Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation Authentisierung des Versicherten ePA

Version:	1.3.0
Revision:	244633
Stand:	30.06.2020
Status:	freigegeben
Klassifizierung:	öffentlich
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		freigegeben	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes 5
1.1 Zielsetzung 5
1.2 Zielgruppe 5
1.3 Geltungsbereich 5
1.4 Abgrenzungen 5
1.5 Methodik 6
2 Systemkontext 7
3 Zerlegung der Komponente 8
4 Übergreifende Festlegungen 9
4.1 Datenschutz und Datensicherheit 9
4.2 Verwendete Standards 10
4.3 Fehlerbehandlung 12
4.4 Protokollierung 13
4.5 Nicht-Funktionale Anforderungen 13
4.6 Identifikation der Akteure 14
5 Funktionsmerkmale 15
5.1 Authentisierung 16
5.1.1 Schnittstellen 16
5.1.1.1 Schnittstelle I_Authentication_Insurant 16
5.1.1.1.1 Operation login 17
5.1.1.1.2 Operation renew 26
5.1.1.1.3 Operation logout 29
5.1.1.1.4 Operation getAuditEvents 30
5.1.2 Umsetzung 32
5.1.2.1 Schnittstelle I_Authentication_Insurant 32
5.1.2.1.1 Operation login 32
5.1.2.1.2 Operation Renew 36
5.1.2.1.3 Operation Logout 37
5.1.2.1.4 Operation getAuditEvents 38
5.1.3 Lebensdauer der Authentifizierungsbestätigung 39
6 Informationsmodell 41
7 Verteilungssicht 42

8 Anhang A – Verzeichnisse 43
8.1 Abkürzungen 43
8.2 Glossar 44
8.3 Abbildungsverzeichnis 44
8.4 Tabellenverzeichnis 44
8.5 Referenzierte Dokumente 44
8.5.1 Dokumente der gematik 44
8.5.2 Weitere Dokumente 46

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).

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

1.5 Methodik

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

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 Systemkontext

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

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

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

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

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

3 Zerlegung der Komponente

Eine weitere Untergliederung der Aufbaustruktur der Komponente ist nicht erforderlich.

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.**[**<=**]**

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.**[**<=**]**

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.**[**<=**]**

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.**[**<=**]**

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 Enkodierung
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-Enkodierung 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]

- WS-Trust 1.4 [WS-Trust]
- WS-I Basic Profile V2.0 [WSIBP]
- WS Security SAML Token Profile 1.1 [WSS-SAML]
- XSPA Profile of SAML for Healthcare v2.0 [XSPA-SAML]
- SAML V2.0 [SAML2.0]
- WS Security [WSS]

[<=]

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

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

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/wsdl	
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.**[<=]**

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.
[<=]

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

[<=]

4.4 Protokollierung

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

A_13877 - Komponente Authentisierung Versicherter - Verwaltungsprotokollierung
Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf einer der in [gemSpec_DM_ePA#A_14505] aufgelisteten Operationen der Schnittstelle I_Authentication_Insurant je einen Eintrag im Verwaltungsprotokoll für den Versicherten bzw. seinen Vertreter gemäß [gemSpec_DM_ePA#A_14471] vornehmen und die Parameterwerte dabei wie folgt setzen:

Tabelle 3: Tab_Auth_Vers_004 - Operationsabhängige Parameter des Verwaltungsprotokolls

Protokoll-parameter	Parameterwerte gemäß aufgerufener Operation
UserID	KVNR (im SubjectDN des bestätigten C.CH.AUT bzw. C.CH.AUT_ALT Zertifikats enthalten, s. Kap. 4.6).
UserName	subjectDN des als Parameter der Operation übergebenen C.CH.AUT bzw. C.CH.AUT_ALT Zertifikats.
ObjectID	[nicht belegt]
ObjectName	[nicht belegt]
DeviceID	[nicht belegt]
übrige Protokolldaten	s. [gemSpec_DM_ePA#A_14471]

Die nicht aufgelisteten Operationen der Schnittstelle I_Authentication_Insurant werden nicht protokolliert.
[<=]

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

4.5 Nichtfunktionale 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 Krankenversichertennummer (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 neunstellige 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.

5 Funktionsmerkmale

Die Komponente Authentisierung Versicherter realisiert ein Funktionsmerkmal über eine Schnittstelle:

Tabelle 4: Tab_Auth_Vers_005 - Schnittstellenübersicht der Komponente 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

Die Operation 'login' wird sowohl zur initialen Erstellung der Authentifizierungsbestätigung als auch nach Ablauf der Gültigkeit der ursprünglichen Authentifizierungsbestätigung zur Erstellung einer neuen Authentifizierungsbestätigung aufgerufen.
Die Operation 'renew' erstellt eine neue Authentifizierungsbestätigung, wenn eine gültige Authentifizierungsbestätigung vorgelegt wird, zu der noch kein 'logout' stattgefunden hat.
Die Operation 'logout' beendet die Erneuerbarkeit einer Authentifizierungsbestätigung.

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

Tabelle 5: 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]

5.1 Authentisierung

5.1.1 Schnittstellen

5.1.1.1 Schnittstelle I_Authentication_Insurant

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

A_14228 - Komponente Authentisierung Versicherter - I_Authentication_Insurant:login/renew/logout nach WS-Trust
Die Komponente "Authentisierung Versicherter" MUSS einen Webservice-Endpunkt AuthInsurantService bereitstellen, welcher die logischen Schnittstellen I_Authentication_Insurant:login, I_Authentication_Insurant:renew und I_Authentication_Insurant:logout durch die folgenden angebotenen Operationen realisiert:

Tabelle 6: Tab_Auth_Vers_007 - Schnittstellenübersicht der Authentisierung des 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) angegeben 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 Nachrichtenfelder 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>
```

```
</soap:Envelope>
```

LoginCreateToken, Request:

```
      <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
        <soap:Header>
          <wsse:Security xmlns:wsse="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
      wssecurity-utility-1.0.xsd" soap:mustUnderstand="true">
            <wsse:BinarySecurityToken EncodingType="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
      1.0#Base64Binary" ValueType="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
      wsu:Id="X509-c3b3a51c-a22b-4682-85a2-
      5537d56ba5e2">MIIEzTCCA7WgA...</wsse:BinarySecurityToken>
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      Id="SIG-f1f0472f-2f0d-468d-b425-0b1f5c78cc5a">
              <ds:SignedInfo>
                <ds:CanonicalizationMethod
      Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                  <ec:InclusiveNamespaces
      xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
      PrefixList="soap"/>
                </ds:CanonicalizationMethod>
                <ds:SignatureMethod
      Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
                <ds:Reference URI="#id-6c68f4bd-153d-42fb-a640-
      890c5cc14771">
                  <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-
      exc-c14n#"/>
                  </ds:Transforms>
                  <ds:DigestMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>9Et/DvvJ1Sb0Z1SEequKHmOYTEizKYCKZlAEiDILG
      FU=</ds:DigestValue>
                </ds:Reference>
              </ds:SignedInfo>
              <ds:SignatureValue>P21t+FT2tA...</ds:SignatureValue>
              <ds:KeyInfo Id="KI-bd93fc63-8828-46ad-8a6c-df08acabe5ce">
                <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
      wssecurity-utility-1.0.xsd" wsu:Id="STR-d16144ef-1a31-45b8-b061-
      537a93fbd515">
                  <wsse:Reference URI="#X509-c3b3a51c-a22b-4682-85a2-
      5537d56ba5e2" ValueType="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
      1.0#X509v3"/>
                </wsse:SecurityTokenReference>
              </ds:KeyInfo>
            </ds:Signature>
          </wsse:Security>

    <Action xmlns="http://www.w3.org/2005/08/addressing">http://docs.oasis-
open.org/ws-sx/ws-trust/200512/RSTR/ChallengeFinal</Action>
    <To
```

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

   <RequestSecurityTokenResponse xmlns="http://docs.oasis-open.org/ws-
sx/ws-trust/200512">
      <SignChallengeResponse>
        <Challenge>JemuBWS-...</Challenge>
      </SignChallengeResponse>
    </RequestSecurityTokenResponse>
  </soap:Body>
</soap:Envelope>
```

LoginCreateToken, 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/RSTRC/IssueFinal</Action>
    <To
xmlns="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addr
essing/anonymous</To>
  </soap:Header>

  <soap:Body>
    <RequestSecurityTokenResponseCollection xmlns="http://docs.oasis-
open.org/ws-sx/ws-trust/200512">
      <RequestSecurityTokenResponse>
        <RequestedSecurityToken>
          <saml2:Assertion ...> ...
          </saml2:Assertion>
        </RequestedSecurityToken>
      </RequestSecurityTokenResponse>
    </RequestSecurityTokenResponseCollection>
  </soap:Body>
</soap:Envelope>
```

Normative Festlegung zum Dialog

A_14053 - Komponente Authentisierung Versicherter - I_Authentication_Insurant:login nach WS-Trust, LoginCreateChallenge
Die Komponente "Authentisierung Versicherter" MUSS die Operation`LoginCreateChallenge` wie folgt anbieten:

Tabelle 7: Tab_Auth_Vers_008 - Signatur der Schnittstelle 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 Operation`LoginCreateToken` wie folgt anbieten:

Tabelle 8: 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			
/RequestSecurityToken ResponseCollection /RequestSecurityToken Response			
/RequestSecurityToken ResponseCollection /RequestSecurityToken Response /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_18985-02 - ePA-Client: Prüfen der AuthenticationAssertion
Ein ePA-Client (ePA-Frontend des Versicherten, ePA FM etc.) MUSS beim Erhalt des Authenticationtokens (AuthenticationAssertion) vergleichen, ob das eineindeutige Merkmal des Nutzers des ePA-Clients (KVNR des Versicherten oder Telematik-ID), der sich gegenüber dem Aktensystem authentifiziert hat, sich im saml2-NameID-Feld als Attribut 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"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
```

```
            <saml2:AttributeValue
xsi:type="xsd:string">Burgund</saml2:AttributeValue>
        </saml2:Attribute>
        <saml2:Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/country"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
            <saml2:AttributeValue
xsi:type="xsd:string">DE</saml2:AttributeValue>
        </saml2:Attribute>
        <saml2:Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
            <saml2:AttributeValue
xsi:type="xsd:string">X110474929</saml2:AttributeValue>
        </saml2:Attribute>
        <saml2:Attribute Name="urn:gematik:subject:subject-id"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
            <saml2:AttributeValue>
                <InstanceIdentifier xmlns="urn:hl7-org:v3"
extension="X110474929" root="1.2.276.0.76.4.8"/>
            </saml2:AttributeValue>
        </saml2:Attribute>
    </saml2:AttributeStatement>
</saml2:Assertion>
```

5.1.1.1.2 Operation renew

Die Operation dient der Erneuerung einer Authentifizierungsbestätigung.

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

Beispiel Dialog

RenewToken, 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/Renew</Action>
    <To xmlns="http://www.w3.org/2005/08/addressing">...</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/Renew</RequestType>
      <RenewTarget>... the token to be renewed ...</RenewTarget>
    </RequestSecurityToken>
  </soap:Body>
```

```
</soap:Envelope>
```

RenewToken, 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/RenewFinal</Action>
    <To xmlns="http://www.w3.org/2005/08/addressing">...</To>
  </soap:Header>



  <soap:Body>
    <RequestSecurityTokenResponse xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
      <RequestedSecurityToken> ... the new token ...
</RequestedSecurityToken>
    </RequestSecurityTokenResponse>
  </soap:Body>
</soap:Envelope>
```

A_17392 - Komponente Authentisierung Versicherter - I_Authentication_Insurant:renew nach WS-Trust, RenewToken
Die Komponente "Authentisierung Versicherter" MUSS die Operation renew wie folgt 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-1.1#SAMLV2.0		n

/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	Dieser Parameter MUSS die in Kap. 5.1.2.1.2 definierte SAML Assertion enthalten Die Signatur der Komponente Authentisierung Versicherterist in der SAML Assertion enthalten.		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.	
wst:UnableToRenew	The requested renewal failed	Das übergebene Token ist abgelaufen oder aus anderen Gründen nicht erneuerbar.	

[<=]

5.1.1.1.3 Operation logout

Die Operation beendet die Erneuerbarkeit einer Authentifizierungsbestätigung.

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

Beispiel Dialog

LogoutToken, 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/Cancel</Action>
    <To xmlns="http://www.w3.org/2005/08/addressing">...</To>
  </soap:Header>
  <soap:Body>
    <RequestSecurityToken xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
      <RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Cancel</RequestType>
      <CancelTarget>... the token to be cancelled ...</CancelTarget>
    </RequestSecurityToken>
  </soap:Body>
</soap:Envelope>
```

LogoutToken, Response:

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

A_17393-01 - Komponente Authentisierung Versicherter - I_Authentication_Insurant:Logout nach WS-Trust, LogoutToken
Die Komponente "Authentisierung Versicherter" MUSS die Operation Logout wie folgt 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.	

[<=]

5.1.1.1.4 Operation getAuditEvents

A_14477 - Komponente Authentisierung Versicherter - I_Authentication_Insurant::getAuditEvents

Die Komponente "Authentisierung Versicherter" MUSS die Operation`I_Authentication_Insurant::getAuditEvents` gemäß der folgenden Tabelle implementieren:

Tabelle 9: Tab_Auth_Vers_010 - Signatur der Schnittstelle 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 Authentifizierungsbestätigung.	SAML Assertion(im WSSE SOAP Header gem. [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.

[<=]

5.1.2 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 10: Tab_Auth_Vers_011 - Ablauf von loginCreateChallenge

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_14053 prüfen	Fehlende Elemente oder falsche Inhalte oder andere Fehler im empfangenen Request.	`wst:InvalidRequest`
Zufallswert für die Responsemessage gem. [gemSpec_Krypt#GS-A_4367] erzeugen	Zufallswert nicht verfügbar oder andere interne Verarbeitungsfehler.	`wst:RequestFailed`

[<=]

A_14229 - Komponente Authentisierung Versicherter - loginCreateToken, Ablauf

Das 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 11: Tab_Auth_Vers_012 - Ablauf von loginCreateToken

Aktion	Fehlerbedingung	Fehlermeldung
Validierung der Eingangsnachricht gegen die WSDL und die zugehörigen Schemadateien	Fehler bei der Validierung.	`wst:InvalidRequest` oder allgemeiner SOAP Fault
Prüfung WS-Security Header	Das Signaturzertifikat ist nicht vorhanden oder das Signaturverfahren entspricht nicht den Vorgaben von [gemSpec_Krypt].	`wst:InvalidRequest`
Prüfung mathematische Korrektheit der Signatur	Signatur nicht korrekt.	`wst:InvalidRequest`
Das Signaturzertifikat muss gemäß [gemSpec_PKI#TUC_PKI_018] geprüft werden. Parameter: • PolicyList: oid_egk_aut, oid_egk_aut_alt • intendedKeyUsage: digitalSignature • intendedExtendedKeyUsage: (leer) • OCSP-Graceperiod: 60 Minuten • Offline-Modus: nein • Prüfmodus: OCSP Eine Prüfung der vom TUC zurückgelieferten Rollen-OID ist nicht erforderlich.	Fehlermeldung des aufgerufenen TUC.	`wst:InvalidSecurityToken`
Eingangsparameter des SOAP Body entsprechend A_14059 prüfen	Fehlende Elemente oder falsche Inhalte oder andere Fehler.	`wst:InvalidRequest`

Challenge Element mit abgesendeter *Challenge* in Response zu `loginCreateChallenge` vergleichen	Challenges verschieden.	`wst:InvalidRequest`
AuthenticationAssertion (Token) gem. A_14109 erstellen und in Whitelist für Erneuerung aufnehmen (s. Kap. 5.1.3#A_17395)	Fehler in der internen Verarbeitung.	`wst:RequestFailed`

[<=]

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/xmlenc#sha256"/>
                <ds:DigestValue>TDtN2nJ05NUB1n18GL7AalUyuMVvrIHlEklGKXLho2o
=</ds:DigestValue>
            </ds:Reference>
        </ds:SignedInfo>        <ds:SignatureValue>aA4mAz3W2j7YWTKZmSXH2erR
5MtfzzOroWRLsy0wVwZdSsaK3MXW5pTnVjXE87Wq2dYJ3OFhu1QGGPWwz1qNxnmynBiWlfu21UZ
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 Musterkasse1GKVNOT-
VALID,C=DE</saml2:NameID>
        <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
    </saml2:Subject>
    <saml2:Conditions NotBefore="2018-09-20T11:29:19.884Z"
NotOnOrAfter="2018-09-20T11:44:19.884Z">
        <saml2:AudienceRestriction>
            <saml2:Audience>[ePA_TI_FQDN]</saml2:Audience>
        </saml2:AudienceRestriction>
    </saml2:Conditions>
    <saml2:AuthnStatement AuthnInstant="2018-09-20T11:29:19.878Z">
        <saml2:AuthnContext>

            <saml2:AuthnContextClassRef>

                urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI

            </saml2:AuthnContextClassRef>
        </saml2:AuthnContext>
    </saml2:AuthnStatement>
    <saml2:AttributeStatement>

          ...
        <saml2:Attribute Name="urn:gematik:subject:subject-id"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
            <saml2:AttributeValue>

               <InstanceIdentifier xmlns="urn:hl7-org:v3"
extension="G995030566" root="1.2.276.0.76.4.8"/>

            </saml2:AttributeValue>
        </saml2:Attribute>
    </saml2:AttributeStatement>
</saml2:Assertion>
```

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 [gemSpec_Aktensystem#A_14128] gesetzt werden: den TI-seitigen FQDN für TI-seitige Aufrufe der Schnittstelle I_Authentication_Insurant bzw. den Internet-seitigen FQDN für Internet-seitige Aufrufe der Schnittstelle I_Authentication_Insurant.
- Das Element *saml2:AuthnStatement/saml2: AuthnContext/saml2:AuthnContextClassRef* muss im Falle eines C.CH.AUT-Zertifikats auf den Wert "`urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI`" und im Falle eines C.CH.AUT_ALT-Zertifikats auf den Wert "`urn:oasis:names:tc:SAML:2.0:ac:classes:X509`" gesetzt werden

[<=]

A_15631 - Komponente Authentisierung Versicherter - Behauptungen in der Authentifizierungsbestätigung
Die Komponente "Authentisierung Versicherter" MUSS die für die Operation loginCreateToken erzeugte *Authentifizierungsbestätigung* im Element *AttributeStatement* mit den Behauptungen gemäß [gemSpec_TBAuth#TAB_TBAuth_02_2] befüllen und dabei folgende Vorgaben beachten:

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

[<=]

5.1.2.1.2 Operation Renew

A_17398 - Komponente Authentisierung Versicherter - RenewToken
Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der Operation`RenewToken` die folgenden Aktionen ausführen und bei den genannten Fehlerbedingungen die Fehlermeldungen (vgl. Kap. 5.1.1.1.2) entsprechend setzen:

Tabelle 12: 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 Authenfizierungsbestätigung 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 Operation`LogoutToken` die folgenden Aktionen ausführen und bei den genannten Fehlerbedingungen die Fehlermeldungen (vgl. Kap. 5.1.1.1.3) entsprechend setzen:

Tabelle 13: 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_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)

[<=]

5.1.2.1.4 Operation getAuditEvents

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_14781 - Komponente Authentisierung Versicherter - getAuditEvents, Prüfschritte
Die Komponente "Authentisierung Versicherter" MUSS beim Aufruf der Operation`getAuditEvents` 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 14: 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
Ein Prüfschritt der Inhaltsprüfung gem. [gemSpec_TBAuth#A_15558]/[gemSpec_Authentisierung_Vers#A_14780] bzw. [gemSpec_TBAuth#A_15637] liefert einen Fehler.	ASSERTION_INVALID

[<=]

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.

[<=]

5.1.3 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.

6 Informationsmodell

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

7 Verteilungssicht

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

8 Anhang A – Verzeichnisse

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 11
Tabelle 2: Tab_Auth_Vers_003 - Zuordnung Fehlercodes zu Fehlernamen 12
Tabelle 3: Tab_Auth_Vers_004 - Operationsabhängige Parameter des Verwaltungsprotokolls 13
Tabelle 4: Tab_Auth_Vers_005 - Schnittstellenübersicht der Komponente Authentisierung des Versicherten 15
Tabelle 5: Tab_Auth_Vers_006 - Benutzte Schnittstellen der TI 15
Tabelle 6: Tab_Auth_Vers_007 - Schnittstellenübersicht der Authentisierung des Versicherten 16
Tabelle 7: Tab_Auth_Vers_008 - Signatur der Schnittstelle I_Authentication_Insurant:loginCreateChallenge 20
Tabelle 8: Tab_Auth_Vers_009 - Signatur der Schnittstelle I_Authentication_Insurant:loginCreateToken 22
Tabelle 9: Tab_Auth_Vers_010 - Signatur der Schnittstelle I_Authentication_Insurant::getAuditEvents 31
Tabelle 10: Tab_Auth_Vers_011 - Ablauf von loginCreateChallenge 32
Tabelle 11: Tab_Auth_Vers_012 - Ablauf von loginCreateToken 33
Tabelle 12: Tab_Auth_Vers_015 - Ablauf von RenewToken 36
Tabelle 13: Tab_Auth_Vers_015 - Ablauf von RenewToken 38
Tabelle 14: Tab_Auth_Vers_013 - Prüfschritte bei getAuditEvents 38

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

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 1.1Binding 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