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

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.

Element	Beschreibung
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) • 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

Element	Beschreibung
	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 = 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. <u>XML-Signatur:</u> Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation. <u>CMS-Signatur:</u> 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

Element	Beschreibung
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) • 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. 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)