Elektronische Gesundheitskarte und Telematikinfrastruktur

Ergänzung zur Spezifikation Konnektor (PTV4)

Version:	1.1.0 ~~0~~ CC
Revision:	~~293621~~348696
Stand:	~~05.11.2020~~22.03.2021
Status:	Freigegeben für interne QS
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemSpec_Kon_KomfSig

Dokumentinformationen

Änderungen zur Vorversion

Es handelt sich um die Erstversion des Dokumentes.

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	~~05~~03.11.2020		freigegeben	gematik
1.1.0 CC	22.03.21		Einarbeitung Konn_Maintenance_21.2	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes

1.1 Zielsetzung

Das vorliegende Dokument ergänzt das Dokument [gemSpec_Kon_V5.9.0] um die Funktionalität "Komfortsignatur". Das Ziel ist, alle Anforderungen zur Herstellung, Test und Betrieb des Produkttyps "Konnektor PTV4Plus mit Komfortsignatur" bereitzustellen.

1.2 Zielgruppe

Das Dokument richtet sich an Konnektorhersteller sowie Hersteller und Anbieter von Produkttypen, die hierzu eine Schnittstelle besitzen.

1.3 Geltungsbereich

Dieses Dokument enthält normative Festlegungen zur Telematikinfrastruktur des deutschen Gesundheitswesens. Der Gültigkeitszeitraum der vorliegenden Version und deren Anwendung in Zulassungs- oder Abnahmeverfahren wird durch die gematik GmbH in gesonderten Dokumenten (z.B. Dokumentenlandkarte, Produkttypsteckbrief, Leistungsbeschreibung) festgelegt und bekanntgegeben.

Schutzrechts-/Patentrechtshinweis

Die nachfolgende Spezifikation ist von der gematik allein unter technischen Gesichtspunkten erstellt worden. Im Einzelfall kann nicht ausgeschlossen werden, dass die Implementierung der Spezifikation in technische Schutzrechte Dritter eingreift. Es ist allein Sache des Anbieters oder Herstellers, durch geeignete Maßnahmen dafür Sorge zu tragen, dass von ihm aufgrund der Spezifikation angebotene Produkte und/oder Leistungen nicht gegen Schutzrechte Dritter verstoßen und sich ggf. die erforderlichen Erlaubnisse/Lizenzen von den betroffenen Schutzrechtsinhabern einzuholen. Die gematik GmbH übernimmt insofern keinerlei Gewährleistungen.

1.4 Abgrenzungen

Spezifiziert werden in dem Dokument die von dem Produkttyp bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert.

Die vollständige Anforderungslage für den Produkttyp ergibt sich aus weiteren Konzept- und Spezifikationsdokumenten, diese sind in dem Produkttypsteckbrief des Produkttyps "Konnektor PTV4Plus mit Komfortsignatur" verzeichnet.

1.5 Methodik

1.5.1 Anforderungen

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

Sie werden im Dokument wie folgt dargestellt:

<AFO-ID> - <Titel der Afo>
Text / Beschreibung
[<=]

Dabei umfasst die Anforderung sämtliche innerhalb der Afo-ID und der Textmarke angeführten Inhalte.

1.5.2 Hinweise zur Benutzung dieses Ergänzungsdokuments

In diesem Dokument stehen nur die geänderten Passagen zur zugrunde liegenden Konnektor-Spezifikation. Alle anderen Teile gelten wie in [gemSpec_Kon_V5.9.0] beschrieben. Die Kapitelstruktur von [gemSpec_Kon_V5.9.0] wurde beibehalten, um dem Leser die Zuordnung der für die Komfortsignatur geänderten Anforderungen zu erleichtern. Bei Kapiteln ohne Änderungen ist nur die Überschrift genannt.

2 Systemüberblick

Die Komfortsignaturfunktion stellt einen Modus des Konnektors bereit, bei dem für die QES mit ein- und denselben HBA mehrere vom Clientsystem initiierte Signaturaufträge (Einzel- oder Stapelsignatur) abgearbeitet werden, ohne dass der Inhaber des HBA für jeden einzelnen dieser Signaturaufträge die PIN.QES am Kartenterminal eingegeben muss.

3 Übergreifende Festlegungen

3.1 Konnektoridentität und gSMC-K

3.2 Bootup-Phase

3.3 Betriebszustand (Kap 3.3)

[Hinweis: Die Anforderung TIP1-A_4510-03 wird nach der Anforderung TIP1A_4510 eingefügt und ersetzt TIP1-A_4510-02]

TIP1-A_4510-03 **-** Sicherheitskritische Fehlerzustände
Der Konnektor MUSS bei eingetretenem Fehlerzustand aus Tabelle Tab_ Kon_503 Betriebszustand_Fehlerzustandsliste mit Severity=Fatal dafür sorgen, dass von den Operationen der Basisdienste und Technische Use Cases (TUCs) der Basisdienste, die relevant für Fachanwendungen sind, nur erlaubte Operationen und TUCs gestartet und ausgeführt werden.
Welche Operationen und TUCs je eingetretenem Fehlerzustand ausgeführt werden dürfen, legt Tabelle „TAB_KON_504 Ausführungserlaubnis für Dienste in kritischen Fehlerzuständen" fest: Jede Erlaubnis ist dort durch ein „x" definiert.
Abweichend zu Angaben in der Tabelle TAB_KON_504 DÜRFEN folgende Operationen und TUCs NICHT im Zustand EC_Firewall_Not_Reliable ausgeführt werden:

- TUC_KON_000 PrüfeAufrufkontext
- TUC_KON_041 Einbringen der Endpunktinformationen während der Bootup-Phase
- GetCardTerminals
- GetCards
- GetResourceInformation
- Subscribe
- RenewSubscription
- Unsubscribe
- GetSubscription
- ReadCardCertificate
- CheckCertificateExpiration
- VerifyCertificate

Sind mehrere Fehlerzustände gleichzeitig eingetreten, dürfen nur die Operationen und TUCs ausgeführt werden, die für alle eingetretenen Fehlerzustände erlaubt sind. Der Konnektor muss Anfragen, die auf Grund eines kritischen Fehlerzustandes nicht ausgeführt oder abgebrochen werden, mit einem Fehler (Fehlercode 4002) beantworten.

Tabelle 1: TAB_KON_502 Fehlercodes „Betriebszustand"

Fehlercode	ErrorType	Severity	Fehlertext
4002	Security	Fatal	Der Konnektor befindet sich in einem kritischen Betriebszustand

[<=]

[Hinweis: In der Tabelle TAB_KON_504 werden für den Bereich "Operation der Basisdienste" im Signaturdienst drei Operationen ergänzt: ActivateComfortSignature, DeactivateComfortSignature, GetSignatureMode]

Tabelle 2: 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_Period	EC_TSL_Trust_Anchor_Out_Of_Date	EC_Secure_KeyStore_Not_Available	EC_FW_Not_Valid_Status_Blocked
Technische Use Cases (TUCs) der Basisdienste relevant für Fachanwendung und die Kommunikation mit Weiteren Anwendungen und SIS										
.....										
Operationen der Basisdienste										
....										
Signaturdienst										
SignDocument	-	-	-	-	-	x	x	x	-	x

VerifyDocument	-	-	-	-	-	x	x	x	-	x
GetJobNumber	-	-	-	-	-	x	x	x	-	x
StopSignature	-	-	-	-	-	x	x	x	-	x
ActivateComfortSignature	-	-	-	-	-	x	x	x	-	x
DeactivateComfortSignature	-	-	-	-	-	x	x	x	-	x
GetSignatureMode	-	-	-	-	-	x	x	x	-	x
.....										

241

4 Funktionsmerkmale

4.1 Anwendungskonnektor

4.1.1 Kartendienst

[Hinweis: In der Tabelle TAB_KON_531 bleiben die Einträge zu "CM_CARD_LIST" unberührt; im Abschnitt "CARD.CARDSESSION_ LIST" wird der Parameter "CARDSESSION. SIGNMODE" ergänzt.]

Der Kartendienst verwaltet mindestens die in der informativen Tabelle TAB_KON_531 ausgewiesenen Parameter, weitere herstellerspezifische Parameter sind möglich. Die normative Festlegung wann welche Parameter wie belegt werden, erfolgt in den folgenden Abschnitten und Unterkapiteln.

Tabelle 3: TAB_KON_531 Parameterübersicht des Kartendienstes

ReferenzID	Belegung	Zustandswerte
CM_CARD_LIST	Liste von Card-Objekten	Eine Liste von Repräsentanzen (CardObjects) der dem Konnektor bekannten Karten. Die Attribute der Card-Objekte sind im Folgenden gelistet.
CARD.CARDHANDLE		vom Konnektor vergebenen eindeutigen Identifikator (Handle).
CARD.CTID		Kartenterminal, in dem die Karte steckt
CARD.SLOTNO		Slot, in dem die Karte steckt
CARD.ICCSN		ICCSN der Karte (sofern auslesbar),
CARD.TYPE		Typ der Karte gemäß Tabelle TAB_KON_500 Wertetabelle Kartentypen
CARD.CARDVERSION		die Versionsinformationen zum Produkttyp der Karte und den gespeicherten Datenstrukturen gemäß [gemSpec_Karten_Fach_TIP].
CARD.CARDVERSION. COSVERSION		Produkttypversion des COS

CARD.CARDVERSION. OBJECTSYSTEMVERSION		Produkttypversion des Objektsystems
CARD.CARDVERSION. CARDPTPERSVERSION		Produkttypversion der Karte bei Personalisierung
CARD.CARDVERSION. DATASTRUCTUREVERSION		Version der Speicherstrukturen (aus EF.Version)
CARD.CARDVERSION. LOGGINGVERSION		Version der Befüllvorschrift für EF.Logging
CARD.CARDVERSION. ATRVERSION		Version der Befüllvorschrift für EF.ATR
CARD.CARDVERSION. GDOVERSION		Version der Befüllvorschrift für EF.GDO
CARD.CARDVERSION. KEYINFOVERSION		Version der Befüllvorschrift für KeyInfo
CARD.INSERTTIME	Timestamp	Zeitpunkt, an dem die Karte gesteckt wurde
CARD.CARDHOLDERNAME	String	Name des Karteninhabers bzw. der Institution/Organisation (subject.commonName)
CARD.KVNR	String	Versicherten-ID (unveränderbarer Teil der KVNR)
CARD. CERTEXPIRATIONDATE		Ablaufdatum des AUT-Zertifikats der Karte
CARD.CARDSESSION_ LIST	Liste von CardSession-Objekten	Eine Liste von Repräsentanzen (CardSession-Objects) der pro Karte vorhandenen Kartensitzungen. Die Attribute der CardSession-Objekte sind im Folgenden gelistet. Das Tripel aus MandantID, CSID und UserID bildet den Kontext ab, in welchem diese Kartensitzung initiiert wurde.
CARDSESSION. AUTHSTATE	Liste von Einträgen aus a)C2C:KeyRef, Role oder b) CHV: PINRef	Liste von erreichten Sicherheitszuständen. Jeder einzelne Sicherheitszustand kann entweder über C2C gegen KeyRef (mit einer bestimmten Rolle gemäß [gemSpec_PKI_TI#Tab_PKI_918]) oder Card Holder Verification (CHV) gegen eine referenzierte PIN erreicht worden sein. Für eGK G2.0 wird der Zustand der MRPINs nicht in AuthState gespeichert.
CARDSESSION. MANDANTID		Mandant-ID
CARDSESSION.CSID		Clientsystem-ID

CARDSESSION.USERID		Nutzer-ID
CARDSESSION.AUTHBY	Referenz auf CardSession	Kartensitzung, über die diese Karte freigeschaltet wurde (nur für eGK belegt)
CARDSESSION. SIGNMODE	„PIN" oder „Comfort"	Signaturmodus „PIN": Komfortsignaturmodus ist für die Karte ausgeschaltet "Comfort": Komfortsignaturmodus ist eingeschaltet Default-Wert="PIN" Nur relevant für den HBA

256

257 **4.1.1.1 Funktionsmerkmalweite Aspekte**

258

259 TIP1-A_4561-02 **-** Terminal-Anzeigen für PIN-Operationen
260 Der Konnektor MUSS im Rahmen des interaktiven PIN-Handlings die folgenden
261 Displaymessages für die Anzeige im Kartenterminal verwenden:

262 **Tabelle 4: TAB_KON_090 Terminalanzeigen beim Eingeben der PIN am Kartenterminal**

Karte/ Kontext	PIN-Referenz	I/ O	Terminal-Anzeige	ANW (max.Anz. Zeichen)
eGK /PIN-Eingabe für Vertreter-PIN	PIN.AMTS_REP	I	Vertreter-PIN•*0x0B*für•*0x0B*ANW *0x0F*Vertr-PIN:	22
eGK /PIN-Eingabe für Vertreter-PIN ändern	PIN.AMTS_REP	I	Vertreter-PIN•*0x0B*ändern *0x0F*PIN.eGK:	
eGK /PIN-Eingabe für Vertreter-PIN entsperren	PIN.AMTS_REP	I	Vertreter-PIN•*0x0*entsperren *0x0F*PIN.eGK:	
eGK /PIN-Eingabe für PIN-Schutz einschalten	MRPIN.NFD, MRPIN.DPE, MRPIN.AMTS, MRPIN.GDD	I	PIN-Schutz•*0x0B*ANW•*0x0B*einschalten *0x0F*PIN.eGK:	16
eGK /PIN-Eingabe für PIN-Schutz abschalten	MRPIN.NFD, MRPIN.DPE, MRPIN.AMTS, MRPIN.GDD	I	PIN-Schutz•*0x0B*ANW•*0x0B*abschalten *0x0F*PIN.eGK:	16

eGK /Sonstige	ALLE (außer PIN.AMTS_REP)	I	PIN•***0x0B***für•***0x0B***ANW ***0x0F***PIN.eGK:	32
HBAx	PIN.CH	I	Eingabe•***0x0B***Freigabe-PIN•***0x0B***HBA ***0x0F***PIN.HBA:	
	PIN.QES (Signatur auslösen)	I	#*UVW-XYZ*•***0x0B***Eingabe•***0x0B***Signatur-PIN•***0x0B***HBA ***0x0F***PIN.QES:	
HBA	PIN.QES (Komfortsignatur aktivieren)	I	Komfortsignatur•***0x0B***aktivieren•***0x0B***HBA ***0x0F***PIN.QES:	
SMC-B	PIN.SMC	I	Eingabe•***0x0B***PIN•SMC-B•***0x0B***SLOT:*X* ***0x0F***PIN.SMC:	
ANDERE	BELIEBIG	I	Herstellerspezifisch	
Erfolgreiche PIN-Eingabe	ALLE	O	PIN•***0x0B***erfolgreich•***0x0B***verifiziert!	
Fehlerhafte PIN-Eingabe	ALLE	O	PIN•***0x0B***falsch•***0x0B***oder•***0x0B***gesperrt!	
PUK-Eingabe	eGK PUK.CH	I	Eingabe•***0x0B***Versicherten-***0x0B***PUK ***0x0F***PUK.eGK:	
	HBAx PUK.CH	I	Eingabe•***0x0B***Freigabe-PUK•***0x0B***HBA ***0x0F***PUK.HBA:	
	HBAx PUK.QES	I	Eingabe•***0x0B***Signatur-PUK•***0x0B***HBA ***0x0F***PUK.QES:	
	SMC-B PUK.SMC	I	Eingabe•***0x0B***PUK•SMC-B•***0x0B***SLOT:*X* ***0x0F***PUK.SMC:	
Erfolgreiche PUK-Eingabe	ALLE	O	PIN•***0x0B***erfolgreich•***0x0B***entsperrt!	
Fehlerhafte PUK-Eingabe	ALLE	O	PUK•***0x0B***falsch•***0x0B***oder•***0x0B***gesperrt!	
Eingabe einer neuen PIN	eGK ALLE (außer PIN.AMTS_REP)	I	Eingabe• ***0x0B***neue•***0x0B***Versicherten-***0x0B***PIN• ***0x0B***(6-8•Ziffern) ***0x0F***PIN.eGK:	
	eGK PIN.AMTS_REP	I	Eingabe• ***0x0B***neue•***0x0B***Vertreter-PIN• ***0x0B***(6-8•Ziffern) ***0x0F***Vertr-PIN:	
	HBAx PIN.CH	I	Eingabe•***0x0B***neue•***0x0B***Freigabe-PIN•***0x0B***HBA•***0x0B***(6-8•Ziffern) ***0x0F***PIN.HBA:	
	HBAx PIN.QES	I	Eingabe• ***0x0B***neue•***0x0B***Signatur-PIN•***0x0B***HBA•***0x0B***(6-8•Ziffern) ***0x0F***PIN.QES:	

	SMC-B PIN.SMC	I	Eingabe•***0x0B***neue•***0x0B***PIN•SMC-B•***0x0B***SLOT:*X*•***0x0B***(6-8•Ziffern)***0x0F***PIN.SMC:
Eingabe einer Transport-PIN	eGK PIN.CH	I	Eingabe•***0x0B***Transport-***0x0B***Versicherten-***0x0B***PIN***0x0F***T-PIN.eGK:
	HBAx PIN.CH	I	Eingabe•***0x0B***Transport-***0x0B***PIN•***0x0B***HBA***0x0F***T-PIN.HBA:
	HBAx PIN.QES	I	Eingabe•***0x0B***Transport-***0x0B***PIN•***0x0B***HBA***0x0F***T-PIN.QES:
	SMC-B PIN.SMC	I	Eingabe•***0x0B***Transport-***0x0B***PIN•SMC-B•***0x0B***SLOT:*X* ***0x0F***T-PIN.SMC:
Wieder-holung einer neuen PIN	eGK PIN.CH	I	Eingabe•***0x0B***Versicherten-***0x0B***PIN•***0x0B***wiederholen!***0x0F***PIN.eGK:
	eGK PIN.AMTS_REP	I	Eingabe•***0x0B***neue•***0x0B***Vertreter-PIN•***0x0B*** wiederholen!***0x0F***Vertr-PIN:
	HBAx PIN.CH	I	Eingabe•***0x0B***für•HBA•***0x0B***wiederholen!***0x0F***PIN.HBA:
	HBAx PIN.QES	I	Eingabe•***0x0B***für•HBA•***0x0B***wiederholen!***0x0F***PIN.QES:
	SMC-B PIN.SMC	I	Eingabe•***0x0B***PIN.SMC•***0x0B***SLOT:*X*•***0x0B***wiederholen!***0x0F***PIN.SMC:
Ungleichheit bei der Wieder-holung der Eingabe der neuen PIN	ALLE	O	PINs•***0x0B***nicht•***0x0B***identisch!•***0x0B***Abbruch!
Erfolgreiche PIN-Änderung	ALLE	O	PIN•0x0Berfolgreich•0x0Bgeändert!
Anzeigen am lokalen Terminal beim Remote-PIN-Verfahren für das Ergebnis der Verschlüsselung durch die gSMC-KT			
Erfolgreiche Verschlüsselung	ALLE	O	Eingabe•0x0Bwird•0x0Bbearbeitet.
Fehler bei der Verschlüsselung	ALLE	O	Eingabe•0x0Bfehlgeschlagen.

[<=]

4.1.2 Dokumentvalidierungsdienst

4.1.3 Dienstverzeichnisdienst

4.1.4 Kartenterminaldienst

4.1.5 Kartendienst

4.1.6 Systeminformationsdienst

4.1.7 Verschlüsselungsdienst

4.1.8 Signaturdienst (Kap 4.1.8)

4.1.8.1 Funktionsmerkmalweite Aspekte

4.1.8.1.1 Dokumentensignatur

4.1.8.1.2 Signaturrichtlinien

4.1.8.1.3 Signaturzeitpunkt

4.1.8.1.4 Jobnummer

4.1.8.1.5 Komfortsignatur (Kap. 4.1.8.1.5 - neu)

Für die QES unterstützt der Konnektor die Komfortsignaturfunktion. In diesem Modus können für ein- und denselben HBA mehrere vom Clientsystem initiierte Signaturaufträge (Einzel- oder Stapelsignatur) abgearbeitet werden, ohne dass der Inhaber des HBA für jeden einzelnen dieser Signaturaufträge die PIN.QES am Kartenterminal eingegeben muss.

Im Auslieferungszustand ist die Komfortsignaturfunktion ausgeschaltet (`SAK_COMFORT_SIGNATURE = Disabled`), d. h. mit dem Konnektor können zunächst keine Komfortsignaturen durchgeführt werden. Die Komfortsignaturfunktion kann vom Administrator eingeschaltet werden. Dies ist nur möglich, wenn an der Clientsystemschnittstelle des Konnektors verpflichtend TLS mit Clientauthentisierung (Konfigurationsvariante SOAP1 und SOAP2 in TAB_KON_852) konfiguriert ist. Das Einschalten der Komfortsignaturfunktion im Konnektor hat zur Folge, dass alle Operationen an der Clientsystemschnittstelle nur über TLS mit Clientauthentisierung angesprochen werden können (außer ggf. Dienstverzeichnisdienst).

Bei eingeschalteter Komfortsignaturfunktion können potentiell alle HBAs in der Umgebung, in der der Konnektor eingesetzt ist, Komfortsignaturen durchführen. Die

eigentliche Aktivierung der Komfortsignatur muss separat für jeden einzelnen HBA erfolgen.

Durch Aufruf der Operation ActivateComfortSignature des Konnektors durch das Primärsystem wird die Nutzung der Komfortsignatur für einen HBA (Komfortsignaturmodus) aktiviert. Dazu muss der HBA-Inhaber die PIN.QES eingeben.

Der Konnektor merkt sich für die Cardsession des HBA, dass die Komfortsignatur aktiviert wurde. Bei den folgenden Aufrufen von ~~SignDocument~~~~werden~~SignDocument werden dann Komfortsignaturen ausgeführt, solange bis eines der folgenden Abbruchkriterien eintritt:

- Die vom HBA (entsprechend Personalisierung) oder die vom Konnektor (entsprechend Konfiguration SAK_COMFORT_SIGNATURE_MAX) durchgesetzte maximale Anzahl von Signaturen wurde erreicht.
- Das konfigurierte Zeitintervall für die Komfortsignatur (entsprechend Konfiguration SAK_COMFORT_SIGNATURE_TIMER) ist für die Cardsession abgelaufen - (aktuell ggf. laufende Signaturaufträge / -Stapel werden beendet).
- Der Komfortsignaturmodus wurde für die betroffene Cardsession deaktiviert.
- Der HBA wurde gezogen.
- Der Sicherheitszustand des HBA wurde zurückgesetzt.
- Die Komfortsignaturfunktion wurde für den Konnektor durch den Administrator deaktiviert.

A_19945 - Unterstützte Signaturvarianten bei Komfortsignatur
Der Signaturdienst MUSS bei der Komfortsignatur die Signaturvarianten für die QES gemäß TAB_KON_778 unterstützen.[<=]

A_18597 - Sicherheitszustand der PIN.QES bei Komfortsignatur
Bei der Komfortsignatur DARF der Konnektor den Sicherheitszustand der PIN.QES NICHT selbsttätig zurücksetzen, außer wenn dies explizit spezifikatorisch gefordert wird.[<=]

A_18597 kann z. B. umgesetzt werden, indem

- ein dedizierter logischer Kanal des HBA für die Komfortsignatur verwendet wird und
- im dedizierten logischen Kanal des HBA die Selektion von DF.QES solange beibehalten wird, bis ein Verlassen von DF.QES durch die Spezifikation explizit gefordert wird.

A_18686-01~~A_18686~~ - Komfortsignatur-Timer
Der Konnektor MUSS für jede HBA-Kartensitzung mit eingeschalteter Komfortsignatur einen Komfortsignatur-Timer gemäß konfiguriertem Zeitintervall SAK_COMFORT_SIGNATURE_TIMER einrichten ~~und~~.
Der Konnektor DARF nach Erreichen des Maximalwerts des Timers NICHT weitere Signaturaufträge annehmen.
Der Konnektor MUSS den Sicherheitszustand des HBA nach Erreichen des Maximalwertes des Timers bzw. nach Abarbeitung eines ggf. laufenden Signaturauftrages zurücksetzen.
[<=]

A_19100 - Komfortsignatur-Zähler

Kommentiert [DS1]: C_10629

Kommentiert [DS2]: C_10629

Der Konnektor MUSS für jeden gesteckten HBA mit eingeschalteter Komfortsignatur die an die Karte gesendeten Signaturaufträge zählen und nach Erreichen des Maximalwerts den Sicherheitszustand des HBA zurücksetzen.**[<=]**

A_19258 **-** Secure Messaging bei Komfortsignatur
Bei der Komfortsignatur MUSS der Signaturdienst die zu signierenden Daten (DTBS) über Secure Messaging vom Konnektor zum HBA übertragen. Dieser Secure Messaging-Kanal MUSS über die gSMC-K zum HBA mittels C.SAK.AUTD_CVC aufgebaut werden.**[<=]**

A_20073-01~~**A_20073**~~ **-** Prüfung der Länge der UserId
Der Konnektor MUSS die beim Aktivieren des Komfortsignaturmodus vom PS übermittelte UserId für die Kartensitzung des HBA, für den der Modus aktiviert wird, auf die ausreichende Länge von 128 Bit im Format einer UUID nach RFC4122 prüfen und die Aktivierung mit Fehler 4272 ablehnen, wenn die UserId nicht ausreichend lang ist.**[<=]**

Kommentiert [DS3]: C_10623

A_20074 **-** UserId über 1.000 Vorgänge eindeutig
Der Konnektor MUSS die Eindeutigkeit der UserId sicherstellen. Wird die Operation ActivateComfortSignature mit einer UserId im Aufrufkontext aufgerufen, die innerhalb der vorangegangenen 1.000 Vorgänge bereits verwendet wurde, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4270 abbrechen. Die Zählung der Aufrufe erfolgt dabei unabhängig vom Aufrufkontext.**[<=]**

A_19101 **-** Handbuch-Hinweis zu Nutzerauthentisierung am Clientsystem bei Komfortsignatur
Das Handbuch des Konnektors MUSS einen Hinweis enthalten, dass die Authentifizierung des HBA-Inhabers für die Komfortsignatur vom Clientsystem vorgenommen wird und dass die Authentifizierung des Nutzers am Clientsystem einen unverzichtbaren Beitrag zur Sicherheit der Lösung leistet.**[<=]**

4.1.8.2 Durch Ereignisse ausgelöste Reaktionen

keine

4.1.8.3 Interne TUCs, nicht durch Fachmodule nutzbar

Abbildung PIC_KON_102 Use Case Diagramm Signaturdienst (Komfortsignatur) beschreibt die Aufrufbeziehungen der TUCs des Signaturdienstes für die Komfortsignatur.

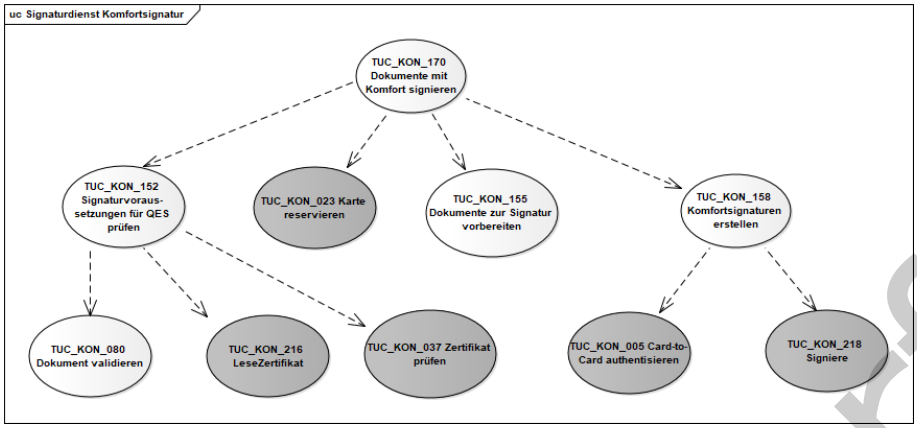


Abbildung 1: PIC_KON_102 Use Case Diagramm Signaturdienst (Komfortsignatur)

4.1.8.3.1 TUC_KON_158 "Komfortsignaturen erstellen" (Kap 4.1.8.3.7 - neu)

Der TUC_KON_158 führt die Komfortsignatur für ein Dokument oder mehrere Dokumente eines Stapels aus. Da die Komfortsignatur auf der Zielkarte passende CVC voraussetzt, die auf den HBA-Vorläuferkarten nicht vorhanden sind, unterstützt dieser TUC nur den HBA.

A_19102-02~~A_19102-01~~ - TUC_KON_158 „Komfortsignaturen erstellen“
Der Konnektor MUSS den technischen Use Case TUC_KON_158 „Komfortsignaturen erstellen” umsetzen.

Tabelle 5: TAB_KON_870 – TUC_KON_158 „Komfortsignaturen erstellen"

Element	Beschreibung
Name	TUC_KON_158 „Komfortsignaturen erstellen“
Beschreibung	Die Data To Be Signed (DTBS) werden erzeugt, an die Signaturkarte gesendet und dort signiert. Die Übertragung der DTBS erfolgt mit Secure Messaging. Die Abarbeitung der Signatur erfolgt im SE#2.
Auslöser	TUC_KON_170 „Dokumente mit Komfort signieren“
Vorbedingungen	Die Ressource Signaturkarte ist für den Vorgang reserviert. DF.QES ist selektiert. PIN.QES ist initial verifiziert

Eingangsdaten	• Zu signierendes Dokument bzw. zu signierende Dokumente • cardSession (nur HBA erlaubt) • zu verwendende Identität (Zertifikatsreferenz) • crypt [SIG_CRYPT_QES] - *optional*; *default und* Werte*bereich*: siehe TAB_KON_862-01 (Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden.) • WorkplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBA)
Ausgangsdaten	• Signierte Dokumente

Standardablauf	1. Wenn noch nicht erfolgt, wird basierend auf SAK.AUTD_CVC und HPC.AUTD_SUK_CVC und den zugehörigen privaten Schlüsseln ein sicherer Kanal zwischen der gSMC-K des Konnektors und dem HBA aufgebaut mittels Aufruf TUC_KON_005 „Card-to-Card authentisieren“ {
	sourceCardSession = gSMC-K;
	targetCardSession = CardSession;
	authMode = „gegenseitig+TC“}
	
	Die folgenden Schritte werden für jedes Dokument des Stapels durchgeführt.
	2. Das zu signierende Dokument wird, soweit noch nicht erfolgt, für die Signatur gemäß des entsprechenden Formats vorbereitet. Ein Ergebnis dieser Vorbereitung sind die Data To Be Signed (DTBS): der Hash-Wert (Digest des SignedInfo-Elementes), der zur Signatur an die Karte gesendet werden soll.
	3. Es wird geprüft, ob der Komfortsignatur-Zähler der cardSession den Wert SAK_COMFORT_SIGNATURE_MAX überschritten hat .
	~~4. Es wird geprüft, ob der Komfortsignatur-Timer der cardSession (SAK_COMFORT_SIGNATURE_TIMER) abgelaufen ist.~~
	~~5~~
	4. Für das zu signierende Dokument werden die DTBS zur Signatur im sicheren Kanal an den HBA übermittelt (Aufruf TUC_KON_218 „Signiere“). Dabei werden der Schlüssel und der Algorithmusidentifier über die Tabelle TAB_KON_900 bestimmt.
	~~6~~5. Der Komfortsignatur-Zähler der cardSession wird um 1 erhöht.
	~~7~~6. Die erstellte Signatur wird mathematisch geprüft.
	~~8~~7. Der ermittelte Signaturwert wird in den zuvor vorbereiteten Signaturprototypen eingefügt.

	~~98~~. Der Konnektor löst TUC_KON_256 {"SIG/SIGNDOC/NEXT_SUCCESSFUL"; Op; Info; „$Jobnummer"; noLog; $doInformClients } aus.
Varianten/ Alternatien	Keine
Fehlerfälle	In den Fehlerfällen, die zum Abbruch des Komfortsignaturmodus mit Fehlercode 4271 führen, wird vor dem Abbruch TUC_KON_172 für das cardHandle des HBA ausgeführt. (->3) Der Komfortsignatur-Zähler der cardSession hat den Maximalwert überschritten: Fehlercode 4271 ~~(->4) Der Komfortsignatur-Timer der cardSession ist abgelaufen: Fehlercode 4271~~ ~~(->5~~ (->4) Der PIN.QES-Nutzungszähler der Karte ist abgelaufen (erkennbar z. B. daran, dass die Karte einen Autorisierungsfehler zurückmeldet): Fehlercode 4271 (->~~5~~4) Fehler im Signaturvorgang führen zum Abbruch des gesamten Signaturvorgangs: Fehlercode 4123 (->~~7~~6) Fehler in mathematischer Prüfung der Signatur führen zum Abbruch des Signaturvorgangs: Fehlercode 4120 Das weitere Verhalten des TUCs bei einem Fehlerfall oder beim Abbruch durch den Benutzer ist in TAB_KON_192, Verhalten des Konnektors beim Abbruch einer Stapelsignatur, beschrieben.
Sicherheitsanforderungen	Zum Aufbau des sicheren Kanals bzw. zur Aushandlung des symmetrischen Schlüssels DARF DF.QES NICHT verlassen werden. Benötigte CVCs des HBA MÜSSEN also bereits vor dem Signaturvorgang eingelesen und gecached werden. Dies KANN bereits beim Stecken des HBA geschehen. Komfortsignaturen MÜSSEN im SE#2 abgearbeitet werden. Die in [gemSpec_Krypt] angegebenen Festlegungen der zu unterstützenden Algorithmen MÜSSEN berücksichtigt werden.

Tabelle 6: TAB_KON_873 Fehlercodes TUC_KON_158 „Komfortsignaturen erstellen"

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4120	Security	Error	Kartenfehler
4123	Security	Error	Fehler bei Signaturerstellung
~~4271~~	~~Technical~~	~~Error~~	~~Komfortsignaturmodus abgebrochen~~

~~[<=]~~[<=]

Kommentiert [DS4]: C_10629

4.1.8.4 Interne TUCs, auch durch Fachmodule nutzbar (Kap. 4.1.8.4)

4.1.8.4.1 TUC_KON_170 „Dokumente mit Komfort signieren" (Kap. 4.1.8.4.7 - neu)

A_19103-04~~A_19103-03~~ - TUC_KON_170 "Dokumente mit Komfort signieren"
Der Konnektor MUSS den technischen Use Case TUC_KON_170 „Dokumente mit Komfort signieren" umsetzen.

Tabelle 7: TAB_KON_871 – TUC_KON_170 „Dokumente mit Komfort signieren"

Element	Beschreibung
Name	TUC_KON_170 "Dokumente mit Komfort signieren"
Beschreibung	Im Rahmen von Fachanwendungen werden ein oder mehrere Dokumente mit einer Komfortsignatur versehen. Es werden die QES_DocFormate unterstützt.
Auslöser	Aufruf durch ein Clientsystem (Operation SignDocument) oder ein Fachmodul.
Vorbedingungen	Die Signaturkarte muss gesteckt sein.
Eingangsdaten	• signRequests (Liste von Signaturaufträgen) Jeder Signaturauftrag (SignRequest) kapselt: • documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente); darin u.a. documentFormat (Formatangabe für das zu signierende Dokument) • optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur, siehe Operation SignDocument, Parameter dss:OptionalInputs); darin u.a. signatureType (URI für den Signaturtyp XML-, CMS-, PDF-Signatur) • includeRevocationInfo [Boolean]: – optional; Default: true (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur; siehe Operation SignDocument, Parameter SIG:IncludeRevocationInfo) • cardSession (Kartensitzung. Unterstützte Kartentypen: HBA) • crypt [SIG_CRYPT_QES] - *optional*; default und Wertebereich: siehe TAB_KON_862-01 (Dieser Parameter steuert, ob RSA-basierte oder ECC-basierte Signaturen erzeugt werden.)

	• workplaceId
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBA)
Ausgangsdaten	• signedDocuments (Liste der signierten Dokumente)
Standardablauf	Der Konnektor KANN die Schritte 1 bis 4 in einer beliebigen Reihenfolge durchführen. 1. Prüfe `SAK_COMFORT_SIGNATURE = Enabled` 2. Prüfe, ob der Komfortsignatur-Timer der cardSession (`SAK_COMFORT_SIGNATURE_TIMER`) abgelaufen ist. ~~2~~3. Der Signaturtyp und die Signaturvariante werden für jedes Dokument der Liste entsprechend signatureType und SignatureVariant festgelegt (ggf. in optionalInputs enthalten). Wenn SignatureType oder SignatureVariant nicht übergeben wurden, wird das dem Dokumentformat entsprechende Default-Verfahren gewählt (siehe TAB_KON_583 – Default-Signaturverfahren). ~~3~~4. Für alle Dokumente des Stapels wird die Zulässigkeit des Kartentyps geprüft. Das für die Signatur zu nutzende Zertifikat wird anhand des Kartentyps und des Parameters crypt ausgewählt. ~~4~~5. Es werden die Voraussetzungen für die Signatur geprüft. Dies erfolgt im TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“. Wenn includeRevocationInfo=true, dann setze ocspResponses auf Rückgabewert von TUC_KON_152. ~~5~~6. Die am Signaturvorgang beteiligte RessourceSignaturkarte wird für die exklusive Nutzung durch diesen Signaturvorgang reserviert. Die Reservierung der Signaturkarte erfolgt durch Aufruf von TUC_KON_023 „Karte reservieren“ { cardSession; doLock = true }. ~~6~~7. Zum Vorbereiten der Dokumente für die Signatur wird TUC_KON_155 „Dokumente zur Signatur vorbereiten“ mit ocspResponses aufgerufen. Die Zugriffe auf die Signaturkarte im Schritt 7 müssen im DF.QES erfolgen. DF.QES darf am Ende des TUCs nicht verlassen werden. ~~7~~8. Die Signaturen werden erstellt. Dies erfolgt gemäß TUC_KON_158 „Komfortsignaturen erstellen“. ~~8~~9. Die reservierte Ressource Signaturkarte wird wieder freigegeben. Zur Freigabe der Signaturkarte wird TUC_KON_023 „Karte reservieren“ cardSession; doLock = false } aufgerufen.

	~~9~~10. Die signierten Dokumente werden an den Aufrufer zurückgegeben.
Varianten/ Alternativen	keine
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (->1) Komfortsignaturfunktion im Konnektor nicht aktiviert: Fehlercode 4263 (->2) Der Komfortsignatur-Timer der cardSession ist abgelaufen: Fehlercode 4271 (->3) Ungültige Angabe des Signaturtyps oder Signaturvariante: Fehlercode 4111 Übergabe eines für die QES nicht unterstützten Dokumentformats: Fehlercode 4110 (->~~3~~4) Kartentyp nicht zulässig für Signatur: Fehlercode 4126 (->~~5~~6) Fehler bei der Reservierung der Signaturkarte: Fehlercode 4060 (->~~7~~8) Karte ist kein HBA, sondern HBA-Vorläuferkarte: Fehlercode 4274 Im Fehlerfall: a) … DARF DF.QES NICHT verlassen werden b) … MÜSSEN alle reservierten Ressourcen freigegeben werden c) … MUSS der Fehler immer an das Clientsystem zurückgemeldet werden
Sicherheits-anforderungen	Der Konnektor MUSS sicherstellen, dass der erhöhte Sicherheitszustand der PIN.QES nur für die Komfortsignatur mittels TUC_KON_170 innerhalb einer Kartensitzung nachgenutzt werden darf.

Tabelle 8: TAB_KON_872 Fehlercodes TUC_KON_170 „Dokumente mit Komfort signieren“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4060	Technical	Error	Ressource belegt
4110	Technical	Error	ungültiges Dokumentformat (%Format%) Der Parameter Format enthält das übergebene Dokumentformat.
4111	Technical	Error	ungültiger Signaturtyp oder Signaturvariante
4126	Security	Error	Kartentyp nicht zulässig für Signatur
4049	Technical	Error	Abbruch durch den Benutzer
4263	Technical	Error	Komfortsignaturfunktion nicht aktiviert

4271	Technical	Error	Komfortsignaturmodus abgebrochen
4274	Technical	Error	Komfortsignaturen werden nur für den HBA unterstützt

~~[<=]~~[<=]

Kommentiert [DS5]: C_10629

4.1.8.4.2 TUC_KON_171 „Komfortsignatur einschalten" (Kap 4.1.8.4.8 - neu)

A_19104-02~~A_19104-01~~ - TUC_KON_171 „Komfortsignatur einschalten"
Der Konnektor MUSS den technischen Use Case TUC_KON_171 „Komfortsignatur einschalten" umsetzen.

Tabelle 9: TAB_KON_883 – TUC_KON_171 „Komfortsignatur einschalten"

Element	Beschreibung
Name	TUC_KON_171 „Komfortsignatur einschalten"
Beschreibung	Zum Einschalten des Komfortsignaturmodus wird die PIN.QES verifiziert und der Signaturmodus „Comfort" für die cardSession gesetzt.
Auslöser	• Operation ActivateComfortSignature • Aufruf durch ein Fachmodul
Vorbedingungen	Der Karte muss gesteckt sein.
Eingangsdaten	• cardSession (nur HBA erlaubt)
Komponenten	Konnektor, Kartenterminal, Karte (HBA)
Ausgangsdaten	• signatureMode

Standardablauf	1. Prüfe `SAK_COMFORT_SIGNATURE = Enabled`
	2. Die am Vorgang beteiligten Ressourcen (Karte sowie PIN-Pad und Display des PIN-Eingabe-Kartenterminals) werden für die exklusive Nutzung durch diesen Vorgang reserviert. Die Reservierung der Karte erfolgt durch Aufruf von
	TUC_KON_023 „Karte reservieren" {
	cardSession;
	doLock = true }
	Der Zugriff auf die Karte im Schritt 3 muss im DF.QES erfolgen. Das DF.QES darf danach nicht verlassen werden, damit der PIN-Status der PIN.QES erhalten bleibt.
	3. Die Einschaltung der Komfortsignatur wird durch den Anwender autorisiert. Dies erfolgt durch Aufruf von TUC_KON_012 „PIN verifizieren" { cardSession;
	workplaceId;
	pinRef = PIN.QES;
	verificationType = Mandatorisch }
	Für die Anzeige am Kartenterminal ist die Displaymessage

	für „Komfortsignatur aktivieren“ aus TAB_KON_090 zu verwenden. 4. Setze CARDSESSION.SIGNMODE = Comfort 5. Starte Komfortsignatur-Timer für die cardSession bei „0“ 6. Die reservierten Ressourcen (Karte sowie PIN-Pad und Display des PIN-Eingabe-Kartenterminals) werden wieder freigegeben. Zur Freigabe der Karte wird TUC_KON_023 „Karte reservieren“ cardSession; doLock = false } aufgerufen.
Varianten/ Alternativen	Keine
Fehlerfälle	Fehler in den folgenden Schritten des Standardablaufs führen zum Abbruch mit den ausgewiesenen Fehlercodes: (->1) Komfortsignaturfunktion im Konnektor nicht aktiviert: Fehlercode 4263 (->2) Fehler bei der Reservierung von Ressourcen: Fehlercode 4060 (->3) Karte ist kein HBA, sondern HBA-Vorläuferkarte: Fehlercode 4274 (->3) pinResult = BLOCKED: Fehlercode 4275 (->3) pinResult = REJECTED: Fehlercode 4276 (->4) Fehler beim Setzen des Signaturmodus: Fehlercode 4267 (->5) Fehler beim Starten des Komfortsignatur-Timers: Fehlercode 4267 Im Fehlerfall, inklusive Timeout bei der PIN-Eingabe, oder bei Abbruch durch den Benutzer (Fehler 4049): a) … MUSS (ab Schritt 3) DF.QES verlassen werden b) … MÜSSEN alle reservierten Ressourcen freigegeben werden c) … MUSS der Fehler immer an das Clientsystem zurückgemeldet werden

413 **Tabelle 10: TAB_KON_886 Fehlercodes TUC_KON_171 „Komfortsignatur einschalten“**

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4049	Technical	Error	Abbruch durch den Benutzer
4060	Technical	Error	Ressource belegt
4263	Technical	Error	Komfortsignaturfunktion nicht aktiviert
4267	Technical	Error	Fehler beim Aktivieren des Komfortsignaturmodus <cardHandle>

4274	Technical	Error	Komfortsignaturen werden nur für den HBA unterstützt
~~[<=]~~4275	Technical	Error	Security Error PIN jetzt gesperrt (BLOCKED)
4276	Technical	Error	Security Error PIN falsch (REJECTED)

[<=]

Kommentiert [DS6]: C_10628

4.1.8.4.3 TUC_KON_172 „Komfortsignatur ausschalten" (Kap 4.1.8.4.9 - neu)

A_19105 - TUC_KON_172 „Komfortsignatur ausschalten"
Der Konnektor MUSS den technischen Use Case TUC_KON_172 „Komfortsignatur ausschalten" umsetzen.

Tabelle 11: TAB_KON_884 – TUC_KON_172 „Komfortsignatur ausschalten"

Element	Beschreibung
Name	TUC_KON_172 „Komfortsignatur ausschalten"
Beschreibung	Zum Ausschalten des Komfortsignaturmodus werden die Sicherheitszustände der Karte(n), die im Konnektor verwalteten Sicherheitszustände und der Signaturmodus der cardSession(s) zurückgesetzt.
Auslöser	• Operation DeactivateComfortSignature • TUC_KON_158 • Der Administrator setzt SAK_COMFORT_SIGNATURE = Disabled • Aufruf durch ein Fachmodul
Vorbedingungen	Die Karten müssen gesteckt sein.
Eingangsdaten	Bei Auslösen des TUCs durch den Administrator: • Keine Ansonsten: • cardHandles : Liste von cardHandles (nur HBA erlaubt)
Komponenten	Konnektor, Kartenterminal, Karte (HBA)
Ausgangsdaten	Keine
Standardablauf	1. Wenn der TUC nicht durch den Administrator ausgelöst wurde: Prüfe `SAK_COMFORT_SIGNATURE = Enabled` 2. Wenn der TUC durch den Administrator ausgelöst wurde: Ermittle die cardHandles aller gesteckten HBA. 3. Für jedes übergebene bzw. ermittelte cardHandle: 4. Ermittle cardSessions zu cardHandle

	5. Für jede ermittelte cardSession: a. Setze den PIN-Status der PIN.QES zurück (z. B. durch Verlassen von DF.QES für alle logischen Kanäle der Karte) b. Lösche den im Konnektor verwalteten Sicherheitszustand aus `CARDSESSION.AUTHSTATE (PINRef=PIN.QES)` c. Setze `CARDSESSION.SIGNMODE = PIN` d. Stoppe Komfortsignatur-Timer für die cardSession
Varianten/ Alternativen	Keine
Fehlerfälle	(->1) Komfortsignaturfunktion im Konnektor nicht aktiviert: Fehlercode 4263 Fehler und Warnungen in den folgenden Schritten werden über alle cardHandle akkumuliert und die <komma-separierte Liste von cardHandle> für den jeweiligen Fehlertext erzeugt. (->3) Bei einem ungültigen cardHandle wird mit dem nächsten cardHandle aus cardHandles fortgesetzt. Fehlercode 4265 (->4) Ist zu einem cardHandle keine cardSession vorhanden wird mit dem nächsten cardHandle fortgesetzt. Fehlercode 4266 (->5) Tritt in Schritt 4 ein Fehler auf wird mit dem nächsten cardHandle fortgesetzt. Fehlercode 4268

Tabelle 12: TAB_KON_887 Fehlercodes TUC_KON_172 „Komfortsignatur ausschalten“

Fehlercode	ErrorType	Severity	Fehlertext
4263	Technical	Fehler	Komfortsignaturfunktion nicht aktiviert
4265	Technical	Warning	Karten-Handle ungültig <komma-separierte Liste von cardHandle>
4266	Technical	Warning	Keine Kartensitzung vorhanden <komma-separierte Liste von cardHandle>
4268	Technical	Fehler	Fehler beim Deaktivieren des Komfortsignaturmodus <komma-separierte Liste von cardHandle>

[<=]

4.1.8.4.4 TUC_KON_173 „Liefere Signaturmodus” (Kap. 4.1.8.4.10 -neu)

A_19106-01~~A_19106~~ - TUC_KON_173 „Liefere Signaturmodus“
Der Konnektor MUSS den technischen Use Case TUC_KON_173 „Liefere Signaturmodus“ umsetzen.

Tabelle 13: TAB_KON_885 – TUC_KON_173 „Liefere Signaturmodus“

Element	Beschreibung
Name	TUC_KON_173 „Liefere Signaturmodus“
Beschreibung	Der aktuell konfigurierte Status der Komfortsignaturfunktion im Konnektor und, falls vorhanden, Informationen zu der ~~aktuelle Signaturmodus für alle dem~~aktuell im Konnektor ~~bekannten Aufrufkontexte zu den übergebenen HBA-CardHandles wird~~existierenden Komfortsignatursession werden ermittelt und an den Aufrufer zurückgegeben.
Auslöser	• Operation GetSignatureMode • Aufruf durch ein Fachmodul
Vorbedingungen	Keine
Eingangsdaten	• ~~Liste von cardHandles (nur~~cardSession (Kartensitzung. Unterstützte Kartentypen: HBA~~ erlaubt~~)
Komponenten	Konnektor, Kartenterminal, Signaturkarte (HBA)
Ausgangsdaten	• comfortSignatureStatus • comfortSignatureMax • comfortSignatureTimer • ~~signatureModes:~~sessionInfo (optional): Struktur aus ~~• Liste von cardHandles (nur HBA erlaubt)~~ ~~• Liste von Tupeln (signatureContext,~~ signatureMode, countRemaining, timeRemaining~~)~~
Standardablauf	1. Ermittle den Status der Komfortsignaturfunktion: comfortSignatureStatus=`SAK_COMFORT_SIGNATURE` 2. Ermittle comfortSignatureMax= `SAK_COMFORT_SIGNATURE_MAX` 3. Ermittle comfortSignatureTimer= `SAK_COMFORT_SIGNATURE_Timer` ~~1. Für jedes übergebene cardHandle:~~ 4. Ermittle ~~cardSession zu cardHandle~~sessionInfo ~~a. Für jede ermittelte cardSession:~~ ~~i. Ermittle den Kontext (signatureContext) der cardSession aus `CARDSESSION.MANDANTID, CARDSESSION.CSID, CARDSESSION.USERID`~~ a. ~~ii.~~ Ermittle den Signaturmodus (signatureMode) aus `CARDSESSION.SIGNMODE` b. ~~iii.~~ Ermittle Differenz von `SAK_COMFORT_SIGNATURE_MAX` und Komfortsignatur-Zähler der cardSession (countRemaining)

	c. ~~iv.~~ Ermittle verbleibende Zeit aus SAK_COMFORT_SIGNATURE_TIMER und Komfortsignatur-Timer der cardSession (timeRemaining) 5. Wenn signatureMode = "Comfort" wird sessionInfo an den Aufrufer zurückgegeben.
Varianten/ Alternativen	Keine~~Wenn SAK_COMFORT_SIGNATURE = Disabled (->4-b iii) countRemaining = 0 (->4-b iv) timeRemaining = 0~~
Fehlerfälle	~~(->2) Bei einem ungültigen cardHandle wird mit dem nächsten cardHandle aus cardHandles fortgesetzt. Fehlercode 4265 (->2a) Ist zu einem cardHandle keine cardSession vorhanden wird mit dem nächsten cardHandle fortgesetzt. Fehlercode 4266 (->2b) Tritt in Schritt 2b ein Fehler auf wird mit dem nächsten cardHandle fortgesetzt. Fehlercode 4269 Die Fehler und Warnungen werden über alle cardHandle akkumuliert und die <komma-separierte Liste von cardHandle> für den jeweiligen Fehlertext erzeugt.~~Wenn im Standardablauf ein Fehler auftritt, wird mit Fehler 4269 abgebrochen.

Tabelle 14: TAB_KON_888 Fehlercodes TUC_KON_173 „Liefere Signaturmodus“

Fehlercode	ErrorType	Severity	Fehlertext
~~4265~~	~~Technical~~	~~Warning~~	~~Karten-Handle ungültig <komma-separierte Liste von cardHandle>~~
~~4266~~	~~Technical~~	~~Warning~~	~~Keine Kartensitzung vorhanden <komma-separierte Liste von cardHandle>~~
4269	Technical	Error	Fehler beim Ermitteln des Signaturmodus <~~komma-separierte Liste von~~ cardHandle>

~~[<=]~~[<=]

Kommentiert [DS7]: C_10555

4.1.8.5 Operationen an der Außenschnittstelle (Kap. 4.1.8.5)

TIP1-A_4676-08~~**TIP1-A_4676-05**~~ **-** Basisdienst Signaturdienst (nonQES und QES)
Der Konnektor MUSS Clientsystemen den Basisdienst Signaturdienst (nonQES und QES) anbieten.

Tabelle 15: TAB_KON_197 Basisdienst Signaturdienst (nonQES und QES)

Name	SignatureService
Version (KDV)	7.4.0 (WSDL-Version), 7.4.2 (XSD-Version) 7.4.2 (WSDL-Version), 7.4.4 (XSD-Version) 7.5.~~2~~4 (WSDL- und XSD-Version)

	Siehe Anhang D	
Namensraum	Siehe Anhang D	
Namensraum-Kürzel	SIG für Schema und SIGW für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	SignDocument	Dokument signieren
	VerifyDocument	Signatur verifizieren
	StopSignature	Signieren eines Dokumentenstapels abbrechen
	GetJobNumber	Liefert eine Jobnummer für den nächsten Signiervorgang
	ActivateComfortSignature	Aktiviert die Komfortsignatur für einen HBA
	DeactivateComfortSignature	Deaktiviert die Komfortsignatur für einen oder mehrere HBA
	GetSignatureMode	Liefert den Status der Komfortsignaturfunktion und ~~den Signaturmodus für einen oder mehrere~~Informationen zur Komfortsignatursession eines HBA
WSDL	SignatureService_V7_5_~~2~~4.wsdl SignatureService_V7_4_2.wsdl SignatureService.wsdl (WSDL-Version 7.4.0)	
Schema	SignatureService_V7_5_~~2~~4.xsd SignatureService_V7_4_4.xsd SignatureService.xsd (XSD-Version 7.4.2)	

~~[<=]~~[<=]

Kommentiert [DS8]: C_10555

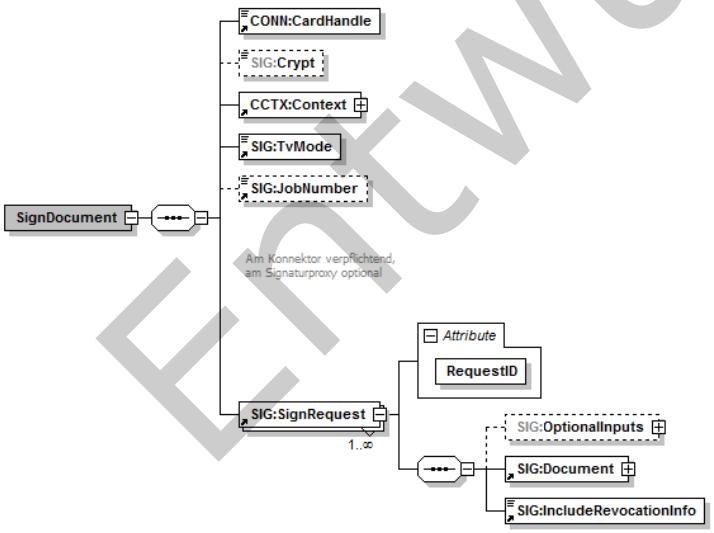
Kommentiert [DS9]: C_10614 - wirkt mit 7.5.4 er Schema

4.1.8.5.1 SignDocument (nonQES und QES) (Kap. 4.1.8.5.1)

TIP1-A_5010-06 **-** Operation SignDocument (nonQES und QES)
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation SignDocument anbieten**.**

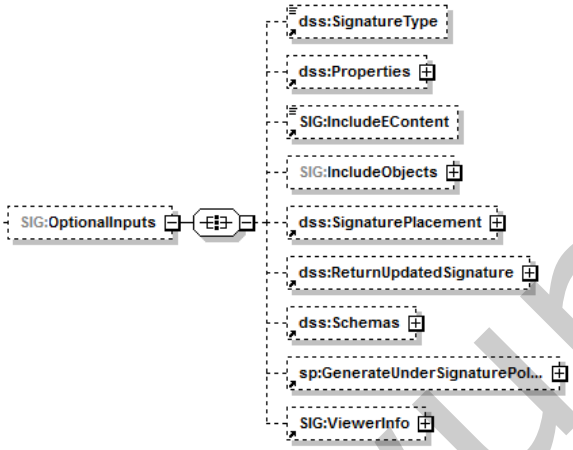
Tabelle 16: TAB_KON_065 Operation SignDocument (nonQES und QES)

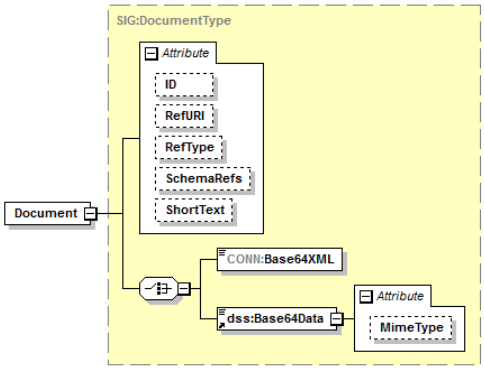
Name	**SignDocument**

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

Name	Beschreibung

	CONN: Card Handle	Identifiziert die zu verwendende Signaturkarte. Die Operation DARF die Signatur mit der eGK NICHT unterstützen. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4126 abbrechen.
	SIG: Crypt	Der Parameter crypt steuert die Auswahl der Zertifikate und Schlüssel für die Signaturerstellung abhängig von der durch cardHandle adressierten Karte gemäß TAB_KON_900. Defaultwert: • gemäß TAB_KON_862-01für die QES • gemäß TAB_KON_863 für die nonQES.
	CCTX: Context	Aufrufkontext QES mit HBAx: MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend Aufrufkontext nonQES mit SM-B: MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
	TvMode	Der Parameter wird im Konnektor nicht ausgewertet.
	SIG: JobNumber	Die Nummer des Jobs, unter der der nächste Signaturvorgang gestartet wird. Parameter ist verpflichtend.
	SIG: Sign Request	Ein `SignRequest` kapselt den Signaturauftrag für ein Dokument. Das verpflichtende XML-Attribut `RequestID` identifiziert einen `SignRequest` innerhalb eines Stapels von `SignRequests` eindeutig. Es dient der Zuordnung der `SignResponse` zum jeweiligen `SignRequest`.

	SIG: OptionalInputs	Enthält optionale Eingangsparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7): 

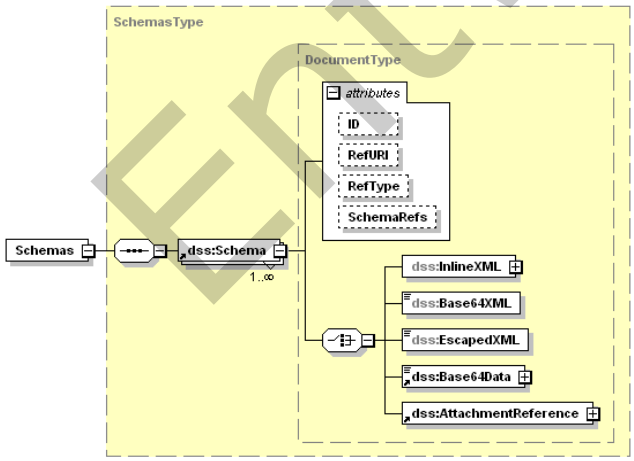
	SIG: Document	 Dieses an das dss:Document Element aus [OASIS-DSS] Section 2.4.2 angelehnte Element enthält das zu signierende Dokument, wobei die Kindelemente CONN:Base64XML und dss:Base64Data auftreten können. Bei den als dss:Base64Data übergebenen Dokumenten werden folgende (Klassen von) MIME-Types unterschieden: • ”application/pdf-a” – für PDF/A-Dokumente, • ”text/plain”, ”text/plain; charset=iso-8859-15” oder ”text/plain; charset=utf-8” – für Text-Dokumente, • ”image/tiff” – für TIFF-Dokumente und • ein beliebiger anderer MIME-Type für nicht näher unterschiedene Binärdaten des spezifizierten Typs. Der MIME-Type „text/plain“ wird interpretiert als „text/plain; charset=iso-8859-15”. Das Element enthält ein Attribut ShortText Es muss für QES-Signaturen bei jedem Aufruf vom Clientsystem übergeben werden, für nonQES-Signaturen ist es optional. Über das Attribut RefURI kann gemäß [OASIS-DSS] (Abschnitt 2.4.1) ein zu signierender Teilbaum eines XML-Dokuments ausgewählt werden. Wenn die Signatur eines Teilbaums für die Signaturvariante nicht unterstützt wird, muss der Signaturauftrag mit Fehler 4111 abgelehnt werden.

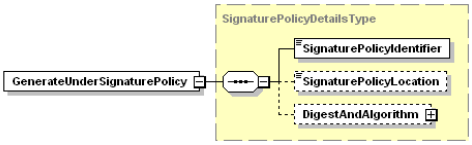
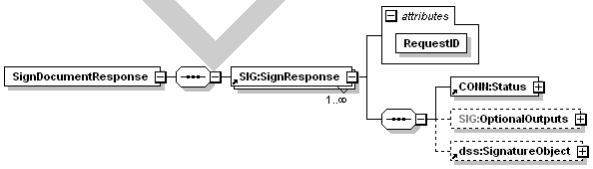
	SIG: Include Revocation Info	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturerstellung vorliegenden Sperrinformationen anfordern. Es wird ausschließlich die zu erstellende Signatur betrachtet, d.h. es erfolgt keine Einbettung von Sperrinformationen für bereits enthaltene Signaturen. Für nicht-qualifizierte elektronische Signaturen (nonQES) wird diese Funktionalität nicht unterstützt. Für PDF-Signaturen werden keine Sperrinformationen eingebettet.

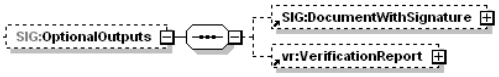
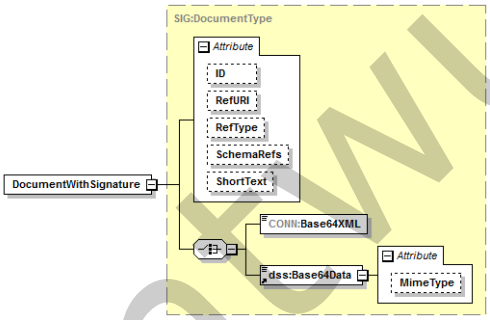
	`dss:SignatureType`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element kann der generelle Typ der zu erzeugenden Signaturen spezifiziert werden. Hierbei MÜSSEN folgende Signaturtypen unterstützt werden: • **XML-Signatur** Durch Übergabe der URI urn:ietf:rfc:3275 wird die Erstellung von XML-Signaturen gemäß [RFC3275], [XMLDSig] angestoßen. Das zu verwendende Profil ist XAdES-BES ([XAdES]). Die Rückgabe einer solchen Signatur erfolgt als `ds:Signature`-Element. • **CMS-Signatur** Durch Übergabe der URI urn:ietf:rfc:5652 wird eine CMS-Signatur gemäß [RFC5652] angestoßen. Das zu verwendende Profil ist CAdES-BES ([CAdES]). Die Signatur wird als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert. • **S/MIME-Signatur** Durch Übergabe der URI „urn:ietf:rfc:5751" wird eine S/MIME-Signatur gemäß [RFC5751] angestoßen. Die CMS-Signatur der übergebenen MIME-Nachricht erfolgt konform der Vorgaben zur CMS-Signatur. Das Rückgabedokument ist eine MIME-Nachricht vom Typ „application/pkcs7-mime" mit einer CMS-Struktur vom Typ_SignedData. Ist das übergebene Dokument keine MIME-Nachricht, so wie der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. • **PDF-Signatur** Durch Übergabe der URI http://uri.etsi.org/02778/3 wird die Erzeugung einer PAdES-Basic Signatur gemäß [PAdES-3]angestoßen, wobei das Dokument mit der integrierten Signatur als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert wird. Handelt es sich beim übergebenen Dokument nicht um ein `Base64Data`-Element mit MIME-Type „application/pdf-a", so wird ein Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder Signaturvariante). Die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur" DÜRFEN für QES der HBAx

		nur mit dem QES-Zertifikat erfolgen, für nonQES nur mit dem OSIG-Zertifikat der SM-B. In jedem diese Anforderung verletzenden Fall MUSS der Fehler 4058 (Aufruf nicht zulässig) zurückgeliefert werden. Fehlt dieses Element, so wird der Signaturtyp gemäß TAB_KON_583 – Default-Signaturverfahren aus dem Dokumententyp abgeleitet.

	dss: Properties	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.5) definierte Element können zusätzliche signierte und unsignierte Eigenschaften (Properties) bzw. Attribute in die Signatur eingefügt werden. Unterstützt werden genau folgende Attribute: Im CMS-Fall (SignatureType = urn:ietf:rfc:5652) kann es XML-Elemente ./SignedProperties/Property/Value/CMSAttribute und ./UnsignedProperties/Property/Value/CMSAttribute enthalten. Ein solches XML-Element CMSAttribute muss ein vollständiges, base64/DER-kodiertes ASN.1-Attribute enthalten, definiert in [CMS#5.3.SignerInfo Type]. Es muss bei der Erstellung des CMS-Containers unverändert unter SignedAttributes bzw. UnsignedAttributes aufgenommen werden.
	SIG: Include EContent	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.7), definierte Element kann bei einer CMS-basierten Signatur das Einfügen des signierten Dokumentes in die Signatur angefordert werden. Die Verwendung dieses Parameters bei anderen Signaturtypen führt zu einem Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante).
	SIG: Include Object	Dieses Element enthält zum Anfordern einer Enveloping XML Signatur ein dss:IncludeObject-Element gemäß [OASIS-DSS] (Abschnitt 3.5.6). Ist das Element vorhanden und ein anderer Signaturtyp als eine XML-Signatur angefordert, so wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.
	dss: Signature Placement	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.8) definierte Element kann bei XML-basierten Signaturen gemäß [RFC3275] die Platzierung der Signatur im Dokument angegeben werden. Die in [OASIS-DSS] (Abschnitt 2.5, XPath c) beschriebene Deklaration von Namespace-Prefixes im dss:SignaturePlacement-Element muss nicht unterstützt werden. Bei anderen Signaturtypen wird das Element ignoriert und eine Warnung (Fehlercode 4197, Parameter SignaturePlacement wurde ignoriert) zurückgeliefert.

	`dss: Return Updated Signature`	Durch dieses in [OASIS-DSS] (Abschnitt 4.5.8) definierte Element kann eine übergegebene XML- oder CMS-Signatur mit zusätzlichen Informationen und Signaturen (Parallel- und Gegensignaturen) versehen werden. Hierbei sind folgende Ausprägungen für das `Type`-Attribut vorgesehen: • http://ws.gematik.de/conn/sig/sigupdate/parallel Hierdurch wird eine Parallelsignatur zu einer bereits existierenden Signatur erzeugt und entsprechend zurückgeliefert. • http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt. Bei anderen `Type`-Attributen wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.
	`dss: Schemas`	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schemata übergeben werden, die zur Validierung der übergebenen XML-Dokumente verwendet werden können.
		

	dss:Schema	Dieses Element enthält ein XML-Schema zur Validierung des übergebenen XML-Dokuments. Das Attribut RefURI ist verpflichtend. Es kennzeichnet dabei den Namensraum des XML-Schemas entsprechend [OASIS-DSS] (Abschnitt 2.8.5)
	sp: GenerateUnderSignaturePolicy	 Über dieses in [OASIS-SP], Kapitel 2.2.1.1.1.1 Optional Input <GenerateUnderSignaturePolicy>, definierte Element wird die erforderliche Singnaturrichlinie ausgewählt. Die im Element sp:SignaturePolicyIdentifier übergebene URI identifiziert die Signaturrichtlinie. Die XML-Elemente SignaturePolicyLocation DigestAndAlgorithm werden nicht verwendet. Wenn eine nach TAB_KON_778 notwendige Signaturrichtlinie fehlt oder die übergebene Signaturrichtlinie unbekannt ist, wird Fehler 4111 zurückgeliefert.
	SIG: Viewer Info	Enthält optional die vom Konnektor in die Signatur einzubeziehende Referenzen für die Stylesheets zur Anzeige.
Rückgabe		
	SIG: Sign Response	Eine SignResponse kapselt den ausgeführten Signaturauftrag pro Dokument. Die Zuordnung zwischen SignRequest und SignResponse erfolgt über die

		RequestID.
	CONN: Status	Enthält den Status der ausgeführten Operation pro SignRequest.
	SIG: Optional Outputs	Enthält (angelehnt an `dss:OptionalOutputs`) optionale Ausgangsparameter: 
	SIG: Document With Signature	 Pro `SignResponse` wird ein Element `SIG:DocumentWithSignature` gemäß [OASIS-DSS] (Abschnitt 3.5.8) zurückgeliefert, in dem das Dokument mit Signatur enthalten ist. Dabei werden die XML-Attribute des Elements `SIG:Document` auf dem zugehörigen `SignRequest` übernommen. Ist die Signatur nicht im Dokument enthalten, wird ein leeres Element `Base64XML` oder `Base64Data` zurückgegeben. Die Signatur wird dann im Element `dss:SignatureObject` abgelegt. Wenn die Signatur im Dokument enthalten ist, wird das signierte Dokument im Feld `Base64XML` bzw. `Base64Data` zurückgeliefert. In diesem Fall MUSS die `dss:SignaturePtr`-Alternative in `dss:SignatureObject` (vgl. [OASIS-DSS] Abschnitt 2.5) dazu genutzt werden, auf die in den Dokumenten enthaltenen Signaturen zu verweisen.

	`vr: VerificationReport`	Vom Konnektor nicht befüllt.
	`dss: SignatureObject`	Enthält im Erfolgsfall die erzeugte Signatur pro `SignRequest` in Form eines `dss:SignatureObject`-Elementes gemäß [OASIS-DSS] (Abschnitt 3.2).
Vorbedingungen	Keine	
Nachbedingungen	Keine	

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

Tabelle 17: TAB_KON_756 Ablauf Operation SignDocument (nonQES und QES)

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

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

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

Im Fall QES wird Schritt 4 ausgeführt. Im Fall nonQES wird Schritt 5 ausgeführt.		
4a)	Prüfe Signaturdienst-Modul	Prüfe, ob MGM_LU_SAK=Enabled. Ist dies nicht der Fall, so bricht die Operation mit Fehler 4125 ab.
Wenn für die CardSession die Komfortsignatur aktiviert ist (CARDSESSION.SIGNMODE = Comfort) wird Schritt 4 c) ausgeführt. Andernfalls wird Schritt 4 b) ausgeführt.		
4b)	TUC_KON_150 „Dokumente QES signieren“	Die QES wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.
4c)	TUC_KON_170 „Dokumente mit Komfort signieren“	Eine Komfortsignatur wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.
5)	TUC_KON_160 „Dokumente nonQES signieren“	Die nonQES wird erzeugt. Tritt hierbei ein Fehler auf, bricht die Operation ab.

Tabelle 18: TAB_KON_757 Fehlercodes „SignDocument (nonQES und QES)“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4111	Technical	Error	ungültiger Signaturtyp oder Signaturvariante
4126	Security	Error	Kartentyp nicht zulässig für Signatur
4125	Technical	Error	LU_SAK nicht aktiviert
4197	Technical	Warning	Parameter SignaturePlacement wurde ignoriert

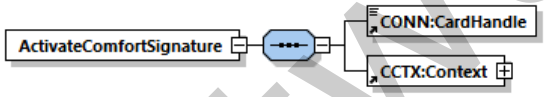
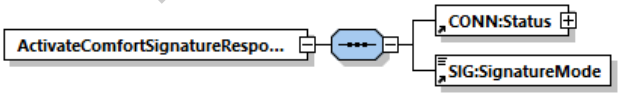
4252	Technical	Error	Jobnummer wurde in den letzten 1.000 Aufrufen bereits verwendet und ist nicht zulässig

Die zulässigen Zertifikate und Schlüssel sind in TAB_KON_900 aufgelistet.
[<=]

4.1.8.5.2 ActivateComfortSignature (Kap. 4.1.8.5.5 - neu)

A_19107 **-** Operation ActivateComfortSignature
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation ActivateComfortSignature anbieten.

Tabelle 19: TAB_KON_874 ActivateComfortSignature

<table>
<tr><td>Name</td><td colspan="2">ActivateComfortSignature</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Diese Operation aktiviert die Komfortsignatur für einen HBA bezogen auf einen Aufrufkontext.</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td>CONN:
Card
Handle</td><td>Identifiziert die zu adressierende Karte.
Es wird nur der HBA unterstützt.</td></tr>
<tr><td></td><td>CCTX:Context</td><td>MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend zu übergeben;
MandantId, WorkplaceId nicht ausgewertet</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td></td><td>CONN:Status</td><td>Enthält den Ausführungsstatus der Operation.</td></tr>
<tr><td></td><td>SIG:
SignatureMode</td><td>Signaturmodus des HBA
Enthält bei erfolgreicher Ausführung der Operation den Wert „COMFORT“</td></tr>
<tr><td>Vorbedingungen</td><td colspan="2">Keine</td></tr>
</table>

Nachbedingungen	Keine

469 **Tabelle 20: TAB_KON_877 Ablauf ActivateComfortSignature**

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 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		userId = $context.userId;
		cardHandle = $cardHandle }
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

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

4.	TUC_KON_171 „Komfortsignatur einschalten“	Der Komfortsignaturmodus wird für das Tupel (CardHandle, CardSession) eingeschaltet. Tritt hierbei ein Fehler auf, bricht die Operation ab.

470

471 **Tabelle 21: TAB_KON_879 Fehlercodes ActivateComfortSignature**

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4270	Technical	Error	UserId wurde in den letzten 1.000 Vorgängen bereits verwendet
4272	Technical	Error	UserId nicht zulässig

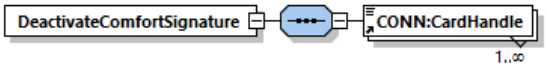
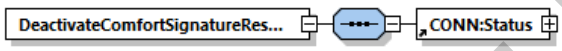
472 **[<=]**

473

474 *4.1.8.5.3 DeactivateComfortSignature (Kap. 4.1.8.5.6 - neu)*

475 A_19108 **-** Operation DeactivateComfortSignature
476 Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation
477 DeactivateComfortSignature anbieten.
478

479 **Tabelle 22: TAB_KON_875 DeactivateComfortSignature**

Name	DeactivateComfortSignature	
Beschreibung	Diese Operation deaktiviert die Komfortsignatur für einen oder mehrere HBA.	
Aufrufparameter		
	Name	Beschreibung
	CONN: Card Handle	Identifiziert die zu adressierende Karte. Es wird nur der HBA unterstützt.
Rückgabe		
	CONN:Status	Enthält den Ausführungsstatus der Operation.
Vorbedingungen	Keine	
Nachbedingungen	Keine	

480 **Tabelle 23: TAB_KON_878 Ablauf DeactivateComfortSignature**

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_172 „Komfortsignatur ausschalten“	Der Komfortsignaturmodus wird für alle Karten aus der CardHandle-Liste ausgeschaltet.

481

482 **Tabelle 24: TAB_KON_880 Fehlercodes DeactivateComfortSignature**

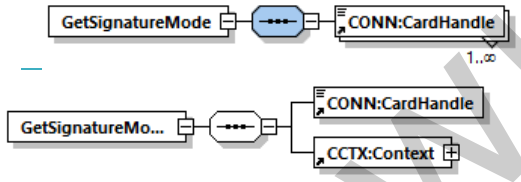
Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen TUCs können folgende weiteren Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler

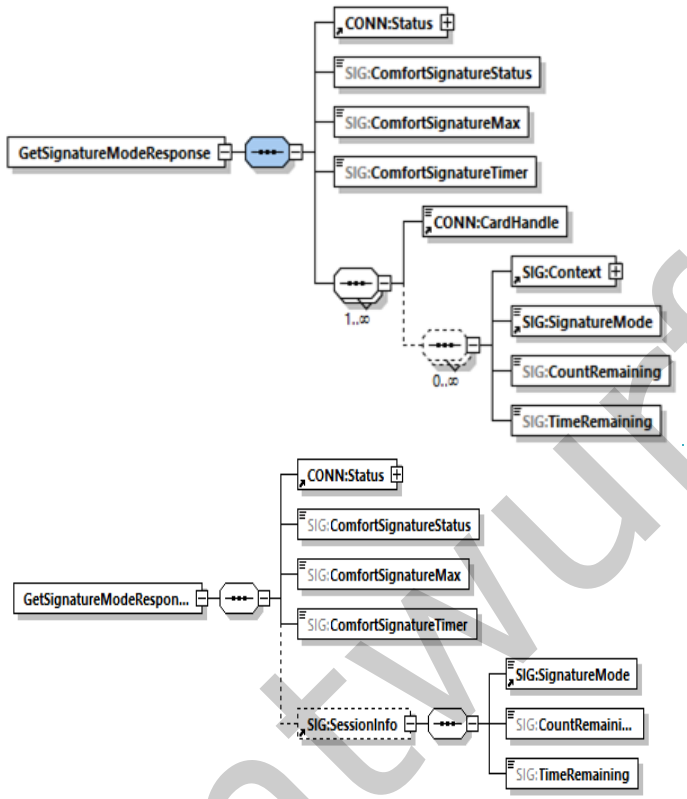
483 **[<=]**

4.1.8.5.4 GetSignatureMode (Kap. 4.1.8.5.7 - neu)

A_19109-01~~A_19109~~ - Operation GetSignatureMode
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine Operation GetSignatureMode anbieten.

Tabelle 25: TAB_KON_876 GetSignatureMode

Name	GetSignatureMode	
Beschreibung	Diese Operation liefert den aktuell konfigurierten Status der Komfortsignaturfunktion im Konnektor und ~~die~~ , falls vorhanden, Informationen zu der aktuell im Konnektor ~~aktuell hinterlegten Signaturmodus zu allen HBA aus der übergebenen~~existierenden Komfortsignatursession für das CardHandle ~~Liste~~ und den Aufrufkontext.	
Aufrufparameter		
	Name	Beschreibung
	CONN: ~~Card~~ ~~Handle~~CardHandle	Identifiziert die zu adressierende Karte. Es wird nur der HBA unterstützt.
	CCTX:Context	MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend zu übergeben

Rückgabe		
	`CONN:Status`	Enthält den Ausführungsstatus der Operation.
	`SIG: ComfortSignatureStatus`	Komfortsignatur-Konfigurationsstatus des Konnektors
	`SIG:ComfortSignatureMax`	Im Konnektor konfigurierte Anzahl von Komfortsignaturen, die ohne erneute PIN-Eingabe ausgeführt werden dürfen[7]
	`SIG: ComfortSignatureTimer`	Im Konnektor konfiguriertes Zeitintervall, in dem Komfortsignaturen

		ohne erneute PIN-Eingabe ausgeführt werden dürfen, Format: "PTnHnMnS" (gemäß Datenttyp xsd:duration)
	~~CONN:CardHandle~~ —SIG:SessionInfo	~~Liste von HBA-CardHandles~~Falls vorhanden, Informationen zu der aktuell im Konnektor existierenden Komfortsignatursession für das CardHandle und den Aufrufkontext
	SIG:~~Context~~ —SignatureMode	~~Liste von im Konnektor hinterlegten Aufrufkontexten für das jeweilige HBA-CardHandle MandantId, ClientSystemId, UserId verpflichtend~~Signaturmodus der Komfortsignatursession (="Comfort")
	SIG:~~SignatureMode~~CountRemaining	~~Im Konnektor hinterlegter Signaturmodus für den jeweiligen Aufrufkontext~~Verbleibende Anzahl von Komfortsignaturen, die ohne erneute PIN-Eingabe ausgeführt werden dürfen
	SIG:~~CountRemaining~~TimeRemaining	Verbleibende ~~Anzahl von~~Zeit, in der Komfortsignaturen~~, die~~ ohne erneute PIN-Eingabe ausgeführt werden dürfen Format:

		"PTnHnMnS" (gemäß Datenttyp xsd:duration)
	~~SIG:TimeRemaining~~	~~Verbleibende Zeit, in der Komfortsignaturen ohne erneute PIN-Eingabe ausgeführt werden dürfen Format: "PTnHnMnS" (gemäß Datenttyp xsd:duration)~~
Vorbedingungen	Keine	
Nachbedingungen	Keine	

491 **Tabelle 26: TAB_KON_882 Ablauf GetSignatureMode**

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 { mandantId = $context.mandantId; clientsystemId = $context.clientsystemId; workplaceId = $context.workplaceId; userId = $context.userId; cardHandle = $cardHandle } Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 { mandatId =$context.mandantId; clientsystemId = $context.clientsystemId; cardHandle = $context.cardHandle; userId = $context.userId }
~~2~~4.	TUC_KON_173 „Liefere Signaturmodus“	Der Komfortsignatur-Konfigurationsstatus des Konnektors und ~~die~~der im Konnektor ~~hinterlegten~~hinterlegte Signaturmodus werden für ~~alle~~den dem Konnektor bekannten ~~Aufrufkontexte der~~Aufrufkontext des HBA aus ~~der~~dem übergebenen CardHandle ~~Liste zurückgeliefert~~ zurück geliefert.

492

Tabelle 27: TAB_KON_881 Fehlercodes GetSignatureMode

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

~~[<=]~~[<=]

Kommentiert [DS10]: C_10555

4.1.8.6 Betriebsaspekte (Kap 8.1.8.6)

TIP1-A_4680-03~~TIP1-A_4680-02~~ - Konfigurationswerte des Signaturdienstes
Die Managementschnittstelle MUSS es einem Administrator ermöglichen Konfigurationsänderungen gemäß Tabelle TAB_KON_596 vorzunehmen:

Tabelle 28: TAB_KON_596 Konfigurationswerte des Signaturdienstes (Administrator)

ReferenzID	Belegung	Bedeutung und Administrator-Interaktion
SAK_SIMPLE_ SIGNATURE_ MODE	SE#1 SE#2	Aktivierung/Deaktivierung des „Einfachsignaturmodus" für alle HBAx für die Durchführung von Einfachsignaturen im SecurityEnvironment #1 (SE#1) für Dokumentenstapel der Größe 1 anstelle der Verwendung des SE#2. Default-Wert = SE#1 ~~Der Parameter ist nur relevant, wenn die Komfortsignaturfunktion nicht aktiviert ist (SAK_COMFORT_SIGNATURE = Disabled).~~
SAK_COMFORT_ SIGNATURE	Enabled/ Disabled	Aktivierung/Deaktivierung der Komfortsignaturfunktion im Konnektor Default-Wert = Disabled Die Komfortsignaturfunktion darf nur aktiviert sein, wenn ANCL_TLS_MANDATORY = Enabled und ANCL_CAUT_MANDATORY = Enabled

SAK_COMFORT_ SIGNATURE_MAX	[1 – 250]	Anzahl von Komfortsignaturen, die ohne erneute PIN-Eingabe ausgeführt werden dürfen Default-Wert = 100 Der Parameter ist nur relevant, wenn die Komfortsignaturfunktion aktiviert ist (SAK_COMFORT_SIGNATURE = Enabled).
SAK_COMFORT_ SIGNATURE_TIMER	[1 – 24 h]	Zeitintervall, in dem Komfortsignaturen ohne erneute PIN-Eingabe ausgeführt werden dürfen Der Timer startet mit Eingabe der PIN.QES für die Komfortsignatur. Default-Wert = 6 h Der Parameter ist nur relevant, wenn die Komfortsignaturfunktion aktiviert ist (SAK_COMFORT_SIGNATURE = Enabled).

[<=]

Kommentiert [DS11]: C_10625

5 Anhang D – Übersicht über die verwendeten Versionen

[konsolidierte Übersicht der zu unterstützenden Versionen von SignatureService für den PTV4Plus Komfortsignatur]

Tabelle 29: TAB_KON_688 Version der Schemas aus dem Namensraum des Konnektors

Schemas aus dem Namensraum des Konnektors „http://ws.gematik.de/conn“		
.....		
	XSD Name	SignatureService_V7_5_2.xsd
	XSD Schemaversion	siehe XSD Name
	TargetNamespace	http://ws.gematik.de/conn/SignatureService/v7.5
	XSD Name	SignatureService_V7_4_4.xsd
	XSD Schemaversion	siehe XSD Name
	TargetNamespace	http://ws.gematik.de/conn/SignatureService/v7.4
	XSD Name	SignatureService.xsd
	XSD Schemaversion	7.4.2
	TargetNamespace	http://ws.gematik.de/conn/SignatureService/v7.4

Tabelle 30: TAB_KON_798 Schnittstellenversionen

Pro Dienst mit Operationen an der Außenschnittstelle: WSDLs des Konnektors und verwendete XSDs aus dem Namensraum der gematik http://ws.gematik.de
.....

Signaturdienst (SignatureService)		
	WSDL Name	SignatureService_V7_5_2.wsdl
	WSDL-Version	siehe WSDL Name
	TargetNamespace	http://ws.gematik.de/conn/SignatureService/WSDL/v7.5
	verwendete XSDs	../tel/error/TelematikError.xsd, ConnectorContext.xsd, SignatureService_V7_5_2.xsd
Signaturdienst (SignatureService)		
	WSDL Name	SignatureService_V7_4_2.wsdl
	WSDL-Version	siehe WSDL Name
	TargetNamespace	http://ws.gematik.de/conn/SignatureService/WSDL/v7.4
	verwendete XSDs	../tel/error/TelematikError.xsd, ConnectorContext.xsd, SignatureService.xsd
Signaturdienst (SignatureService)		
	WSDL Name	SignatureService.wsdl
	WSDL-Version	7.4.0
	TargetNamespace	http://ws.gematik.de/conn/SignatureService/WSDL/v7.4
	verwendete XSDs	../tel/error/TelematikError.xsd, ConnectorContext.xsd, SignatureService.xsd

.....

515