

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

Entfernen der Prüfung von qualifizierten Zeitstempeln

Da auf absehbare Zeit in der TI keine Möglichkeit bestehen wird, qualifizierte Zeitstempel zu erstellen, wird eine Prüfung qualifizierter Zeitstempel durch den Konnektor aktuell nicht benötigt. Dabei wird auch die Inkonsistenz beseitigt, dass die Spezifikation nicht die Anforderung aus gemKPT_Arch_TIP umsetzt, da es sich um unterschiedliche Zeitstempel handelt.

Anpassungen in gemKPT_Arch_TIP:

TIP1-A_2255-01

[...]

~~In den SignedData enthaltene qualifizierte Zeitstempel werden ausgewertet.~~

Anpassungen in gemSpec_Kon:

4.1.8.1.3 Signaturzeitpunkt

Bezogen auf den vom Konnektor für die Signaturprüfung anzunehmenden Signaturerstellungszeitpunkt werden in dieser Spezifikation die Bezeichner Ermittelter_Signaturzeitpunkt und Benutzerdefinierter_Zeitpunkt verwendet.

Ermittelter_Signaturzeitpunkt: Vom Konnektor ermittelter Zeitpunkt, zu dem eine Signatur geprüft wird. Es werden folgende Signaturzeitpunkte ermittelt:

1. Ermittelter_Signaturzeitpunkt_Eingebettet:
in der Signatur eingebetteter Zeitpunkt (falls vorhanden)
- ~~2. Ermittelter_Signaturzeitpunkt_Qualifiziert:
qualifizierter Zeitstempel über die Signatur (falls vorhanden)~~
2. Ermittelter_Signaturzeitpunkt_System:
Systemzeit des Konnektors bei Signaturprüfung

Anmerkung: Bei vom Konnektor selbst erstellten Signaturen ist immer ein in der Signatur eingebetteter Zeitpunkt vorhanden, jedoch kein qualifizierter Zeitstempel, da in der TI keine qualifizierten Zeitstempel ausgestellt werden. Sollte ein Dokument mit einem qualifizierten Zeitstempel versehen sein, so wird dieser nicht für die Ermittlung des Signaturzeitpunkts herangezogen.

Benutzerdefinierter_Zeitpunkt: Vom Benutzer beim Aufruf der Signaturprüfoperation als Parameter an den Konnektor übergebener Zeitpunkt, zu dem eine Signatur geprüft werden soll.

4.1.8.4.6 TUC_KON_151 „QES Dokumentensignatur prüfen“

TIP1-A_4672_01 - TUC_KON_151 „QES-Dokumentensignatur prüfen“

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

Tabelle 216: TAB_KON_591 - TUC_KON_151 „QES-Dokumentensignatur prüfen“

Element	Beschreibung
Name	TUC_KON_151 „QES-Dokumentensignatur prüfen“
Beschreibung	Es wird die QES eines Dokuments geprüft. Dabei werden die Signaturverfahren laut Tabelle TAB_KON_582 – Signaturverfahren unterstützt. Sind mehrere Signaturen vorhanden, so werden alle geprüft. Auch Parallel- und Gegensignaturen MÜSSEN unterstützt werden.
Eingangsanforderung	keine
Auslöser	Aufruf durch ein Clientsystem (Operation VerifyDocument) oder durch ein Fachmodul im Konnektor
Vorbedingungen	keine
Eingangsdaten	<ul style="list-style-type: none"> signedDocument – <i>optional</i> (QES-signiertes Dokument vom Typ QES_DocFormate -> siehe Definition in Operation VerifyDocument mit SIG:Document) signatureObject – <i>optional</i> (-> siehe Definition in Operation VerifyDocument mit dss:SignatureObject. Es werden Parallel- und Gegensignaturen unterstützt.) optionalInputParams (optionale Eingabeparameter, siehe Operation VerifyDocument, Parameter SIG:OptionalInputs) certificates – <i>optional/falls diese nicht im signierten Dokument enthalten sind, sondern nur referenziert werden</i> (X.509-Zertifikate). xmlSchemas – <i>optional/nur für XML-Dokumente</i> (XMLSchema und ggf. weitere vom Hauptschema benutzte Schemata) includeRevocationInfo [Boolean]: – <i>optional; Default: false</i> (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur)
Komponenten	Konnektor
Ausgangsdaten	<ul style="list-style-type: none"> verificationResult [VerificationResult] (Ergebnis der Signaturprüfung) optionalOutput – <i>optional</i> (weitere Ausgabedaten gemäß SIG:OptionalOutput)

Standardablauf	<p>1. „DocumentValidation“: Das signierte Dokument wird validiert mit Aufruf TUC_KON_080 „Dokument validieren“{ ... }.</p> <p>Treten Fehler bei der Validierung der Typkonformität auf, wenn die Signatur im Dokument eingebettet ist, wird die Prüfung mit einem Fehler abgebrochen.</p> <p>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.)</p> <p>2. „CoreValidation“:</p> <p>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.</p> <p><u>XML-Signatur:</u> Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation.</p> <p><u>CMS-Signatur:</u> Die Core Validation erfolgt entsprechend Cryptographic Message Syntax (CMS) Kapitel 5.6 Signature Verification Process [RFC5652].</p> <p><u>PDF-Signatur:</u> Die Core Validation erfolgt entsprechend [PAdES-3] Kapitel 4.6 Signature Validation aus PAdES-BES Part 3.</p> <p>Auch wenn die Validierung fehlschlägt, werden die folgenden Prüfschritte durchgeführt, so dass ein vollständiges Prüfprotokoll erstellt werden kann.</p> <p>3. „CheckSignatureCertificate“:</p> <p>Teil 1: Signaturzertifikat ermitteln</p> <p>XML-Signatur: Das Signaturzertifikat ist im XMLDSig Element <code>ds:KeyInfo/ds:X509Data</code> gespeichert [XMLDSig] oder wird als Eingangsparameter übergeben.</p> <p>CMS-Signatur: Das Signaturzertifikat für CAdES ist im Feld <code>certificates</code> im <code>SignedData</code> Container gespeichert [CAdES] oder wird als Eingangsparameter übergeben.</p> <p>PDF-Signatur: Das PDF Signaturzertifikat für PAdES ist im Feld <code>SignedData.certificates</code> entsprechend Kapitel 6.1.1 „Placements of the signing certificate“ [PAdES Baseline Profile] gespeichert oder wird als Eingangsparameter übergeben.</p> <p>Teil 2: Signaturzeitpunkt bestimmen</p> <p>Der Signaturzeitpunkt Ermittelter_Signaturzeitpunkt_Eingebettet wird wie folgt selektiert: XML-Signatur:</p>
----------------	--

	<p>Das XML element <code>SigningTime</code> spezifiziert den Signaturzeitpunkt entsprechend Kapitel 7.2.1 XAdES [XAdES].</p> <p>CMS-Signatur: Das Attribut <code>SigningTime</code> spezifiziert den Signaturzeitpunkt entsprechend Kapitel 11.3 CMS [CMS].</p> <p>PDF-Signatur: Der Signaturzeitpunkt kann dem M Eintrag des Signature Dictionary entnommen werden [PAdES Baseline Profile] Kapitel 6.2.1 Signing time.</p> <p>Der Signaturzeitpunkt Ermittelter_Signaturzeitpunkt_Qualifiziert wird wie folgt selektiert: XML-Signatur: Das XML element <code>SignatureTimeStamp</code> spezifiziert den Signaturzeitpunkt entsprechend Kapitel 4.4.3.1 XAdES [XAdES]. CMS-Signatur und PDF-Signatur: Das Attribut <code>signature-time-stamp</code> spezifiziert den Signaturzeitpunkt entsprechend Kapitel 6.1 CAdES [CAdES]. Der Signaturzeitpunkt <code>Benutzerdefinierter_Zeitpunkt</code> liegt gegebenenfalls als Aufrufparameter vor. Der Signaturzeitpunkt <code>Ermittelter_Signaturzeitpunkt_System</code> wird ermittelt.</p> <p>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“ { certificate = C.HP.QES; qualifiedCheck = required; baseTime = Signaturzeitpunkt; offlineAllowNoCheck = true; validationMode = OCSP; ocspResponses = OCSP-Response; getOCSPResponses = includeRevocationInfo }. Sind OCSP-Responses in der Signatur eingebettet, ist die jüngsten OCSP-Response des EE-Zertifikats, die für die Zertifikatsprüfung notwendig ist, beim Aufruf von TUC_KON_037 zu übergeben. Sofern der Aufruf von TUC_KON_037 <code>ocspResponses</code> zurückgibt, wird die OCSP-Response des EE-Zertifikats in die Signatur eingebettet. Auch wenn die Zertifikatsprüfung fehlschlägt, werden die folgenden Prüfungen durchgeführt.</p> <p>4. „CheckPolicyConstraints“:</p> <p>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“ und Tabelle TAB_KON_778 „Einsatzbereich der Signaturvarianten“ zu erfüllen.</p>
--	---

	<p>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.</p> <p>5. Das Prüfergebnis (VerificationResult, OptionalOutput) wird an den Aufrufer zurückgegeben (siehe TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur).</p>
Varianten/Alternativen	Keine
Fehlerfälle	<p>Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_592 Fehlercodes TUC_KON_151 „QES Dokumentensignatur prüfen“ beschrieben.</p> <p>(->1) keine Signatur in signedDocument und signatureObject vorhanden: 4253.</p> <p>(->2 „CoreValidation“) Interner Fehler: 4001, Signatur des Dokuments ungültig: 4115, Signatur umfasst nicht das gesamte Dokument: 4262</p> <p>(->3 „CheckSignatureCertificate“) Interner Fehler: 4001, Signaturzertifikat ermitteln ist fehlgeschlagen: 4206.</p> <p>(->4 „CheckPolicyConstraints“) Interner Fehler: 4001, Dokument nicht konform zu Regeln für QES: 4124, Dokument nicht konform zu Profilierung der Signaturformate: 4208.</p>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	keine

[...]

Tabelle 218: TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur

VerificationResult für gesamtes Dokument (VerificationResult/HighLevelResult)	
Wert	Bedeutung
VALID	Wenn VerificationResult für alle Signaturen zum Dokument VALID
INVALID	Wenn VerificationResult für eine Signatur zum Dokument INVALID
INCONCLUSIVE	in allen anderen Fällen
VerificationResult pro Signatur (VerificationReport/IndividualReport/Result)	
Wert	Bedeutung mögliche Ausprägungen im VerificationReport
VALID	Die Signatur wurde gemäß den Regeln für die QES geprüft und für gültig befunden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor =

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

[<=]

TIP1-A_5540_01 - QES-Signaturprüfergebnis bezogen auf Signaturzeitpunkt

Der Konnektor MUSS zur QES-Signaturprüfung ein Prüfergebnis, das sich auf genau einen angenommenen Signaturzeitpunkt bezieht, an den Aufrufer zurückgeben.

Die Auswahl des angenommenen Signaturzeitpunkts, auf den sich das Signaturergebnis bezieht, erfolgt hierarchisch:

- Benutzerdefinierter_Zeitpunkt
falls vorhanden, sonst
- Ermittelter_Signaturzeitpunkt_Eingebettet
falls vorhanden, sonst
- ~~Ermittelter_Signaturzeitpunkt_Qualifiziert~~
~~falls vorhanden, sonst~~
- Ermittelter_Signaturzeitpunkt_System

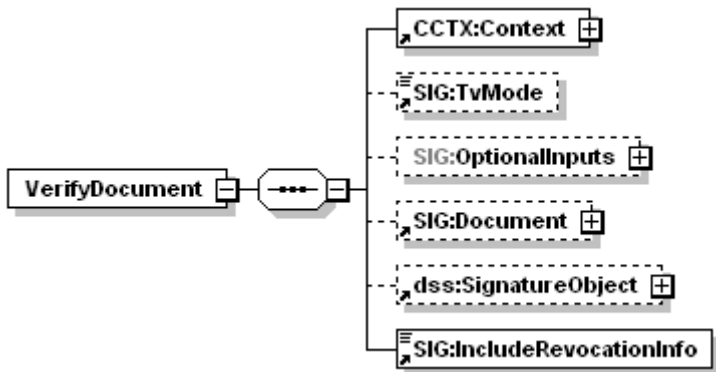
[<=]

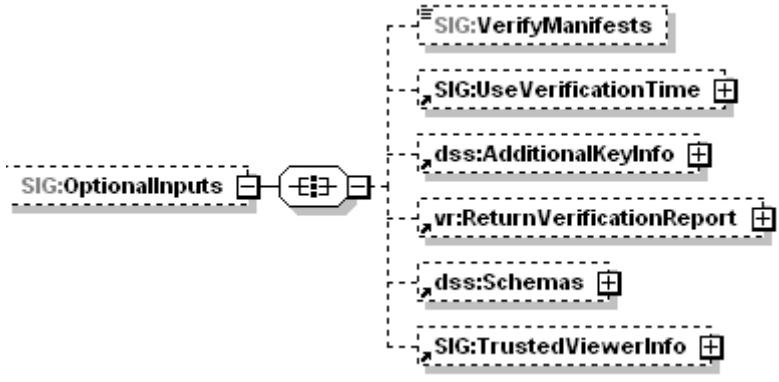
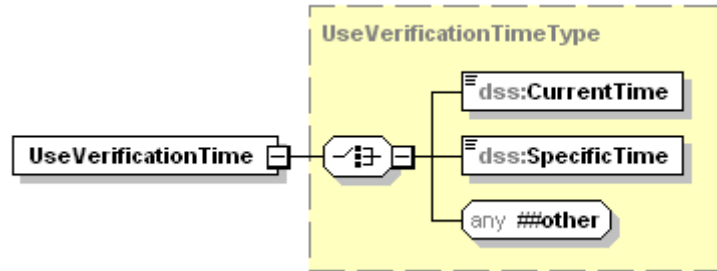
4.1.8.5.2 VerifyDocument (nonQES und QES)

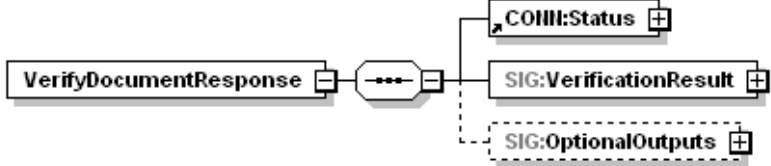
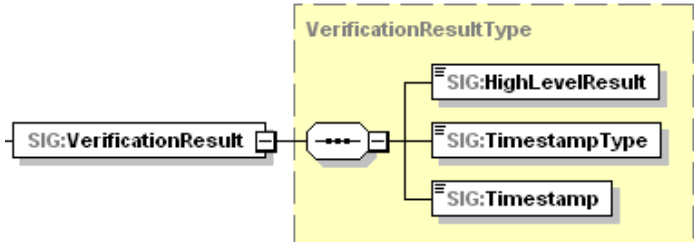
TIP1-A_5034-03 - Operation VerifyDocument (nonQES und QES)

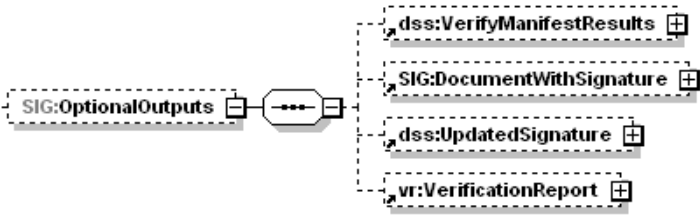
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation VerifyDocument (nonQES und QES) anbieten.

Tabelle 223: TAB_KON_066 Operation VerifyDocument (nonQES und QES)

Name	VerifyDocument	
Beschreibung	<p>Diese Operation verifiziert die Signatur eines Dokumentes.</p> <p>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</p> <p>Das Ergebnis der Prüfung wird, wenn gefordert, in Form eines standardisierten Prüfberichts in einer VerificationReport-Struktur gemäß [OASIS-VR] zurückgeliefert.</p>	
Aufruf- parameter		
	Name	Beschreibung
	CCTX: Context	MandantId, ClientSystemId, WorkplacId verpflichtend; UserId nicht ausgewertet
	TVMode	Der Parameter wird im Konnektor nicht ausgewertet.
	SIG: Optional Inputs	Enthält optionale Eingabeparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7): Die zulässigen optionalen Eingabeparameter sind unten erläutert.
	SIG: Document	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).
	dss: Signa ture Object	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

		<p>Type-Attribut (siehe <i>SignatureType</i>, Operationen <i>SignDocument</i> und <i>ExternalAuthenticate</i>) übergeben, wobei die nachfolgenden Werte unterstützt werden müssen:</p> <ul style="list-style-type: none"> • CMS-Signatur urn:ietf:rfc:5652 • S/MIME-Signatur urn:ietf:rfc:5751 • PDF-Signatur http://uri.etsi.org/02778/3 • PKCS#1-Signatur (siehe Operation <i>ExternalAuthenticate</i>) urn:ietf:rfc:3447 • ECC-Signatur (siehe Operation <i>ExternalAuthenticate</i>) urn:bsi:tr:03111:ecdsa
	<p>SIG: Include Revocat ionInfo</p>	<p>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. Für PDF-Signaturen erfolgt keine Einbettung von Sperrinformationen. Der Konnektor nimmt die Warnung 4261 in die Antwort auf.</p>
	<p>SIG: OptionalInputs</p>	
	<p>SIG: Verify Mani fests</p>	<p>Durch das in [OASIS-DSS] (Abschnitt 4.5.1) definierte Element kann die Prüfung eines ggf. vorhandenen Manifests angefordert werden.</p>
	<p>UseVerificationTime</p>	

	SIG: Use Verifi cation Time	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: Addit ional KeyInfo	Durch das in [OASIS-DSS] (Abschnitt 4.5.4) spezifizierte Element kann zusätzliches, für die Prüfung benötigtes, Schlüsselmaterial übergeben werden.
	vr: Return Verifi cation Report	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: Viewer Info	Der Parameter wird im Konnektor nicht ausgewertet.
Rückgabe		
	Status	Enthält den Ausführungsstatus der Operation.
	SIG: Verifi cation Result	 <p>Das Element Sig:VerificationResult enthält das Ergebnis der Prüfung als Ampel, den Typ des zugehörigen angenommenen Signaturzeitpunkts und der angenommene Signaturzeitpunkt selbst.</p>
	SIG: High Level Result	<p>Das Ergebnis der Prüfung (Ampelschaltung) mit folgenden Werten:</p> <ul style="list-style-type: none"> • VALID: alle Signaturen sind gültig • INVALID: mindestens eine der Signaturen ist ungültig • INCONCLUSIVE: in allen anderen Fällen

	SIG: Time stamp Type	<p>Der Typ des angenommenen Signaturzeitpunkts mit folgenden Werten:</p> <ul style="list-style-type: none"> • SIGNATURE_EMBEDDED_TIMESTAMP: in der Signatur eingebetteter 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 <p>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.</p>
	SIG: Time stamp	Im Element SIG:Timestamp wird der zu SIG:TimestampType gehörende Zeitstempel zurückgegeben.
	SIG: Optio nal Outputs	<p>Enthält (angelehnt an dss:OptionalOutputs, wie in Abschnitt 2.7 von [OASIS-DSS] beschrieben) optionale Ausgangselemente:</p> 
	dss: Verify Manifest Results	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: Document With Signa ture	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: Updated Signa ture	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: Verifi- cation Report	Dieses in [OASIS-VR] spezifizierte Element wird zurückgeliefert, falls das ReturnVerificationReport-Element als Eingabeparameter verwendet wurde. Die Profilierung von Anhang B3 MUSS beachtet werden.
Vorbe- dingungen	Keine	
Nachbe- dingungen	Keine	

Tabelle 224: TAB_KON_760 Ablauf Operation VerifyDocument (nonQES und QES)

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 Zugriffs- berechtigung“	Die Prüfung erfolgt durch den Aufruf <pre>TUC_KON_000 { mandantId = \$context.mandantId; clientsystemId = \$context.clientsystemId; workplaceId = \$context.workplaceId; needCardSession= false; }</pre> Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
3.	prüfe, ob QES oder nonQES	Ist im jeweiligen Signaturzertifikat mindestens ein QCStatement mit dem OID id-etsi-qcs-QcCompliance (0.4.0.1862.1.1) enthalten, handelt es sich um eine QES-Signatur, andernfalls liegt eine nonQES-Signatur vor.
Für QES-Signaturen wird Schritt 4 ausgeführt. Für nonQES-Signaturen wird Schritt 5 ausgeführt.		
4.a	Prüfe Signaturdienst-Modul	Prüfe, ob MGM_LU_SAK=Enabled. Ist dies nicht der Fall, so bricht die Operation mit Fehler 4125 ab.
4.b	TUC_KON_151 „QES Dokumentensignatur prüfen“	Die QES wird geprüft. Tritt hierbei ein Fehler auf, bricht die Operation ab.
5.	TUC_KON_161 „nonQES Dokumentensignatur prüfen“	Die nonQES wird geprüft. Tritt hierbei ein Fehler auf, bricht die Operation ab.

6.3 Profilierung VerificationReport

...

Sonderfälle:

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.~~