

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

Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation Authentisierung des Versicherten ePA

Version: 1.34.0 CC
Revision: 244633304702
Stand: 30.0609.12.2020
Status: zur Abstimmung freigegeben
Klassifizierung: öffentlich Entwurf
Referenzierung: gemSpec_Authentisierung_Vers

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	18.12.18		initiale Erstellung	gematik
1.1.0	15.05.19		Einarbeitung Änderungsliste P18.1	gematik
1.2.0	02.03.20		Einarbeitung Änderungsliste P21.1	gematik
1.2.1	26.05.20		Einarbeitung Änderungsliste P21.3	gematik
			Einarbeitung offener Punkte für Release 4.0.0	
1.3.0.	30.06. 2020 <u>20</u>		freigegeben	gematik
<u>1.4.0 CC</u>	<u>09.12.20</u>		<u>Einarbeitung Änderungsliste P22.5</u>	<u>gematik</u>

Inhaltsverzeichnis

1 Einordnung des Dokumentes	6
1.1 Zielsetzung	6
1.2 Zielgruppe	6
1.3 Geltungsbereich	6
1.4 Abgrenzungen	6
1.5 Methodik	7
2 Systemkontext	8
3 Zerlegung der Komponente	9
4 Übergreifende Festlegungen	10
4.1 Datenschutz und Datensicherheit	10
4.2 Verwendete Standards	11
4.3 Fehlerbehandlung	13
4.4 Protokollierung	14
4.5 Nichtfunktionale Anforderungen	16
4.6 Identifikation der Akteure	16
5 Funktionsmerkmale	17
5.1 Authentisierung	18
5.1.1 Schnittstellen	18
5.1.1.1 Schnittstelle I_Authentication_Insurant	18
5.1.1.1.1 Operation login	19
5.1.1.1.2 Operation renew	28
5.1.1.1.3 Operation logout	31
5.1.1.1.4 Operation getAuditEvents	33
5.1.2 Umsetzung	36
5.1.2.1 Schnittstelle I_Authentication_Insurant	36
5.1.2.1.1 Operation login	36
5.1.2.1.2 Operation Renew	40
5.1.2.1.3 Operation Logout	41
5.1.2.1.4 Operation getAuditEvents	42
5.1.3 Lebensdauer der Authentifizierungsbestätigung	44
6 Informationsmodell	45
7 Verteilungssicht	46

65	8 Anhang A – Verzeichnisse	47
66	8.1 Abkürzungen	47
67	8.2 Glossar	48
68	8.3 Abbildungsverzeichnis	48
69	8.4 Tabellenverzeichnis	48
70	8.5 Referenzierte Dokumente	49
71	8.5.1 Dokumente der gematik	49
72	8.5.2 Weitere Dokumente	50
73	1 Einordnung des Dokumentes	6
74	1.1 Zielsetzung	6
75	1.2 Zielgruppe	6
76	1.3 Geltungsbereich	6
77	1.4 Abgrenzungen	6
78	1.5 Methodik	7
79	2 Systemkontext	8
80	3 Zerlegung der Komponente	9
81	4 Übergreifende Festlegungen	10
82	4.1 Datenschutz und Datensicherheit	10
83	4.2 Verwendete Standards	11
84	4.3 Fehlerbehandlung	13
85	4.4 Protokollierung	14
86	4.5 Nicht-Funktionale Anforderungen	16
87	4.6 Identifikation der Akteure	16
88	5 Funktionsmerkmale	17
89	5.1 Authentisierung	18
90	5.1.1 Schnittstellen	18
91	5.1.1.1 Schnittstelle I Authentication Insurant	18
92	5.1.1.1.1 Operation login	19
93	5.1.1.1.2 Operation renew	28
94	5.1.1.1.3 Operation logout	31
95	5.1.1.1.4 Operation getAuditEvents	33
96	5.1.1.1.5 Operation getSignedAuditEvents	34
97	5.1.2 Umsetzung	36
98	5.1.2.1 Schnittstelle I Authentication Insurant	36
99	5.1.2.1.1 Operation login	36
100	5.1.2.1.2 Operation Renew	40

101	5.1.2.1.3 Operation Logout.....	41
102	5.1.2.1.4 Operation getAuditEvents	42
103	5.1.2.1.5 Operation getSignedAuditEvents	43
104	5.1.3 Lebensdauer der Authentifizierungsbestätigung	44
105	6 Informationsmodell	45
106	7 Verteilungssicht	46
107	8 Anhang A – Verzeichnisse	47
108	8.1 Abkürzungen	47
109	8.2 Glossar	48
110	8.3 Abbildungsverzeichnis.....	48
111	8.4 Tabellenverzeichnis	48
112	8.5 Referenzierte Dokumente.....	49
113	8.5.1 Dokumente der gematik.....	49
114	8.5.2 Weitere Dokumente.....	50
115		

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die Anforderungen an die Teilkomponente "Authentisierung Versicherter" der Komponente "Zugangsgateway" (s.a. [gemSpec_Zugangsgateway_Vers]) des Produkttyps ePA-Aktensystem (s.a. [gemSpec_Aktensystem]).

Die Teilkomponente "Authentisierung Versicherter" ist zuständig für die Authentisierung von Versicherten und deren Vertretern innerhalb der Fachanwendung ePA (s.a. [gemSysL_ePA]).

1.2 Zielgruppe

Das Dokument ist maßgeblich für Anbieter und Hersteller des Produkttyps ePA-Aktensystem sowie für Anbieter und Hersteller von Produkten, die die Schnittstellen der Teilkomponente "Authentisierung Versicherter" nutzen.

1.3 Geltungsbereich

Dieses Dokument enthält normative Festlegungen zur Telematikinfrastruktur des Deutschen Gesundheitswesens. Der Gültigkeitszeitraum der vorliegenden Version und deren Anwendung in Zulassungs- oder Abnahmeverfahren wird durch die gematik GmbH in gesonderten Dokumenten (z.B. Dokumentenlandkarte, Produkttypsteckbrief, Leistungsbeschreibung) fest-gelegt und bekannt gegeben.

Schutzrechts-/Patentrechtshinweis

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

1.4 Abgrenzungen

Spezifiziert werden in dem Dokument die von dem Produkttyp bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert (siehe auch Kapitel 8.5).

150 Die vollständige Anforderungslage für den Produkttyp ergibt sich aus weiteren Konzept-
151 und Spezifikationsdokumenten, diese sind in dem Produkttypsteckbrief des Produkttyps
152 ePA-Aktensystem verzeichnet.

153 **1.5 Methodik**

154 Anforderungen als Ausdruck normativer Festlegungen werden durch eine eindeutige ID
155 sowie die dem RFC 2119 [RFC2119] entsprechenden, in Großbuchstaben geschriebenen
156 deutschen Schlüsselworte MUSS, DARF NICHT, SOLL, SOLL NICHT, KANN
157 gekennzeichnet.

158

159 Anforderungen werden im Dokument wie folgt dargestellt:

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

161 Text / Beschreibung

162 [\leq]

163 Dabei umfasst die Anforderung sämtliche zwischen Afo-ID und Textmarke

164 [\leq] angeführten Inhalte.

165

166

2 Systemkontext

167 Die Teilkomponente "Authentisierung Versicherter" der Komponente "Zugangsgateway"
168 des ePA-Aktensystems ist Teil des Produkttyps ePA. Der Systemüberblick ist
169 in [gemSysL_ePA] dargestellt.

170 Von der dezentralen Fachlogik im "ePA-Frontend des Versicherten" und dem Fachmodul
171 ePA wird die Komponente verwendet, um die Authentifizierung von Versicherten und
172 deren berechtigten Vertretern zu bestätigen.

173 Auf Anwendungsebene findet dabei ein Dialog zwischen aufrufendem Client (C) und der
174 Komponente "Authentisierung Versicherter" (S) statt:

- 175 • C fordert S auf, einen Authentisierungs-Token zu erstellen.
- 176 • S antwortet C mit der Aufforderung (Challenge), eine Zufallszahl zu signieren, um
177 sicherzustellen, dass die nachfolgende Authentisierungsnachricht frisch erzeugt
178 wird.
- 179 • C antwortet auf die Challenge mit einer Signatur für die Zufallszahl aus der
180 Challenge. Die Signatur erzeugt er mittels der Authentisierungsidentität
181 ID.CH.AUT der eGK oder der alternativen Versichertenidentität ID.CH.AUT_ALT.
- 182 • S authentifiziert C durch Prüfung der Signatur.
183 S stellt eine Authentifizierungsbestätigung aus und sendet sie an C.

184 Um Prüfungen durchzuführen, greift die Komponente auf Dienste der TI-Plattform zentral
185 zurück.

186

3 Zerlegung der Komponente

187

Eine weitere Untergliederung der Aufbaustruktur der Komponente ist nicht erforderlich.

ENTWURF

4 Übergreifende Festlegungen

Die Komponente "Authentisierung Versicherter" stellt eine X-User Assertion (XUA) gemäß [IHE#ITI-40] aus.

4.1 Datenschutz und Datensicherheit

A_14773 - Komponente Authentisierung Versicherter - Authentisierungsschlüssel

Die Komponente "Authentisierung Versicherter" MUSS die erstellten Authentifizierungsbestätigungen mit dem privaten Schlüssel der Ausstelleridentität ID.FD.SIG signieren. Das zugehörige Zertifikat C.FD.SIG MUSS die Rolle "oid_epa_authn" enthalten. [\leq]

Hinweis: Da die Identität ID.FD.SIG nur durch das Aktensystem selbst verwendet wird ist dafür die Schlüsselgeneration ECDSA zu verwenden (s. [gemSpec_Krypt]).

A_15091 - Komponente Authentisierung Versicherter - Verwendung eines HSM

Die Komponente "Authentisierung Versicherter" MUSS das private Schlüsselmaterial der Ausstelleridentität C.FD.SIG und der TLS-Server-Identität C.FD.TLS-S in einem HSM speichern. [\leq]

Zur Absicherung der Schnittstelle muss der Transport der SOAP-Nachrichten mittels HTTPS erfolgen. Dabei sind die Vorgaben zu TLS gem. [gemSpec_Krypt#3.3.2] und [gemSpec_PKI#8.4.1] umzusetzen.

Die Verbindung zum ePA-Frontend des Versicherten wird auf Transportebene mit TLS abgesichert. Auf dieser Ebene erfolgt eine serverseitige Authentisierung durch die Komponente "Authentisierung Versicherter" wie in [gemSpec_Zugangsgateway_Vers#Kapitel4.2] beschrieben.

Verbindungen innerhalb der TI werden ebenfalls auf Transportebene mit TLS abgesichert. Dabei werden Zertifikate der TI verwendet.

A_14227 - Komponente Authentisierung Versicherter - TLS-Authentisierung innerhalb der TI

Die Komponente "Authentisierung Versicherter" MUSS für alle innerhalb der TI zur Verfügung gestellten Schnittstellen ausschließlich Verbindungen mit TLS akzeptieren und dabei die einseitige Serverauthentisierung unter Nutzung des X.509-Komponentenzertifikats für TLS C.FD.TLS-S und der Rolle "oid_epa_authn" umsetzen. [\leq]

A_14801 - Komponente Authentisierung Versicherter - XML Schema-Validierung für SOAP-Eingangsnachrichten

Die Komponente "Authentisierung Versicherter" MUSS vor einer Weiterverarbeitung sämtliche SOAP 1.2-Eingangsnachrichten einer XML Schema-Validierung unterziehen und gemäß [SOAP] verarbeiten. Sind Nachrichten nicht wohlgeformt oder gültig, MUSS die Komponente "Authentisierung Versicherter" die Nachricht mit einem HTTP-Statuscode 400 gemäß [RFC7231] quittieren. [\leq]

A_14777 - Komponente Authentisierung Versicherter - Prüfung des Signaturzertifikats von Authentifizierungsbestätigungen

Die Komponente "Authentisierung Versicherter" MUSS sicherstellen, dass Authentifizierungsbestätigungen nur akzeptiert werden, wenn das zugehörige

Signaturzertifikat zeitlich gültig ist, nicht gesperrt wurde und nach dem Zertifikatsprofil C.FD.SIG für die Identität der Komponente Authentisierung Versicherter selbst ausgestellt wurde.

Dies kann durch eine aktuell gehaltene Konfiguration vertrauenswürdiger Zertifikate umgesetzt werden und ersetzt eine detaillierte Prüfung des Signaturzertifikats gem. [gemSpec_TBAuth#A_15557].

[<=]

A_14780 - Komponente Authentisierung Versicherter - Aussteller von Authentifizierungsbestätigungen

Die Komponente "Authentisierung Versicherter" MUSS sicherstellen, dass die Authentifizierungsbestätigung von der Komponente "Authentisierung Versicherter" selbst ausgestellt wurde (s.a. [gemSpec_TBAuth#GS-A_5494]).

[<=]

A_15605-01 - Komponente Authentisierung Versicherter - Ablehnung von SOAP 1.2-Nachrichten ohne UTF-8 Encodierung

Die Komponente "Authentisierung Versicherter" MUSS SOAP 1.2-Nachrichten mit einem HTTP-Statuscode 415 gemäß [RFC7231] quittieren, sofern die Zeichenkodierung im HTTP Header nicht UTF-8 benennt (Content-Type: charset=utf-8). [<=]

Diese Festlegungen zur UTF-8-Encodierung überschreibt die Festlegungen aus [WSIBP].

A_15613 - Komponente Authentisierung Versicherter – Erkennung von Denial-of-Service-Angriffen hinsichtlich dem Parsen von SOAP 1.2-Nachricht

Die Komponente "Authentisierung Versicherter" MUSS die folgenden Angriffstypen in eingehenden SOAP 1.2-Nachrichten erkennen und mit einem HTTP-Statuscode 400 gemäß [RFC7231] quittieren:

- XML Injection
- XPath Query Tampering
- XML External Entity Injection

[<=]

4.2 Verwendete Standards

Für die Übertragung von Nachrichten an den Schnittstellen der Komponente "Authentisierung Versicherter" wird SOAP in Verbindung mit HTTP verwendet.

A_14352 - Komponente Authentisierung Versicherter - Grundlegende Standards

Die Komponente "Authentisierung Versicherter" MUSS folgende Standards umsetzen, soweit diese im Rahmen der zu implementierenden Operationen verwendet werden und sofern sie nicht durch konkrete Anforderungen überschrieben werden:

- IHE ITI-40 Transaction "Provide X-User Assertion" [IHE#ITI-40]
- HTTP/1.1 [RFC7231]
- SOAP 1.2 [SOAP]
- WSDL 1.1 [WSDL]
- WSDL 1.1 Binding Extension for SOAP 1.2 [WSDL11SOAP12]

- 273 • WS-Trust 1.4 [WS-Trust]
- 274 • WS-I Basic Profile V2.0 [WSIBP]
- 275 • WS Security SAML Token Profile 1.1 [WSS-SAML]
- 276 • XSPA Profile of SAML for Healthcare v2.0 [XSPA-SAML]
- 277 • SAML V2.0 [SAML2.0]
- 278 • WS Security [WSS]

279 [**<=**]

280 Generell ist [gemSpec_Krypt] für alle Algorithmen und sonstigen kryptographischen
281 Vorgaben zu beachten.

282 Für die Schnittstellen der Komponente "Authentisierung Versicherter" werden die in der
283 folgenden Tabelle definierten XML-Präfixe verwendet.

284 **Tabelle 1: Tab_Auth_Vers_002 - Verwendete Namensräume und Präfixe**

Präfix	Namensraum	Referenz
phra	http://ws.gematik.de/fd/phrs/I_Authentication_Insurant/v1.1	
phr	http://ws.gematik.de/fa/phr/v1.0	
xs	http://www.w3.org/2001/XMLSchema	
saml	urn:oasis:names:tc:SAML:2.0:assertion	SAML 2.0 [SAML2.0]
soap	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2 [SOAP]
wsoap12	http://schemas.xmlsoap.org/wsdl/soap12/	[WSDL11SOAP12]
wsdl	http://schemas.xmlsoap.org/wsdl/	WSDL 1.1 [WSDL]
ds	http://www.w3.org/2000/09/xmldsig#	
xenc	http://www.w3.org/2001/04/xmlenc#	
wst	http://docs.oasis-open.org/ws-sx/ws-trust/200512	WS-Trust 1.4 [WS-Trust]
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	

wsaw	http://www.w3.org/2006/05/addressing/wsd	
tel	http://ws.gematik.de/tel/error/v2.0	

A_15604 - Komponente Authentisierung Versicherter - Kodierung in UTF-8

Die Komponente "Authentisierung Versicherter" MUSS bei der Erstellung von XML-Fragmenten das Encoding UTF-8 verwenden. [\leq]

4.3 Fehlerbehandlung

Bei Fehlern in der internen Verarbeitung oder bei fachlichen Fehlern in der Nutzung der bereitgestellten Schnittstellen liefert die Komponente "Authentisierung Versicherter" Fehlermeldungen zurück. Deren Struktur hängt davon ab, ob der Meldungsablauf auf [WS-Trust] basiert oder nicht.

Aufrufe mit Meldungen nach [WS-Trust] werden entsprechend auch mit Fehlermeldungen gemäß dem Standard beantwortet.

Andere Aufrufe werden als SOAP-Fault gemäß [gemSpec_OM] strukturiert und enthalten die in den Schnittstellendefinitionen angegebenen Fehlermeldungsinhalte innerhalb einer GERROR-Struktur gemäß [TelematikError.xsd].

A_14415 - Komponente Authentisierung Versicherter - Verwendung von Webservice-Fehlern

Die Komponente "Authentisierung Versicherter" MUSS an der Schnittstelle I_Authentication_Insurant:login den in [WS-Trust#Kapitel11] festgelegten SOAP-Fault-Mechanismus umsetzen. [\leq]

A_15138 - Komponente Authentisierung Versicherter - Inhalte der Fehlermeldungen

Die Komponente "Authentisierung Versicherter" MUSS in einer GERROR-Fehlermeldung gemäß [TelematikError.xsd] die Felder wie folgt mit den Fehlermeldungsinhalten der Schnittstellenbeschreibung befüllen:

- Fehlername Name: tel:Error/tel:Trace/tel:EventID
- Fehlerdetailtext Fehlertext: tel:Error/tel:Trace/tel:ErrorText
- Fehlercode: in tel:Error/tel:Trace/tel:Code entsprechend dem Fehlernamen gem. folgender Tabelle:

Tabelle 2: Tab_Auth_Vers_003 - Zuordnung Fehlercodes zu Fehlernamen

Name	Fehlercode
INTERNAL_ERROR	7720
SYNTAX_ERROR	7730

ASSERTION_INVALID	7740
-------------------	------

316 [\leq]

317 4.4 Protokollierung

318 Die Anforderungen an die Protokollierung für die Komponente leiten sich
319 aus [gemSysL_ePA#2.5.5] ab.

320 **A 13877-01A-13877 - Komponente Authentisierung Versicherter -** 321 **Verwaltungsprotokollierung**

322 Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf einer der in
323 [gemSpec_DM_ePA#A_14505] aufgelisteten Operationen der Schnittstelle
324 I_Authentication_Insurant unter der Voraussetzung, dass der Aufruf erfolgreich war, je
325 einen Eintrag im Verwaltungsprotokoll für den Versicherten ~~bzw. seinen Vertreter~~ gemäß
326 [gemSpec_DM_ePA#A_14471] vornehmen und die Parameterwerte dabei wie folgt
327 setzen:

328 **Tabelle 3: Tab_Auth_Vers_004 - Operationsabhängige Parameter des**
329 **Verwaltungsprotokolls**

Protokoll- parameter	Parameterwerte gemäß aufgerufener Operation
UserID	<p>KVNR (im SubjectDN des bestätigten C.CH.AUT bzw. C.CH.AUT_ALT Zertifikats enthalte</p> <p><u>Ermittlung für Operation</u> <u>I_Authentication_Insurant::getAuditEvents,</u> <u>I_Authentication_Insurant::getSignedAuditEvents,</u> <u>I_Authentication_Insurant::logoutToken</u> <u>über den folgenden XPath-Ausdruck zur "Subject ID" der im Operationsaufruf übergebenen</u> <u>//*[local-name()='Assertion' and namespace-uri() = 'urn:oasis:names:tc:SAML:2.0:assertion'</u> <u>name()='Attribute' and namespace-uri() = 'urn:oasis:names:tc:SAML:2.0:assertion'</u> <u>'urn:gematik:subject:subject-id']/*[local-name()='AttributeValue']/*[local-name()='InstanceIdentifier']/data(@extension)</u></p> <p><u>Ermittlung für die Operation I_Authentication_Insurant::loginCreateToken aus dem Inhalt des</u> <u>C.CH.AUT bzw. C.CH.AUT ALT Zertifikats, siehe Kap. 4.6).</u></p>
UserName	<p><u>subjectDN des als Parameter der Operation übergebenen C.CH.AUT bzw. C.CH.AUT_ALT Zertifikats</u></p> <p><u>Ermittlung für Operation</u> <u>I_Authentication_Insurant::getAuditEvents,</u> <u>I_Authentication_Insurant::getSignedAuditEvents,</u> <u>I_Authentication_Insurant::logoutToken über den folgenden XPath-Ausdruck zur Behauptung</u> <u>übergebenen Authentication Assertion:</u> <u>//*[local-name()='Assertion' and namespace-uri() = 'urn:oasis:names:tc:SAML:2.0:assertion'</u> <u>name()='Attribute' and namespace-uri() =</u> <u>'urn:oasis:names:tc:SAML:2.0:assertion'][@Name='http://schemas.xmlsoap.org:soap:envelop</u> <u>name()='AttributeValue']</u></p>

	<u>Ermittlung für die Operation I Authentication Insurant::loginCreateToken aus dem InH übergebenen C.CH.AUT bzw. C.CH.AUT ALT Zertifikats.</u>	
ObjectID	[nicht belegt]	
ObjectName	[nicht belegt]	
DeviceID	[nicht belegt]	
<u>ObjectDetail</u>	<u>gilt nur bei Operation I Authentication Insurant::loginCreateToken:</u>	
	<u>type</u>	<u>value</u>
	<u>AuthenticationType</u>	"eGK", wenn polic "alternative Auth <oid egk aut alt policyIdentifizier ist des übergebenen
übrige Protokolldaten	s. [gemSpec DM ePA#A 14471]	

Die nicht in [[gemSpec DM ePA#A 14505](#)] aufgelisteten Operationen der Schnittstelle I_Authentisierung_Insurant werden nicht protokolliert.

[<=]

A 21104 - Komponente Authentisierung Versicherter - Verwaltungsprotokollierung, wenn loginCreateToken fehlschlägt

Falls der Aufruf der Operation loginCreateToken fehl schlägt, MUSS die Komponente "Authentisierung Versicherter" einen Eintrag im Verwaltungsprotokoll für den Versicherten bzw. seinen Vertreter gemäß [[gemSpec DM ePA#A 14471](#)] vornehmen und zusätzlich zu den Vorgaben von A 13877 die Parameterwerte wie in Tab Auth Vers 0016 setzen. Es wird die Gesamtanzahl der fehlgeschlagenen Aufrufe der Operation loginCreateToken jeweils für eGK und al.vi gezählt und genau ein Protokolleintrag je Datumstag geschrieben.

**Tabelle 4 : Tab Auth Vers 0016- Operationsabhängige Parameter des
Verwaltungsprotokolls bei fehlerhaftem Aufruf der Operation loginCreateToken**

<u>Protokoll</u> <u>=</u> <u>paramet</u> <u>er</u>	<u>Parameterwerte gemäß aufgerufener Operation</u>	
<u>ObjectDet</u> <u>ail</u>	<u>type</u>	<u>value</u>
	<u>ErrorCounter_e</u> <u>GK</u>	<u>Anzahl der fehlgeschlagenen Aufrufe der</u> <u>Operation loginCreateToken mit eGK (policyIdentifizier =</u>

		<u><oid egk aut>) für diesen Tag korrespondierend ZU phr:AuditMessage/phr:EventIdentification/@EventDateTime</u>
	<u>ErrorCounter alvi</u>	<u>Anzahl der fehlgeschlagenen Aufrufe der Operation loginCreateToken mit al.vi (policyIdentifier = <oid egk aut alt>) für diesen Tag korrespondierend ZU phr:AuditMessage/phr:EventIdentification/@EventDateTime</u>

[<=]

Zur Protokollierung sind auch die Vorgaben in [gemSpec_Aktensystem#5.2] zu beachten.

4.5 ~~Nichtfunktionale~~ Nicht-Funktionale Anforderungen

Die die Komponente "Authentisierung Versicherter" betreffenden Anforderungen zu Skalierbarkeit, Performance und Mengengerüst sind in [gemSpec_Perf] zu finden.

4.6 Identifikation der Akteure

Der Versicherte bzw. der von ihm berechtigte Vertreter im Sinne der Fachanwendung ePA werden über ihre Krankenversicherungsnummer (KVNR) eindeutig identifiziert (vgl. [gemSysL_ePA#2.4.1]). Die KVNR besteht aus einem unveränderlichen Teil (Versicherten-ID) und einem veränderlichen Teil. In diesem Dokument ist mit der Abkürzung KVNR immer nur der unveränderliche Teil (Versicherten-ID) gemeint.

In den Zertifikaten einer eGK bzw. einer alternativen Versichertenidentität ist der unveränderliche Teil der KVNR in einem Feld organizationalUnitName des SubjectDN enthalten (s. [gemSpec_PKI#5.1]). Dabei ist zu beachten, dass das Feld organizationalUnitName im SubjectDN in zwei Ausprägungen auftritt (s. [gemSpec_PKI#4.2]):

- das zehnstellige alphanumerische Feld organizationalUnitName beinhaltet den unveränderlichen Teil der KVNR
- das neunstelligen numerische Feld organizationalUnitName beinhaltet das Institutionskennzeichen (Kassenzugehörigkeit)

Demzufolge muss für Versicherte bzw. deren berechtigte Vertreter der unveränderliche Teil der KVNR aus dem zehnstelligen alphanumerischen Feld organizationalUnitName von den Zertifikaten entnommen und zur Identifikation herangezogen werden.

370

5 Funktionsmerkmale

371 Die Komponente Authentisierung Versicherter realisiert ein Funktionsmerkmal über eine
372 Schnittstelle:

373 **Tabelle 5: Tab_Auth_Vers_005 - Schnittstellenübersicht der Komponente**
374 **Authentisierung des Versicherten**

Schnittstelle	Beschreibung und Operationen	
I_Authentication_Insurant	Schnittstelle zur Authentifizierung eines Versicherten	
	Logische Operation	Beschreibung
	login	Authentifizierung eines Versicherten
	renew	Erneuern der Authentifizierungsbestätigung für einen Versicherten auf Basis einer vorliegenden Authentifizierungsbestätigung
	logout	Beenden der Erneuerbarkeit der Authentifizierungsbestätigung für einen Versicherten
	getAuditEvents	Abruf der Verwaltungsprotokolleinträge
	<u>getSignedAuditEvents</u>	<u>Abruf der signierten Liste des Verwaltungsprotokolls</u>

375 Die Operation 'login' wird sowohl zur initialen Erstellung der
376 Authentifizierungsbestätigung als auch nach Ablauf der Gültigkeit der ursprünglichen
377 Authentifizierungsbestätigung zur Erstellung einer neuen Authentifizierungsbestätigung
378 aufgerufen.

379 Die Operation 'renew' erstellt eine neue Authentifizierungsbestätigung, wenn eine gültige
380 Authentifizierungsbestätigung vorgelegt wird, zu der noch kein 'logout' stattgefunden
381 hat.

382 Die Operation 'logout' beendet die Erneuerbarkeit einer Authentifizierungsbestätigung.

383 Die Komponente "Authentisierung Versicherter" nutzt die in der folgenden Tabelle
384 aufgeführten Schnittstellen der Telematikinfrastruktur.

385 **Tabelle 6: Tab_Auth_Vers_006 - Benutzte Schnittstellen der TI**

Schnittstelle	Bemerkung
I_IP_Transport	Definition in [gemSpec_Net]

I_DNS_Name_Resolution	Definition in [gemSpec_Net]
I_NTP_Time_Information	Definition in [gemSpec_Net]
I_OCSP_Status_Information	Definition in [gemSpec_PKI]
I_TSL_Download	Definition in [gemSpec_TSL]
I_Cert_provisioning	Definition in [gemSpec_X.509_TSP]
I_Cert_Revocation	Definition in [gemSpec_X.509_TSP]

386 **5.1 Authentisierung**

387 **5.1.1 Schnittstellen**

388 **5.1.1.1 Schnittstelle I_Authentication_Insurant**

389 Das Interface I_Authentication_Insurant stellt die in [gemSysL_ePA] definierte
390 Schnittstelle bereit.

391 **A_14228 - Komponente Authentisierung Versicherter -**

392 **I_Authentication_Insurant:login/renew/logout nach WS-Trust**

393 Die Komponente "Authentisierung Versicherter" MUSS einen Webservice-Endpunkt
394 AuthInsurantService bereitstellen, welcher die logischen Schnittstellen
395 I_Authentication_Insurant:login, I_Authentication_Insurant:renew
396 und I_Authentication_Insurant:logout durch die folgenden angebotenen Operationen
397 realisiert:

398 **Tabelle 7: Tab_Auth_Vers_007 - Schnittstellenübersicht der Authentisierung des**
399 **Versicherten**

Name	AuthInsurantService	
Version	1.0.0	
Namensraum	http://docs.oasis-open.org/ws-sx/ws-trust/200512	
Operationen	Name	Kurzbeschreibung
	LoginCreateChallenge	Login Teil 1 - Bereitgestellt über AuthInsurantService Request: RequestSecurityToken Response: RequestSecurityTokenResponse mit einer SignChallenge

	LoginCreateToken	Login Teil 2 - Bereitgestellt über AuthInsurantService Request: RequestSecurityTokenResponse mit einer SignChallengeResponse Response: RequestSecurityTokenResponseCollection
	RenewToken	Renew - Bereitgestellt über AuthInsurantService Request: RequestSecurityToken Response: RequestSecurityTokenResponse
	LogoutToken	Logout - Bereitgestellt über AuthInsurantService Request: RequestSecurityToken Response: RequestSecurityTokenResponse
WSDL	AuthenticationService.wsdl	

Die als SAML-Assertion zurückgelieferte Authentifizierungsbestätigung ist zur Vorlage bei den im Element *Audience* (s. Kap. 5.1.2.1.1) angegebenen Webservices bestimmt und kann durch den Aufrufer als opakes Token behandelt werden. Es ist mit der Identität der Komponente "Authentisierung Versicherter" signiert. [**<=**]

5.1.1.1.1 Operation login

Die Operation dient der Ausstellung von Authentifizierungsbestätigungen für Versicherte auf der Basis des Zertifikats C.CH.AUT oder C.CH.AUT_ALT des Versicherten.

Die Authentifizierungsbestätigung hat folgende wesentlichen Eigenschaften:

- Sie enthält das Zertifikat des Versicherten C.CH.AUT bzw. C.CH.AUT_ALT . Der Subject-DN aus diesem Zertifikat ist in ihr als Subjekt aufgeführt und enthält in einem der Felder *OrganizationalUnitName* die KVNR (s. Kap. 4.6).
- Der Authentication-Kontext im Feld *saml2:AuthnContextClassRef* der erzeugten Authentifizierungsbestätigung hängt vom Typ des übergebenen Zertifikats (C.CH.AUT oder C.CH.AUT_ALT) ab.
- Sie enthält in einem Attribut die aus dem Zertifikat extrahierte KVNR separat.
- Sie wird mit einer Gültigkeit von 5 Minuten ausgestellt.
- Sie legt als Methode zur *SubjectConfirmation* "Bearer" fest.

Voraussetzung für den Dialog auf Anwendungsebene ist eine etablierte TLS-Verbindung auf Transportebene.

Analog zu [WS-Trust#8] wird auf Anwendungsebene ein Signature Challenge Dialog implementiert. Abweichend von [WS-Trust#8.2] bzw. [WS-Trust#Appendix B] liegt der

Endpunkt auch für den Austausch der Signaturchallenge auf der Seite der Komponente "Authentisierung Versicherter", d.h. der Meldungsablauf ist in zwei durch den Aufrufer initiierte Meldungspaare aufgeteilt, deren Inhalte gemäß [WS-Trust] strukturiert sind.

Die logische Operation Login setzt sich daher auf Ebene der Webservices aus einer Abfolge der zwei Operationen LoginCreateChallenge und LoginCreateToken zusammen.

Die Fehlerbehandlung für diese beiden Operationen wird gemäß [WS-Trust#11] durchgeführt (vgl. Kap. 4.3).

Im Request zur Operation LoginCreateToken wird die Signatur des Versicherten über die von der Komponente "Authentisierung Versicherter" erstellten Challenge übertragen. Diese Übertragung erfolgt per WS-Security im SOAP-Header.

Die Bestückung der Nachrichtfelder wird an einem Beispiel illustriert und dann normativ festgelegt.

Beispiel Dialog

LoginCreateChallenge, Request:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    <Action xmlns="http://www.w3.org/2005/08/addressing">http://docs.oasis-
open.org/ws-sx/ws-trust/200512/RST/Issue</Action>
    <To
xmlns="http://www.w3.org/2005/08/addressing">https://localhost:9094/authn</
To>
  </soap:Header>
  <soap:Body>
    <RequestSecurityToken xmlns="http://docs.oasis-open.org/ws-sx/ws-
trust/200512">
      <TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.1#SAMLV2.0</TokenType>
      <RequestType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</RequestType>
    </RequestSecurityToken>
  </soap:Body>
</soap:Envelope>
```

LoginCreateChallenge, Response:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    <Action xmlns="http://www.w3.org/2005/08/addressing">http://docs.oasis-
open.org/ws-sx/ws-trust/200512/RSTR/Challenge</Action>
    <To
xmlns="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addr
essing/anonymous</To>
  </soap:Header>
  <soap:Body>
    <RequestSecurityTokenResponse xmlns="http://docs.oasis-open.org/ws-
sx/ws-trust/200512">
      <SignChallenge>
        <Challenge>JemuBWS...</Challenge>
      </SignChallenge>
    </RequestSecurityTokenResponse>
  </soap:Body>
```

474 </soap:Envelope>

475

476 LoginCreateToken, Request:

```
477     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
478         <soap:Header>
479             <wsse:Security xmlns:wsse="http://docs.oasis-
480 open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
481 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
482 wssecurity-utility-1.0.xsd" soap:mustUnderstand="true">
483                 <wsse:BinarySecurityToken EncodingType="http://docs.oasis-
484 open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
485 1.0#Base64Binary" ValueType="http://docs.oasis-
486 open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
487 wsu:Id="X509-c3b3a51c-a22b-4682-85a2-
488 5537d56ba5e2">MIIEzTCCA7WgA...</wsse:BinarySecurityToken>
489                 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
490 Id="SIG-f1f0472f-2f0d-468d-b425-0b1f5c78cc5a">
491                     <ds:SignedInfo>
492                         <ds:CanonicalizationMethod
493 Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
494                             <ec:InclusiveNamespaces
495 xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
496 PrefixList="soap"/>
497                         </ds:CanonicalizationMethod>
498                         <ds:SignatureMethod
499 Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
500                         <ds:Reference URI="#id-6c68f4bd-153d-42fb-a640-
501 890c5cc14771">
502                             <ds:Transforms>
503                                 <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-
504 exc-c14n#" />
505                             </ds:Transforms>
506                             <ds:DigestMethod
507 Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256"/>
508                             <ds:DigestValue>9Et/DvvJlSb0ZlSEequKHmOYTEizKYCKZlAEiDILG
509 FU=</ds:DigestValue>
510                         </ds:Reference>
511                     </ds:SignedInfo>
512                     <ds:SignatureValue>P21t+FT2tA...</ds:SignatureValue>
513                     <ds:KeyInfo Id="KI-bd93fc63-8828-46ad-8a6c-df08acabe5ce">
514                         <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-
515 open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
516 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
517 wssecurity-utility-1.0.xsd" wsu:Id="STR-d16144ef-1a31-45b8-b061-
518 537a93fbd515">
519                             <wsse:Reference URI="#X509-c3b3a51c-a22b-4682-85a2-
520 5537d56ba5e2" ValueType="http://docs.oasis-
521 open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
522 1.0#X509v3"/>
523                         </wsse:SecurityTokenReference>
524                     </ds:KeyInfo>
525                 </ds:Signature>
526             </wsse:Security>
527         <Action xmlns="http://www.w3.org/2005/08/addressing">http://docs.oasis-
528 open.org/ws-sx/ws-trust/200512/RSTR/ChallengeFinal</Action>
529     </To>
```

```

530 xmlns="http://www.w3.org/2005/08/addressing">https://localhost:9094/authn</
531 To>
532 </soap:Header>
533 <soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
534 200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="id-6c68f4bd-153d-42fb-a640-
535 890c5cc14771">

536 <RequestSecurityTokenResponse xmlns="http://docs.oasis-open.org/ws-
537 sx/ws-trust/200512">
538 <SignChallengeResponse>
539 <Challenge>JemuBWS-...</Challenge>
540 </SignChallengeResponse>
541 </RequestSecurityTokenResponse>
542 </soap:Body>
543 </soap:Envelope>

```

544

545 **LoginCreateToken, Response:**

```

546 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
547 <soap:Header>
548 <Action xmlns="http://www.w3.org/2005/08/addressing">http://docs.oasis-
549 open.org/ws-sx/ws-trust/200512/RSTRC/IssueFinal</Action>
550 <To
551 xmlns="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addr
552 essing/anonymous</To>
553 </soap:Header>

554 <soap:Body>
555 <RequestSecurityTokenResponseCollection xmlns="http://docs.oasis-
556 open.org/ws-sx/ws-trust/200512">
557 <RequestSecurityTokenResponse>
558 <RequestedSecurityToken>
559 <saml2:Assertion ...> ...
560 </saml2:Assertion>
561 </RequestedSecurityToken>
562 </RequestSecurityTokenResponse>
563 </RequestSecurityTokenResponseCollection>
564 </soap:Body>
565 </soap:Envelope>
566

```

567 **Normative Festlegung zum Dialog**

568 **A_14053 - Komponente Authentisierung Versicherter -**
569 **I_Authentication_Insurant:login nach WS-Trust, LoginCreateChallenge**
570 Die Komponente "Authentisierung Versicherter" MUSS die
571 OperationLoginCreateChallenge wie folgt anbieten:

572 **Tabelle 8: Tab_Auth_Vers_008 - Signatur der Schnittstelle**
573 **I_Authentication_Insurant:loginCreateChallenge**

Operation	loginCreateChallenge
Beschreibung	Login Teil 1 (Erzeugen der Challenge) Request: RequestSecurityToken

	Response: RequestSecurityTokenResponse mit einer SignChallenge		
Eingangsparameter			
Name	Beschreibung	Typ	opt.
/RequestSecurityToken	Request Security Token		n
/RequestSecurityToken /TokenType	Typ des Security Tokens. Wert: http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0		n
/RequestSecurityToken /RequestType	Angeforderte Funktion des Requests. Wert: http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue		n
Ausgangsparameter			
Name	Beschreibung	Typ	opt.
/RequestSecurityToken Response			n
/RequestSecurityToken Response /SignChallenge			n
/RequestSecurityToken Response /SignChallenge /Challenge	Enthält einen Zufallswert. Der Inhalt wird vom Aufrufer nicht ausgewertet.	String	n

Fehlermeldungen		
Fault/Code/Subcode/Value	Fault/Reason/Text	Details
wst:RequestFailed	The specified request failed	Interner Fehler in der Verarbeitungslogik
wst:InvalidRequest	The request was invalid or malformed	Es wurde ein fehlerhafter Aufrufparameter übergeben.

[<=]

A_14059 - Komponente Authentisierung Versicherter -

I_Authentication_Insurant:login nach WS-Trust, LoginCreateToken

Die Komponente "Authentisierung Versicherter" MUSS die OperationLoginCreateToken wie folgt anbieten:

**Tabelle 9: Tab_Auth_Vers_009 - Signatur der Schnittstelle
I_Authentication_Insurant:loginCreateToken**

Operation		loginCreateToken	
Beschreibung		Login Teil 2 Request: RequestSecurityTokenResponse mit einer SignChallengeResponse Response: RequestSecurityTokenResponseCollection	
Eingangsparameter			
Name	Beschreibung	Typ	opt.
/wsse:Security	Der WSSE SOAP Header enthält die Signatur über den Body sowie das zugehörige Zertifikat.		n
/wsse:Security /wsse:BinarySecurityToken	Zertifikat C.CH.AUT oder C.CH.AUT_ALT als BinarySecurityToken (s. [WSS#Kapitel6.3]) Hinweis: dabei kann es sich um ein Zertifikat der Schlüsselgeneration RSA oder ECDSA		n

	handeln (vgl. [gemSpec_Krypt]).		
/wsse:Security /ds:Signature	Signatur über den SOAP Body durch die Identität ID.CH.AUT bzw. ID.CH.AUT_ALT und Referenz auf das Zertifikat (s. [WSS#Kapitel8] und [WSS-X509#Kapitel3.2])		n
/RequestSecurityTokenResponse			n
/RequestSecurityTokenResponse /SignChallengeResponse /Challenge	Unveränderter Wert der vom Aufrufer in der Meldung zuvor empfangenen Challenge.		n
Ausgangsparameter			
Name	Beschreibung	Typ	opt.
/RequestSecurityTokenResponseCollection			
/RequestSecurityTokenResponseCollection /RequestSecurityTokenResponse			
/RequestSecurityTokenResponseCollection /RequestSecurityTokenResponse /RequestedSecurityToken	Dieser Parameter MUSS die in Kap. 5.1.2.1.1 definierte SAML Assertion enthalten Die Signatur der Komponente Authentisierung Versicherter ist in der SAML Assertion enthalten.		

/RequestSecurityToken ResponseCollection /RequestSecurityToken Response /RequestedSecurityToken /saml2:Assertion	Angeforderte AuthenticationAssertion als SAML Assertion		
Fehlermeldungen			
Fault/Code/Subcode/Value	Fault/Reason/Text	Details	
wst:RequestFailed	The specified request failed	Interner Fehler in der Verarbeitungslogik	
wst:InvalidRequest	The request was invalid or malformed	Es wurde ein fehlerhafter Aufrufparameter übergeben oder die Signatur der Eingangsnachricht ist nicht korrekt.	
wst:InvalidSecurityToken	Security token has been revoked	Das als BinarySecurityToken übergebene Zertifikat ist ungültig oder gesperrt.	

[<=]

A_14350 - Komponente Authentisierung Versicherter - I_Authentication_Insurant:login, Challenge Response Prüfung

Die Komponente "Authentisierung Versicherter" MUSS sicherstellen, dass die in der *SignChallengeResponse* verwendete *Challenge* folgende Eigenschaften hat:

- der Wert in der *Challenge* im Request der Operation *LoginCreateToken* muss identisch dem Wert aus der *Challenge* in der Response der Operation *LoginCreateChallenge* sein.
- der Zeitraum zwischen Erzeugung des Zufallswertes in der *Challenge* und dem Eintreffen der Nachricht Request der Operation *LoginCreateToken* darf nicht größer als 1 Minute sein.

[<=]

A_18985-03A_18985-02 - ePA-Client: Prüfen der AuthenticationAssertion

Ein ePA-Client (ePA-Frontend des Versicherten, ePA FM etc.) MUSS beim Erhalt des Authenticationtokens (AuthenticationAssertion) vergleichen, ob ~~das-eindeutige~~ sich die KVN-R als eindeutiges Merkmal des Nutzers ~~des ePA-Clients (KVN-R des Versicherten oder Telematik-ID), der, welcher~~ sich gegenüber dem Aktensystem authentifiziert hat, sich in den Behauptungen mit saml2-NameID-Feld als ~~Attribut:Attribute~~ Name="urn:gematik:subject:subject-id" wiederfindet (vgl. A_18985-02 Beispiel-1)). Falls die Prüfung ein negatives Ergebnis liefert, so MUSS der ePA-Client den Vorgang (Einloggen ins Aktensystem) abbrechen.

[<=]

A_18985-02-Beispiel-1:

```
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="_3886aa57-deb8-
422d-b0f8-54ff207d089a" IssueInstant="2020-05-11T16:59:53.420Z"
Version="2.0" xsi:type="saml2:AssertionType">
  <saml2:Issuer>https://aktor-gateway.gematik.de/authn</saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    ...
  </ds:Signature>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">CN=Dr. Emilio von BurgundTEST-ONLY, T=Dr.,
GIVENNAME=Emilio von, SURNAME=Burgund, OU=109500969, OU=X110474929, O=Test
GKV-SVNOT-VALID, C=DE</saml2:NameID>
    <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
  </saml2:Subject>
  <saml2:Conditions NotBefore="2020-05-11T16:59:53.420Z"
NotOnOrAfter="2020-05-11T17:04:53.420Z">
    <saml2:AudienceRestriction>
      <saml2:Audience>https://aktor-
gateway.gematik.de</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml2:Conditions>
  <saml2:AuthnStatement AuthnInstant="2020-05-11T16:53:53.322Z">
    <saml2:AuthnContext>
      <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:clas
ses:SmartcardPKI</saml2:AuthnContextClassRef>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
    <saml2:Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
      <saml2:AttributeValue xsi:type="xsd:string">Dr. Emilio von
BurgundTEST-ONLY</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
      <saml2:AttributeValue xsi:type="xsd:string">Emilio
von</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
```

```
655 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
656     <saml2:AttributeValue
657 xsi:type="xsd:string">Burgund</saml2:AttributeValue>
658     </saml2:Attribute>
659     <saml2:Attribute
660 Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/country"
661 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
662     <saml2:AttributeValue
663 xsi:type="xsd:string">DE</saml2:AttributeValue>
664     </saml2:Attribute>
665     <saml2:Attribute
666 Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"
667 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
668     <saml2:AttributeValue
669 xsi:type="xsd:string">X110474929</saml2:AttributeValue>
670     </saml2:Attribute>
671     <saml2:Attribute Name="urn:gematik:subject:subject-id"
672 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
673     <saml2:AttributeValue
674     <InstanceIdentifier xmlns="urn:hl7-org:v3"
675 extension="X110474929" root="1.2.276.0.76.4.8"/>
676     </saml2:AttributeValue>
677     </saml2:Attribute>
678     </saml2:AttributeStatement>
679 </saml2:Assertion>
```

680
681
682
683

684 5.1.1.1.2 Operation renew

685 Die Operation dient der Erneuerung einer Authentifizierungsbestätigung.

686 Die Bestückung der Nachrichtfelder wird an einem Beispiel illustriert und dann
687 normativ festgelegt.

688

689 Beispiel Dialog

690 RenewToken, Request:

```
691 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
692     <soap:Header>
693         <Action xmlns="http://www.w3.org/2005/08/addressing"> http://docs.oasis-
694 open.org/ws-sx/ws-trust/200512/RST/Renew</Action>
695         <To xmlns="http://www.w3.org/2005/08/addressing">...</To>
696     </soap:Header>
697     <soap:Body>
698         <RequestSecurityToken xmlns="http://docs.oasis-open.org/ws-sx/ws-
699 trust/200512">
700             <TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-
701 profile-1.1#SAMLV2.0</TokenType>
702             <RequestType>http://docs.oasis-open.org/ws-sx/ws-
703 trust/200512/Renew</RequestType>
704             <RenewTarget>... the token to be renewed ...</RenewTarget>
705         </RequestSecurityToken>
```

```

706     </soap:Body>
707 </soap:Envelope>
708
709 RenewToken, Response:
710 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
711     <soap:Header>
712         <Action xmlns="http://www.w3.org/2005/08/addressing"> http://docs.oasis-
713 open.org/ws-sx/ws-trust/200512/RSTR/RenewFinal</Action>
714         <To xmlns="http://www.w3.org/2005/08/addressing">...</To>
715     </soap:Header>
716
717
718     <soap:Body>
719         <RequestSecurityTokenResponse xmlns="http://docs.oasis-open.org/ws-
720 sx/ws-trust/200512">
721             <RequestedSecurityToken> ... the new token ...
722         </RequestedSecurityToken>
723         </RequestSecurityTokenResponse>
724     </soap:Body>
725 </soap:Envelope>

```

726
727 **A 17392-01A-17392 - Komponente Authentisierung Versicherter -**
728 **I_Authentication_Insurant:renew nach WS-Trust, RenewToken**
729 Die Komponente "Authentisierung Versicherter" MUSS die Operation renew wie folgt
730 anbieten:

Operation		RenewToken		
Beschreibung		renew —Bereitgestellt über AuthInsurantService Request: RequestSecurityToken Response: RequestSecurityTokenResponse		
Eingangsparameter				
Name	Beschreibung	Typ	opt.	
/RequestSecurityToken	Request Security Token		n	
/RequestSecurityToken /TokenType	Typ des Security Tokens. Wert: http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-		n	

	1.1#SAMLV2.0		
/RequestSecurityToken /RequestType	Angeforderte Funktion des Requests. Wert: http://docs.oasis-open.org/ws-sx/ws-trust/200512/Renew		n
/RequestSecurityToken /RenewTarget	Der Token, der verlängert werden soll		n
Ausgangsparameter			
Name	Beschreibung	Typ	opt.
/RequestSecurityToken Response			n
/RequestSecurityToken Response /RequestSecurityToken/RequestSecurityToken Response /RequestedSecurityToken	Dieser Parameter MUSS die in Kap. 5.1.2.1.2 definierte SAML Assertion enthalten Die Signatur der Komponente Authentisierung Versicherter ist in der SAML Assertion enthalten.		n
Fehlermeldungen			
Fault/Code/Subcode/Value	Fault/Reason/T ext	Details	

wst:RequestFailed	The specified request failed	Interner Fehler in der Verarbeitungslogik
wst:InvalidRequest	The request was invalid or malformed	Es wurde ein fehlerhafter Aufrufparameter übergeben.
wst:UnableToRenew	The requested renewal failed	Das übergebene Token ist abgelaufen oder aus anderen Gründen nicht erneuerbar.

731 [**<=**]

732 5.1.1.1.3 Operation logout

733 Die Operation beendet die Erneuerbarkeit einer Authentifizierungsbestätigung.

734 Die Bestückung der Nachrichtfelder wird an einem Beispiel illustriert und dann
735 normativ festgelegt.

736 **Beispiel Dialog**

737 LogoutToken, Request:

```

739 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
740   <soap:Header>
741     <Action xmlns="http://www.w3.org/2005/08/addressing"> http://docs.oasis-
742 open.org/ws-sx/ws-trust/200512/RST/Cancel</Action>
743     <To xmlns="http://www.w3.org/2005/08/addressing">...</To>
744   </soap:Header>
745   <soap:Body>
746     <RequestSecurityToken xmlns="http://docs.oasis-open.org/ws-sx/ws-
747 trust/200512">
748       <RequestType>http://docs.oasis-open.org/ws-sx/ws-
749 trust/200512/Cancel</RequestType>
750       <CancelTarget>... the token to be cancelled ...</CancelTarget>
751     </RequestSecurityToken>
752   </soap:Body>
753 </soap:Envelope>
754
```

755 LogoutToken, Response:

```

756 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
757   <soap:Header>
758     <Action xmlns="http://www.w3.org/2005/08/addressing">...</Action>
759     <To xmlns="http://www.w3.org/2005/08/addressing"> http://docs.oasis-

```

```

760 open.org/ws-sx/ws-trust/200512/RSTR/CancelFinal</To>
761 </soap:Header>
762 <soap:Body>
763   <RequestSecurityTokenResponse xmlns="http://docs.oasis-open.org/ws-
764   sx/ws-trust/200512">
765     <RequestedTokenCancelled/>
766   </RequestSecurityTokenResponse>
767 </soap:Body>
768 </soap:Envelope>

```

769 **A_17393-01 - Komponente Authentisierung Versicherter -**

770 **I_Authentication_Insurant:Logout nach WS-Trust, LogoutToken**

771 Die Komponente "Authentisierung Versicherter" MUSS die Operation Logout wie folgt
772 anbieten:

Operation		LogoutToken	
Beschreibung	Logout Request: RequestSecurityToken Response: RequestSecurityTokenResponse		
Eingangsparameter			
Name	Beschreibung	Typ	opt.
/RequestSecurityToken	Request Security Token		n
/RequestSecurityToken /RequestType	Angeforderte Funktion des Requests. Wert: http://docs.oasis-open.org/ws-sx/ws-trust/200512/Cancel		n
/RequestSecurityToken /CancelTarget	Der Token, für den der Logout erfolgen soll.		n
Ausgangsparameter			
Name	Beschreibung	Typ	opt.
/RequestSecurityToken Response			n
/RequestSecurityToken Response /RequestedTokenCancelled			n

Fehlermeldungen		
Fault/Code/Subcode/Value	Fault/Reason/Text	Details
wst:RequestFailed	The specified request failed	Interner Fehler in der Verarbeitungslogik
wst:InvalidRequest	The request was invalid or malformed	Es wurde ein fehlerhafter Aufrufparameter übergeben.

773 [**<=**]

774

775 5.1.1.1.4 Operation *getAuditEvents*

776 **A_14477 - Komponente Authentisierung Versicherter -**
777 **I_Authentication_Insurant::getAuditEvents**

778 Die Komponente "Authentisierung Versicherter" MUSS die
779 Operation *I_Authentication_Insurant::getAuditEvents* gemäß der folgenden Tabelle
780 implementieren:
781

782 **Tabelle 10: Tab_Auth_Vers_010 - Signatur der Schnittstelle**
783 **I_Authentication_Insurant::getAuditEvents**

Operation	I_Authentication_Insurant::getAuditEvents		
Beschreibung	Mit dieser Operation kann ein authentifizierter Versicherter bzw. ein berechtigter Vertreter das Verwaltungsprotokoll der Komponente "Authentisierung Versicherter" auslesen. Es werden nur Protokolleinträge zurückgegeben, die der authentifizierten Person zuzuordnen sind.		
Formatvorgaben	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthenticationService.xsd].		
Eingangsparameter			
Name	Beschreibung	Typ	opt.
AuthenticationAssertion	Die AuthenticationAssertion ist eine zuvor von der Komponente "Authentisierung Versicherter" ausgestellte	SAML Assertion(im WSSE SOAP Header gem.	-

	Authentifizierungsbestätigung.	[WSS-SAML#3.3])	
Ausgangsparameter			
Name	Beschreibung	Typ	opt.
AuditEventList	Liste der Verwaltungsprotokolleinträge, die sich auf die KVNR beziehen, die in dem zugehörigen Attribut der übergebenen Authentication-Assertion enthalten ist.	AuditMessage[0..*]	-
Fehlermeldungen			
Name	Fehlertext	Details	
INTERNAL_ERROR	Es ist ein interner Fehler aufgetreten.	Interner Fehler in der Verarbeitungslogik.	
ASSERTION_INVALID	Die übergebene AuthenticationAssertion ist ungültig.	z. B abgelaufen oder ungültige Signatur des Tokens.	
SYNTAX_ERROR	Fehlerhafte Aufrufparameter.	Es wurde ein fehlerhafter Aufrufparameter übergeben.	

[<=]

5.1.1.1.5 Operation *getSignedAuditEvents*

A 21162 - Komponente Authentisierung Versicherter -

I Authentication Insurant::getSignedAuditEvents

Die Komponente "Authentisierung Versicherter" MUSS die

Operation *I Authentication Insurant::getSignedAuditEvents* gemäß der folgenden

Tabelle implementieren:

Tabelle 11: Tab Auth Vers 016 - Signatur der Schnittstelle

I Authentication Insurant::getSignedAuditEvents

<u>Operation</u>	<u>I Authentication Insurant::getSignedAuditEvents</u>
-------------------------	---

<u>Beschreibung</u>	Mit dieser Operation kann ein authentifizierter Versicherter das signierte Verwaltungsprotokoll der Komponente "Authentisierung Versicherter" auslesen. Das signierte Verwaltungsprotokoll enthält alle Protokolleinträge, die der authentifizierten Person zuzuordnen sind.		
<u>Formatvorgaben</u>	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthenticationService.xsd].		
<u>Eingangsparameter</u>			
<u>Name</u>	<u>Beschreibung</u>	<u>Typ</u>	<u>opt.</u>
<u>AuthenticationAssertion</u>	Die AuthenticationAssertion ist eine zuvor von der Komponente "Authentisierung Versicherter" ausgestellte Authentifizierungsbestätigung.	<u>SAML Assertion(im WSSE SOAP Header gem. [WSS-SAML#3.3])</u>	=
<u>Ausgangsparameter</u>			
<u>Name</u>	<u>Beschreibung</u>	<u>Typ</u>	<u>opt.</u>
<u>SignedAuditEventList</u>	Signierte Liste der Verwaltungsprotokolleinträge, die sich auf die KVNR beziehen, die in dem zugehörigen Attribut der übergebenen Authentication-Assertion enthalten ist.	<u>Signiertes PDF/A-Dokument</u>	=
<u>Fehlermeldungen</u>			
<u>Name</u>	<u>Fehlertext</u>	<u>Details</u>	
<u>INTERNAL_ERROR</u>	Es ist ein interner Fehler aufgetreten.	<u>Interner Fehler in der Verarbeitungslogik.</u>	
<u>ASSERTION_INVALID</u>	Die übergebene AuthenticationAssertion ist ungültig.	<u>z. B abgelaufen oder ungültige Signatur des Tokens.</u>	
<u>SYNTAX_ERROR</u>	Fehlerhafte Aufrufparameter.	<u>Es wurde ein fehlerhafter</u>	

		<u>Aufrufparameter übergeben.</u>
--	--	---------------------------------------

[<=]

5.1.2 Umsetzung

5.1.2.1 Schnittstelle I_Authentication_Insurant

5.1.2.1.1 Operation login

A_15052 - Komponente Authentisierung Versicherter - loginCreateChallenge, Ablauf

Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der Operation `loginCreateChallenge` die folgenden Aktionen ausführen und bei den genannten Fehlerbedingungen die Fehlermeldungen (vgl. Kap. 5.1.1.1.1) entsprechend setzen:

Tabelle 12: Tab_Auth_Vers_011 - Ablauf von loginCreateChallenge

Aktion	Fehlerbedingung	Fehlermeldung
Validierung der Eingangsnachricht gegen die WSDL und die zugehörigen Schemadateien	Fehler bei der Validierung.	<code>wst:InvalidRequest</code> oder allgemeiner SOAP-Fault
Eingangsparameter entsprechend A_14053 prüfen	Fehlende Elemente oder falsche Inhalte oder andere Fehler im empfangenen Request.	<code>wst:InvalidRequest</code>
Zufallswert für die Responsemessage gem. [gemSpec_Krypt#GS-A_4367] erzeugen	Zufallswert nicht verfügbar oder andere interne Verarbeitungsfehler.	<code>wst:RequestFailed</code>

[<=]

A_14229 - Komponente Authentisierung Versicherter - loginCreateToken, Ablauf

Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der Operation `loginCreateToken` die folgenden Aktionen ausführen und bei den genannten Fehlerbedingungen die Fehlermeldungen (vgl. Kap. 5.1.1.1.1) entsprechend setzen:

Tabelle 13: Tab_Auth_Vers_012 - Ablauf von loginCreateToken

Aktion	Fehlerbedingung	Fehlermeldung
--------	-----------------	---------------

Validierung der Eingangsnachricht gegen die WSDL und die zugehörigen Schemadateien	Fehler bei der Validierung.	<code>wst:InvalidRequest</code> oder allgemeiner SOAP Fault
Prüfung WS-Security Header	Das Signaturzertifikat ist nicht vorhanden oder das Signaturverfahren entspricht nicht den Vorgaben von [gemSpec_Krypt].	<code>wst:InvalidRequest</code>
Prüfung mathematische Korrektheit der Signatur	Signatur nicht korrekt.	<code>wst:InvalidRequest</code>
<p>Das Signaturzertifikat muss gemäß [gemSpec_PKI#TUC_PKI_018] geprüft werden. Parameter:</p> <ul style="list-style-type: none"> • PolicyList: oid_egk_aut, oid_egk_aut_alt • intendedKeyUsage: digitalSignature • intendedExtendedKeyUsage: (leer) • OCSP-Graceperiod: 60 Minuten • Offline-Modus: nein • Prüfmodus: OCSP <p>Eine Prüfung der vom TUC zurückgelieferten Rollen-OID ist nicht erforderlich.</p>	Fehlermeldung des aufgerufenen TUC.	<code>wst:InvalidSecurityToken</code>
Eingangsparameter des SOAP Body entsprechend A_14059 prüfen	Fehlende Elemente oder falsche Inhalte oder andere Fehler.	<code>wst:InvalidRequest</code>
<i>Challenge</i> Element mit abgesendeter <i>Challenge</i> in Response zu <code>loginCreateChallenge</code> vergleichen	Challenges verschieden.	<code>wst:InvalidRequest</code>
AuthenticationAssertion (Token) gem. A_14109 erstellen und in Whitelist für Erneuerung aufnehmen (s. Kap. 5.1.3#A_17395)	Fehler in der internen Verarbeitung.	<code>wst:RequestFailed</code>

[<=]

Die Bestückung der Authentifizierungsbestätigung wird an einem Beispiel illustriert und dann normativ festgelegt.

Beispiel Authentifizierungsbestätigung

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="_108c30ac-bbcb-
42c9-b306-a61c39a6d890" IssueInstant="2018-09-20T11:29:19.858Z"
Version="2.0" xsi:type="saml2:AssertionType">
  <saml2:Issuer>https://[ePA_TI_FQDN]/authn</saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod
Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1" />
      <ds:Reference URI="#_108c30ac-bbcb-42c9-b306-a61c39a6d890">
        <ds:Transforms>
          <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#">
            <ec:InclusiveNamespaces
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="xsd" />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmenc#sha256" />
        <ds:DigestValue>TDtN2nJ05NUB1n18GL7AalUyuMVvrIHlEk1GKXLho2o
=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>aA4mAz3W2j7YWTKZmSXH2erR
5MtfzzOroWRLsy0wVwZdSsaK3MXW5pTnVjXE87Wq2dYJ30FhulQGGPWwz1qNxmynBiWlfu21UZ
NuroQycQCIOjHqw+wguYkZJQAA7exfyDAQYG8lgQbg4YiaIHWvy7l/VPu8fKaU/BgGobbnYyLuX
wg2DrTilD1XbunBpj25Hps4z6cS5zJZPPIIx8ZqOQ/keyz4Z+gcykj9Djv87lb/UZciBqtNR7nW
v9PhDwvFti9VvD3KbNixgoyNozGbgAdlc9qo4gLgmDXuMhZLrOADzVwDolmdx3/6rp+4vyMODdZ
GtIMA97EqPam+QF0DQ==</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>MIID...zA==</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName" NameQualifier="http://cxf.apache.org/sts">CN=Harald
Graf HünschTEST-
ONLY,2.5.4.42=#0c0b486172616c642047726166,2.5.4.4=#0c0748c3bc6e736368,OU=99
9567890,OU=X110446869,O=gematik MusterkasselGKVNOT-
VALID,C=DE</saml2:NameID>
    <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer" />
```

```
871     </saml2:Subject>
872     <saml2:Conditions NotBefore="2018-09-20T11:29:19.884Z"
873 NotOnOrAfter="2018-09-20T11:44:19.884Z">
874         <saml2:AudienceRestriction>
875             <saml2:Audience>[ePA_TI_FQDN]</saml2:Audience>
876         </saml2:AudienceRestriction>
877     </saml2:Conditions>
878     <saml2:AuthnStatement AuthnInstant="2018-09-20T11:29:19.878Z">
879         <saml2:AuthnContext>
880             <saml2:AuthnContextClassRef>
881                 urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
882             </saml2:AuthnContextClassRef>
883         </saml2:AuthnContext>
884     </saml2:AuthnStatement>
885     <saml2:AttributeStatement>
886         ...
887         <saml2:Attribute Name="urn:gematik:subject:subject-id"
888 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
889             <saml2:AttributeValue>
890                 <InstanceIdentifier xmlns="urn:hl7-org:v3"
891 extension="G995030566" root="1.2.276.0.76.4.8"/>
892             </saml2:AttributeValue>
893         </saml2:Attribute>
894     </saml2:AttributeStatement>
895 </saml2:Assertion>
896
897
```

A_14109-01 - Komponente Authentisierung Versicherter - Befüllung der Authentifizierungsbestätigung bei Login

Die Komponente "Authentisierung Versicherter" MUSS die für die Operation
loginCreateToken erzeugte *Authentifizierungsbestätigung* als SAML2-Assertion gemäß
[gemSpec_TBAuth#TAB_TBAuth_03] umsetzen und dabei folgende Vorgaben beachten:

- Das *Issuer*-Element muss als Aussteller des Token \$ePA_TI_FQDN/authn
enthalten, wobei \$ePA_TI_FQDN der anbieterspezifische FQDN in der TI ist.
- Die eingebettete Signatur *ds:Signature* wird mit der Identität der Komponente
Authentisierung Versicherter erstellt und das Element
ds:Signature/ds:KeyInfo/ds:X509Data/ds:X509Certificate muss das zugehörige
C.FD.SIG Zertifikat enthalten.
- Das Element *saml2:Subject/saml2:NameID* muss mit dem Subject-DN des
C.CH.AUT- bzw. C.CH.AUT_ALT-Zertifikats befüllt werden.
- Das Attribut *saml2:Subject/saml2:SubjectConfirmation/@Method* muss auf den
Wert "urn:oasis:names:tc:SAML:2.0:cm:bearer" gesetzt werden.
- Das Attribut *saml2:Conditions/@NotBefore* muss auf die Systemzeit gesetzt
werden.
- Das Attribut *saml2:Conditions/@NotOnOrAfter* muss auf (Systemzeit + 5 Minuten)
gesetzt werden.
- Das Element *saml2:Conditions/saml2:AudienceRestriction/saml2:Audience* muss
auf den FQDN des Anbieters des ePA-Aktensystems gemäß

919 [gemSpec_AktenSystem#A_14128] gesetzt werden: den TI-seitigen FQDN für TI-seitige
920 Aufrufe der Schnittstelle I_Authentication_Insurant bzw. den Internet-seitigen FQDN für
921 Internet-seitige Aufrufe der Schnittstelle I_Authentication_Insurant.

- 922 • Das Element
923 *saml2:AuthnStatement/saml2: AuthnContext/saml2:AuthnContextClassRef* muss
924 im Falle eines C.CH.AUT-Zertifikats auf den Wert
925 "urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI" und
926 im Falle eines C.CH.AUT_ALT-Zertifikats auf den Wert
927 "urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
928 gesetzt werden

929 [**<=**]

930 **A_15631 - Komponente Authentisierung Versicherter - Behauptungen in der** 931 **Authentifizierungsbestätigung**

932 Die Komponente "Authentisierung Versicherter" MUSS die für die Operation
933 loginCreateToken erzeugte *Authentifizierungsbestätigung* im Element *AttributeStatement*
934 mit den Behauptungen gemäß [gemSpec_TBAuth#TAB_TBAuth_02_2] befüllen und dabei
935 folgende Vorgaben beachten:

- 936 • Die Behauptungen müssen auf Basis des C.CH.AUT bzw. C.CH.AUT_ALT Zertifikats
937 gebildet werden.
- 938 • Die Behauptung "urn:gematik:subject:subject-id" muss enthalten sein und
939 basierend auf dem unveränderlichen Anteil der KVNR gebildet werden. Das
940 Attribut *Attribute/@NameFormat* muss dabei den Wert
941 "urn:oasis:names:tc:SAML:2.0:attrname-format:uri" haben.
- 942 • Die Behauptung "urn:gematik:subject:authreference" muss mit der
943 Seriennummer des C.CH.AUT- bzw. C.CH.AUT_ALT-Zertifikats gebildet werden.

944 [**<=**]

945 **5.1.2.1.2 Operation Renew**

946 **A_17398 - Komponente Authentisierung Versicherter - RenewToken**

947 Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der
948 Operation *RenewToken* die folgenden Aktionen ausführen und bei den genannten
949 Fehlerbedingungen die Fehlermeldungen (vgl. Kap. 5.1.1.1.2) entsprechend setzen:

950 **Tabelle 14: Tab_Auth_Vers_015 - Ablauf von RenewToken**

Aktion	Fehlerbedingung	Fehlermeldung
Validierung der Eingangsnachricht gegen die WSDL und die zugehörigen Schemadateien	Fehler bei der Validierung.	wst:InvalidRequest oder allgemeiner SOAP-Fault
Eingangsparameter entsprechend A_17392 prüfen	Fehlende Elemente oder falsche Inhalte oder andere Fehler im empfangenen Request.	wst:InvalidRequest

Prüfung gegen WhiteList entsprechend A_17395 und Entfernen des Tokens aus der Whitelist	Token nicht in Whitelist vorhanden	wst:UnableToRenew
Erstellung der neuen Authentifizierungsbestätigung gemäß A_17793 und ggf. Aufnahme in Whitelist für Erneuerung (gem. Kap. 5.1.3#A_17395)	Fehler in der internen Verarbeitung.	wst:RequestFailed

[<=]

A_17793 - Komponente Authentisierung Versicherter - Befüllung der Authentifizierungsbestätigung bei Renew

Die Komponente "Authentisierung Versicherter" MUSS die für die Operation RenewToken erzeugte Authentifizierungsbestätigung als SAML2-Assertion gemäß [gemSpec_TBAuth#TAB_TBAuth_03] umsetzen und dabei folgende Vorgaben beachten:

- Das Attribut saml2:Conditions/@NotBefore muss auf die Systemzeit gesetzt werden.
- Das Attribut saml2:Conditions/@NotOnOrAfter muss auf (Systemzeit+5 Minuten) gesetzt werden.
- Alle anderen Attribute werden aus der zu verlängernden Authentifizierungsbestätigung aus der Whitelist (s. Kap. 5.1.3 übernommen). Insbesondere betrifft dies auch das Element saml2:AuthnStatement mit dem Attribut AuthnInstant.
- Die eingebettete Signatur ds:Signature wird mit der Identität der Komponente Authentisierung Versicherter erstellt und das Element ds:Signature/ds:KeyInfo/ds:X509Data/ds:X509Certificate muss das zugehörige C.FD.SIG Zertifikat enthalten.

[<=]

5.1.2.1.3 Operation Logout

A_17412 - Komponente Authentisierung Versicherter - LogoutToken

Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der OperationLogoutToken die folgenden Aktionen ausführen und bei den genannten Fehlerbedingungen die Fehlermeldungen (vgl. Kap. 5.1.1.1.3) entsprechend setzen:

Tabelle 15: Tab_Auth_Vers_015 - Ablauf von RenewToken

Aktion	Fehlerbedingung	Fehlermeldung
Validierung der Eingangsnachricht gegen die	Fehler bei der Validierung.	wst:InvalidRequest oder allgemeiner SOAP- Fault

WSDL und die zugehörigen Schemadateien		
Eingangsparameter entsprechend A_17393 prüfen	Fehlende Elemente oder falsche Inhalte oder andere Fehler im empfangenen Request.	wst:InvalidRequest
Authentifizierungsbestätigung (Token) aus Whitelist für Erneuerung entfernen	Authentifizierungsbestätigung nicht in Whitelist vorhanden	(keine Fehlermeldung)

978
979 [**<=**]

980

981 **5.1.2.1.4 Operation *getAuditEvents***

982 Die Vorgaben zur Erstellung der Protokolleinträge sind in Kap. 4.4 beschrieben. Zur
983 Prüfung der Berechtigung des Abrufs des Protokolls wird die übergebene
984 Authentifizierungsbestätigung geprüft.

985 **A_14781 - Komponente Authentisierung Versicherter - *getAuditEvents*,**
986 **Prüfschritte**

987 Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der
988 Operation *getAuditEvents* die Prüfschritte für *Authentifizierungsbestätigungen* gem. Kap.
989 4.2 mit der als Eingangsparameter übergebenen *Authentifizierungsbestätigung* ausführen
990 und die Fehlermeldung (vgl. Kap. 5.1.1.1.2) wie folgt setzen:

991 **Tabelle 16: Tab_Auth_Vers_013 - Prüfschritte bei *getAuditEvents***

Fehlerbedingung	Fehlermeldung
Fehler bei Validierung der Eingangsnachricht gegen die WSDL oder die zugehörigen Schemadateien	SYNTAX_ERROR oder allgemeiner SOAP Fault
Fehler im empfangenen Request	SYNTAX_ERROR
Interner Fehler in der Verarbeitungslogik	INTERNAL_ERROR
Ein Prüfschritt der Signaturprüfung gem. [gemSpec_TBAuth#A_15556] bzw. [gemSpec_Authentisierung_Vers#A_14777] liefert einen Fehler.	ASSERTION_INVALID ID
Ein Prüfschritt der Inhaltsprüfung gem. [gemSpec_TBAuth#A_15558]/[gemSpec_Authentisierung_Vers#A_14780] bzw. [gemSpec_TBAuth#A_15637] liefert einen Fehler.	ASSERTION_INVALID ID

[<=]

A_14803 - Komponente Authentisierung Versicherter - Umsetzung getAuditEvents

Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der Operation `getAuditEvents` die Liste aller Verwaltungsprotokolleinträge gemäß [\[gemSpec DM ePA#A 14471\]](#) zurückliefern, die der Identität in der übergebenen *Authentifizierungsbestätigung* entsprechen.

[<=]

5.1.2.1.5 Operation `getSignedAuditEvents`

Die Vorgaben zur Erstellung der Protokolleinträge sind in Kap. 4.4 beschrieben. Zur Prüfung der Berechtigung des Abrufs des Protokolls wird die übergebene Authentifizierungsbestätigung geprüft.

A 21163 - Komponente Authentisierung Versicherter - `getSignedAuditEvents`, Prüfschritte

Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der Operation `getSignedAuditEvents` die Prüfschritte für *Authentifizierungsbestätigungen* gem. Kap. 4.2 mit der als Eingangsparameter übergebenen *Authentifizierungsbestätigung* ausführen und die Fehlermeldung (vgl. Kap. 5.1.1.1.2) wie folgt setzen:

Tabelle 17: Tab Auth Vers 017 - Prüfschritte bei `getSignedAuditEvents`

<u>Fehlerbedingung</u>	<u>Fehlermeldung</u>
<u>Fehler bei Validierung der Eingangsnachricht gegen die WSDL oder die zugehörigen Schemadateien</u>	<u>SYNTAX_ERROR oder allgemeiner SOAP Fault</u>
<u>Fehler im empfangenen Request</u>	<u>SYNTAX_ERROR</u>
<u>Interner Fehler in der Verarbeitungslogik</u>	<u>INTERNAL_ERROR</u>
<u>Ein Prüfschritt der Signaturprüfung gem. [gemSpec TBAuth#A 15556] bzw. [gemSpec Authentisierung Vers#A 14777] liefert einen Fehler.</u>	<u>ASSERTION_INVALID</u>
<u>Ein Prüfschritt der Inhaltsprüfung gem. [gemSpec TBAuth#A 15558]/[gemSpec Authentisierung Vers#A 14780] bzw. [gemSpec TBAuth#A 15637] liefert einen Fehler.</u>	<u>ASSERTION_INVALID</u>

[<=]

A 21164 - Komponente Authentisierung Versicherter - Umsetzung getSignedAuditEvents

Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der Operation `getSignedAuditEvents` ein signiertes PDF/A-Dokument zurückliefern, welches alle Verwaltungsprotokolleinträge gemäß [\[gemSpec DM ePA#A 14471\]](#) enthält, die der

Identität in der übergebenen Authentifizierungsbestätigung entsprechen, wobei für die Signatur der Liste der private Schlüssel der Ausstelleridentität ID.FD.SIG genutzt wird, dessen zugehöriges Zertifikat C.FD.SIG die Rolle "oid_ePA_logging" enthält. [≤]

Es wird das gesamte PDF-Dokument signiert. Beim Anlegen des PDF-Dokuments muss Platz für die Signatur vorgesehen werden.

5.1.3 Lebensdauer der Authentifizierungsbestätigung

Die Authentifizierungsbestätigung (Token) wird mit einer kurzen Lebensdauer erstellt. Innerhalb dieser Lebensdauer kann über die Operation Renew ein neuer Token wieder mit einer kurzen Lebensdauer ausgestellt werden. Durch Aufruf der Logout Operation wird die Möglichkeit eines erneuten Renew unterbunden. Die Gesamtlebensdauer, über die ein Renew erfolgen kann, wird beschränkt.

A_17395 - Komponente Authentisierung Versicherter - Whitelist

Die Komponente "Authentisierung Versicherter" MUSS eine Whitelist der aktiven Authentifizierungsbestätigungen (Token) mit folgenden Eigenschaften führen:

- Authentifizierungsbestätigungen (Token), die als Ergebnis von Login oder Renew zurückgeliefert werden, werden in die Whitelist eingetragen, sofern die Zeit im Attribut *saml2:Conditions/@NotOnOrAfter* weniger als 120 Minuten später liegt als die Zeit im Attribut *saml2:AuthnStatement@AuthnInstant*.
- Authentifizierungsbestätigungen (Token), die als Eingangsparameter von Renew verlängert werden sollen oder deren Verlängerbarkeit als Eingangsparameter von Logout beendet wird, werden aus der Whitelist entfernt
- Authentifizierungsbestätigungen (Token), die zeitlich abgelaufen sind (d.h. die aktuelle Systemzeit liegt später als *saml2:Conditions/@NotOnOrAfter*) werden aus der Whitelist entfernt

[≤]

Die Whitelist wirkt somit ausschließlich als Einschränkung für die Operation Renew:

- Token, die nicht auf der Whitelist stehen, werden nicht verlängert und
- Token, für die der Authentifizierungszeitpunkt länger als die gegebene Zeitspanne zurückliegt, werden ebenfalls nicht verlängert.

Für die konkrete Ausgestaltung der Aktualisierung der Whitelist werden keine Vorgaben gemacht. Die Anforderungen in dieser Spezifikation stellen nur das logische Modell des Verhaltens der Whitelist dar. Umsetzungen sind spezifikationskonform, sofern dieses Verhalten an der Schnittstelle der Komponente reproduziert wird.

1056

6 Informationsmodell

1057 Ein gesondertes Informationsmodell der durch den Produkttypen verarbeiteten Daten
1058 wird nicht benötigt.

ENTWURF

1059

7 Verteilungssicht

- 1060 Eine Darstellung der hardwareseitigen Verteilung des Produkttyps bzw. seiner
1061 Teilsysteme und der Einbettung in die physikalische Umgebung wird nicht benötigt.

ENTWURF

1062

8 Anhang A – Verzeichnisse

1063

8.1 Abkürzungen

Kürzel	Erläuterung
CDA	Clinical Document Architecture
eGK	elektronische Gesundheitskarte
ePA	elektronische Patientenakte
FdV	ePA-Frontend des Versicherten
FQDN	Fully-Qualified Domain Name
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
IHE	Integrating the Healthcare Enterprise
KVNR	Krankenversichertennummer (vgl. Kap. 4.6)
OID	Object Identifier
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
TI	Telematikinfrastruktur
TLS	Transport Layer Security
TUC	Technical Use Case
VAU	Vertrauenswürdige Ausführungsumgebung
W3C	World Wide Web Consortium
WS-I	Web-Services Interoperability Consortium
WSDL	Web Services Description Language
XACML	eXtensible Access Control Markup Language
XSPA	Cross-Enterprise Security and Privacy Authorization Profile
XUA	Cross-Enterprise User Assertion Profile

8.2 Glossar

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

8.3 Abbildungsverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

8.4 Tabellenverzeichnis

Tabelle 1: Tab_Auth_Vers_002	Verwendete Namensräume und Präfixe.....	12
Tabelle 2: Tab_Auth_Vers_003	Zuordnung Fehlercodes zu Fehlernamen	13
Tabelle 3: Tab_Auth_Vers_004	Operationsabhängige Parameter des Verwaltungsprotokolls.....	14
Tabelle 4: Tab_Auth_Vers_005	Schnittstellenübersicht der Komponente Authentisierung des Versicherten	17
Tabelle 5: Tab_Auth_Vers_006	Benutzte Schnittstellen der TI.....	17
Tabelle 6: Tab_Auth_Vers_007	Schnittstellenübersicht der Authentisierung des Versicherten	18
Tabelle 7: Tab_Auth_Vers_008	Signatur der Schnittstelle I_Authentication_Insurant:loginCreateChallenge	22
Tabelle 8: Tab_Auth_Vers_009	Signatur der Schnittstelle I_Authentication_Insurant:loginCreateToken	24
Tabelle 9: Tab_Auth_Vers_010	Signatur der Schnittstelle I_Authentication_Insurant::getAuditEvents	33
Tabelle 10: Tab_Auth_Vers_011	Ablauf von loginCreateChallenge	36
Tabelle 11: Tab_Auth_Vers_012	Ablauf von loginCreateToken	36
Tabelle 12: Tab_Auth_Vers_015	Ablauf von RenewToken	40
Tabelle 13: Tab_Auth_Vers_015	Ablauf von RenewToken	41
Tabelle 14: Tab_Auth_Vers_013	Prüfschritte bei getAuditEvents	42
Tabelle 1: Tab_Auth_Vers_002	Verwendete Namensräume und Präfixe.....	12
Tabelle 2: Tab_Auth_Vers_003	Zuordnung Fehlercodes zu Fehlernamen	13
Tabelle 3: Tab_Auth_Vers_004	Operationsabhängige Parameter des Verwaltungsprotokolls.....	14
Tabelle 4 : Tab_Auth_Vers_0016	Operationsabhängige Parameter des Verwaltungsprotokolls bei fehlerhaftem Aufruf der Operation loginCreateToken	15
Tabelle 5: Tab_Auth_Vers_005	Schnittstellenübersicht der Komponente Authentisierung des Versicherten	17
Tabelle 6: Tab_Auth_Vers_006	Benutzte Schnittstellen der TI.....	17

<u>Tabelle 7: Tab Auth Vers 007 - Schnittstellenübersicht der Authentisierung des Versicherten</u>	<u>18</u>
<u>Tabelle 8: Tab Auth Vers 008 - Signatur der Schnittstelle I Authentication Insurant:loginCreateChallenge</u>	<u>22</u>
<u>Tabelle 9: Tab Auth Vers 009 - Signatur der Schnittstelle I Authentication Insurant:loginCreateToken</u>	<u>24</u>
<u>Tabelle 10: Tab Auth Vers 010 - Signatur der Schnittstelle I Authentication Insurant::getAuditEvents</u>	<u>33</u>
<u>Tabelle 11: Tab Auth Vers 016 - Signatur der Schnittstelle I Authentication Insurant::getSignedAuditEvents</u>	<u>34</u>
<u>Tabelle 12: Tab Auth Vers 011 - Ablauf von loginCreateChallenge</u>	<u>36</u>
<u>Tabelle 13: Tab Auth Vers 012 - Ablauf von loginCreateToken</u>	<u>36</u>
<u>Tabelle 14: Tab Auth Vers 015 - Ablauf von RenewToken</u>	<u>40</u>
<u>Tabelle 15: Tab Auth Vers 015 - Ablauf von RenewToken</u>	<u>41</u>
<u>Tabelle 16: Tab Auth Vers 013 - Prüfschritte bei getAuditEvents</u>	<u>42</u>
<u>Tabelle 17: Tab Auth Vers 017 - Prüfschritte bei getSignedAuditEvents</u>	<u>43</u>

8.5 Referenzierte Dokumente

8.5.1 Dokumente der gematik

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

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Glossar der Telematikinfrastruktur
[gemSysL_ePA]	gematik: Systemspezifisches Konzept ePA
[gemSpec_Aktensystem]	gematik: Spezifikation Aktensystem ePA
[gemSpec_DM_ePA]	gematik: Datenmodell ePA
[gemSpec_Zugangsgateway_Vers]	gematik: Spezifikation Zugangsgateway des Versicherten ePA
[gemSpec_Autorisierung]	gematik: Spezifikation Autorisierung ePA

[gemSpec_FM_ePA]	gematik: Spezifikation Fachmodul ePA
[gemSpec_ePA_FdV]	gematik: Spezifikation Frontend des Versicherten ePA
[gemKPT_Arch_TIP]	gematik: Konzept Architektur der TI-Plattform
[gemKPT_PKI_TIP]	gematik: Konzept PKI der TI-Plattform
[gemSpec_Net]	gematik: Übergreifenden Spezifikation Netzwerk
[gemSpec_Perf]	gematik: Spezifikation Performancevorgaben und Mengengerüst
[gemSpec_OID]	gematik: Spezifikation Festlegung von OIDs
[gemSpec_OM]	gematik: Übergreifende Spezifikation Operations und Maintenance
[gemSpec_PKI]	gematik: Spezifikation PKI
[gemSpec_Krypt]	gematik: Übergreifende Spezifikation - Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur
[gemSpec_X.509_TSP]	gematik: PKI für X.509-Zertifikate: Spezifikation Trust Service Provider X.509
[gemSpec_TSL]	gematik: Spezifikation TSL-Dienst

1126 8.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[RFC2119]	IETF (1997): Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, http://tools.ietf.org/html/rfc2119
[RFC7231]	IETF (2014): Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, RFC 7231, https://tools.ietf.org/html/rfc7231
[SOAP]	W3C (2007): SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), https://www.w3.org/TR/soap12-part1/
[SAML2.0]	Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 http://docs.oasis-open.org/security/saml/v2.0/
[WSDL]	W3C: Web Services Description Language (WSDL) 1.1 https://www.w3.org/TR/wSDL.html
[WSDL11SOAP12]	W3C (2006): WSDL 1.1 Binding Extension for SOAP 1.2, https://www.w3.org/Submission/wSDL11soap12/

[WSIBP]	Web-Services Interoperability Consortium (2010): WS-I Basic Profile V2.0 (final material), http://ws-i.org/Profiles/BasicProfile-2.0-2010-11-09.html
[WSS]	OASIS (2006): Web Services Security: SOAP Message Security 1.1 (WS-Security 2004), http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf
[WSS-SAML]	OASIS (2006): Web Services Security: SAML Token Profile 1.1, https://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTOKENProfile.pdf
[WS-Trust]	WS-Trust 1.4 OASIS Standard incorporating Approved Errata01 25.04.2012 http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/errata01/os/ws-trust-1.4-errata01-os-complete.doc
[XSPA-SAML]	OASIS: Cross-Enterprise Security and Privacy Authorization (XSPA) Profile of Security Assertion Markup Language (SAML) for Healthcare Version 2.0 http://docs.oasis-open.org/xspa/saml-xspa/v2.0/saml-xspa-v2.0.html
[IHE#ITI-40]	IHE IT Infrastructure Technical Framework Volume 2b (ITI TF-2b) – Transactions Part B, Revision 15.0, Section 3.40 Provide X-User Assertion [ITI-40] http://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2b.pdf

1127