Elektronische Gesundheitskarte und Telematikinfrastruktur

Errata 3 zum Konnektor PTV1 (VSDM) und PTV2 Online-Produktivbetrieb (Stufe 1)

Version:	1.0.0
Stand:	18.05.2018
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemErrata_3_Kon_PTV1_PTV2]

Betroffene Produkttypen
gemProdT_Kon_PTV1
gemProdT_Kon_PTV2

ID	Dokument	Quelle	Beschreibung der Änderung	**Anpassung** an Afos; TUCs, Tabellen, Korrekturen	**von Änderung betroffene Dokumente**
C_6326	gemSpec_PKI gemSpec_TSL gemSpec_OCSP-Proxy gemKPT_PKI_TIP		**Änderung - TSL ohne QES-Zertifikate / dadurch Anpassungen an QES-Zertifikatsprüfung** Im Test wurden Fehlersituationen bei TI-Produkten erkannt, die kein QES auswerten sollen bzw. können, ausgelöst durch nicht-RSA-QES-CA-Zertifikate in der TSL. Um diese Fehler zukünftig zu vermeiden, werden QES-CA-Elemente aus der TSL entfernt. Diese waren dort nur vor Inkrafttreten von eIDAS notwendig, werden aber nicht mehr benötigt, da sie in der BNetzA-VL enthalten sind. In der TSL verbleiben (allerdings an anderer Stelle) für die zugelassenen VDAs die Adress-Informationen zu deren OCSP-Respondern in der TI. Die Entfernung von QES-Zertifikaten in der TSL erfordert geringfügige Anpassungen der QES-Zertifikatsprüfungs-Mechanismen für QES-Konnektoren und stützt sich damit vollständig auf die Einträge in der Vertrauensliste der BNetzA, die im Zuge der TI-Anpassungen für eIDAS (Rel. 1.6.3) bereits für Konnektoren in der TI verfügbar gemacht wurde. Die klarere Trennung der Vertrauensdomänen (TSL für nonQES / BNetzA-VL für QES) ist dabei das Ziel, dass Sicherheit und Architektur verfolgen und zukunftsweisend umgesetzt werden kann, solange noch kein QES-Konnektor im Feld umgesetzt ist. Ohne die Anpassung würde es weiterhin ein Risiko bezüglich Auswertung der TSL bei nonQES-TI-Produkten geben (kompletter Abbruch bei Fehlern bei der TSL-Auswertung).	siehe C_6326_Anlage	gemSpec_PKI gemSpec_TSL gemSpec_OCSP-Proxy gemKPT_PKI_TIP gemProdT_Kon_PTV2
C_6346	gemSpec_Kon_SigProxy	TIP1-A_4662 TIP1-A_4663	**SigProxy: Fehlercode für Abbruch durch Benutzer** Für den Fall, dass der Benutzer am Signaturproxy einen Signaturauftrag im Bestätigungsmodus während der dafür konfigurierten Zeitspanne nicht freigibt oder einen Signaturvorgang abbricht, ist nicht spezifiziert, welchen Fehler der Signaturproxy an das aufrufende Clientsystem zurückgibt. Für den Fall, dass der Benutzer am Signaturproxy das Prüfergebnis für einen Signaturprüfauftrag im Ansichtsmodus während der dafür konfigurierten Zeitspanne nicht freigibt (TIP1-A_5701: Bewertung von INCONCLUSIVE), ist nicht spezifiziert, welchen Fehler der Signaturproxy an das aufrufende Clientsystem zurückgibt. Um Interoperabilität für diesen Fall zu gewährleisten, sind die Fehlercodes festzulegen.	siehe C_6346_Anlage	gemSpec_Kon_SigProxy gemProdT_Kon_PTV2
C_6358	SignatureService.xsd		**Optimierung - SignatureService.xsd - Typprüfung für Timestamp** In der Antwort auf einen Aufruf der Operation VerifyDocument wird ein Element Timestamp zurückgegeben. Für dieses Element fehlt im XML-Schema SignatureService.xsd der Typ. Im FAQ Q_0389 wurde der Typ festgelegt. Der so festgelegte Typ wird nun im XML-Schema festgeschrieben.	Änderung in SignatureService.xsd: alt …version="7.4.1" … <element name="Timestamp"/> neu … version="7.4.2" … <element name="Timestamp" type="dateTime"/> Änderung in gemSpec_Kon: alt: XSD Name SignatureService.xsd XSD Schemaversion 7.4.1 neu: XSD Name SignatureService.xsd XSD Schemaversion 7.4.2	SignatureService.xsd gemSpec_Kon gemProdT_Kon_PTV2

© gematik – öffentlich

C_6359	TIFF_25MB gemSpec_Kon_SigProxy		**Korrektur - Referenzdokument zur Performance TIFF_25MB** Die Datei TIFF_25MB ist konform zu DF_BV_TIFF intendiert, wobei DF_BV_TIFF das Dokumentenformat für die vollständige Anzeige von TIFF-Dokumenten im Signaturproxy ist. Anhang C2 in gemSpec_Kon_SigProxy definiert folgende Eigenschaft zu DF_BV_TIFF: "das Dokument enthält keine optionalen Felder". Das Referenzdokument zur Performance TIFF_25MB_V1.0.1.tif enthält aber das private tag "ICCProfile", was nicht konform zu DF_BV_TIFF ist. Es ist genauer zu definieren, was mit "das Dokument enthält keine optionalen Felder" gemeint ist und die Datei TIFF_25MB ist anzupassen.	Änderung: TIFF_25MB_V1.0.1.tif wird durch TIFF_25MB_V1.0.2.tif ohne das Tag "ICCProfile" ersetzt. Anhang C2 in gemSpec_Kon_SigProxy wird geändert: alt: das Dokument enthält keine optionalen Felder neu: das Dokument enthält ausschließlich solche Tags, die in [TIFF6], Part 1: Baseline TIFF, aufgeführt sind.	TIFF_25MB gemSpec_Kon_SigProxy gemProdT_Kon_PTV2
C_6369	gemSpec_Perf		**Optimierung der Performance-Angaben für VZD** Bisher fließen in der Schnittstellenoperation I_Directory_Query::search_Directory nicht die Bearbeitungszeiten für den TLS-Aufbau zwischen Konnektor und VZD ein. Hierbei handelt es sich um einen fachlichen Fehler, der behoben werden muss.	siehe C_6369_Anlage	gemSpec_Perf gemProdT_Kon_PTV2
C_6388	gemSpec_Kon gemSpec_HBA_ObjSys_G2.1	TIP1-A_4646 TIP1-A_4649 TIP1-A_4672 TIP1-A_5010 TIP1-A_4696 TIP1-A_4700	**Behandlung Attributzertifikate entfernen** Der Bedarf für die Behandlung von Attributzertifikaten besteht nicht mehr. Daher wird die Behandlung von Attributzertifikaten aus den Konzepten und Spezifikationen entfernt.	Für die Änderungen an anderen Dokumenten siehe C_6388_Anlage.	gemKPT_Arch_TIP gemSpec_Kon gemSpec_HBA_ObjSys_G2.1 gemProdT_Kon_PTV2 gemILF_PS
C_6434	gemSpec_Kon	TUC_KON_031	**Korrektur: Inkonsistenz - Fehlernummern im Konnektor vertauscht** Die Fehlernummern 4133 und 4129 werden in den Tabellen TAB_KON_618 und TAB_KON_619 jeweils vertauscht verwendet.	in Tabelle TAB_KON_618 werden die Nummern 4133 und 4129 vertauscht.	gemSpec_Kon
C_6438	gemSpec_Kon	TIP1-A_4644 TIP1-A_5010 TIP1-A_5667	**Änderung: Eindeutigkeit einer Jobnummer** TIP1-A_4644 fordert dass eine Jobnummer über 1000 Vorgänge eindeutig sein muss. Es ist hierbei nicht genau ersichtlich zu welchem Zeitpunkt diese Eindeutigkeit sichergestellt werden muss. Der Konnektor generiert Jobnummern. Clientsysteme können aber auch Jobnummern bei der Signaturerstellung an den Konnektor übergeben, die nicht vom Konnektor erzeugt wurden. Dies wird präzisiert. Es wird ein gematik-Fehler definiert, für den Fall, dass eine Jobnummer nicht eindeutig ist. Für den Fall dass Clientsysteme nicht den Signaturproxy nutzen, müssen Clientsysteme die Forderung, dass eine Jobnummer über 1000 Vorgänge eindeutig sein muss, selbst umsetzen. Hierfür werden Anforderungen an Clientsysteme spezifiziert.	siehe C_6438_Anlage	gemSpec_Kon gemILF_PS.doc gemProdT_Kon_PTV2
C_6453	gemSpec_Kon	TUC_KON_150	**Korrektur der Nummerierung in TUC_KON_150** Die Nummerierung und dazugehörige Referenzierung beim Standardablauf des TUC_KON_150 sind fehlerhaft.	Siehe C_6453_Anlage	gemSpec_Kon gemProdT_Kon_PTV2

© gematik – öffentlich

C_6460	gemSpec_Kon	TIP1-A_4654 TIP1-A_4672	**Präzisierung Status VerificationResult pro Signatur** Im Rahmen der Signaturprüfung für ein Dokument wird ein übergreifender Statuswert (VALID/INVALID/INCONCLUSIVE) aus den Statuswerten (VALID/INVALID/INCONCLUSIVE) für jede einzelne Signatur auf dem Dokument ermittelt. Der Statuswert zu jeder einzelnen Signatur korrespondiert zu den Werten resultmajor/resultminor innerhalb des VerifikationReports zu den einzelnen Signaturen. Hier besteht Präzisierungsbedarf. Zum einen bestehen Lücken, z.B. ist die Behandlung von Manifests nicht berücksichtigt. Zum anderen sind einzelne Punkte nicht klar definiert, so die Wirkung einer nicht durchführbaren OCSP-Abfrage.	siehe C_6460_Anlage	gemSpec_Kon gemProdT_Kon_PTV2
C_6490	gemSpec_Kon EventService.wsdl EventService.xsd	Kap.3.5.3 Kap.3.3.3 4.1.6.5 4.1.6.5.8	**Verzicht der Funktionalität CORS** Für die Pflege der Basisdaten im Verzeichnisdienst (VZD) wird nicht mehr die Lösung verfolgt, dass der VZD eine Webanwendung zur Verfügung stellt. Da es gegenwärtig keine Anwendungen gibt, die die Funktionalität CORS benötigen, gibt es auch nicht den Bedarf CORS im Konnektor zu unterstützen. Deshalb wird die Funktionalität CORS vollständig aus gemSpec_Kon ausgebaut.	siehe C_6490_Anlage	gemSpec_Kon gemProdT_Kon_PTV2

© gematik – öffentlich

Änderungsbedarf:

Im Test wurden Fehlersituationen bei TI-Produkten erkannt, die kein QES auswerten sollen bzw. können, ausgelöst durch nicht-RSA-QES-CA-Zertifikate in der TSL.

Um diese Fehler zukünftig zu vermeiden, werden QES-CA-Elemente aus der TSL entfernt. Diese waren dort nur vor Inkrafttreten von eIDAS notwendig, werden aber nicht mehr benötigt, da sie in der BNetzA-VL enthalten sind. In der TSL verbleiben (allerdings an anderer Stelle) für die zugelassenen VDAs die Adress-Informationen zu deren OCSP-Respondern in der TI.

Die Entfernung von QES-Zertifikaten in der TSL erfordert geringfügige Anpassungen der QES-Zertifikatsprüfungs-Mechanismen für QES-Konnektoren und stützt sich damit vollständig auf die Einträge in der Vertrauensliste der BNetzA, die im Zuge der TI-Anpassungen für eIDAS (Rel. 1.6.3) bereits für Konnektoren in der TI verfügbar gemacht wurde.

Die klarere Trennung der Vertrauensdomänen (TSL für nonQES / BNetzA-VL für QES) ist dabei das Ziel, dass Sicherheit und Architektur verfolgen und zukunftsweisend umgesetzt werden kann, solange noch kein QES-Konnektor im Feld umgesetzt ist.

Diese Änderung verhindert das Risiko, dass zukünftige QES-CA-Zertifikate, die von VDAs in die TSL eingebracht werden, zu Fehlern bei anderen Systemen in der TI führen. Hintergrund: Verwendete Signatur-Algorithmen können für QES-CA-Zertifikate der VDAs nicht von der gematik vorgegeben werden und sind in der BNetzA-VL enthalten.

QES-Konnektoren werden vorbereitet, um mit allen QES-CA-Zertifikaten in der BNetzA-VL umgehen zu können.

Die Änderungen berücksichtigen ferner die weggefallene Notwendigkeit, Attributzertifikate bei Zertifikatsprüfungen zu berücksichtigen. Siehe als Referenz dazu auch den Änderungseintrag C_6388.

Die folgenden Spezifkation werden dahingehend angepasst:

- gemSpec_PKI
- gemSpec_TSL
- gemSpec_OCSP_Proxy
- gemKPT_PKI_TIP

Änderungen in gemSpec_PKI

[…]

~~3.5 Sperrungen von QES-CAs~~

~~Im Falle von sicherheitskritischen Incidents bei QES-CA-Zertifikaten (s. Definition und Beschreibung in [gemKPT_PKI_TIP#2.3.3.5]) müssen innerhalb der TI besondere Maßnahmen ergriffen werden, wenn die betroffene QES-CA Zertifikate für die TI ausstellt und durch die BNetzA gesperrt wurde.~~

~~⊠ GS-A_5065 Sperrung einer QES-CA~~

~~Der Gesamtbetriebsverantwortliche der TI MUSS im Falle der Sperrung einer QES-CA in Kooperation mit der BNetzA die erforderlichen Maßnahmen für die TI festlegen und den TSL-Dienst entsprechend anweisen.~~ ~~⊠~~

[…]

8 Prüfung von Zertifikaten

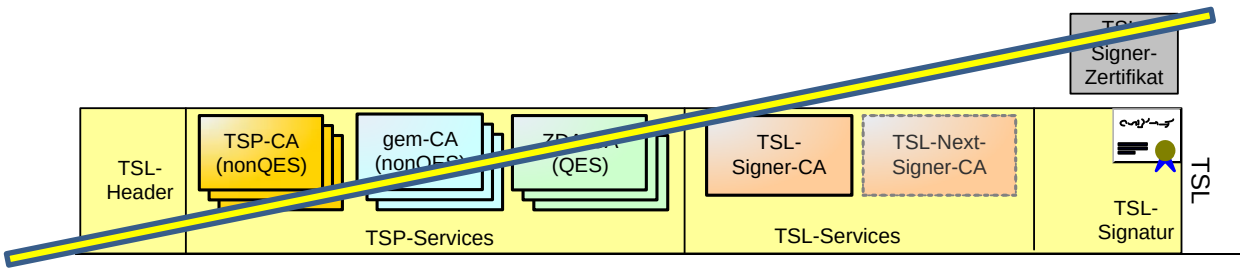
Für die Nutzung und Statusprüfung von Zertifikaten in der TI gilt:

- Das TSL-Signer-CA-Zertifikat bildet den Vertrauensanker für die TI.
- Die TSL stellt (i. S. einer Whitelist) den Vertrauensraum für die in der TI zugelassenen Aussteller-CA.
- nonQES-Aussteller-CA-Zertifikate werden ausschließlich gegen die TSL geprüft
- QES-Aussteller-CA-Zertifikate werden
  ~~(a) hinsichtlich ihrer Zulassung in der TI gegen die TSL und~~
  ~~(b)~~ hinsichtlich ihres VDA Qualifikationsstatus gemäß [eIDAS] gegen die BNetzA-VL geprüft.
- Als Vertrauensanker für die BNetzA-VL fungieren jeweils die aktuell publizierten BNetzA-VL-Signer-Zertifikate. Diese werden mittels TSL in die QES-prüfenden Systeme (Konnektoren) eingebracht und aktualisiert.
- End-Entity-Zertifikate werden gegen den OCSP-Dienst der Aussteller-CA geprüft, außer die Statusprüfung für einen bestimmten Zertifikatstyp ist explizit optional oder nicht vorgesehen.

[…]

8.1 Vertrauensraum der TI

[…]



~~Abbildung 6: Aufbau der TSL~~

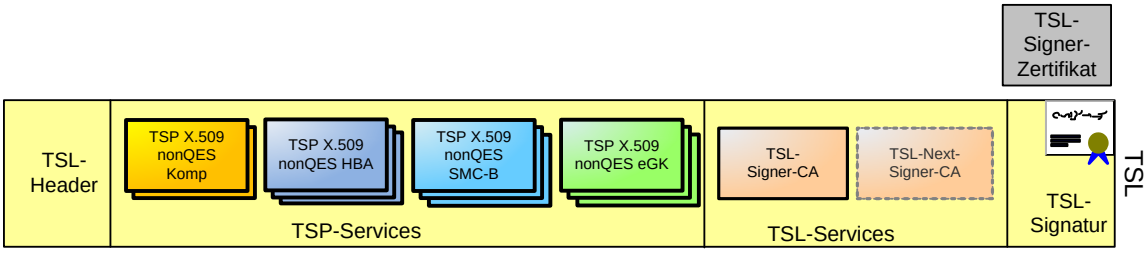


Abbildung 6: Aufbau der TSL

[…]

8.5 Zertifikatsprüfung X.509 QES

[…]

⌦ **GS-A_4750 TUC_PKI_030 „QES-Zertifikatsprüfung“**

Alle Produkttypen, die QES-Zertifikate prüfen, MÜSSEN TUC_PKI_030 zur Prüfung der QES-Zertifikate umsetzen. ⌫

8.5.1 TUC_PKI_030 „QES-Zertifikatsprüfung“

Tabelle 101: TUC_PKI_030 „QES-Zertifikatsprüfung“

Element	Beschreibung
Name	TUC_PKI_030 „QES-Zertifikatsprüfung“
Beschreibung	In diesem Use Case wird die Prüfung von Zertifikaten mit qualifizierter Signatur beschrieben. Die Prüfung von QES-Zertifikaten setzt sich aus den in [Common-PKI#Part5] und [Common-PKI#9 beschriebenen Schritten zusammen], sofern sie den Vorgaben von [eIDAS] nicht widersprechen. ~~Die Prüfungen der korrespondierenden Attributzertifikate (falls übergeben) müssen die Vorgaben gemäß [baekAttr], [RFC5755], [Recommendation ITU-T X.509 | ISO/IEC 9594-8] und [Common-PKI#Part9#2.2] berücksichtigen.~~ Zusätzlich werden folgende Schritte in diesem Technical Use Case (TUC) durchgeführt.
Anwendungsumfeld	System, das Zertifikate verwendet
Vorbedingungen	aktuelle TSL-Informationen im Truststore (inkl. OCSP-Adressen in der TI für die zugelassenen VDAs), eine aktuell gültige BNetzA-VL.
Auslöser	Zertifikats-Check
Eingangsdaten	• QES-Zertifikat • ~~Liste der korrespondierenden Attributzertifikate (optional, zum QES-Zertifikat als Basis-Zertifikat)~~ • Referenzzeitpunkt (refTime): Zeitpunkt, für den das Zertifikat geprüft werden soll • Offline-Modus (ja/nein)

Element	Beschreibung
	• Beigefügte OCSP-Response, die zur Prüfung des angefragten QES-Zertifikates erforderlich ist (optional; z. B. in Signatur eingebettet) • Nonce (optional; Wert ausschließlich zur Verwendung bei der OCSP-Prüfung des zu prüfenden QES-Zertifikates) • ~~TSL-Informationen (Adressen für OCSP-Abfragen)~~ • Timeout-Parameter für OCSP-Abfragen (Default: 10s)
Komponenten	System
Ausgangsdaten	• Status der Prüfung~~,~~ • OCSP-Response zum angefragten QES-Zertifikat~~,~~ • im Zertifikat enthaltene Rollen-OIDs • im Zertifikat enthaltene QCStatements-Einträge
Standardablauf	~~0.~~ ~~Die QES-Zertifikatsprüfung setzt sich aus den in [Common-PKI#Part5] und [Common-PKI#9] beschriebenen Schritten zusammen.~~ ~~Die Prüfungen der korrespondierenden Attributzertifikate müssen die Vorgaben gemäß [baekAttr], [RFC5755] und [Recommendation ITU-T X.509 | ISO/IEC 9594-8] berücksichtigen.~~ ~~(Zusätzlich zu den in [Common-PKI] beschriebenen Schritten werden folgende Schritte durchgeführt:~~ 1. [System] Auslesen und Ausgabe aller gesetzten Elemente der Extension QCStatements des Zertifikates ~~1~~ 2. [System:] Anhand der End-Entity-Zertifikatsdaten wird die BNetzA-VL durchsucht, um das passende QES-CA-Zertifikat zu finden. *Hinweis: Das Verfahren zum Finden des QES-CA-Zertifikates in BNetzA-VL verläuft analog zum Finden des nonQES-CA-Zertifikates in der TSL mittels TUC_PKI_003.* ~~1~~ 3. [System:] Prüfung, ob das ausstellende QES-CA-Zertifikat für die QES-Prüfung zum Referenzzeitpunkt in der BNetzA-VL gemäß [eIDAS] und [ETSI TS 119 612#5.5.4 und #Annex J] qualifiziert und als gültig gekennzeichnet ist. *Hinweis: Für gültige Status siehe Anmerkungszeile zu diesem TUC* ~~2.~~ ~~[System:] Prüfung, ob das ausstellende QES-CA-Zertifikat des QES-Zertifikates in der TSL enthalten ist (TUC_PKI_003 "CA-Zertifikat in TSL-Informationen finden").~~ ~~3.~~ ~~[System:] Prüfung, ob das QES-CA-Zertifikat (zum Referenzzeitpunkt) in der TSL als gültig gekennzeichnet ist.~~ 4. [System:] Ermittlung der OCSP-Adresse aus ~~der TSL (TUC_PKI_005 "Adresse für Status- und Sperrprüfung ermitteln")~~ dem AIA-Feld des QES-EE-Zertifikates. Dabei handelt es sich um eine öffentlich aufrufbare URL im Internet. Wird für die ermittelte OCSP-URL in der TSL derselbe Wert im InformationValue-Element von AdditionalServiceInformation von BNetzA-VL-Service (mit ServiceTypeIdentifier http://uri.telematik/TrstSvc/Svctype/TrustedList/schemerules/DE) gefunden, so wird die dahinter folgende (nach Leerzeichen) URL als

Element	Beschreibung
	Adresse für die OCSP-Anfrage verwendet. Andernfalls wird die zuvor ermittelte OCSP-Adresse aus dem AIA-Feld für die OCSP-Anfrage verwendet. *Hinweis: Details zu den TSL-Einträgen für URLs für OCSP-Responder in der TI unter gemSpec_TSL# TIP1-A_7219* 5. [System:] Die abzufragenden Statusinformationen zu QES-Zertifikaten werden unter Verwendung der ~~extrahierten~~ ermittelten OCSP-Adressen~~n~~ eingeholt. *Hinweis: Details zur OCSP-Statusprüfung siehe Anmerkungszeile zu diesem TUC* 6. [System:] Ermittlung der Rolle (TUC_PKI_009 "Rollenermittlung") 7. [System:] Ende des Use Case mit Rückgabe des/der im Zertifikat enthaltenen Rollen-OID(s)
Varianten/Alternativen	Der Standardablauf stellt die üblichen Schritte dar, die durchgeführt werden müssen. Eine Trennung in zwei Prozesse oder eine Umstrukturierung, bei der alle notwendigen Schritte erfolgen, ist zulässig. 4a. [System:] Der Offline-Modus ist aktiviert. Es werden keine Statusinformationen eingeholt. (Schritte 4 und 5 entfallen.) ~~0.~~ 5a. [System:] Wird im optionalen Parameter Nonce ein Wert übergeben, dann muss für QES-Zertifikate dieser Wert als OCSP-Parameter in den OCSP-Request integriert und im Response geprüft werden. ~~4a. 5b.~~ ~~[System:] Der Offline-Modus ist aktiviert. Es werden keine Statusinformationen eingeholt. (Schritte 4 und 5 entfallen.)~~ 5~~a~~ c. [System:] Eine OCSP-Response zu dem zu prüfenden Zertifikat wurde im Aufruf mit übergeben. Falls dieses zum Referenzzeitpunkt gültig ist, werden keine OCSP-Requests erzeugt, sondern die beigefügte OCSP-Response zur weiteren Prüfung verwendet.
Fehlerfälle	In jedem der beschriebenen Schritte können Fehler auftreten. Diese sind durch das System zu melden und der Prozess muss beendet werden. ~~0b.~~ ~~In [Common-PKI#Part5] und [Common-PKI#9] sind Fehlerfälle für die QES-Zertifikatsprüfung und die Prüfung von Attributzertifikaten beschrieben. Diese beinhalten:~~ ~~- Prüfung von QCStatements: QC_STATEMENT_ERROR~~ ~~- Prüfung von QES-Basiszertifikat assoziierten Attributzertifikaten: ATTR_CERT_MISMATCH.~~ ~~0c.~~ ~~Wenn die in einer OCSP-Response zurückgelieferte Nonce nicht mit der Nonce des OCSP-Requests für ein QES-Zertifikat übereinstimmt, wird die Prüfung abgebrochen mit der Fehlermeldung OCSP_NONCE_MISMATCH.~~ 1a. Ist die Extension QCStatements nicht auslesbar, leer oder enthält keine auslesbaren Elemente, bricht der TUC mit dem Fehler QC_STATEMENT_ERROR ab ~~1~~ 3a.

Element	Beschreibung
	Ist das QES-CA-Zertifikat in der BNetzA-VL nicht vorhanden oder zum Referenzzeitpunkt nicht mit einem gültigen Status gekennzeichnet, muss der TUC mit einer Fehlermeldung CA_CERTIFICATE_NOT_QES_QUALIFIED abbrechen. 3~~a~~ b. [System:] QES-CA-Zertifikat des QES-Zertifikates ist in der ~~TSL~~ BNetzA-VL als revoked gekennzeichnet und QES-Zertifikat ist nach Sperrzeitpunkt erstellt worden. Abbruch mit Fehlermeldung (CA_CERTIFICATE_REVOKED_IN_~~TSL~~BNETZA-VL). 4a. [System:] Warnmeldung, dass keine Online-Statusprüfung durchgeführt wurde (NO_OCSP_CHECK). 5b. [System:]. Der zuständige OCSP-Responder ist nicht erreichbar. Abbruch mit Fehlermeldung (OCSP_NOT_AVAILABLE). 5c. [System:] OCSP-Responses zu dem zu prüfenden Zertifikat wurden im Aufruf mit übergeben, ergaben bei den weiteren Prüfschritten jedoch kein gültiges Ergebnis. Eine erneute Prüfung wird in diesem Fall durchgeführt, als wären keine OCSP-Responses beigefügt. In den Rückgabewerten dieses TUC wird die Warnmeldung (PROVIDED_OCSP_RESPONSE_NOT_VALID) an die aufrufende Funktion übergeben. ~~5d.~~ ~~[System:] Warnmeldung, dass keine Online-Statusprüfung durchgeführt wurde (NO_OCSP_CHECK).~~ 5e. Wenn die in einer OCSP-Response zurückgelieferte Nonce nicht mit der Nonce des OCSP-Requests (vgl. optionalen Eingangsparameter „Nonce“) für ein QES-Zertifikat übereinstimmt, wird die Prüfung abgebrochen mit der Fehlermeldung OCSP_NONCE_MISMATCH.
Technische Fehlermeldung	Das System meldet entsprechende Fehlercodes.
Sicherheits-anforderungen	
Anmerkungen	Gültige Status zu Schritt 1 sind gemäß [ETSI TS 119 612#5.5.4 und #Annex J] *granted, accredited, undersupervision, supervisionincessation* Die Einträge der QES-CA-Zertifikate in der BNetzA-VL besitzen gemäß [ETSI TS 119 612#5.5.1.1] die Extension AdditionalServiceInformation http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures. Die Einträge der QES-CA-Zertifikate in der TSL und der BNetzA-VL besitzen den ServiceTypeIdentifier http://uri.etsi.org/TrstSvc/Svctype/CA/QC. Schritt 2 stellt eine TI-spezifische Sperrprüfung des QES-CA-Zertifikats gemäß Kettenmodell dar. ~~Vgl. Kap. 8.1.1 Initialisierung TI-Vertrauensraum.~~ ~~Schritt 5 enthält die QES-spezifischen Statusprüfungen.~~ Zusätzlich zu den Vorgaben gemäß [eIDAS#Artikel 24, Abs. (2) Buchstabe (k), Abs. (3) und (4)] muss Schritt 5 folgende Anforderungen bei der QES-spezifischen Statusprüfungen erfüllen: • Zur Auswertung der OCSP-Response siehe auch [Common-PKI#Part4#3 und #Part9#4] • Zur Prüfung der certHash-Erweiterung siehe auch [Common-PKI#Part4#3.1.2] und [Common-PKI#Part5#2.3] sowie [gemSpec_Krypt#GS-A_4393] • Zur Prüfung der OCSP-Response auf Integrität (Signatur): Das

Element	Beschreibung
	OCSP-Signer-Zertifikat wird gemäß RFC 6960 durch die ausstellende QES-CA bereitgestellt
Zugehörige Diagramme/Tabelle	

[…]

Tabelle 104: Tab_PKI_274 Fehlercodes des SubCompTyps PKI bei TSL- und Zertifikatsprüfung

Code	Severity	ErrorType	ErrorText	Detail	Meldungskürzel
…					
1062	Error	Security	Das QES-EE-Zertifikat ist ungültig. Es wurde nach der Sperrung der ausgebenden QES-CA ausgestellt		CA_CERTIFICATE_REVOKED_IN_BNETZA_VL
…					

9.1.1 Schnittstelle I_OCSP_Status_Information

[…]

9.1.1.2 Umsetzung

[…]

⌦ GS-A_5517 Schlüsselgenerationen der OCSP-Signer-Zertifikate

Die Produkttypen TSL-Dienst, gematik Root-CA~~, TSP-X.509 QES~~ und TSP-X.509 nonQES MÜSSEN sicherstellen, dass zum Signieren von OCSP-Responses für Zertifikate einer bestimmten Schlüsselgeneration, ausschliesslich ein OCSP-Signer-Zertifikat derselben Schlüsselgeneration (gemäß [gemSpec_Krypt#GS-A_4357] bzw. [gemSpec_Krypt#GS-A_4358]) verwendet wird.⌫

[…]

B5.2 – Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
…	
~~[baekAttr]~~	~~Zertifikatsprofile für X.509 Attributzertifikate; Version 2.3.1 vom 29.05.2009~~
~~[BÄK_ZPX.509B]~~	~~Bundesärztekammer (28.05.2009):~~ ~~Zertifikatsprofile für X.509 Basiszertifikate V 2.3.1; Zertifikatsaufbau und –hierarchie, Gültigkeitsmodell für Zertifikatstypen: ENC, AUT, QES, ATT sowie die Root, Cross- und CA-Zertifikate, die CRL-Signer und die OCSP-Zertifikate~~
…	

[…]

Änderungen in gemKPT_PKI_TIP

[…]

2.3.1 Vertrauensmodell für QES

[…]
~~Außerdem werden die TSP-X.509 QES mit ihren zugelassenen Diensten in den Vertrauensraum der TI aufgenommen~~**~~Fehler! Textmarke nicht definiert.~~**.

[…]

~~⌦ TIP1-A_2035 Bereitstellung QES-CAs durch TSL-Dienst~~

~~Der TSL-Dienst MUSS im Kontext der QES die CA-Zertifikate der VDA für QES in der TSL bereitstellen.⌫~~

2.3.3 Vertrauensraum mittels TSL – Umsetzung in der TI

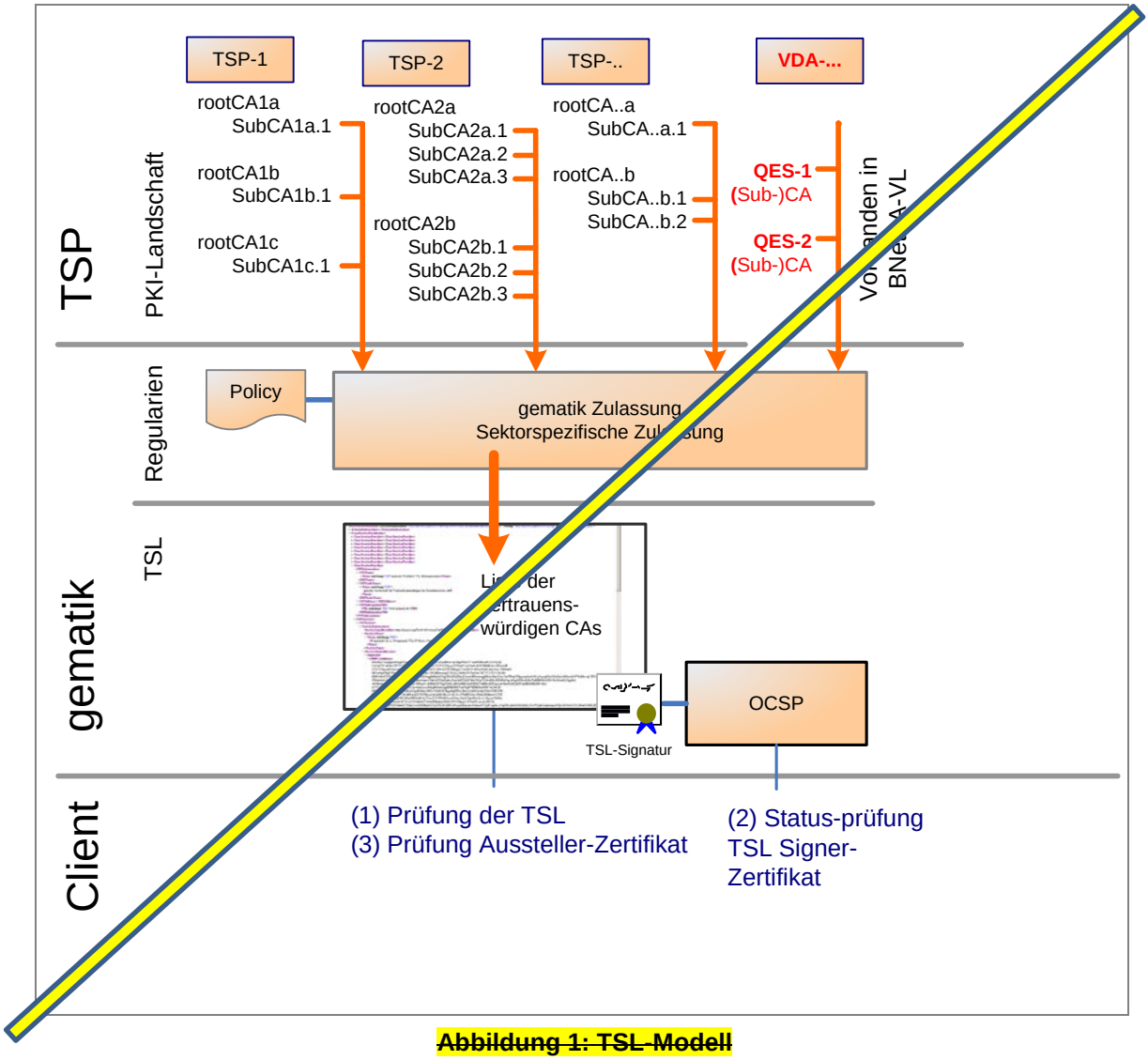
[…]

~~Die Aufnahme eines QES-CA-Zertifikates in die TSL repräsentiert die TI-Zulassung dieses VDA, um QES-Zertifikate für HBA in der TI für einen oder mehrere Sektoren produzieren zu dürfen. Für die Nutzung von QES-Zertifikaten außerhalb der TI hat die TSL keine Relevanz.~~

[…]

2.3.3.2 Struktur, Signatur und Inhalt der TSL

Die folgende Abbildung verdeutlicht die Anwendung der TSL in der TI.



~~Abbildung 1: TSL-Modell~~

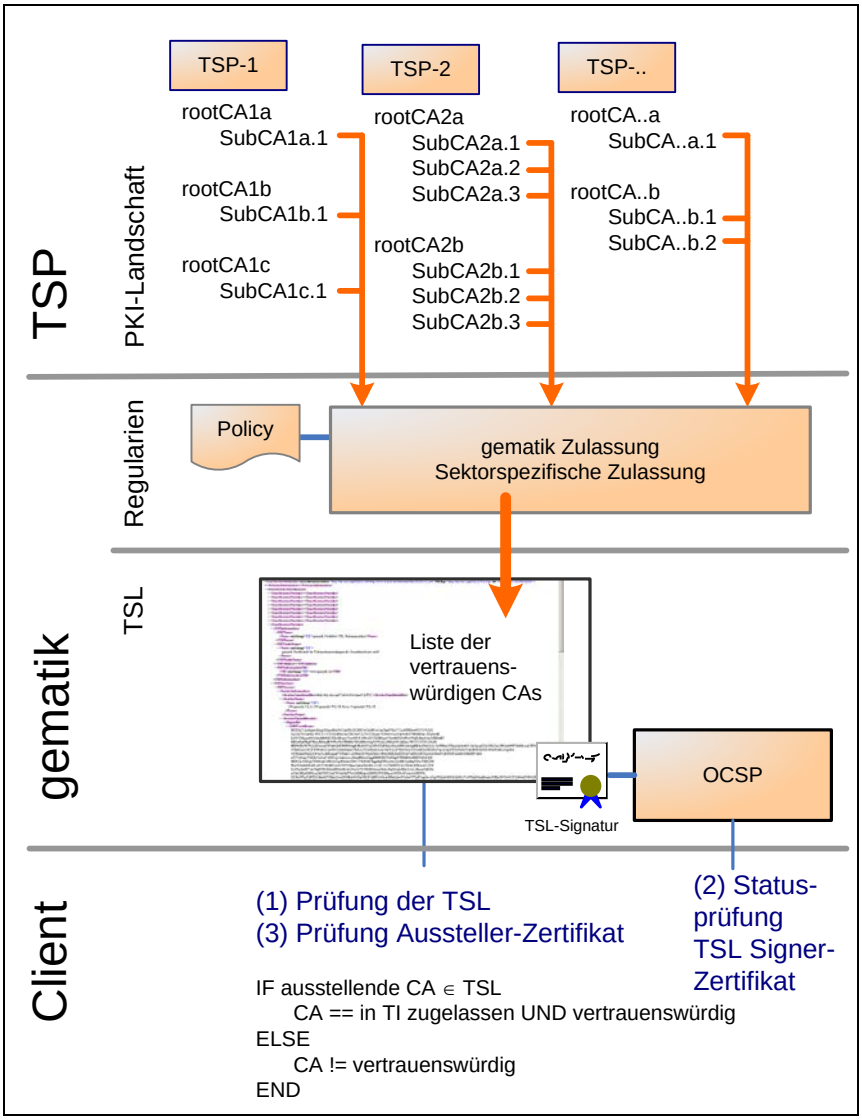
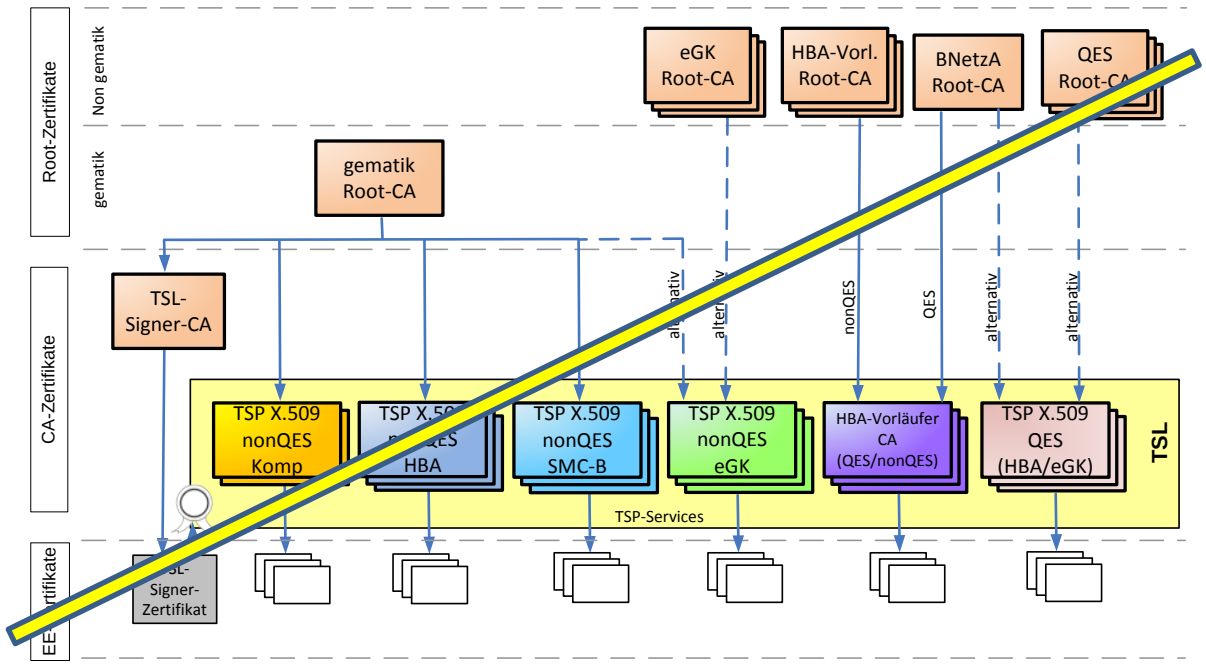


Abbildung 2: TSL-Modell

~~**Abbildung 3**~~ zeigt die grundsätzlichen Hierarchiestufen der Zertifikate und soll verdeutlichen, dass jeweils die CA-Zertifikate in die TSL als Vertrauensraum aufzunehmen sind, unabhängig davon, ob diese aus einer übergeordneten Root-CA abgeleitet sind oder es sich um sog. „self-signed“ CAs handelt.



~~Abbildung 3: Zertifikatshierarchien und deren Abbildung in der TSL~~

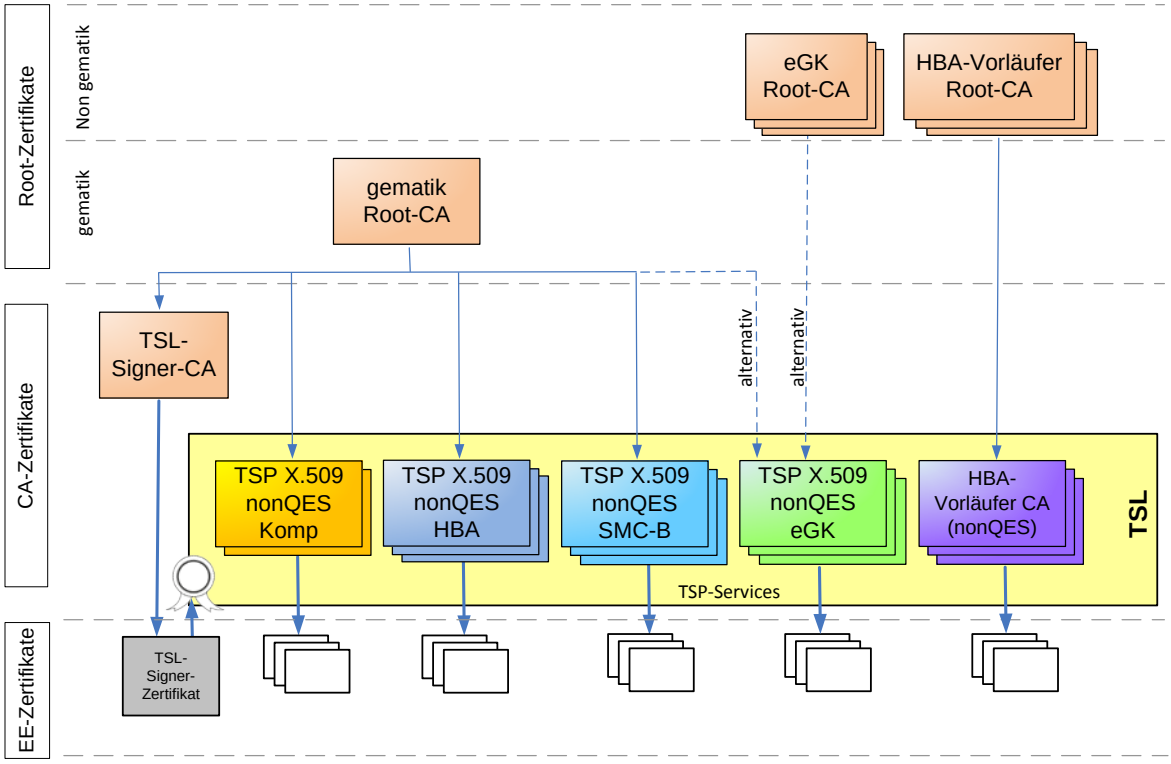
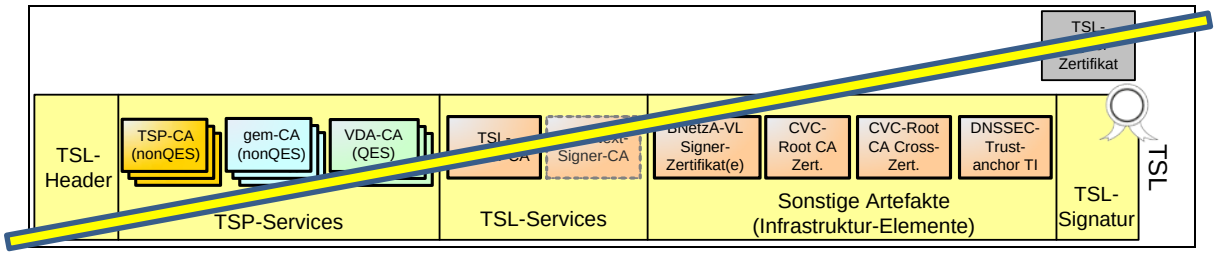


Abbildung 4: Zertifikatshierarchien und deren Abbildung in der TSL

In **~~Abbildung 5~~** ist der interne Aufbau der TSL gezeigt. Nach Verwaltungsinformationen im TSL-Header werden die CA-Zertifikate der TSP-X.509, ob self-signed oder aus überge-

ordneter Root, als „TSP-Services“ in die TSL aufgenommen. „TSL-Services“ sind die Zertifikate, die zur Absicherung der TSL selbst erforderlich sind, z. B. das TSL-Signer-CA-Zertifikat. Im Falle der Aktualisierung des Vertrauensankers wird auch der neue, zukünftige Vertrauensanker aufgenommen. Das TSL-Signerzertifikat ist in die XML-Signatur der TSL eingebettet. Das TSL-Signerzertifikat wird ausgegeben von der TSL-Signer-CA, die wiederum von der gematik Root-CA zertifiziert wird.

Neben diesen Basis-Elementen, werden in der TSL auch noch weitere Infrastruktur-Elemente als Artefakte[1] mitgeführt. Dazu gehören vor allem auch die publizierten Signer-Zertifikate der BNetzA-VL[2].



~~Abbildung 5: Aufbau der TSL~~

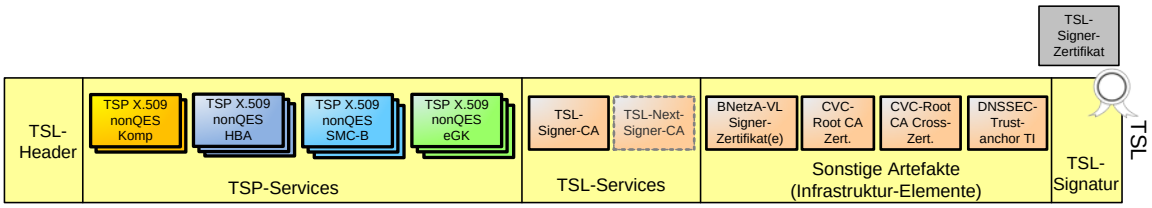


Abbildung 6: Aufbau der TSL

[…]

~~⊠ TIP1-A_2044 Beibehalten von abgelaufenen QES-CA-Zertifikaten in der TSL~~

~~Der TSL-Dienst MUSS QES-CA-Zertifikate, deren Gültigkeitszeitraum überschritten ist, noch 30 Jahre in der TSL belassen, nachdem das letzte von dieser CA ausgestellte QES-EE-Zertifikat zeitlich abgelaufen ist.⊠~~

[…]

2.3.3.5 Sperrung von CA-Zertifikaten in der TSL

[…]

~~⊠ TIP1-A_5172 Vorkehrungen bei Sperrung einer QES-CA~~

[1] Siehe auch Hinweis in Kapitel 2.3.3 „Vertrauensraum mittels TSL – Umsetzung in der TI“.

[2] Siehe auch Kapitel 2.3.1 „Vertrauensmodell für QES“

~~Die TI-Plattform MUSS Vorkehrungen treffen, welche Maßnahmen in der TI im Falle der Sperrung einer QES-CA durch die BNetzA getroffen werden.~~ ⌫

[…]

Bei ~~QES-CAs und~~ nonQES-CAs für HBA und SMC-B wird eine solche Sperrung gemäß ~~QES-Kettenmodell (s. Kap. 2.4.3) bzw. dem~~ Kompromissmodell (s. Kap. 2.4.2) umgesetzt, indem diese CAs in der TSL auf den Status „revoked“ gesetzt werden.

[…]

⌦ **TIP1-A_2066 Verhinderung der Neuausstellung von EE-Zertifikaten bei Incidents**

~~TSP-X.509QES und~~ Ein TSP-X.509nonQES ~~MÜSSEN~~ MUSS ab dem Zeitpunkt der Feststellung eines nicht-sicherheitskritischen Incidents sicherstellen, dass von der betroffenen CA keine neuen EE-Zertifikate für den Einsatz in der TI ausgestellt werden. ⌫

[…]

⌦ **TIP1-A_2069 Bestätigung des Widerrufs von CA-Zertifikaten in der TSL**

Bei dem Widerruf einer CA-Zulassung durch die gematik ~~MÜSSEN~~ MUSS der betroffene ~~TSP-X.509QES und~~ TSP-X.509nonQES den Widerruf sowie die korrekte Durchführung der Auflagen (z. B. Sperrung von CA- und EE-Zertifikaten oder Sicherstellung des weiteren Betriebs von OCSP- und Sperrdienst) schriftlich gegenüber der gematik dokumentieren und die Umsetzung bestätigen. ⌫

2.3.3.6 Aktualisierung des Vertrauensraumes

[…]

Änderungen im Inhalt der TSL können sich ergeben durch:

- Initiale Aufnahme eines ~~TSP-X.509QES oder~~ TSP-X.509nonQES mit erstem CA-Zertifikat
- Aufnahme neuer CA-Zertifikate für einen bereits enthaltenen ~~TSP-X.509QES oder~~ TSP-X.509nonQES
- Update bestehender Aussteller-CA-Zertifikate nach Ablauf deren Gültigkeitszeitraumes
- Wechsel des Status eines bestehenden Aussteller-CA-Zertifikates auf „revoked“ wegen eines nicht-sicherheitskritischen Wegfalls der Zulassung für den ~~TSP-X.509QES oder~~ TSP-X.509nonQES und die betreffende CA ab einem bestimmten Datum.
- Entfernen eines Aussteller-CA-Zertifikates im Status „revoked“ nach Ablauf des regulären Gültigkeitszeitraumes
- Entfernen eines Aussteller-CA-Zertifikats aufgrund eines sicherheitskritischen Incidents
- Aufnahme oder Entfernung von BNetzA-VL-Signerzertifikaten aufgrund Änderungen der EU-LOTL

[…]

6.6.1 Konzeptionelle Festlegungen zur QES-Zertifikatsprüfung

[…]

~~Die CA-Zertifikate der QES-VDAs werden in die TSL eingebracht als Nachweis, dass diese das Zulassungsverfahren der gematik erfolgreich durchlaufen haben.~~ Die Gültigkeitsprüfung ~~eines in der TSL enthaltenen~~ ~~QES-CA-Zertifikates~~ von CA-Zertifikaten der QES-VDAs erfolgt durch die Prüfung auf sein Vorhandensein in der BNetzA-VL und seines Servicestatus in der ~~TSL und~~ BNetzA-VL.

[…]

Generelle Prüfschritte gemäß [Common-PKI] und [RFC5280], wie die Prüfung der zeitlichen Gültigkeit ~~sowie die unter 6.5.3 genannten Schritte der Prüfung von Verwendungszweck und Zertifikatsrichtlinie~~, sind auch bei der QES-Prüfung obligatorisch. Sie sind in der folgenden Übersicht zwecks Vereinfachung und Übersichtlichkeit nicht aufgeführt. ~~Die Prüfung von qualifizierten Attributzertifikaten in der TI wird unterstützt~~.

Tabelle 11: Tab_PKI_106 Ablaufschritte der QES-Zertifikatsprüfung

Ablaufschritte der QES-Zertifikatsprüfung **Anmerkung: Die Sequenz der Ablaufschritte ist funktional hergeleitet, kann jedoch auch anders gestaltet werden, sofern das Ergebnis der Prüfung qualitativ äquivalent ist.**	
Ablaufschritt 1	Prüfung ob „qualifiziert“
Beschreibung	Prüfung, ob das QES-Zertifikat das spezifische QES-Attribut („QCStatement“) als Merkmal enthält
Vorbedingung	keine
~~Anmerkungen~~	~~Für QES-Zertifikat und optional für Attributzertifikate.~~
~~Ablaufschritt 2~~	~~Prüfung des QES-CA-Zertifikates in der TSL~~
~~Beschreibung~~	~~Ermittlung des QES-CA-Zertifikates in der TSL und Prüfung des Status dieses QES-CA-Zertifikates in der TSL zum Referenzzeitpunkt.~~
~~Vorbedingung~~	~~QES-Zertifikat, TSL~~
~~Anmerkungen~~	~~Zum prüfenden Zertifikat wird das QES-CA-Zertifikat zunächst in der TSL ermittelt. Der Status dieses QES-CA-Zertifikates wird anschließend geprüft.~~
Ablaufschritt ~~3~~ 2	Prüfung des QES-CA-Zertifikates gegen die BNetzA-VL
Beschreibung	Das ~~in der TSL enthaltene~~ QES-CA-Zertifikat wird gegen die BNetzA-VL zum Referenzzeitpunkt geprüft (Prüfung auf Vorhandensein und gültigen Servicestatus des QES-CA-Zertifikates)
Vorbedingungen	BNetzA-VL
Anmerkungen	Gültiger Servicestatus des QES-CA-Zertifikates gemäß [ETSI TS 119 612#Annex J] zum Referenzzeitpunkt
Ablaufschritt 4 3	Validierung der Zertifikatssignatur

Ablaufschritte der QES-Zertifikatsprüfung **<u>Anmerkung</u>: Die Sequenz der Ablaufschritte ist funktional hergeleitet, kann jedoch auch anders gestaltet werden, sofern das Ergebnis der Prüfung qualitativ äquivalent ist.**	
Beschreibung	Prüfung der mathematischen Korrektheit des QES-Zertifikats (Ableitung vom ermittelten QES-CA-Zertifikat)
Vorbedingungen	QES- und QES-CA-Zertifikat vorhanden
Anmerkungen	Die Prüfung erfolgt über Verifikation der Signatur
Ablaufschritt ~~5~~ 4	Prüfung des Sperrstatus
Beschreibung	Prüfung, ob Zertifikat gesperrt ist durch Abfrage des OCSP-Responders, ggf. durch Auswertung von in Signatur eingebetteter OCSP-Response.
Vorbedingung	QES-Zertifikat vorhanden
Anmerkungen	OCSP-Dienste für QES-Zertifikate müssen den Vorgaben von [eIDAS] genügen. Die Ableitung des OCSP-Signer-Zertifikates erfolgt gemäß [RFC6960]

Änderungen in gemSpec_TSL

[…]

2.1 Zweck der TSL

[…]

Um den Vertrauensraum für X.509-Zertifikate in der Telematikinfrastruktur technisch abzubilden, wird die TSL verwendet. Die TSL-Datei ist eine signierte Whitelist der zugelassenen Zertifikatsherausgeber. Das heißt, die TSL-Datei enthält sämtliche nonQES-~~ und QES~~-X.509-CA-Zertifikate, die in der TI verwendet werden.

[…]

6.1.2 Schnittstelle P_TSL-Management

[…]

6.1.2.2.4 Prüfung

TIP1-A_4044 Prüfung auf ungültige Einträge

Der Anbieter des TSL-Dienstes MUSS sicherstellen, dass in der aktuellen Version keine zum Zeitpunkt der TSL-Erstellung abgelaufenen Einträge enthalten sind~~, mit Ausnahme von X.509 QES-CA-Zertifikaten~~. Der Anbieter des TSL-Dienstes MUSS abgelaufene Einträge und andere Normabweichungen an gematik melden.

~~**TIP1-A_4851 Beibehalten von abgelaufenen QES-CA-Zertifikaten in der TSL**~~

~~Der TSL-Dienst MUSS QES-CA-Zertifikate, deren Gültigkeitszeitraum überschritten ist, noch 30 Jahre in der TSL belassen, nachdem das letzte von dieser CA ausgestellte QES-EE-Zertifikat zeitlich abgelaufen ist.~~

[…]

7.3.2 Angaben zum TSP-Dienst

[…]

Es bestehen die folgenden Möglichkeiten hinsichtlich URI im ServiceSupplyPoint-Element:

- Bei Einträgen von X.509-CA-Zertifikaten wird die Adresse des OCSP-Responder (oder CRL-Verteilungspunktes) eingetragen, der die Status- bzw. Sperrinformationen zu den von der CA ausgegebenen Zertifikaten zur Verfügung stellt (sofern vorhanden).

~~Wenn sich ein OCSP-Responder außerhalb der TI befindet, wird die Adresse des OCSP-Proxy eingetragen.~~

[…]

7.5 BNetzA-VL

[…]

⌦ TIP1-A_6767 BNetzA-VL ServiceInformationExtensions

Der TSL-Dienst MUSS für die BNetzA-VL im Element ServiceInformationExtensions ein Element Extension eintragen, welches den Platzhalter-OID (oid_tsl_placeholder) gemäß [gemSpec_OID#Tab_PKI_407] enthält. ⌫

Im Rahmen der QES-Zertifikatsprüfung benötigen zertifikatsprüfende Komponenten neben den Informationen in der BNetzA-VL zu CA-Zertifikaten von VDAs auch Referenzen zu deren OCSP-Respondern in der TI, wenn diese von in der TI zugelassenen VDAs bereitgestellt wurden. Die URLs der OCSP-Responder in der TI werden in der TSL anhand von Einträgen im Feld AdditionalServiceInformation als Tupel der Elemente URI und InformationValue innerhalb des Services mit ServiceTypeIdentifier für die BNetzA-VL-Signer realisiert.

⌦ TIP1-A_7219 BNetzA-VL AdditionalServiceInformation für Umleitung von OCSP-Responder-Adressen in der TI

Der TSL-Dienst MUSS für jede von der gematik bereitgestellte OCSP-Responder-URL für QES-Zwecke je ein Element AdditionalServiceInformation innerhalb des ServiceInformationExtensions Elementes als Tupel der Elemente URI und InformationValue erstellen.

Das URI Element MUSS dabei jeweils folgenden Eintrag enthalten: <URI>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures/</URI>.

Das InformationValue Element MUSS jeweils den von der gematik bereitgestellten Text-Eintrag als String (URLs getrennt durch Leerzeichen) enthalten.⌫

Der Text-Eintrag für das InformationValue Element enthält demnach als Umsetzung der OCSP-Responder-Adresse die Quell-URL (aus dem Zertifikat in der Extension AIA) und die Ziel-URL (URL aus dem Namensraum der TI, unter der der OCSP-Responder in der TI erreichbar ist).

Ein TSL-Eintrag für ~~ein~~ die BNetzA-VL Referenzierung inklusive der URLs für OCSP-Responder für VDAs in der TI sieht also dergestalt aus:

```
<TrustServiceProvider>
   <TSPInformation>
      <TSPName>
         <Name xml:lang="de">Bundesnetzagentur für Elektrizität, Gas, Telekommunikation,
Post und Eisenbahnen</Name>
      </TSPName>
      <TSPTradeName>
         <Name xml:lang="de">Bundesnetzagentur</Name>
      </TSPTradeName>
```

```
        <TSPAddress>
            <PostalAddresses>
                <PostalAddress xml:lang="de">
                    <StreetAddress>Canisiusstr. 21</StreetAddress>
                    <Locality>Mainz</Locality>
                    <StateOrProvince>NW</StateOrProvince>
                    <PostalCode>55122</PostalCode>
                    <CountryName>DE</CountryName>
                </PostalAddress>
            </PostalAddresses>
            <ElectronicAddress>
                <URI>mailto:eIDAS@bnetza.de</URI>
            </ElectronicAddress>
        </TSPAddress>
        <TSPInformationURI>
            <URI xml:lang="en">http://www.bundesnetzagentur.de</URI>
        </TSPInformationURI>
    </TSPInformation>
    <TSPServices>
        <TSPService>
            <ServiceInformation>
                <ServiceTypeIdentifier>
                    http://uri.telematik/TrstSvc/Svctype/TrustedList/schemerules/DE
                </ServiceTypeIdentifier>
                <ServiceName>
                    <Name xml:lang="de">
                      {z.B.“CN=14R-TSL 1:PN,O=Bundesnetzagentur,C=DE”}
                    </Name>
                </ServiceName>
                <ServiceDigitalIdentity>
                    <DigitalId>
                      <X509Certificate>
                            Base64-codiertes BNetzA-VL-Signer-Zertifikat}
                      </X509Certificate>
                    </DigitalId>
                    {weitere DigitalId-Elemente mit Signer-Zertifikaten}
                </ServiceDigitalIdentity>
                <ServiceStatus>
                    http://uri.etsi.org/TrstSvc/Svcstatus/inaccord
                </ServiceStatus>
                    <StatusStartingTime>
                    {z.B. “2016-07-01T00:00:01Z”}
                    </StatusStartingTime>
                <ServiceSupplyPoints>
                    <ServiceSupplyPoint>
                    {primäre TI-Download-Adresse der BNetzA-VL}
                    </ServiceSupplyPoint>
                    <ServiceSupplyPoint>
                    {sekundäre TI-Download-Adresse der BNetzA-VL}
                    </ServiceSupplyPoint>
                </ServiceSupplyPoints>
                <ServiceInformationExtensions>
                    <Extension Critical="false">
                      <ExtensionOID>{oid_tsl_placeholder}</ExtensionOID>
                      <ExtensionValue>oid_tsl_placeholder</ExtensionValue>
                    </Extension>
                    <Extension Critical="false">
                      <AdditionalServiceInformation>
```

```
                        <URI>
                            http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
                        </URI>
                        <InformationValue>
                            {TextFeld: Quell-URL1 Ziel-URL1}
                        </InformationValue>
                    </AdditionalServiceInformation>
                    <AdditionalServiceInformation>
                        <URI>
                            http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
                        </URI>
                        <InformationValue>
                            {TextFeld: Quell-URL2 Ziel-URL2}
                        </InformationValue>
                    </AdditionalServiceInformation>
                    {ggf. weitere AdditionalServiceInformation-Elemente}
                </Extension>
            </ServiceInformationExtensions>
        </ServiceInformation>
    </TSPService>
</TSPServices>
</TrustServiceProvider>
```

Änderungen in gemSpec_OCSP_Proxy

[…]

2 Systemüberblick

Der Produkttyp OCSP-Proxy wird eingesetzt, um die Statusinformation der Zertifikate der zeitlich begrenzt durch die TI unterstützten HBA-Vorläuferkarten in der TI-Plattform verfügbar zu machen.

Zusätzlich ermöglicht er das Weiterleiten von OCSP-Anfragen aus der TI an andere im Internet erreichbare OCSP-Responder für QES-Zwecke.

Im Falle von OCSP-Anfragen für End-Entity- Zertifikate leitet der OCSP-Proxy die Anfrage an den zuständigen OCSP-Responder im Internet weiter und gibt die vom OCSP-Responder zurück gelieferte OCSP-Antwort an die zertifikatsvalidierende Komponente zurück.

[…]

5.1.1.1 Schnittstellendefinition

Der OCSP-Proxy muss die technische Schnittstelle I_OCSP_Status_Information gemäß [gemSpec_PKI#9] implementieren und in der Telematikinfrastruktur anbieten.

Über diese Schnittstelle werden die Statusinformation für Zertifikate der unterstützten HBA-Vorläuferkarten und anderer QES-Verfahren in der TI-Plattform verfügbar gemacht.

[…]

5.1.1.2 Umsetzung

[…]

⌦ **TIP1-A_5849 OCSP-Anfragen aus der TI beantworten**

Der OCSP-Proxy MUSS den technischen Use Case "TUC_OCSP-Proxy_002 OCSP-Anfragen aus der TI beantworten" gemäß Tab_OCSP-Proxy_002 umsetzen. ⌫

[…]

Tabelle 1: Tab_OCSP-Proxy_002 TUC_OCSP-Proxy_002 OCSP-Anfragen aus der TI beantworten

Element	Beschreibung
Name	TUC_OCSP-Proxy_002 "OCSP-Anfragen aus der TI beantworten"

Beschreibung	Dieser Use Case beschreibt den Prozess der Zertifikatsstatusauskunft des OCSP-Proxys bei OCSP-Anfragen für nonQES-EE- und QES-EE-Zertifikate der HBA-Vorläuferkarten und anderer QES-Verfahren.
Auslöser	OCSP-Anfrage einer zertifikatsvalidierenden Komponente aus der TI
Vorbedingungen	
Eingangsdaten	OCSP-Anfrage
Komponenten	Zertifikatsvalidierende Komponenten der TI (z. B. Konnektor), OCSP-Proxy, OCSP-Responder (der HBA-Vorläuferkarten, u.a.)
Ausgangsdaten	OCSP-Antwort eines OCSP-Responders ~~von HBA-Vorläuferkarten~~
Nachbedingungen	
Standardablauf	1. [OCSP-Proxy]: OCSP-Anfrage der zertifikatsvalidierenden Komponente empfangen. 2. [OCSP-Proxy]: OCSP-Anfrage ~~unter Verwendung von "ServiceSupplyPoint Internet"~~ an den entsprechenden OCSP-Responder im Internet weiterleiten unter Verwendung der Ziel-URL im Internet aus der empfangenen OCSP-Anfrage. 3. [OCSP-Proxy]: OCSP-Antwort des OCSP-Responders im Internet empfangen. 4. [OCSP-Proxy]: OCSP-Antwort des OCSP-Responders im Internet an zertifikatsvalidierende Komponente zurückgeben.
Varianten/Alternativen	
Fehlerfälle	2a [OCSP-Proxy]: Der OCSP-Responder im Internet ist nicht erreichbar: OCSP-Response mit einer unsignierten Error-Response des Typs "internalError" (siehe [RFC2560#2.3]) zurückgeben.
Nichtfunktionale Anforderungen	
Anmerkungen	Die URL, mit der der OCSP-Proxy-Responder angesprochen wird, enthält auch die Ziel-URL des OCSP-Responders im Internet (s. [gemSpec_VPN_ZugD#TIP1-A_4322]).
Zugehörige Diagramme	

Änderungen in Produkttypsteckbriefen

Die Änderung betrifft die folgenden Produkttypsteckbriefe:

- gemProdT_TSL
- gemProdT_Kon
- gemProdT_X.509_TSP_QES
- gemProdT_OCSP_Proxy

Änderungsbedarf:

Für den Fall, dass der Benutzer am Signaturproxy einen Signaturauftrag im Bestätigungsmodus während der dafür konfigurierten Zeitspanne nicht freigibt oder einen Signaturvorgang abbricht, ist nicht spezifiziert, welchen Fehler der Signaturproxy an das aufrufende Clientsystem zurückgibt.

Für den Fall, dass der Benutzer am Signaturproxy das Prüfergebnis für einen Signaturprüfauftrag im Ansichtsmodus während der dafür konfigurierten Zeitspanne nicht freigibt (TIP1-A_5701: Bewertung von INCONCLUSIVE), ist nicht spezifiziert, welchen Fehler der Signaturproxy an das aufrufende Clientsystem zurückgibt.

Um Interoperabilität für diesen Fall zu gewährleisten, sind die Fehlercodes festzulegen.

Änderungen in gemSpec_Kon_SigProxy

1. Die Anforderung TIP1-A_4662 wird wie folgt angepasst
alt:
Wird der Signaturproxy im Bestätigungsmodus betrieben, so MUSS der Signaturproxy auf eine Freigabe des Signaturvorganges durch Anwender warten, bevor der Signaturauftrag an den Konnektor gesendet wird. Wird die Freigabe des Signaturvorganges über den Signaturproxy nicht während der konfigurierten Zeitspanne durch den Benutzer erteilt, so MUSS die der Signaturproxy die Signaturerzeugung abbrechen.
neu:
Wird der Signaturproxy im Bestätigungsmodus betrieben, so MUSS der Signaturproxy auf eine Freigabe des Signaturvorganges durch den Anwender warten, bevor der Signaturauftrag an den Konnektor gesendet wird. Wird die Freigabe des Signaturvorganges über den Signaturproxy nicht während der konfigurierten Zeitspanne durch den Benutzer erteilt, so MUSS der Signaturproxy die Signaturerzeugung mit dem Fehler 4116 "Timeout (Benutzer)" abbrechen.

2. Die Anforderung TIP1-A_4663 wird wie folgt angepasst
alt:
Der Signaturproxy MUSS im Bestätigungsmodus dem Benutzer vor Beginn des Signaturvorganges die Möglichkeit zum Abbruch des Vorganges bieten.
neu:
Der Signaturproxy MUSS im Bestätigungsmodus dem Benutzer vor Beginn des Signaturvorganges die Möglichkeit zum Abbruch (Fehler 4049 "Abbruch durch den Benutzer") des Vorganges bieten.

3. In Tabelle TAB_SIG_850 werden 2 Zeilen ergänzt:
4049 | Technical | Error | Abbruch durch den Benutzer
4116 | Technical | Error | Timeout (Benutzer)

4. TAB_SIG_852/Schritt 3 wird wie folgt angepasst
alt:
Das Ergebnis wird angezeigt.
neu:
Das Ergebnis wird angezeigt.
Erfolgt der Aufruf mit TvMode=UNCONFIRMED, muss der Signaturproxy auf eine Freigabe des Prüfergebnisses durch den Anwender warten, bevor das Prüfergebnis an den Aufrufer zurückgesendet wird. Wird die Freigabe des Prüfergebnisses über den

Signaturproxy nicht während einer konfigurierbaren Zeitspanne durch den Benutzer erteilt, so muss der Signaturproxy die Signaturprüfung mit dem Fehler 4116 abbrechen.

5. In Tabelle TAB_SIG_853 wird eine Zeile ergänzt:
4116 | Technical | Error | Timeout (Benutzer)

Änderungsbedarf in gemSpec_Perf

4.1.2 Produkttyp Konnektor

Tabelle 23: Tab_gemSpec_Perf_Konnektor – Last- und Bearbeitungszeitvorgaben

Schnittstellenoperationen	Last		Bearbeitungszeit	
	LE-U	Spitzenlast [1/h]	Größe der Anfragenachricht [kByte]	Mittelwert [msec]
Fachanwendung				
I_VSD_Service				
ReadVSD – mit Akt.-Prüfung, mit Update	1	1		6130
	2	1		
	3	4		
	4	11		
ReadVSD – mit Akt.-Prüfung, ohne Update	1	50		3940
	2	50		
	3	175		
	4	437		
ReadVSD – ohne Akt.-Prüfung				3820
UpdateVSD – automat. Akt.-Prüfung, mit Update				5720
UpdateVSD – automat. Akt.-Prüfung, ohne Update				3130
I_NFD_Management				
NFD von eGK lesen	1	6	10,5	7260
	2	28		
	3	115		
	4	286		
NFD auf eGK schreiben	1	11	10,5	5780
	2	51		
	3	213		
	4	533		
NFD von eGK löschen	1	1	10,5	4800
	2	5		
	3	21		
	4	53		
I_DPE_Management				
DPE von eGK lesen	1	1	1,5	4300
	2	3		
	3	14		
	4	36		
DPE auf eGK schreiben	1	1	1,5	4590
	2	5		
	3	20		
	4	51		
DPE von eGK löschen	1	0,1	1,5	4260

© gematik – öffentlich

		2	0,5		
		3	2		
		4	5		
I_IDP_Auth_Active_Client					
	issue_Identity_Assertion			5	2500
	renew_Identity_Assertion			20	2500
	cancel_Identity_Assertion			20	500
I_IDP_Auth_Passive_Client					
	signin			2	3500
	signout			1	500
I_Local_IDP_Service					
	sign_Token			5	2500
I_AMTS_Service					
	ReadMP			30	5268
	WriteMP (mit C2C)			30	6625
	WriteMP (ohne C2C)			30	4020
Basisdienste					
I_Sign_Operations					
	sign_Document			10	1010
		1	217	100	1030
		2	258		
		3	351		
		4	575		
				1000	1440
	sign_Document (XAdES, XML_25MB, enveloped)		13	25000	10500
	sign_Document (CAdES, TIFF_25MB, detached)			25000	7300
	sign_Document (PAdES, PDFA_2b_25MB)			25000	7300
	verify_Document			10	1570
		1	217	100	1600
		2	258		
		3	351		
		4	575		
				1000	1930
	verify_Document (XAdES, XML_25MB, enveloped, IncludeRevocationInfo=false)		13	25000	9000
	verify_Document (CAdES, TIFF_25MB, IncludeRevocationInfo=false)			25000	9000
	verify_Document (PAdES, PDFA_2b_25MB, IncludeRevocationInfo=false)			25000	10600
	external_Authenticate				885
	get_Certificate				220
I_SAK_Operations					

© gematik – öffentlich

	sign_Document_QES (Stapelgröße 1)			10	3540
		1	17	100	3790
		2	65		
		3	177		
		4	442		
				1000	4070
	sign_Document_QES (XAdES, XML_25MB, enveloped)			25000	12810
	sign_Document_QES (CAdES, TIFF_25MB)			25000	9610
	sign_Document_QES (PAdES, PDFA_2b_25MB)			25000	9610
	sign_Document_QES (Stapelgröße 2, 2 * 100 kB Dokumente)	1	3	200	8870
		2	11		
		3	30		
		4	74		
	verify_Document_QES			10	2580
		1	10	100	2610
		2	39		
		3	113		
		4	282		
				1000	2940
	verify_Document_QES (XAdES, XML_25MB, enveloped, IncludeRevocationInfo=false)			25000	10010
	verify_Document_QES (CAdES, TIFF_25MB, detached IncludeRevocationInfo=false)			25000	10010
	verify_Document_QES (PAdES, PDFA_2b_25MB, IncludeRevocationInfo=false)			25000	11610
I_KV_Card_Unlocking					
	authorize_Card (no Cache)				2020
	authorize_Card (Cache)				1830
I_Crypt_Operations					
	encrypt_Document			10	1860
		1	217	100	1880
		2	258		
		3	351		
		4	575		
				1000	2200
	encrypt_Document (XMLEnc, TIFF_25MB, ein Empfänger)		13	25000	10600
	encrypt_Document (CMS, TIFF_25MB, ein Empfänger)			25000	7800
	decrypt_Document			10	490
		1	217	100	510
		2	258		
		3	351		
		4	575		

© gematik – öffentlich

					1000	820
		decrypt_Document (XMLEnc, TIFF_25MB)		13	25000	8900
		decrypt_Document (CMS, TIFF_25MB)			25000	8900
	I_Cert_Verification					
		verify_Certificate				1150
	I_Directory_Query					
		search_Directory	1	200		~~1220~~ 2220
			2	300		
			3	500		
			4	1000		

B1 – Verteilung der Konnektorbearbeitungszeiten auf Komponenten

Tabelle 45: Tab_gemSpec_Perf_Konnektorbearbeitungszeiten_pro_Komponente

Schnittstellenoperationen	Konnektor Gesamt [msec]	Konnektor intern mit LE-LAN [msec]	Kartenterm. + Karte [msec]	OCSP + Zugangsnetz+ Zentr.Netz [msec]
Lesen VSD mit Onlineprüfung mit Aktualisierung	6130	1250	3780	1100
Lesen VSD mit Onlineprüfung ohne Aktualisierung	3940	790	3150	0
Lesen VSD ohne Onlineprüfung	3820	610	3210	0
Automatische Onlineprüfung mit Aktualisierung der VSD	5720	1030	3590	1100
Automatische Onlineprüfung ohne Aktualisierung der VSD	3130	460	2670	0
NFD von eGK lesen	7260	1070	4080	2110
NFD auf eGK schreiben	5780	850	4930	0
NFD von eGK löschen	4800	810	3990	0
DPE von eGK lesen	4300	935	3365	0
DPE auf eGK schreiben	4590	975	3615	0
DPE von eGK löschen	4260	810	3450	0
I_AMTS_Service::ReadMP	5268	1010	4258	0
I_AMTS_Service::WriteMP (mit C2C)	6625	1120	5505	0
I_AMTS_Service::WriteMP (ohne C2C)	4020	1020	3000	0
I_Sign_Operations::sign_Document (10 kB)	1010	300	710	0
I_Sign_Operations::sign_Document (100 kB)	1030	320	710	0

© gematik – öffentlich

Schnittstellenoperationen	Konnektor Gesamt [msec]	Konnektor intern mit LE-LAN [msec]	Kartenterm. + Karte [msec]	OCSP + Zugangsnetz+ Zentr.Netz [msec]
I_Sign_Operations::sign_Document (1 MB) (XAdES, XML_1MB, enveloped) (CAdES, TIFF_1MB, detached) (PAdES, PDFA_2b_1MB_Komplex)	1440	730	710	0
				0
I_Sign_Operations::sign_Document (XAdES, XML_25MB, enveloped)	10500	9790	710	
I_Sign_Operations::sign_Document (CAdES, TIFF_25MB, detached)	7300	6590	710	
I_Sign_Operations::sign_Document (PAdES, PDFA_2b_25MB_Bilder_und_Text)	7300	6590	710	
I_Sign_Operations::verify_Document (10 kB)	1570	470	0	1100
I_Sign_Operations::verify_Document (100 kB)	1600	500	0	1100
I_Sign_Operations::verify_Document (1 MB) (XAdES, XML_1MB, enveloped) (CAdES, TIFF_1MB, detached) (PAdES, PDFA_2b_1MB_Komplex)	1930	830	0	1100
I_Sign_Operations::verify_Document (XAdES, XML_25MB, enveloped, IncludeRevocationInfo=false)	9000	7900	0	1100
I_Sign_Operations::verify_Document (CAdES, TIFF_25MB, IncludeRevocationInfo=false)	9000	7900	0	1100
I_Sign_Operations::verify_Document (PAdES, PDFA_2b_25MB, IncludeRevocationInfo=false)	10600	9500	0	1100
I_SAK_Operations::sign_Document_QES (10KB)	3540	520	910	2110
I_SAK_Operations::sign_Document_QES (100KB, Stapelgröße 1, SE#1)	3790	770	910	2110
I_SAK_Operations::sign_Document_QES (100KB, Stapelgröße 2, SE#2)	8870	1430	5330	2110
I_SAK_Operations::sign_Document_QES (1MB)	4070	1050	910	2110
I_SAK_Operations::sign_Document_QES (25MB)				
I_SAK_Operations::sign_Document_QES (XAdES, XML_25MB, enveloped)	12810	9790	910	2110
I_SAK_Operations::sign_Document_QES (CAdES, TIFF_25MB)	9610	6590	910	2110
I_SAK_Operations::sign_Document_QES (PAdES, PDFA_2b_25MB)	9610	6590	910	2110
I_SAK_Operations::verify_Document_QES (10KB)	2580	470	0	2110
I_SAK_Operations::verify_Document_QES (100KB)	0		0	2110

© gematik – öffentlich

Schnittstellenoperationen	Konnektor Gesamt [msec]	Konnektor intern mit LE-LAN [msec]	Kartenterm. + Karte [msec]	OCSP + Zugangsnetz+ Zentr.Netz [msec]
	2610	500		
I_SAK_Operations::verify_Document_QES (1 MB)	2940	830	0	2110
I_SAK_Operations::verify_Document_QES (XAdES, XML_25MB, enveloped, IncludeRevocationInfo=false)	10010	7900	0	2110
I_SAK_Operations::verify_Document_QES (CAdES, TIFF_25MB, IncludeRevocationInfo=false)	10010	7900	0	2110
I_SAK_Operations::verify_Document_QES (PAdES, PDFA_2b_25MB, IncludeRevocationInfo=false)	11610	9500	0	2110
I_KV_Card_Unlocking::authorize_Card (no Cache)	2020	100	1920	0
I_KV_Card_Unlocking::authorize_Card (Cache)	1830	100	1730	0
I_Crypt_Operations::encrypt_Document (10 kB)	1860	760	0	1100
I_Crypt_Operations::encrypt_Document (100 kB)	1880	780	0	1100
I_Crypt_Operations::encrypt_Document (1 MB)	2200	1100	0	1100
I_Crypt_Operations::encrypt_Document (XMLEnc, XML_25MB, ein Empfänger)	10600	9500	0	1100
I_Crypt_Operations::encrypt_Document (CMS, TIFF_25MB, ein Empfänger)	7800	6700	0	1100
I_Crypt_Operations::decrypt_Document (10 kB)	490	150	340	0
I_Crypt_Operations::decrypt_Document (100 kB)	510	170	340	0
I_Crypt_Operations::decrypt_Document (1 MB)(XMLEnc, XML_1MB)(CMS, TIFF_1MB)	820	480	340	0
I_Crypt_Operations::decrypt_Document (XMLEnc, XML_25MB)	8900	8560	340	
I_Crypt_Operations::decrypt_Document (CMS, TIFF_25MB)	8900	8560	340	
I_Cert_Verification::verify_Certificate	1150	50	0	1100
I_Directory_Query::search_Directory	~~1220~~ 2220	~~1220~~ 2000	0	~~0~~ 220

© gematik – öffentlich

Änderungsbedarf:

Ein Bedarf für die Behandlung von Attributzertifikaten besteht nicht mehr. Daher wird die Behandlung von Attributzertifikaten aus der Konnektorspezifikation und der HBA Kartenspezifikation entfernt. Da die Änderungen in den Schemadateien keine Wirkung auf Anfragen haben, die keine Attributzertifikate behandeln, und Attributzertifikate bisher nicht verwendet wurden, wird die Änderung als kompatibel bewertet und die Version der betroffenen Schemadateien in der dritten Stelle hochgezählt.

Änderungen in gemKPT_Arch_TIP

Die Anforderung TIP1-A_5760 entfällt.

Änderungen in gemSpec_Kon

4.1.8.3.1 TUC_KON_155 ”Dokumente zur Signatur vorbereiten”

⌧ **TIP1-A_4646 TUC_KON_155 „Dokumente zur Signatur vorbereiten“**

Der Konnektor MUSS den technischen Use Case TUC_KON_155 „Dokumente zur Signatur vorbereiten” umsetzen.

Tabelle 1: TAB_KON_748 - TUC_KON_155 „Dokumente zur Signatur vorbereiten“

Element	Beschreibung
Name	TUC_KON_155 ”Dokumente zur Signatur vorbereiten”
Beschreibung	Die zu signierenden Dokumente werden entsprechend den Erfordernissen der Signaturverfahren für die QES oder nonQES vorbereitet.
Anwendungsumfeld	Erstellung von qualifizierten elektronischen Signaturen (QES) und nicht-qualifizierten elektronischen Signaturen (nonQES)
Auslöser	Aufruf durch TUC_KON_150 „Dokumente QES signieren“ oder TUC_KON_160 „Dokumente nonQES signieren“
Vorbedingungen	keine
Eingangsdaten	• signatureMode (Signaturart: QES | nonQES) • documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente) und pro Dokument: • documentFormat (Formatangabe für das zu signierende Dokument) • optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur (siehe Operation SignDocument, Parameter dss:OptionalInputs), darin u.a. ▪ signatureType

Element	Beschreibung
	(URI für den Signaturtyp XML-, CMS-, S/MIME-o PDF-Signatur) • certificate (Signaturzertifikat) • ocspResponses – *optional* (OCSP-Response des EE-Zertifikats, das bei der Signaturerstellung in die Signatur eingebettet wird.)
Komponenten	Konnektor
Ausgangsdaten	• preProcessedDocuments (Aufbereitetes zu signierendes Dokument bzw. aufbereitete zu signierende Dokumente)
Standardablauf	signatureType = XMLDSig (XAdES) Entsprechend den Regeln für die QES und die nonQES werden zunächst weitere Signatureigenschaften zum jeweiligen Dokument in Form von `QualifyingProperties` (siehe [XAdES]) hinzugefügt. ~~Falls erforderlich, werden die Attributzertifikate hinzugefügt.~~ Die Systemzeit des Anwendungskonnektors muss in das XML-Element `SigningTime` (siehe [XAdES]) eingetragen werden. Die Signatur wird anschließend entsprechend [XMLDSig] vorbereitet. D. h., es wird je Dokument nach Erzeugung der Reference Elemente das SignedInfo Element aufgebaut. Dessen Inhalt ergibt dann nach erfolgter XML-Kanonisierung und Hashing die DTBS (Data To Be Signed), die später zur Karte gesendet werden. certificate wird im Element `ds:KeyInfo/ds:X509Data` gespeichert. Im Fall signatureMode = QES können neben den reinen Nutzdaten auch alle weiteren Elemente in die Signatur einbezogen werden, die für die Rekonstruktion der ursprünglich dargestellten Daten in der sicheren Anzeige erforderlich sind. Für XML-Dokumente sind das, falls vorhanden, das/die XML-Schema(ta). Für diese werden Referenzen (Hash + URI) in die Signatur eingebettet. Die URI ist im Fall übergebener XML-Schemata der übergebene *signatureType* - Parameter. Die URI ist im Fall der im Konnektor im Rahmen einer Signaturrichtlinie hinterlegten XML-Schemata/XSL-Stylesheets die URI der Signaturrichtlinie, ergänzt um den Dateinamen mit Pfad, wie in der Signaturrichtlinie festgelegt. (Beispiel: URI für Schemadatei CDA.xsd der Signaturrichtlinie SR_DF_SV_VHITG_ARZTBRIEF lautet: urn:gematik:fa:sak:cda:r1:v1:fa:sak:cda:schemas:CDA.xsd) Das Einbetten der Referenzen erfolgt über das XML-Element ds:object/ds:manifest (XMLDSig) mit eingebetteten XML-Elementen ds:Reference, die eine URI (RefURI) als Identifier für die jeweilige Datei und einen Hash über die jeweilige Resource enthalten. Der ShortTextClientsystem muss in die Signatur in das `DataObjectFormat/Description`-Element gemäß [XAdES] (Abschnitt 7.2.5) eingebettet werden. Falls durch den Aufrufparameter `SIG:IncludeRevocationInfo` angefordert, wird die für die Offline-Prüfung notwendige OCSP-Antwort im Sinne vom ES-X-L vom Konnektor in die Signatur eingebettet: Die base-64 kodierte OCSP-Response wird im Feld `QualifyingProperties/UnsignedProperties/UnsignedSignatu`

Element	Beschreibung
	`reProperties/RevocationValues/OCSPValues/EncapsulatedOCSPValue` (selbst DER-kodiert) gespeichert. signatureType = CMS (CAdES) Etwaig einzubettende XML-Schemata werden zunächst wie für XAdES definiert in ein ds:manifest-Element eingebettet. Die so erzeugte Zeichenkette wird als genau ein ASN.1 Character String vom Typ UTF8String verpackt. Dieser wird als contentDescription in einen Content-Hints Attributwert vom Typ ContentHints verpackt, wobei der contentType=id-data gemäß [CAdES]. Der ShortTextClientsystem muss in die Signatur in das `content-hints.ContentDescription`-Attribut gemäß [CAdES] (Abschnitt 5.10.3) eingebettet werden. Ist die Einbettung von OCSP-Responses gefordert, wird die für die Offline-Prüfung notwendige OCSP-Antwort des EE-Zertifikats im Attribut SingedData.crls.other abgelegt. signatureType = PDF/A (PAdES) Der ShortTextClientsystem muss bei einer PDF-Signatur in das `Reason`-Feld eingebettet werden. OCSP-Response des EE-Zertifikats ist im Document Security Store gemäß [PAdES-4] einzubetten. Es sind die Vorgaben zum Signaturprofil gemäß Tabelle TAB_KON_779 „Profilierung der Signaturformate“ zu erfüllen. Die aufbereiteten zu signierenden Dokumente werden an den Aufrufer zurückgegeben.
Varianten/Alternativen	keine
Fehlerfälle	Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_586 Fehlercodes TUC_KON_155 „Dokumente zur Signatur vorbe¬reiten“ „PDF/A (PAdES)“ Es ist nicht genügend Speicherplatz im PDF-Dokument verfügbar: 4205
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	keine

Tabelle 2: TAB_KON_586 Fehlercodes TUC_KON_155 „Dokumente zur Signatur vorbereiten“

Fehlercode	ErrorType	Severity	Fehlertext
4205	Technical	Error	Es ist nicht genügend Speicherplatz im PDF-Dokument verfügbar.

⌫

4.1.8.3.1 TUC_KON_152 ”Signaturvoraussetzungen für QES prüfen”

⌦ TIP1-A_4649 TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“

Der Konnektor MUSS den technischen Use Case TUC_KON_152 „Signaturvoraussetzungen für QES prüfen” umsetzen.

Tabelle 3: TAB_KON_751 – TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“

Element	Beschreibung
Name	TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“
Beschreibung	Es werden die Voraussetzungen an die zu signierenden Dokumente und das Signaturzertifikat geprüft. Es werden die `QES_DocFormate` unterstützt.
Auslöser	TUC_KON_150 „Dokumente QES signieren“
Vorbedingungen	keine
Eingangsdaten	1. Zu signierende Dokumente 2. optionale Eingabeparameter zur Steuerung der Details der Signaturerstellung 3. CardSession Signaturkarte 4. zu verwendende Identität (Zertifikatsreferenz) 5. includeRevocationInfo [Boolean] - optional; Default: true (Dieser Parameter steuert die Einbettung von OCSP-Responses in die Signatur. true: Die Sperrinformationen werden in ocspResponses zurückgegeben.)
Komponenten	Konnektor, Kartenterminal, Signaturkarte
Ausgangsdaten	1. Prüfergebnis 2. Signaturzertifikat 3. ocspResponses - optional/nur wenn includeRevocationInfo = true (OCSP-Response des EE-Zertifikats, die beim Aufruf von TUC_KON_037 „Zertifikat prüfen“ zurückgegeben wird)
Standardablauf	1. Abhängig von seinem Typ werden für jedes Dokument die typabhängigen Dokumentvalidierungsschritte durchgeführt (Aufruf TUC_KON_080 „Dokument validieren“). Wird der Aufruf von TUC_KON_080 mit einem Fehler beendet, wird die Prüfung im laufenden TUC mit diesem Fehler abgebrochen. 2. Durch Aufruf von TUC_KON_216 „Lese Zertifikat“ ~~werden~~ wird das Signaturzertifikat ~~und – falls für mindestens ein Dokument benötigt – die Attributzertifikate~~ von der Signaturkarte gelesen. 3. Das Signaturzertifikat ~~und die Attributzertifikate werden~~ wird durch Aufruf von TUC_KON_037 „Zertifikat prüfen“ {C.HP.QES; required; true; oid_hba_qes; OCSP; getOCSPResponses = includeRevocationInfo ~~; Liste der Attributzertifikate~~} geprüft.
Varianten/Alternativen	keine
Fehlerfälle	keine
Nichtfunktionale	keine

Element	Beschreibung
Anforderungen	
Zugehörige Diagramme	keine

Tabelle 4: TAB_KON_588 Fehlercodes TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können keine weiteren Fehlercodes auftreten.			

⌫

4.1.8.4.6 TUC_KON_151 „QES Dokumentensignatur prüfen“

⌦ **TIP1-A_4672 TUC_KON_151 „QES-Dokumentensignatur prüfen“**

Der Konnektor MUSS den technischen Use Case TUC_KON_151 „QES-Dokumentensignatur prüfen” umsetzen.

Tabelle 215: TAB_KON_591 - TUC_KON_151 „QES-Dokumentensignatur prüfen“

Element	Beschreibung
Name	TUC_KON_151 „QES-Dokumentensignatur prüfen”
Beschreibung	Es wird die QES eines Dokuments geprüft. Dabei werden die Signaturverfahren laut Tabelle 187: TAB_KON_582 – Signaturverfahren unterstützt. Sind mehrere Signaturen vorhanden, so werden alle geprüft. Auch Parallel- und Gegensignaturen MÜSSEN unterstützt werden.
Eingangsanforderung	keine
Auslöser	Aufruf durch ein Clientsystem (Operation VerifyDocument) oder durch ein Fachmodul im Konnektor
Vorbedingungen	keine
Eingangsdaten	• QES-signiertes Dokument vom Typ QES_DocFormate • QES Signatur(en). Es werden Parallel- und Gegensignaturen unterstützt. • optionale Eingabeparameter (siehe Operation VerifyDocument, Parameter SIG:OptionalInputs) • X.509-Zertifikate (falls diese nicht im signierten Dokument enthalten sind, sondern nur referenziert werden). Für XML-Dokumente: • Liste von XML-Schemata (optional) • includeRevocationInfo [Boolean] - optional; Default: false (Dieser optionale Parameter steuert die Einbettung von OCSP-Responses in die Signatur)
Komponenten	Konnektor

Element	Beschreibung
Ausgangsdaten	• VerificationResult • SIG:OptionalOutput (optional)
Standardablauf	1. **„DocumentValidation“**: Das signierte Dokument wird validiert mit Aufruf TUC_KON_080 „Dokument validieren“{ … }. Treten Fehler bei der Validierung der Typkonformität auf, wenn die Signatur im Dokument eingebettet ist, wird die Prüfung mit einem Fehler abgebrochen. Treten bei der Typkonformität, wenn die Signatur nicht im Dokument eingebettet ist, Fehler auf, so bricht der TUC nicht ab, sondern führt die folgenden Schritte soweit sinnvoll möglich durch. (Die Entscheidung über das sinnvoll Durchführbare liegt beim Hersteller des Konnektors.) 2. **„CoreValidation“**: Es erfolgt die mathematische Prüfung der Signatur, bestehend aus der Prüfung der Hash-Kette bis zum signierten Hashwert und der Prüfung der Signatur unter Verwendung des öffentlichen Schlüssels, des Signaturwertes und des signierten Hashwertes. XML-Signatur: Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation. CMS-Signatur: Die Core Validation erfolgt entsprechend Cryptographic Message Syntax (CMS) Kapitel 5.6 Signature Verification Process [RFC5652]. PDF-Signatur: Die Core Validation erfolgt entsprechend [PAdES-3] Kapitel 4.6 Signature Validation aus PAdES-BES Part 3. Auch wenn die Validierung fehlschlägt, werden die folgenden Prüfschritte durchgeführt, so dass ein vollständiges Prüfprotokoll erstellt werden kann. 3. **„CheckSignatureCertificate“:** **Teil 1: Signaturzertifikat ermitteln** XML-Signatur: Das Signaturzertifikat ist im XMLDSig Element `ds:KeyInfo/ds:X509Data` gespeichert [XMLDSig] oder wird als Eingangsparameter übergeben. CMS-Signatur: Das Signaturzertifikat für CAdES ist im Feld `certificates` im `SignedData` Container gespeichert [CAdES] oder wird als Eingangsparameter übergeben. PDF-Signatur: Das PDF Signaturzertifikat für PAdES ist im Feld SignedData.certificates entsprechend Kapitel 6.1.1 „Placements of the signing certificate“ [PAdES Baseline Profile] gespeichert oder wird als Eingangsparameter übergeben. **Teil 2: Signaturzeitpunkt bestimmen** Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_Eingebettet` wird wie folgt selektiert: XML-Signatur: Das XML element `SigningTime` spezifiziert den Signaturzeitpunkt

Element	Beschreibung
	entsprechend Kapitel 7.2.1 XAdES [XAdES]. CMS-Signatur: Das Attribut `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 11.3 CMS [CMS]. PDF-Signatur: Der Signaturzeitpunkt kann dem M Eintrag des Signature Dictionary entnommen werden [PAdES Baseline Profile] Kapitel 6.2.1 Signing time. Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_Qualifiziert` wird wie folgt selektiert: XML-Signatur: Das XML element `SignatureTimeStamp` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 4.4.3.1 XAdES [XAdES]. CMS-Signatur und PDF-Signatur: Das Attribut `signature-time-stamp` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 6.1 CAdES [CAdES]. Der Signaturzeitpunkt `Benutzerdefinierter_Zeitpunkt` liegt gegebenenfalls als Aufrufparameter vor. Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_System` wird ermittelt. **Teil 3: Signaturzertifikatsprüfung:** Bei der folgenden Signaturzertifikatsprüfung sind die Signaturzeitpunkte gemäß [TIP1-A_5540] zu berücksichtigen. Die Signaturzertifikatsprüfung erfolgt durch Aufruf von TUC_KON_037 „Zertifikat prüfen“ {C.HP.QES; required; Signaturzeitpunkt; true; oid_hba_qes; OCSP; OCSP-Response; getOCSPResponses = includeRevocationInfo~~; Liste der Attributzertifikate~~}. Sind OCSP-Responses in der Signatur eingebettet, ist die jüngsten OCSP-Response des EE-Zertifikats, die für die Zertifikatsprüfung notwendig ist, beim Aufruf von TUC_KON_037 zu übergeben. Sofern der Aufruf von TUC_KON_037 ocspResponses zurückgibt, wird die OCSP-Response des EE-Zertifikats in die Signatur eingebettet. ~~Sofern Attributzertifikate in der Signatur vorhanden sind, werden diese beim Aufruf von TUC_KON_037 übergeben.~~ Auch wenn die Zertifikatsprüfung fehlschlägt, werden die folgenden Prüfungen durchgeführt. 4. **„CheckPolicyConstraints“**: In diesem Schritt wird das signierte Dokument entsprechend der Profilierung der Signaturformate (siehe Anhang B.2) geprüft. Es sind die Vorgaben für die Prüfung von Signaturen aus den Standards für AdES [XAdES], [XAdES Baseline], [CAdES], [CAdES Baseline], [PAdES-3] und [PAdES Baseline] umzusetzen. Dabei sind die Vorgaben aus Tabelle TAB_KON_779 „Profilierung der Signaturformate“ zu erfüllen. Auch wenn nicht alle Anforderungen an das Format des signierten Dokuments erfüllt werden, wird die Prüfung mit den folgenden Schritten fortgesetzt, um ein vollständiges Prüfungsprotokoll zu erhalten.

Element	Beschreibung
	5. Das Prüfergebnis (VerificationResult, OptionalOutput) wird an den Aufrufer zurückgegeben (siehe TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur).
Varianten/Alternativen	Keine
Fehlerfälle	Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_592 Fehlercodes TUC_KON_151 „QES Dokumentensignatur prüfen“ beschrieben. (→ 2 **„CoreValidation“**) Interner Fehler: 4001, Signatur des Dokuments ungültig: 4115 (→3 **„CheckSignatureCertificate“**) Interner Fehler: 4001, Signaturzertifikat ermitteln ist fehlgeschlagen: 4206. (→4 **„CheckPolicyConstraints“**) Interner Fehler: 4001, Dokument nicht konform zu Regeln für QES: 4124, Dokument nicht konform zu Profilierung der Signaturformate: 4208.
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	keine

Tabelle 216: TAB_KON_592 Fehlercodes TUC_KON_151 „QES Dokumentensignatur prüfen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4001	Technical	Error	interner Fehler
4115	Security	Error	Signatur des Dokuments ungültig. Prüfung der Hashwertkette bzw. Prüfung der kryptographischen Signatur fehlgeschlagen.
4124	Technical	Error	Dokument nicht konform zu Regeln für QES
4206	Technical	Error	Signaturzertifikat ermitteln ist fehlgeschlagen
4208	Technical	Error	Dokument nicht konform zu Profilierung der Signaturformate

Das Gesamtergebnis (VerificationResult) für die Prüfung einer Dokumentensignatur fasst die Ergebnisse der obigen Einzelschritte zusammen. Welche nicht erfüllten Kriterien einzelner Prüfschritte zu einem Fehler oder einer Warnung führen, wird durch die Regeln für die QES (siehe Anhang B.2) bestimmt.

Tabelle 217: TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur

VerificationResult pro Signatur		
	Wert	**Bedeutung**
	VALID	Die Signatur wurde gemäß den Regeln für die QES geprüft und für gültig befunden.

	INVALID	Die Signatur des Dokuments ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.
	INCONCLUSIVE	Die Signatur wurde gemäß den Regeln für die QES geprüft. Allerdings konnten einige Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail. Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.

VerificationResult für gesamtes Dokument	
Wert	**Bedeutung**
VALID	Wenn VerificationResult für alle Signaturen zum Dokument VALID
INVALID	Wenn VerificationResult für eine Signatur zum Dokument INVALID
INCONCLUSIVE	in allen anderen Fällen

Das VerificationResult pro Signatur MUSS wie folgt zur Belegung des Elements VerficationReport/IndividualReport/Result korrespondieren:

- wenn
  ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success AND
  ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
- INVALID, wenn
  ResultMajor UNGLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success ODER
  (ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success UND
  ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature)⌫

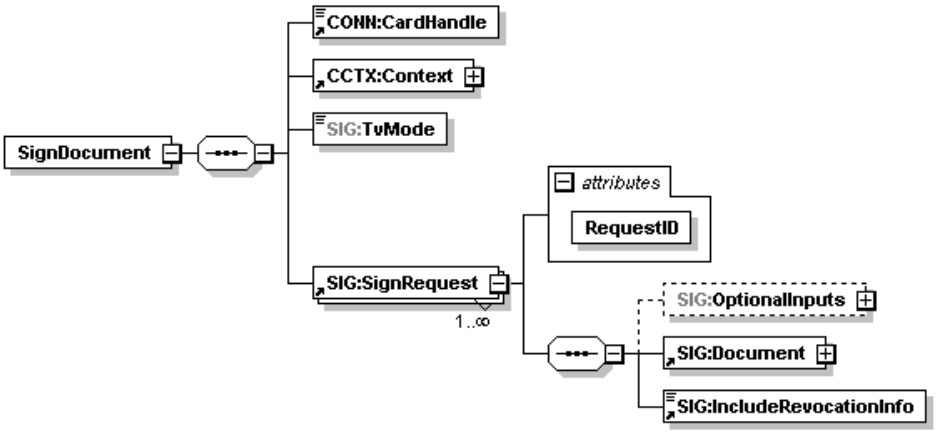
4.1.8.5.1 SignDocument (nonQES und QES)

⌦ **TIP1-A_5010 Operation SignDocument (nonQES und QES)**

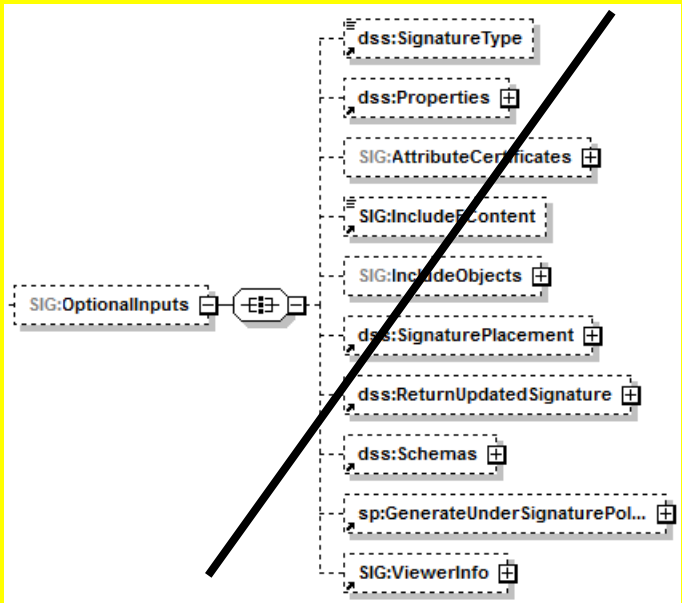
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation SignDocument anbieten**.**

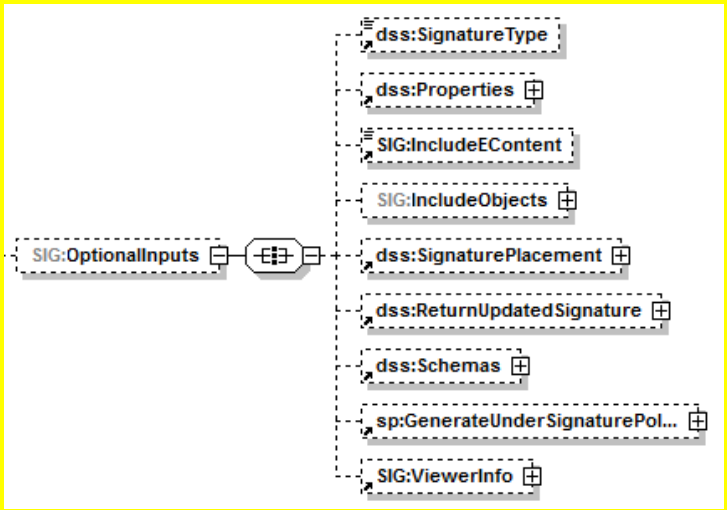
Tabelle 5: TAB_KON_065 Operation SignDocument (nonQES und QES)

Name	`SignDocument`
Beschreibung	Diese Operation lehnt sich an [OASIS-DSS] an. Sie enthält voneinander unabhängige SignRequests. Jeder SignRequest erzeugt eine Signatur für ein Dokument. Für die qualifizierte elektronische Signatur (QES) werden die `QES_DocFormate` unterstützt. Für nicht-qualifizierte elektronische Signaturen (nonQES) werden die `nonQES_DocFormate` unterstützt. Zur Signaturerzeugung werden Schlüssel und Zertifikate einer Chipkarte benutzt. Unterstützte Karten sind für die QES der HBAx mit dem QES-Zertifikat. Für die nonQES wird für die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur“ die SM-B mit dem OSIG–Zertifikat unterstützt. Bei der Erstellung von XML-Signaturen MUSS Canonical XML 1.1 verwendet werden [CanonXML1.1]. Es soll der Common-PKI-Standard eingesetzt werden, siehe [Common-PKI]. In Summe für die Größe der Dokumente in allen SignRequests innerhalb einer SignDocument-Anfrage MUSS der Konnektor eine Gesamtgröße von <= 250 MB

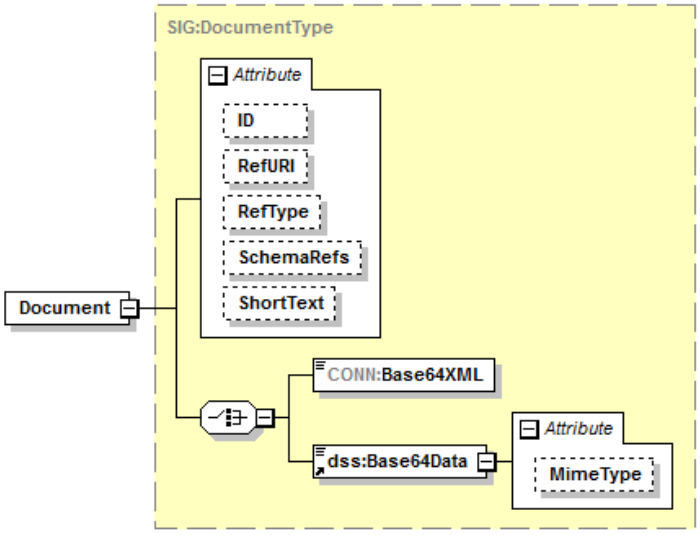
<table>
<tr><td></td><td>unterstützen.</td></tr>
<tr><td>Aufruf-parameter</td><td>


<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td><code>CONN:CardHandle</code></td><td>Identifiziert die zu verwendende Signaturkarte.
Die Operation DARF die Signatur mit der eGK NICHT unterstützen. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4126 abbrechen.</td></tr>
<tr><td><code>CCTX:Context</code></td><td>Aufrufkontext QES mit HBAx:
MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend
Aufrufkontext nonQES mit SM-B:
MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet</td></tr>
<tr><td><code>TvMode</code></td><td>Der Parameter wird im Konnektor nicht ausgewertet.</td></tr>
<tr><td><code>SIG:JobNumber</code></td><td>Die Nummer des Jobs, unter der der nächste Signaturvorgang gestartet wird.
Parameter muss gefüllt werden.</td></tr>
<tr><td><code>SIG:SignRequest</code></td><td>Ein <code>SignRequest</code> kapselt den Signaturauftrag für ein Dokument.
Das verpflichtende XML-Attribut <code>RequestID</code> identifiziert einen <code>SignRequest</code> innerhalb eines Stapels von <code>SignRequests</code> eindeutig. Es dient der Zuordnung der <code>SignResponse</code> zum jeweiligen <code>SignRequest</code>.</td></tr>
<tr><td><code>SIG:OptionalInputs</code></td><td>Enthält optionale Eingangsparameter (angelehnt an <code>dss:OptionalInputs</code> gemäß [OASIS-DSS] Section 2.7):</td></tr>
</table>
</td></tr>
</table>





`SIG:Document`

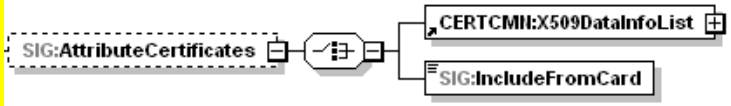


Dieses an das `dss:Document` Element aus [OASIS-DSS]

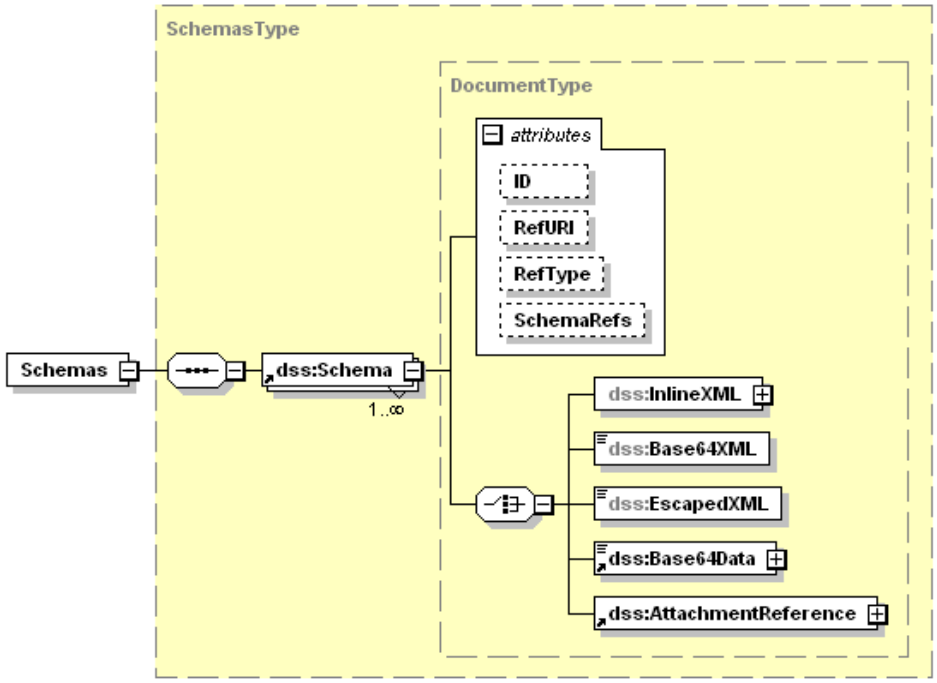
		Section 2.4.2 angelehnte Element enthält das zu signierende Dokument, wobei die Kindelemente `CONN:Base64XML` und `dss:Base64Data` auftreten können. Bei den als `dss:Base64Data` übergebenen Dokumenten werden folgende (Klassen von) MIME-Types unterschieden: ”application/pdf-a” – für PDF/A-Dokumente, ”text/plain”, ”text/plain; charset=iso-8859-15” oder ”text/plain; charset=utf-8” – für Text-Dokumente, ”image/tiff” – für TIFF-Dokumente und ein beliebiger anderer MIME-Type für nicht näher unterschiedene Binärdaten des spezifizierten Typs. Der MIME-Type „text/plain“ wird interpretiert als „text/plain; charset=iso-8859-15”. Das Element enthält ein Attribut `ShortText`. Es muss für QES-Signaturen bei jedem Aufruf vom Clientsystem übergeben werden, für nonQES-Signaturen ist es optional. Über das Attribut `RefURI` kann gemäß [OASIS-DSS] (Abschnitt 2.4.1) ein zu signierender Teilbaum eines XML-Dokuments ausgewählt werden.
	`SIG:IncludeRevocationInfo`	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturerstellung vorliegenden Sperrinformationen anfordern. Es wird ausschließlich die zu erstellende Signatur betrachtet, d.h. es erfolgt keine Einbettung von Sperrinformationen für bereits enthaltene Signaturen. Für nicht-qualifizierte elektronische Signaturen (nonQES) wird diese Funktionalität nicht unterstützt.
	`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element kann der generelle Typ der zu erzeugenden Signaturen spezifiziert werden. Hierbei MÜSSEN folgende Signaturtypen unterstützt werden: **XML-Signatur** Durch Übergabe der URI urn:ietf:rfc:3275 wird die Erstellung von XML-Signaturen gemäß [RFC3275], [XMLDSig] angestoßen. Das zu verwendende Profil ist XAdES-BES ([XAdES]). Die Rückgabe einer solchen Signatur erfolgt als `ds:Signature`-Element. **CMS-Signatur** Durch Übergabe der URI urn:ietf:rfc:5652 wird eine CMS-Signatur gemäß [RFC5652] angestoßen. Das zu verwendende Profil ist CAdES-BES ([CAdES]). Die Signatur wird als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert. **S/MIME-Signatur** Durch Übergabe der URI „urn:ietf:rfc:5751” wird eine S/MIME-Signatur gemäß [RFC5751] angestoßen. Die CMS-Signatur der übergebenen MIME-Nachricht erfolgt konform der Vorgaben zur CMS-Signatur. Das Rückgabedokument ist eine MIME-Nachricht vom Typ „application/pkcs7-mime“ mit einer CMS-Struktur vom Typ SignedData. Ist das übergebene Dokument keine MIME-Nachricht, so wie

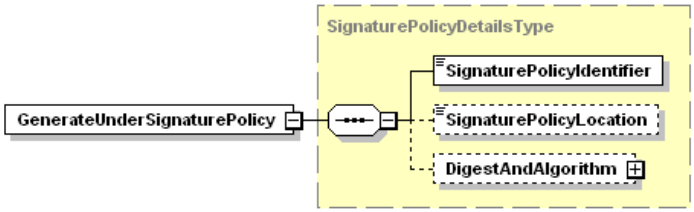
		der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. **PDF-Signatur** Durch Übergabe der URI http://uri.etsi.org/02778/3 wird die Erzeugung einer PAdES-Basic Signatur gemäß [PAdES-3]angestoßen, wobei das Dokument mit der integrierten Signatur als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert wird. Handelt es sich beim übergebenen Dokument nicht um ein `Base64Data`-Element mit MIME-Type „application/pdf-a“, so wird ein Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder Signaturvariante). Die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur“ DÜRFEN für QES der HBAx nur mit dem QES-Zertifikat erfolgen, für nonQES nur mit dem OSIG-Zertifikat der SM-B. In jedem diese Anforderung verletzenden Fall MUSS der Fehler 4058 (Aufruf nicht zulässig) zurückgeliefert werden. Fehlt dieses Element, so wird der Signaturtyp gemäß TAB_KON_583 – Default-Signaturverfahren aus dem Dokumententyp abgeleitet.

	`dss:Properties`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.5) definierte Element können zusätzliche signierte und unsignierte Eigenschaften (Properties) bzw. Attribute in die Signatur eingefügt werden.
		Unterstützt werden genau folgende Attribute:
		Im CMS-Fall (SignatureType = urn:ietf:rfc:5652) kann es XML-Elemente
		`./SignedProperties/Property/Value/CMSAttribute`
		und
		`./UnsignedProperties/Property/Value/CMSAttribute`
		enthalten. Ein solches XML-Element `CMSAttribute` muss ein vollständiges, base64/DER-kodiertes ASN.1-`Attribute` enthalten, definiert in [CMS#5.3.SignerInfo Type]. Es muss bei der Erstellung des CMS-Containers unverändert unter `SignedAttributes bzw. UnsignedAttributes` aufgenommen werden.

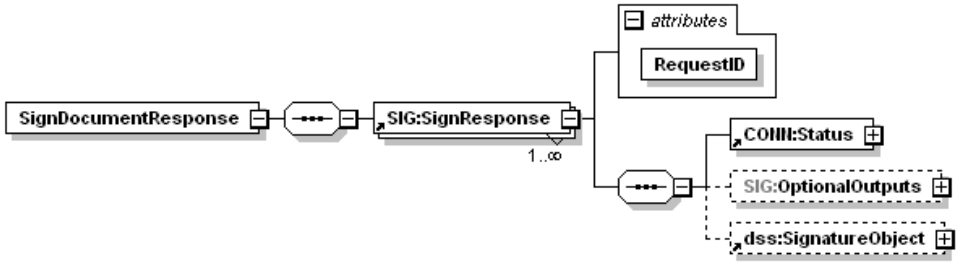
	~~`SIG:AttributeCertificates`~~	 ~~Dieses Element ist nur im Fall einer QES-Signatur relevant. Ist dieses Element nicht vorhanden, werden keine Attributzertifikate eingebetttet.~~ ~~Ist dieses Element vorhanden, können über das Element CERTCMN:X509DataInfoList Attributzertifikate beim Aufruf in dem Format mitgegeben werden, wie es die Operation ReadCardCertificate des Zertifikatsdienstes liefert.~~ ~~Alternativ kann der Konnektor über das Element SIG:IncludeFromCard angewiesen werden, alle Attributzertifikate zum Basiszertifikat von der signierenden Karte zu lesen.~~

		~~Die Zertifikate bettet der Konnektor für XAdES unter `ds:Signature/ds:Object/QualifyingProperties/SignedProperties/SignedSignatureProperties/SignerRole/CertifiedRoles/CertifiedRole` in die Signatur ein und für CAdES und PAdES unter `SignedData/signerInfos/SignerInfo/SignedAttrs/SignerAttributes/CertifiedAttributes` Es muss mindestens eine Anzahl von 10 Attributzertifikaten unterstützt werden.~~
	`SIG:IncludeEContent`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.7), definierte Element kann bei einer CMS-basierten Signatur das Einfügen des signierten Dokumentes in die Signatur angefordert werden. Die Verwendung dieses Parameters bei anderen Signaturtypen führt zu einem Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante).
	`SIG:IncludeObject`	Dieses Element enthält zum Anfordern einer Enveloping XML Signatur ein dss:IncludeObject-Element gemäß [OASIS-DSS] (Abschnitt 3.5.6). Ist das Element vorhanden und ein anderer Signaturtyp als eine XML-Signatur angefordert, so wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.
	`dss:SignaturePlacement`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.8) definierte Element kann bei XML-basierten Signaturen gemäß [RFC3275] die Platzierung der Signatur im Dokument angegeben werden. Bei anderen Signaturtypen wird das Element ignoriert und eine Warnung (Fehlercode 4197, Parameter SignaturePlacement wurde ignoriert) zurückgeliefert.
	`dss:ReturnUpdatedSignature`	Durch dieses in [OASIS-DSS] (Abschnitt 4.5.8) definierte Element kann eine übergegebene XML- oder CMS-Signatur mit zusätzlichen Informationen und Signaturen (Parallel- und Gegensignaturen) versehen werden. Hierbei sind folgende Ausprägungen für das `Type`-Attribut vorgesehen: • http://ws.gematik.de/conn/sig/sigupdate/parallel Hierdurch wird eine Parallelsignatur zu einer bereits existierenden Signatur erzeugt und entsprechend zurückgeliefert. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentincluding Hierdurch wird eine dokumentinkludierende Gegensignatur für das Dokument und alle vorhandenen parallelen Signaturen erzeugt. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt. Bei anderen `Type`-Attributen wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.
	`dss:Schemas`	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schemata übergeben werden, die zur Validierung der übergebenen XML-Dokumente verwendet werden können.

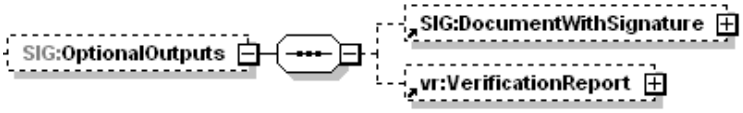
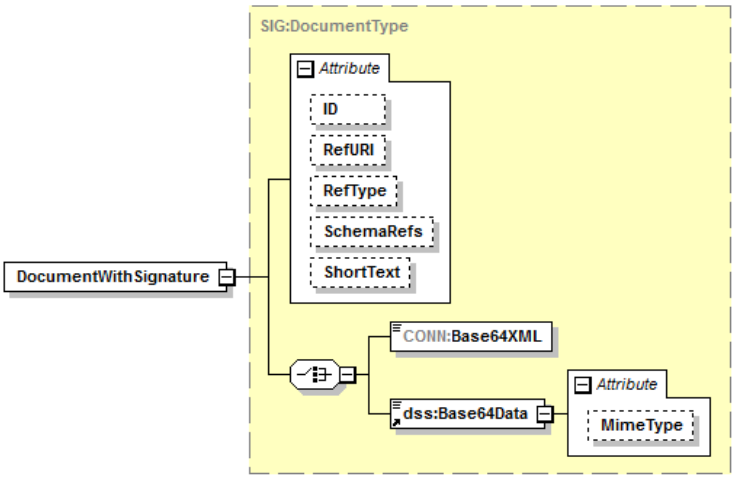


`dss:Schema`	Dieses Element enthält ein XML-Schema zur Validierung des übergebenen XML-Dokuments. Das Attribut `RefURI` ist verpflichtend. Es kennzeichnet dabei den Namensraum des XML-Schemas entsprechend [OASIS-DSS] (Abschnitt 2.8.5)
`sp:GenerateUnderSignaturePolicy`	 Über dieses in [OASIS-SP], Kapitel 2.2.1.1.1.1 Optional Input <GenerateUnderSignaturePolicy>, definierte Element wird die erforderliche Singnaturrichlinie ausgewählt. Die im Element `sp:SignaturePolicyIdentifier` übergebene URI identifiziert die Signaturrichtlinie.
`SIG:ViewerInfo`	Enthält optional die vom Konnektor in die Signatur einzubeziehende Referenzen für die Stylesheets zur Anzeige.

Rückgabe



`SIG:SignResponse`	Eine `SignResponse` kapselt den ausgeführten Signaturauftrag pro Dokument. Die Zuordnung zwischen

		SignRequest und SignResponse erfolgt über die RequestID.
	CONN:Status	Enthält den Status der ausgeführten Operation pro SignRequest.
	SIG:OptionalOutputs	Enthält (angelehnt an dss:OptionalOutputs) optionale Ausgangsparameter: 
	SIG:DocumentWithSignature	 Pro SignResponse wird ein Element SIG:DocumentWithSignature gemäß [OASIS-DSS] (Abschnitt 3.5.8) zurückgeliefert, in dem das Dokument mit Signatur enthalten ist. Dabei werden die XML-Attribute des Elements SIG:Document auf dem zugehörigen SignRequest übernommen. Ist die Signatur nicht im Dokument enthalten, wird ein leeres Element Base64XML oder Base64Data zurückgegeben. Die Signatur wird dann im Element dss:SignatureObject abgelegt. Wenn die Signatur im Dokument enthalten ist, wird das signierte Dokument im Feld Base64XML bzw. Base64Data zurückgeliefert. In diesem Fall MUSS die dss:SignaturePtr-Alternative in dss:SignatureObject (vgl. [OASIS-DSS] Abschnitt 2.5) dazu genutzt werden, auf die in den Dokumenten enthaltenen Signaturen zu verweisen.
	vr:VerificationReport	Vom Konnektor nicht befüllt.
	dss:SignatureObject	Enthält im Erfolgsfall die erzeugte Signatur pro SignRequest in Form eines dss:SignatureObject-Elementes gemäß [OASIS-DSS] (Abschnitt 3.2).
Vorbe-dingungen	Keine	
Nachbe-	Keine	

dingungen	

4.1.9.4.1 TUC_KON_037 ”Zertifikat prüfen”

⌦ TIP1-A_4696 TUC_KON_037 „Zertifikat prüfen“

Der Konnektor MUSS den technischen Use Case „Zertifikat prüfen“ gemäß TUC_KON_037 „Zertifikat prüfen“ umsetzen.⌫

Tabelle 6: TAB_KON_769 TUC_KON_037 „Zertifikat prüfen“

Element	Beschreibung
Name	TUC_KON_037 „Zertifikat prüfen“
Beschreibung	Der TUC beschreibt • die Prüfung der Gültigkeit eines CV-Zertifikates, indem die Korrektheit der Signatur des CV- und des CVC-CA-Zertifikates geprüft wird. • die Prüfung eines X.509- Zertifikats gegen den Vertrauensraum
Auslöser	• Aufruf in einem Fachmodul oder • technischen Use Case
Vorbedingungen	• aktuelle TSL-Informationen im Truststore vorhanden • für QES X.509-Prüfung: eine aktuell gültige BNetzA-VL
Eingangsdaten	• CV-Zertifikat und der öffentliche Schlüssel der zugehörigen ausstellenden CVC-CA oder X.509-Zertifikat • Bei X.509-Prüfung: o QUALIFIED={not_required|required| if_QC_present} o Referenzzeitpunkt: Zeitpunkt, für den das Zertifikat geprüft werden soll (optional; bei Nichtangabe Verwendung der Systemzeit des Konnektors) o OFFLINE_ALLOW_NOCHECK. (true/false; Default: false) o PolicyList: Zugelassene Zertifikatstyp-OID gemäß [gemSpec_OID#GS-A_4445] o Nur für nonQES-Zertifikate: ▪ Vorgesehene KeyUsage (intendedKeyUsage) ▪ Vorgesehene ExtendedKeyUsage (intendedExtendedKeyUsage) ▪ Grace Period: maximal zulässiger Zeitraum, den letzte OCSP-Antwort aus dem Cache bezüglich des Referenzzeitpunkts zurückliegen darf (optional; Default-Wert CERT_OCSP_DEFAULT_GRACE_PERIOD_NONQES ▪ Prüfmodus: • OCSP: Es wird mittels OCSP geprüft. Dabei wird, falls die Grace Period noch nicht abgelaufen ist, die OCSP-Antworten aus dem Cache des Konnektors verwendet. • CRL: Es wird gegen die aktuelle CRL auf dem

Element	Beschreibung
	Konnektor geprüft. • NONE: Keine Prüfung von Statusinformationen o OCSP-Response - optional; o getOCSPResponses [Boolean] - optional; Default: false (liefert die Information, ob die OCSP-Antwort des geprüften Zertifikats an den Aufrufer zurückzugegeben ist) ~~o Liste von Attributzertifikaten (optional, QES)~~
Komponenten	Konnektor
Ausgangsdaten	• Status und Liste von Warnungen/Fehlern bei der Zertifikatsprüfung • Ermittelte Rolle (X.509-Zertifikate) Werte ermittelt aus dem Zertifikat aus „Tab_PKI_406 OID-Festlegung technische Rolle in X.509-Zertifikaten“ oder „Tab_PKI_402 OID-Festlegung Rolle im X.509-Zertifikat für Berufsgruppen“ oder Tab_PKI_403 OID-Festlegung Institutionen im X.509-Zertifikat der SMC-B [gemSpec_OID] • QCStatements des Zertifikats • ocspResponsesRenewed – optional / verpflichtend, wenn Eingabeparameter getOCSPResponses = true
Standardablauf	1. Abhängig davon, ob ein CV-Zertifikat oder ein X.509-Zertifikat geprüft werden soll, werden folgende Schritte ausgeführt: A. CV-Zertifikat: CV-Zertifikate sind entsprechend [gemSpec_PKI#8.7 “Zertifikatsprüfung CVC“] und [gemSpec_PKI#8.8 “Zertifikatsprüfung CV-Zertifikate der 2. Generation“] zu prüfen. i. Die Signatur des CVC-CA-Zertifikats (von der zu prüfenden Karte) wird mit dem öffentlichen Root-Schlüssel der ausstellenden CVC-Root-CA geprüft. Der benötigte Root-Key für G2 befindet sich im Truststore des Konnektors und für G1+ auf der gSMC-K in der Datei EF.PuK.RCA.CS.R2048. ii. Die Signatur des CV-Zertifikats wird mit dem öffentlichen Schlüssel der ausstellenden CVC-CA (per Message Recovery in Schritt i aus dem CVC-CA-Zertifikat gewonnen) geprüft. B. X.509-Zertifikat: B.1 Wenn das X.509-Zertifikat von einem CA-Zertifikat in CERT_IMPORTED_CA_LIST ausgestellt wurde, erfolgt eine Zertifikatsprüfung analog zu den Festlegungen in TUC_PKI_018 “Zertifikatsprüfung”. Dabei sind zu prüfen: Zeitliche Gültigkeit, mathematische Prüfung der Zertifikatssignatur, die Prüfung der Zweckbindung gemäß der im Zertifikat hinterlegten keyUsage. TSL-bezogene Prüfungen im TUC_PKI_018 werden in diesem Fall nicht durchgeführt. Ebenso erfolgt keine

Element	Beschreibung
	OCSP-Prüfung. B.2 Wenn das zum X.509-Zertifikat gehörende CA-Zertifikat nicht in CERT_IMPORTED_CA_LIST enthalten ist, werden, abhängig vom Parameter QUALIFIED, folgende TUCs unter Weitergabe aller Eingangsparameter sowie der Negation des Werts von MGM_LU_ONLINE als Parameter „Offline-Modus“ aufgerufen: a) Für QUALIFIED=not_required: TUC_PKI_018 “Zertifikatsprüfung in der TI” Ist der Eingangsparameter ocspResponses mit einer OCSP-Antwort gefüllt, so wird dieser übergeben. Die aktuell aus der OCSP-Abfrage resultierte OCSP-Antwort, falls vorhanden, wird an den Aufrufer weitergegeben. b) Für QUALIFIED=required: TUC_PKI_030 „QES-Zertifikatsprüfung“ Dabei wird ~~werden~~ das Basiszertifikat ~~und die Attributzertifikate~~ übergeben. Ist Eingangsparameter ocspResponses mit einer OCSP-Response gefüllt, so wird diese übergeben. Die aktuell aus der OCSP-Abfrage resultierende OCSP-Response, falls vorhanden, wird an den Aufrufer weitergegeben. c) Für QUALIFIED=if_QC_present: Ist im jeweiligen Signaturzertifikat mindestens ein QCStatement mit dem OID id-etsi-qcs-QcCompliance (0.4.0.1862.1.1) enthalten, handelt es sich um eine QES-Zertifikatsprüfung mittels TUC_PKI_030 „QES-Zertifikatsprüfung“, sonst um eine nonQES-Zertifikatsprüfung mittels TUC_PKI_018 „Zertifikatsprüfung“ Als Timeout wird beim Aufruf von TUC_PKI_018 der Wert von CERT_OCSP_TIMEOUT_NONQES bzw. beim Aufruf von TUC_PKI_030 der Wert von CERT_OCSP_TIMEOUT_QES übergeben (siehe auch Eingangsdaten von diesen TUCs in [gemSpec_PKI]). Als TOLERATE_OCSP_FAILURE wird OFFLINE_ALLOW_NOCHECK verwandt. Für die QES-Zertifikatsprüfung wird das zu prüfende QES-Zertifikat an TUC_PKI_030 „QES-Zertifikatsprüfung“ übergeben. Wird im Aufruf der Eingangsparameter getOCSPResponses=false mit übergeben, wird keine OCSP-Response an den Aufrufer zurückgegeben. Wenn der Eingangsparameter „Prüfmodus“ den Wert NONE hat, werden die TUC_PKI_018 Eingangsparameter • „Offline-Modus“ unabhängig von

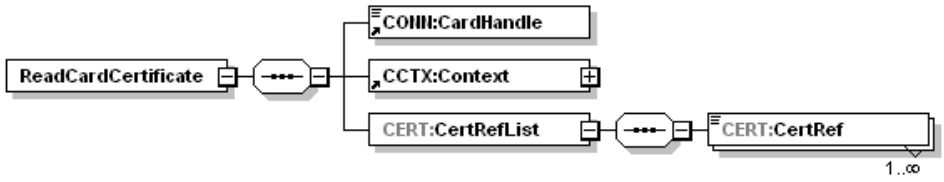
Element	Beschreibung
	MGM_LU_ONLINE auf „ja“ gesetzt und • „Prüfmodus“ auf „OCSP“. Der Status der Prüfung wird zurückgegeben.
Varianten/Alternativen	
Fehlerfälle	(→1.A.) Fehler bei der CV-Zertifikatsprüfung: Fehlercode 4196.
Nichtfunktionale Anforderungen	Der Konnektor MUSS unter Einhaltung aller anderen Anforderungen an die Zertifikatsprüfung die Anzahl der OCSP-Abfragen minimieren. Dies MUSS durch Caching (unter Berücksichtigung der Grace Period) und DARF NICHT durch Bündelung von OCSP-Anfragen geschehen.
Zugehörige Diagramme	keine

4.1.9.5.2 ReadCardCertificate

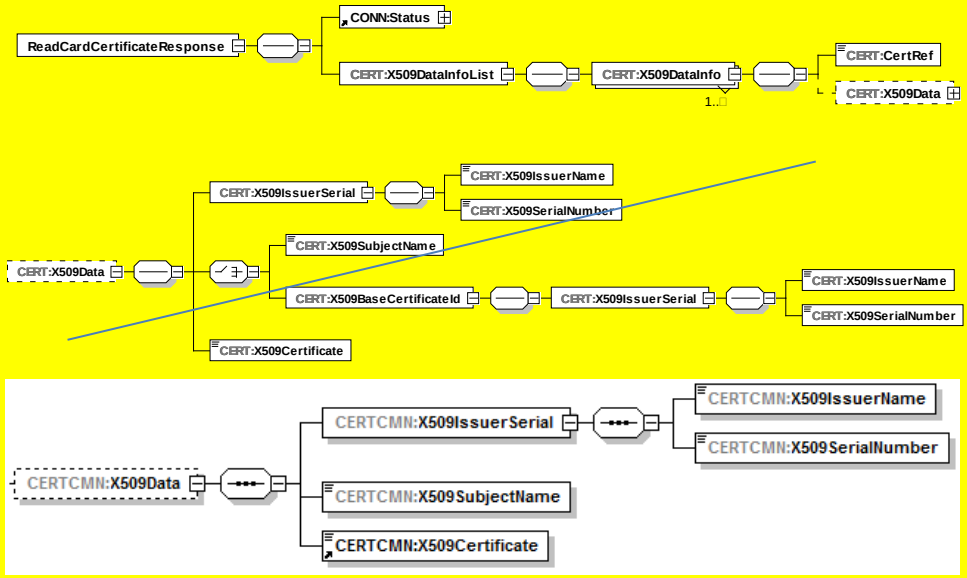
TIP1-A_4700 Operation ReadCardCertificate

Die Basisanwendung Zertifikatsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation ReadCardCertificate wie in Tabelle 7: TAB_KON_678 Operation ReadCardCertificate beschrieben anbieten**.**

Tabelle 7: TAB_KON_678 Operation ReadCardCertificate

<table>
<tr><td>Name</td><td colspan="2">ReadCardCertificate</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Liest X.509-Zertifikate von einer Karte.</td></tr>
<tr><td>Aufruf-parameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>CardHandle</td><td>Gibt die Karte an, von der das Zertifikat gelesen werden soll.
Es können Zertifikate von HBAx (HBA, HBA-VK), SM-B ausgelesen werden.
Die Operation ReadCardCertificate DARF das Lesen von Zertifikaten der eGK NICHT unterstützen.</td></tr>
<tr><td></td><td>Context</td><td>Aufrufkontext (Mandant)</td></tr>
<tr><td></td><td>CertRefList</td><td>Gibt an, welche(s) Zertifikat(e) gelesen werden soll. Mögliche Werte für CertRef sind:

C.AUT, C.ENC, C.SIG, C.QES, ~~C.QES-AC1, C.QES-AC2, C.QES-AC3~~</td></tr>
</table>

<table>
<tr><td rowspan="8">Rückgabe</td><td colspan="3"> 
</td></tr>
<tr><td>Status</td><td colspan="2">Enthält den Ausführungsstatus der Operation.</td></tr>
<tr><td>CertRef</td><td colspan="2">Dieses Element beinhaltet die Referenz des Zertifikats, welches bei der Anfrage übergeben wurde.</td></tr>
<tr><td>X509Data</td><td colspan="2">Inhalt des über die CertRef referenzierten Zertifikats. Ist das referenzierte Zertifikat nicht vorhanden, so wird dieses Element nicht vom Konnektor gefüllt.</td></tr>
<tr><td></td><td>X509IssuerName</td><td>Enthält den Issuer-Name des Zertifikats. Bezüglich des Encodings sind die in [XMLDSig#4.4.4.4.1] angegebenen Regeln zu beachten (Escaping von Sonderzeichen etc.)</td></tr>
<tr><td></td><td>X509SerialNumber</td><td>Enthält die serialNumber des Zertifikats.</td></tr>
<tr><td></td><td>X509SubjectName</td><td>Enthält ~~im Falle eines Public-Key-Zertifikats~~ das Feld subject.CommonName. Bezüglich des Encodings sind die in [XML DSig#4.4.4.4.1] angegebenen Regeln zu beachten (Escaping von Sonderzeichen etc.)</td></tr>
<tr><td></td><td>~~X509BaseCertificateId~~</td><td>~~Enthält im Falle eines Attributzertifikats Issuer-Name und Seriennummer des Basiszertifikates, falls vorhanden.~~</td></tr>
<tr><td></td><td>X509Certificate</td><td>Enthält das base64-codierte Zertifikat, dessen Binärstruktur wiederum ASN.1-codiert (gemäß [COMMON_PKI]) vorliegt.</td></tr>
<tr><td>Vorbe-dingungen</td><td colspan="3">Keine</td></tr>
<tr><td>Nachbe-dingungen</td><td colspan="3">Keine</td></tr>
</table>

Der Ablauf der Operation ReadCardCertificate ist in Tabelle 8: TAB_KON_679 Ablauf ReadCardCertificate beschrieben:

Tabelle 8: TAB_KON_679 Ablauf ReadCardCertificate

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab. Wurde als Zielkarte eine eGK adressiert, wird Fehlercode 4090 zurückgeliefert.

2.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle
		}
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 {
		mandatId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		cardHandle = $context.cardHandle;
		userId = $context.userId }

4.	getEF	Für jedes Paar von CertRef und CardHandle wird gemäß folgender Tabelle das zu lesende File (EF) bestimmt: Referenz EF C.AUT HBA-VK: EF.C.HP.AUT HBA: EF.C.HP.AUT.R2048 SM-B: EF.C.HCI.AUT C.ENC HBA-VK: EF.C.HP.ENC HBA: EF.C.HP.ENC.2048 SM-B: EF.C.HCI.ENC.R2048 C.SIG SM-B: EF.C.HCI.OSIG.R2048 C.QES HBA-VK: EF.C.HP.QES HBA: EF.C.HP.QES.R2048 ~~C.QES-AC1 HBAx: EF.C.HP.QES-AC1~~ ~~C.QES-AC2 HBAx: EF.C.HP.QES-AC2~~ ~~C.QES-AC3 HBAx: EF.C.HP.QES-AC3~~ Ist die übergebene Zertifikatsreferenz ungültig, wird Fehlercode 4149 zurückgegeben. Das Lesen von Zertifikaten der eGK ist aus Sicherheitsgründen für Clientsysteme nicht zulässig.
5.	TUC_KON_216 „LeseZertifikat“	Für jedes Paar von CardHandle und EF wird nun durch Aufruf von TUC_KON_216 „LeseZertifikat“ das Zertifikat ausgelesen.

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
6.	Zertifikatsattribute extrahieren	Aus jedem Zertifikat werden die zu liefernden Attribute extrahiert. Die Ergebnisstruktur wird mit den erhaltenen Rückgabewerten gefüllt.

Tabelle 9: TAB_KON_604 Fehlercodes „ReadCardCertificate“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4149	Technical	Error	Ungültige Zertifikatsreferenz
4090	Security	Error	Zugriff auf eGK nicht gestattet

⌫

B3 – Profilierung VerificationReport

~~**Signatur mit Attributzertifikaten**~~

~~Es wird für jedes Attributzertifikat, das von der Signatur umfasst wird, ein Element~~

~~/VerificationReport/~~
~~IndividualReport/~~
~~SignedObjectIdentifier/~~
~~SignedProperties/~~
~~SignedSignatureProperties/~~
~~SignerRole/~~
~~CertifiedRoles/~~
~~AttributeCertificateValidity~~

~~angelegt. Attributzertifikate außerhalb der Signatur werden in einem eigenen IndividualReport behandelt, bei allDetails mit IndividualAttributeCertificateReport.~~

Anhang D - Übersicht über die verwendeten Versionen

XSD Name	SignatureService.xsd
XSD Schemaversion	~~7.4.1~~ 7.4.2
TargetNamespace	http://ws.gematik.de/conn/SignatureService/v7.4

XSD Name	CertificateServiceCommon.xsd
XSD Schemaversion	~~2.0.0~~ 2.0.1
TargetNamespace	http://ws.gematik.de/conn/CertificateServiceCommon/2.0

SignatureService.xsd

```
<schema ... version="7.4.17.4.2">
  …
<element name="SignRequest">
      <complexType>
        <sequence>
          <element name="OptionalInputs" minOccurs="0">
            <complexType>
              <all>
                <element ref="dss:SignatureType" minOccurs="0"/>
                <element ref="dss:Properties" minOccurs="0"/>
                <element name="AttributeCertificates" minOccurs="0">
                  <complexType>
                    <choice>
                      <element ref="CERTCMN:X509DataInfoList"/>
                      <element name="IncludeFromCard" type="boolean"/>
                    </choice>
                  </complexType>
                </element>
                <element ref="SIG:IncludeEContent" minOccurs="0"/>
                <element name="IncludeObjects" minOccurs="0">
                  <complexType>
                    <sequence>
                      <element ref="dss:IncludeObject" maxOccurs="unbounded"/>
                    </sequence>
                  </complexType>
                </element>
                <element ref="dss:SignaturePlacement" minOccurs="0"/>
                <element ref="dss:ReturnUpdatedSignature" minOccurs="0"/>
                <element ref="dss:Schemas" minOccurs="0"/>
                <element ref="sp:GenerateUnderSignaturePolicy" minOccurs="0"/>
                <element ref="SIG:ViewerInfo" minOccurs="0"/>
              </all>
            </complexType>
          </element>
          <element ref="SIG:Document"/>
          <element ref="SIG:IncludeRevocationInfo"/>
        </sequence>
        <attribute name="RequestID" type="string" use="required"/>
      </complexType>
    </element>
```

CertificateServiceCommon.xsd

```
<schema … version="2.0.02.0.1"    >

   <simpleType name="CertRefEnum">
      <restriction base="token">
         <enumeration value="C.AUT"/>
         <enumeration value="C.ENC"/>
         <enumeration value="C.SIG"/>
         <enumeration value="C.QES"/>
         <enumeration value="C.QES-AC1"/>
         <enumeration value="C.QES-AC2"/>
         <enumeration value="C.QES-AC3"/>
      </restriction>
```

```
</simpleType>

<complexType name="X509DataInfoListType">
  <sequence>
    <element name="X509DataInfo" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="CertRef" type="CERTCMN:CertRefEnum"/>
          <element name="X509Data" minOccurs="0">
            <complexType>
              <sequence>
                <element name="X509IssuerSerial">
                  <complexType>
                    <sequence>
                      <element name="X509IssuerName" type="string"/>
                      <element name="X509SerialNumber" type="string"/>
                    </sequence>
                  </complexType>
                </element>
                <choice>
                  <element name="X509SubjectName" type="string"/>
                  <element name="X509BaseCertificateId">
                    <complexType>
                      <sequence>
                        <element name="X509IssuerSerial">
                          <complexType>
                            <sequence>
                              <element name="X509IssuerName" type="string"/>
                              <element name="X509SerialNumber" type="string"/>
                            </sequence>
                          </complexType>
                        </element>
                      </sequence>
                    </complexType>
                  </element>
                </choice>
                <element ref="CERTCMN:X509Certificate"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
```

Änderungen in gemSpec_HBA_ObjSys_G2.1

Die Attributzertifikate C.HP.QES-AC1, C.HP.QES-AC2 und C.HP.QES-AC3 in den Ausprägungen R2048 und E256 werden durch die Telematikinfrastruktur nicht mehr unterstützt.

Die vorhandene Beschreibung der Objekte zur Ablage dieser Attributzertifikate auf dem HBA der Generation G2.1 soll aus der Spezifikation des Objektssystems des HBA G2.1 entfernt werden.

Die Spezifikation gemSpec_HBA_ObjSys_G2.1 soll wie folgt geändert werden:

1. Entfernen HBA-Objekte für Attributzertifikate RSA

Das Kapitel ”5.5.2.5MF / DF.QES / EF.C.HP.QES-AC1, MF / DF.QES / EF.C.HP.QES-AC2 und MF / DF.QES / EF.C.HP.QES-AC3” und alle enthaltenen Unterkapitel werden vollständig entfernt.

~~5.5.2.5 MF / DF.QES / EF.C.HP.QES-AC1, MF / DF.QES / EF.C.HP.QES-AC2 und MF / DF.QES / EF.C.HP.QES-AC3~~

2. Entfernen HBA-Objekte für Attributzertifikate ELC

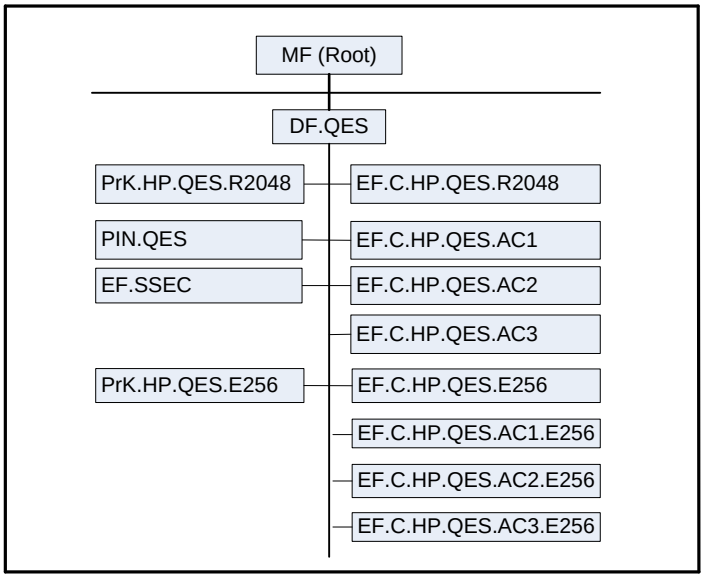
Das Kapitel ” 5.5.2.8 MF / DF.QES / EF.C.HP.QES-AC1.E256, MF / DF.QES / EF.C.HP.QES-AC2.E256 und MF / DF.QES / EF.C.HP.QES-AC3.E256” und alle enthaltenen Unterkapitel werden vollständig entfernt

~~5.5.2.8 MF / DF.QES / EF.C.HP.QES-AC1.E256, MF / DF.QES / EF.C.HP.QES-AC2.E256 und MF / DF.QES / EF.C.HP.QES-AC3.E256~~

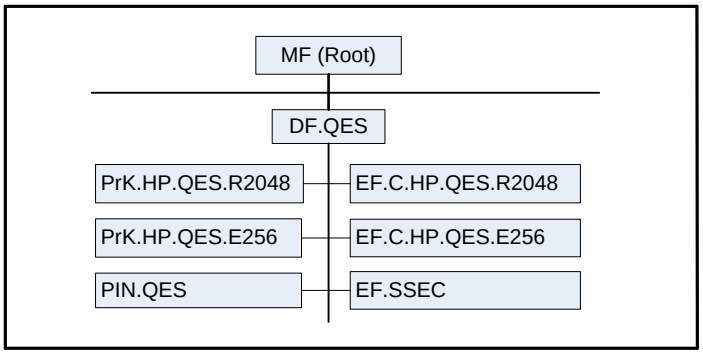
3. Anpassung der Strukturgrafik DF.QES

In Kapitel 5.5.1 ”Dateistruktur und Dateiinhalt” wird die Abbildung Abb_HBA_ObjSys_003 wie folgt geändert.

Bisher:



Neu:



4. Verweis auf Attributzertifikate im Text entfernen

In Kapitel 5.5.1 wird die Beschreibung wie folgt geändert:

Die QES-Anwendung beinhaltet~~besitzt~~ EFs für die~~das~~ X.509-QES-Zertifikate ~~und maximal drei Attributzertifikate, jeweils~~ für die Kryptographie mit RSA und mit elliptischen Kurven. Zusätzlich ist ein EF zur Anzeige des unterstützten Maximalwertes des SSEC angelegt.

5. Anpassung Größe und Inhalt EF.CD der Cryptografic Information Application (CIA) für die Anwendung QES

In Kapitel 5.7.1.5 MF/DF.CIA.QES/EF.CD (Certificate Directory) wird der Inhalt von Tab_HBA_ObjSys_063 wie folgt geändert

Tabelle 92: Tab_HBA_ObjSys_063 Initialisierte Attribute von MF/ DF.CIA.QES / EF.CD (Certificate Directory)

Attribute	Wert	Bemerkung
Objekttyp	transparentes Elementary File	
fileIdentifier	’50 38’	
shortFileIdentifier	‘16’ = 22	
numberOfOctet	~~’01 50’ Oktett = 336 Oktett~~ ’4E’ Oktett = 78 Oktett	
positionLogicalEndOfFile	~~’01 50’ Oktett = 336 Oktett~~ ’4E’ Oktett = 78 Oktett	
flagTransactionMode	False	
flagChecksum	True	
lifeCycleStatus	Operational state (activated)	
shareable	True	

body	`30 25`
	`  30 0A`
	`  0C 08 43 2E 48 50 2E 51 45 53`
	`  30 03`	

```
      04 01 84
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 00
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
         2D 41 43 31
   30 03
      04 01 84
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 01
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
         2D 41 43 32
   30 03
      04 01 84
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 02
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
         2D 41 43 33
   30 03
      04 01 84
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 03
30 25
   30 0A
      0C 08 43 2E 48 50 2E 51 45 53
   30 03
      04 01 86
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 06
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
         2D 41 43 31
   30 03
      04 01 86
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 07
```

	~~A0 29~~ ~~30 0E~~ ~~0C 0C 43 2E 48 50 2E 51 45 53~~ ~~2D 41 43 32~~ ~~30 03~~ ~~04 01 86~~ ~~A1 12~~ ~~30 10~~ ~~30 0E~~ ~~A1 0C~~ ~~4F 06 D2 76 00 00 66 01~~ ~~04 02 C0 08~~ ~~A0 29~~ ~~30 0E~~ ~~0C 0C 43 2E 48 50 2E 51 45 53~~ ~~2D 41 43 33~~ ~~30 03~~ ~~04 01 86~~ ~~A1 12~~ ~~30 10~~ ~~30 0E~~ ~~A1 0C~~ ~~4F 06 D2 76 00 00 66 01~~ ~~04 02 C0 09~~	

Hinweis

Durch die Änderungen in gemSpec_HBA_ObjSys_G2.1 entfallen die Anforderungen:

- Card-G2-A_2094-01
- Card-G2-A_3302
- Card-G2-A_2095-01
- Card-G2-A_3303
- Card-G2-A_2096-01
- Card-G2-A_3304
- Card-G2-A_3633
- Card-G2-A_3634
- Card-G2-A_3635
- Card-G2-A_3636
- Card-G2-A_3637
- Card-G2-A_3638

Durch die Änderungen in gemSpec_HBA_ObjSys_G2.1 wird folgende Anforderung inhaltlich geändert:

- Card-G2-A_2123-01

Die dargestellten Änderungen in gemSpec_HBA_ObjSys_G2.1 ergeben Veränderungen in den folgenden Produkttypsteckbriefen.

Änderungen in gemProdT_HBA_G2.1

Tabelle 10: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
GS-A_5020	Einbringung des Komponentenzertifikats durch den Kartenherausgeber	gemRL_TSL_SP_CP
TIP1-A_2578	Korrekte ICCSN der Chipkarte	gemSpec_CVC_TSP
TIP1-A_2589	Personalisierung des CVC-CA-Zertifikats	gemSpec_CVC_TSP
Card-G2-A_3673	K_Personalisierung und K_Initialisierung:Unterstützung RSA CV-Zertifikate	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3009	K_HBA: Zusatzanforderungen für kontaktlose Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2038	K_Personalisierung: Druck der CAN auf den HBA bei Verwendung der optionalen kontaktlosen Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2673	K_Personalisierung und K_Initialisierung: Wert von „positionLogicalEndOfFile“	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3005	Absicherung der Kartenadministration	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2040-01	K_Personalisierung und K_Initialisierung: Wert des Attributes answerToReset	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2041	K_Personalisierung: Wert des Attributes iccsn8	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3395	K_Personalisierung: personalisierter Wert von pointInTime	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2043	K_Personalisierung und K_Initialisierung: ATR-Kodierung	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2044-01	K_Personalisierung und K_Initialisierung: TC1 Byte im ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3627	K_Personalisierung und K_Initialisierung: T0 Byte im ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3015	K_Personalisierung und K_Initialisierung: Historical Bytes im ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2045	K_Personalisierung und K_Initialisierung: Vorgaben für Historical Bytes	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2046	K_Personalisierung: Kompatibilität zu G1-Karten	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2058-01	K_Personalisierung: Personalisiertes Attribut von EF.GDO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3281	K_Personalisierung: Personalisierte Attribute von MF / EF.C.CA_HPC.CS.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3282	K_Personalisierung: Personalisierte Attribute von MF / EF.C.CA_HPC.CS.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3385	K_Personalisierung: Festlegung von CHR in MF / EF.C.HPC.AUTR_CVC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3283	K_Personalisierung: Personalisierte Attribute von MF / EF.C.HPC.AUTR_CVC.R2048	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_3386	K_Personalisierung: Festlegung von CHR in MF / EF.C.HPC.AUTR_CVC.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3284	K_Personalisierung: Personalisierte Attribute von MF / EF.C.HPC.AUTR_CVC.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3387	K_Personalisierung: Festlegung von CHR in MF / EF.C.HPC.AUTD_SUK_CVC.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3285	K_Personalisierung: Personalisierte Attribute von MF / EF.C.HPC.AUTD_SUK_CVC.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3326	K_Personalisierung: Personalisierte Attribute von MF / PuK.RCA.CS.R2048 für Testkarten	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3327-01	K_Personalisierung: Personalisierte Attribute von MF / PuK.RCA.CS.E256 für Testkarten	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3297	K_Personalisierung: Personalisierte Attribute von MF / SK.CAN	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_2869	K_Personalisierung: Generierung der CAN bei Verwendung der optionalen kontaktlosen Schnittstelle des HBA	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3301	K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES.R2048	gemSpec_HBA_ObjSys_ G2.1
~~Card-G2-A_3302~~	~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1~~	~~gemSpec_HBA_ObjSys_ G2.1~~
~~Card-G2-A_3303~~	~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2~~	~~gemSpec_HBA_ObjSys_ G2.1~~
~~Card-G2-A_3304~~	~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3~~	~~gemSpec_HBA_ObjSys_ G2.1~~
Card-G2-A_3630	K_Personalisierung: Personalisierte Attribute von MF / DF.QES / PrK.HP.QES.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3632	K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES.E256	gemSpec_HBA_ObjSys_ G2.1
~~Card-G2-A_3634~~	~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1.E256~~	~~gemSpec_HBA_ObjSys_ G2.1~~
~~Card-G2-A_3636~~	~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2.E256~~	~~gemSpec_HBA_ObjSys_ G2.1~~
~~Card-G2-A_3638~~	~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3.E256~~	~~gemSpec_HBA_ObjSys_ G2.1~~
Card-G2-A_3307	K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.R2048	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3308	K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.R2048	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3640	K_Personalisierung: Personalisierte MF / DF.ESIGN / PrK.HP.AUT.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3642	K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3644	K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3646	K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.E256	gemSpec_HBA_ObjSys_ G2.1
Card-G2-A_3317-01	K_Personalisierung: Personalisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO1.R3072	gemSpec_HBA_ObjSys_ G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_3318-01	K_Personalisierung: Personalisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO2.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3479	Kodierung von Versionskennungen	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3480	Kodierung von Produktidentifikatoren	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3481	Ausschluss für die Kodierung von Produktidentifikatoren	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3487	K_Initialisierung und K_Personalisierung: DO_HistoricalBytes in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3492	K_Personalisierung: DO_PT_Pers in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3494	K_Personalisierung: DO_PI_Kartenkörper in EF.ATR-Personalisierung	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3495	K_Personalisierung: DO_PI_Personalisierung in EF.ATR-Personalisierung	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3496	K_Initialisierung: Weitere Datenobjekte in DO_HistoricalBytes in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3497	K_Personalisierung: Vollständige Befüllung von EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3498	K_Personalisierung: DO_ICCSN in EF.GDO	gemSpec_Karten_Fach_TIP_G2.1
GS-A_4377	Card-to-Card-Authentisierung G1	gemSpec_Krypt
GS-A_3695	Grundlegender Aufbau Versionsnummern	gemSpec_OM
GS-A_5026	Versionierung von Karten durch die Produktidentifikation	gemSpec_OM
GS-A_5140	Inhalt der Selbstauskunft von Karten	gemSpec_OM
GS-A_4559	Versionierung der Karten der TI	gemSpec_OM
GS-A_4560	Versionierung von Datenstrukturen der Karten der TI	gemSpec_OM
GS-A_4583	Berufsgruppenkennzeichen für HBA	gemSpec_PKI
GS-A_4587	Gesamtlänge der Telematik-ID	gemSpec_PKI
GS-A_4710	Präfix der Telematik-ID	gemSpec_PKI
GS-A_4711	Separator der Telematik-ID	gemSpec_PKI
GS-A_4974	CV-Ausstattung von Smartcards der TI	gemSpec_PKI
GS-A_4621	Zugriffsprofil von HBA und SM-B (SMC-B, HSM-B)	gemSpec_PKI
GS-A_4623	Zugriffsprofil eines HBA	gemSpec_PKI

Änderungen in gemProdT_HBA_ObjSys_G2.1

Tabelle 11: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_2232	Größe Chip-Modul	gemSpec_eGK_OPT

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_2036	K_Initialisierung Anzahl logischer Kanäle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3006	K_HBA: USB-Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2867	K_HBA: Vorhandensein einer USB-Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3007	K_HBA: Kontaktlose Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3009	K_HBA: Zusatzanforderungen für kontaktlose Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3010	K_Initialisierung und K_Personalisierung: Kontaktlose Schnittstelle wird nicht genutzt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3011	K_Initialisierung: Kontaktlose Schnittstelle im COS nicht vorhanden	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3277	K_Personalisierung und K_Initialisierung: Konformität kontaktlose Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3014	K_HBA: Vorhandensein Kryptobox	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3182	K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, kein konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3184	K_Initialisierung: Verhalten der Objekte, konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3185	K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2033-01	K_Initialisierung: Ordnerattribute	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3626	K_Initialisierung: Herstellerspezifischer FileIdentifier	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2673	K_Personalisierung und K_Initialisierung: Wert von „positionLogicalEndOfFile“	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2035	K_Initialisierung: Zugriffsregeln für besondere Kommandos	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2039	K_Initialisierung: Wert des Attributes root	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2040-01	K_Personalisierung und K_Initialisierung: Wert des Attributes answerToReset	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2042	K_Initialisierung: Inhalt persistentPublicKeyList	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3266-01	K_Initialisierung: Wert von pointInTime	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2043	K_Personalisierung und K_Initialisierung: ATR-Kodierung	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2044-01	K_Personalisierung und K_Initialisierung: TC1 Byte im ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3627	K_Personalisierung und K_Initialisierung: T0 Byte im ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3015	K_Personalisierung und K_Initialisierung: Historical Bytes im ATR	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_2045	K_Personalisierung und K_Initialisierung: Vorgaben für Historical Bytes	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2047	K_Initialisierung: Initialisierte Attribute von MF	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2048-01	K_Initialisierung: Initialisierte Attribute von MF / EF.ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3278	K_Initialisierung: Initialisiertes Attribut numberOfOctet von MF / EF.ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3199	K_Initialisierung: Initialisierte Attribute von MF / EF.CardAccess	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3628	K_Initialisierung: Inhalt der Records von EF.DIR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2055-01	K_Initialisierung: Initialisierte Attribute von MF / EF.DIR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2057	K_Initialisierung: Initialisierte Attribute von MF / EF.GDO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2059-01	K_Initialisierung: Attribute von MF / EF.Version2	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2060	K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2061	K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2063	K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2064	K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2067	K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTD_SUK_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2069	K_Initialisierung: Initialisierte Attribute von MF / PIN.CH	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2071	K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2072	K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2075	K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTD_SUK_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2077-01	K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2078-01	K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3016-01	K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.ADMINCMS.CS.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2080-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES128	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2081-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3293-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES128	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_3295-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2868	K_Initialisierung: Initialisierte Attribute von MF / SK.CAN	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2082	K_Initialisierung: Initialisierte Attribute von MF / DF.HPA	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2083	K_Initialisierung: Initialisierte Attribute von MF / DF.HPA / EF.HPD	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2084	K_Initialisierung: Initialisierte Attribute von MF / DF.QES	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2085-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2088-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PIN.QES	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2089	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.SSEC	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2090-01	K_Initialisierung: Inhalt von EF.SSEC	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2091-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.R2048	gemSpec_HBA_ObjSys_G2.1
~~Card-G2-A_2094-01~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_2095-01~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_2096-01~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3~~	~~gemSpec_HBA_ObjSys_G2.1~~
Card-G2-A_3629-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3631	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.E256	gemSpec_HBA_ObjSys_G2.1
~~Card-G2-A_3633~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1.E256~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_3635~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2.E256~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_3637~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3.E256~~	~~gemSpec_HBA_ObjSys_G2.1~~
Card-G2-A_2097	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2098-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2101	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2107	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2110-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3639-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.E256	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_3641-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3643	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3645	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2117	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2119-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CIA.CIAInfo	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2120-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.OD (Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2121-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.AOD (Authentication Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2122-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.PrKD (Private Key Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2123-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CD(Certificate Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2118	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3320-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CIA.CIAInfo	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3321-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.OD (Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3322-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.AOD (Authentication Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3323-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.PrKD (Private Key Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3324-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CD (Certificate Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2124	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2125-01	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PrK.HP.AUTO.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2128	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PIN.AUTO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2129	K_Initialisierung: Initialisierte Attribute von MF / DF:ATUO / PIN.SO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2130-01	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO1.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2131-01	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO2.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3479	Kodierung von Versionskennungen	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3480	Kodierung von Produktidentifikatoren	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3481	Ausschluss für die Kodierung von Produktidentifikatoren	gemSpec_Karten_Fach_TIP_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_3483-01	K_Initialisierung: Inhalt body von EF.Version2	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3484	K_Initialisierung: Reihenfolge der Datenobjekte in body von EF.Version2	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3485	K_Initialisierung: Datenobjekte in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3486	K_Initialisierung: DO_BufferSize in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3487	K_Initialisierung und K_Personalisierung: DO_HistoricalBytes in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3488	K_Initialisierung: DO_PT_COS in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3489	K_Initialisierung: DO_PI_CHIP in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3490	K_Initialisierung: DO_PI_COS in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3491	K_Initialisierung: DO_PI_InitialisiertesObjSys in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3493	K_Initialisierung DO_PI_Kartenkörper in EF.ATR-Initialisierung	gemSpec_Karten_Fach_TIP
GS-A_4377	Card-to-Card-Authentisierung G1	gemSpec_Krypt
GS-A_3695	Grundlegender Aufbau Versionsnummern	gemSpec_OM
GS-A_3700	Versionierung von Produkten auf Basis von dezentralen Produkttypen der TI-Plattform durch die Produktidentifikation	gemSpec_OM
GS-A_5026	Versionierung von Karten durch die Produktidentifikation	gemSpec_OM
GS-A_5054	Versionierung von Produkten durch die Produktidentifikation erweitert um Klartextnamen	gemSpec_OM
GS-A_5140	Inhalt der Selbstauskunft von Karten	gemSpec_OM
GS-A_4559	Versionierung der Karten der TI	gemSpec_OM
GS-A_4668	Prüfung der mathematischen Korrektheit bei CV-Zertifikaten der Generation G1	gemSpec_PKI

Tabelle 12: Anforderungen zur sicherheitstechnischen Eignung "Sich.techn. Eignung: Zertifizierung nach Technischer Richtlinie"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_2036	K_Initialisierung Anzahl logischer Kanäle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3009	K_HBA: Zusatzanforderungen für kontaktlose Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3277	K_Personalisierung und K_Initialisierung: Konformität kontaktlose Schnittstelle	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2032	K_Initialisierung: Änderung von Zugriffsregeln	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3182	K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, kein konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_3183	K_Initialisierung: Eigenschaften der Objekte in anderen SEs, kein konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3184	K_Initialisierung: Verhalten der Objekte, konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3185	K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3186	K_Initialisierung: Eigenschaften der Objekte in anderen SEs, konkretes SE genannt	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2033-01	K_Initialisierung: Ordnerattribute	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2034	K_Initialisierung: Dateiattribute	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2673	K_Personalisierung und K_Initialisierung: Wert von „positionLogicalEndOfFile“	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2035	K_Initialisierung: Zugriffsregeln für besondere Kommandos	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3325	K_Initialisierung und K_Personalisierung: Abweichung von Festlegungen zum Zwecke der Personalisierung	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2039	K_Initialisierung: Wert des Attributes root	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2040-01	K_Personalisierung und K_Initialisierung: Wert des Attributes answerToReset	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2042	K_Initialisierung: Inhalt persistentPublicKeyList	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3181	K_Initialisierung: Größe persistentPublicKeyList	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3266-01	K_Initialisierung: Wert von pointInTime	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2043	K_Personalisierung und K_Initialisierung: ATR-Kodierung	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2044-01	K_Personalisierung und K_Initialisierung: TC1 Byte im ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3015	K_Personalisierung und K_Initialisierung: Historical Bytes im ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2045	K_Personalisierung und K_Initialisierung: Vorgaben für Historical Bytes	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2047	K_Initialisierung: Initialisierte Attribute von MF	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2048-01	K_Initialisierung: Initialisierte Attribute von MF / EF.ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3278	K_Initialisierung: Initialisiertes Attribut numberOfOctet von MF / EF.ATR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3199	K_Initialisierung: Initialisierte Attribute von MF / EF.CardAccess	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2055-01	K_Initialisierung: Initialisierte Attribute von MF / EF.DIR	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2057	K_Initialisierung: Initialisierte Attribute von MF / EF.GDO	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_2059-01	K_Initialisierung: Attribute von MF / EF.Version2	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2060	K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2061	K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2063	K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2064	K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2067	K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTD_SUK_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2069	K_Initialisierung: Initialisierte Attribute von MF / PIN.CH	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2071	K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2072	K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2075	K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTD_SUK_CVC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2077-01	K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2078-01	K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3016-01	K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.ADMINCMS.CS.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2080-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES128	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2081-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3293-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES128	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3295-01	K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2868	K_Initialisierung: Initialisierte Attribute von MF / SK.CAN	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2082	K_Initialisierung: Initialisierte Attribute von MF / DF.HPA	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2083	K_Initialisierung: Initialisierte Attribute von MF / DF.HPA / EF.HPD	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2084	K_Initialisierung: Initialisierte Attribute von MF / DF.QES	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2085-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2088-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PIN.QES	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2089	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.SSEC	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_2091-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.R2048	gemSpec_HBA_ObjSys_G2.1
~~Card-G2-A_2094-01~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_2095-01~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_2096-01~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3~~	~~gemSpec_HBA_ObjSys_G2.1~~
Card-G2-A_3629-01	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3631	K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.E256	gemSpec_HBA_ObjSys_G2.1
~~Card-G2-A_3633~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1.E256~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_3635~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2.E256~~	~~gemSpec_HBA_ObjSys_G2.1~~
~~Card-G2-A_3637~~	~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3.E256~~	~~gemSpec_HBA_ObjSys_G2.1~~
Card-G2-A_2097	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2098-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2101	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2107	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2110-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.R2048	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3639-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3641-01	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3643	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3645	K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.E256	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2117	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2119-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CIA.CIAInfo	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2120-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.OD (Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2121-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.AOD (Authentication Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2122-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.PrKD (Private Key Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2123-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CD(Certificate Directory)	gemSpec_HBA_ObjSys_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_2118	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3320-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CIA.CIAInfo	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3321-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.OD (Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3322-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.AOD (Authentication Object Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3323-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.PrKD (Private Key Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3324-01	K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CD (Certificate Directory)	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2124	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2125-01	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PrK.HP.AUTO.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2128	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PIN.AUTO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3270	K_Initialisierung: CHANGE REFERENCE DATA bei Nutzung der Leer-PIN für PIN.AUTO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2129	K_Initialisierung: Initialisierte Attribute von MF / DF:ATUO / PIN.SO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3271	K_Initialisierung: CHANGE REFERENCE DATA bei Nutzung der Leer-PIN für PIN.SO	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2130-01	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO1.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2131-01	K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO2.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_2675-01	K_Initialisierung: Initialisierte: Wert von PrK.HP.AUTO.R3072	gemSpec_HBA_ObjSys_G2.1
Card-G2-A_3479	Kodierung von Versionskennungen	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3480	Kodierung von Produktidentifikatoren	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3481	Ausschluss für die Kodierung von Produktidentifikatoren	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3483-01	K_Initialisierung: Inhalt body von EF.Version2	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3484	K_Initialisierung: Reihenfolge der Datenobjekte in body von EF.Version2	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3485	K_Initialisierung: Datenobjekte in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3486	K_Initialisierung: DO_BufferSize in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3487	K_Initialisierung und K_Personalisierung: DO_HistoricalBytes in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3488	K_Initialisierung: DO_PT_COS in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
Card-G2-A_3489	K_Initialisierung: DO_PI_CHIP in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3490	K_Initialisierung: DO_PI_COS in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3491	K_Initialisierung: DO_PI_InitialisiertesObjSys in EF.ATR	gemSpec_Karten_Fach_TIP_G2.1
Card-G2-A_3493	K_Initialisierung DO_PI_Kartenkörper in EF.ATR-Initialisierung	gemSpec_Karten_Fach_TIP_G2.1
GS-A_3695	Grundlegender Aufbau Versionsnummern	gemSpec_OM
GS-A_3696	Zeitpunkt der Erzeugung neuer Versionsnummern	gemSpec_OM
GS-A_3700	Versionierung von Produkten auf Basis von dezentralen Produkttypen der TI-Plattform durch die Produktidentifikation	gemSpec_OM
GS-A_5026	Versionierung von Karten durch die Produktidentifikation	gemSpec_OM
GS-A_5140	Inhalt der Selbstauskunft von Karten	gemSpec_OM
GS-A_4559	Versionierung der Karten der TI	gemSpec_OM

Änderung: Eindeutigkeit einer Jobnummer

Änderungsbedarf

TIP1-A_4644 fordert, dass eine Jobnummer über 1000 Vorgänge eindeutig sein muss. Es ist hierbei nicht genau ersichtlich zu welchem Zeitpunkt diese Eindeutigkeit sichergestellt werden muss. Der Konnektor generiert Jobnummern. Clientsysteme können aber auch Jobnummern bei der Signaturerstellung an den Konnektor übergeben, die nicht vom Konnektor erzeugt wurden. Dies wird präzisiert. Es wird ein gematik-Fehler definiert, für den Fall, dass eine Jobnummer nicht eindeutig ist.

Für den Fall, dass Clientsysteme nicht den Signaturproxy nutzen, müssen Clientsystem die Forderung, dass eine Jobnummer über 1000 Vorgänge eindeutig sein muss, selbst umsetzen. Hierfür werden Anforderungen an Clientsysteme spezifiziert.

Änderung in gemSpec_Kon

4.1.8.1.4 Jobnummer

Da die eHealth-Kartenterminals dezentral über eine Netzwerkschnittstelle am Konnektor betrieben werden, fehlt die Möglichkeit zur direkten physischen und vom Anwender kontrollierbaren Zuordnung eines solchen Terminals zu einem Arbeitsplatz, auf dem sich das Clientsystem befindet.

Daher ist es bei einer fehlerhaften Zuordnung eines eHealth-Kartenterminals zu einem Arbeitsplatz möglich, dass die PIN-Eingabeaufforderung – beispielsweise zu einem Signaturauftrag – an ein entferntes Kartenterminal weitergeleitet wird. Diese fehlerhafte Zuordnung kann durch einen Fehler des Clientsystems oder den Versuch eines Angriffes hervorgerufen werden.

Die Jobnummern werden vom Konnektor erzeugt und können durch Clientsystem oder Signaturproxy abgerufen werden. Der Konnektor stellt jedoch keine Verbindung zwischen erzeugten und verwendeten Jobnummern her. Es wird also nicht geprüft, ob nur Jobnummern verwendet werden, die vorher vom Konnektor erzeugt wurden, oder ob alle Jobnummern verwendet werden, die vom Konnektor erzeugt wurden.

⌦ **TIP1-A_4639 Generierung von Jobnummern für PIN-Eingaben**

Um Fehler- und Angriffsmöglichkeiten auszuschließen, MUSS der Konnektor bei bestimmten PIN-Verifikationen vor der Aufforderung zur PIN-Eingabe an einem eHealth-Kartenterminal eine hinreichend eindeutige Nummer – die Jobnummer – generieren, welche den Auftrag kennzeichnet, für dessen Verarbeitung die PIN-Eingabe erfolgen soll. Bei welchen PIN-Verifikationen dies der Fall ist, kann den PIN-Prompts in Tabelle 56: TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal entnommen werden.⌫

TIP1-A_4640 Anzeige der Jobnummern für PIN-Eingaben

Diese Jobnummer MUSS vom Konnektor im Display des eHealth-Kartenterminals neben der PIN-Eingabeaufforderung angezeigt werden.⌫

TIP1-A_4992 Guidance zur Jobnummer

Das Handbuch des Konnektors MUSS den Benutzer über den korrekten Gebrauch der Jobnummer informieren. Es MUSS ihm verdeutlichen, dass er seine PIN über die Tastatur des eHealth-Kartenterminals nur eingeben darf, wenn am Signaturproxy bzw. Primärsystem und am Display des Kartenterminals die gleiche Jobnummer angezeigt wird. Stimmen die beiden Nummern nicht überein, so soll der Benutzer seine PIN nicht eingeben und stattdessen weitergehende Schritte zur Klärung des aufgetretenen Fehlverhaltens einleiten. ⌫

TIP1-A_4642 Ableitung der Jobnummer von einem Zufallswert

Zur hinreichend eindeutigen Kennzeichnung des Vorganges MUSS eine Jobnummer von einem Zufallswert abgeleitet sein, wobei die Vorgaben an einen solchen Zufallswert beachtet werden MÜSSEN [gemSpec_Krypt#2.2].⌫

TIP1-A_4643 Beschaffenheit der Jobnummer

Zur Wahrung der Benutzerfreundlichkeit MUSS eine Reduzierung der Jobnummer auf eine Länge von sechs Zeichen erfolgen. Diese sechs Zeichen MÜSSEN in zwei Zeichengruppen mit je drei Zeichen, getrennt durch einen Bindestrich (0x2D), dargestellt werden. Die erste Zeichengruppe MUSS ausschließlich die Zeichen „A-Z“ ~~Buchstaben~~ beinhalten, die zweite Zeichengruppe MUSS aus Ziffern „0-9“ bestehen. Die Länge der resultierenden, reduzierten Jobnummer ist sieben und wird durch den Umfang der darstellbaren Zeichen auf dem Display des eHealth-Kartenterminals beschränkt.⌫

TIP1-A_4644 Jobnummer über 1.000 Vorgänge eindeutig

~~Der Konnektor MUSS sicherstellen, dass eine einmal angezeigte Jobnummer nicht innerhalb der nächsten 1.000 Vorgänge erneut Anwendung findet.~~

Der Konnektor MUSS die Eindeutigkeit einer Jobnummer sicherstellen:

- Bei Aufruf der Operation GetJobnumber MUSS der Konnektor innerhalb von 1000 Aufrufen eine eindeutige Jobnummer generieren. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext.
- Wird die Operation SignDocument mit einer Jobnummer aufgerufen, die innerhalb der vorangegangenen 1.000 Vorgänge verwendet wurde, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4252 abbrechen. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext. ⌫

TIP1-A_4645 Zeichen der Jobnummer

Die einzelnen Zeichen der Jobnummer MÜSSEN für die Anzeige am Kartenterminal gemäß dem Zeichensatz ISO 646DE/DIN66003, bzw. ISO 646 US codiert werden. Aus diesem Zeichensatz dürfen nur die Zeichen „A-Z“ (0x41 bis 0x5A) und die Ziffern „0-9“ (0x30 bis 0x39) für die Anzeige der Jobnummer verwendet werden. ⌫

Beispiele für eine Jobnummer sind ABC-475 und HZF-696.

Die Einbettung der Jobnummer in den Nachrichtentext für den Bildschirm des Kartenlesers wird in Tabelle 56: TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminalbeschrieben.

4.1.8.5.1 SignDocument (nonQES und QES)

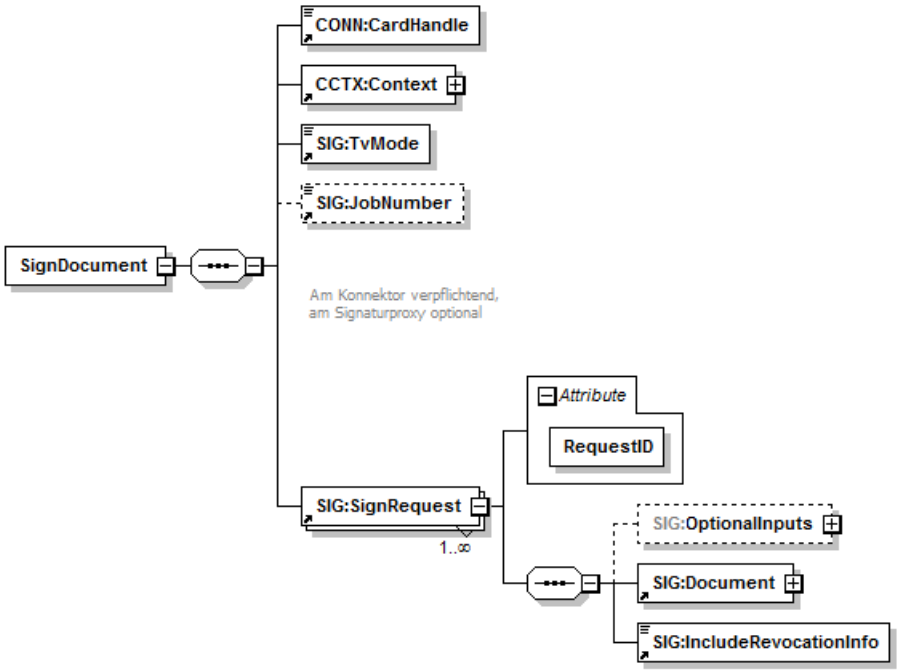
⌦ TIP1-A_5010 Operation SignDocument (nonQES und QES)

Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation SignDocument anbieten**.**

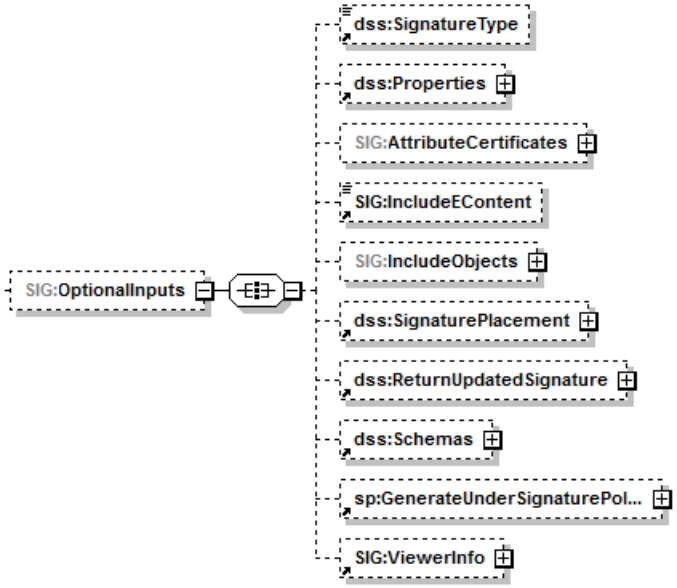
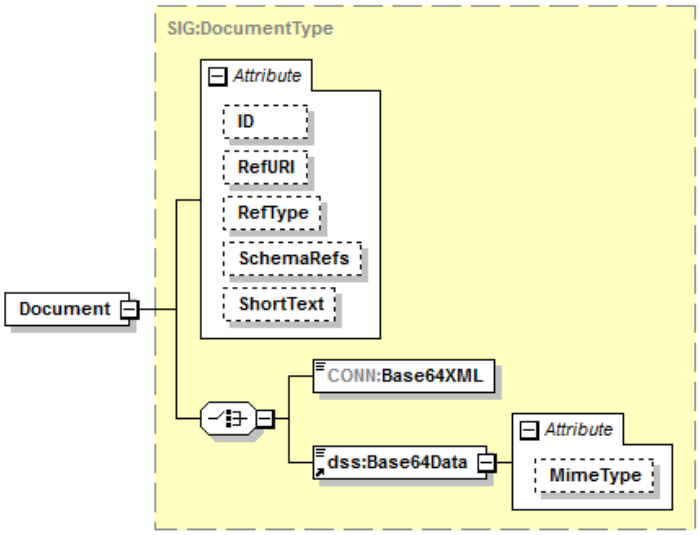
Tabelle 220: TAB_KON_065 Operation SignDocument (nonQES und QES)

Name	`SignDocument`
Beschreibung	Diese Operation lehnt sich an [OASIS-DSS] an. Sie enthält voneinander unabhängige SignRequests. Jeder SignRequest erzeugt eine Signatur für ein Dokument. Für die qualifizierte elektronische Signatur (QES) werden die `QES_DocFormate` unterstützt. Für nicht-qualifizierte elektronische Signaturen (nonQES) werden die `nonQES_DocFormate` unterstützt. Zur Signaturerzeugung werden Schlüssel und Zertifikate einer Chipkarte benutzt. Unterstützte Karten sind für die QES der HBAx mit dem QES-Zertifikat. Für die nonQES wird für die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur“ die SM-B mit dem OSIG–Zertifikat unterstützt. Bei der Erstellung von XML-Signaturen MUSS Canonical XML 1.1 verwendet werden [CanonXML1.1]. Es soll der Common-PKI-Standard eingesetzt werden, siehe [Common-PKI]. In Summe für die Größe der Dokumente in allen SignRequests innerhalb einer SignDocument-Anfrage MUSS der Konnektor eine Gesamtgröße von <= 250 MB unterstützen.

Aufruf-parameter



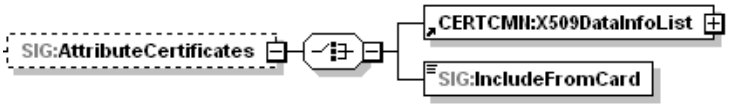
Name	Beschreibung
`CONN:CardHandle`	Identifiziert die zu verwendende Signaturkarte. Die Operation DARF die Signatur mit der eGK NICHT unterstützen. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4126 abbrechen.
`CCTX:Context`	Aufrufkontext QES mit HBAx: MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend Aufrufkontext nonQES mit SM-B: MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
`TvMode`	Der Parameter wird im Konnektor nicht ausgewertet.
`SIG:JobNumber`	Die Nummer des Jobs, unter der der nächste Signaturvorgang gestartet wird. Parameter ist verpflichtend.~~muss gefüllt werden.~~
`SIG:SignRequest`	Ein `SignRequest` kapselt den Signaturauftrag für ein Dokument. Das verpflichtende XML-Attribut `RequestID` identifiziert einen `SignRequest` innerhalb eines Stapels von `SignRequests` eindeutig. Es dient der Zuordnung der `SignResponse` zum jeweiligen `SignRequest`.
`SIG:OptionalInputs`	Enthält optionale Eingangsparameter (angelehnt an `dss:OptionalInputs` gemäß [OASIS-DSS] Section 2.7):

		
	`SIG:Document`	 Dieses an das `dss:Document` Element aus [OASIS-DSS] Section 2.4.2 angelehnte Element enthält das zu signierende Dokument, wobei die Kindelemente `CONN:Base64XML` und `dss:Base64Data` auftreten können. Bei den als `dss:Base64Data` übergebenen Dokumenten werden folgende (Klassen von) MIME-Types unterschieden: ”application/pdf-a” – für PDF/A-Dokumente, ”text/plain”, ”text/plain; charset=iso-8859-15” oder ”text/plain; charset=utf-8” – für Text-Dokumente, ”image/tiff” – für TIFF-Dokumente und ein beliebiger anderer MIME-Type für nicht näher unterschiedene Binärdaten des spezifizierten Typs. Der MIME-Type „text/plain“ wird interpretiert als „text/plain; charset=iso-8859-15”. Das Element enthält ein Attribut `ShortText`. Es muss für

		QES-Signaturen bei jedem Aufruf vom Clientsystem übergeben werden, für nonQES-Signaturen ist es optional. Über das Attribut `RefURI` kann gemäß [OASIS-DSS] (Abschnitt 2.4.1) ein zu signierender Teilbaum eines XML-Dokuments ausgewählt werden.
	`SIG:IncludeRevocationInfo`	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturerstellung vorliegenden Sperrinformationen anfordern. Es wird ausschließlich die zu erstellende Signatur betrachtet, d.h. es erfolgt keine Einbettung von Sperrinformationen für bereits enthaltene Signaturen. Für nicht-qualifizierte elektronische Signaturen (nonQES) wird diese Funktionalität nicht unterstützt.
	`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element kann der generelle Typ der zu erzeugenden Signaturen spezifiziert werden. Hierbei MÜSSEN folgende Signaturtypen unterstützt werden: **XML-Signatur** Durch Übergabe der URI urn:ietf:rfc:3275 wird die Erstellung von XML-Signaturen gemäß [RFC3275], [XMLDSig] angestoßen. Das zu verwendende Profil ist XAdES-BES ([XAdES]). Die Rückgabe einer solchen Signatur erfolgt als `ds:Signature`-Element. **CMS-Signatur** Durch Übergabe der URI urn:ietf:rfc:5652 wird eine CMS-Signatur gemäß [RFC5652] angestoßen. Das zu verwendende Profil ist CAdES-BES ([CAdES]). Die Signatur wird als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert. **S/MIME-Signatur** Durch Übergabe der URI „urn:ietf:rfc:5751" wird eine S/MIME-Signatur gemäß [RFC5751] angestoßen. Die CMS-Signatur der übergebenen MIME-Nachricht erfolgt konform der Vorgaben zur CMS-Signatur. Das Rückgabedokument ist eine MIME-Nachricht vom Typ „application/pkcs7-mime" mit einer CMS-Struktur vom Typ SignedData. Ist das übergebene Dokument keine MIME-Nachricht, so wie der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. **PDF-Signatur** Durch Übergabe der URI http://uri.etsi.org/02778/3 wird die Erzeugung einer PAdES-Basic Signatur gemäß [PAdES-3]angestoßen, wobei das Dokument mit der integrierten Signatur als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert wird. Handelt es sich beim übergebenen Dokument nicht um ein `Base64Data`-Element mit MIME-Type „application/pdf-a", so wird ein Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder

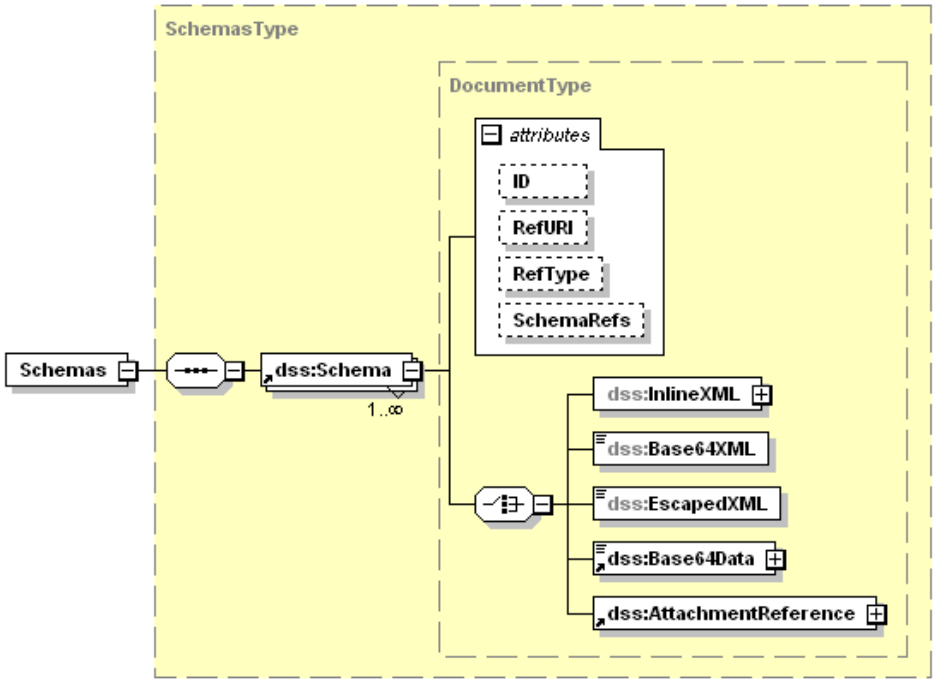
		Signaturvariante). Die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur“ DÜRFEN für QES der HBAx nur mit dem QES-Zertifikat erfolgen, für nonQES nur mit dem OSIG-Zertifikat der SM-B. In jedem diese Anforderung verletzenden Fall MUSS der Fehler 4058 (Aufruf nicht zulässig) zurückgeliefert werden. Fehlt dieses Element, so wird der Signaturtyp gemäß TAB_KON_583 – Default-Signaturverfahren aus dem Dokumententyp abgeleitet.

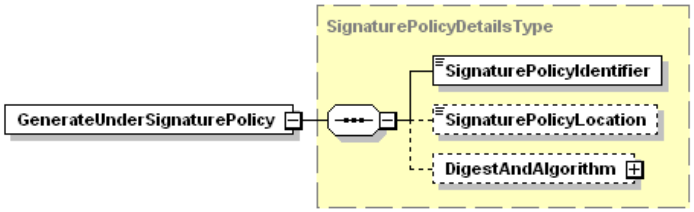
	`dss:Properties`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.5) definierte Element können zusätzliche signierte und unsignierte Eigenschaften (Properties) bzw. Attribute in die Signatur eingefügt werden.
		Unterstützt werden genau folgende Attribute:
		Im CMS-Fall (SignatureType = urn:ietf:rfc:5652) kann es XML-Elemente
		`./SignedProperties/Property/Value/CMSAttribute`
		und
		`./UnsignedProperties/Property/Value/CMSAttribute`
		enthalten. Ein solches XML-Element `CMSAttribute` muss ein vollständiges, base64/DER-kodiertes ASN.1-`Attribute` enthalten, definiert in [CMS#5.3.SignerInfo Type]. Es muss bei der Erstellung des CMS-Containers unverändert unter `SignedAttributes bzw. UnsignedAttributes` aufgenommen werden.

	`SIG:AttributeCertificates`	
		Dieses Element ist nur im Fall einer QES-Signatur relevant.
		Ist dieses Element nicht vorhanden, werden keine Attributzertifikate eingebetttet.
		Ist dieses Element vorhanden, können über das Element CERTCMN:X509DataInfoList Attributzertifikate beim Aufruf in dem Format mitgegeben werden, wie es die Operation ReadCardCertificate des Zertifikatsdienstes liefert.
		Alternativ kann der Konnektor über das Element SIG:IncludeFromCard angewiesen werden, alle Attributzertifikate zum Basiszertifikat von der signierenden Karte zu lesen.
		Die Zertifikate bettet der Konnektor für XAdES unter
		`ds:Signature/ds:Object/QualifyingProperties/SignedProperties/SignedSignatureProperties/SignerRole/CertifiedRoles/CertifiedRole`
		in die Signatur ein und für CAdES und PAdES unter
		`SignedData/signerInfos/SignerInfo/SignedAttrs/SignerAttributes/CertifiedAttributes`
		Es muss mindestens eine Anzahl von 10 Attributzertifikaten unterstützt werden.

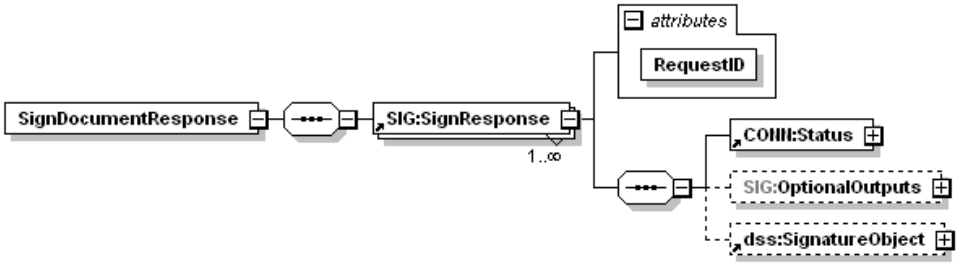
	`SIG:IncludeEContent`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.7), definierte Element kann bei einer CMS-basierten Signatur das Einfügen des signierten Dokumentes in die Signatur angefordert werden.

		Die Verwendung dieses Parameters bei anderen Signaturtypen führt zu einem Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante).
	`SIG:IncludeObject`	Dieses Element enthält zum Anfordern einer Enveloping XML Signatur ein dss:IncludeObject-Element gemäß [OASIS-DSS] (Abschnitt 3.5.6). Ist das Element vorhanden und ein anderer Signaturtyp als eine XML-Signatur angefordert, so wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.
	`dss:SignaturePlacement`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.8) definierte Element kann bei XML-basierten Signaturen gemäß [RFC3275] die Platzierung der Signatur im Dokument angegeben werden. Bei anderen Signaturtypen wird das Element ignoriert und eine Warnung (Fehlercode 4197, Parameter SignaturePlacement wurde ignoriert) zurückgeliefert.
	`dss:ReturnUpdatedSignature`	Durch dieses in [OASIS-DSS] (Abschnitt 4.5.8) definierte Element kann eine übergegebene XML- oder CMS-Signatur mit zusätzlichen Informationen und Signaturen (Parallel- und Gegensignaturen) versehen werden. Hierbei sind folgende Ausprägungen für das `Type`-Attribut vorgesehen: • http://ws.gematik.de/conn/sig/sigupdate/parallel Hierdurch wird eine Parallelsignatur zu einer bereits existierenden Signatur erzeugt und entsprechend zurückgeliefert. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentincluding Hierdurch wird eine dokumentinkludierende Gegensignatur für das Dokument und alle vorhandenen parallelen Signaturen erzeugt. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt. Bei anderen `Type`-Attributen wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.
	`dss:Schemas`	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schemata übergeben werden, die zur Validierung der übergebenen XML-Dokumente verwendet werden können.

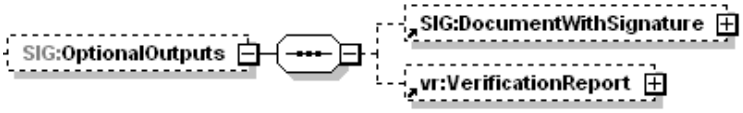
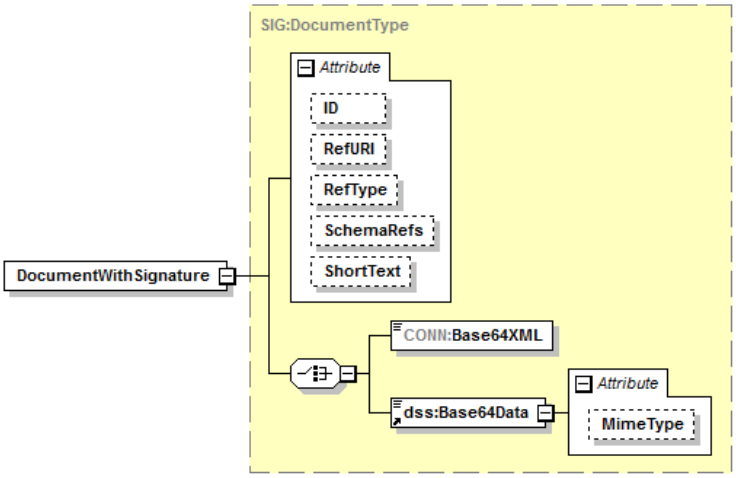


`dss:Schema`	Dieses Element enthält ein XML-Schema zur Validierung des übergebenen XML-Dokuments. Das Attribut `RefURI` ist verpflichtend. Es kennzeichnet dabei den Namensraum des XML-Schemas entsprechend [OASIS-DSS] (Abschnitt 2.8.5)
`sp:GenerateUnderSignaturePolicy`	 Über dieses in [OASIS-SP], Kapitel 2.2.1.1.1.1 Optional Input <GenerateUnderSignaturePolicy>, definierte Element wird die erforderliche Singnaturrichlinie ausgewählt. Die im Element `sp:SignaturePolicyIdentifier` übergebene URI identifiziert die Signaturrichtlinie.
`SIG:ViewerInfo`	Enthält optional die vom Konnektor in die Signatur einzubeziehende Referenzen für die Stylesheets zur Anzeige.

Rückgabe



`SIG:SignRespo`	Eine `SignResponse` kapselt den ausgeführten

	nse	Signaturauftrag pro Dokument. Die Zuordnung zwischen SignRequest und SignResponse erfolgt über die RequestID.
	CONN:Status	Enthält den Status der ausgeführten Operation pro SignRequest.
	SIG:OptionalOutputs	Enthält (angelehnt an dss:OptionalOutputs) optionale Ausgangsparameter: 
	SIG:DocumentWithSignature	 Pro SignResponse wird ein Element SIG:DocumentWithSignature gemäß [OASIS-DSS] (Abschnitt 3.5.8) zurückgeliefert, in dem das Dokument mit Signatur enthalten ist. Dabei werden die XML-Attribute des Elements SIG:Document auf dem zugehörigen SignRequest übernommen. Ist die Signatur nicht im Dokument enthalten, wird ein leeres Element Base64XML oder Base64Data zurückgegeben. Die Signatur wird dann im Element dss:SignatureObject abgelegt. Wenn die Signatur im Dokument enthalten ist, wird das signierte Dokument im Feld Base64XML bzw. Base64Data zurückgeliefert. In diesem Fall MUSS die dss:SignaturePtr-Alternative in dss:SignatureObject (vgl. [OASIS-DSS] Abschnitt 2.5) dazu genutzt werden, auf die in den Dokumenten enthaltenen Signaturen zu verweisen.
	vr:VerificationReport	Vom Konnektor nicht befüllt.
	dss:SignatureObject	Enthält im Erfolgsfall die erzeugte Signatur pro SignRequest in Form eines dss:SignatureObject-Elementes gemäß [OASIS-DSS] (Abschnitt 3.2).
Vorbe-	Keine	

dingungen	
Nachbe-dingungen	Keine

Der Ablauf der Operation SignDocument ist in Tabelle 221: TAB_KON_756 Ablauf Operation SignDocument (nonQES und QES) beschrieben:

Tabelle 221: TAB_KON_756 Ablauf Operation SignDocument (nonQES und QES)

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Anhand des Kartentyps wird ermittelt, ob eine QES oder eine nonQES erzeugt werden soll. Alle übergebenen Parameterwerte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs- berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 {
		mandatId =$context.mandantId;
		clientsystemId = $context.clientsystemId;
		cardHandle = $context.cardHandle;
		userId = $context.userId }

Im Fall QES wird Schritt 4 ausgeführt. Im Fall nonQES wird Schritt 5 ausgeführt.		
4a)	Prüfe Signaturdienst-Modul	Prüfe, ob MGM_LU_SAK=Enabled. Ist dies nicht der Fall, so bricht die Operation mit Fehler 4125 ab.
4b)	TUC_KON_150 „Dokumente QES signieren“	Die QES wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.
5)	TUC_KON_160 „Dokumente nonQES signieren“	Die nonQES wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.

Tabelle 1: TAB_KON_757 Fehlercodes „SignDocument (nonQES und QES)“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4126	Security	Error	Kartentyp nicht zulässig für Signatur

Fehlercode	ErrorType	Severity	Fehlertext
4125	Technical	Error	LU_SAK nicht aktiviert
4197	Technical	Warning	Parameter SignaturePlacement wurde ignoriert
4252	Technical	Error	Jobnummer wurde in den letzten 1.000 Aufrufen bereits verwendet und ist nicht zulässig

Die folgende Tabelle führt die zulässigen Zertifikate und Schlüssel für die nonQES auf:

Tabelle 223: TAB_KON_758 Zertifikat und privater Schlüssel je Karte für Sign/VerifyDocument (nonQES)

Karte	Zertifikat (Verify)	Schlüssel (Sign)
SM-B	EF.C.HCI.OSIG.R2048 in DF.ESIGN	PrK.HCI.OSIG.R2048 in DF.ESIGN

Die folgende Tabelle führt die zulässigen Zertifikate und Schlüssel für die QES auf:

Tabelle 224: TAB_KON_759 Zertifikat und privater Schlüssel je Karte für Sign/VerifyDocument (QES)

Karte	Zertifikat (Verify)	Schlüssel (Sign)
HBA-VK	EF.C.HP.QES in DF.QES	PrK.HP.QES in DF.QES
HBA	DF.C.HP.QES.R2048 in DF.QES	PrK.HP.QES.R2048 in DF.QES

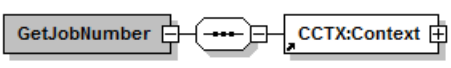
⌫

4.1.8.5.4 GetJobNumber

⌦ TIP1-A_5667 Operation GetJobNumber

Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation GetJobNumber anbieten**.**

Tabelle 231: TAB_KON_843 Operation GetJobNumber

Name	`GetJobNumber`	
Beschreibung	Diese Operation liefert eine Jobnummer zur Verwendung in der Operation SignDocument. Die Jobnummer MUSS nach den Vorgaben von Kapitel 4.1.8.1.4 erstellt werden.	
Aufrufparameter		
	Name	Beschreibung
	`CCTX:Context`	MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet

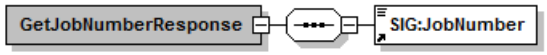
Rückgabe		
	`SIG: JobNumber`	Jobnummer zur Verwendung in „SignDocument“
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Tabelle 232: TAB_KON_844 Ablauf Operation GetJobNumber

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
2.	Generiere und liefere eine Jobnummer	Eine innerhalb von 1000 Aufrufen eindeutige Jobnummer wird generiert und geliefert. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext.

Tabelle 233: TAB_KON_845 Fehlercodes „GetJobNumber“

Fehlercode	ErrorType	Severity	Fehlertext
Folgende Fehlercodes können auftreten:			
4000	Technical	Error	Syntaxfehler

⌫

Änderung in gemILF_PS

4.4 Signaturerstellung und Verschlüsselung <Konnektor PTV2>

Die Nutzung der in diesem Kapitel geschilderten Funktionalität ist abhängig von der Verfügbarkeit eines QES-fähigen Konnektors.

Der Konnektor stellt generische Schnittstellen zur Erstellung und Prüfung von Signaturen und zur Verschlüsselung von Dokumenten zur Verfügung (`SignatureService`, `AuthSignatureService`, `EncryptionService`, `CertificateService`). Diese Schnittstellen können vom Primärsystem in einer Vielzahl von Szenarien genutzt werden.

Die Nutzung dieser Services ist optional. Die Vielfalt der bereitgestellten Funktionalität eröffnet dem Primärsystem eine Reihe von Möglichkeiten, die Signaturserviceschnittstelle im

Rahmen ihrer Workflows zu nutzen. Dem Primärsystem wird kein spezielles Nutzungsszenario zur Realisierung vorgeschlagen.

Diese Services bieten folgende Funktionalitäten an:

- Signieren von Dokumenten (XML, MIME, Text, Tiff, PDF/A, Binär)
- Prüfung von signierten Dokumenten (XML, MIME, Text, Tiff, PDF/A, Binär)
- Verschlüsseln von Dokumenten (XML, MIME, Text, Tiff, PDF/A, Binär), eGK kann verwendet werden
- Entschlüsseln von Dokumenten
- Zertifikatsabfragen von Smartcards
- Prüfung von Zertifikaten

Die Operationen dieser Dienste können einzeln genutzt werden. Sie ermöglichen, Dokumente mithilfe von Zertifikats- und Verschlüsselungsmaterial von Smartcards zu verschlüsseln und zu signieren. Wenn es sich bei der Smartcard um eine sichere Signaturerstellungseinheit für qualifizierte Signaturen handelt, so wird das Niveau einer qualifizierten elektronischen Signatur (QES) erreicht.

Das Primärsystem kann den Leistungsumfang des Signaturdienstes des Konnektors nur nutzen, wenn am Konnektor der entsprechende Parameter konfiguriert ist.

Zur Unterstützung bei der Signaturerstellung und Signaturprüfung kann der Signaturproxy des Konnektors eingesetzt werden. Der Signaturproxy ist eine Softwarekomponente auf dem Clientsystem und übernimmt Funktionen zur Prüfung und lokalen Anzeige. Wenn diese Funktionen nicht im Primärsystem umgesetzt sind, wird der Einsatz des Signaturproxys dringend empfohlen.

Der Signaturproxy bietet eine eIDAS-konforme Anzeige. Um diese lokale Anzeige für die Signaturerstellung und Signaturprüfung zu realisieren, ermittelt der Signaturproxy alle Informationen, die für die Anzeige notwendig sind und bereitet die Informationen sowie das Dokument zur Anzeige auf. Im Rahmen der Anzeige bietet der Signaturproxy dem Anwender Möglichkeiten, mit dem Signaturvorgang zu interagieren. Dazu gehört auch die Möglichkeit, die Verarbeitung einer Stapelsignatur abzubrechen.

Der Signaturproxy ist als lokale Anzeigesoftware zur Installation auf dem Clientrechner konzipiert. Der Signaturproxy stellt dem Clientsystem seine Schnittstellen auf dem lokalen Interface (localhost) zur Verfügung. Eine Transportsicherung (TLS) zwischen Clientsystem und Signaturproxys ist daher nicht nötig.

Alternativ kann die eIDAS-konforme Anzeige statt im Signaturproxy im Clientsystem selbst umgesetzt werden. In diesem Umsetzungsszenario kommuniziert das Clientsystem direkt mit dem Konnektor. Die Notwendigkeit für den Einsatz eines Signaturproxys entfällt. Es wird empfohlen, in diesem Umsetzungsszenario die Funktionalität der Anzeige und der Benutzerinteraktion im Clientsystem an der Spezifikation des Signaturproxy [gemSpec_Kon_SigProxy] auszurichten.

Damit die für Anzeige und Benutzerinteraktion verantwortliche Komponente die Verarbeitung einer Stapelsignatur abbrechen kann, stellt der Konnektor einen besonderen Mechanismus bereit: Der Konnektor gibt über die Operation `GetJobNumber` eine Jobnummer heraus, die

beim Aufruf der Operation `SignDocument` am Konnektor als Aufrufparameter mitgegeben werden muss und mit der eine laufende Verarbeitung durch Aufruf der Operation `StopSignature` am Konnektor abgebrochen werden kann. In der Schnittstelle zwischen Clientsystem und Signaturproxy entfällt die Notwendigkeit eine `Jobnummer` beim Aufruf der Operation `SignDocument` mitzugeben, weil der Signaturproxy die Benutzerinteraktion zur Stapelsignatur kapselt.

Kommuniziert das Clientsystem direkt mit dem Konnektor, wird empfohlen die Jobnummer durch den Konnektor mit der Operation `GetJobNumber` generieren zu lassen. Erzeugt das Clientsystem die Jobnummer selbst, so muss das Primärsystem die Eindeutigkeit der Jobnummer, wie vom Konnektor verlangt, sicherstellen.

A_13482 Eindeutigkeit der Jobnummer

Das Primärsystem, welches Jobnummern selbst erzeugt, MUSS die Eindeutigkeit der Jobnummer innerhalb der letzten 1000 Aufrufe über alle Arbeitsplätze sicherstellen. ⌫

Der Konnektor kann den Revocation-Status von Zertifikaten im Rahmen des Signatur- und Verschlüsselungsdienstes nur dann überprüfen, wenn der Konnektor die volle Online-Funktionalität nutzt.

Formate von Dokumenten sind dem Clientsystem bekannt und müssen an den unten beschriebenen Schnittstellenaufrufen auch dem Konnektor bekannt gegeben werden, damit dieser die dokumententypspezifischen Verarbeitungsschritte durchführen kann.

Die nicht-XML-Formate werden dabei nach MIME-Typ-Klassen unterschieden:

- „PDF/A" für MIME-Typ „application/pdf-a",
- „Text" für MIME-Typ „text/plain",
- „TIFF" für MIME-Typ „image/tiff"
- „Binär" für alle übrigen MIME-Typen.

Änderungsbedarf:

Die Nummerierung und dazugehörige Referenzierung beim Standardablauf des TUC_KON_150 sind fehlerhaft.

Die Nummerierung und Referenzierung wird korrigiert.

Dazu werden folgende Änderungen vorgesehen.

Änderungen in gemSpec_Kon

4.1.8.4.4 TUC_KON_150 „Dokumente QES signieren"

⌧ **TIP1-A_4655 TUC_KON_150 „Dokument QES signieren„**

Der Konnektor MUSS den technischen Use Case TUC_KON_150 „Dokumente QES signieren" umsetzen.

Tabelle 1: TAB_KON_755 – TUC_KON_150 „Dokumente QES signieren"

Element	Beschreibung
Name	TUC_KON_150 "Dokumente QES signieren"
Beschreibung	Im Rahmen von Fachanwendungen werden ein oder mehrere Dokumente mit einer qualifizierten elektronischen Signatur versehen. Es werden die QES_DocFormate unterstützt.
Auslöser	Aufruf durch ein Clientsystem (Operation SignDocument) oder ein Fachmodul.
Vorbedingungen	Die Signaturkarte muss gesteckt sein.
Eingangsdaten	• SignRequests. Jeder SignRequest kapselt: o Zu signierendes Dokument (Document) bzw. zu signierende Dokumente o weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur (siehe Operation SignDocument, Parameter dss:OptionalInputs) o includeRevocationInfo [Boolean] - optional; Default: true Dieser optionale Parameter steuert die Einbettung von OCSP-Responses in die Signatur (siehe Operation SignDocument, Parameter SIG:IncludeRevocationInfo) • CardSession (HBAx) • WorkplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBAx)
Ausgangsdaten	Signierte Dokumente
Standardablauf	Der Konnektor KANN die Schritte 1 bis 4 in einer beliebigen Reihenfolge durchführen. 1. Der Signaturtyp und die Signaturvariante werden für jedes

Element	Beschreibung
	Dokument der Liste entsprechend signatureType und SignatureVariant festgelegt (ggf. in optionalInputs enthalten). Wenn SignatureType oder SignatureVariant nicht übergeben wurden, wird das dem Dokumentformat entsprechende Default-Verfahren gewählt (siehe TAB_KON_583 – Default-Signaturverfahren). 2. Für alle Dokumente des Stapels wird die Zulässigkeit des Kartentyps geprüft. Das für die Signatur zu nutzende Zertifikat wird anhand des Kartentyps implizit ausgewählt. 3. Es werden die Voraussetzungen für die Signatur geprüft. Dies erfolgt im TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“. Wenn includeRevocationInfo=true, dann setze ocspResponses auf Rückgabewert von TUC_KON_152. 4. Die am Signaturvorgang beteiligten Ressourcen (Signaturkarte sowie PIN Pad und Display des PIN-Eingabe-Kartenterminals) werden für die exklusive Nutzung durch diesen Signaturvorgang reserviert. Die Reservierung der Signaturkarte erfolgt durch Aufruf von TUC_KON_023. 5. Zum Vorbereiten der Dokumente für die Signatur wird TUC_KON_155 „Dokumente zur Signatur vorbereiten“ mit ocspResponses aufgerufen. Die Zugriffe auf die Signaturkarte in den Schritten ~~86~~ bis ~~97~~ müssen im DF.QES erfolgen. 6. Die Signaturerstellung wird durch den Anwender autorisiert. Dies erfolgt durch Aufruf von TUC_KON_012. Wenn nur ein zu signierendes Dokument vorhanden ist und der Einfachsignaturmodus aktiviert ist (siehe Konfigurationsparameter SAK_SIMPLE_SIGNATURE_ MODE), wird Schritt ~~97~~a) durchgeführt, ansonsten Schritt ~~97~~b). ~~97~~a) Die Signatur wird erstellt. Dies erfolgt gemäß TUC_KON_168 „Einzelsignatur QES erstellen“. ~~97~~b) Die Signaturen werden erstellt. Dies erfolgt gemäß TUC_KON_154 „QES-Signaturen erstellen“. ~~108~~ Es wird DF.QES verlassen, um den PIN-Status der PIN.QES zurückzusetzen. Der im Konnektor verwaltetete Sicherheitszustand (CARDSESSION.AUTHSTATE) ist zu aktualisieren. ~~119~~ Die reservierten Ressourcen (Signaturkarte sowie PIN-Pad und Display des PIN-Eingabe-Kartenterminals) werden wieder freigegeben. Zur Freigabe der Signaturkarte wird TUC_KON_023 „Karte reservieren“ { cardSession; doLock = false } aufgerufen. ~~12~~10 Die signierten Dokumente werden an den Aufrufer zurückgegeben.
Varianten/Alternativen	Der Nutzer kann den Vorgang bei der Autorisierung (Schritt ~~96~~) abbrechen. Hierbei sind die gleichen Regeln anzuwenden wie im Fehlerfall (s. Fehlerfälle).
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes:

Element	Beschreibung
	(→1) Ungültige Angabe des Signaturtyps oder Signaturvariante: Fehlercode 4111 Übergabe eines für die QES nicht unterstützten Dokumentformats: Fehlercode 4110 (→2) Kartentyp nicht zulässig für Signatur: Fehlercode 4126 (→5) Fehler bei der Reservierung von Ressourcen: Fehlercode 4060 (→~~9~~7b) Karte ist kein HBA, sondern HBA-Vorläuferkarte: Fehlercode 4118 Im Fehlerfall, inklusive Timeout bei der PIN-Eingabe, oder bei Abbruch durch den Benutzer (Fehler 4049): a) … MUSS DF.QES verlassen werden b) … MÜSSEN alle reservierten Ressourcen freigegeben werden c)… MUSS der Fehler immer an das Clientsystem zurückgemeldet werden
Nichtfunktionale Anforderungen	keine
Zugehörige Diagramme	Abbildung 18: PIC_KON_114 Aktivitätsdiagramm zu „Dokument QES signieren“

Änderungsbedarf

Im Rahmen der Signaturprüfung für ein Dokument wird ein übergreifender Statuswert (VALID/INVALID/INCONCLUSIVE) aus den Statuswerten (VALID/INVALID/INCONCLUSIVE) für jede einzelne Signatur auf dem Dokument ermittelt. Der Statuswert zu jeder einzelnen Signatur korrespondiert zu den Werten resultmajor/resultminor innerhalb des VerifikationReports zu den einzelnen Signaturen. Hier besteht Präzisierungsbedarf. Zum einen bestehen Lücken, z.B. ist die Behandlung von Manifests nicht berücksichtigt. Zum anderen sind einzelne Punkte nicht klar definiert, so die Wirkung einer nicht durchführbaren OCSP-Abfrage.

Änderungen in gemSpec_Kon

4.1.8.4.2. TUC_KON_161 „nonQES Dokumentsignatur prüfen"

TIP1-A_4654 TUC_KON_161 „nonQES Dokumentsignatur prüfen"

Der Konnektor MUSS den technischen Use Case TUC_KON_161 „nonQES Dokumentsignatur prüfen" umsetzen.

…

Das Gesamtergebnis (VerificationResult) für die Prüfung einer Dokumentensignatur fasst die Ergebnisse aller Prüfungsschritte in einem einzelnen Statuswert zusammen. ~~der obigen Einzelschritte zusammen. Welche nicht erfüllten Kriterien einzelner Prüfschritte zu einem Fehler oder einer Warnung führen, wird durch die Regeln für die nonQES (Anhang B) bestimmt.~~

Tabelle 1: TAB_KON_754 Übersicht Status für Prüfung einer Dokumentensignatur

~~VerificationResult pro Signatur~~		
	~~**Wert**~~	~~**Bedeutung**~~
	~~VALID~~	~~Die Signatur wurde gemäß den Regeln für die nonQES geprüft und für gültig befunden.~~
	~~INVALID~~	~~Die Signatur des Dokuments ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.~~
	~~INCONCLUSIVE~~	~~Die Signatur wurde gemäß den Regeln für die nonQES geprüft. Allerdings konnten einige Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail.~~ ~~Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.~~

VerificationResult für gesamtes Dokument (VerificationResult/HighLevelResult)	
Wert	**Bedeutung**
VALID	Wenn VerificationResult für alle Signaturen zum Dokument VALID
INVALID	Wenn VerificationResult für eine Signatur zum Dokument INVALID

INCONCLUSIVE	in allen anderen Fällen
VerificationResult pro Signatur (VerificationReport/IndividualReport/Result)	
Wert	**Bedeutung** **mögliche Ausprägungen im VerificationReport**
VALID	Die Signatur wurde gemäß den Regeln für die nonQES geprüft und für gültig befunden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:HasManifestResults
INVALID	Die Signatur ~~des Dokuments~~ ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidSignatureTimestamp
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:ResponderError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:CertificateChainNotComplete
INCONCLUSIVE	Die Signatur wurde gemäß den Regeln für die nonQES geprüft. Allerdings konnten eine oder mehrere ~~einige~~ Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail. Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:OcspNotAvailiable Hinweis: Das Erreichen dieses Zustandes hängt davon ab, ob eine OCSP-Abfrage nicht durchgeführt werden konnte, unabhängig davon, ob die Ursache dafür die Offlineschaltung des Konnektors (MGM_LU_ONLINE = Disabled) oder die Nichterreichbarkeit des OCSP-Responders im Online-Betrieb (MGM_LU_ONLINE = Enabled) ist.

~~Das VerificationResult pro Signatur MUSS wie folgt zur Belegung des Elements VerficationReport/IndividualReport/Result korrespondieren:~~

- ~~VALID, wenn~~
  ~~ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success AND~~
  ~~ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments~~
- ~~INVALID, wenn~~
  ~~ResultMajor UNGLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success ODER~~
  ~~(ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success UND~~
  ~~ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature)~~

⌫

4.1.8.4.2. TUC_KON_151 „QES-Dokumentsignatur prüfen”

⌦ TIP1-A_4672 TUC_KON_151 „QES-Dokumentensignatur prüfen“

Der Konnektor MUSS den technischen Use Case TUC_KON_151 „QES-Dokumentensignatur prüfen” umsetzen.

...

Das Gesamtergebnis (VerificationResult) für die Prüfung einer Dokumentensignatur fasst die Ergebnisse aller Prüfungsschritte in einem einzelnen Statuswert zusammen. ~~der obigen Einzelschritte zusammen. Welche nicht erfüllten Kriterien einzelner Prüfschritte zu einem Fehler oder einer Warnung führen, wird durch die Regeln für die nonQES (Anhang B) bestimmt.~~

Tabelle 2: TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur

~~VerificationResult pro Signatur~~		
	~~Wert~~	**~~Bedeutung~~**
	~~VALID~~	~~Die Signatur wurde gemäß den Regeln für die nonQES geprüft und für gültig befunden.~~
	~~INVALID~~	~~Die Signatur des Dokuments ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.~~
	~~INCONCLUSIVE~~	~~Die Signatur wurde gemäß den Regeln für die nonQES geprüft. Allerdings konnten einige Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail.~~ ~~Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.~~

VerificationResult für gesamtes Dokument (VerificationResult/HighLevelResult)	
Wert	**Bedeutung**
VALID	Wenn VerificationResult für alle Signaturen zum Dokument VALID
INVALID	Wenn VerificationResult für eine Signatur zum Dokument INVALID
INCONCLUSIVE	in allen anderen Fällen

VerificationResult pro Signatur (VerificationReport/IndividualReport/Result)	
Wert	**Bedeutung** **mögliche Ausprägungen im VerificationReport**
VALID	Die Signatur wurde gemäß den Regeln für die ~~non~~QES geprüft und für gültig befunden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:HasManifestResults
INVALID	Die Signatur ~~des Dokuments~~ ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature

	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidSignatureTimestamp
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:ResponderError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:CertificateChainNotComplete
INCONCLUSIVE	Die Signatur wurde gemäß den Regeln für die QES geprüft. Allerdings konnten eine oder mehrere ~~einige~~ Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail. Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:OcspNotAvailiable Hinweis: Das Erreichen dieses Zustandes hängt davon ab, ob eine OCSP-Abfrage nicht durchgeführt werden konnte, unabhängig davon, ob die Ursache dafür die Offlineschaltung des Konnektors (MGM_LU_ONLINE = Disabled) oder die Nichterreichbarkeit des OCSP-Responders im Online-Betrieb (MGM_LU_ONLINE = Enabled) ist.

~~Das VerificationResult pro Signatur MUSS wie folgt zur Belegung des Elements VerfcationReport/IndividualReport/Result korrespondieren:~~

- ~~VALID, wenn~~
  ~~ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success AND~~
  ~~ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments~~
- ~~INVALID, wenn~~
  ~~ResultMajor UNGLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success ODER~~
  ~~(ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success UND~~
  ~~ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature)~~ ⌫

Ausbau der Funktionalität CORS

Für die Pflege der Basisdaten im Verzeichnisdienst (VZD) wird nicht mehr die Lösung verfolgt, dass der VZD eine Webanwendung zur Verfügung stellt.

Da es gegenwärtig keine Anwendungen gibt, die die Funktionalität CORS benötigen, gibt es auch nicht den Bedarf CORS im Konnektor zu unterstützen. Deshalb wird die Funktionalität CORS vollständig aus gemSpec_Kon ausgebaut.

Änderungsbedarf in gemSpec_Kon

~~3.5.3 Unterstützung von Webanwendungen~~

Wird vollständig gestrichen

~~3.5.5 Betriebsaspekte~~

Wird vollständig gestrichen

4.1.6.5 Operationen an der Außenschnittstelle

TIP1-A_4603 - Basisanwendung Systeminformationsdienst

Der Konnektor MUSS für Clients eine Basisanwendung Systeminformationsdienst anbieten.

Tabelle TAB_KON_029 Basisanwendung Systeminformationsdienst

Name	**EventService**	
Version	7.2.0 ~~7.2.1~~ Siehe Anhang D (WSDL-Version)	
Namensraum	Siehe Anhang D	
Namensraum-Kürzel	EVT für Schema und EVTW für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	GetCardTerminals	Auflistung der verfügbaren Kartenterminals
	GetCards	Auflistung der gesteckten Karten
	~~GetLeCards~~	~~Auflistung der gesteckten Karten der Kartentypen HBAx und SM-B~~
	GetResourceInformation	Liefert Details zu einer Ressource (Kartenterminal, Karte, HSM)
	Subscribe	Anmeldung der Zustellung von Ereignissen
	Unsubsribe	Abmelden von der Zustellung von Ereignissen
	RenewSubscriptions	Gültigkeit bestehender

		Subscriptions verlängern
	GetSubscriptions	Abfrage der angemeldeten Zustellungen von Ereignissen
WSDL	EventService.wsdl	
Schema	EventService.xsd	

⌫

~~4.1.6.5.8 GetLeCards~~

Wird vollständig gestrichen

Anhang D – Übersicht über die verwendeten Versionen

TAB_KON_688 Version der Schemas aus dem Namensraum des Konnektors

Schemas aus dem Namensraum des Konnektors „http://ws.gematik.de/conn“		
	…	
	XSD Name	EventService.xsd
	XSD Schemaversion	~~7.2.2~~ 7.2.1
	TargetNamespace	http://ws.gematik.de/conn/EventService/ v7.2
	…	

TAB_KON_798 Schnittstellenversionen

~~Systeminformationsdienst (EventService)~~		
	~~WSDL Name~~	~~EventService.wsdl~~
	~~WSDL-Version~~	~~7.2.1~~
	~~TargetNamespace~~	~~http://ws.gematik.de/conn/EventService/ WSDL/v7.2~~
	~~verwendete XSDs~~	~~CardService.xsd,~~ ~~CardServiceCommon.xsd,~~ ~~CardTerminalInfo.xsd,~~ ~~ConnectorCommon.xsd,~~ ~~ConnectorContext.xsd,~~ ~~EventService.xsd,~~ ~~ProductInformation.xsd,~~ ~~TelematikError.xsd~~
Systeminformationsdienst (EventService)		
	WSDL Name	EventService.wsdl
	WSDL-Version	7.2.0

	TargetNamespace	http://ws.gematik.de/conn/EventService/ WSDL/v7.2
	verwendete XSDs	CardService.xsd, CardServiceCommon.xsd, CardTerminalInfo.xsd, ConnectorCommon.xsd, ConnectorContext.xsd, EventService.xsd, ProductInformation.xsd, TelematikError.xsd

Anhang A1

~~CORS~~	~~Cross-Origin Resource Sharing~~

Anhang A5.2

~~[CORS]~~	~~W3C Recommendation (16.01.2014):~~ ~~Cross-Origin Resource Sharing~~ ~~http://www.w3.org/TR/2014/REC-cors-20140116/~~

Änderungsbedarf in WSDL und Schemata

Es wird auf EventService.Wsdl V7.2.0 und EventService.xsd V7.2.1 zurückgegangen.

Änderungsbedarf in PTV2

TIP1-A_6727 wird vollständig entfernt