

Telematikinfrastruktur 2.0

Feature: Zero Trust

Version:	1.0.01_CC
Revision:	887428925986
Stand:	15.0407.06.2024
Status:	zur Abstimmung freigegeben
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemF_Zero-Trust

Dokumentinformationen

Änderungen zur Vorversion

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

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0_CC	15.04.2024		initiale Erstellung	gematik
1.0.1_CC	07.06.2024		ZT für LE-Zugang hinzugefügt	gematik

Inhaltsverzeichnis

1 Einordnung des Dokuments	8
1.1 Zielsetzung	8
1.2 Zielgruppe	8
1.3 Abgrenzungen	8
1.4 Methodik	9
1.4.1 Anforderungen.....	9
2 Epics und User Stories	10
2.1 Clientregistrierung	10
2.1.1 Wiedererkennung bekannter Clients.....	10
2.1.2 Device Security Rating.....	10
2.2 Policy Enforcement.....	10
2.2.1 Zugriffsschutz.....	11
2.2.2 Forward Proxy	11
2.3 Decide from Policies	11
2.3.1 Maschinenlesbare Zugriffsregeln.....	11
2.3.2 Ein reproduzierbares Ja/Nein/Vielleicht.....	11
2.3.3 Policies nach Betroffenheit.....	11
2.4 Policy Information und Administration	11
2.4.1 Policy Verwaltung.....	11
2.4.2 Monitoring	12
2.5 Client Authorization.....	12
2.5.1 Autorisierung auf Basis von Policy Entscheidungen.....	12
2.5.2 Client Authentication	12
3 Einordnung in die TI 2.0	13
4 Technisches Konzept	17
4.1 Policy Enforcement Point (PEP).....	17
4.2 Policy Decision Point (PDP).....	18
4.3 Trust Client.....	19
4.4 Policy Information und Administration	19
4.4.1 Policy Information Point (PIP)	19
4.4.2 Policy Administration Point (PAP).....	20
4.5 Clientregistrierung	20
4.6 Monitoring	24
4.6.1 Security Information and Event Management (SIEM):	24
4.6.2 Shared Signals	24
4.6.3 Telemetrie, Monitoring und Logging.....	25
4.7 Zusammenspiel mit IdP.....	25
4.8 Fachdienst Backend	26

5 Spezifikation	27
5.1 Übergreifende Anforderungen für Datenschutz und Sicherheit	27
5.1.1 Sicherheits- und Datenschutzanforderungen an Logging und Monitoring	28
5.1.2 Sicherheits- und Datenschutz-Anforderungen an das Security Monitoring	29
5.1.3 Sicherheits- und Datenschutz-Anforderungen an die Verarbeitung von personenbezogenen medizinische Daten	31
5.1.4 Sicherheits- und Datenschutz-Anforderungen an dem Trust Client	32
5.2 Anforderungen an Clientsysteme	33
5.2.1 Hersteller und Herausgeber	33
5.2.2 Verbindungsaufbau	33
5.2.3 Clientregistrierung	34
5.2.4 Nutzerauthentifizierung	35
5.2.5 Session Management	36
5.3 Anforderungen an Policy Enforcement Points	37
5.3.1 PEP-Dynamic Client Registration	37
5.3.1.1 Sicherheits- und Datenschutz-Anforderungen an dem PEP-Client Registration	40
5.3.2 PEP-Relying Party	40
5.3.3 PEP-Authorization Server	41
5.3.4 PEP-Upstream HTTP Proxy	42
5.3.5 Sicherheits- und Datenschutz-Anforderungen an dem PEP	49
5.4 Anforderungen an den Policy Decision Point	50
5.5 Anforderungen an den PIP und PAP Dienst	53
5.6 Anforderungen an den Betrieb der Zero-Trust Komponenten	55
5.6.1 Anforderungen für nahtlose Aktualisierungen	55
5.6.2 Anforderungen für Steuerung durch Feature-Flags	56
5.6.3 Anforderungen zur Überwachung des Betriebsstatus	56
5.7 Anforderungen für die Testbarkeit	57
6 Dokumentenhaushalt	62
6.1 Neue Dokumente	62
6.2 Übersicht betroffener Dokumente	62
6.3 Übersicht Produkt- und Anbietertypen	62
7 Beispiele und Referenzimplementierungen	63
8 Anhang A – Verzeichnisse	64
8.1 Abkürzungen	64
8.2 Abbildungsverzeichnis	64
8.3 Tabellenverzeichnis	65
8.4 Referenzierte Dokumente	65
8.4.1 Dokumente der gematik	65
8.4.2 Weitere Referenzen	67
1 Einordnung des Dokuments	8
1.1 Zielsetzung	8

- 1.2 Zielgruppe 8**
- 1.3 Abgrenzungen 8**
- 1.4 Methodik 9**
 - 1.4.1 Anforderungen..... 9
- 2 Features und Epics 10**
 - 2.1 Clientregistrierung 10**
 - 2.1.1 Wiedererkennung bekannter Clients..... 10
 - 2.1.2 Device Security Rating..... 10
 - 2.2 Policy Enforcement..... 10**
 - 2.2.1 Zugriffsschutz..... 11
 - 2.2.2 http Proxy..... 11
 - 2.3 Decide from Policies 11**
 - 2.3.1 Maschinenlesbare Zugriffsregeln..... 11
 - 2.3.2 Ein reproduzierbares Ja/Nein/Vielleicht..... 11
 - 2.3.3 Policies nach Betroffenheit..... 11
 - 2.4 Policy Information und Administration 11**
 - 2.4.1 Policy Verwaltung..... 11
 - 2.4.2 Monitoring 12
 - 2.5 Client Authorization..... 12**
 - 2.5.1 Autorisierung auf Basis von Policy Entscheidungen 12
 - 2.5.2 Client Authentication 12
- 3 Einordnung in die TI 2.0 13**
- 4 Technisches Konzept 17**
 - 4.1 Zero Trust Cluster..... 17**
 - 4.2 Policy Enforcement Point (PEP)..... 17**
 - 4.3 Policy Decision Point (PDP) 18**
 - 4.4 Trust Client..... 19**
 - 4.5 Policy Information und -Administration 19**
 - 4.5.1 Policy Information Point (PIP) 19
 - 4.5.2 Policy Administration Point (PAP)..... 20
 - 4.6 Clientregistrierung 20**
 - 4.7 Monitoring 24**
 - 4.7.1 Security Information and Event Management (SIEM): 24
 - 4.7.2 Shared Signals 24
 - 4.7.3 Telemetrie, Monitoring und Logging 25
 - 4.8 Zusammenspiel mit IdP..... 25**
 - 4.9 Fachdienst-Backend 26**
- 5 Spezifikation 27**
 - 5.1 Übergreifende Anforderungen für Datenschutz und Sicherheit..... 27**
 - 5.1.1 Sicherheits- und Datenschutzanforderungen an Logging und Monitoring..... 28
 - 5.1.2 Sicherheits- und Datenschutz-Anforderungen an das Security Monitoring 29

5.1.3 Sicherheits- und Datenschutz-Anforderungen an die Verarbeitung von Daten mit dem Schutzbedarf "sehr hoch"	31
5.1.4 Sicherheits- und Datenschutz Anforderungen an dem Trust Client	32
5.2 Anforderungen an Clientsysteme.....	33
5.2.1 Hersteller und Herausgeber	33
5.2.2 Verbindungsaufbau	33
5.2.3 Clientregistrierung	34
5.2.4 Nutzerauthentifizierung.....	35
5.2.5 Session Management	36
5.3 Zero Trust Cluster.....	36
5.4 Anforderungen an Policy Enforcement Points	37
5.4.1 PEP Client Registry	37
5.4.1.1 Sicherheits- und Datenschutz-Anforderungen an dem PEP Client Registration.....	40
5.4.2 PEP Relying Party.....	40
5.4.3 PEP Authorization Server.....	41
5.4.3.1 Service Discovery.....	43
5.4.3.2 Ablauf der SM-B Authentifizierung mit DPoP.....	44
5.4.4 PEP http Proxy.....	48
5.4.5 Sicherheits- und Datenschutz-Anforderungen an dem PEP.....	49
5.4.6 Konfiguration	50
5.5 Anforderungen an den Policy Decision Point	50
5.6 Anforderungen an den PIP und PAP Service	53
5.7 Anforderungen an den Betrieb der Zero Trust Komponenten.....	55
5.7.1 Anforderungen für nahtlose Aktualisierungen	55
5.7.2 Anforderungen für Steuerung durch Feature-Flags	56
5.7.3 Anforderungen zur Überwachung des Betriebsstatus	56
5.7.4 Betriebliche Schnittstellendefinition der Zero Trust-Komponenten	57
5.8 Anforderungen an den Test der Zero-Trust Komponenten	59
5.8.1 Testartefakte.....	59
5.8.2 Testtreiberschnittstelle und Testunterstützung	59
5.8.3 Bereitstellung der Testkomponenten und Testartefakte.....	60
5.8.4 Testumgebungen und Quality Gates.....	61
6 Dokumentenhaushalt.....	62
6.1 Neue Dokumente	62
6.2 Übersicht betroffener Dokumente	62
6.3 Übersicht Produkt- und Anbietertypen	62
7 Beispiele und Referenzimplementierungen	63
8 Anhang A – Verzeichnisse	64
8.1 Abkürzungen	64
8.2 Abbildungsverzeichnis.....	64
8.3 Tabellenverzeichnis	65
8.4 Referenzierte Dokumente.....	65
8.4.1 Dokumente der gematik.....	65

| 8.4.2 Weitere Referenzen 67

1 Einordnung des Dokuments

Dieses Dokument stellt eine übergreifende Spezifikation dar, ohne einen ersten konkreten Bezug zu einem Produkttypen oder zu Schnittstellen 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 ~~sind eine Blaupause für die dienen der Umsetzung der Paradigmen des Zero Trust in einer der "Telematikinfrastruktur 2.0". Zero Trust stellt keinen Produkttyp innerhalb der Telematikinfrastruktur (TI) dar.~~

~~Sämtliche Ausführungen sind zurückzuführen auf das Konzept [gemKPT_ZeroTrust] über die Einführung von Zero Trust zur Modernisierung der TI und konkretisieren die dortigen Ansätze zur Umsetzung in einer Fachanwendung, und revidieren an einigen Stellen vorzeitige Vorfestlegungen bzw. wählen zwischen dort vorgeschlagenen Varianten einer möglichen Zero Trust Umsetzung.~~

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, Backendservices, Produkttypen, Komponenten und Dienste, die sich untereinander über das Internet vernetzen, im Gegensatz zur bestehenden TI als geschlossenes VPN. Dieses Pattern wird bisweilen auch als TI 2.0 bezeichnet.

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 ~~und User Stories~~

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 Clientsystemen 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, soll 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 ~~so~~**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 (DSR - Device Security Rating), soweit es die gegebenen Plattformmechanismen erlauben ~~und die Usability der im Fokus stehenden Anwendung nicht karikiert.~~

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 Forwardhttp Proxy

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

2.3 Decide from Policies

Die Menge an Regeln für die Gewähr eines Zugriffs auf Daten oder Schnittstellen ~~ist groß~~ ~~und~~ 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.

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

Regeln können sich ändern und Regeln beeinflussen Regeln.

2.4.1 Policy Verwaltung

Eine Policyentscheidung 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 Policyentscheidungen

zu Informationen über das Gesamtsystem, die als Eingangsdaten für nachfolgende Policyentscheidungen 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 Telemetriedaten wird die Durchsetzung von Policies sowie die Auswirkung möglicher Policyänderungen transparent.

2.5 Client Authorization

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

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 ~~bzw. Geräte~~ werden anhand sicherer Merkmale authentifiziert, die Identifikation ist nachrangig bzw. in nachgelagerten fachlichen Anwendungsfällen bzw. in fachlichen Zugriffsregeln relevant.

Kann ein Mensch oder Client ~~bzw. Gerät~~ 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 mit ihrem geschlossenen Netz eine reale Infrastruktur. Das Konzept einer TI 2.0 löst sich von einer real existierenden, dedizierten Infrastruktur und setzt auf die Vernetzung über das Internet.

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

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

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

Figure 4-1 General ZTA Reference Architecture

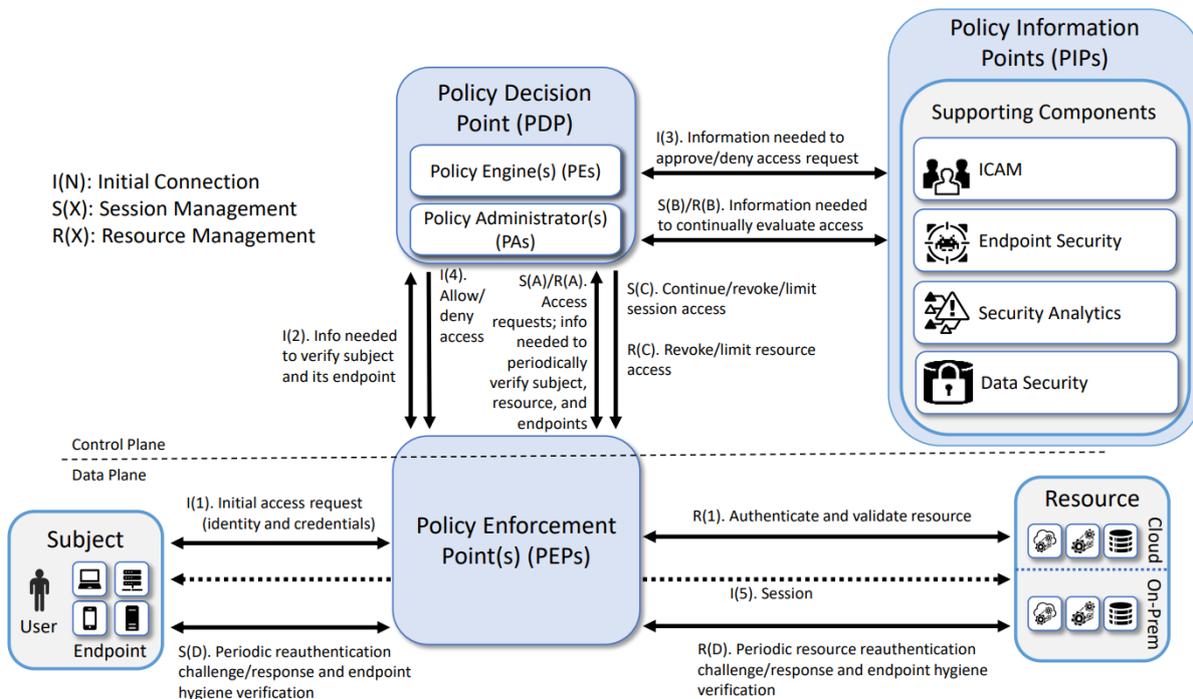


Abbildung 1: NIST Zero Trust Referenzarchitektur

Im Architekturkonzept der TI 1.0 werden konkrete Umgebungsannahmen zu Consumer Zonen, Secure Consumer Zonen, Plattformzonen, Personal Zonen usw. getroffen, in

denen kein (Personal Zone) bzw. ein gewisses Sicherheitsniveau (überall sonst) axiomatisch angenommen wird. Das Zero Trust Konzept löst sich von der Aufteilung in verschiedene Zonen, insbesondere, da weniger (teilweise gar keine mehr) TI-Plattform-Produkttypen zwischen den Datenaustauschen unter Clients mit Backendservices involviert werden. Im Folgenden ist eine Produkttypzerlegung für die Umsetzung der NIST-Referenzarchitektur einer generischen ~~Fachwendung~~-~~Fachanwendung~~ dargestellt.

In diesem Pattern greift ein Nutzer über ein ~~Client-System~~~~Client~~system auf Daten ~~einer Fachanwendung~~ eines TI 2.0 Dienstes zu, ~~die von einem Anbieter des fachanwendungsspezifischen Fachdienstes bereitgestellt wird~~. Das folgende Bild zeigt eine Übersicht der beteiligten Komponenten in der Vernetzung zwischen einem ~~Client~~Clientsystem (links grün) und einem Backendservice (rechts grün: Ressource ~~Backend~~Server).

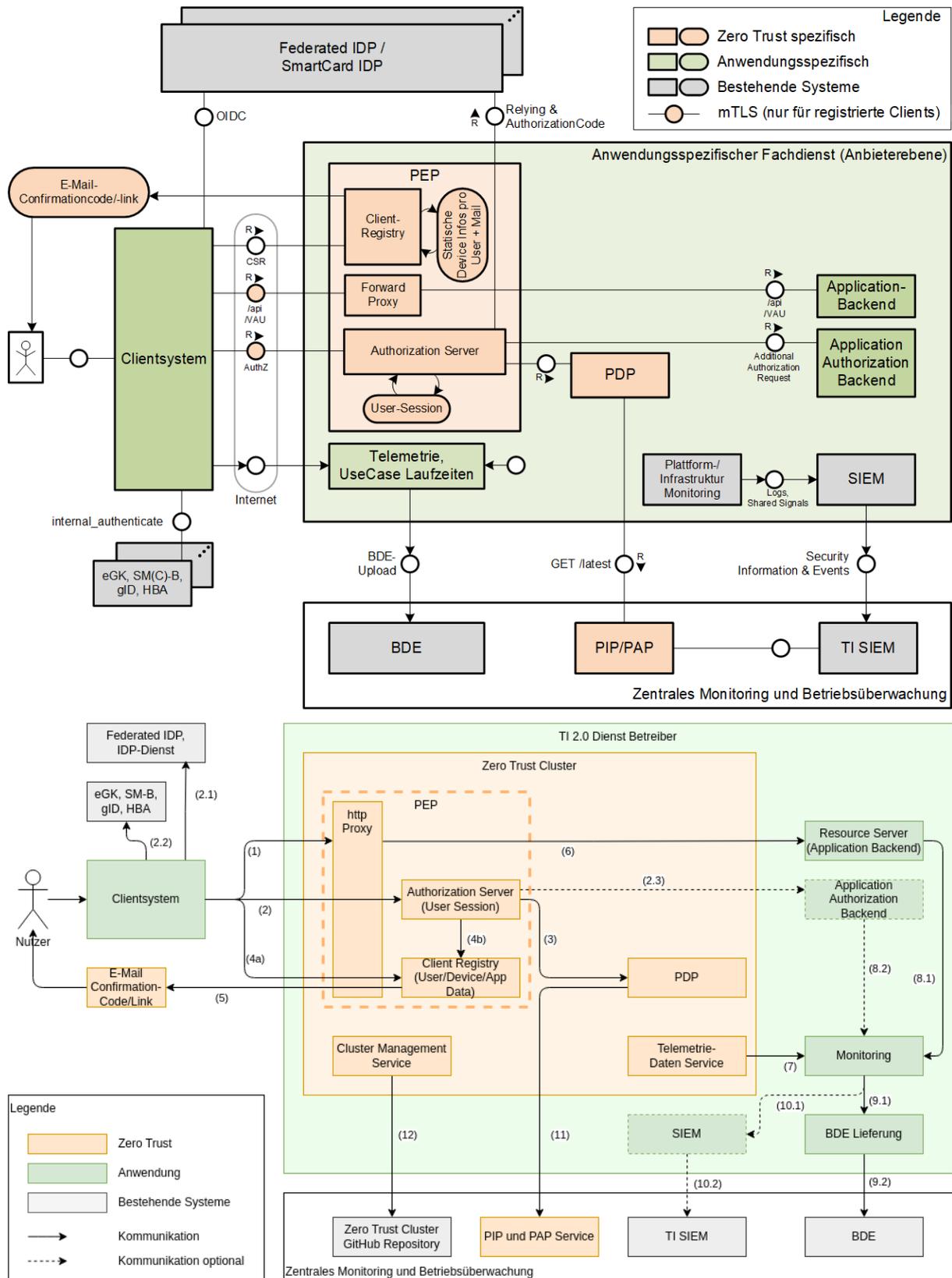


Abbildung 2 : Komponenten zwischen Client und Server
Zero_Trust_Architektur_der_TI_2.0

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

Kurzbeschreibung der Komponenten und Schnittstellen

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

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

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

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

(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 Betreibers 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 Betriebs-Daten-Erfassung gebildet und versendet.

(10.1) und (10.2): Wenn der Betreiber 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 GitHub Repository gespeicherte Konfiguration ausgeführt wird.

4 Technisches Konzept

Im Kapitel zuvor wurden zwei Abbildungen vorgestellt, welche technischen Zero Trust-Komponenten (orange) an der Umsetzung fachlicher Anwendungsfälle von Clients, Komponenten und Backendservices von Fachanwendungen (grün) beteiligt sind. Im Folgenden werden diese technischen Komponenten genauer beschrieben und eingeordnet, welche Rolle sie in einer Architektur nach dem Zero Trust-Paradigma einnehmen. ~~Dieser Komponentenschnitt ist nicht notwendigerweise ein Schnitt in konkrete Produkttypen einer Fachanwendung. Je nach gegebenem Rahmen setzen Produkttypen einer Fachanwendung diese Komponenten um oder nutzen Schnittstellen zu diesen Komponenten in anderen (bestehenden) Produkttypen der TI 2.0.~~

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

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

4.1 Zero Trust Cluster

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

~~4.1~~ 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 ~~Forwardproxy~~ http Proxy, der den Datenverkehr zwischen Clientanwendungen und den zu schützenden Ressourcen kontrolliert. Dadurch kann der PEP den gesamten Datenverkehr überwachen und filtern, um sicherzustellen, dass er den festgelegten Sicherheitsrichtlinien entspricht.
- Der PEP ist als vertrauenswürdige Relying Party im föderierten Identitätsmanagement registriert. Dadurch kann der PEP Identitätsinformationen von Benutzern sicher und vertrauenswürdig beziehen und bei Bedarf eine (erneute) Nutzer-Authentifizierung an die IDPs delegieren. Dadurch stellt der PEP

sicher, dass nur authentifizierte Benutzer Zugriff auf die geschützten Ressourcen erhalten.

- Der PEP fungiert als OAuth2 ~~Autorisierungsdienst~~ [Authorization Server](#) und verwaltet die Autorisierung von Benutzeranfragen auf geschützte Ressourcen. Zudem überwacht der PEP die Benutzersessions, um sicherzustellen, dass sie gültig sind und den Sicherheitsrichtlinien entsprechen.
- Der PEP ermöglicht die dynamische Registrierung von Clients ~~bzw. Geräten~~, die auf geschützte Ressourcen zugreifen möchten. Dies umfasst auch die Offband-Bestätigung, bei der zusätzliche Sicherheitsmechanismen (Verifikation via E-Mail oder SMS) verwendet werden, um die Identität und Integrität (plattformabhängig) der registrierten Clients zu überprüfen.

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

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

4.24.3 Policy Decision Point (PDP)

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

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

~~Insgesamt fungiert der PDP als zentraler Mechanismus zur Durchsetzung von Sicherheitsrichtlinien im Rahmen des Zero Trust Modells.~~ Durch die Analyse von Policies und die Bewertung von Zugriffsanfragen auf Plattformebene trägt der PDP dazu bei, sicherzustellen, dass nur autorisierte Benutzer und Geräte Zugriff auf geschützte Ressourcen erhalten, ~~unabhängig von ihrem Standort oder anderen Kontextfaktoren.~~

~~In der TI 2.0 ist der PDP entweder Teil der Betriebsumgebung eines Fachdienstes einer Fachanwendung oder wird durch einen zentralen Fachdienst realisiert, der die (erste) Autorisierungsentscheidung für Fachdienste weiterer Fachanwendungen übernimmt – bspw. als Nachweis einer Proof-of-Patient-Presence.~~

~~4.34.4~~ Trust Client

Im Kontext von Zero Trust stellt der "Trust Client" eine logische Komponente innerhalb einer Clientanwendung (Primärsystem (PS), Frontend des Versicherten (FdV) etc.) dar, die im Rahmen des Zero Trust-Paradigmas als vertrauenswürdig eingestuft wird. Dies steht im Gegensatz zu der traditionellen Annahme, dass ein, eine Schnittstelle aufrufendes, Clientsystem automatisch als vertrauenswürdig betrachtet wird. Das Vertrauen in Clientanwendungen erwächst beispielsweise durch regelmäßige Softwareupdates, authentische und verschlüsselte Kommunikation und die Verwendung von Zertifikaten, die die Einhaltung dieser Maßnahmen beweisen.

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

~~4.44.5~~ 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 der PDP anwendet, um zu entscheiden, ob eine Kommunikationsanfrage zulässig ist.

~~4.4.14.5.1~~ 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.

4.4.24.5.2 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.

4.54.6 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-/Anbierebene

Client Eigenschaft	Beschreibung
Produkt-Id	Eindeutige ID der Client-Software,, vergeben durch gematik über einen ITSM Prozess
Produkt-Name	Produkt-Name, vergeben durch den Hersteller
Hersteller-Id	Kennung des Herstellers aus TI-ITSM
Hersteller-Name	Name des Herstellers aus TI-ITSM
Produkt-Plattform	Zunächst werden zwei Plattformen gesondert behandelt: Android und Apple (iOS, macOS etc). Diese beide Plattformen bieten Mechanismen für die Attestation der Client-Instanzen und der Umgebung, in welcher diese Clients ausgeführt werden. Alle andere Clients werden zunächst als generische Software-Produkte eingestuft.

Client Eigenschaft	Beschreibung
Produkt-Plattform-Id	<p>Plattformspezifisch eindeutige Kennung der Client-Software</p> <p>Android: Package-Name und Signer-Zertifikat Fingerprint Apple: Bundle-ID und Apple-ID Software: Registriert durch gematik , analog zu ClientIds in E-Rezept</p>
Attestation-Methode	<p>Die Methode, nach welcher die Client-Software und die Ablaufumgebung attestiert werden kann.</p> <p>Zunächst werden Android und Apple unterstützt, weil diese Plattformen entsprechende Mechanismen zur Remote-Attestation anbieten.</p> <p>Perspektivisch ist es geplant, weitere Plattformen zur Attestation der Clients einzuführen, z. B. über TPM 2.0 auf Windows und Linux. Nicht attestierbare Software-Clients müssen zunächst einen weiteren Faktor verwenden, z. B. SM-B.</p>

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

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

Tabelle 2 : Eigenschaften Clientsysteme auf Instanzebene (pro Installation

Client-Instanz-Eigenschaften	Beschreibung
Produkt-Version	Aktuelle Version der Client-Software
Client-BesitzerNutzer (Owner)	Informationen über den Client-BesitzerNutzer
Client-Eigenschaften (Posture)	<p>Aktuelle Eigenschaften der Ablaufumgebung des Clients, insbesondere:</p> <ul style="list-style-type: none"> • Betriebssystem • Betriebssystem-Version

Client-Instanz-Eigenschaften	Beschreibung
Client-Attestation	<p>Falls die Plattform die Attestation des Clients ermöglicht, wird hier plattformspezifische Attestation angegeben.</p> <ul style="list-style-type: none">• Android: Key ID Attestation• Apple: DCAppAttest• Software: keine Attestation, nur BesitzerNutzer-Bindung

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

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

Android

Tabelle 3: Verwendete Device Claims für Android-Geräte

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

iOS

Tabelle 4: Verwendete Device Claims für iOS-Geräte

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

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

4-64.7 Monitoring

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

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

~~4-6-1~~4.7.1 Security Information and Event Management (SIEM):

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

~~4-6-2~~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.

4.6.34.7.3 Telemetrie, Monitoring und Logging

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

4.74.8 Zusammenspiel mit IdP

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

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

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

4.84.9 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.

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]). [\leq]

A_25401 - ~~Zero Trust-Komponenten - Darstellung der Voraussetzungen für sicheren Betrieb des Produkts im Handbuch~~ 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 ~~Handbuch~~Betriebshandbuch leicht ersichtlich darstellen, welche Voraussetzungen vom Betreiber und der Betriebsumgebung erfüllt werden müssen, damit ein sicherer Betrieb des Produktes gewährleistet werden kann. [\leq]

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. [\leq]

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. [\leq]

A_25404 - Zero Trust-Komponenten - Angriffe erkennen

Alle Zero Trust-Komponenten MÜSSEN Maßnahmen zur Erkennung, Kategorisierung und Protokollierung bzw. Meldung von Angriffen umsetzen. [\leq]

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. [\leq]

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. [\leq]

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 unbefugter Profilbildung

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

A_25409 - Zero Trust-Komponenten - Privacy by Design

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

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

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

A_25411 - Zero Trust-Komponenten - Verbot vom dynamischen Inhalt

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

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

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

A_25413 - Zero Trust-Komponenten - Ordnungsgemäße IT-Administration

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

A_25718 - Zero Trust-Komponenten - Bereitstellung Security-KPIs

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

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

5.1.1 Sicherheits- und Datenschutzerfordernungen an Logging und Monitoring

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

A_25744 - Zero Trust-Komponenten - Datenschutzkonformes Logging und Monitoring

Die Zero Trust-Komponenten MÜSSEN die für den Betrieb des Zero Trust erforderlichen Logging- und Monitoring-Informationen in solcher Art und Weise erheben und verarbeiten, dass mit technischen Mitteln ausgeschlossen ist, dass dem Betreiber der

Zero Trust-Komponenten vertrauliche oder zur unautorisierten Profilbildung geeignete Daten zur Kenntnis gelangen. [≤]

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

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

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

A_25747 - Zero Trust-Komponenten - Löschfristen Protokolle

Der Betreiber 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 Betreiber des ~~Monitoring-Systems~~ TI 2.0 Dienstes MUSS sicherstellen, dass das Monitoring System die folgenden ~~Arten von Anomalien erkannt werden können~~ 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 Betreiber des Security Monitorings sicherstellen, dass das Security Monitoring solche Missbrauchssignale von dem Fachdienst empfangen und verarbeitet werden kann. [\leq]

A_25420 - ~~Security Monitoring – Anomalien signalisieren~~ Security Monitoring - Kommunikationsmerkmale signalisieren

Der Betreiber des Monitoring Systems MUSS sicherstellen, dass bei ~~der~~Erkennung ~~von den folgenden Anomalien~~folgender Kommunikationsmerkmale die erforderlichen Informationen automatisiert ~~ein Abbruchsignal~~an ~~dem~~den PEP ~~und PDP~~gesendet wird: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)

[\leq]

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

A_25484 - Security Monitoring - Security KPIs

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

- Anzahl versuchter Zugriffe von nicht registrierten Geräten (hier muss die KPIs zwischen Fachdienst APIs und Clientregistrierung APIs unterscheiden)
- Anzahl von Netzwerk-Protokoll-Anomalien
- Anzahl von Zugriffen von Botnetzen
- Anzahl von Zugriffen aus jedem Land gezählt plus weitere Zugriffe, die separat in Versicherte und LE ausgewiesen werden
- Anzahl fehlerhafter Gerätefreischaltungen plus weitere breakdown in Versicherte und LE

- Anzahl von Impossible travel Zugriffen (inkl. Land- und Ortsdaten) plus weitere breakdown in Versicherte und LE
- Anzahl von Zugriffen über TOR Netzwerke plus weitere breakdown in Versicherte und LE
- Anzahl von Zugriffen über Proxies plus weitere breakdown in Versicherte und LE
- Anzahl von Zugriffen über VPNs plus weitere breakdown in Versicherte und LE
- Traffic Volumes plus weitere breakdown in Versicherte und LE
- Anzahl erkannte Angriffe in Kategorie (siehe [A_25404](#)) plus weitere breakdown in Versicherte und LE
- Anzahl fehlerhafte Authorization Codes vom IDP.

[<=]

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

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

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

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

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

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

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

[<=]

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

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

~~A_25608 - PEP und PDP—Verarbeitung von personenbezogenen medizinische Daten in einer VAUPEP und PDP - Verarbeitung von Daten mit Schutzbedarf "sehr hoch"~~

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

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

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

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

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

A_25765 - Zero Trust-Komponenten - Einsatz zertifizierter HSM

~~Alle Zero Trust Komponenten MÜSSEN~~Falls der ZT Cluster Daten mit dem Schutzbedarf „sehr hoch“ verarbeitet und der Betreiber keine Kenntnis über die in dem ZT Cluster verarbeiteten Daten erlangen darf, MUSS der Betreiber beim Einsatz eines HSMs sicherstellen, dass dessen Eignung durch eine erfolgreiche Evaluierung nachgewiesen wurde. Als Evaluierungsschemata kommen dabei Common Criteria, ITSEC 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. [\leq]

A_26065 - Nur zugelassene Images in Produktion

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

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

5.1.4 Sicherheits- und Datenschutz Anforderungen an dem Trust Client

A_25802 - ~~Trust Client – Einhaltung der BSI-Prüfvorschrift~~ Trust Client - Einhaltung der BSI-Prüfvorschrift

Der Trust Client MUSS die Prüfaspekte aus ~~der BSI-Prüfvorschrift~~ [~~BSI-Prüfvorschrift~~BSI [TR-03161]] erfüllen, sofern sie für das Trust Client anwendbar sind. [\leq]

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-Prüfvorschrift~~ [TR-03161] 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-Prüfvorschrift~~ [TR-03161] aus Sicht des Gutachters erfüllt sind.

5.2 Anforderungen an Clientsysteme

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.

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 [für wenigstens 5 Jahre](#) 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 alternative Plattformattestierung verwenden](#), um Nachweise über die Geräteintegrität einer jeden Clientsysteminstallation [über die Google Play Integrity API](#) beziehen zu können. [<=]

5.2.2 Verbindungsaufbau

A_25338 - Clientsystem - User-Agent

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

- <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 \[0-9a-zA-Z\-\.\]](#)

des Clientsystems befüllen. [<=]

A_25339 - Clientsystem - Exponential Backoff

Das Clientsystem 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. [<=]

A_25340 - Clientsystem - Zertifikatsprüfung

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

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

A_25341 - Clientsystem - TLS- oder JWT-Clientauthentisierung

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

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

als legitimer Client authentisieren. Verfügt das Clientsystem über kein gültiges Clientauthentisierungsmittel, MUSS es den Bezug über die Clientregistrierung starten.[\leq]

5.2.3 Clientregistrierung

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

A_25432 - Clientsystem - Ablauf Clientregistrierung

Das Clientsystem MUSS, sofern es an Schnittstellen der Telematikinfrastruktur wegen einer ungültigen/fehlenden [Gerätregistrierung](#) Geräte/App-Registrierung abgewiesen wird, eine [Clientregistrierung](#) Registrierung am Service der [Dynamic-Client Registration](#) 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-Password über einen zweiten Kommunikationsweg (z. B. E-Mailbestätigung) startet

[\leq]

A_25766 - Clientsystem - Client Credentials in TI Qualität

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

A_25769 - Clientsystem - Client Credentials sicher generieren und schützen

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

A_25770 - Clientsystem - Client Credentials Rotation

Das Clientsystem 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, ~~SMC-B~~.

A_25767 - Clientsystem - Clientkey in JWT oder Zertifikat

Das Clientsystem MUSS wahlweise Private Key JWT (~~[RFC7521]~~ und [RFC7523]) sowie DPoP [RFC9449] oder mutual TLS (~~[RFC8705]~~) zur Authentifizierung unterstützen. [<=]

A_25434 - Clientsystem - Clientregistrierung mit bestätigten Umgebungseigenschaften Android

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

A_25768 - Clientsystem - Clientregistrierung mit bestätigten Umgebungseigenschaften Apple

Das Clientsystem für eine Apple-basierte Betriebsumgebung (iOS, macOS) MUSS die Client-Credentials, App Integrität und Authentizität über DCAppAttest gegenüber PEP Client Registrierung bestätigen. Eigenschaften der Ablaufumgebung Laufzeitumgebung MÜSSEN durch das Clientsystem über einen geprüften Prozess bestätigt werden. [<=]

A_25758 - Clientsystem - Erfassung Kontaktinformation für Offband-Verifikation

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

A_25732 - Clientsystem - Unterstützung des Nutzers bei der Registrierung

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

A_25733 - Clientsystem - Clientverwaltung und manuelle De-Registrierung

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

A_25734 - Clientsystem - Zugriffsprotokoll Clientregistrierung

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

A_25735 - Clientsystem - Aktivierung Push-Benachrichtigung

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

5.2.4 Nutzerauthentifizierung

A_25761 - Clientsystem - Nutzerauthentifizierung mittels etablierter Standards

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

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

A_25762 - Clientsystem - Nutzerauthentifizierung - Unterstützung etablierter Identitäten und Dienste

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

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

[<=]

5.2.5 Session Management

A_25781 - Clientsysteme - OAuth2 Autorisierung

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

[<=]

A_25782 - Clientsystem - OAuth2 Session Management

Das Clientsystem MUSS

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

[<=]

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

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

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

umsetzen.[<=]

5.3 Zero Trust Cluster

Die Software des Zero Trust Clusters wird im Auftrag der gematik entwickelt und als signierte Docker Container in einer gematik Container Registry sowie mit Kubernetes Manifest Dateien in einem gematik GitHub Repository bereitgestellt, sodass die Betreiber

von TI 2.0 Diensten ihren spezifischen ZT Cluster darauf aufbauend als Kubernetes Cluster konfigurieren können. Die ZT Cluster-Konfiguration muss in ein GitHub Repository der gematik als git submodule verlinkt werden. Der Cluster Management Service des Zero Trust Clusters setzt durch, dass nur die im gematik GitHub Repository verlinkte ZT Cluster Konfiguration ausgeführt werden kann. Dadurch ist es möglich, dass der Betreiber seine Cluster-Konfiguration selbständig anpassen und die gematik den korrekten Einsatz des ZT Clusters prüfen kann.

A_26106 - ZT Cluster, Verwendung der gematik Docker Container

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

A_26105 - ZT Cluster, Durchsetzung der Konfiguration

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

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

5.3.5.4 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. Gemäß oben vorgeschlagener Architekturzerlegung, empfiehlt es sich, den PEP in mehreren Teilkomponenten zu realisieren.

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

5.3.15.4.1 PEP ~~Dynamic~~ Client RegistrationRegistry

A_25644 - PEP Client Registration - Mobile Attestation

Die Komponente ~~Dynamic~~ Client ~~RegistrationRegistry~~ MUSS die Clients/Apps bei der Registrierung über folgende Mechanismen attestieren:

- Android Key ID Attestation
- Apple DCAAppAttest.
- ~~[<=]~~ SM-B signiertes Client Assertion JWT

[<=]

A_25645 - PEP Client Registration - User Authentication in Attestation PEP Client Registration - Attestation mittels TI-Smartcard

Die Komponente ~~Dynamic-Client Registration~~Registry MUSS die Registrierung über eine TI-Smartcard durchführen (eGK, SMC-B, HBA), falls die Ausführungsumgebung des Clients keine plattformseitigen ~~Attestationsmechanismen~~Attestation-Mechanismen anbietet. Die Verwendung des zentralen IDP-Dienstes ist für die Nutzerauthentifizierung zulässig. [~~=~~]

A_25646 - PEP Client Registration - SubCA Device Certificate

Die Komponente ~~Dynamic-Client Registration~~ MUSS über ein ~~SubCA-Zertifikat~~ verfügen, um ~~TLS-Client Zertifikate für die Client Instanzen ausstellen zu können.~~ [~~=~~]

Hinweis: Die Mechanismen und Prozesse zur Absicherung des Schlüsselmaterials zu diesem SubCA-Zertifikat haben nach dem Stand der Technik zur Herstellung eines angemessenen Vertrauensniveaus zu erfolgen. Detailfestlegungen erfolgen zu einem späteren Zeitpunkt, wenn die funktionalen Lücken zwischen Clientsystem und der Komponente ~~Dynamic-Client Registration~~ geschlossen werden.

A_25647 - PEP Client Registration - Certificate Signing Request

Die Komponente ~~Dynamic-Client Registration~~ MUSS die automatisierte Erneuerung bzw. Ausstellung neuer ~~TLS-Client Zertifikate~~ über einen ~~Certificate Signing Request~~ unterstützen. [~~=~~]

A_25648 - PEP Client Registration - Device Session Credentials

Die Komponente ~~Dynamic-Client Registration~~Registry MUSS die Client Credentials aufbewahren und Verifikationsmechanismen dem PEP ~~Authorisation~~Authorization Server bereitstellen. [~~=~~]

A_25649 - PEP Client Registration - Regelmäßige Wiederholung der Attestation

Die Komponente ~~Dynamic-Client Registration~~Registry MUSS die Client Attestierung regelmäßig gemäß Festlegungen in der Device Policy wiederholen. [~~=~~]

A_25650 - PEP Client Registration - UserID in Attestation PEP Client Registration - TI-Identität in Attestation

Die Komponente ~~Dynamic-Client Registration~~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. [~~=~~]

A_25651 - PEP Client Registration - Offband User Verification PEP Client Registration - Offband Nutzer Verification

Die Komponente ~~Dynamic-Client Registration~~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_25652 - PEP Client Registration - Push Gateway

Die Komponente ~~Dynamic-Client Registration~~Registry 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 - PEP Client Registration - Push Notification

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

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

A_25653 - PEP Client Registration - Umsetzung der Device Policy

Die Komponente ~~Dynamic-Client~~ ~~RegistrationRegistry~~ 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. [\leq]

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

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

A_25654 - PEP Client Registration - Minimum Device Policy

Die Komponente ~~Dynamic-Client~~ ~~RegistrationRegistry~~ MUSS Client Mindestanforderungen vor der Registrierung verifizieren und kann hierfür eine Schnittstelle des PDP verwenden. [\leq]

~~**A_25737 - PEP Client Registration - Push Notification**~~

~~Die Komponente ~~Dynamic-Client~~ ~~Registration~~ 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. [\leq]~~

A_25738 - PEP Client Registration - Telemetrie Clientregistrierung

Die Komponente ~~Dynamic-Client~~ ~~RegistrationRegistry~~ MUSS in den Telemetriedaten zu jeder versuchten Clientregistrierung folgende Parameter ohne einen Nutzerbezug protokollieren:

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

[\leq]

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

Die Komponente ~~Dynamic-Client~~ ~~RegistrationRegistry~~ 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. [\leq]

A_26064 - Access Token bei Monitoring-Signalen sperren

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

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

5.3.1-15.4.1.1 Sicherheits- und Datenschutz-Anforderungen an dem PEP Client Registration

A_25751 - PEP Client Registration - Anwendungsfälle nur vom registrierten Gerät

Nach der erfolgreichen Registrierung von dem ersten Gerät, Clients (Gerät/App Kombination), MUSS die Komponente Dynamic Client RegistrationRegistry sicherstellen, dass die folgenden Anwendungsfälle nur von einem registrierten Gerät durchgeföhrt werden kann:

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

[<=]

A_25748 - PEP Client Registration - Maximale Anzahl von Geräten

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

A_25749 - PEP Client Registration - Nutzer Protokollierung

Die Komponente Dynamic Client RegistrationRegistry MUSS ein Nutzerprotokoll föhren und die folgenden Anwendungsfälle für den Nutzer protokollieren:

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

[<=]

A_25750 - PEP Client Registration - Nutzer über sicherheitsrelevante Ereignisse informieren

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

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

[<=]

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

5.3-25.4.2 PEP Relying Party

A_25655 - PEP - Relying Party

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

A_25656 - PEP - Entity Statement

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

A_25657 - ~~PEP—User Authentication über sektoralen IDP~~ PEP - Authentication über sektoralen IDP

Die Komponente Policy Enforcement Point (bzw. ihre Subkomponente Authorization Server) MUSS die ~~Anwender~~Nutzer über sektorale IDPs authentifizieren können. [\leq]

A_25658 - ~~PEP—User Authentication über SmartCard IDP~~ PEP - Authentication über SmartCard IDP

Die Komponente Policy Enforcement Point (bzw. ihre Subkomponente Authorization Server) MUSS die ~~Anwender~~Nutzer über den zentralen IDP-Dienst (SmartCard-IDP) authentifizieren können. [\leq]

5.3.35.4.3 PEP Authorization Server**A_25760 - PEP Authorization Server - OAuth2 Schnittstellen**

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

Der Authorization Server MUSS am Token Endpunkt REFRESH_TOKEN entsprechend [RFC6749] ausstellen können. [\leq]

A_25659 - PEP Authorization Server - Check Device Registration

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

- Mutual-TLS Client Authentication gemäß [RFC8705]
- JSON Web Token Client Authentication gemäß [RFC7523] mit DPoP gemäß [RFC9449]

und Anfragen, die weder noch eine der genannten Mechanismen verwendet konsequent ablehnen. [\leq]

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

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. [\leq]

A_25661 - PEP 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. [\leq]

A_25662 - PEP 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. [\leq]

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

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

- TLS Client Zertifikat Binding gemäß [RFC8705]
- OAuth 2.0 Demonstrating Proof of Possession (DPoP) gemäß [RFC9449]

identifizierte Client-Instanz binden, in dem im Token-Binding-Claim die Angabe der verwendeten Clientidentifikation als "jkt" oder "x5t#s256" eineindeutig referenziert wird. [<=]

A_25664 - PEP Authorization Server - Token Laufzeit gemäß Policy

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

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

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

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

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

[<=]

~~5.3.4 PEP Upstream HTTP Proxy~~

~~5.4.3.1 Die Komponente Upstream HTTP Proxy (im Bild oben "Forward Proxy") ist die "letzte" vor das Resource Backend geschaltete Zero Trust Komponente und prüft alle Nachweise (Token, Zertifikate etc.) des~~

~~Clients über den gewährten Zugriff. Sind alle NachweiseService Discovery~~

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

A_26037 - PEP Authorization Server, Well-known

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

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

Das Well-known json Dokument MUSS mit dem Schema

<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. [`<=`]

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 - PEP Authorization Server, Well-known Erstellung

Die Komponente http Proxy MUSS für den Authorization Server 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://github.com/gematik/zero-lab",
"ui_locales_supported": [
  "de",
  "en"
],
"code_challenge_methods_supported": [
  "S256"
]
}
```

5.4.3.2 Ablauf der SM-B Authentifizierung mit DPoP

Die SM-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-B signiertem Client Assertion JWT. Replay-Attacken werden durch die Aufnahme einer server-generierten Nonce als JTI Claim in der Client-Assertion verhindert.

A_26091 - ZT Cluster, SM-B Authentifizierung mit DPoP

Der Zero Trust Cluster MUSS den Ablauf gemäß Abbildung *SM-B Authentifizierung mit DPoP* sowie gemäß [RFC7523] und [RFC9449] unterstützen. [≤]

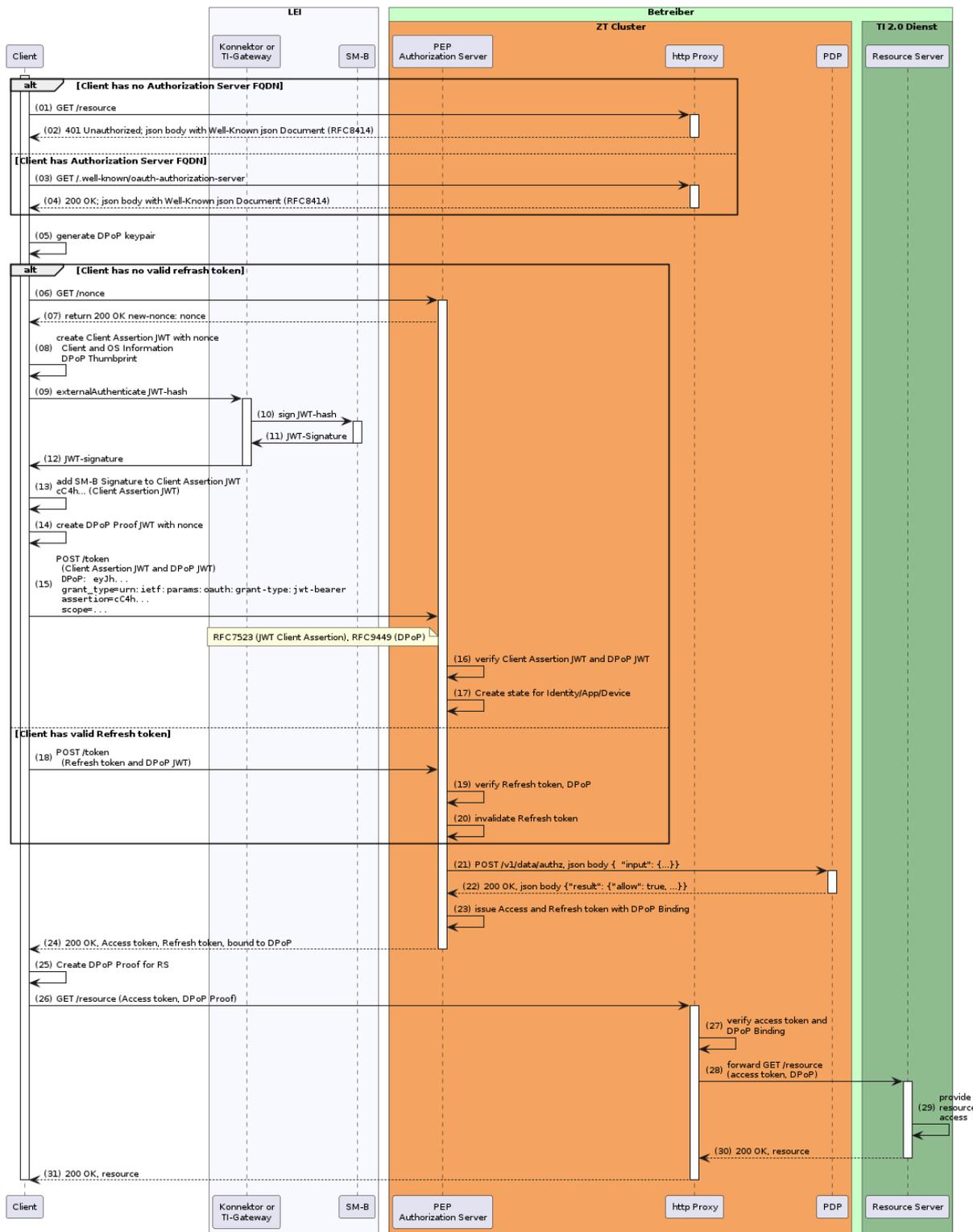


Abbildung 3 SM-B_Authentisierung_mit_DPoP

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

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 auch Informationen über das Gerät, das Betriebssystem und die App (PS) enthalten sind. Das JWT wird mit der SM-B signiert. 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": "18tFrhx-34tV3hRICRDY9zCkDlpBhF42UQUfWVAWBFs",
    "y": "9VE4jfh_Ok_o64zbTT1cuNJajHmt6v9TDVrU0CdvGRDA",
  }
}
```

```

    "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=authorization_code&
code=n0esc3NRze7LTCu7iYzS6a5acc3f0ogp4&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-
type%3Ajwt-bearer&
client_assertion=...JWT Assertion signed by SM-C...

```

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

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

In Schritt (21) und (22) wird durch den PDP 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 den PDP.

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

```

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 (25) und sendet Request, mit Access token und DPoP Proof im Header, an den Resource Server.

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

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

5.4.4 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_25666 - ~~PEP Upstream Proxy~~ — ~~TLS Terminierung~~ PEP http Proxy - TLS Terminierung

Die Komponente ~~Upstream~~ http Proxy MUSS den TLS Kanal terminieren und alle http-Header validieren. [\leq]

A_25667 - ~~PEP Upstream Proxy~~ — ~~Verifikation Access-Token-Binding~~ PEP http Proxy - Verifikation Access-Token Binding

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

- TLS Client Zertifikat Binding gemäß [RFC8705]
- OAuth 2.0 Demonstrating Proof of Possession (DPoP) gemäß [RFC9449],

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

A_25668 - ~~PEP Upstream Proxy~~ — ~~Access-Token-Validierung~~ PEP http Proxy - Access-Token Validierung

Die Komponente ~~Upstream~~ 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 und
- die Angabe *aud* für das Resource Backend korrekt eingetragen

sein. [\leq]

A_25669 - ~~PEP Upstream Proxy~~ — ~~Zusätzliche http-Header~~ PEP http Proxy - ~~Zusätzliche http-Header~~

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

Tabelle 6: PEP ~~Upstream~~ http Proxy - Zusätzliche http-Header

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

Gleichnamige http-Header aus dem ursprünglichen http-Request MÜSSEN entfernt bzw. überschrieben werden. [\leq]

5.3.55.4.5 Sicherheits- und Datenschutz-Anforderungen an dem 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. [\leq]

A_25840 - PEP - Sichere interne Kommunikation

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

A_25448 - PEP - Nutzerzugriff nur von registrierten Geräten

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

A_25449 - ~~PEP~~ Nutzeridentität nur von einem zugelassenem IDP ~~PEP~~ - Nutzeridentität nur von einem zugelassenem IDP

Der PEP MUSS sicherstellen, dass nur Nutzeridentitäten von einem zugelassenen IDP akzeptiert werden. [\leq]

A_25447 - PEP - Kommunikation nur mit authentischen PDP

Der PEP MUSS sicherstellen, dass er mit einem authentischen und korrekt konfigurierten PDP kommuniziert. [\leq]

A_25486 - PEP - Abbruch durch Anomalie Signale

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

5.4.6 Konfiguration

A_26038 - PEP, Konfigurations-Parameter

Der PEP MUSS die folgenden Konfigurations-Parameter unterstützen.

Tabelle 7 PEP_Konfigurations-Parameter

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

[<=]

Hinweis: Weitere Konfigurationsparameter werden zusammen mit dem Zero Trust Hersteller festgelegt und hier ergänzt.

5.4.5 Anforderungen an den Policy Decision Point

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

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

A_25739 - PDP, Open Policy Agent Instanzen

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

Die OPA Instanzen MÜSSEN gemäß Tabelle OPA_Konfiguration konfiguriert werden.

Tabelle 8: OPA_Konfiguration

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

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

[<=]

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 Continuous Delivery Workflow aus GitHub geladen.

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

A_25772 - ~~PDP—persistente-Speicherung-von-decision-logs~~ PDP - persistente Speicherung von decision logs

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

A_25450 - ~~PDP—Policy-nur-von-dem-gematik-PAPPDP - Policy nur vom gematik PIP und PAP Service~~

Der PDP MUSS sicherstellen, dass nur Policies ~~von der~~und Daten vom gematik PIP und PAP Service importiert werden können. [<=]

A_25451 - PDP - Integritätsprüfung der Policies

Der PDP MUSS sicherstellen, dass Policies ~~von dem~~ vom gematik PIP und PAP Service nur nach einer positiven Integritätsprüfung importiert werden können. [\leq]

A_25452 - PDP - Tamper-Proof Protokollierung von Administrationsaktivitäten

Der PDP MUSS ein "Tamper-Proof" Audit-Log von allen administrativen Vorgängen umsetzen. [\leq]

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. [\leq]

A_25775 - PDP - Kontrolle des Audit-Logs

Der Betreiber des ~~PDPs~~ ZT Clusters MUSS das Audit-Log mindestens alle 3 Monate ~~nach~~ ~~maliziösen Administratoraktivitäten~~ durch zwei unabhängige Rollen 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. [\leq ~~einer Zero-Trust-Komponente~~ teilnehmen. [\leq]

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 ~~in der~~ im PDP installiert sind. [\leq]

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. [\leq]

A_25771 - PDP - Automatisierte Prüfung nach Policy-Aktualisierungen

Der PDP muss alle ~~15~~ 5 Minuten prüfen, ob Aktualisierungen der installierten und verwendeten Policy/PIP Daten vorhanden sind. [\leq]

5-55.6 Anforderungen an den PIP und PAP DienstService

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

A_25670 - PIP und PAP - Bereitstellung Download-Endpunkt

Der PIP und PAP DienstService 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>. [\leq]

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 DienstService MUSS sich beim TLS-Verbindungsaufbau am Download-Endpunkt gegenüber Clients mit einem Extended Validation TLS-Zertifikat eines Herausgebers gemäß [CAB-Forum] authentisieren. [\leq]

A_25672 - PIP und PAP - Kompatibilität mit OPA Bundles

Der PIP und PAP DienstService MUSS die Policies und Daten als [OPA Bundle] bereitstellen. [\leq]

A_25680 - PIP und PAP - download path

Der PIP und PAP [DienstService](#) 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. [\leq]

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

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

A_25673 - PIP und PAP - ETags für OPA Bundles

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

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 [DienstService](#) MUSS für alle bereitgestellten OPA Bundles prüfen, ob deren Signatur vorhanden und gültig ist. [\leq]

Hinweis: In dieser Spezifikation wird der Prozess, wie die OPA Bundles sicher in den PIP und PAP [DienstService](#) gelangen, nicht festgelegt.

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

Der PIP und PAP [DienstService](#) MUSS ein "Tamper-Proof" Audit-Log von allen administrativen Vorgängen umsetzen. [\leq]

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

Der PIP und PAP [DienstService](#) MUSS sicherstellen, dass die Löschung eines Auditeintrags den gesetzlichen Vorgaben entspricht und frühestens nach 12 Monaten erfolgt. [\leq]

A_25778 - PAP und PIP - Kontrolle des Audit-Logs

Der Betreiber des PIP und PAP [DienstServices](#) MUSS das Audit-Log mindestens alle 3 Monate ~~nach maliziösen Administratoraktivitäten~~ durch zwei unabhängige Rollen im Vieraugenprinzip kontrollieren. Diese Rollen DÜRFEN NICHT an der Administration des PAPs oder PIPs teilnehmen. *Bei der Kontrolle ist insbesondere auf ungewöhnliche, nicht nachvollziehbare oder maliziöse Administratoraktivitäten zu achten.* [\leq [\leftarrow]]

A_25465 - PAP und PIP - Änderungen nur durch berechtigte Nutzer

Der PIP und PAP [DienstService](#) MUSS sicherstellen, dass nur berechtigte Nutzer Änderungen von Policies oder PIP-Daten durchführen können. [\leq]

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

Der PIP und PAP [DienstService](#) 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. [\leq]

A_25467 - PAP und PIP - Änderungen nur unter 4 Augen

Der PIP und PAP [DienstService](#) MUSS sicherstellen, dass Änderungen in Policies oder PIP-Daten nur im Vieraugenprinzip durchgeführt werden können. [\leq]

A_25791 - PAP und PIP - Zentrale Verwaltung von Feature-Flags

PIP und PAP ~~Dienst~~Service MUSS eine zentrale Schnittstelle oder ein System zur Verwaltung der Feature-Flags bereitstellen. [<=]

5-65.7 Anforderungen an den Betrieb der Zero Trust Komponenten

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

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

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

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

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

5-6-15.7.1 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. [<=]

5.6.25.7.2 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. [<=]

A_25792 - Zero Trust-Komponenten - Protokollierung von Feature-Flag-Änderungen

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

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

A_25793 - Zero Trust-Komponenten - Zugriffskontrolle für Feature-Flag-Verwaltung

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

5.6.35.7.3 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 - Verfügbarkeit der Health Checks für gematik Monitoring

Die Komponente der Zero Trust Architektur MUSS die Schnittstellen zu Health Checks 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. [<=]

~~5.7 Anforderungen für die Testbarkeit~~

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

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

5.7.4 Betriebliche Schnittstellendefinition der Zero Trust-Komponenten

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

Table 1 Tab_gemF_Zero-Trust_Schnittstellendefinition_PEPPDP

<i>Offener Punkt: Details in diesem Kapitel werden im Rahmen der Implementierung zwischen gematik und dem Zero-Trust-Hersteller festgelegt.</i> A-ID	Schnittstellen::Operation / Anwendungsfall	Beschreibung
A01	-	- vorbelegt für Verfügbarkeitsberechnung -
A02	/ .well-known/	Abruf gültiger Autorisierungsserver
A03	GET /nonce/	Nonce abrufen
A04	POST /token <JWT Client Assert>	Autorisierung ohne Refresh Token
A05	POST /token <Refresh Token>	Autorisierung mit Refresh Token

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

Table 2 Tab_gemF_Zero-Trust_Schnittstellendefinition_PIPPAP

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

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

5.8 Anforderungen an den Test der Zero-Trust Komponenten

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

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

5.8.1 Testartefakte

A_26081 - Format von Testszenerarien

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

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

A_26083 - Effiziente Entwicklung von Testartefakten

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

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

A_26089 - Automatisierung von Testartefakten

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

A_26084 - Freie Nutzung und Weiterentwicklung von Testartefakten

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

A_26085 - Support für Testartefakte

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

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

5.8.2 Testtreiberschnittstelle und Testunterstützung

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

A_26086 - Bereitstellen einer Testtreiberschnittstelle

Der Hersteller der Zero Trust Komponenten MUSS in jeder Komponente eine Testtreiberschnittstelle bereitstellen, die den automatisierten Nachweis aller funktionalen

Anforderungen an die jeweilige Komponente in einem Blackbox-Test, d.h. einem Test an den Außenschnittstellen, ermöglicht, die sonst über die normale API der Komponente nicht oder nur mit einem erhöhten Aufwand testbar wäre. [\leq]

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

A_26087 - Keine Testtreiberschnittstelle in produktiv einsetzbaren Komponenten

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

A_26088 - API und Dokumentation der Testtreiberschnittstelle

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

A_26092 - Keine Hardwareabhängigkeiten bei Komponenten in Testumgebungen

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

A_26093 - Kenntnis der privaten Schlüsseln in Testumgebungen

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

A_26094 - Mitwirken bei übergreifenden Testmaßnahmen

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

5.8.3 Bereitstellung der Testkomponenten und Testartefakte

Die hier spezifizierten Komponenten und Dienste haben keinen Selbstzweck, sondern kommen in der Realisierung von Produkttypen einer Fachanwendung zum Einsatz. ~~Sofern sie~~ Sie werden "as Code" bereitgestellt ~~werden, wird angestrebt, sie, und in Build-Pipelines der Komponenten und Produkttypen der Fachanwendung einzubetten, um eingebettet.~~ Weiterhin soll die Testausführung automatisiert werden. Dadurch wird ein automatisches und regelmäßiges Deployment in Testumgebungen ~~zu ermöglichen. Dies unterstützt das Bestreben nach einem~~ im Sinne eines Continuous Testing Testings ermöglicht.

A_26096 - Bereitstellung von Testkomponenten und Testartefakten

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

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

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

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

A_26098 - Konfigurierbarkeit von Testkomponenten und Testartefakten

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

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

5.8.4 Testumgebungen und Quality Gates

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

A_26099 - Bestehen von Quality Gates

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

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

A_26100 - Labeln von Komponenten

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

6 Dokumentenhaushalt

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

6.1 Neue Dokumente

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

6.2 Übersicht betroffener Dokumente

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

6.3 Übersicht Produkt- und Anbietertypen

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

7 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 Betriebssystemplattformen.

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.

8 Anhang A – Verzeichnisse

8.1 Abkürzungen

Tabelle 9: Im Dokument verwendete Abkürzungen

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

8.2 Abbildungsverzeichnis

Abbildung 1: NIST Zero Trust Referenzarchitektur.....	13
Abbildung 2 : Komponenten zwischen Client und Server	15
Abbildung 1: NIST Zero Trust Referenzarchitektur.....	13
Abbildung 2 : Zero_Trust_Architektur_der_TI_2.0.....	15
Abbildung 3 SM-B_Authentisierung_mit_DPoP	45

8.3 Tabellenverzeichnis

Tabelle 1: Statische Eigenschaften Clientsysteme auf Hersteller-/Herausgeber-/Anbierebene	20
Tabelle 2 : Eigenschaften Clientsysteme auf Instanzebene (pro Installation	21
Tabelle 3: Verwendete Device Claims für Android-Geräte	23
Tabelle 4: Verwendete Device Claims für iOS-Geräte	23
Tabelle 5: PEP Authorization Server – Plugin-Schnittstelle Application Authorization Backend.....	42
Tabelle 6: PEP Upstream Proxy – Zusätzliche http-Header	49
Tabelle 7: OPA_Konfiguration.....	51
Tabelle 8: Im Dokument verwendete Abkürzungen	64
Tabelle 9: Referenzierte Dokumente der gematik	66
Tabelle 10: Weitere Referenzen.....	67
Tabelle 1: Statische Eigenschaften Clientsysteme auf Hersteller-/Herausgeber-/Anbierebene	20
Tabelle 2 : Eigenschaften Clientsysteme auf Instanzebene (pro Installation	21
Tabelle 3: Verwendete Device Claims für Android-Geräte	23
Tabelle 4: Verwendete Device Claims für iOS-Geräte	23
Tabelle 5: PEP Authorization Server - Plugin-Schnittstelle Application Authorization Backend.....	42
Tabelle 6: PEP http Proxy - Zusätzliche http-Header	49
Tabelle 7 PEP_Konfigurations-Parameter.....	50
Tabelle 8: OPA_Konfiguration.....	51
Tabelle 9: Im Dokument verwendete Abkürzungen	64
Tabelle 10: Referenzierte Dokumente der gematik	66
Tabelle 11: Weitere Referenzen.....	67

8.4 Referenzierte Dokumente

8.4.1 Dokumente der gematik

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

Tabelle 10: Referenzierte Dokumente der gematik

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

8.4.2 Weitere Referenzen

Tabelle 11: Weitere Referenzen

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[Android Platform Security Model]	The Android Platform Security Model (2023) https://research.google/pubs/the-android-platform-security-model/
[Apple Platform Security Guide]	Einführung in die Sicherheit der Apple-Plattformen https://support.apple.com/de-de/guide/security/seccd5016d31/web
[BSI-Grundschutz]	IT-Grundschutz - Informationssicherheit mit System https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/it-grundschutz_node.html
[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)
[ExpBackoff]	Exponential Backoff https://en.wikipedia.org/wiki/Exponential_backoff
[GPI-API]	Google Play Integrity API https://developer.android.com/google/play/integrity/standard
[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/

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[OWASP-Top-10-Risiken]	OWASP Top 10 https://owasp.org/www-project-top-ten/
[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
[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
[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
[Verified Boot]	Verifizierter Start https://source.android.com/docs/security/features/verifiedboot?hl=de