Einführung der Gesundheitskarte

Errata 2 zum Konnektor PTV1 (VSDM) und PTV2 (QES) Online-Produktivbetrieb (Stufe 1)

Version:	1.0.0
Stand:	19.01.2018
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemErrata_2_Kon_PTV1_PTV2]

Betroffene Produkttypen

gemProdT_Kon_PTV2

ID	Dokument	Quelle Dokument und/oder Kapitel	Beschreibung der Änderung	**Anpassungen** an Afos; TUCs, Tabellen, Korrekturen	von Änderung betroffene Dokumente
C_6087	gemSpec_Kon	TAB_KON_066, Anhang B3	Rückmeldungen der Hersteller im Rahmen der Implementierung machen Anpassungen der Profilierung des VerificationReport nötig.	siehe C_6087_Anlage	gemSpec_Kon, gemProdT_Kon_PTV2 SignatureService.xsd
C_6091	gemSpec_Kon	TIP1-A_4621	Gemäß C_5579 wird bei der CMS-Verschlüsselung nun Authenticated-Enveloped-Data statt Enveloped-Data als Content-Type verwendet. Die entsprechende Anpassung des Aufrufparameters CRYPT:UnprotectedProperties der Operation EncryptDocument fehlt noch.	Die folgende Anpassung für die Verwendung des Aufrufparameters CRYPT:UnprotectedProperties der Operation EncryptDocument wird vorgenommen: alt: Die Elemente ./UnprotectedProperties/Property/Value/CMSAttribute müssen base64/DER-kodiert ein vollständiges ASN.1-Attribute enthalten, definiert in **[CMS#6.1.EnvelopedData Type]**. Es muss bei der Erstellung des CMS-Containers unter **UnprotectedAttributes** aufgenommen werden. Das zugehörige Element ./UnprotectedProperties/Property/Identifier wird nicht ausgewertet. neu: Die Elemente ./UnprotectedProperties/Property/Value/CMSAttribute müssen base64/DER-kodiert ein vollständiges ASN.1-Attribute enthalten, definiert in **[CMS#9.1.AuthenticatedData Type]**. Es muss bei der Erstellung des CMS-Containers unter **"unauthAttrs"** aufgenommen werden. Das zugehörige Element ./UnprotectedProperties/Property/Identifier wird nicht ausgewertet.	gemSpec_Kon, gemProdT_Kon_PTV2

C_6092	gemSpec_Kon	TIP1-A_5010	Der Parameter dss:Properties für den Aufruf der Operation SignDocument ist vorgesehen, um bei fachlichem Bedarf zusätzliche signierte und unsignierte Eigenschaften (Properties) bzw. Attribute in die Signatur einfügen zu können. Es gibt keinen generellen Mechanismus, wie beliebige solche Attribute in gleicher Weise in CAdES, PAdES und XAdES aufgenommen werden können. Die Intention für die Verwendung dieses Parameters ist, dass alle Attribute, für die er verwendet wird, explizit in seiner Beschreibung spezifiziert werden. Derzeit wird der Parameter im Rahmen von KOM-LE verwendet. Da es Verständnisprobleme bzgl. des intendierten Umfangs des Parameters gegeben hat, soll die Formulierung der Beschreibung geschärft werden.	Beschreibung des Parameters dss:Properties in TIP1-A_5010: alt: 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. 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. neu: 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.	gemSpec_Kon, gemProdT_Kon_PTV2
C_6130	gemSpec_Perf	Tab_gemSpec_Perf_Konnektor	**Performancevorgaben QES-Konnektor (25-MB-Dokumente)** Die Performance der Operation SignDocument, VerifyDocument, EncryptDocument und DecryptDocument ist bei großen Dokumenten vom Typ der Dokumente und dem Signatur-/Verschlüsselungsverfahren wesentlich abhängig. Die Performancevorgaben spiegeln das noch nicht wider und werden daher für große Dokumente präzisiert. Für diese Operationen werden weitere Präzisierungen der Rahmenbedingungen vorgenommen - im Einzelnen aufgeführt im Vorspann der C_6130_Anlage.	siehe C_6130_Anlage	gemSpec_Perf, gemProdT_Kon_PTV2

C_6132	gemProdT_Kon_PTV2.doc		**Modellierung der SigProxy-Afos** Die über viele Dokumente verstreuten Signaturproxy-Anforderungen sollen so modelliert werden, dass die Afos im Produkttypsteckbrief des Konnektors in getrennten Tabellen (und dann auch in separaten Kapiteln) erscheinen. Gleichzeitig sollen keine neuen Produkttypsteckbriefe für den Signaturproxy definiert werden.	siehe C_6132_Anlage	gemProdT_Kon_PTV2
C_6181	gemSpec_Kon_SigProxy	TAB_SIG_851	Die beschriebenen UseCases in der SigProxy-Spezifikation im Kapitel 2.5. Verify_Document: f. und g. erwecken den Eindruck, dass der Benutzer das Ergebnis der Signaturprüfung, das an das aufrufende Clientsystem zurückgeliefert wird, nach der Prüfung noch beeinflussen kann. Jedoch ist dies aufgrund der Beschreibung des Parameters TvMode in VerifyDocument nicht möglich.	Aus den Anwendungsfällen (Kap.2.5) und den Ablaufdiagrammen (Kap.2.6 Abbildung 4: Ablauf der Operation VerifyDocument) geht klar hervor, dass eine Interaktion mit dem Anwender zur Bewertung des Prüfergebnisses erfolgen soll. Folgende Anpassungen werden durchgeführt: Neue Afo **TIP1-A_5701** in Kapitel 3 zwischen TIP1-A_4673 und TIP1-A_5405 mit dem Titel "**SigProxy: Bewertung von INCONCLUSIVE bei der Signaturprüfung durch den Nutzer**" und dem Inhalt "**Falls bei der Signaturprüfung im Anzeigemodus (TvMode=UNCONFIRMED) das VerificationResult INCONCLUSIVE ermittelt wurde, KANN der Signaturproxy dem Nutzer die Möglichkeit geben, das Ergebnis INCONCLUSIVE in VALID oder INVALID umzuwandeln. Das resultierende Ergebnis wird im VerificationResult im Element SIG:HighLevelResult an das aufrufende Clientsystem zurückgegeben und der dazugehörige VerificationReport wird hierbei nicht verändert.**" In der TAB_SIG_851 in der Zeile TvMode: *alt:* (siehe weitere Anzeige gemäß TIP1-A_4673 Anzeige verpflichtender Parameter bei Signaturprüfung und Beschreibung Parameter TvMode bei Operation SignDocument). *neu:* (siehe weitere Anzeige gemäß TIP1-A_4673 **SigProxy:** Anzeige verpflichtender Parameter bei Signaturprüfung**, TIP1-A_5701 SigProxy: Optionale Anzeige bei Signaturprüfung** und Beschreibung Parameter TvMode bei Operation SignDocument). Prüfverfahren für TIP1-A_5701: Funktionale Eignung-Test	gemSpec_Kon_SigProxy, gemProdT_Kon_PTV2

C_6240	gemSpec_Kon	TIP1-A_6727	Damit die für die Clientsystemschnittstelle definierten Web-Services auch von einer Webanwendung aus einem Webbrowser heraus genutzt werden können, unterstützt der Konnektor den Cross-Origin-Resource-Sharing (CORS) Mechanismus (TIP1-A_6727). Informationen über gesteckte eGKs dürfen hierbei nicht im genannten Anwendungsfall abrufbar sein. Der Schutz der Versichertendaten wird über folgende mit dem BSI abgestimmte Sicherheitsfunktionen erreicht: 1. Es wird eine neue Operation GetLeCards eingeführt, welche ausschließlich Informationen über Karten vom Typ SMC-B und HBAx bereitstellt. TIP1-A_6727 wird dahingehend geändert, dass GetCards durch GetLeCards ersetzt wird, d.h. die Operation GetLeCards ist durch CORS nutzbar, nicht aber GetCards. 2. Der Administrator hat an der Managementschnittstelle die Möglichkeit CORS abzuschalten. Es wird der Schalter CORS_Mode (Enabled/Disabled; Default: Enabled) zum Aktivieren bzw. Deaktivieren von CORS ergänzt. 3. Der Konnektor darf bei positiven CORS-Responses keine Informationen im http-Header zurückgeben, die dem Browser ein Caching der Antwort ermöglichen. (z.B. Cache-Control: no-cache)	siehe C_6240_Anlage	gemSpec_Kon EventService.xsd EventService.wsdl gemProdT_Kon_PTV2
C_6256	gemSpec_Kon	Kapitel 4.1.8 Signaturdienst	In [gemSpec_Kon], Kapitel 4.1.8 Signaturdienst, sind zwei Dienste enthalten: • Signaturdienst • Authentifizierungsdienst Es gilt beide Dienste ohne fachliche Änderung in separaten Kapiteln darzustellen. Hierzu werden alle Aspekte des Authentifizierungsdienstes aus Kapitel 4.1.8 entfernt und in ein separates Kapitel 4.1.13 ausgelagert.	siehe C_6256_Anlage	gemSpec_Kon

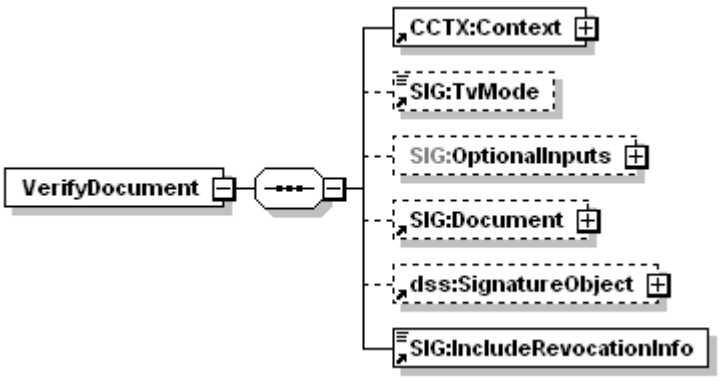
C_6257	gemSpec_Perf	GS-A_5522	Anforderung GS-A_5522 gibt eine maximale Bearbeitungszeit für die Validerung auf einfache und vollständige Anzeigbarkeit der Referenzdokumente TIFF_25MB, TIFF_1MB, PDFA_2b_25MB_Bilder_und_Text, PDFA_2b_1MB_Komplex vor. Die von der gematik aktuell bereitgestellten Dokumente der Version 1.0.0 erfüllen die Vorgaben für einfache Anzeigbarkeit, aber nicht für vollständige Anzeigbarkeit. Wenn die Prüfung auf vollständige Anzeigbarkeit beim ersten Fehler abbricht, läßt sich mit den Referenzdokumenten nicht sicherstellen, dass die Vorgabezeiten für valide Dokumente eingehalten werden. Die Referenzdokumente sind so anzupassen, dass sie die Kriterien für vollständige Anzeigbarkeit erfüllen.	Dokumente TIFF_25MB, TIFF_1MB, PDFA_2b_25MB_Bilder_und_Text, PDFA_2b_1MB_Komplex werden korrigiert und und als Dateien TIFF_25MB_V1.0.1.tif, TIFF_1MB_V1.0.1.tif, PDFA_2b_25MB_Bilder_und_Text_V1.0.1.pdf PDFA_2b_1MB_Komplex_V1.0.1.pdf bereitgestellt. Im Produkttypsteckbrief Konnektor PTV 2 werden in Kapitel 4.3. die Versionsangaben für die korrigierten Dateien von 1.0.0 auf 1.0.1 angepasst.	gemProdT_Kon_PTV2, TIFF_25MB, TIFF_1MB, PDFA_2b_25MB_Bilder_und_Text, PDFA_2b_1MB_Komplex
C_6266	gemSpec_Kon gemSpec_PKI	TUC_KON_160 TUC_KON_152 TUC_KON_151 TUC_KON_037 TUC_PKI_030	Die QES-Zertifikatsprüfung beinhaltet aktuell eine OID-Prüfung über eine Policy-Liste. Diese Prüfung verhindert die Nutzung von HBA-Vorläuferkarten. Für eine eIDAS-konforme QES-Zertifikatsprüfung ist diese OID-Prüfung nicht notwendig. Damit QES-Signaturen auch mit Vorläuferkarten erstellt und mit Vorläuferkarten erstellte Signaturen geprüft werden können, wird die OID-Prüfung in der QES-Zertifikatsprüfung entfernt.	siehe C_6266_Anlage	gemSpec_Kon gemSpec_PKI gemProdT_Kon_PTV2
C_6289	gemSpec_Krypt	GS-A_4358	Im Rahmen der eIDAS-Anpassungen (gemSpec_PKI, gemSpec_Krypt) wurde bisher nicht dargestellt, dass QES-CA-Zertifikate einer Schlüsselgeneration nicht mit dem jeweiligen Schlüsselalgorithmus des Zertifikats signiert sein müssen. Dies bedeutet QES-CA-Zertifikate mit RSA Public Key müssen nicht mit RSA signiert sein und QES-CA-Zertifikate mit ECDSA Public Key müssen nicht mit ECDSA signiert sein. Der VDA kann über den Algorithmus frei entscheiden. Darüber hinaus kann für QES-EE-Zertifikate auch das Paddingverfahren RSASSA-PSS verwendet werden.	siehe C_6289_Anlage	gemSpec_Krypt gemProdT_Kon_PTV2

C_6299	gemSpec_Kon	EncryptDocument	Beim Aufruf der Operation EncryptDocument zum Verschlüsseln eines XML-Dokuments mit dem Verschlüsselungsverfahren CMS gibt es zwei Möglichkeiten das Dokument zu übergeben: 1. In CONN:Document/dss:Base64Data 2. In CONN:Document/CONN:Base64XML. Im ersten Fall wird ein binäres Dokument erwartet, dessen innere Struktur nicht weiter überprüft wird, passend zum Verschlüsselungsverfahren CMS, das keine innere Struktur des Dokuments voraussetzt. Im zweiten Fall wird ein XML-Dokument erwartet, für das die XML-Wohlgeformtheit geprüft wird, obwohl diese für den Verschlüsselungsprozess keine Rolle spielt. Die in dieser Änderung adressierte Optimierung besteht im Streichen der zweiten Variante.	Die Beschreibung des Parameters "EncryptionType" wird in TIP1-A_4621/TAB_KON_071 wie folgt geändert: alt: … In den Fällen CMS und S/MIME wird ein Base64-codiertes Binär-Dokument im Element CONN:Document/dss:Base64Data übergeben oder ein XML-Dokument im Element CONN:Document/CONN:Base64XML. ... neu: … In den Fällen CMS und S/MIME wird ein Base64-codiertes Binär-Dokument im Element CONN:Document/dss:Base64Data übergeben. ...	gemSpec_Kongem ProdT_Kon_PTV2

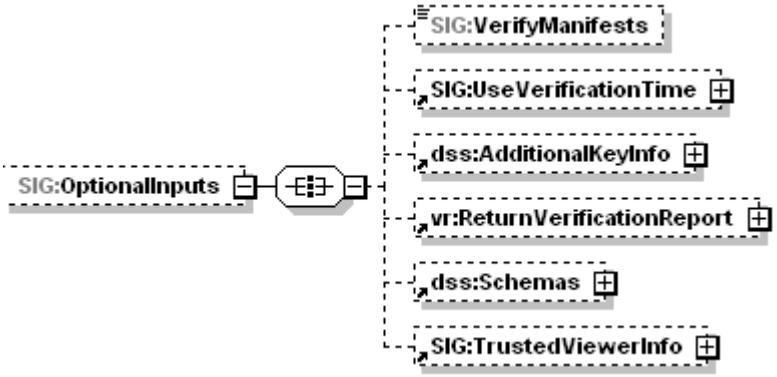
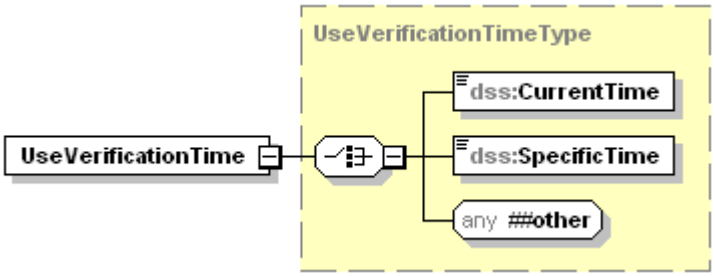
Aus den Rückmeldungen der Hersteller im Rahmen der Umsetzung des QES-Konnektors ergeben sich Anpassungen in der Profilierung des Verification Reports.

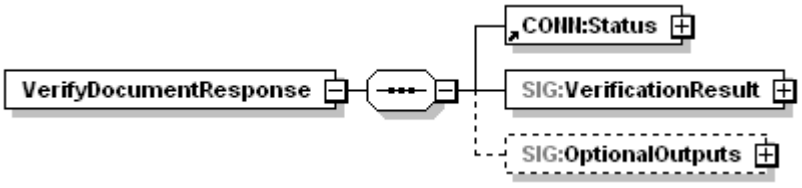
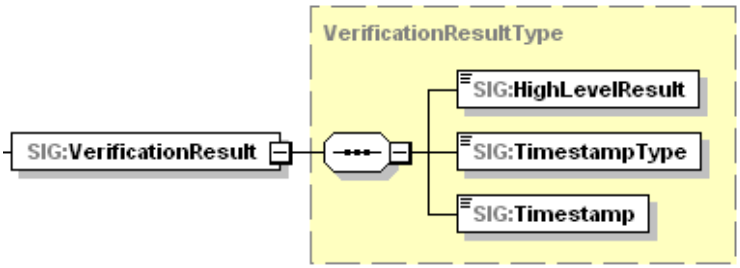
Änderungsbedarf in gemSpec_Kon

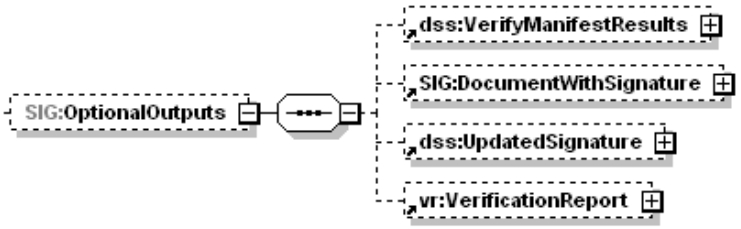
Tabelle 1: TAB_KON_066 Operation VerifyDocument (nonQES und QES)

<table>
<tr><td>Name</td><td colspan="2">VerifyDocument</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Diese Operation verifiziert die Signatur eines Dokumentes.
Der Konnektor MUSS jede konform zur Außenschnittstelle SignDocument erzeugte Signatur durch VerifyDocument prüfen können. Darüber hinaus müssen im Fall QES, falls vorhanden, auch qualifizierte Zeitstempel geprüft werden. Außerdem MÜSSEN die zusätzlich geforderten Signaturverfahren zur Dokumentensignaturprüfung unterstützt werden.
Das Ergebnis der Prüfung wird, wenn gefordert, in Form eines standardisierten Prüfberichts in einer VerificationReport-Struktur gemäß [OASIS-VR] zurückgeliefert.</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 

<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>CCTX:Context</td><td>MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet</td></tr>
<tr><td>TvMode</td><td>Der Parameter wird im Konnektor nicht ausgewertet</td></tr>
<tr><td>SIG:OptionalInputs</td><td>Enthält optionale Eingabeparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7):
Die zulässigen optionalen Eingabeparameter sind unten erläutert.</td></tr>
<tr><td>SIG:Document</td><td>Enthält im Fall der Prüfung von detached oder enveloped Signaturen das zur Signatur gehörende bzw. das diese umschließende Dokument (siehe [OASIS-DSS] Section 2.4.2 und oben)</td></tr>
<tr><td>dss:SignatureObject</td><td>Enthält die zu prüfende Signatur, wenn sie nicht im Dokument selbst eingebettet ist ([OASIS-DSS] Kapitel 4.1). Hierbei werden XML-Signaturen als ds:Signature Element und alle anderen Signaturen als dss:Base64Signature mit entsprechend gesetztem</td></tr>
</table></td></tr>
</table>

		Type-Attribut (siehe `SignatureType`, Operationen SignDocument und ExternalAuthenticate) übergeben, wobei die nachfolgenden Werte unterstützt werden müssen: • **CMS-Signatur** urn:ietf:rfc:5652 • **S/MIME-Signatur** urn:ietf:rfc:5751 • **PDF-Signatur** http://uri.etsi.org/02778/3 • **PKCS#1-Signatur** urn:ietf:rfc:3447
	`SIG:IncludeRevocationInfo`	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturprüfung vorliegenden Sperrinformationen anfordern. Ist bereits eine Sperrinformation eingebettet, so wird die neue Sperrinformation zusätzlich eingebettet. Für in einer Gegensignatur enthaltene Signaturen erfolgt keine Einbettung von Sperrinformationen.
		
	`SIG:VerifyManifests`	Durch das in [OASIS-DSS] (Abschnitt 4.5.1) definierte Element kann die Prüfung eines ggf. vorhandenen Manifests angefordert werden.
		
	`SIG:UseVerificationTime`	Durch das in [OASIS-DSS] (Abschnitt 4.5.2) spezifizierte Element kann die Prüfung der Signatur bezüglich eines durch den Aufrufer bestimmten Zeitpunktes (`Benutzerdefinierter_Zeitpunkt`) erfolgen.
	`dss:AdditionalKeyInfo`	Durch das in [OASIS-DSS] (Abschnitt 4.5.4) spezifizierte Element kann zusätzliches, für die Prüfung benötigtes, Schlüsselmaterial übergeben werden.
	`vr:ReturnVerificationReport`	Durch dieses in [OASIS-VR] spezifizierte Element kann die Erstellung eines ausführlichen Prüfberichtes angefordert werden. Der Konnektor MUSS die Anforderungen der Konformitätsstufe 2 („Comprehensive“) erfüllen und die

		Profilierung aus Anhang B3 beachten.
	`dss:Schemas`	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schematas übergeben werden, die zur Validierung des übergebenen XML-Dokumentes verwendet werden können. Zur Struktur dieses Elements siehe Beschreibung des Parameters `dss:Schemas` der Operation SignDocument.
	`SIG:ViewerInfo`	Der Parameter wird vom Konnektor nicht ausgewertet.
Rückgabe		
	`Status`	Enthält den Ausführungsstatus der Operation.
	`SIG:VerificationResult`	 Das Element `Sig:VerificationResult` enthält das Ergebnis der Prüfung als Ampel, den Typ des zugehörigen angenommenen Signaturzeitpunkts und der angenommene Signaturzeitpunkt selbst.
	`SIG:HighLevelResult`	Das Ergebnis der Prüfung (Ampelschaltung) mit folgenden Werten: • VALID: alle Signaturen sind gültig • INVALID: mindestens eine der Signaturen ist ungültig • INCONCLUSIVE: in allen anderen Fällen
	`SIG:TimestampType`	Der Typ des angenommenen Signaturzeitpunkts mit folgenden Werten: • SIGNATURE_EMBEDDED_TIMESTAMP: in der Signatur eingebetter Zeitpunkt `Ermittelter_Signaturzeitpunkt_Eingebettet` • QUALIFIED_TIMESTAMP: qualifizierter Zeitstempel über die Signatur `Ermittelter_Signaturzeitpunkt_Qualifiziert` • SYSTEM_TIMESTAMP: Systemzeit des Konnektors bei Signaturprüfung `Ermittelter_Signaturzeitpunkt_System`

		• USER_DEFINED_TIMESTAMP: benutzerdefinierter Zeitpunkt `Benutzerdefinierter_Zeitpunkt` Als Format darf jedes zum XML-Typ "dateTime" konforme Format verwendet werden (<element name="Timestamp" type="dateTime"/>). Wenn mehrere Signaturen im Dokument vorhanden sind, wird hier der angenommene Signaturzeitpunkt der jüngsten Signatur angegeben.
	`SIG:Timestamp`	Im Element SIG:Timestamp wird der zu SIG:TimestampType gehörende Zeitstempel zurückgegeben.
	`SIG:OptionalOutputs`	Enthält (angelehnt an dss:OptionalOutputs, wie in Abschnitt 2.7 von [OASIS-DSS] beschrieben) optionale Ausgangselemente: 
	`dss:VerifyManifestResults`	Dieses in Abschnitt 4.5.1 von [OASIS-DSS] definierte Element enthält Informationen zur Prüfung eines ggf. vorhandenen Signaturmanifests und wird zurückgeliefert, sofern beim Aufruf das `dss:VerifyManifest`-Element, aber nicht das `RequestVerificationReport` als optionales Eingabeelement übergeben wurde.
	`SIG:DocumentWithSignature`	Dieses in Abschnitt 4.5.8 von [OASIS-DSS] spezifizierte Element wird zurückgeliefert, falls eine in dem Dokument enthaltene Signatur (Enveloped Signature) in Verbindung mit dem `SIG:IncludeRevocationInfo`-Element geprüft wurde.
	`dss:UpdatedSignature`	Dieses in Abschnitt 4.5.8 von [OASIS-DSS] spezifizierte Element wird zurückgeliefert, falls eine abgesetzte (Detached Signature) oder umschließende (Enveloping Signature) in Verbindung mit dem `SIG:IncludeRevocationInfo`-Element geprüft wurde.
	`vr:VerificationReport`	Dieses in [OASIS-VR] spezifizierte Element wird zurückgeliefert, falls das `ReturnVerificationReport`-Element als Eingabeparameter verwendet wurde. Die Profilierung von Anhang B3 MUSS beachtet werden.
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Anhang B - Profilierung der Signatur- und Verschlüsselungsformate (normativ)

B3 – Profilierung VerificationReport

Anforderung eines ausführlichen Prüfberichts

Folgende Aufrufparameter müssen unterstützt werden:

```
<ReturnVerificationReport
xmlns="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#
                    oasis-dssx-1.0-profiles-vr-cd1.xsd">
    <IncludeVerifier>false</IncludeVerifier>
    <IncludeCertificateValues>true</IncludeCertificateValues>
    <IncludeRevocationValues>true</IncludeRevocationValues>
    <ExpandBinaryValues>false</ExpandBinaryValues>
    <ReportDetailLevel>
        urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:allDetails
    </ReportDetailLevel>
</ReturnVerificationReport>
```

Verwendung des erzeugten VerificationReport.

Für die folgenden Inhalte müssen die angegebenen Strukturen benutzen. Im Standard angegebene Pflichtfelder von erzeugten Strukturen müssen ggf. zusätzlich gefüllt werden:

a) ~~Angenommener Signaturzeitpunkt gemäß TIP1-A_5540 (QES) und TIP1-A_5545 (nonQES)~~ Prüfzeitpunkt (Systemzeit des Konnektors zum Zeitpunkt der Prüfung)

```
/VerificationReport/
    dss:VerificationTimeInfo/
        dss:VerificationTime
```

b) Signaturzeitpunkt (`Ermittelter_Signaturzeitpunkt_Eingebettet`)

```
/VerificationReport/
    IndividualReport/
        SignedObjectIdentifier/
            SignedProperties/
```

SignedSignatureProperties/
XAdES:SigningTime

~~Den zur Prüfung der Signatur verwendeten Signaturzeitpunkt als Lokalzeit ist in der SIG:VerifyDocumentResponse / SIG:VerificationResult/ SIG:Timestamp zurückgemeldet und über SIG:TimestampType qualifiziert.~~

Die Signierzeit SigningTime ist nicht nur für XAdES-Signaturen, sondern allgemein für Signaturen gemäß AdES-Baseline-Profilierung, also auch für CAdES und PAdES zu füllen.

~~c) Der Hashalgorithmus~~
~~/VerificationReport/~~
~~IndividualReport/~~
~~SignedObjectIdentifier/~~
~~DigestAlgAndValue/~~
~~ds:DigestMethod/~~
~~@Algorithm~~

c) Angenommener Signaturzeitpunkt gemäß TIP1-A_5540 (QES) und TIP1-A_5545 (nonQES)

/VerificationReport/
IndividualReport/
Details/
dss:VerificationTimeInfo/
dss:VerificationTime

d) ~~die Daten, auf die sich die Signatur bezieht~~ der binäre Wert der Signatur
/VerificationReport/
IndividualReport/
SignedObjectIdentifier/
SignatureValue

e) Kurztext ~~und andere Signierte Attribute~~

~~Kurztext wird als Teil von SIG:DocumentWithSignature außerhalb des VerificationReport in der SIG:VerifyDocumentResponse/ zurückgemeldet.~~Der signierte Kurztext wird in folgendem XML-Element zurückgegeben:

/VerificationReport/
IndividualReport/
SignedObjectIdentifier/
SignedProperties/
Other/
SIG:ShortText
~~dss:Property~~
~~./dss:Identifier (z.B.: ShortText).~~

~~./dss:Value/AttributValue (Wert)~~

f) Das folgende Element mit den Werten true/false gibt an, ob eine Zertifikatsreferenz gemäß Anhang B2 vorhanden ist (true) oder nicht (false):

/VerificationReport/
IndividualReport/
SignedObjectIdentifier/
SignedProperties/
Other/
SIG:ReferenceToSignerCertificate

g) Sämtliche signierte Attribute, deren Rückgabe nicht explizit über andere Elemente geregelt ist, werden als direkt anzeigbare Key/Value-Paare zurückgeben. Dabei sind sowohl Key und Value bereits für die Anzeige formatiert. Der Key wird in einer Zeile dargestellt. Der Value wird in mehreren Zeilen dargestellt, wobei ein Zeilenumbruch durch 'CARRIAGE RETURN (CR)' 'LINE FEED (LF)' erzeugt wird und keine weiteren Steuerzeichen erlaubt sind.

/VerificationReport/
IndividualReport/
SignedObjectIdentifier/
SignedProperties/
Other/
SIG:DisplayableAttributes

h) das Ergebnis der Signaturprüfung

/VerificationReport/
IndividualReport/
Result

i) ~~im Fall einer Gegensignatur, die Kennzeichnung als Gegensignatur und den Verweis auf die gegensignierte Signatur, im Element~~

~~/DetailedSignatureReport/~~
~~Properties/~~
~~UnsignedProperties/~~
~~UnsignedSignatureProperties/~~
~~CounterSignature/~~

handelt es sich bei der Signatur um eine Gegensignatur wird diese als solche markiert

/DetailedSignatureReport/
Properties/
UnsignedProperties/
Other/

SIG:CounterSignatureMarker

und mit

/DetailedSignatureReport/
Properties/
UnsignedProperties/
Other/
SIG:CounterSignatureMarker/
SignatureValueReference/
@IdRef

auf jede (eine oder mehrere) gegensignierte Signaturen verwiesen. Dabei zeigt IdRef auf den jeweiligen gegensignierten Signaturwert

/VerificationReport/
IndividualReport/
SignedObjectIdentifier/
ds:SignatureValue/
@Id

j) das Ergebnis der Zertifikatsprüfung,

/VerificationReport/
IndividualReport/
Details/
DetailedSignatureReport/
CertificatePathValidity/
PathValiditySummary/
ResultMajor

k) den Inhalt des Zertifikates, auf dem beruhend signiert wurde, ~~sowie den Inhalt in die Signatur eingefügter Attributzertifikate mit dem Prüfergebnis der im Kontext relevanten Rollen, umfasst die OIDs oid_hba_qes zur Identifikation einer QES-Signatur~~

/VerificationReport/
IndividualReport/
Details/
DetailedSignatureReport/
CertificatePathValidity/
PathValidityDetail/

CertificateValidity/
CertificateValue

l) den Signaturalgorithmus der Dokumentensignatur (URI, angelehnt an den Wertebereich des Feldes ds:SignatureMethod),

/VerificationReport/
IndividualReport/
Details/
DetailedSignatureReport/
SignatureOK/ ~~CertificatePathValidity/~~
SignatureAlgorithm ~~PathValidityDetail/~~
~~CertificateValidity/~~
~~SignatureOK/~~
~~SignatureAlgorithm/~~
~~Algorithm~~

m) einen aussagekräftigen Hinweis zum verminderten Beweiswert hinsichtlich Authentizität und Integrität der Signatur, wenn einer der bei der Signaturprüfung identifizierten und unterstützten Algorithmen zum Zeitpunkt der Signaturprüfung ~~Signaturerstellung~~ nicht mehr laut Algorithmenkatalog [ALGCAT] als geeignet eingestuft wird. Auszuwerten sind die Festlegungen des ALGCAT sowohl bezogen auf die Vergangenheit als auch auf die Zukunft

Für alle geprüften Zertifikate:
../
vr:CertificateValidity/
vr:SignatureOK/
vr:SignatureAlgorithm/
vr:Suitability/
./ResultMajor= urn:oasis:names:tc:dss:1.0:detail:invalid ~~indetermined~~
./ResultMessage="Algorithmen seit <Jahr> als unsicher eingestuft"

n) die PathValidity bis zur TrustAnchor-TSL mit
//CertificateValidity/ChainingOK/ResultMajor (ab dem zweiten Zertifikat in der Kette)
//CertificateValidity/CertificateStatus/CertStatusOK/ResultMajor
//CertificateValidity/CertificateValue
Für das Feld TrustAnchor ist
"urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:certDataBase"
zu verwenden.

o) Prüfergebnis des Gültigkeitszeitraums
/VerificationReport/
IndividualReport/
Details/
DetailedSignatureReport/

CertificatePathValidity/
PathValidityDetail/
CertificateValidity/
ValidityPeriodOK/
ResultMajor

p) Prüfung der Extensions

/VerificationReport/
IndividualReport/
Details/
DetailedSignatureReport/
CertificatePathValidity/
PathValidityDetail/
CertificateValidity/
ExtensionsOK/
ResultMajor

q) Zeitstempel und Herkunft der OCSP-Antwort für das Signaturzertifikat

/VerificationReport/
IndividualReport/
Details/
DetailedSignatureReport/
CertificatePathValidity/
PathValidityDetail/
CertificateValidity/
CertificateStatus/
RevocationEvidence/
OCSPValidity/
OCSPIdentifier/
./XAdES:ResponderID/XAdES:ByName
./XAdES:ProducedAt

r) OCSP Antwort für das Signaturzertifikats

/VerificationReport/
IndividualReport/
Details/
/vr:DetailedSignatureReport/
vr:CertificatePathValidity/
vr:PathValidityDetail/
vr:CertificateValidity/
vr:CertificateStatus/
vr:RevocationEvidence/

vr:OCSPValidity/
vr:OCSPValue

Sonderfälle:

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.~~

Dokument mit parallelen Signaturen

Für jede Signatur wird ein IndividualReport erzeugt.

Dokument mit Signatur und Gegensignatur

Für jede Signatur wird ein IndividualReport erzeugt.

Dokument mit Signatur und qualifiziertem Zeitstempel

Für den Zeitstempel wird ein eigener IndividualReport mit IndividualTimeStampReport erzeugt.

Anhang D - Übersicht über die verwendeten Versionen

Tabelle 2: TAB_KON_688 Version der Schemas aus dem Namensraum des Konnektors

Schemas aus dem Namensraum des Konnektors „http://ws.gematik.de/conn“		
	XSD Name	CardEvents.xsd
	XSD Schemaversion	6.0.0
	TargetNamespace	http://ws.gematik.de/conn/CardEvents/v6.0
	XSD Name	CardService.xsd
	XSD Schemaversion	8.1.1
	TargetNamespace	http://ws.gematik.de/conn/CardService/v8.1
	XSD Name	CardServiceCommon.xsd
	XSD Schemaversion	2.0.0
	TargetNamespace	http://ws.gematik.de/conn/CardServiceCommon/v2.0
	XSD Name	CardTerminalInfo.xsd
	XSD Schemaversion	8.1.0
	TargetNamespace	http://ws.gematik.de/conn/CardTerminalInfo/v8.1
	XSD Name	CardTerminalService.xsd
	XSD Schemaversion	1.1.1
	TargetNamespace	http://ws.gematik.de/conn/CardTerminalService/v1.1
	XSD Name	CertificateService.xsd
	XSD Schemaversion	6.0.1
	TargetNamespace	http://ws.gematik.de/conn/CertificateService/v6.0
	XSD Name	CertificateServiceCommon.xsd
	XSD Schemaversion	2.0.0
	TargetNamespace	http://ws.gematik.de/conn/CertificateServiceCommon/2.0
	XSD Name	ConnectorCommon.xsd
	XSD Schemaversion	5.0.0
	TargetNamespace	http://ws.gematik.de/conn/ConnectorCommon/v5.0
	XSD Name	ConnectorContext.xsd
	XSD Schemaversion	2.0.0
	TargetNamespace	http://ws.gematik.de/conn/ConnectorContext/v2.0
	XSD Name	EncryptionService.xsd
	XSD Schemaversion	6.1.1
	TargetNamespace	http://ws.gematik.de/conn/EncryptionService/v6.1
	XSD Name	EventService.xsd
	XSD Schemaversion	7.2.1
	TargetNamespace	http://ws.gematik.de/conn/EventService/ v7.2
	XSD Name	ServiceDirectory.xsd
	XSD Schemaversion	3.1.0

	TargetNamespace	http://ws.gematik.de/conn/ServiceDirectory/v3.1
	XSD Name	SignatureService.xsd
	XSD Schemaversion	~~7.4.0~~ 7.4.1
	TargetNamespace	http://ws.gematik.de/conn/SignatureService/v7.4

Änderung in SignatureService.xsd

```
<!-- Version History
        version: V7.4.1
                        ===========
                        *** C_6087: Elemente ShortText, CounterSignatureMarker,
DiplayableAttributes, ReferenceToSignerCertificate für die Verwendung im VerificationReport
        version: V7.4.0
                        ===========
                        *** Anpassungen für Signaturproxy statt xTV
        version: V7.3.0
                        ===========
                        *** ShortText wird optional (KOM-LE)
        version: V7.2.0
                        ===========
                        *** KeyReference wird gelöscht (P12: C_4528)
        version: V7.1.0
                        ===========
                        *** ServicePolicy durch GenerateUnderSignature ersetzt
         End of Version History--> <schema …
xmlns:SIG="http://ws.gematik.de/conn/SignatureService/v7.4" … version="7.4.1">
        …
        <element name="ShortText">
                <simpleType>
                        <restriction base="string">
                                <maxLength value="1000"/>
                        </restriction>
                </simpleType>
        </element>
        <element name="CounterSignatureMarker">
                <complexType>
                        <sequence>
                                <element name="SignatureValueReference"
maxOccurs="unbounded">
                                        <complexType>
                                                <attribute name="IdRef" type="IDREF"/>
                                        </complexType>
                                </element>
                        </sequence>
                </complexType>
        </element>
        <element name="DiplayableAttributes">
                <complexType>
                        <sequence>
                                <element name="DisplayableAttribute" maxOccurs="unbounded">
                                        <complexType>
                                                <sequence>
                                                        <element name="Key">
```

```
                                                    <simpleType>
                                                        <restriction base="string">
                                                            <maxLength
value="80"/>
                                                        </restriction>
                                                    </simpleType>
                                                </element>
                                                <element name="Value">
                                                    <simpleType>
                                                        <restriction base="string">
                                                            <maxLength
value="8000"/>
                                                        </restriction>
                                                    </simpleType>
                                                </element>
                                            </sequence>
                                        </complexType>
                                    </element>
                                </sequence>
                            </complexType>
                        </element>
                        <element name="ReferenceToSignerCertificate" type="boolean"/>
                        </schema>
```

Die Performancevorgaben für den Konnektor werden in folgenden Punkten präzisiert:

- Die Abhängigkeit der Bearbeitungszeiten von der Dokumentengröße wird präzisiert. Diese ist bei 25 MB großen Dokumenten vom Typ der Dokumente abhängig.
- Inzwischen sind LAN-Anbindung mit 1Gbit/s Stand der Technik, so dass vom Konnekor verlangt wird, diesen Stand zu unterstützen. Damit die Konnektorhersteller motiviert sind diese Bandbreite soweit möglich auch bereitzustellen, wird als Rahmenbedingung für die Performancemessungen die LAN-Bandbreite auf 1 Gbit/s hochgesetzt. Entsprechend werden die Bearbeitungszeiten (relevant für große Nachrichten) angepasst.
- Die Karten- und Kartenterminalreferenzzeiten in Tabelle *Tab_gemSpec_Perf* werden an die aktuell in der COS-Spezifikation vorgegebenen G2-Kartenzeiten angepasst. Die Zeiten für Kartenoperationen bei QES werden nachgeschärft.
- Für QES-Signaturen war in den Vorgabezeiten die Zeit für die Prüfung des Signaturzertifikats nicht berücksichtigt. Sie wird ergänzt.
- Die Lastvorgabe für QES-Stapelsignaturen wird korrigiert entsprechend der Annahme, dass 25% der QES-Signaturen als Stapelsignaturen erfolgen. Dabei wird berücksichtigt, dass mit einem Signaturauftrag zwei dokumente signiert werden.
- Für QES-Stapelsignaturen (Stapel > 1) wird die Zeit zum Aufbau und zur Nutzung der sicheren Kanäle zwischen den Karten zur Übertragung der PIN und des zu signierenden Hashes genauer berücksichtigt.
- VerifyDocument wird präzisiert: Im typischen Fall IncludeRevocationInfo=false wird das Dokument mit Signatur nicht von Konnektor zum Clientsystem zurücktransportiert, so dass die Größe der Antwortnachricht unabhängig von der Dokumentengröße ist. Das reduziert die Antwortzeit.
- Bisher haben sich die Werte in der Spalte ”Konnektor Gesamt” aus der Summe der Werte der übrigen Spalten zuzüglich 50 msec Puffer berechnet. Dieser Puffer ist dem Konnektor zuzurechnen und wird daher in der Spalte ”Konnektor intern” ergänzt.
- Da Karten und Kartenterminalzeiten bereits in Tabelle *Tab_gemSpec_Perf_Konnektorbearbeitungszeiten_pro_Komponenten* berücksichtigt sind, ist Tabelle *Tab_gemSpec_Perf_Normierte_Karten_Kartenterminal_Bearbeitungszeiten* nicht erforderlich. Tabelle *Tab_gemSpec_Perf_Normierte_Karten_Kartenterminal_Bearbeitungszeiten* wird daher gestrichen.
- Die Antwortzeiten des Konnektor werden über Mittelwertvorgaben an die Bearbeitungszeit im Bereich der Erwartungen gehalten. Dadurch werden insbesondere auch die Schwankungen eingegrenzt. Es hat sich herausgestellt, dass eine zusätzliche Einschränkung durch die 99%-Quantilvorgaben nicht praktikabel ist und keinen tatsächlichen Mehrwert bringt. Sie wird daher für den Konnektor gestrichen.

- encrypt_Document_Symmetric und decrypt_Document_Symmetric wird aus Tab_gemSpec_Perf_Konnektor gestrichen, weil die Operationen nicht an der Außenschnittstelle des Konnektors angeboten werden.

Änderungsbedarf in gemSpec_Perf

Kapitel 4.1.2. Produkttyp Konnektor

Skalierbarkeit

⌦ **GS-A_5327 Performance - Konnektor - Skalierbarkeit**

Der Konnektor MUSS die von 8 durchschnittlichen Anwendungen erzeugte Last im vorgegebenen Bearbeitungszeitrahmen für die vorgesehene Leistungserbringerumgebung bedienen können. Dabei wird die erzeugte Last einer durchschnittlichen Anwendung als die durch Tabelle Tab_gemSpec_Perf_Konnektor definierte Last (VSDM, KOM-LE, QES) geteilt durch 3 definiert. ⌫

Der Test von [GS-A_5327] erfolgt für den VSDM-Konnektor anhand eines QES-Produktmusters. Das QES-Produktmuster muss dafür funktional nur soweit implementiert sein, dass eine Überprüfung der Bearbeitung paralleler Requests unter der Ziellast möglich ist. Welche Tests durchgeführt werden und welche Eigenschaften dafür beim QES-Produktmuster erforderlich sind, beschreibt „Anhang D – Performancerelevante Produktmustereigenschaften des QES-Konnektors“.

Der Test von [GS-A_5327] erfolgt für den QES-Konnektor vom Verfahren her analog den Tests für den VSDM-Konnektor. Getestet wird an Hand eines breiteren Spektrums von Signatur- und Verschlüsselungsverfahren, beschrieben in „Anhang E – Testverfahren zur Prüfung der Skalierungsfähigkeit des QES-Konnektors“.

Tabelle 18 Tab_gemSpec_Perf_Konnektor – Last- und Bearbeitungszeitvorgaben

Schnittstellenoperationen	Last		Bearbeitungszeit		
	LE-U	Spitzenlast [1/h]	Größe der Anfragenachricht [kByte]	Mittelwert [msec]	~~99%-Quantil [msec]~~
Fachanwendung					
I_VSD_Service					
ReadVSD – mit Akt.-Prüfung, mit Update	1	1		6130	~~6700~~
	2	1			
	3	4			
	4	11			
ReadVSD – mit Akt.-Prüfung, ohne Update	1	50		3940	~~4480~~
	2	50			
	3	175			
	4	437			
ReadVSD – ohne Akt.-Prüfung				3820	~~4500~~
UpdateVSD – automat. Akt.-				5720	~~6260~~

Schnittstellenoperationen	Last		Bearbeitungszeit		
	LE-U	Spitzenlast [1/h]	Größe der Anfragenachricht [kByte]	Mittelwert [msec]	~~99%-Quantil [msec]~~
Prüfung, mit Update					
UpdateVSD – automat. Akt.-Prüfung, ohne Update				3130	~~3540~~
Basisdienste					
I_Sign_Operations					
sign_Document			10	~~1460~~ 1010	~~1740~~
	1	217	100	~~1480~~ 1030	~~1760~~
	2	258			
	3	351			
	4	575			
			1000	~~1670~~ 1440	~~1970~~
		~~13~~	~~25000~~	~~6680~~	~~7280~~
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	~~1280~~ 1570	~~1550~~
	1	217	100	~~1300~~ 1600	~~1570~~
	2	258			
	3	351			
	4	575			
			1000	~~1490~~ 1930	~~1770~~
		~~13~~	~~25000~~	~~6500~~	~~7100~~
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	~~1110~~
get_Certificate				220	~~330~~
I_SAK_Operations					
sign_Document_QES (Stapelgröße 1)			10	~~1580~~ 3540	~~1890~~
	1	17	100	~~1860~~ 3790	~~2210~~
	2	65			
	3	177			
	4	442			

Schnittstellenoperationen	Last		Bearbeitungszeit		
	LE-U	Spitzen-last [1/h]	Größe der Anfrage-nachricht [kByte]	Mittelwert [msec]	~~99%-Quantil [msec]~~
			1000	~~2380~~ 4070	~~2780~~
			~~25000~~	~~7740~~	~~8410~~
sign_Document_QES (XAdES, XML_25MB, enveloped)			25000	12810	~~17070~~
sign_Document_QES (CAdES, TIFF_25MB)			25000	9610	~~11920~~
sign_Document_QES (PAdES, PDFA_2b_25MB)			25000	9610	~~12540~~
sign_Document_QES (Stapelgröße 2, 2 * 100 kB Dokumente)	1	3	~~100~~ 200	~~3480~~ 8870	~~3960~~
	2	~~33~~ 11			
	3	~~89~~ 30			
	4	~~221~~ 74			
verify_Document_QES			10	~~2300~~ 2580	~~2660~~
	1	10	100	~~2340~~ 2610	~~2710~~
	2	39			
	3	113			
	4	282			
			1000	~~2560~~ 2940	~~2940~~
			~~25000~~	~~7720~~	~~8370~~
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	~~2350~~
authorize_Card (Cache)				1830	~~2140~~
I_Crypt_Operations					
encrypt_Document			10	~~1870~~ 1860	~~2190~~
	1	217	100	~~1890~~ 1880	~~2210~~
	2	258			
	3	351			
	4	575			
			1000	~~2080~~ 2200	~~2410~~
		~~13~~	~~25000~~	~~6970~~	~~7580~~
encrypt_Document (XMLEnc, TIFF_25MB, ein Empfänger)		13	25000	10600	
encrypt_Document			25000	7800	

Schnittstellenoperationen			Last		Bearbeitungszeit		
			LE-U	Spitzenlast [1/h]	Größe der Anfragenachricht [kByte]	Mittelwert [msec]	~~99%-Quantil [msec]~~
		(CMS, TIFF_25MB, ein Empfänger)					
		decrypt_Document			10	~~1110~~ 490	~~1350~~
			1	217	100	~~1120~~ 510	~~1370~~
			2	258			
			3	351			
			4	575			
					1000	~~1310~~ 820	~~1570~~
				~~13~~	~~25000~~	~~6200~~	~~6780~~
		decrypt_Document (XMLEnc, TIFF_25MB)		13	25000	8900	
		decrypt_Document (CMS, TIFF_25MB)			25000	8900	
		~~encrypt_Document_Symmetric~~			~~10~~	~~510~~	~~670~~
					~~1000~~	~~710~~	~~900~~
		~~decrypt_Document_Symmetric~~			~~10~~	~~160~~	~~250~~
					~~1000~~	~~360~~	~~500~~
	I_Cert_Verification						
		verify_Certificate				1150	~~1400~~
	I_Directory_Query						
		search_Directory	1	200		1220	~~1470~~
			2	300			
			3	500			
			4	1000			

[…]

- ~~Das Leistungserbringer-LAN wird mit einer Bandbreite von 1Gbit/sec 100 Mbit/sec, einer TCP-Effizienz von 1309/1500 = 87% und einer Latenz (eine Strecke) von 2 msec normiert angesetzt.~~ Für die Messung wird eine Bandbreite von 1Gbit/sec zwischen Clientsystem und Konnektor angenommen.

- Die Performancevorgaben aus Tab_gemSpec_Perf_Konnektor für die Basisdienste I_Sign_Operations und I_Crypt_Operations ~~im Fall von XML-Dokumenten~~ sind an Hand folgender Referenzdokumente nachzuweisen:

  - XML_25MB
  - XML_1MB
  - XML_100KB
  - XML_10KB
  - TIFF_25MB

- o TIFF_1MB
- o PDFA_2b_25MB_Bilder_und_Text
- o PDFA_2b_1MB_Komplex
- o TEXT_100KB
- o TEXT_10KB

Die konkreten Dokumente zu diesen Bezeichnern legt die Dokumentenlandkarte fest.

Stapelsignatur und gSMC-Ks

Bei der Operation sign_Document_QES in Tabelle Tab_gemSpec_Perf_Konn wurde gemäß Lastmodell aus Kapitel 3.1.4 davon ausgegangen, dass 25% der Signaturen per Stapelsignatur (Annahme Lastmodell: Stapelgröße 2) erfolgen. Tabelle 1 stellt für diese Situation dar, wie groß die Wahrscheinlichkeit ist, dass n Stapelsignaturen oder mehr parallel erfolgen müssen.

Tabelle 1 Tab_gemSpec_Perf_Konnektor_Stapelsignatur – Parallelverarbeitung gemäß Lastmodell

Lastvorgaben		Mittelwert Bearbeitungs-zeit [msec]	Sp.Last * Mittelwert Bearbeitungs-zeit [msec]	Wahrscheinlichkeit für n oder mehr parallele Bearbeitungen					
LE-U	Spitzenlasten [1/h]			n=1	n=2	n=3	n=4	n=5	n=6
1	3	4280	0,00	0,3%	0,0%	0,0%	0,0%	0,0%	0,0%
2	32		0,04	3,8%	0,1%	0,0%	0,0%	0,0%	0,0%
3	89		0,11	10,0%	0,5%	0,0%	0,0%	0,0%	0,0%
4	221		0,26	23,1%	2,9%	0,2%	0,0%	0,0%	0,0%

Lastvorgaben		Mittelwert Bearbeitungs-zeit [msec]	Sp.Last * Mittelwert Bearbeitungs-zeit [msec]	Wahrscheinlichkeit in % für n oder mehr parallele Bearbeitungen					
LE-U	Spitzenlasten [1/h]			n=1	n=2	n=3	n=4	n=5	n=6
1	3	8870	0,01	1	0	0	0	0	0
2	11		0,03	3	0	0	0	0	0
3	30		0,07	7	0	0	0	0	0
4	74		0,18	17	1	0	0	0	0

In Tabelle 1 sind alle Wahrscheinlichkeiten ~~unter 1 % grün markiert, da beim 99%-Quantil 1% der Antwortzeiten von der Vorgabe abweichen dürfen. Die Wahrscheinlichkeiten~~ über 1% ~~sind~~ rot markiert, weil hier davon ausgegangen wird, dass die Vorgaben nur erreicht werden können, wenn eine vollständige parallele Verarbeitung der Anfragen erfolgt. Geht man davon aus, dass pro gSMC-K drei logische Kanäle für die parallele Verarbeitung von Stapelsignaturen zur Verfügung stehen, dann folgt daraus, dass für das angenommene Lastszenario der Einsatz einer gSMC-K ausreichend ist.

Der Konnektor muss jedoch auch auf ein geändertes Nutzungsverhalten vorbereitet sein, wie es durch verstärkte Nutzung oder systematische Häufung von Anfragen gegen Schichtende oder durch eine verstärkte Nutzung der Stapelsignatur hervorgerufen werden kann. Angenommen in einer Leistungserbringerumgebung wird dadurch ~~werden nur Stapelsignaturen ausgeführt und es gibt eine zusätzliche Lasterhöhung von Faktor 10~~ (zusätzlich zum angenommenen Spitzenlastfaktor) die Last um den Faktor 30 erhöht, dann stellt sich die Situation aus Tabelle 26 wie folgt dar:

Tabelle 2 Tab_gemSpec_Perf_Konnektor_Stapelsignatur_Perspektivisch - Parallelverarbeitung perspektivisch

~~Last~~		~~B.zeit [msec]~~	~~Sp.Last * B.zeit~~	~~Wahrscheinlichkeit für n oder mehr parallele Bearbeitungen~~											
~~LE-U~~	~~Spitzen-lasten [1/h]~~			~~n=1~~	~~n=2~~	~~n=3~~	~~n=4~~	~~n=5~~	~~n=6~~	~~n=7~~	~~n=8~~	~~n=9~~	~~n=10~~	~~n=11~~	~~n=12~~
~~1~~	~~170~~	~~4220~~	~~0,2~~	~~18%~~	~~2%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~
~~2~~	~~648~~		~~0,8~~	~~54%~~	~~18%~~	~~4%~~	~~1%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~
~~3~~	~~1770~~		~~2,1~~	~~88%~~	~~62%~~	~~35%~~	~~16%~~	~~6%~~	~~2%~~	~~1%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~	~~0%~~
~~4~~	~~4421~~		~~5,3~~	~~99%~~	~~97%~~	~~90%~~	~~77%~~	~~60 %~~	~~43%~~	~~28%~~	~~16%~~	~~9%~~	~~4%~~	~~2%~~	~~1%~~

Last		Mittelwert Bearbeitungs-zeit [msec]	Sp.Last * Mittelwert Bearbeitungs-zeit [msec]	Wahrscheinlichkeit in % für n oder mehr parallele Bearbeitungen											
LE-U	Spitzen-lasten [1/h]			1	2	3	4	5	6	7	8	9	10	11	12
1	90	8870	0,2	19	2	0	0	0	0	0	0	0	0	0	0
2	330		0,8	55	19	5	1	0	0	0	0	0	0	0	0
3	900		2,2	89	64	37	18	7	2,4	1	0	0	0	0	0
4	2220		5,4	100	97	91	79	63	46	31	18	10	5	2	1

Anhang B – Modelldetails

B1 – Normierte Karten- und Kartenterminalbearbeitungszeiten für den Messaufbau

In den Vorgaben für die Bearbeitungszeiten des Konnektors werden pro Kartenoperation normierte Bearbeitungszeiten für Kartenterminal und Karte in Summe angesetzt. Sie werden durch Tabelle 45 festgelegt.

Tabelle 3 Tab_gemSpec_Perf_Normierte_Karten_Kartenterminal_Bearbeitungszeiten

Kartenoperationen	Bearbeitungszeiten (Karte + Kartenterminal)	
	volumen-unabhängig [msec]	volumen-abhängig [msec/kByte]
VERIFY	122	0
RESET RETRY COUNTER	130	0
CHANGE REFERENCE DATA	130	0
GET PIN STATUS	60	0
Select	65	0
PSO Decipher	865	0
MSE Set	65	0
PSO Compute Digital Signature	800	0
READ BINARY	69	80
UPDATE BINARY	74	146
Text im Display des Kartenterminals anzeigen (inkl. RoundTrip)	160	0
Reset ICC	210	0
Select DF.ESIGN	42	0
Read Binary EF.C.CH.AUT mit SFID	156	0
MSE SET Puk.RCA.CS PSO Verify Certificate aus EF.C.CA_SMC.CS der SMC bzw. EF.C.CA_eGK.CS der eGK	330	0
MSE SET Public Key aus EF.C.CA_SMC.CS der SMC bzw. EF.C.CA_eGK.CS der eGK PSO Verify Certificate aus EF.C.SMC.AUTR_CVC der SMC bzw. EF.C.eGK.AUTR_CVC der eGK	330	0
MSE SET Public Key aus EF.C.SMC.AUTR_CVC der SMC bzw. aus EF.C.eGK.AUT_CVC der eGK Get Challenge MSE SET PrK.SMC.AUTR_CVC bzw. PrK.eGK.AUT_CVC Internal Authenticate External Authenticate	1070	0
MSE Set PuK.SMC.AUTR_CVC	25	0
Internal Authenticate	750	0
ReadBinary EF.ATR	144	0
Select DF.HCA	50	0
MSE SET SK.VSD	31	0

Kartenoperationen	Bearbeitungszeiten (Karte + Kartenterminal)	
	volumen-unabhängig [msec]	volumen-abhängig [msec/kByte]
Get Challenge	24	0
MutualAuthenticate	179	0
Update Binary EF.StatusVD mit SFID	88	0
Update Binary EF.PD mit SFID	161	0
Update Binary EF.VD mit SFID	222	0
Update Binary EF.GVD mit SFID	131	0
Read Binary EF.StatusVD mit SFID	40	0
Read Binary EF.PD mit SFID	112	0
Read Binary EF.VD mit SFID	157	0
Read Binary EF.GVD mit SFID	86	0
Append Record mit SFID EF.Logging	98	0
Update Binary mit SFID EF.Prüfungsnachweis	96	0
Read Binary mit SFID EF.Prüfungsnachweis	61	0

Tabelle 40 Tab_gemSpec_Perf_Konnektorbearbeitungszeiten_pro_Komponente

Schnittstellenoperationen	Konnektor Gesamt [msec]	Konnektor intern mit LE-LAN [msec]	Kartenterm. + Karte [msec]	LE-LAN [msec]	OCSP + Zugangsnetz+ Zentr.Netz [msec]
Lesen VSD mit Onlineprüfung mit Aktualisierung	6140	1210 1260	3780	0	1100
Lesen VSD mit Onlineprüfung ohne Aktualisierung	3940	740 790	3150	0	0
Lesen VSD ohne Onlineprüfung	3820	560 610	3210	0	0
Automatische Onlineprüfung mit Aktualisierung der VSD	5720	980 1030	3590	0	1100
Automatische Onlineprüfung ohne Aktualisierung der VSD	3130	410 460	2670	0	0
I_Sign_Operations::sign_Document (10 kB)	1460 1010	300	1100 710	10	0
I_Sign_Operations::sign_Document (100 kB)	1470 1030	300 320	1100 710	20	0
I_Sign_Operations::sign_Document (1 MB) (XAdES, XML_1MB, enveloped) (CAdES, TIFF_1MB, detached) (PAdES, PDFA_2b_1MB_Komplex)	1670 1440	330 730	1100 710	190	0
I_Sign_Operations::sign_Document (25 MB)	6680	930	1100	4600	0
I_Sign_Operations::sign_Document (XAdES, XML_25MB, enveloped)	10500	9790	710	0	0

Schnittstellenoperationen	Konnektor Gesamt [msec]	Konnektor intern mit LE-LAN [msec]	Kartenterm. + Karte [msec]	~~LE-LAN [msec]~~	OCSP + Zugangsnetz+ Zentr.Netz [msec]
I_Sign_Operations::sign_Document (CAdES, TIFF_25MB, detached)	7300	6590	710	0	0
I_Sign_Operations::sign_Document (PAdES, PDFA_2b_25MB_Bilder_und_Text)	7300	6590	710	0	0
I_Sign_Operations::verify_Document (10 kB)	~~1290~~ 1570	~~110~~ 470	~~20~~ 0	~~10~~	1100
I_Sign_Operations::verify_Document (100 kB)	~~1310~~ 1600	~~110~~ 500	~~20~~ 0	~~30~~	1100
I_Sign_Operations::verify_Document (1 MB) (XAdES, XML_1MB, enveloped) (CAdES, TIFF_1MB, detached) (PAdES, PDFA_2b_1MB_Komplex)	~~1500~~ 1930	~~140~~ 830	~~20~~ 0	~~190~~	1100
~~I_Sign_Operations::verify_Document (25 MB)~~	~~6510~~	~~740~~	~~20~~	~~4600~~	~~1100~~
I_Sign_Operations::verify_Document (XAdES, XML_25MB, enveloped, IncludeRevocationInfo=false)	9000	7900	0	~~2350~~	1100
I_Sign_Operations::verify_Document (CAdES, TIFF_25MB, IncludeRevocationInfo=false)	9000	7900	0	~~2350~~	1100
I_Sign_Operations::verify_Document (PAdES, PDFA_2b_25MB, IncludeRevocationInfo=false)	10600	9500	0	~~2350~~	1100
I_SAK_Operations::sign_Document_QES (10KB)	~~1580~~ 3540	~~470~~ 520	~~1050~~ 910	~~10~~	~~0~~ 2100
I_SAK_Operations::sign_Document_QES (100KB, Stapelgröße 1, SE#1)	~~1860~~ 3790	~~720~~ 770	~~1050~~ 910	~~40~~	~~0~~ 2100
I_SAK_Operations::sign_Document_QES (100KB, Stapelgröße 2, SE#2)	~~3480~~ 8870	~~1380~~ 1430	~~1980~~ 5330	~~70~~	~~0~~ 2110
I_SAK_Operations::sign_Document_QES (1MB)	~~2380~~ 4070	~~1000~~ 1050	~~1050~~ 910	~~280~~	~~0~~ 2110
~~I_SAK_Operations::sign_Document_QES (25MB)~~	~~7740~~	~~1850~~	~~1050~~	~~4790~~	~~0~~
I_SAK_Operations::sign_Document_QES (XAdES, XML_25MB, enveloped)	12810	9790	~~1050~~ 910	~~4700~~	2110
I_SAK_Operations::sign_Document_QES (CAdES, TIFF_25MB)	9610	6590	~~1050~~ 910	~~4700~~	2110
I_SAK_Operations::sign_Document_QES (PAdES, PDFA_2b_25MB)	9610	6590	~~1050~~ 910	~~4700~~	2110
I_SAK_Operations::verify_Document_QES (10KB)	~~2300~~ 2580	~~130~~ 470	0	~~10~~	2110
I_SAK_Operations::verify_Document_QES (100KB)	~~2340~~ 2610	~~160~~ 500	0	~~20~~	2110
I_SAK_Operations::verify_Document_QES (1 MB)	~~2560~~	~~210~~	0	~~190~~	2110

Schnittstellenoperationen	Konnektor Gesamt [msec]	Konnektor intern mit LE-LAN [msec]	Kartenterm. + Karte [msec]	~~LE-LAN [msec]~~	OCSP + Zugangsnetz+ Zentr.Netz [msec]
	2940	830			
~~I_SAK_Operations::verify_Document_QES (25 MB)~~	~~7720~~	~~960~~	~~0~~	~~4600~~	~~2110~~
I_SAK_Operations::verify_Document_QES (XAdES, XML_25MB, enveloped, IncludeRevocationInfo=false)	~~9600~~ 10010	~~5120~~ 7900	0	~~2350~~	2110
I_SAK_Operations::verify_Document_QES (CAdES, TIFF_25MB, IncludeRevocationInfo=false)	~~9600~~ 10010	~~2420~~ 7900	0	~~2350~~	2110
I_SAK_Operations::verify_Document_QES (PAdES, PDFA_2b_25MB, IncludeRevocationInfo=false)	~~10200~~ 11610	~~5720~~ 9500	0	~~2350~~	2110
I_KV_Card_Unlocking::authorize_Card (no Cache)	2020	~~50~~ 100	1920	~~0~~	0
I_KV_Card_Unlocking::authorize_Card (Cache)	1830	~~50~~ 100	1730	~~0~~	0
I_Crypt_Operations::encrypt_Document (10 kB)	~~1880~~ 1860	~~700~~ 760	~~20~~ 0	~~10~~	1100
I_Crypt_Operations::encrypt_Document (100 kB)	~~1900~~ 1880	~~700~~ 780	~~20~~ 0	~~30~~	1100
I_Crypt_Operations::encrypt_Document (1 MB)	~~2080~~ 2200	~~720~~ 1100	~~20~~ 0	~~190~~	1100
~~I_Crypt_Operations::encrypt_Document (25 MB)~~	~~6970~~	~~1200~~	~~20~~	~~4600~~	~~1100~~
I_Crypt_Operations::encrypt_Document (XMLEnc, XML_25MB, ein Empfänger)	10600	9500	0	~~4700~~	1100
I_Crypt_Operations::encrypt_Document (CMS, TIFF_25MB, ein Empfänger)	7800	6700	0	~~4700~~	1100
I_Crypt_Operations::decrypt_Document (10 kB)	~~1110~~ 490	~~100~~ 150	~~950~~ 340	~~10~~	0
I_Crypt_Operations::decrypt_Document (100 kB)	~~1120~~ 510	~~100~~ 170	~~950~~ 340	~~20~~	0
I_Crypt_Operations::decrypt_Document (1 MB) (XMLEnc, XML_1MB) (CMS, TIFF_1MB)	~~1310~~ 820	~~120~~ 480	~~950~~ 340	~~190~~	0
~~I_Crypt_Operations::decrypt_Document (25 MB)~~	~~6200~~	~~600~~	~~950~~	~~4600~~	~~0~~
I_Crypt_Operations::decrypt_Document (XMLEnc, XML_25MB)	8900	8560	~~930~~ 340	~~4700~~	0
I_Crypt_Operations::decrypt_Document (CMS, TIFF_25MB)	8900	8560	~~930~~ 340	~~4700~~	0
~~I_Crypt_Operations::encrypt_Document_Symmetric (10 kB)~~	~~510~~	~~450~~	~~0~~	~~10~~	~~0~~
~~I_Crypt_Operations::encrypt_Document_Symmetric (1 MB)~~	~~710~~	~~470~~	~~0~~	~~190~~	~~0~~
~~I_Crypt_Operations::decrypt_Document_Symmetric (10 kB)~~	~~160~~	~~100~~	~~0~~	~~10~~	~~0~~

Schnittstellenoperationen	Konnektor Gesamt [msec]	Konnektor intern mit LE-LAN [msec]	Kartenterm. + Karte [msec]	~~LE-LAN [msec]~~	OCSP + Zugangsnetz+ Zentr.Netz [msec]
~~I_Crypt_Operations::decrypt_Document_Symmetric (1 MB)~~	~~360~~	~~120~~	~~0~~	~~190~~	~~0~~
I_Cert_Verification::verify_Certificate	1150	50	0	0	1100
I_Directory_Query::search_Directory	1220	1220	0	0	0

Anhang E – Testverfahren zur Prüfung der Skalierungsfähigkeit des QES-Konnektors

Entsprechend der Lastvorgaben aus [GS-A_5327] für 8 Anwendungen wird das Messverfahren festgelegt. Auf Grund der unterschiedlichen Lastanforderungen für die beiden Ausprägungsformen „Einbox-Konnektor“ und „HighSpeed-Konnektor“ wird das Verfahren für beide Fälle dargestellt. Für beide Ausprägungsformen werden die Signaturverfahren CAdES, XAdES, PAdES und die Verschlüsselungsverfahren XMLEnc und CMS unterschieden.

Es gelten die Bearbeitungszeitvorgaben aus Tabelle Tab_gemSpec_Perf_QES-Konnektor_Skalierungsfähigkeit_Bearbeitungszeitvorgaben.

Tab_gemSpec_Perf_QES-Konnektor_Skalierungsfähigkeit_Bearbeitungszeitvorgaben

	Mittlere Bearbeitungszeit $\mu_o{}^{SOLL}$ [ms]		
	CMS, CAdES	XMLEnc, XAdES	CMS, PAdES
I_Sign_Operations::sign_Document (100 kB)	1100	1100	1100
I_Sign_Operations::sign_Document (25 MB)	7300	10500	7300
I_Sign_Operations::verify_Document (100 kB)	500	500	500
I_Sign_Operations::verify_Document (25 MB)	7900	7900	9500
I_Crypt_Operations::encrypt_Document (100 kB)	780	780	780
I_Crypt_Operations::encrypt_Document (25 MB)	6700	9500	6700
I_Crypt_Operations::decrypt_Document (100 kB)	510	510	510
I_Crypt_Operations::decrypt_Document (25 MB)	8900	8900	8900

Einbox-Konnektor

In der Lastsituation für 8 Anwendungen ergeben sich verschiedene Situationen in Bezug auf die parallele Bearbeitung von Anfragen, dargestellt in Tabelle Tab_gemSpec_Perf_Einbox_QES-Konnektor_Lastsituationen. In Situation 1 bearbeitet der Konnektor weder Operationen mit 25-MB-Dokumenten noch solche mit 100-kB-Dokumenten. In den Situationen 2 und 5 bearbeitet der Konnektor genau jeweils ein Dokument. In den übrigen Situationen liegt parallele Verarbeitung vor.

Die Situationen sind getrennt für die folgenden drei Verfahrensgruppen zu betrachten:

- Verschlüsselungsverfahren CMS und Signaturverfahren CAdES,
- Verschlüsselungsverfahren XMLEnc und Signaturverfahren XAdES,
- Verschlüsselungsverfahren CMS und Signaturverfahren PAdES.

Tab_gemSpec_Perf_Einbox_QES-Konnektor_Lastsituationen

Situationen i					
i	25 MB [Anzahl]	100 kB [Anzahl]	Wahrscheinlichkeiten p_i		
			CMS, CAdES	XMLEnc, XAdES	CMS, PAdES
1	0	0	39	37	38
2	0	1	25	24	25
3	0	2	8	8	8
4	0	3	2	2	2
5	1	0	12	13	12
6	1	1	7	8	8
7	1	2	2	3	2

Für jede der Lastsituationen i in Tab_gemSpec_Perf_Einbox_QES-Konnektor_Lastsituationen ist eine Messreihe zu erstellen. In jeder Messreihe sind vom Clientsystem jeweils ein Aufruferthread pro parallele Bearbeitung zu starten, der 100mal sign_Document, encrypt_Document, decrypt_Document und verify_Document sequentiell, direkt nacheinander aufruft. In Lastsituation 8 sind es beispielsweise 1 Thread, der 25 MB große Dokumente bearbeitet, und 3 Threads, die 100 kB große Dokumente bearbeiten.

Für jede der Lastsituationen i und der Operationen o sind die Mittelwerte ${\mu_{i,o}}^{IST}$ der Bearbeitungszeiten für die beiden Klassen 25-MB-Dokumente und 100-kB-Dokumente zu bestimmen.

Durch den Test ist pro Verfahrengruppe nachzuweisen, dass die über die Lastsituationen gemittelte Bearbeitungszeit ${\mu_o}^{IST}$ für jede Operation o kleiner als die vorgegebene Bearbeitungszeit ${\mu_o}^{SOLL}$ gemäß Tab_gemSpec_Perf_QES-Konnektor_Skalierungsfähigkeit_Bearbeitungszeitvorgaben ist:

$$\mu_o{}^{IST} < \mu_o{}^{SOLL}$$

$\mu_o{}^{IST}$ wird für 100-kB-Dokumente wie folgt gemittelt:

$$\mu_o{}^{IST} = \frac{p_2\mu_{2,o}{}^{IST} + p_3\mu_{3,o}{}^{IST} + p_4\mu_{4,o}{}^{IST} + p_6\mu_{6,o}{}^{IST} + p_7\mu_{7,o}{}^{IST}}{p_2 + p_3 + p_4 + p_6 + p_7}$$

$\mu_o{}^{IST}$ wird für 25-MB-Dokumente wie folgt gemittelt:

$$\mu_o{}^{IST} = \frac{p_5\mu_{5,o}{}^{IST} + p_6\mu_{6,o}{}^{IST} + p_7\mu_{7,o}{}^{IST}}{p_5 + p_6 + p_7}$$

HighSpeed-Konnektor

In der Lastsituation für 8 Anwendungen ergeben sich verschiedene Situationen in Bezug auf die parallele Bearbeitung von Anfragen, dargestellt in Tabelle Tab_gemSpec_Perf_HighSpeed_QES-Konnektor_Lastsituationen.

Tab_gemSpec_Perf_HighSpeed_QES-Konnektor_Lastsituationen

Situationen i					
i	25 MB [Anzahl]	100 kB [Anzahl]	Wahrscheinlichkeiten p_i		
			CMS, CAdES	XMLEnc, XAdES	CMS, PAdES
1	0	0	12	11	14
2	0	1	22	21	23
3	0	2	20	20	19
4	0	3	12	12	11
5	0	4	6	6	5
6	0	5	2	2	2
7	1	0	3	4	4
8	1	1	6	7	7
9	1	2	6	6	6
10	1	3	4	4	3
11	1	4	2	2	1
12	2	2	3	4	4

Für jede der Lastsituationen i in Tab_gemSpec_Perf_HighSpeed_QES-Konnektor_Lastsituationen ist eine Messreihe zu erstellen. In jeder Messreihe sind vom Clientsystem jeweils ein Aufruferthread pro parallele Bearbeitung zu starten, der 100 mal sign_Document, encrypt_Document, decrypt_Document und verify_Document sequentiell, direkt nacheinander aufruft. In Lastsituation 12 sind es beispielsweise 2 Threads, die 25 MB große Dokumente bearbeiten, und 2 Threads, die 100 kB große Dokumente bearbeiten.

Für jede der Lastsituationen i und die Operationen o sind die Mittelwerte ${\mu_{i,o}}^{IST}$ der Bearbeitungszeiten für die beiden Klassen 25 MB-Dokumente und 100 kB-Dokumente zu bestimmen.

Durch den Test ist nachzuweisen, dass die über die Lastsituationen gemittelte Bearbeitungszeit ${\mu_o}^{IST}$ für jede Operation o kleiner als die vorgegebene Bearbeitungszeit ${\mu_o}^{SOLL}$ gemäß Tab_gemSpec_Perf_QES-Konnektor_Skalierungsfähigkeit_Bearbeitungszeitvorgaben ist:

$$ {\mu_o}^{IST} < {\mu_o}^{SOLL} $$

${\mu_o}^{IST}$ wird für 100 kB Dokumente wie folgt gemittelt:

$$ {\mu_o}^{IST} = \frac{\sum_{i=2,3,4,5,6,8,9,10,11,12} p_i {\mu_{i,o}}^{IST}}{\sum_{i=2,3,4,5,6,8,9,10,11,12} p_i} $$

${\mu_0}^{IST}$ wird für 25 MB Dokumente wie folgt gemittelt:

$$ {\mu_o}^{IST} = \frac{\sum_{i=7}^{12} p_i {\mu_{i,o}}^{IST}}{\sum_{i=7}^{12} p_i} $$

Rahmenbedingungen

Folgende konkretisierende Rahmenbedingungen gelten für Einbox-Konnektoren und HighSpeed-Konnektoren gleichermaßen zusätzlich zu den generellen Rahmenbedingungen für die Messungen aus Kapitel 4.1.2:

- Die Messungen werden mit den Referenzdokumenten TIFF_25MB und TEXT_100KB durchgeführt.
- Es wird im Offline-Modus (MGM_LU_ONLINE = Disabled) getestet.
- Pro Aufruferthread wird eine Karte und ein Kartenterminal für Signatur und Entschlüsselung eingesetzt.
- Für die einzelnen Operationen wird konkretisiert:
    - sign_Document: nonQES
    - verify_Document: Signatur verifizieren, die in sign_Document erzeugt wurde, IncludeRevocationInfo=false
    - encrypt_Document: ein Empfänger
    - decrypt_Document: Dokument entschlüsseln, das mit encrypt_Document verschlüsselt wurde.

Folgende Afos werden verschoben von
Tabelle 3: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test"
in eine neue Tabelle
Tabelle 4: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test" spezifisch für den Signaturproxy

TIP1-A_5150	SigProxy: Anzeige definierter Dokumentenformate im Signaturproxy	gemSpec_Kon_SigProxy
TIP1-A_5531	SigProxy: PDF-Anzeige von XML-Dokumenten	gemSpec_Kon_SigProxy
TIP1-A_5687	SigProxy: Unterstützte Versionen bei PDF-Anzeige von XML-Dokumenten	gemSpec_Kon_SigProxy
TIP1-A_5688	SigProxy: XSL-FO bei PDF-Anzeige von XML-Dokumenten	gemSpec_Kon_SigProxy
TIP1-A_5404	SigProxy: Anzeige Kurztext bei Signaturerstellung	gemSpec_Kon_SigProxy
TIP1-A_5532	SigProxy: HTML/CSS-Anzeige von XML-Dokumenten	gemSpec_Kon_SigProxy
TIP1-A_5689	SigProxy: Stylesheets bei HTML/CSS-Anzeige von XML-Dokumenten	gemSpec_Kon_SigProxy
TIP1-A_5668	SigProxy: Durchreichen von Contextparametern	gemSpec_Kon_SigProxy
TIP1-A_5669	SigProxy: Interface für Operationen des Signaturproxy	gemSpec_Kon_SigProxy
TIP1-A_5686	SigProxy: Keine Transportsicherung am Signaturproxyinterface	gemSpec_Kon_SigProxy
TIP1-A_4634	SigProxy: Verbindung zwischen Konnektor und Signaturproxy	gemSpec_Kon_SigProxy
TIP1-A_5692	SigProxy: Installation des Signaturproxy	gemSpec_Kon_SigProxy
TIP1-A_5693	SigProxy: Vertrauensankerwechsel für TLS-Verbindungen	gemSpec_Kon_SigProxy
TIP1-A_5670	SigProxy: Information zur Anbindung Signaturproxy	gemSpec_Kon_SigProxy
TIP1-A_4650	SigProxy: TUC_SIG_153 “Dokumentenliste im Signaturproxy anzeigen”	gemSpec_Kon_SigProxy
TIP1-A_4656	SigProxy: Anzeige der Parameter bei QES-Signaturerstellung	gemSpec_Kon_SigProxy
TIP1-A_5683	SigProxy: Anzeige der Jobnummer	gemSpec_Kon_SigProxy
TIP1-A_4657	SigProxy: Anzeige der Vertrauenswürdigkeit von Signaturalgorithmen	gemSpec_Kon_SigProxy
TIP1-A_4658	SigProxy: Anzeige und Deselektion von Daten bei Stapelsignatur	gemSpec_Kon_SigProxy
TIP1-A_4659	SigProxy: Fortschrittsanzeige bei Stapelsignatur	gemSpec_Kon_SigProxy
TIP1-A_4660	SigProxy: Reihenfolge der Dokumente bei Stapelsignatur	gemSpec_Kon_SigProxy
TIP1-A_4661	SigProxy: Kennzeichnung unterschiedlicher Dokumententypen	gemSpec_Kon_SigProxy
TIP1-A_4662	SigProxy: Bestätigungsmodus: Warten auf Freigabe	gemSpec_Kon_SigProxy
TIP1-A_4663	SigProxy: Bestätigungsmodus: Möglichkeit zum Abbruch geben	gemSpec_Kon_SigProxy
TIP1-A_5671	SigProxy: Abbruchmöglichkeit bei Stapelsignaturverarbeitung	gemSpec_Kon_SigProxy
TIP1-A_5680	SigProxy: Löschen von Anzeigen nach Zeitablauf	gemSpec_Kon_SigProxy
TIP1-A_5681	SigProxy: Löschen von Anzeigen durch Benutzerinteraktion	gemSpec_Kon_SigProxy
TIP1-A_4664	SigProxy: Ansichtsmodus: Allein die PIN-Eingabe am Kartenterminal ist maßgeblich	gemSpec_Kon_SigProxy

TIP1-A_4665	SigProxy: Ansichtsmodus: Muss darin verbleiben wenn alles anzeigbar	gemSpec_Kon_SigProxy
TIP1-A_4666	SigProxy: Ansichtsmodus: Muss in Bestätigungsmodus umschalten wenn nicht alles anzeigbar	gemSpec_Kon_SigProxy
TIP1-A_4668	SigProxy: Bestätigung von Fehlern durch die Benutzer	gemSpec_Kon_SigProxy
TIP1-A_4673	SigProxy: Anzeige der Parameter bei Signaturprüfung	gemSpec_Kon_SigProxy
TIP1-A_5405	SigProxy: Anzeige Kurztext bei Signaturprüfung	gemSpec_Kon_SigProxy
TIP1-A_5690	SigProxy: Basisdienst Signaturdienst (nonQES und QES)	gemSpec_Kon_SigProxy
TIP1-A_5672	SigProxy: Basisdienst Dienstverzeichnisdienst (nonQES und QES)	gemSpec_Kon_SigProxy
TIP1-A_5673	SigProxy: TUC_SIG_192 “Anzeigbarkeit des Dokuments prüfen”	gemSpec_Kon_SigProxy
TIP1-A_4631	SigProxy: Bereitstellung der Anzeige	gemSpec_Kon_SigProxy
TIP1-A_5695	SigProxy: SOAP Message Transmission Optimization Mechanism	gemSpec_Kon_SigProxy
TIP1-A_5684	SigProxy: SOAP-Faults melden	gemSpec_Kon_SigProxy
TIP1-A_5691	SigProxy: Protokollierung spezifizierter Fehler	gemSpec_Kon_SigProxy
TIP1-A_5674	SigProxy: Operation SignDocument (nonQES und QES)	gemSpec_Kon_SigProxy
TIP1-A_5675	SigProxy: Operation VerifyDocument (nonQES und QES)	gemSpec_Kon_SigProxy
TIP1-A_5676	SigProxy: Bereitstellen des Dienstverzeichnisdienstes	gemSpec_Kon_SigProxy
TIP1-A_5677	SigProxy: Protokollierung personenbezogener und medizinischer Daten	gemSpec_Kon_SigProxy
TIP1-A_5678	SigProxy: Keine Protokollierung vertraulicher Daten	gemSpec_Kon_SigProxy
TIP1-A_5679	SigProxy: Starten des Signaturproxy in einer Terminal-Server-Umgebung	gemSpec_Kon_SigProxy
GS-A_5519	SigProxy: Performance - TLS-Handshake	gemSpec_Perf
GS-A_5520	SigProxy: Performance - TLS Session Resumption 1	gemSpec_Perf
GS-A_5521	SigProxy: Performance - Weiterleiten von Nachrichten	gemSpec_Perf
GS-A_5522	SigProxy: Performance - Validierung auf Anzeigbarkeit	gemSpec_Perf

Folgende Afos werden verschoben von
Tabelle 4: Anforderungen zur funktionalen Eignung "Herstellererklärung"
in eine neue Tabelle
Tabelle 6: Anforderungen zur funktionalen Eignung "Herstellererklärung"
spezifisch für den Signaturproxy

TIP1-A_5685	SigProxy: Softwareergonomie	gemSpec_Kon_SigProxy

Folgende Afos werden zusätzlich zu bestehenden Prüfverfahren auch aufgenommen in eine neue Tabelle
Tabelle 4: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test" spezifisch für den Signaturproxy

TIP1-A_4514	Verfügbarkeit einer TLS Schnittstelle	gemSpec_Kon
TIP1-A_4515	Verpflichtung zur Nutzung der TLS-Verbindung	gemSpec_Kon
GS-A_5525	TLS-Renegotiation Konnektor	gemSpec_Krypt
GS-A_5345	TLS-Verbindungen Konnektor	gemSpec_Krypt
GS-A_5542	TLS-Verbindungen (fatal Alert bei Abbrüchen)	gemSpec_Krypt
GS-A_4359	X.509-Identitäten für die Durchführung einer TLS-Authentifizierung	gemSpec_Krypt
GS-A_4385	TLS-Verbindungen, Version 1.2	gemSpec_Krypt
GS-A_4386	TLS-Verbindungen, optional Version 1.1	gemSpec_Krypt
GS-A_5530	TLS-Verbindungen, Version 1.1	gemSpec_Krypt
GS-A_4387	TLS-Verbindungen, nicht Version 1.0	gemSpec_Krypt
GS-A_4384	TLS-Verbindungen	gemSpec_Krypt
GS-A_5322	Weitere Vorgaben für TLS-Verbindungen	gemSpec_Krypt
TIP1-A_4500	Dokumentgrößen von 25 MB	gemSpec_Kon
TIP1-A_5694	SOAP Message Transmission Optimization Mechanism	gemSpec_Kon
GS-A_5038	Festlegungen zur Vergabe einer Produktversion	gemSpec_OM
TIP1-A_6727	Cross-Origin Resource Sharing	gemSpec_Kon
TIP1-A_5446	Zusätzliche Signaturverfahren für Dokumentensignaturprüfung	gemSpec_Kon

1. Änderungen in gemSpec_Kon

1.1 Änderung in “TIP1-A_6727 Cross-Origin Resource Sharing”

Die Operation GetCards wird durch eine neue Operation GetLeCards ersetzt und ist damit über CORS nicht mehr erreichbar.

Der Konnektor darf bei positiven CORS Responses keine Informationen im http-Header zurückgeben, die dem Browser ein Caching der Antwort ermöglichen. (z.B. Cache-Control: no-cache)

⌦ TIP1-A_6727 Cross-Origin Resource Sharing

Der Konnektor MUSS Cross-Origin Resource Sharing gemäß [CORS] für sämtliche für die Clientsystemschnittstelle definierten Web-Services unterstützen.

Dabei MUSS der Konnektor über den Access-Controll-Allow-Origin HTTP-Header ausschließlich explizit zugelassenen ORIGINs den Zugriff auf explizit zugelassene Operationen erlauben. Der Konnektor DARF NICHT über den Eintrag "*" im Access-Controll-Allow-Origin HTTP-Header den Zugriff über jede ORIGIN ermöglichen.

Explizit zugelassene ORIGINs sind alle ORIGINS, die sämtliche der folgenden Bedingungen erfüllen:

- ORIGIN, wie vom Aufrufer angegeben
- ORIGIN ist gemäß [RFC3986] wie folgt definiert: “https://<host>[:<port>]“
- <host> ist ein FQDN aus dem Namensraum der TI, d.h. er hat die Top Level Domain DNS_TOP_LEVEL_DOMAIN_TI

Explizit zugelassene Operationen sind alle in TAB_KON_803 aufgeführten Operationen.

Tabelle 1 TAB_KON_803 Erlaubte Operationen beim CORS-Zugriff

Name des Service	Operation
ConnectorServiceDirectory	GET /connector.sds
~~EventService~~	~~GetCards~~
EventService	GetLeCards
~~SignatureService~~ AuthSignatureService	ExternalAuthenticate
CertificateService	ReadCardCertificate
CardService	GetPinStatus
CardService	VerifyPin

Der Konnektor MUSS bei Responses auf Cross-Origin Requests den http-Header so zurückgeben, dass kein Caching der Antwort beim Browser erfolgt (z.B. Cache-Control: no-cache).⌫

1.2 Neuer Konfigurationsparameter zum Aktivieren/Deaktivieren von CORS (Einfügen in Kap.3.5.5)

⌦ TIP1-A_7223 Aktivieren/Deaktivieren von CORS

Die Managementschnittstelle MUSS es einem Administrator ermöglichen, die Funktionalität Cross-Origin Resource Sharing [CORS] zu aktivieren bzw. zu deaktivieren gemäß TAB_KON_xxx Konfigurationswerte CORS.

Tabelle xx TAB_KON_xxx Konfigurationswerte CORS

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
`CORS_Mode`	`Enabled/Disabled`	Der Administrator MUSS CORS aktivieren bzw. deaktivieren können. `Enabled`: Cross-Origin Resource Sharing gemäß [CORS] ist für die Web-Services der Clientsystemschnittstelle NICHT erlaubt. `Disabled`: Cross-Origin Resource Sharing gemäß [CORS] ist für die erlaubten Operationen der Clientsystemschnittstelle gemäß TAB_KON_803 erlaubt. Default-Wert: Enabled

⌫

1.3 Neue Operation GetLeCards im Systeminformationsdienst

⌦ TIP1-A_4603 Basisanwendung Systeminformationsdienst

Der Konnektor MUSS für Clients eine Basisanwendung Systeminformationsdienst anbieten.

Tabelle 2 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	

⌫

1.3 Neue Operation GetLeCards

Es wird Kapitel 4.1.6.5.8 neu aufgenommen:

4.1.6.5.8 GetLeCards

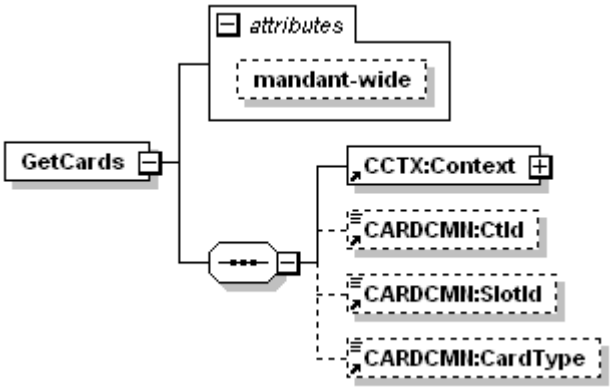
⌦ **TIP1-A_7224 Operation GetLeCards**

Der Konnektor MUSS an der Außenschnittstelle eine Operation GetLeCards, wie in Tabelle TAB_KON_xxx „Operation GetLeCards“ beschrieben, anbieten und MUSS dabei Kartentypen aus Tabelle 1: TAB_KON_500 Wertetabelle Kartentypen unterscheiden.

Tabelle 3: Tab_KON_xxx Operation GetLeCards

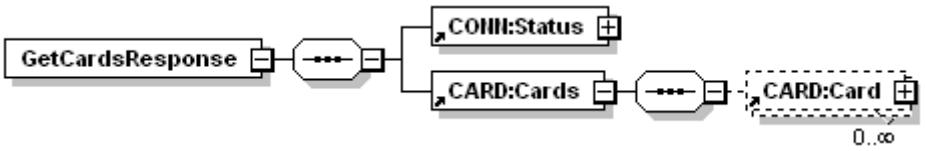
Name	GetLeCards
Beschreibung	Liefert Informationen zu den in den Kartenterminals verfügbaren Karten des Leistungerbringers zurück, die in Kartenterminals stecken, auf die Mandant und Clientsystem zugreifen dürfen. Insbesondere umfasst die Information die sog. Karten-Handles. Die Karten-Handles können bei anderen Operationen an der Außenschnittstelle des Konnektors zur Adressierung von Karten genutzt werden. Es werden ausschließlich Informationen der Kartentypen HBAx und SM-B zurückgegeben.

Aufrufparameter

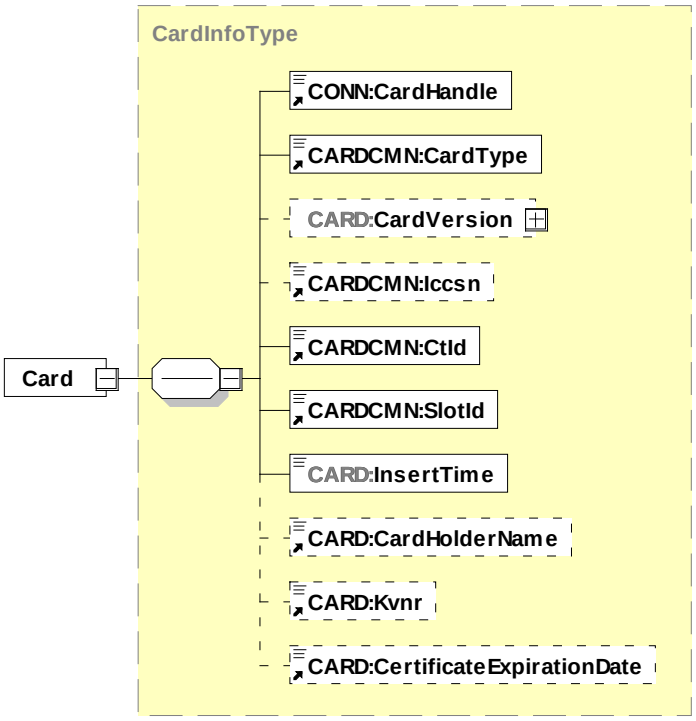


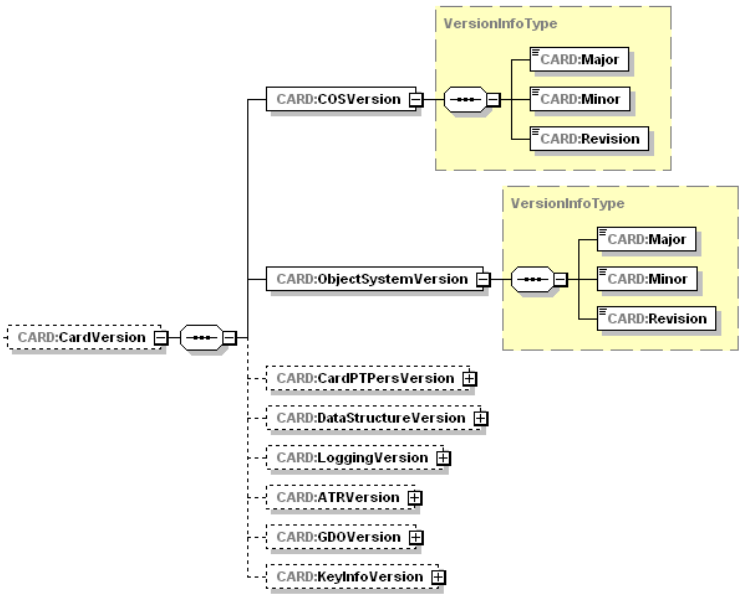
Name	Beschreibung
`@mandant-wide`	Wenn „true“, werden alle Karten zurückgegeben, auf die der Mandant und das aufrufende Clientsystem zugreifen darf. Wenn „false“ (Standardbelegung), werden nur Karten zurückgegeben, auf die von dem im Aufrufkontext spezifizierten Arbeitsplatz zugriffen werden darf.
`Context`	Aufrufkontext
`CtId`	Identifikation des Kartenterminals. Wenn angegeben, werden nur die Karten zurückgeliefert, die in diesem Kartenterminal verfügbar sind und von einem der Kartentypen KVK, HBAx oder SM-B entsprechen.
`SlotId`	Nummer des Slots, beginnend bei 1.
`CardType`	Ein Kartentyp gemäß Tabelle TAB_KON_500 „Wertetabelle Kartentypen“ als optionaler Filter. Wenn angegeben, werden nur Karten vom spezifizierten Typ zurückgegeben. Unterstützt werden die Kartentypen HBAx und SM-B.

Antwort



Name	Beschreibung
`Status`	Ergebnis der Operation
Im Element `Cards` wird die Liste der gesteckten Karten zurückgegeben. Für jede Karte wird dabei ein `Card`-Element angegeben. Leere Slots der Kartenterminals sind in dieser Liste nicht enthalten.	



Name	Beschreibung
`CardHandle`	Handle, mit dem die Karte in Folgeaufrufen adressiert werden kann. Der Konnektor garantiert, dass dieses Handle die gesteckte Karte eindeutig identifiziert und bei Entfernen der Karte aus dem Kartenterminal ungültig wird.
`CardType`	Erkannter Typ der Karte. Siehe Tabelle 1: TAB_KON_500 Wertetabelle Kartentypen
`CardVersion`	 Der Konnektor MUSS in CardVersion bei HBA und SM-B der Generation 2 die Versionsinformationen gemäß [gemSpec_Karten_Fach_TIP] übergeben, für G1+ aus

		/MF/EF.Version. Bei HBA-VK MUSS das Element weggelassen werden.
	Iccsn	Falls auslesbar, die ICC-Serial-Number der Karte.
	CtId	Identifikation des Kartenterminals, in dem die Karte steckt.
	SlotId	Nummer des Slots (beginnend bei 1), in dem die Karte steckt.
	InsertTime	Gibt den Zeitpunkt an, zu dem der Konnektor die Karte erkannt hat. Die Zeit wird mit dem Datentyp DateTime in folgendem Format angegeben: yyyy-mm-ddThh:mm:ss+hh:mm Es ist also – gemäß ISO 8601 [ISO8601] – die lokale Zeit und die Differenz zur UTC anzugeben.
	CardHolderName	Name des Karteninhabers bzw. der Institution/Organisation (subject.commonName).
	Kvnr	Das Element MUSS weggelassen werden.
	CertificateExpirationDate	Ablaufdatum des Zertifikates (AUT bzw. OSIG).
Vorbedingungen	Keine.	
Nachbedingungen	Der Zustand der Karten und der Kartenterminals bleibt unverändert.	
Hinweise	Der Aufruf darf nur den im Konnektor verwalteten aktuellen Zustand der Karte liefern und keine Abfragen an die Kartenterminals absetzen.	

Der Ablauf der Operation GetLeCards ist in Tabelle 4: TAB_KON_xxx Ablauf LeOnly beschrieben:

Tabelle 4: TAB_KON_xxx Ablauf GetLeCards

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.	TUC_KON_000 „Prüfe Zugriffsberechtigung“	Die Prüfung erfolgt durch den Aufruf TUC_KON_000 {$context.mandantId; $context.clientsystemId; $context.workplaceId; doNotNeedCardSession; @mandant-wide} Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
3.	TUC_KON_253 „Liefere Karten_Liste“	Die Liste der Karten wird erstellt und zurückgegeben. Tritt hierbei ein Fehler auf, so bricht die Operation mit dem Fehler des TUCs ab. Wurde CardType nicht angegeben, dann ermittle die Liste der Karten für CardType=HBAx und für CardType=SM-B Wenn @mandant-wide=true dann ermittle die Liste der Karten für

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
		alle Arbeitsplätze des Mandanten für das angegebene Clientsystem durch den Aufruf TUC_KON_253 „Liefere Karten_Liste“ { Clientsystem ID = $context.clientsystemId; Kartenterminal-ID = CtId; Slot-ID = SlotId; Mandanten ID = $context.mandantId; CardType = CardType } Wenn @mandant-wide=false dann ermittle die Liste der Karten für den Arbeitsplatz des Mandanten für das angegebene Clientsystem entsprechend $context durch den Aufruf TUC_KON_253 „Liefere Karten_Liste“ { Arbeitsplatz ID = $context.workplaceId; Clientsystem ID = $context.clientsystemId; Kartenterminal-ID = CtId; Slot-ID = SlotId; Mandanten ID = $context.mandantId; CardType = CardType }

Die Fehlerfälle der Operation GetLeCards sind in Tabelle 5 TAB_KON_xxx Fehlerfälle LeOnly dargestellt:

Tabelle 5 TAB_KON_xxx Fehlerfälle GetLeCards

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

⌫

1.4 Anpassung der Version in „Anhang D – Übersicht über die verwendeten Versionen“

TAB_KON_688 Version der Schemas aus dem Namensraum des Konnektors

	~~XSD Name~~	~~EventService.xsd~~
	~~XSD Schemaversion~~	~~7.2.1~~
	~~TargetNamespace~~	~~http://ws.gematik.de/conn/EventService/ v7.2~~
	XSD Name	EventService.xsd
	XSD Schemaversion	7.2.2
	TargetNamespace	http://ws.gematik.de/conn/EventService/ v7.2

Tabelle 6: TAB_KON_798 Schnittstellenversionen

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
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

1.5 Anhang E „Übersicht Konfigurationsparameter und Zustandswerte“ wird erweitert um folgenden Tabelleneintrag:

Zustandswert?	ReferenzID	Belegung	Bedeutung und Administrator-Interaktion bzw. Zustandswerte
Interne Mechanismen			
	`CORS_Mode`	`Enabled/ Disabled`	Der Administrator MUSS CORS aktivieren bzw. deaktivieren können. `Enabled`: Cross-Origin Resource Sharing gemäß [CORS] ist für die Web-Services der Clientsystemschnittstelle NICHT erlaubt. `Disabled`: Cross-Origin Resource Sharing gemäß [CORS] ist für die erlaubten Operationen der Clientsystemschnittstelle gemäß TAB_KON_803 erlaubt. Default-Wert: `Enabled`

2. Änderungen in EventService.xsd

Das Schema EventService.xsd wird erweitert um die neue Operation `GetLeCards`.

Die Schnittstellenerweiterung ist abwärts kompatibel.

EventService.xsd (alt: version="7.2.1", neu: version="7.2.2") wird wie folgt erweitert:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:EVT="http://ws.gematik.de/conn/EventService/v7.2"
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
```

```
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:CARD="http://ws.gematik.de/conn/CardService/v8.1"
xmlns:CT="http://ws.gematik.de/conn/CardTerminalInfo/v8.0"
xmlns:CARDCMN="http://ws.gematik.de/conn/CardServiceCommon/v2.0"
xmlns:HSM="http://ws.gematik.de/conn/HsmInfo/v8.0"
targetNamespace="http://ws.gematik.de/conn/EventService/v7.2" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="7.2.2">

…
        <element name="GetLeCards">
                <annotation>
                        <documentation>Abfragen der in Kartenterminals verfügbaren Karten vom Kartentyp
        HBAx und SM-B</documentation>
                </annotation>
                <complexType>
                        <sequence>
                                <element ref="CCTX:Context"/>
                                <element ref="CARDCMN:CtId" minOccurs="0"/>
                                <element ref="CARDCMN:SlotId" minOccurs="0"/>
                                <element ref="CARDCMN:CardType" minOccurs="0"/>
                        </sequence>
                        <attribute name="mandant-wide" type="boolean" use="optional" default="false"/>
                </complexType>

        </element>
        <element name="GetLeCardsResponse">
                <annotation>
                        <documentation>Antwort des Aufrufs GetLeCards</documentation>
                </annotation>
                <complexType>
                        <sequence>
                                <element ref="CONN:Status"/>
                                <element ref="CARD:Cards"/>
                        </sequence>
                </complexType>

        </element>

…

</schema>
```

3. Änderungen in EventService.wsdl

EventService.wsdl (alt: version="7.2.0", neu: version="7.2.1") wird wie folgt erweitert:

Das Schema EventService.wsdl wird erweitert um die neue Operation `GetLeCards`:

```
<definitions ...>
…
        <message name="GetLeCardsRequestMessage">
                <part name="parameter" element="EVT:GetLeCards"/>

        </message>
        <message name="GetLeCardsResponseMessage">
                        <part name="parameter" element="EVT:GetLeCardsResponse"/>

        </message>

…
        <portType …>
        …
                <operation name="GetLeCards">
                        <input message="EVTW:GetLeCardsRequestMessage"/>
```

```
                <output message="EVTW:GetLeCardsResponseMessage"/>
                <fault name="FaultMessage" message="EVTW:FaultMessage"/>

        </operation>

...

</portType>

<binding …>

...
        <operation name="GetLeCards">
                <soap:operation
        soapAction="http://ws.gematik.de/conn/EventService/v7.2#GetLeCards"/>
                <input>
                        <soap:body use="literal"/>
                </input>
                <output>
                        <soap:body use="literal"/>
                </output>
                <fault name="FaultMessage">
                        <soap:fault name="FaultMessage" use="literal"/>
                </fault>

        </operation>

...

</binding>
```

4. Änderungen in gemProdT_Kon_PTV2

Anmerkung: Die Anforderungen der folgenden Tabelle stellen einen Auszug dar und verteilen sich innerhalb der Tabelle des Originaldokuments [gemProdT_Kon_PTV2]. Alle Anforderungen der Tabelle des Originaldokuments, die in der folgenden Tabelle nicht ausgewiesen sind, bleiben unverändert bestehenden.

3.1.1 Produkttest / Produktübergreifender Test

In diesem Abschnitt sind alle funktionalen und nichtfunktionalen Anforderungen an den technischen Teil des Produkttyps verzeichnet, deren Umsetzung im Zuge von Zulassungstests durch die gematik geprüft wird.

Tabelle 3: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
TIP1-A_7224	Operation GetLeCards	gemSpec_Kon
TIP1-A_7223	Aktivieren/Deaktivieren von CORS	gemSpec_Kon

3.2.1 CC-Evaluierung

Der Produkttyp erfordert eine Zertifizierung nach ITSEC [ITSEC] oder Common Criteria (CC) [BSI_2006a] auf der Grundlage der Protection Profiles [PP_NK] und [PP_KON].

Für die Evaluierung sind die Inhalte der Schutzprofile normativ führend. Der Nachweis der im Folgenden aufgeführten Anforderungen erfolgt implizit durch die Vorlage des IT-Sicherheitszertifikats bei der gematik.

Wenn der Hersteller die Funktion CORS (TIP1-A_7223) umsetzt, muss diese auch evaluiert und zertifiziert werden. Dazu muss der Hersteller die CORS-spezifischen Inhalte aus dem [PP_KON]#7.2 in sein Security Target übernehmen.

Tabelle 5: Anforderungen zur sicherheitstechnischen Eignung "CC-Evaluierung"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
TIP1-A_7223	Aktivieren/Deaktivieren von CORS	gemSpec_Kon

In [gemSpec_Kon], Kapitel 4.1.8 Signaturdienst, sind zwei Dienste enthalten:

- Signaturdienst
- Authentifizierungsdienst

Es gilt beide Dienste ohne fachliche Änderung in separaten Kapiteln darzustellen. Hierzu werden alle Aspekte des Authentifizierungsdienstes aus Kapitel 4.1.8 entfernt und in ein separates Kapitel 4.1.13 ausgelagert.

Änderungsbedarf in gemSpec_Kon

Es wird ein neues Kapitel 4.1.13 ergänzt

4.1.13 Authentifizierungsdienste

Der Authentifizierungsdienst bietet Clientsystemen und Fachmodulen eine Schnittstelle zum Signieren von Binärstrings zum Zweck der externen Authentisierung.

Innerhalb des Authentifizierungsdienstes werden folgende Präfixe für Bezeichner verwendet:

- Events (Topic Ebene 1): *keine Events vorhanden*
- Konfigurationsparameter: *keine Konfigurationsparameter vorhanden*

Eine Prüfung der Signatur bietet der Authentifizierungsdienst nicht an. Sie wird im Rahmen der Operation VerifyDocument des Signaturdienstes angeboten.

4.1.13.1 Funktionsmerkmalweite Aspekte

4.1.13.1.1 Externe Authentisierung

⌦ TIP1-A_5437 Signaturverfahren für externe Authentisierung

Der Signaturdienst MUSS das Signaturverfahren PKCS#1 entsprechend TAB_KON_780 – Signaturverfahren Externe Authentisierung unterstützen. ⌫

Tabelle 1: TAB_KON_780 – Signaturverfahren Externe Authentisierung

Signaturformat	Standard	Dokumentformate	QES/ nonQES	Bemerkung
PKCS#1 (V2.1)	[RFC3447]	Binär	nonQES	Dieses Signaturformat DARF NUR in Verbindung mit dem zur Authentisierung vorgesehenen Schlüssel des HBAx und des SM-B genutzt werden. Die Nutzung ist auf Dokumente (Hash) von maximal 512 bit Länge beschränkt.

⌦ TIP1-A_5149 PKCS#1-Schnittstelle nur für Authentisierung mit HBAx und SM-B nutzen

Der Hersteller des Konnektors MUSS den Anwender (Clientsystem) im Handbuch des Konnektors geeignet und ausreichend darüber informieren, dass das Signaturformat PKCS#1 nur zu Authentisierungszwecken mit dem Authentisierungsschlüssel des HBAx und des SM-B verwendet werden darf.⌫

4.1.13.2 Durch Ereignisse ausgelöste Reaktionen

keine

4.1.13.3 Interne TUCs

keine

4.1.13.4 Operationen an der Außenschnittstelle

⌦ TIP1-A_5665 Basisdienst Authentifizierungsdienst

Der Konnektor MUSS Clientsystemen den Basisdienst Authentifizierungsdienst anbieten.

Tabelle 2: TAB_KON_839 Basisdienst Authentifizierungsdienst

Name	`AuthSignatureService`	
Version (KDV)	Siehe Anhang D (WSDL-Version)	
Namensraum	Siehe Anhang D	
Namensraum-Kürzel	`SIG für Schema und SIGW für WSDL`	
Operationen	**Name**	**Kurzbeschreibung**
	`ExternalAuthenticate`	Binärstring signieren (nonQES)
WSDL	`AuthSignatureService.wsdl`	
Schema	`Kein`	

⌫

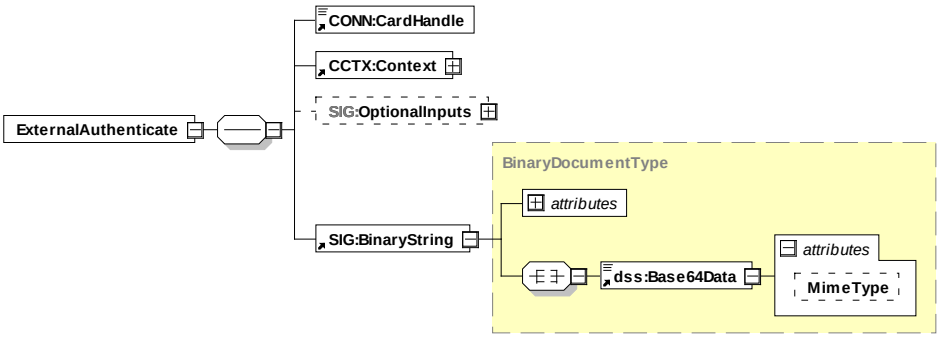
4.1.13.4.1 ExternalAuthenticate

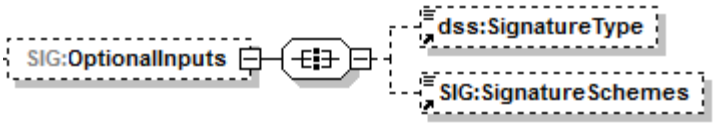
⌦ TIP1-A_5439 Operation ExternalAuthenticate

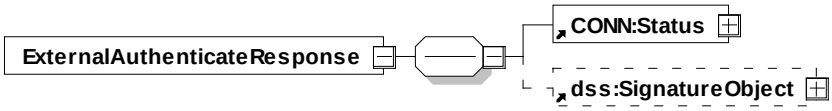
Der Authentifizierungsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation ExternalAuthenticate anbieten**.**

Tabelle 3: TAB_KON_781 Operation ExternalAuthenticate

Name	`ExternalAuthenticate`
Beschreibung	Diese Operation versieht einen Binärstring der maximalen Länge 512 Bit mit

<table>
<tr><td></td><td>einer nicht-qualifizierten elektronischen Signatur (nonQES).
Dazu wird das Signaturverfahren PKCS#1 verwendet. Das AUT-Zertifikat der SM-B und das AUT-Zertifikat des HBAx werden unterstützt.</td></tr>
<tr><td>Aufrufparameter</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 unterstützt HBAx und SM-B.</td></tr>
<tr><td><code>CCTX:Context</code></td><td>Aufrufkontext für HBAx:
MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend
Aufrufkontext für SM-B:
MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet</td></tr>
<tr><td><code>SIG:OptionalInputs</code></td><td>Enthält optionale Eingangsparameter:
 
</td></tr>
<tr><td><code>SIG:BinaryString</code></td><td>Dieses Element enthält im Kindelement <code>dss:Base64Data</code> den zu signierenden Binärstring.
Das XML Attribut SIG:BinaryString/dss:Base64Data/@MimeType MUSS den Wert "application/octet-stream" haben.
Die maximale Länge des Binärstrings beträgt 512 Bit entsprechend der maximal zu erwartenden Hash-Größe.
Aus der Länge des Binärstrings wird auf das verwendete Hashverfahren geschlossen. Es werden folgende Längen unterstützt:
• 256 Bit: SHA-256 (OID 2.16.840.1.101.3.4.2.1)
• 384 Bit: SHA-384 (OID 2.16.840.1.101.3.4.2.2)
• 512 Bit: SHA-512 (OID 2.16.840.1.101.3.4.2.3)
Im Falle des Signaturverfahrens RSASSA-PKCS1-v1_5 werden SHA-256, SHA-384 und SHA-512 unterstützt.
Im Falle des Signaturverfahrens RSASSA-PSS wird SHA-256 unterstützt.
Für die Signaturerstellung gilt:</td></tr>
</table>
</td></tr>
</table>

		• Im Falle des Signaturverfahrens RSASSA-PKCS1-v1_5 beginnt der Konnektor die Ausführung der Methode EMSA-PKCS1-v1_5-ENCODE nach [RFC3447], Abschnitt 9.2, mit Schritt 2, Erstellung des DigestInfo-Datenfeldes. • Im Falle des Signaturverfahrens RSASSA-PSS beginnt der Konnektor die Ausführung der Methode EMSA-PSS-ENCODE nach [RFC3447], Abschnitt 9.1.1, mit Schritt 3.
	`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element wird der Typ der zu erzeugenden Signatur bestimmt. Als Signaturtyp wird unterstützt : • **PKCS#1-Signatur** Durch Übergabe der URI urn:ietf:rfc:3447 wird eine PKCS#1 (Version 2.1) Signatur gemäß [RFC3447] erzeugt, die als `dss:Base64Signature` mit der oben genannten URI zurückgeliefert wird. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder Signaturvariante). Fehlt dieses Element, so wird ebenfalls der Signaturtyp PKCS#1-Signatur verwendet.
	`SIG:SignatureSchemes`	Durch dieses Element wird für PKCS#1-Signaturen zwischen den folgenden SignatureScheme-Optionen unterschieden: • RSASSA-PSS • RSASSA-PKCS1-v1_5 Fehlt dieses Element, so wird als Default-SignatureScheme RSASSA-PSS gewählt.
Rückgabe		
	`CONN:Status`	Enthält den Status der ausgeführten Operation.

	`dss:SignatureObject`	Enthält im Erfolgsfall die erzeugte Signatur in Form eines `dss:SignatureObject`-Elements gemäß [OASIS-DSS] (Abschnitt 3.2).
		Der Signaturwert wird im XML-Element `dss:SignatureObject/dss:Base64Signature` übergeben. Das XML-Attribut `dss:SignatureObject/dss:Base64Signature/@Type` kennzeichnet durch den Wert urn:ietf:rfc:3447 den Signatur-Typ.
		Die XML-Elemente
		`dss:SignatureObject/ds:Signature`
		`dss:SignatureObject/dss:Timestamp`
		`dss:SignatureObject/dss:SignaturePtr`
		`dss:SignatureObject/dss:Other`
		werden nicht verwendet.

Vorbedingungen	Keine
Nachbedingungen	Keine

Der Ablauf der Operation ExternalAuthenticate ist in Tabelle TAB_KON_782 beschrieben:

Tabelle 4: TAB_KON_782 Ablauf Operation ExternalAuthenticate

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	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 {$context.mandantId; $context.clientsystemId; $context.workplaceId; $context.userId; $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 { MandantId, CsId, CardHandle, UserId }

4.	TUC_KON_218 „Signiere“	Signaturberechnung durch Aufruf des TUC_KON_218 {
		PinRef = PIN.CH bzw. PIN.SMC;
		KeyRef = PrK.HP.AUT bzw. PrK.HCI.AUT;
		AlgorithmusID = signPKCS1_V1_5 oder signPSS;
		DTBS = Binärstring
		}

Tabelle 5: TAB_KON_783 Übersicht Fehler Operation ExternalAuthenticate

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4058	Security	Error	Aufruf nicht zulässig

Die folgende Tabelle führt die zulässigen privaten Schlüssel für die Operation ExternalAuthenticate auf:

Tabelle 6: TAB_KON_784 Privater Schlüssel je Karte für ExternalAuthenticate

Karte	Schlüssel
SM-B	PrK.HCI.AUT in DF.ESIGN
HBAx	PrK.HP.AUT in DF.ESIGN

⌫

4.1.13.5 Betriebsaspekte

Keine

Aus Kapitel 4.1.8 werden die Bestandteile des Authentifizierungsdienstes entfernt.

4.1.8 Signaturdienst

Der Signaturdienst bietet Clientsystemen und Fachmodulen eine Schnittstelle zum Signieren von Dokumenten und Prüfen von Dokumentensignaturen ~~und zum Signieren von Binärstrings zum Zweck der externen Authentisierung~~.

[…]

~~*4.1.8.1.4. Externe Authentisierung*~~

~~**TIP1-A_5437 Signaturverfahren für externe Authentisierung**~~

~~Der Signaturdienst MUSS das Signaturverfahren PKCS#1 entsprechend TAB_KON_780 – Signaturverfahren Externe Authentisierung unterstützen.~~

~~**Tabelle 7: TAB_KON_780 – Signaturverfahren Externe Authentisierung**~~

~~Signaturformat~~	~~Standard~~	~~Dokumentformate~~	~~QES/ nonQES~~	~~Bemerkung~~
~~PKCS#1 (V2.1)~~	~~[RFC3447]~~	~~Binär~~	~~nonQES~~	~~Dieses Signaturformat DARF NUR in Verbindung mit dem zur Authentisierung vorgesehenen Schlüssel des HBAx und des SM-B genutzt werden. Die Nutzung ist auf Dokumente (Hash) von maximal 512 bit Länge beschränkt.~~

~~**TIP1-A_5149 PKCS#1-Schnittstelle nur für Authentisierung mit HBAx und SM-B nutzen**~~

~~Der Hersteller des Konnektors MUSS den Anwender (Clientsystem) im Handbuch des Konnektors geeignet und ausreichend darüber informieren, dass das Signaturformat PKCS#1 nur zu Authentisierungszwecken mit dem Authentisierungsschlüssel des HBAx und des SM-B verwendet werden darf.~~

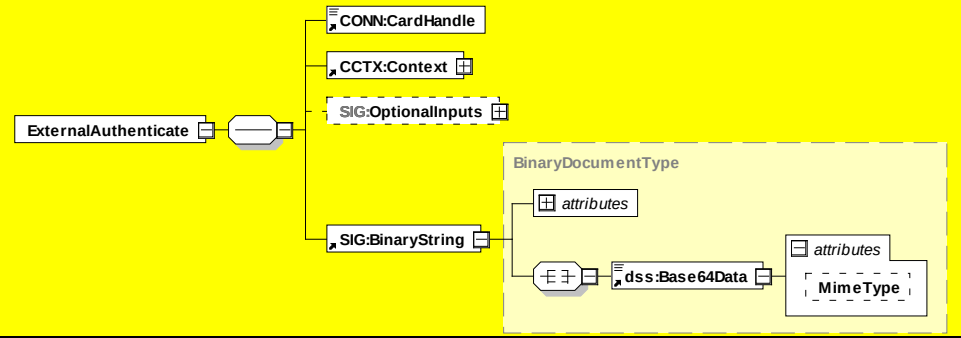
[…]

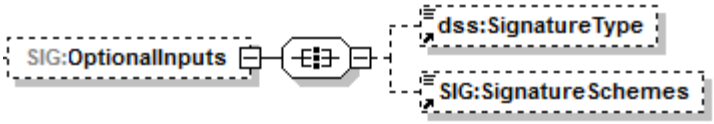
4.1.8.5.5 ExternalAuthenticate

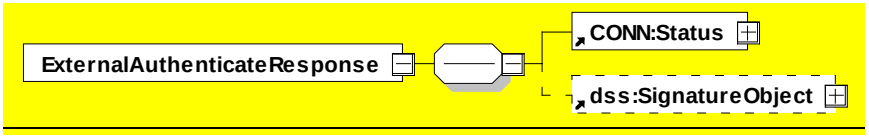
TIP1-A_5439 Operation ExternalAuthenticate

Der Authentifizierungsdienst des Konnektors MUSS an der Clientschnittstelle eine Operation ExternalAuthenticate anbieten.

Tabelle 8: TAB_KON_781 Operation ExternalAuthenticate

<table>
<tr><td>Name</td><td>ExternalAuthenticate</td></tr>
<tr><td>Beschreibung</td><td>Diese Operation versieht einen Binärstring der maximalen Länge 512 Bit mit einer nicht-qualifizierten elektronischen Signatur (nonQES).
Dazu wird das Signaturverfahren PKCS#1 verwendet. Das AUT-Zertifikat der SM-B und das AUT-Zertifikat des HBAx werden unterstützt.</td></tr>
<tr><td>Aufrufparameter</td><td>


<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>CONN:CardHandle</td><td>Identifiziert die zu verwendende Signaturkarte.
Die Operation unterstützt HBAx und SM-B.</td></tr>
<tr><td>CCTX:Context</td><td>Aufrufkontext für HBAx:
MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend
Aufrufkontext für SM-B:
MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet</td></tr>
<tr><td>SIG:OptionalInputs</td><td>Enthält optionale Eingangsparameter:
 
</td></tr>
<tr><td>SIG:BinaryString</td><td>Dieses Element enthält im Kindelement dss:Base64Data den zu signierenden Binärstring.
Das XML-Attribut SIG:BinaryString/dss:Base64Data/@MimeType MUSS den Wert "application/octet-stream" haben.
Die maximale Länge des Binärstrings beträgt 512 Bit entsprechend der maximal zu erwartenden Hash-Größe.
Aus der Länge des Binärstrings wird auf das verwendete Hashverfahren geschlossen. Es werden folgende Längen unterstützt:</td></tr>
</table>
</td></tr>
</table>

		• 256 Bit: SHA-256 (OID 2.16.840.1.101.3.4.2.1) • 384 Bit: SHA-384 (OID 2.16.840.1.101.3.4.2.2) • 512 Bit: SHA-512 (OID 2.16.840.1.101.3.4.2.3) Im Falle des Signaturverfahrens RSASSA-PKCS1-v1_5 werden SHA-256, SHA-384 und SHA-512 unterstützt. Im Falle des Signaturverfahrens RSASSA-PSS wird SHA-256 unterstützt. Für die Signaturerstellung gilt: • Im Falle des Signaturverfahrens RSASSA-PKCS1-v1_5 beginnt der Konnektor die Ausführung der Methode EMSA-PKCS1-v1_5-ENCODE nach [RFC3447], Abschnitt 9.2, mit Schritt 2, Erstellung des DigestInfo-Datenfeldes. • Im Falle des Signaturverfahrens RSASSA-PSS beginnt der Konnektor die Ausführung der Methode EMSA-PSS-ENCODE nach [RFC3447], Abschnitt 9.1.1, mit Schritt 3.
	`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element wird der Typ der zu erzeugenden Signatur bestimmt. Als Signaturtyp wird unterstützt : • **PKCS#1-Signatur** Durch Übergabe der URI urn:ietf:rfc:3447 wird eine PKCS#1 (Version 2.1) Signatur gemäß [RFC3447] erzeugt, die als `dss:Base64Signature` mit der oben genannten URI zurückgeliefert wird. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder Signaturvariante). Fehlt dieses Element, so wird ebenfalls der Signaturtyp PKCS#1-Signatur verwendet.
	`SIG:SignatureSchemes`	Durch dieses Element wird für PKCS#1-Signaturen zwischen den folgenden SignatureScheme-Optionen unterschieden: • RSASSA-PSS • RSASSA-PKCS1-v1_5 Fehlt dieses Element, so wird als Default-SignatureScheme RSASSA-PSS gewählt.
Rückgabe		
	`CONN:Status`	Enthält den Status der ausgeführten Operation.
	`dss:SignatureObject`	Enthält im Erfolgsfall die erzeugte Signatur in Form eines `dss:SignatureObject`-Elements gemäß [OASIS-DSS] (Abschnitt 3.2).

		Der Signaturwert wird im XML-Element `dss:SignatureObject/dss:Base64Signature` übergeben. Das XML-Attribut `dss:SignatureObject/dss:Base64Signature/@Type` kennzeichnet durch den Wert urn:ietf:rfc:3447 den Signatur-Typ.
		Die XML-Elemente
		`dss:SignatureObject/ds:Signature`
		`dss:SignatureObject/dss:Timestamp`
		`dss:SignatureObject/dss:SignaturePtr`
		`dss:SignatureObject/dss:Other`
		werden nicht verwendet.

Vorbedingungen	Keine	
Nachbedingungen	Keine	

Der Ablauf der Operation ExternalAuthenticate ist in Tabelle TAB_KON_782 beschrieben:

Tabelle 9: TAB_KON_782 Ablauf Operation ExternalAuthenticate

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
5.	checkArguments	Alle übergebenen Parameterwerte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.
6.	TUC_KON_000 „Prüfe Zugriffs-berechtigung“	Die Prüfung erfolgt durch den Aufruf TUC_KON_000 {$context.mandantId; $context.clientsystemId; $context.workplaceId; $context.userId; $cardHandle} Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
7.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 { MandantId, CsId, CardHandle, UserId }

8.	TUC_KON_218 „Signiere“	Signaturberechnung durch Aufruf des TUC_KON_218 {
		PinRef = PIN.CH bzw. PIN.SMC;
		KeyRef = PrK.HP.AUT bzw. PrK.HCI.AUT;
		AlgorithmusID = signPKCS1_V1_5 oder signPSS;
		DTBS = Binärstring
		}

Tabelle 10: TAB_KON_783 Übersicht Fehler Operation ExternalAuthenticate

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4058	Security	Error	Aufruf nicht zulässig

~~Die folgende Tabelle führt die zulässigen privaten Schlüssel für die Operation ExternalAuthenticate auf:~~

~~Tabelle 11: TAB_KON_784 Privater Schlüssel je Karte für ExternalAuthenticate~~

~~Karte~~	~~Schlüssel~~
~~SM-B~~	~~PrK.HCI.AUT in DF.ESIGN~~
~~HBAx~~	~~PrK.HP.AUT in DF.ESIGN~~

~~⌫~~

In Kapitel 3.3 wird Tabelle TAB_KON_504 korrigiert

Tabelle 12 TAB_KON_504 Ausführungserlaubnis für Dienste in kritischen Fehlerzuständen

	EC_Software_Integrity_Check_Failed	EC_Random_Generator_Not_Reliable	EC_Security_Log_Not_Writable	EC_Time_Sync_Pending_Critical	EC_Time_Difference_Intolerable	EC_CRL_Out_Of_Date	EC_TSL_Out_Of_Date_Beyond_Grace	EC_TSL_Trust_Anchor_Out_Of_Date	EC_Firewall_Not_Reliable	EC_Secure_KeyStore_Not_Available
Technische Use Cases (TUCs) der Basisdienste relevant für Fachanwendung und die Kommunikation mit Bestandsnetz und SIS										
Zugriffsberechtigungsdienst										
TUC_KON_000 PrüfeAufrufkontext	-	x	x	x	x	x	x	x	x	x
Dienstverzeichnisdienst										
TUC_KON_041 Einbringen der Endpunktinformationen während der Bootup-Phase	-	-	-	x	x	x	x	x	x	x
Kartenterminaldienst										
TUC_KON_051 Mit Anwender über Kartenterminal interagieren	-	-	-	-	-	x	x	x	-	-
Kartendienst										
TUC_KON_005 Card-to-Card authentisieren	-	-	-	-	-	x	x	x	-	-
TUC_KON_006 Datenzugriffsaudit eGK schreiben	-	-	-	-	-	x	x	x	-	-
TUC_KON_018 eGK-Sperrung prüfen	-	-	-	-	-	x	x	x	-	-
TUC_KON_024 Karte zurücksetzen	-	-	-	-	-	x	x	x	-	-
TUC_KON_026 Liefere CardSession	-	-	-	-	-	x	-	x	-	-
TUC_KON_200 SendeAPDU	-	-	-	-	-	x	x	x	-	-
TUC_KON_202 LeseDatei	-	-	-	-	-	x	x	x	-	-
TUC_KON_203 SchreibeDatei	-	-	-	-	-	x	x	x	-	-

	EC_Software_Integrity_Check_Failed	EC_Random_Generator_Not_Reliable	EC_Security_Log_Not_Writable	EC_Time_Sync_Pending_Critical	EC_Time_Difference_Intolerable	EC_CRL_Out_Of_Date	EC_TSL_Out_Of_Date_Beyond_Grace	EC_TSL_Trust_Anchor_Out_Of_Date	EC_Firewall_Not_Reliable	EC_Secure_KeyStore_Not_Available
TUC_KON_209 LeseRecord	-	-	-	-	-	x	x	x	-	-
Systeminformationsdienst										
TUC_KON_256 Systemereignis absetzen	-	x	x	x	x	x	x	x	x	x
Verschlüsselungsdienst										
TUC_KON_072 Daten symmetrisch verschlüsseln	-	-	-	x	x	x	x	x	-	-
TUC_KON_073 Daten symmetrisch entschlüsseln	-	-	-	x	x	x	x	x	-	-
Zertifikatsdienst										
TUC_KON_034 Zertifikatsinformationen extrahieren	-	-	-	x	x	x	x	x	-	-
Protokollierungsdienst										
TUC_KON_271 Schreibe Protokolleintrag	-	x	x	x	x	x	x	x	x	x
TLS-Dienst										
TUC_KON_110 Kartenbasierte TLS-Verbindung aufbauen	-	-	-	-	-	-	-	-	-	-
Verbindung zum VPN-Konzentrator										
TUC_KON_321 Verbindung zu dem VPN-Konzentrator der TI aufbauen	-	-	-	-	-	-	-	-	-	-
TUC_KON_322 Verbindung zum dem VPN-Konzentrator des SIS aufbauen	-	-	-	-	-	-	-	-	-	-
Operationen der Basisdienste										
Kartendienst										
VerifyPin	-	-	-	-	-	x	x	x	-	-
UnblockPin	-	-	-	-	-	x	x	x	-	-
ChangePin	-	-	-	-	-	x	x	x	-	-
GetPinStatus	-	-	-	-	-	x	x	x	-	-
Systeminformationsdienst										
Schnittstelle der Ereignissenke	-	x	x	x	x	x	x	x	x	x
GetCardTerminals	-	x	x	x	x	x	x	x	x	x
GetCards	-	x	x	x	x	x	x	x	x	x
GetResourceInformation	-	x	x	x	x	x	x	x	x	x
Subscribe	-	x	x	x	x	x	x	x	x	x
RenewSubscription	-	x	x	x	x	x	x	x	x	x
Unsubscribe	-	x	x	x	x	x	x	x	x	x
GetSubscription	-	x	x	x	x	x	x	x	x	x
Verschlüsselungsdienst										
EncryptDocument	-	-	-	-	-	x	x	x	-	-
DecryptDocument	-	-	-	-	-	x	x	x	-	-
Signaturdienst										

			EC_Software_Integrity_Check_Failed	EC_Random_Generator_Not_Reliable	EC_Security_Log_Not_Writable	EC_Time_Sync_Pending_Critical	EC_Time_Difference_Intolerable	EC_CRL_Out_Of_Date	EC_TSL_Out_Of_Date_Beyond_Grace	EC_TSL_Trust_Anchor_Out_Of_Date	EC_Firewall_Not_Reliable	EC_Secure_KeyStore_Not_Available
		SignDocument	-	-	-	-	-	x	x	x	-	-
		VerifyDocument	-	-	-	-	-	x	x	x	-	-
		GetJobNumber	-	-	-	-	-	x	x	x	-	-
		StopSignature	-	-	-	-	-	x	x	x	-	-
	Authentifizierungsdienst											
		ExternalAuthenticate	-	-	-	-	-	x	x	x	-	-
	Zertifikatsdienst											
		ReadCardCertificate	-	-	-	-	-	x	x	x	x	x
		CheckCertificateExpiration	-	-	-	-	-	x	x	x	x	x
		VerifyCertificate	-	-	-	-	-	x	x	x	x	x
	Zeitdienst											
		I_NTP_Time_Information	-	-	-	-	-	x	x	x	-	x
	Konnektormanagement											
		Softwareaktualisierung	x	x	x	x	x	x	x	x	x	x
		Protokolleinsicht	x	x	x	x	x	x	x	x	x	x
		Werksreset	x	x	x	x	x	x	x	x	x	x
		Sonstiges	-	x	x	x	x	x	x	x	x	x

In Kapitel 4.1.5.4.16 wird Tabelle TAB_KON_231 korrigiert

Tabelle 13: TAB_KON_231 – TUC_KON_218 „Signiere“

Element	Beschreibung
Name	TUC_KON_218 „Signiere“
Beschreibung	Dieser Use Case beschreibt das Anwenden eines privaten Schlüssels einer Karte zur Signatur oder Authentisierung.
Auslöser	• Aufruf einer der Operationen SignDocument ~~oder ExternalAuthenticate~~ des Signaturdienstes oder ExternalAuthenticate des Authentifizierungsdienstes durch das Clientsystem. • Aufruf durch Fachmodul

In Anhang H – Mapping von „Architektur der TI-Plattform“ auf Konnektorspezifikation wird Tabelle TAB_KON_712 korrigiert:

Tabelle 14 - TAB_KON_712 Architektur der TI-Plattform, Berechtigt Clientsysteme

Interface	Operation	→	Funktionsmerkmal	Interface / TUC	Operation
I_Crypt_Operations	decrypt_Document	→	Verschlüsselungsdienst	EncryptionService	DecryptDocument
	encrypt_Document	→	Verschlüsselungsdienst	EncryptionService	EncryptDocument
I_DNS_Name_Resolution	get_FQDN	→	Namensdienst und Dienstlokalisierung	GetFQDN	I_DNS_Name_Resolution
	get_IP_Address	→	Namensdienst und Dienstlokalisierung	GetIPAddress	
I_IP_Transport	send_Data_External	→	Anbindung LAN/WAN		AFOs: Routing der IP-Pakete von Client --> VPN_SIS
I_KV_Card_Handling	discard_Card_Usage_Reference	→	---		*--- keine Umsetzung notwendig. Erfolgt implizit*
	get_Card_Usage_Reference	→	---		*--- keine Umsetzung notwendig. Erfolgt implizit*
I_KV_Card_Unlocking					
	change_PIN	→	Kartendienst	CardService	ChangePin
	get_PIN_Status	→	Kartendienst	CardService	GetPinStatus
	initialize_PIN	→	Kartendienst	CardService	ChangePin
	unblock_PIN	→	Kartendienst	CardService	UnblockPin
	verify_PIN	→	Kartendienst	CardService	VerifyPin
I_Poll_System_Information	get_Ressource_Information	→	Systeminformationsdienst	EventService	GetResourceInformation
	get_Ressource_List	→	Systeminformationsdienst	EventService	GetCardTerminals
	get_Ressource_List	→	Systeminformationsdienst	EventService	GetCards

Interface	Operation	→	Funktionsmerkmal	Interface / TUC	Operation
I_Reg_Notification	register_for_Notifications	→	Systeminformationsdienst	EventService	Subscribe
		→	Systeminformationsdienst	EventService	Unsubscribe
		→	Systeminformationsdienst	EventService	GetSubscription
I_SAK_Operations	sign_Document_QES	→	Signaturdienst	SignatureService	SignDocument
	verify_Document_QES	→	Signaturdienst	SignatureService	VerifyDocument
I_Sign_Operations	sign_Document	→	Signaturdienst	SignatureService	SignDocument
	external_Authenticate	→	~~Signaturdienst~~ Authentifizierungsdienst	~~SignatureService~~ AuthSignatureService	ExternalAuthenticate
	verify_Document	→	Signaturdienst	SignatureService	VerifyDocument
	get_Certificate	→	Zertifikatsdienst	CertificateService	ReadCardCertificate
I_NTP_Time_Information	sync_Time	→	Zeitdienst		TIP1-A_2331 „I_NTP_Time_Information“
I_Directory_Query	search_Directory	→	LDAP-Proxy	LDAP-Operation	TIP1-A_5521

Änderungsbedarf:

Im Lastenheft wird gefordert, dass der Konnektor die von HBA-Vorläuferkarten qualifiziert signierten Dokumente validieren muss. Dies kann er zurzeit nicht leisten, da die mit HBA-Vorläuferkarten erstellten Signaturen nicht zulässige bzw. gar keine Zertifikatstyp-OIDs, im Folgenden als OID bzw. OIDs bezeichnet, für das EE-Zertifikat enthalten. Die Zulässigkeit der OIDs wird vom Konnektor geprüft und ergibt sich aus der Vorgabe einer Liste an erwarteten OIDs, die für HBA-Zertifikate in der gemSpec_OID spezifiziert sind. OIDs für Zertifikate von HBA-Vorläuferkarten sind dort nicht enthalten.

Die Einschränkung, dass die zu validierenden, qualifizierten elektronischen Signaturen (QES) die in gemSpec_OID aufgeführten OIDs enthalten müssen, soll entfallen, da diese Prüfung für QES-Zertifikate bereits gegen die Vertrauensliste der Bundesnetzagentur (BNetzA_VL) erfolgt.

Die momentan vorhandene Inkonsistenz in der Spezifikation wird beseitigt, indem die Liste der erlaubten OIDs (PolicyList) als Eingangsparameter für Signaturvalidierung und -Erstellung als optional definiert und bei der Validierung bzw Erstellung von QES mit HBAx nicht übergeben wird.

Folgende Konzepte und Spezifkationen werden dahingehend angepasst:

gemSpec_Kon

gemSpec_PKI

Änderungen in gemSpec_Kon

4.1.8.4.1 TUC_KON_160 “Dokumente nonQES signieren”

⌧ TIP1-A_4653 TUC_KON_160 „Dokumente nonQES signieren“

Der Konnektor MUSS den technischen Use Case TUC_KON_160 “Dokumente nonQES signieren” umsetzen.

Tabelle 180: TAB_KON_753 - TUC_KON_160 „Dokumente nonQES signieren“

Element	Beschreibung
Name	TUC_KON_160 ”Dokumente nonQES signieren”
Beschreibung	Im Rahmen von Fachanwendungen werden ein oder mehrere Dokumente mit einer nicht-qualifizierten elektronischen Signatur (nonQES) versehen. Es werden die nonQES_DocFormate unterstützt.
Auslöser	Aufruf durch ein Clientsystem (Operation SignDocument) oder ein Fachmodul.
Vorbedingungen	Die Signaturkarte muss gesteckt sein.
Eingangsdaten	• Zu signierendes Dokument (Document) bzw. zu signierende Dokumente • CardSession (SM-B~~, HBAx~~ oder bei Aufruf durch Fachmodul auch zusätzlich eGK)

Element	Beschreibung
	• WorkplaceId Weitere optionale Eingabeparameter (siehe Operation SignDocument, Parameter dss:OptionalInputs)

4.1.8.3.4 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 174: 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	• Zu signierende Dokumente • optionale Eingabeparameter zur Steuerung der Details der Signaturerstellung • CardSession Signaturkarte • zu verwendende Identität (Zertifikatsreferenz) • 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	• Prüfergebnis • Signaturzertifikat • ocspResponses - optional / nur wenn includeRevocationInfo = true (OCSPResponse 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 das Signaturzertifikat und – falls für mindestens ein Dokument benötigt – die Attributzertifikate von der Signaturkarte gelesen. 3. Das Signaturzertifikat und die Attributzertifikate werden durch Aufruf von TUC_KON_037 „Zertifikat prüfen“{C.HP.QES; required; true; ~~oid_hba_qes;~~ OCSP; getOCSPResponses =

Element	Beschreibung
	includeRevocationInfo ; Liste der Attributzertifikate} geprüft.
Varianten/Alternativen	Keine
Fehlerfälle	Keine
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

4.1.8.4.5 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 188: TAB_KON_591 - TUC_KON_151 „QES-Dokumentensignatur prüfen“

(…)

Standardablauf	1. **„DocumentValidation“**: Das signierte Dokument wird validiert (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 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 jüngste

	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 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. 5. Das Prüfergebnis (VerificationResult, OptionalOutput) wird an den Aufrufer zurückgegeben (siehe TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur).

(…)

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 223: TAB_KON_769 TUC_KON_037 „Zertifikat prüfen“

(…)

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: o PolicyList: Zugelassene Zertifikatstyp-OID gemäß [gemSpec_OID#GS-A_4445] o Vorgesehene KeyUsage (intendedKeyUsage) o Vorgesehene ExtendedKeyUsage

	(intendedExtendedKeyUsage) o 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 o 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 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)

(…)

Änderungen in gemSpec_PKI

⌦ **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 102: 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.
Anwendungsumfeld	System, das Zertifikate verwendet
Vorbedingungen	aktuelle TSL-Informationen im Truststore, 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 ~~• PolicyList: Liste der im aktuellen Aufruf zulässigen Zertifikatstyp-OIDs. Die Liste muss mindestens eine OID enthalten.~~ • 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)

(…)

Standardablauf	0. [System:] 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], [RFC*5755*] 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:] Prüfung, ob das 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 ist. 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 er¬mitteln") 5. [System:] Die abzufragenden Statusinformationen zu QES-Zertifikaten werden unter Verwendung der extrahierten OCSP-Adressen eingeholt. 6. [System:] Ermittlung der Rolle (TUC_PKI_009 "Rollenermittlung") ~~7.~~ ~~[System:] Prüfung, ob eine der übergebenen Zertifikatstyp-OIDs (aus der Parameter PolicyList) im Zertifikat enthalten ist (TUC_PKI_007 "Prüfung Zertifikatstyp"). Zur Prüfung muss die Liste (PolicyList s.o.) mindestens eine OID enthalten.~~ ~~8~~ 7. [System:] Ende des Use Case mit Rückgabe des/der im Zertifikat enthaltenen Rollen-OID(s)

Änderungsbedarf:

Änderungen in [gemSpec_Krypt]

[...]

⌦ **GS-A_4358 X.509-Identitäten für die Erstellung und Prüfung qualifizierter elektronischer Signaturen**

Alle Produkttypen, die X.509-Identitäten für die Erstellung oder Prüfung von qualifizierten elektronischen Signaturen verwenden, MÜSSEN mindestens alle in Tabelle Tab_KRYPT_003 aufgeführten Algorithmen unterstützen und die Tabellenvorgaben erfüllen.
TSP-X.509-QES, die qualifizierte Zertifikate (X.509-Identitäten) auf Basis der Schlüsselgeneration „ECDSA“ (vgl. Abschnitt 5.1) erstellen oder verwenden MÜSSEN die in Tab_KRYPT_003a aufgeführten Algorithmen und die Tabellenvorgaben erfüllen. ⌫

Tabelle 4: Tab_KRYPT_003 Algorithmen für X.509-Identitäten zur Erstellung qualifizierter elektronischer Signaturen für die Schlüsselgeneration „RSA“

Anwendungs~~fall~~fälle	Vorgabe	Schlüssellänge
Signatur des VDA-Zertifikats	Nachdem die eIDAS-Verordnung das Signaturgesetz vollständig abgelöst hat, steht es einem VDA frei zu entscheiden welche Signatur (bspw. signiert von einer beliebigen VDA-internen CA) sein VDA-Zertifikat haben soll. Insbesondere kann die Signatur mit einem Nicht-RSA-Verfahren erstellt werden. Eine auswertende Komponente muss mit beliebigen (also auch nicht-RSA basierten) Signaturen eines VDA-Zertifikats umgehen können (bspw. Signatur des VDA-Zertifikats nicht auswerten, Authentizität und Integrität des Zertifikats wird über die Vertrauensliste sichergestellt).	
Art und Kodierung des öffentlichen EE-Schlüssels	RSA Entweder OID 1.2.840.113549.1.1.1 (rsaEncryption) oder OID 1.2.840.113549.1.1.10 (id-RSASSA-PSS) [RFC-5756]. Die Auswahl obliegt dem EE-Zertifikatsausgebenden VDA.	2048 Bit, zulässig bis Ende 2022
Signatur eines Zertifikats, Signatur einer OCSP-Response oder Signatur eines OCSP-Responder-Zertifikates	sha256withRSAEncryption (OID 1.2.840.113549.1.1.11)	2048 Bit, zulässig bis Ende 20~~22~~18
	id-RSASSA-PSS (1.2.840.113549.1.1.10) [RFC-5756] Die Hashfunktion für die Hashwertberechnung der TBSCertificate-Datenstruktur MUSS eine nach [ALGCAT] zulässige Hashfunktion sein. Als Hashfunktion SOLL SHA-256 [FIPS-180-4] verwendet werden. Als MGF MUSS MGF1 [PKCS#1] verwendet werden. Die innerhalb der MGF1 verwendete Hashfunktion MUSS die gleiche Hashfunktion sein, wie die Hashfunktion der Hashwertberechnung der TBSCertificate-Datenstruktur.[1]	2048 Bit, zulässig bis Ende 2022

[1] Dies entspricht der Empfehlung aus [RFC-5756] bzw. [RFC-4055, 3.1] und dient der Komplexitätsreduktion.

Anwendungs~~fall~~fälle	Vorgabe	Schlüssellänge
	Die Saltlänge MUSS mindestens 256 Bit betragen.[2]	

Tabelle 5: Tab_KRYPT_003a Algorithmen für X.509-Identitäten zur Erstellung qualifizierter Signaturen für die Schlüsselgeneration „ECDSA“

Anwendungsfall	Vorgabe	Domainparameter / Schlüssellänge
Signatur des VDA-Zertifikats	Nachdem die eIDAS-Verordnung das Signaturgesetz vollständig abgelöst hat, steht es einem VDA frei zu entscheiden welche Signatur (bspw. signiert von einer beliebigen VDA-internen CA) sein VDA-Zertifikat haben soll. Insbesondere kann die Signatur mit einem Nicht-ECDSA-Verfahren erstellt werden. Eine auswertende Komponente muss mit beliebigen (also auch nicht-ECDSA basierten) Signaturen eines VDA-Zertifikats umgehen können (bspw. Signatur des VDA-Zertifikats nicht auswerten, Authentizität und Integrität des Zertifikats wird über die Vertrauensliste sichergestellt).	
Art und Kodierung des öffentlichen EE-Schlüssels	ecPublicKey {OID 1.2.840.10045.2.1} auf der Kurve brainpoolP256r1 [RFC-5639#3.4, brainpoolP256r1] Die Kodierung des öffentlichen Punkt erfolgt nach [RFC5480, Abschnitt 2], vgl. Beispiel auf S. 48f)	Kurve: brainpoolP256r1 Der privater Schlüssel muss zufällig und gleichverteilt aus {1, …, q} gewählt werden. (q ist die Ordnung des Basispunkts und $ceil(\log_2 q) = 256$). zulässig bis Ende 2023+
Signatur eines Zertifikats, Signatur einer OCSP-Response oder Signatur eines OCSP-Responder-Zertifikates	ecdsa-with-SHA256 [RFC-3279] {OID 1.2.840.10045.4.3.2} auf Kurve der brainpoolP256r1 [RFC-5639#3.4, brainpoolP256r1] vgl. Beispiel auf S. 48f	s. o.

[2] Die Maximallänge des Salts ergibt sich nach [PKCS#1] in Abhängigkeit von der Länge des Moduls.