitet wird.**[<=]**

Hinweis: Die Abfrage der Client-Daten kann zusammen mit der Abfrage der Nutzer-Daten in einem Request an die PDP Datenbank erfolgen.

A_26974 - PEP http Proxy - Fehler vom Resource Server
Die Komponente http Proxy MUSS die Response vom Resource Server als Fehler des http Proxy werten, wenn der Resource Server den Response Header `ZTA-Cause: Proxy` gesetzt hat (der Resource Server hat einen Fehler im Request festgestellt und vermutet die Ursache beim http Proxy) und DARF diese Response nicht an den Client weiterleiten. Der entsprechende Request des Clients MUSS in diesem Fall mit http 500 beantwortet werden.**[<=]**

5.4.2 Sicherheits- und Datenschutz-Anforderungen an den PEP

A_25445 - PEP - Zugriffsentscheidung nur über PDP
Der PEP MUSS sicherstellen, dass Zugriffe auf den Fachdienst nur durch eine positive Zugriffsentscheidung vom PDP möglich sind. **[<=]**

Hinweis: Die positive Zugriffsentscheidung auf den Fachdienst ist gegeben, wenn der Client im Request Authorization Header ein gültiges Access Token mit passendem scope - ausgestellt von einem Authorization Server, zu dem eine Vertrauensbeziehung besteht - vorweisen kann. Durch das gültige Access Token ist sichergestellt, dass der Zugriff von dem PDP freigegeben wird.

Offener Punkt: Es muss noch geklärt werden, welche Validierungen notwendig sind, wenn DPoP plus ein TPM Schlüssel von einem stationären Gerät zum Einsatz kommt. Details werden im Rahmen der Implementierung zwischen gematik und dem Zero Trust-Hersteller festgelegt und in einer Folgeversion veröffentlicht.

5.5 Anforderungen an den Policy Decision Point

5.5.1 Policy Engine

Der PDP implementiert die Policy Engine als [Open Policy Agent] (OPA). Die Policies und die zugehörigen Daten erhält die Policy Engine per Download vom PIP und PAP Service. Aus den Input-Daten vom Authorization Server, den Daten vom PIP und den Policies vom PAP ermittelt die Policy Engine eine Entscheidung und gibt diese zurück an den Authorization Server.

Neben der OPA Instanz, die die Entscheidung für den Authorization Server trifft (aktive Instanz), ob eine Kommunikation zulässig ist, implementiert die Policy Engine noch eine zweite OPA Instanz, die mit einem zweiten OPA Bundle vom PIP und PAP Service arbeitet, aber die getroffenen Entscheidungen nicht an den Authorization Server zurückgibt. Diese Instanz wird Simulations-Instanz genannt.

A_25739 - PDP, Open Policy Agent Instanzen
Der PDP MUSS als Policy Engine zwei Open Policy Agent (OPA) Instanzen bereitstellen, wobei eine Instanz die Entscheidung für den Authorization Server trifft (aktive Instanz), und eine Instanz eine Entscheidung trifft, diese aber nicht an den Authorization Server sendet (Simulations-Instanz).
Die OPA Instanzen MÜSSEN gemäß Tabelle OPA_Konfiguration konfiguriert werden.

Tabelle 7: OPA - Konfiguration

Konfiguration	Aktive OPA Instanz	Simulations-OPA Instanz

`services:`
`  - name: <service name>`
`  url: <PIP und PAP service>`
	`<service name>`
	Innerhalb der OPA 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 OPA 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 OPA Instanz heruntergeladen.
	Default:
	`/policies/<TI service>/latest`
		`<path>`
		Durch das label `latest-sim` wird die neueste `bundle.tar.gz` Datei für die Simulations-Instanz der Policy Engine 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 OPAInstanz

`bundles:`
`  authz:`
`  signing:`
 	
`keyid: <PIP_and_PAP_key>`
`  scope: read`
	`<PIP_and_PAP_key>`
	Die `keyid` des Schlüssels, mit dem die OPA Bundles signiert sind.
		wie aktive OPA Instanz

`decision_logs:`
`  service: <service name>`
`  resource: $`
	`${DL_REMOTE_URL}`
	Die URL des Remote Servers, zu dem die decision logs
		wie aktive OPA Instanz

{DL_REMOTE_URL}
  reporting:
  min_delay_seconds:
<min>
  max_delay_seconds:
<max>
	gesendet werden. Dieser Parameter wird per environment Variable übergeben, sodass jeder Anbieter des ZT Cluster 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	

[<=]

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

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

Die Bundles sind immer signiert. Der OPA prüft die Signatur mit dem zur konfigurierten `keyid` passenden Schlüssel. Gültige Schlüssel und deren `keyid` werden über einen Downloadpunkt des PIP und PAP Service als JWKS geladen.

Die decision logs werden an die vom Anbieter des ZT Cluster festgelegte URL gesendet.

A_26664 - PDP - Durchsetzung der Policy-Konfiguration
Der PDP MUSS sicherstellen, dass nur die in der Konfiguration festgelegten Eigenschaften angewendet werden, insbesondere Herkunft der Policies und Daten sowie Signaturprüfung der Policies und Daten.**[<=]**

A_25450 - PDP - Policy nur vom gematik PIP und PAP Service
Der Anbieter des PDP MUSS sicherstellen, dass in den Policy Engines nur Policies und Daten vom gematik PIP und PAP Service importiert werden können. Eine unberechtigte Änderung der Konfiguration des PDP MUSS technisch ausgeschlossen werden.**[<=]**

A_25452 - PDP - Tamper-Proof Protokollierung von Administrationsaktivitäten
Der PDP MUSS ein "Tamper-Proof" Audit-Log von allen administrativen Vorgängen umsetzen. **[<=]**

A_25774 - PDP - Löschfristen für Auditeinträge des Admin Audit-Logs
Der PDP MUSS sicherstellen, dass die Löschung eines Auditeintrags den gesetzlichen Vorgaben entspricht und frühestens nach 12 Monaten erfolgt.**[<=]**

A_25775 - PDP - Kontrolle des Audit-Logs
Der Anbieter des ZT Clusters MUSS das Audit-Log mindestens alle 3 Monate im Vieraugenprinzip kontrollieren. Diese Rollen DÜRFEN NICHT an der Administration des ZT Clusters teilnehmen. Bei der Kontrolle ist insbesondere auf ungewöhnliche, nicht nachvollziehbare oder maliziöse Administratoraktivitäten zu achten. **[<=]**

A_25453 - PDP - Transparenz der installierte Policies
Der PDP MUSS sicherstellen, dass die gematik zu jeder Zeit feststellen kann, welche Policies und welche Policy-Versionen im PDP installiert sind.**[<=]**

A_25490 - PDP - Sicherheitsmeldung bei Änderungen und Aktualisierung
Der PDP MUSS sicherstellen, dass bei Aktualisierung und Änderungen der Policies oder PIP-Daten eine Sicherheitsmeldung an das Security Monitoring automatisiert übermittelt wird. **[<=]**

A_25771 - PDP - Automatisierte Prüfung nach Policy-Aktualisierungen
Der PDP muss alle 5 Minuten prüfen, ob Aktualisierungen der installierten und verwendeten Policy/PIP-Daten vorhanden sind.**[<=]**

A_25451 - PDP - Integritätsprüfung der Policies
Der PDP MUSS in der Policy Engine sicherstellen, dass Policies vom gematik PIP und PAP Service nur nach einer positiven Integritätsprüfung importiert werden können.**[<=]**

5.5.2 PDP Client Registry

A_26670 - PDP Client Registry - TLS Terminierung
Die Komponente PDP Client Registry MUSS TLS für eingehende Verbindungen von außerhalb des ZT Clusters terminieren können. Alternativ wird die TLS-Verbindung an der Komponente Ingress terminiert.**[<=]**

A_25644 - PDP Client Registry - Mobile Attestation
Die Komponente Client Registry MUSS die Clients/Apps bei der Registrierung über folgende Mechanismen attestieren:

- Android Key ID Attestation (für Google-Android Geräte)
- Apple DCAppAttest.
- SM(C)-B signiertes Client Assertion JWT
- TPM signiertes Client Assertion JWT (für die Authentisierung am TI-Gateway)
- ClientZertifikat plus Client Assertion JWT (nur für Dienst-zu-Dienst Kommunikation)

[<=]

A_25645 - PDP Client Registry - Attestation mittels TI-Smartcard
Die Komponente Client Registry MUSS die Registrierung über eine TI-Smartcard durchführen (SMC-B oder SM(C)-B),falls die Ausführungsumgebung des Clients keinen plattformseitigen Attestation-Mechanismen anbietet. Die Verwendung des zentralen IDP-Dienstes ist für die Nutzerauthentifizierung zulässig.**[<=]**

A_25648 - PDP Client Registry - Device Session Credentials
Die Komponente Client Registry MUSS die Client Credentials aufbewahren und Verifikationsmechanismen dem PDP Authorization Server bereitstellen.**[<=]**

A_25649 - PDP Client Registry - Regelmäßige Wiederholung der Attestation
Die Komponente Client Registry MUSS die Client Attestierung regelmäßig gemäß Festlegungen in der Device Policy wiederholen.**[<=]**

A_25650 - PDP Client Registry - TI-Identität in Attestation
Die Komponente Client Registry MUSS den registrierten Client an eine TI-Identität (KVNR oder TelematikID, festgestellt z. B. über die Einbindung des zentralen IDP-Dienstes oder eines Sektoralen IDP) binden.**[<=]**

Offener Punkt: Wie im Fall der Dienst-zu-Dienst Kommunikation die Bindung an eine Dienst-Identität erfolgt, wird in einer folgenden Version des Dokuments festgelegt (z. B. über Client-Registry, JWKS und Policy Engine oder [SPIFFE und SPIRE]*).*

A_25651 - PDP Client Registry - Offband Nutzer Verification

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

A_25653 - PDP Client Registry - 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 - PDP Client Registry - 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 - PDP Client Registry - Minimum Device Policy
Die Komponente Client Registry MUSS Client Mindestanforderungen vor der Registrierung verifizieren und kann hierfür eine Schnittstelle des PDP verwenden.**[<=]**

A_25738 - PDP Client Registry - Telemetrie Clientregistrierung
Die Komponente Client Registry MUSS in den Telemetrie-Daten 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 - PDP Client Registry - 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 PDP 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.

A_26585 - PDP Client Registry - Client-Daten
Die Komponente Client Registry MUSS die Client-Daten gemäß [client-instance.yaml] in der PDP Datenbank verwalten.
Die Client-Daten MÜSSEN bei Anfragen des Authorization-Servers an die Policy Engine im Request mit übergeben werden.**[<=]**

A_26668 - PDP Client Registry - Rate Limit
Die Komponente Client Registry MUSS für den Endpunkt ein Rate Limit konfigurierbar einstellen können. Wenn ein Rate Limit konfiguriert ist, dann muss der Client über Response Header-Informationen informiert werden ((das erlaubte Limit), (verbleibende Anfragen) und (Zeitpunkt, an dem das Limit zurückgesetzt wird)).**[<=]**

5.5.2.1 Sicherheits- und Datenschutz-Anforderungen an die PDP Client Registry

A_25751 - PDP Client Registry - Anwendungsfälle nur für registrierte mobile Clients
Nach der erfolgreichen Registrierung des ersten mobilen Clients (Gerät/App Kombination), MUSS die Komponente Client Registry sicherstellen, dass die folgenden Anwendungsfälle nur von einem registrierten mobilen Client durchgeführt werden können:

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

[<=]

A_25748 - PDP Client Registry - Maximale Anzahl von Geräten
Die Komponente Client Registry MUSS sicherstellen, dass ein Nutzer maximal 256 Clients registrieren kann. Der Wert muss konfigurierbar sein.**[<=]**

A_25749 - PDP Client Registry - Nutzer Protokollierung
Die Komponente Client Registry MUSS ein Nutzerprotokoll führen und die folgenden Anwendungsfälle für den Nutzer protokollieren:

- Client hinzufügen
- Client löschen
- Client umbenennen
- E-Mail-Adresse hinzufugen
- E-Mail-Adresse aktualisieren

[<=]

A_25750 - PDP Client Registry - Nutzer über sicherheitsrelevante Ereignisse informieren
Die Komponente Client Registry MUSS sicherstellen, dass der Nutzer bei folgenden Anwendungsfällen informiert wird:

- Client hinzufügen
- Client löschen
- Client 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.

5.5.3 PDP Relying Party

A_25655 - PDP - Relying Party

Der PDP Authorization Server MUSS in der TI-Föderation als Relying Party registriert sein. **[<=]**

A_25656 - PDP - Entity Statement
Der PDP Authorization Server MUSS die Redirect-URLs aller zulässigen Clients als erlaubte Redirect-URLs im Entity Statement ausweisen.**[<=]**

A_25657 - PDP - Authentication über sektoralen IDP
Der PDP Authorization Server MUSS die Nutzer über sektorale IDPs authentifizieren können.**[<=]**

A_25658 - PDP - Authentication über SmartCard IDP
Der PDP Authorization Server MUSS die Nutzer über den zentralen IDP-Dienst (SmartCard-IDP) authentifizieren können.**[<=]**

5.5.4 PDP Authorization Server

A_25760 - PDP Authorization Server - OAuth2 Schnittstellen
Der PDP Authorization Server MUSS eine OAuth2 Schnittstelle gemäß [RFC6749] und [RFC7636] implementieren.
Der Authorization Server MUSS am Token Endpunkt REFRESH_TOKEN entsprechend [RFC6749] ausstellen können.**[<=]**

A_26669 - PDP Authorization Server - TLS Terminierung
Die Komponente PDP Authorization Server MUSS TLS für eingehende Verbindungen von außerhalb des ZT Clusters terminieren können. Alternativ wird die TLS-Verbindung an der Komponente Ingress terminiert.**[<=]**

A_25659 - PDP Authorization Server - Check Device Registration
Der PDP Authorization Server MUSS die Client Instanzen über den Mechanismus JSON Web Token Client Authentication gemäß [RFC7523] mit DPoP gemäß [RFC9449] authentifizieren und Anfragen, die den Mechanismus nicht verwenden, ablehnen.**[<=]**

A_25660 - PDP Authorization Server - Session Management mittels Access Token und Refresh Token
Die Komponente Authorization Server MUSS ein Session Management mittels OAuth2 und Ausgabe, Verwaltung und Entzug von Access und Refresh Token gemäß [RFC6749#1.5] unterstützen.
Die Komponente Authorization Server MUSS die Session-Daten gemäß [session.yaml] und die User-Daten gemäß [user-info.yaml] in der PDP Datenbank verwalten.**[<=]**

A_26944 - PDP Authorization Server - Access Token Inhalt
Die Komponente Authorization Server MUSS Access Token mit Attributen gemäß [access-token.yaml] ausstellen.**[<=]**

A_26945 - PDP Authorization Server - Refresh Token Inhalt
Die Komponente Authorization Server MUSS Refresh Token mit Attributen gemäß [refresh-token.yaml] ausstellen.**[<=]**

A_25661 - PDP Authorization Server - Umsetzung der Policy Decision
Die Komponente Authorization Server MUSS die Zugriffs-Entscheidung eines PDP mittels der Ausgabe eines Access und eines Refresh Tokens umsetzen bzw. eine Zugriffsverweigerung mit einem http-Statuscode 403 quittieren.
Die Claims im Access und im Refresh Token MÜSSEN zur Entscheidung der Policy Engine für die angefragten Claims passen.**[<=]**

A_25662 - PDP Authorization Server - Refresh Token Rotation
Die Komponente Authorization Server MUSS eine Refresh Token Rotation gemäß [RFC6749#10.4] erzwingen und MUSS sicherstellen, dass ein Refresh- Token nur einmal gegen ein Access Token und ein Refresh Token getauscht werden kann.

Die Komponente Authorization Server MUSS erzwingen, dass der Nutzer eine Authentisierung durchführen muss, wenn seit der letzten Authentisierung die Zeit der Gültigkeitsdauer des Refresh Tokens abgelaufen ist.**[<=]**

A_25663 - PDP Authorization Server - Token-Binding an Device-Registration
Die Komponente Authorization Server MUSS auszugebende Access Token und Refresh Token über den Mechanismus OAuth 2.0 Demonstrating Proof of Possession (DPoP) gemäß [RFC9449] an die identifizierte Client-Instanz binden, indem im Token-Binding-Claim die Angabe der Clientidentifikation als `"jkt"` eineindeutig referenziert wird.**[<=]**

A_25664 - PDP 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.**[<=]**

A_25665 - PDP 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 8: PDP Authorization Server - Plugin-Schnittstelle Application Authorization Backend

Operation	Operation Kennung	Input	Output
Benachrichtigung über die Ablehnung des Zugriffs durch PDP	notifiyAccessDenied	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	-

[<=]

A_26586 - PDP Authorization Server - Session- und Nutzer-Daten
Die Komponente Authorization Server MUSS nach jeder vollständigen und erfolgreichen Authentifizierung Session-Daten gemäß [session.yaml] und Nutzer-Daten gemäß [user-info.yaml] in einer Datenbank des PDP speichern.
Die Session-Daten und Nutzer-Daten MÜSSEN bei Anfragen des Authorization-Servers an die Policy Engine im Request mit übergeben werden.**[<=]**

A_26972 - PDP Authorization Server - Nutzer-Daten aus der SM(C)-B
Die Komponente Authorization Server MUSS im Falle der SM(C)-B Authentifizierung die folgenden Daten aus dem SM(C)-B Zertifikat auslesen und als Nutzer-Daten gemäß [user-info.yaml] in der PDP Datenbank speichern:

Tabelle 9 SM(C)-B_Nutzer-Daten

SM(C)-B Daten (C.HCI.OSIG gemäß [gemSpec_PKI])	Nutzer-Daten gemäß [user-info.yaml]	Beschreibung
Extension Admission, registrationNumber	identifier	Telematiik-ID Eindeutige ID der Organisation des Gesundheitswesens
subject, commonName	commonName	Wird übernommen, wenn im Zertifikat vorhanden. „Kurzname“ der Institution, so wie sie sich selbst auf dem Anschriftenfeld findet.
Extension Admission, professionOID	professionOID	OID der Institution gemäß [gemSpec_OID#GS-A_4443-*]
subject, organizationName	organizationName	Wird übernommen, wenn im Zertifikat vorhanden. Name der Organisation/Einrichtung des Gesundheitswesens

[<=]

A_26973 - DP Authorization Server - Nutzer-Daten aus id_token
Die Komponente Authorization Server MUSS im Falle der OIDC Authentifizierung für Versicherte alle Claims aus dem id_token gemäß [gemSpec_IDP_Sek] auslesen und als Nutzer-Daten gemäß [user-info.yaml] in der PDP Datenbank speichern.
Dabei gilt das folgende Mapping für die Pflicht-Nutzer-Daten.

Tabelle 10 id_token_Nutzer-Daten

id_token claims (gemäß [gemSpec_IDP_Sek])	Nutzer-Daten gemäß [user-info.yaml]	Beschreibung
urn:telematik:claims:id	identifier	Für Versicherte der unveränderliche Anteil der KVNR
urn:telematik:claims:profession	professionOID	OID für Versicherte

[<=]

5.5.4.1 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.
Host: <FQDN des Authorization Servers>
```

Zusätzlich können Clients das Well-known json Dokument unter

```
GET /.well-known/oauth-protected-resource
 Host: <FQDN des Resource Servers>
```

vom Resource Server abfragen (siehe https://www.ietf.org/archive/id/draft-ietf-oauth-resource-metadata-13.html).

A_26037 - PEP http Proxy, Well-known für PDP Authorization Server
Die Komponente http Proxy MUSS für den Authorization Server des ZT Clusters 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 https://raw.githubusercontent.com/gematik/spec-t20r/main/src/schemas/as-well-known.yaml validiert werden können.
Das Attribut `scopes_supported` MUSS mindestens die folgenden Werte enthalten:
- `zero:register`
- `zero:manage`

Weitere Werte für das Attribut `scopes_supported` müssen per Konfiguration ergänzt werden können.
Es MUSS ein Attribut `nonce_endpoint` enthalten sein, dass die URL zur Abfrage neuer Nonce-Werte angibt.
Es MUSS ein Attribut `openid_providers_endpoint` enthalten sein, dass die URL zur Abfrage der unterstützten OpenID Provider enthält.
Es MUSS möglich sein, weitere Authorization Server zu konfigurieren, denen der PEP http Proxy vertraut. Für diese MUSS ein eigenes Well-known json Dokument nach den oben genannten Regeln erzeugt werden. Die URL zum Download des Well-known json Dokuments MUSS einen Pfadanteil haben: https://<as-fqdn>/<path>/.well-known/oauth-authorization-server .**[<=]**

Hinweis: Der FQDN des Authorization Servers wird vom Anbieter des TI 2.0 Dienstes vergeben.
Die unterstützten OpenID Provider sind in der PU https://idp.app.ti-dienste.de/directory/fed_idp_list und in der RU https://idp-ref.app.ti-dienste.de/directory/fed_idp_list.

A_26090 - PDP Authorization Server, Well-known Erstellung
Die Komponente http Proxy MUSS für den Authorization Sever das Well-known json Dokument aus den Konfigurationsdaten des Zero Trust-Clusters erzeugen.
[<=]

Beispiel Well-known json Dokument des Authorization Servers:

```
{
  "issuer": "https://zerobin.zt.dev.ccs.gematik.solutions",
  "authorization_endpoint":
"https://zerobin.zt.dev.ccs.gematik.solutions/auth",
  "token_endpoint": "https://zerobin.zt.dev.ccs.gematik.solutions/token",
  "jwks_uri": "https://zerobin.zt.dev.ccs.gematik.solutions/jwks",
  "nonce_endpoint": "https://zerobin.zt.dev.ccs.gematik.solutions/nonce"
  "openid_providers_endpoint":
"https://idp.app.ti-dienste.de/directory/fed_idp_list"
  "scopes_supported": [
    "zero:register",
    "zero:manage"
  ],
  "response_types_supported": [
    "code"
  ],
  "response_modes_supported": [
    "query"
```

```
  ],
  "grant_types_supported": [
    "authorization_code"
  ],
  "token_endpoint_auth_methods_supported": [
    "none"
  ],
  "token_endpoint_auth_signing_alg_values_supported": [
    "ES256"
  ],
  "service_documentation": "https://gihub.com/gematik/zero-lab",
  "ui_locales_supported": [
    "de",
    "en"
  ],
  "code_challenge_methods_supported": [
    "S256"
  ]
}
```

5.5.4.2 Ablauf der SM(C)-B Authentifizierung mit DPoP

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

Die SM(C)-B Authentifizierung erfolgt bei freigeschalteter SM(C)-B automatisch (ohne Aktion des Leistungserbringers) durch den Trust Client gesteuert. Die Clientregistrierung erfolgt implizit während der Authentifizierung.

A_26091 - ZT Cluster, SM(C)-B Authentifizierung mit DPoP
Der Zero Trust-Cluster MUSS den Ablauf gemäß Abbildung SM(C)-B_*Authentisierung_mit_DPoP* sowie gemäß [RFC7523] und [RFC9449] unterstützen.**[<=]**

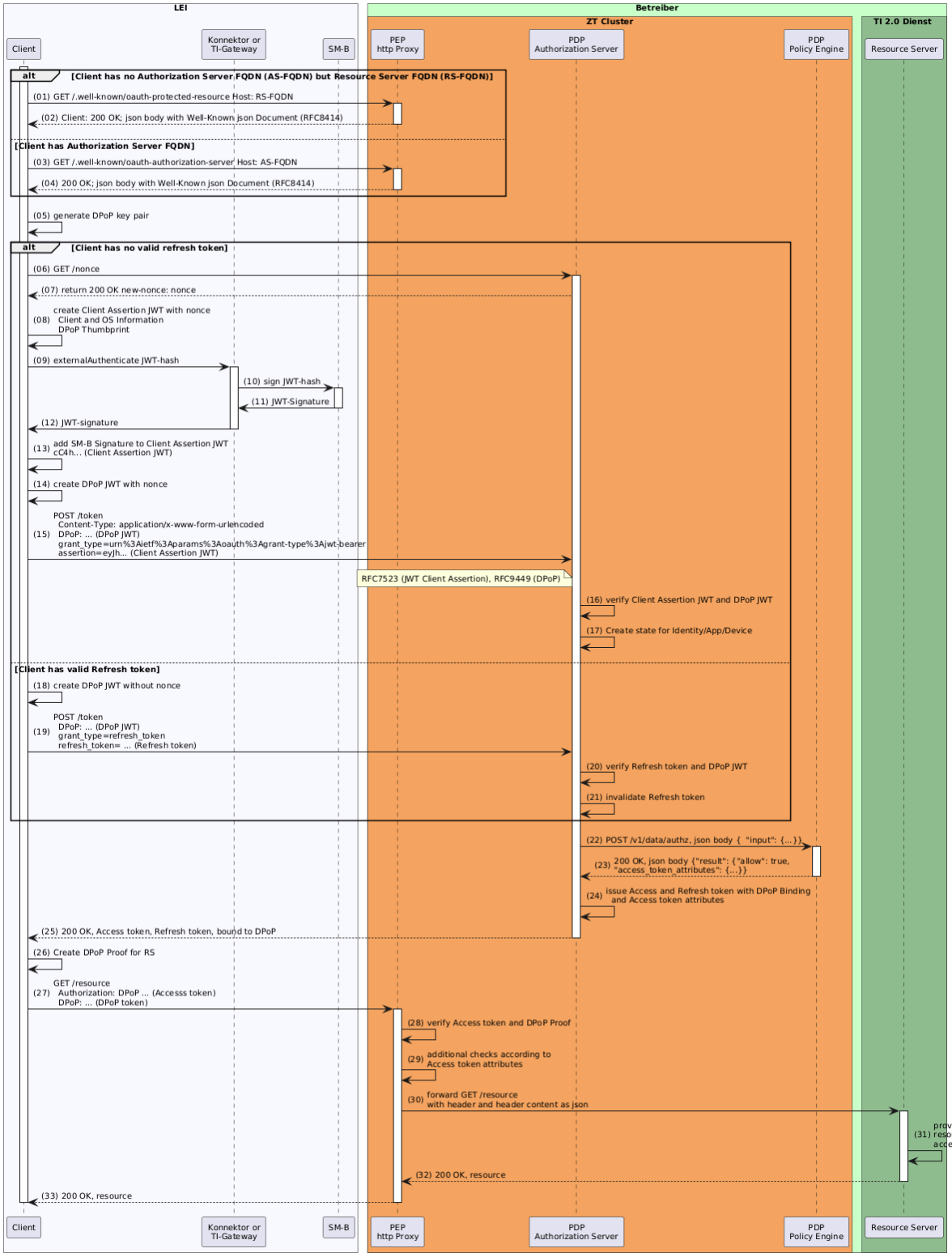

Abbildung 3: SM(C)-B Authentisierung mit DPoP

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

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

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

```
{
     "crv": "P-256",
     "kty": "EC",
     "x": "...",
     "y": "...",
     "d": "..."
}
```

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

```
HEAD /nonce http/1.1
Host: as.example.com

http/1.1 200 OK
new-nonce: ...Nonce from the AS...
```

Danach erzeugt der Client in Schritt (08) bis (13) das Client Assertion JWT, in dem gemäß [client-instance.yaml] auch Informationen über das Gerät, das Betriebssystem und die App (PS) enthalten sind. Das JWT wird mit der SM(C)-B signiert. Als Algorithmus wird `"alg": "BP256R1"`

verwendet. Im Attribut `"jkt"` ist der Hash des öffentlichen Schlüssels des Client DPoP Keys enthalten.

```
{
    "nonce": "...Nonce from the AS...",
    "iss":"urn:telematik:telematik-id:9-123456789", // Telematik ID from
X.509 certificate
    "sub":"...Client ID...",
    "aud":" https://as.example.com", // AS URL
    "iat":1562262611,
    "exp":1562266216,
    "cnf": {
        "jkt":"..thumbprint of the DPoP key..."
    },
    "urn:telematik:client-self-assessment": {
        "product_id": "PS-000",
        "product_version": "0.5.0",
        "manufacturer_id": "HRST-001",
        "platform": "software"
        "runtime": {
            "os": "Microsoft Windows",
            "os_version": "10.0.19045.4291",
            "os_arch": "x86",
        },
}
```

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

```
{
  "typ":"dpop+jwt",
  "alg":"ES256",
  "jwk": {
    "kty":"EC",
```

```
    "x":"l8tFrhx-34tV3hRICRDY9zCkDlpBhF42UQUfWVAWBFs",
    "y":"9VE4jf_Ok_o64zbTTlcuNJajHmt6v9TDVrU0CdvGRDA",
    "crv":"P-256"
  }
}
.
{
  "jti":"-BwC3ESc6acc2lTc",
  "htm":"POST",
  "htu":"https://server.example.com/token",
  "nonce":"...nonce from the AS..."
  "iat":1562262616
}
.
ES256 signature of the DPoP JWT
```

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

```
POST /token HTTP/1.1
Host: as.example.com
grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer
assertion=...Client Assertion JWT signed by SM-C...
DPoP: eyJh...
scope=...
```

Der Authorization Server prüft die im Request übergebenen Client Assertion JWT und DPoP JWT (16) und erzeugt oder aktualisiert den Eintrag für den SM(C)-B Nutzer und seine App in der Client Registry (17). Der Authorization Server muss die Signatur des Client Assertion JWT mit `"alg": "BP256R1"` prüfen können.

Wenn ein gültiges Refresh Token vorhanden ist, dann kann der Client anstatt der Schritte (06) bis (17) die Schritte (18) und (19) nutzen, um das Refresh Token gegen ein neues Access Token und ein neues Refresh Token einzutauschen. Das verwendete Refresh Token wird geprüft (20) und verliert sofort seine Gültigkeit (21).

In Schritt (22) und (23) wird durch die Policy Engine die Entscheidung getroffen, ob die vom Client übergebenen Header-Daten hinreichend sind, um dem Authorization Server zu erlauben, für den Client neue Access und Refresh Token auszustellen. Wenn ein Refresh Token verwendet wurde, übergibt der Authorization Server zusätzlich Daten aus dem Eintrag in der Client Registry an die Policy Engine. Die Policy Engine kann zusätzliche Attribute übergeben, die vom Authorization Server in das Access Token übernommen werden. Mit diesen Attributen wird gesteuert, welche Prüfschritte der PEP ausführt.

Wenn die Erlaubnis erteilt wurde, stellt der Authorization Server neue Access und refresh token mit DPoP Binding aus (24) und sendet sie an den Client (25).

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "...",
    "token_type": "DPoP",
    "expires_in": 3600,
    "refresh_token": "...",
    "scope": "..."
}
```

```
Inhalt des Access Token
```

```
{
    "alg": "ES256",
```

```
    "kid": "...",
}
.
{
    "iss": " https://as.example.com",
    "sub": "...Client ID...",
    "aud": "...Client ID...",
    "exp": 1562266216,
    "cnf": {
        "jkt": "...thumbprint of the DPoP key..."
    }
}
.
ES256 signature of the access token
```

Der Client erzeugt ein neues DPoP Proof JWT für den Zugriff auf Daten des Resource Servers (26) und sendet den Request, mit Access Token und DPoP Proof im Header, an den Resource Server (27).

Der http Proxy prüft das Access Token, das DPoP Binding (28) und führt weitere Prüfschritte gemäß den Attributen im Access Token aus (29). Wenn alle Prüfungen erfolgreich waren, leitet der PEP den Request an den Resource Server weiter (30).

Der Resource Server empfängt den Request, führt seine Fachlogik aus (31) und sendet eine Antwort an den http Proxy (32), der diese an den Client weiterleitet (33).

Offener Punkt: Details zur Signaturprüfung der JWT werden in einer folgenden Version des Dokuments festgelegt. Grundsätzlich werden die folgenden Schritte durchgeführt.

1. ***JWT Header-Überprüfung:***
   - *Algorithmus: Zuerst überprüfen, ob der Algorithmus akzeptabel ist, um unnötige Verarbeitung zu vermeiden. Gültige Algorithmen sind in [gemSpec_Krypt] festgelegt.*
2. ***Payload-Überprüfung:***
   - *Ablaufdatum (exp): prüfen, um abgelaufene Tokens sofort abzulehnen.*
   - *Audience (aud): prüfen, ob das Token für den beabsichtigten Empfänger bestimmt ist.*
   - *Issuer (iss): Sicherstellen, dass der Aussteller vertrauenswürdig ist.*
3. ***Integrität der Nachricht:***
   - *Hash-Überprüfung: Sicherstellen, dass die Nachricht nicht manipuliert wurde.*
4. ***Schlüsselvalidierung:***
   - *Öffentlicher Schlüssel: Überprüfen, ob der Schlüssel gültig und vertrauenswürdig ist.*
5. ***Vertrauenswürdigkeit der Schlüsselkette:***
   - *Federation Master: Bestätigen, dass der Schlüssel von einer vertrauenswürdigen Quelle stammt.*
6. ***Spezifische Prüfungen für TI-Zertifikate:***
   - *Ablaufdatum des Zertifikats: Sicherstellen, dass das Zertifikat noch gültig ist.*
   - *Widerrufsstatus des Zertifikats (OCSP): Prüfen, ob das Zertifikat nicht widerrufen wurde. Die OCSP Response wird für eine konfigurierbare Dauer*

zwischengespeichert, um zu häufige OCSP Anfragen zu verhindern. Die Dauer soll der Gültigkeitsdauer des Refresh Token entsprecchen.

- *Vertrauenswürdigkeit der Zertifikatskette: Sicherstellen, dass die Kette zu einer vertrauenswürdigen Root-CA führt.*

Offener Punkt: Für DPoP Schlüssel und Token gelten folgende Vorgaben. Weitere Details werden in einer folgenden Version des Dokuments festgelegt.

1. DPoP Schlüssel:

- *Zufälligkeit: Die Schlüssel müssen unter Verwendung eines starken Zufallszahlengenerators erzeugt werden. Eine unzureichende Zufälligkeit kann die Sicherheit des Systems kompromittieren.*
- *Schlüsselauswahl: Gültige Algorithmen sind in [gemSpec_Krypt] festgelegt.*
- *Schlüsselspeicherung: Die privaten Schlüssel müssen sicher gespeichert werden.*
- *Regelmäßige Erneuerung: Die Schlüssel sollten regelmäßig erneuert werden, um das Risiko eines Kompromittierens zu verringern. Das Auslaufdatum soll entsprechend der Gültigkeitsdauer des Refresh Token gesetzt sein.*
- *Sicheres Verfahren: Der Prozess der Schlüsselerneuerung muss sicher gestaltet sein.*

2. DPoP Token Signaturüberprüfung:

- *Typ: Prüfen, dass der JWT-Typ dpop+jwt ist.*
- *Gültigkeit der Signatur: Die Signatur des DPoP Tokens muss mit dem öffentlichen Schlüssel des Clients verifiziert werden.*
- *Algorithmus: Der verwendete Signaturalgorithmus muss mit dem im öffentlichen Schlüssel angegebenen Algorithmus übereinstimmen.*
- *Auslaufdatum: Das Auslaufdatum für den Schlüssel muss überprüft werden. Das Auslaufdatum soll entsprechend der Gültigkeitsdauer des Refresh Token gesetzt sein.*
- *Nonce: Die Nonce muss geprüft werden.*

5.5.4.3 Ablauf der Authentifizierung bei Dienst-zu-Dienst Kommunikation

Offener Punkt: Grundsätzlich gilt, dass Dienste, die auf einen TI 2.0 Dienst zugreifen, sich mit mTLS (z. B. mit SPIFFE und SPIRE) oder Client Assertion JWT + DPoP (ähnlich wie SM(C)-B Authentifizierung mit DPoP.; anstatt der SM(C)-B wird ein Signaturschlüssel des Dienstes verwendet) authentifizieren müssen. Details zum Ablauf der Authentifizierung bei Dienst-zu-Dienst Kommunikation und zu den Schlüsseln werden in einer folgenden Version des Dokuments festgelegt.

A_26902 - PDP Authorization Server - Rate Limit
Die Komponente Authorization Server MUSS an seinen Endpunkten ein Rate Limit konfigurierbar einstellen können. Wenn ein Rate Limit konfiguriert ist, dann muss der Client über Response Header-Informationen informiert werden ((das erlaubte Limit), (verbleibende Anfragen) und (Zeitpunkt, an dem das Limit zurückgesetzt wird)).**[<=]**

5.5.5 PDP Datenbank

Die Datenbank speichert Session-, Nutzer- und Client-Daten. Die Struktur der Daten ist in den Schemadateien [session.yaml], [user-info.yaml], und [client-instance.yaml] festgelegt.

Die Session Daten gemäß [session.yaml] enthalten die Attribute `Access Token_jti` zur Verlinkung der Session mit dem aktuell gültigen Access Token (Attribut `jti`) und `Refresh Token_jti` zur Verlinkung der Session mit dem aktuell gültigen Refresh Token (Attribut `jti`). Das Attribut `subject` enthält die Verlinkung mit den User-Daten (Attribut `subject`) und das Attribut `client_id` enthält die Verlinkung mit den Client-Daten (Attribut `client_id`).

Abbildung 4: Beziehungen zwischen Session-, Nutzer- und Client-Daten sowie Token

A_26587 - PDP Datenbank - Schnittstellen
Die Komponente Datenbank MUSS REST Schnittstellen anbieten, sodass

- der Authorization Server die Operationen create, retrieve, update und delete für Session-Daten gemäß [session.yaml] ausführen kann,
- der Authorization Server die Operationen create, retrieve, update und delete für User-Daten gemäß [user-info.yaml] ausführen kann,
- die Client Registry die Operationen create, retrieve, update und delete für Client-Daten gemäß [client-instance.yaml] ausführen kann und
- der PEP http Proxy die Operation retrieve für User-Daten und optional im gleichen Request für Client-Daten ausführen kann. Anhand des Parameters jti aus dem Access Token des Requests MUSS der http Proxy die User- und Client-Daten der zugehörigen Session aus der Datenbank abfragen können.

[<=]

A_26588 - PDP Datenbank - Konfiguration
Die Komponente Datenbank MUSS so konfigurierbar sein, dass für alle Attribute der Client-Daten gemäß [client-instance.yaml] eingestellt werden kann, welche Attribute an den PEP http Proxy ausgegeben werden.
Die default Einstellung ist, dass die Attribute `platform`, `product_name` und `product_version` sowie aus dem Objekt posture die Attribute `system_name`, `system_version` und `device_model` ausgegeben werden.**[<=]**

5.5.6 Sicherheits- und Datenschutzanforderungen an den PDP

A_25449 - PDP- Nutzeridentität nur von einem zugelassenem IDP

Der PDP MUSS sicherstellen, dass nur Nutzeridentitäten von einem zugelassenen IDP akzeptiert werden.**[<=]**

A_25447 - PDP Authorization Server - Kommunikation nur mit authentischer Policy Engine
Der PDP Authorization Server MUSS sicherstellen, dass er mit einer authentischen und korrekt konfigurierten Policy Engine kommuniziert.**[<=]**

A_25486 - PDP - Abbruch durch Anomalie Signale
Falls das Security Monitoring eine Anomalie beim Zugriff eines Clients signalisiert, MUSS der PDP Authorization Server das Access Token des Clients annullieren und damit die aktuelle fachliche Operation abbrechen. **[<=]**

A_26269 - PDP - Tamper-Proof Protokollierung von Administrationsaktivitäten
Der PDP MUSS ein "Tamper-Proof" Audit-Log von allen administrativen Vorgängen umsetzen.**[<=]**

A_26281 - PDP - kurzlebige Zertifikate
Der PDP Authorization Server MUSS für die Signatur von Token kurzlebige Self-signed Zertifikate verwenden.
Die Gültigkeitsdauer der Signatur-Zertifikate MUSS in den ZT Cluster-Manifest-Dateien konfigurierbar sein.**[<=]**

Hinweis: Initial wird eine Gültigkeitsdauer von 48 Stunden für die Zertifikate festgelegt. Der Wert kann entsprechend den aktuellen betrieblichen Anforderungen durch die gematik geändert werden.

5.5.7 Konfiguration

A_26038 - PDP, Konfigurations-Parameter
Der PDP MUSS die folgenden Konfigurations-Parameter unterstützen.

Tabelle 11: PDP - Konfigurations-Parameter

Konfigurations-Parameter	Default Wert	Beschreibung
as-fqdn	n/a	FQDN des PDP Authorization Servers
scopes_supported		vom PDP 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.

[<=]

Offener Punkt: Weitere Konfigurationsparameter werden in einer zukünftigen Version des Dokuments festgelegt und hier ergänzt.

5.6 Anforderungen an den PIP und PAP Service

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

A_25670 - PIP und PAP - Bereitstellung Download-Endpunkt
Der PIP und PAP Service MUSS Download-Endpunkte für OPA Bundles gemäß OpenAPI Spezifikation [pip-pap-service.yaml] Version 1.0.0 in den folgenden Instanzen bereitstellen:
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. **[<=]**

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

A_25671 - PIP und PAP - TLS am Download-Endpunkt
Der PIP und PAP Service MUSS sich beim TLS-Verbindungsaufbau am Download-Endpunkt gegenüber Clients mit einem Extended Validation TLS-Zertifikat eines Herausgebers gemäß [CAB-Forum] authentisieren.**[<=]**

A_25672 - PIP und PAP - Kompatibilität mit OPA Bundles
Der PIP und PAP Service MUSS die Policies und Daten als [OPA Bundle] 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 die aktive Policy Engine bereitgestellt. Unter dem label "latest-sim" werden Bundles für die Simulations-Policy Engine bereitgestellt.
Der PIP und PAP Service bezieht die OPA Bundles aus einem Git Repository der gematik. Die Bundles werden in einem GitOps CI Prozess mit Quality Gates entwickelt und für die Policy Engines 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 gemäß RFC7232 verwenden, das aus dem SHA-256 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 Andminstrationsaktivitäten
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 Anbieter des PIP und PAP Services MUSS das Audit-Log mindestens alle 3 Monate 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.**[<=]**

5.7 Anforderungen an den Betrieb der Zero Trust-Komponenten

Die Komponenten des ZT Cluster werden als Kubernetes (k8s) Cluster betrieben. In einem von der gematik vorgegebenen Git Repository werden die Konfigurationsdateien des k8s Clusters bereitgestellt, mit denen die aktuelle Version des Clusters erstellt und ausgeführt werden kann. Im Cluster ist zusätzlich ein Cluster Management Service (eine Continuous Delivery (CD) Komponente) enthalten, der den Betriebszustand des Clusters überwacht und regelmäßig prüft, ob eine neuere Version des Clusters im Git Repository verfügbar ist, und ggf. das Cluster automatisch aktualisiert.

A_25773 - Zero Trust-Cluster - Nutzung der von der gematik bereitgestellten Zero Container Images
Der Anbieter eines Dienstes der TI 2.0 MUSS die von der gematik bereitgestellten Container Images im Zero Trust-Cluster verwenden, um den Zugang zum TI 2.0 Dienst zu kontrollieren. **[<=]**

A_25776 - Zero Trust-Cluster - Änderung der Konfiguration
Der Anbieter eines Dienstes der TI 2.0 MUSS die Kubernetes Manifeste (Konfiguration) des Zero Trust-Clusters aus dem git Repository der gematik verwenden. **[<=]**

Hinweis: Die Kubernetes Manifeste (Konfiguration) des ZT Clusters werden per git submodule in das git Repository der gematik integriert, sodass der Anbieter des ZT Clusters seine Konfiguration selbständig anpassen kann.

5.7.1 Anforderungen für nahtlose Aktualisierungen

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

A_25784 - Zero Trust-Komponenten - Download von Aktualisierungen im Hintergrund
Die Komponente der Zero Trust-Architektur MUSS in der Lage sein, Aktualisierungen im Hintergrund herunterzuladen, ohne den laufenden Betrieb zu beeinträchtigen.**[<=]**

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

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

A_26104 - Zero Trust-Komponenten - Protokollieren von Änderungen
Die Komponente der Zero Trust-Architektur MUSS jede Änderung protokollieren, einschließlich des Zeitpunkts der Änderung und des Administrators, der die Änderung vorgenommen hat.**[<=]**

A_25786 - Zero Trust-Komponenten - Abschluss von Transaktionen vor Aktualisierung
Die Komponente der Zero Trust-Architektur MUSS sicherstellen, dass alle aktuellen Transaktionen und Anfragen abgeschlossen oder ordnungsgemäß übernommen werden, bevor ein Update finalisiert wird.**[<=]**

A_25787 - Zero Trust-Komponenten - Gewährleistung der Systemintegrität während Aktualisierungen
Die Komponente der Zero Trust-Architektur MUSS während des gesamten Aktualisierungsprozesses die Systemintegrität und Sicherheitsrichtlinien aufrechterhalten. **[<=]**

A_25788 - Zero Trust-Komponenten - Unterstützung von Rollbacks
Die Komponente der Zero Trust-Architektur MUSS die Fähigkeit besitzen, zu einer stabilen Vorversion zurückzukehren, sollte eine Aktualisierung fehlerhaft sein oder abgebrochen werden müssen.**[<=]**

A_25789 - Zero Trust-Komponenten - Schnelle Rollback-Durchführung
Die Komponente der Zero Trust-Architektur MUSS Rollbacks schnell und ohne manuelle Eingriffe durchführen können.**[<=]**

5.7.2 Anforderungen für Steuerung durch Feature-Flags

A_25790 - Zero Trust-Komponenten - Aktivierung/Deaktivierung von Funktionen in Echtzeit
Die Komponente der Zero Trust-Architektur MUSS Funktionen oder Verhaltensweisen zur Laufzeit durch Feature-Flags aktivieren oder deaktivieren können, ohne dass ein Neustart erforderlich ist.**[<=]**

Hinweis: Feature Flags dürfen nur unter 4 Augen und unter strenger Einhaltung des Change-Management Prozesses geändert werden.

5.7.3 Anforderungen zur Überwachung des Betriebsstatus

A_25794 - Zero Trust-Komponenten - Implementierung von Health Checks
Die Komponente der Zero Trust-Architektur MUSS Health Checks implementieren, um ihren aktuellen Zustand und ihre Verfügbarkeit zu überwachen. **[<=]**

A_25797 - Zero Trust-Komponenten - Health Check Schnittstelle für gematik Monitoring
Der Anbieter des ZT Cluster MUSS die Schnittstellen zu Health Checks des ZT Cluster dem gematik Monitoring zur Verfügung stellen.**[<=]**

A_25795 - Zero Trust-Komponenten - Automatische Antwort auf Health Check Anfragen
Die Komponente der Zero Trust-Architektur MUSS automatisch auf Health Check Anfragen antworten können, um ihre Funktionalität und Verfügbarkeit zu 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. **[<=]**

Hinweis: Im ZT Cluster ist dafür der Telemetrie-Daten Service vorgesehen, der die Daten an ein Monitoring System des Anbieters weitergibt.

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 im Rahmen der Zero Trust-Komponenten nicht an. Diese sind bei Bedarf vom jeweiligen TI 2.0 Dienst eigenständig einzusetzen.

5.7.4 Leistungs-Anforderungen

A_26486 - PEP http Proxy - Performance
Die Komponente PEP http Proxy MUSS bei einem Deployment in eine Cloud Umgebung ermöglichen, dass eingehende Requests unter Last innerhalb von 0,1 Sekunden geprüft und bei erfolgreicher Prüfung weitergeleitet werden.**[<=]**

A_26487 - PEP http Proxy -Skalierbarkeit
Die Komponente PEP http Proxy MUSS horizontal skalierbar sein, sodass mit jedem zusätzlichen Pod mindestens 75% der Leistung eines einzigen Pods verfügbar werden. **[<=]**

A_26488 - PEP http Proxy - Last
Die Komponente PEP http Proxy MUSS pro Pod mehr als 300 Websocket Verbindungen und mehr als 300 Requests pro Sekunde unterstützen können.**[<=]**

A_26489 - PDP Authorization Server - Performance
Die Komponente PDP Authorization Server MUSS bei einem Deployment in eine Cloud Umgebung ermöglichen, dass eingehende Requests unter Last innerhalb von 0,2 Sekunden bearbeitet und beantwortet werden.**[<=]**

A_26490 - PDP Authorization Server - Skalierbarkeit
Die Komponente PDP Authorization Server MUSS horizontal skalierbar sein, sodass mit jedem zusätzlichen Pod mindestens 75% der Leistung eines einzigen Pods verfügbar werden.**[<=]**

A_26491 - PDP Authorization Server - Last
Die Komponente PDP Authorization Server MUSS pro Pod über alle Endpunkte zusammen mehr als 300 Requests pro Sekunde unterstützen können.**[<=]**

Hinweis: Anfragen an abhängige Dienste im Hintergrund, um einen Request vollständig zu bearbeiten (z.B. OCSP), werden bei den Bearbeitungszeiten mit berücksichtigt, jedoch nicht dem abfragenden Dienst zur Last gelegt.

5.7.5 Betriebliche Schnittstellendefinition der Zero Trust-Komponenten

Die Komponenten des ZT Cluster 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 Schnittstellendefinitionen aus [gemKPT_Betr].

Tabelle 12: Tab_gemF_Zero-Trust_Schnittstellendefinition_ZT_Cluster

Endpunkt / Anwendungsfall	Beschreibung
/.well-known/	Abruf gültiger Autorisierungsserver
GET /nonce/	Nonce abrufen
POST /token <JWT Client Assert>	Autorisierung ohne Refresh Token
POST /token <Refresh Token>	Autorisierung mit Refresh Token

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

Tabelle 13: Tab_gemF_Zero-Trust_Schnittstellendefinition_PIPPAP

Endpunkt / Anwendungsfall	Beschreibung
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 Dienste der TI

Dienste der TI (Resource Server), die durch einen ZT Cluster geschützt sind, erhalten nur Requests von Clients oder anderen Diensten, wenn der PDP des ZT Cluster den Zugriff gewährt und ein Access Token ausgestellt hat. Der PEP des ZT Clusters setzt durch, dass nur Requests mit gültigem Access Token zum Resource Server gelangen.

5.9 Anforderungen an den Test der Zero Trust-Komponenten

Die Teststrategie der gematik für die TI 2.0 Anwendungen befindet sich aktuell in der Abstimmung mit den Gesellschaftern. Das daraus abzuleitende konkrete Testkonzept für Zero Trust und die daraus resultierenden Anforderungen an die Zero Trust Umsetzung sind dadurch noch nicht für eine Vorveröffentlichung verbindlich festgelegt.
Sobald die Teststrategie der gematik abgestimmt ist, wird dieses Kapitel entsprechend angepasst.

6 Beispiele und Referenzimplementierungen

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

Das Projekt https://dsr.gematik.solutions demonstriert eine Attestation mobiler Anwendungen auf gängigen mobilen Betriebsystemplattformen.

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

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

7 Anhang A – Verzeichnisse

7.1 Abkürzungen

Tabelle 14: Im Dokument verwendete Abkürzungen

Kürzel	Erläuterung
API	Application Programming Interface
BDE	Betriebsdatenerfassung
CD	Continuous Delivery
CTS	Compatibility Test Suite
DSR	Device Security Rating
eGK	elektronische Gesundheitskarte
ePA	elektronische Patientenakte
FBE	File Based Encryption
FDE	Full Disk Encryption
FIPS	Federal Information Processing Standards
GesundheitsID	Digitale Identität
IDP	Identity Provider
IDS	Intrusion Detection System
ISMS	Informationssicherheitsmanagementsystem
JWT	JSON Web Token
k8s	Kubernetes
mTLS	Mutual Transport Layer Security
OCSP	Online Certificate Status Protocol
PAP	Policy Administration Point

PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
PU	Produktivumgebung
SIEM	Security Information and Event Management
SPIFFE	Secure Production Identity Framework for Everyone
SPIRE	SPIFFE Runtime Environment
TI	Telematikinfrastruktur
TI-ITSM	IT-Service-Management der TI
TPM	Trusted Platform Module
VAU	Vertrauenswürdige Ausführungsumgebung

7.2 Glossar

Tabelle 15: Glossar der explizit im Dokument verwendeten Begriffe

Begriff	Erläuterung
Authorization Server	Ein Server, der Zugriffstoken ausgibt, nachdem er die Identität eines Benutzers authentifiziert und die Berechtigungen überprüft hat. In OAuth 2.0-basierten Systemen ist der Authorization Server eine zentrale Komponente zur Verwaltung von Zugriffsrechten.
Client Registry	Eine Datenbank oder ein Verzeichnis, das Informationen über registrierte Clients enthält, die auf eine API oder einen Dienst zugreifen dürfen. Es verwaltet Details wie Client-IDs, Geheimnisse und andere relevante Informationen, die für die Authentifizierung und Autorisierung erforderlich sind.
Cluster Management Service	Ein Dienst zur Verwaltung und Orchestrierung von Cluster-Ressourcen, insbesondere in containerisierten Umgebungen wie Kubernetes. Er ermöglicht die Verwaltung von Services, Pods und anderen Ressourcen innerhalb eines Clusters, um die Verfügbarkeit, Skalierbarkeit und Sicherheit der Anwendungen zu gewährleisten.
Demonstrating Proof of Possession (DPoP)	Ein DPoP Token ist ein Sicherheitsmechanismus im OAuth 2.0-Protokoll, der den Besitz eines kryptografischen Schlüssels nachweist. Es stellt sicher, dass der Client, der ein Access Token verwendet, auch den zugehörigen privaten Schlüssel besitzt, um Missbrauch durch Dritte zu

	verhindern.
Identity and Access Management (IAM)	Prozesse und Technologien zur Verwaltung digitaler Identitäten und deren Zugriff auf Unternehmensressourcen.
Open Policy Agent (OPA)	Eine Open-Source-Policy-Engine, die es ermöglicht, Richtlinien als Code zu definieren und durchzusetzen, um Entscheidungslogik zentralisiert und flexibel in verschiedensten Software-Systemen und Anwendungen zu implementieren.
Policy Administration Point (PAP)	Eine Komponente, die Sicherheitsrichtlinien erstellt, verwaltet und verteilt. Der PAP definiert und verwaltet die Richtlinien, die von der PDP Policy Engine bei der Entscheidungsfindung verwendet werden.
Policy Decision Point (PDP)	Der PDP trifft die Entscheidung, ob ein Access Token ausgestellt werden darf, basierend auf den definierten Richtlinien und Informationen über den Anfragenden und den Client.
Policy Enforcement Point (PEP)	Ein Punkt in einem Netzwerk, an dem Sicherheitsrichtlinien durchgesetzt werden. Der PEP überwacht und kontrolliert den Zugriff auf Ressourcen basierend auf den Entscheidungen, die vm Policy Decision Point (PDP) getroffen werden.
Policy Information Point (PIP)	Eine Quelle von Attributen oder Kontextinformationen, die für die Entscheidungsfindung des Policy Decision Point (PDP) erforderlich sind. Der PIP stellt die notwendigen Daten zur Verfügung, um Zugriffsanfragen entsprechend den festgelegten Richtlinien zu bewerten.
Security Information and Event Management (SIEM)	Technologien und Prozesse zur Sammlung, Analyse und Korrelation von Sicherheitsdaten aus verschiedenen Quellen, um Sicherheitsvorfälle zu erkennen und darauf zu reagieren.
Telemetrie-Daten	Telemetrie-Daten sind Daten, die von entfernten oder verteilten Systemen, Geräten oder Anwendungen gesammelt und an ein zentrales System zur Überwachung, Analyse und Verwaltung übertragen werden. Im Kontext der IT spielen Telemetrie-Daten eine entscheidende Rolle bei der Überwachung und Sicherung von Netzwerken und Systemen. Merkmale und Arten von Telemetrie-Daten: - System- und Leistungsmetriken: Informationen über die Leistung und den Zustand von Hardware und Software, wie CPU-Auslastung, Speichernutzung, Netzwerkbandbreite und Festplattenkapazität. - Benutzeraktivitätsdaten: Protokolle und Aufzeichnungen über Benutzeraktionen und -verhalten, einschließlich Anmeldungen, Dateizugriffe, Anwendungsnutzung und andere Interaktionen. - Sicherheitsereignisse: Daten über sicherheitsrelevante Vorfälle, wie fehlgeschlagene Anmeldeversuche, erkannte Malware, unerlaubte Zugriffsversuche und andere sicherheitsbezogene Anomalien. - Netzwerkverkehrsdaten: Informationen über den Datenfluss im

	Netzwerk, einschließlich IP-Adressen, Ports, Protokolle, Datenmengen und Verbindungen zwischen verschiedenen Systemen und Diensten. - Konfigurationsdaten: Details zu den aktuellen Einstellungen und Konfigurationen von Systemen und Anwendungen, einschließlich Softwareversionen, installierte Patches und Sicherheitsrichtlinien.
Telemetrie-Daten-Service	Ein Dienst, der Telemetrie-Daten sammelt, verarbeitet und analysiert. Telemetrie-Daten umfassen Informationen über die Nutzung, Leistung und Zustände von Systemen und Anwendungen. Der Dienst hilft dabei, Einblicke in das Verhalten und die Gesundheit der Infrastruktur zu gewinnen, um proaktive Maßnahmen zur Optimierung und Sicherheit zu ergreifen.
Zero Trust (ZT)	Ein Sicherheitskonzept, das davon ausgeht, dass keine Entität (intern oder extern) automatisch vertraut wird. Alle Zugriffsanfragen werden überprüft, unabhängig von ihrem Ursprung.

7.3 Abbildungsverzeichnis

Abbildung 1: NIST Zero Trust-Referenzarchitektur....11
Abbildung 2: Zero Trust-Architektur der TI 2.0....12
Abbildung 3: SM(C)-B Authentisierung mit DPoP....52
Abbildung 4: Beziehungen zwischen Session-, Nutzer- und Client-Daten sowie Token....57

7.4 Tabellenverzeichnis

Tabelle 1: Statische Eigenschaften Clientsysteme auf Hersteller-/Herausgeber-/Anbieterebene....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....20
Tabelle 5 ZT_http_Statuscodes....33
Tabelle 6: PEP http Proxy - Zusätzliche http-Header....40
Tabelle 7: OPA - Konfiguration....41
Tabelle 8: PDP Authorization Server - Plugin-Schnittstelle Application Authorization Backend....48
Tabelle 9 SM(C)-B_Nutzer-Daten....49
Tabelle 10 id_token_Nutzer-Daten....49
Tabelle 11: PDP - Konfigurations-Parameter....58
Tabelle 12: Tab_gemF_Zero-Trust_Schnittstellendefinition_ZT_Cluster....63
Tabelle 13: Tab_gemF_Zero-Trust_Schnittstellendefinition_PIPPAP....63

Tabelle 14: Im Dokument verwendete Abkürzungen....66
Tabelle 15: Glossar der explizit im Dokument verwendeten Begriffe....67
Tabelle 16: Referenzierte Dokumente der gematik....70
Tabelle 17: Weitere Referenzen....71

7.5 Referenzierte Dokumente

7.5.1 Dokumente der gematik

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

Tabelle 16: 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/
[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

7.5.2 Weitere Referenzen

Tabelle 17: Weitere Referenzen

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[access-token.yaml]	Schema access-token.yaml https://raw.githubusercontent.com/gematik/spec-t20r/refs/heads/main/src/schemas/access-token.yaml
[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
[BSI-Prüfvorschrift]	Prüfvorschrift für den Produktgutachter des „ePA-Frontend des Versicherten“ und des „E-Rezept-Frontend des Versicherten. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/DigitaleGesellschaft/Pruefvorschrift_Produktgutachter_ePA-Frontend.html
[CAB-Forum]	Certification Authority Browser Forum (CA/Browser Forum) https://cabforum.org/
[CAPEC OWASP]	CAPEC: OWASP Related Patterns CAPEC - CAPEC-659: OWASP Related Patterns (Version 3.9) (mitre.org)
[client-instance.yaml]	Schema client-instance.yaml https://raw.githubusercontent.com/gematik/spec-t20r/refs/heads/main/src/schemas/client-instance.yaml
[ExpBack]	Exponential Backoff https://en.wikipedia.org/wiki/Exponential_backoff

[GitHub ZT Schemas]	Schemas für zusätzliche http-Header https://github.com/gematik/spec-t20r/tree/main/src/schemas
[ISMS]	Spezifikation Datenschutz- und Sicherheitsanforderungen der TI an Anbieter (Abschnitt 3.3) https://gemspec.gematik.de/docs/gemSpec/gemSpec_DS_Anbieter/latest/#3.3
[OPA Bundle]	Open Policy Agent, Bundles https://www.openpolicyagent.org/docs/latest/management-bundles/
[Open Policy Agent]	Open Policy Agent https://www.openpolicyagent.org/docs/latest/
[OWASP-Top-10-Risiken]	OWASP Top 10 https://owasp.org/www-project-top-ten/
[refresh-token.yaml]	Schema refresh-token.yaml https://raw.githubusercontent.com/gematik/spec-t20r/refs/heads/main/src/schemas/refresh-token.yaml
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels https://datatracker.ietf.org/doc/html/rfc2119
[RFC2986]	PKCS #10: Certification Request Syntax Specification https://datatracker.ietf.org/doc/html/rfc2986
[RFC6066]	Transport Layer Security (TLS) Extensions: Extension Definitions https://datatracker.ietf.org/doc/html/rfc6066
[RFC6749]	The OAuth 2.0 Authorization Framework https://datatracker.ietf.org/doc/html/rfc6749
[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
[session.yaml]	Schema session.yaml https://raw.githubusercontent.com/gematik/spec-t20r/refs/heads/main/src/schemas/session.yaml
[SPIFFE und SPIRE]	Universal identity control plane for distributed systems https://spiffe.io/
[TR-03161]	BSI TR-03161 Anforderungen an Anwendungen im Gesundheitswesen https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03161/tr-03161.html
[TR-03107-1]	BSI TR-03107 Elektronische Identitäten und Vertrauensdienste im E-Government https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03107/TR-03107-1.pdf
[user-info.yaml]	Schema user-info.yaml https://raw.githubusercontent.com/gematik/spec-t20r/refs/heads/main/src/schemas/user-info.yaml
[Verified Boot]	Verifizierter Start https://source.android.com/docs/security/features/verifiedboot?hl=de