gematik
Gesellschaft für Telematikanwendungen der Gesundheitskarte mbH

Einführung der Gesundheitskarte

Begleitdokument

Spezifikation Konnektor OPB2.1

Version:	1.2.0
Revision:	\main\rel_ors2\24
Stand:	18.12.2017
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemSpec_Kon_Begleit]

Dokumentinformationen

Änderungen zur Vorversion

Änderungen zur Vorversion beruhen auf P15.1

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.1.0	05.10.17		freigegeben	gematik
			Einarbeitung von P15.1	
1.2.0	18.12.17		freigegeben	gematik

Inhaltsverzeichnis

1 Einordnung des Dokuments

1.1 Zielsetzung

Die Spezifikation des Konnektors wurde für die Integration der Fachmodule NFDM~~, AdV~~ und AMTS erweitert. Gleichzeitig wurden die Schnittstellen der Technischen Use Cases (TUC) zu den Fachmodulen des Konnektors formal angepasst und vereinheitlicht. Hiervon betroffen sind die Darstellung der Namen und Datentypen der Ein- und Ausgangsparameter dieser TUCs. Die Formalisierung der Schnittstellen dient der Vereinfachung einer Prüfung auf Konsistenz zwischen dem Aufrufer und dem Anbieter der Schnittstelle. Sie ist rein redaktionell und begründet keine funktionalen Änderungen sowie keine Auswirkung auf die Sicherheitsevaluierung. Solche Änderungen sind daher in der Spezifikation des Konnektors nicht markiert.

Ziel des vorliegenden Dokuments ist es, die genannten rein formalen Unterschiede aufzuzeigen. Dazu werden im Kapitel 2 „Überblick über die formalisierten TUC-Schnittstellen“ die Änderungen tabellarisch gegenübergestellt.

1.2 Zielgruppe

Das Dokument richtet sich an Konnektorhersteller.

1.3 Geltungsbereich

Dieses Dokument gilt in Zusammenhang mit [gemSpec_Kon_OPB2.1].

1.4 Abgrenzung des Dokuments

Das Dokument beinhaltet die in Kapitel 1.1 erwähnten Unterschiede zwischen [gemSpec_Kon] und [gemSpec_Kon_OPB2.1].

1.5 Methodik

Die Formalisierung der Schnittstellen wird durch tabellarische Gegenüberstellung der Eingangs- und Ausgangsparameter der TUCs veranschaulicht.

Das Ausgangsdokument ist die Konnektorspezifikation [gemSpec_Kon].

1.5.1 Darstellungsweise

Um eine bessere Übersicht über die funktionalen Änderungen in [gemSpec_Kon_OPB2.1] gegenüber [gemSpec_Kon] zu erhalten und sie von redaktio-

nellen und sonstigen Änderungen zu unterscheiden, die sich nicht auf den Leistungsumfang oder das Verhalten des Konnektors auswirken, wurden nicht alle Änderungen farblich markiert.

Bezogen auf die den Vergleich von [gemSpec_Kon] und [gemSpec_Kon_OPB2.1] gilt:

Änderungen von OPB Release 1.6.4 nach Release 2.1.0 sind gelb markiert.

Änderungen aus den OPB Releases 1.6.4-1, 1.6.4-2 und 1.6.4-3 sind pink markiert.

Änderungen von OPB Release 2.1.0 nach Release 2.1.1 sind grün markiert.

Nicht farblich markiert sind Änderungen:

- redaktioneller Art, die dem besseren Verständnis, der Übersichtlichkeit oder der Vollständigkeit dienen,
- der Eingangs- und Ausgangsdaten der TUCs, die der formalen Beschreibung der Schnittstellen dienen (siehe Festlegungen zur Schreibweise von Eingangs- und Ausgangsdaten [gemSpec_Kon_OPB2.1#1.5.4.2]). Dabei wurde die Anzahl und Art der Eingangs- und Ausgangsparameter nicht verändert, lediglich die Darstellungsweise der Eingangs- und Ausgangsdaten wurde überarbeitet.
- der Ablaufbeschreibungen, die an die neuen Parameternamen angepasst wurden.

Die bei der Definition der Eingangs- und Ausgangsdaten verwendete Syntax ist in [gemSpec_Kon_OPB2.1#1.5.4.2] beschrieben.

2 Überblick über die formalisierten TUC-Schnittstellen

Dieses Kapitel stellt die Eingangs- und Ausgangsdaten der TUCs von [gemSpec_Kon] und [gemSpec_Kon_OPB2.1] gegenüber.

Die Nummerierung der Kapitel folgt derjenigen in [gemSpec_Kon_OPB2.1].

4.1.1 Zugriffsberechtigungsdienst

4.1.1.4.1 TUC_KON_000 „Prüfe Zugriffsberechtigung“

TAB_KON_511 - TUC_KON_000 „Prüfe Zugriffsberechtigung“

OPB1	OPB2.1
Eingangsdaten	
• mandantId	• mandantId
• clientSystemId.	• clientSystemId.
• workplaceId	• workplaceId
• userId (optional)	• userId – *optional*
• ctId (optional)	• ctId – *optional* (Kartenterminalidentifikator)
• cardHandle (optional)	• cardHandle - *optional*
• needCardSession (needCardSession=true; doNot-NeedCardSession=false; default: true; optional; wenn der Parameter leer ist, gilt der Default-Wert) Verwendet der aufrufende TUC eine Kartensitzung ist der Wert true, verwendet er keine Kartensitzung ist der Wert false. Die Berechtigungsprüfung geht im Default-Fall davon aus, dass eine Kartensitzung benötigt wird, und prüft für diesen Fall die Berechtigung mit.	• needCardSession [Boolean] – *optional*; *default: true* („needCardSession“=true; „doNotNeed-CardSession“=false) Dieser Schalter gibt an, ob eine Kartensitzung benötigt wird - true, der aufrufende TUC verwendet eine Kartensitzung - false, der aufrufende TUC verwendet keine Kartensitzung Die Berechtigungsprüfung geht im Default-Fall davon aus, dass eine Kartensitzung benötigt wird, und prüft für diesen Fall die Berechtigung mit.
• allWorkplaces (allWorkplaces=true; allWorkplace=false; default: false; optional; wenn der Parameter leer ist, gilt der Default-Wert) Dieser Parameter muss dann (true) gesetzt werden, wenn die Berechtigungsprüfung	• allWorkplaces [Boolean] – *optional*; *default: false* Dieser Schalter gibt an, ob eine mandantenweite Zugriffsberechtigung gemeint ist. Dieser Parameter muss dann (true) gesetzt werden, wenn die Berechtigungs-

OPB1	OPB2.1
nicht auf die vom angegebenen Arbeitsplatz erreichbaren Kartenterminals beschränkt ist, sondern sich auf alle vom Clientsystem (clientSystemId) und dem Mandant (mandantId) insgesamt erreichbaren Kartenterminals beziehen soll. Ist dieser Schalter gleich true, wird die Berechtigung unabhängig vom Eingangsparameter workplaceId geprüft.	prüfung nicht auf die vom angegebenen Arbeitsplatz erreichbaren Kartenterminals beschränkt ist, sondern sich auf alle vom Clientsystem (clientSystemId) und dem Mandant (mandantId) insgesamt erreichbaren Kartenterminals beziehen soll. Ist dieser Schalter gleich true, wird die Berechtigung unabhängig vom Eingangsparameter workplaceId geprüft.
	• ~~serviceName - *optional*~~ ~~Name des SOAP-Services der aufgerufenen Außenoperation, für die die Prüfung der Zugriffsberechtigung (berechtigter Client) erfolgt.~~ ~~Wird dieser TUC nicht im Kontext einer Außenoperation und deren Parametern sondern im Kontext eines Fachmoduls aufgerufen, bleibt dieser Parameter leer.~~
Ausgangsdaten	
• keine (Autorisierung erteilt)	• keine
• Fehler (Autorisierung nicht erteilt, siehe technische Fehlermeldung)	

Hinweis: „Autorisierung erteilt“ wurde als Nachbedingung eingepflegt

4.1.2 Dokumentvalidierungsdienst

4.1.2.4.1 TUC_KON_080 “Dokument validieren”

TAB_KON_143 - TUC_KON_080 „Dokument validieren“

OPB1	OPB2.1
Eingangsdaten	
• Zu validierendes Dokument.	• documentToBeValidated (Zu validierendes Dokument.)
• Formatangabe für das Dokument (Dokumentformat)	• documentFormat (mögliche Werte siehe Definition `Alle_DocFormate;` Formatangabe für das Dokument)
	Optional für XML-Dokumente: • signaturePolicyIdentifier – *optional/nur für XML-Formate gemäß einer referenzierten* Signaturrichtlinie

OPB1	OPB2.1
	(URI identifiziert die Signaturrichtlinie)
Ausgangsdaten	
• Prüfprotokoll (DocumentValidation) Die Ausprägung dieses Konnektor internen Parameters erfolgt herstellerspezifisch.	• documentValidationProtocol (Prüfprotokoll) Die Ausprägung dieses Konnektor internen Parameters erfolgt herstellerspezifisch.

4.1.3 Dienstverzeichnisdienst

4.1.4.3.1 TUC_KON_041 „Einbringen der Endpunktinformationen während der Bootup-Phase“

TAB_KON_519 - TUC_KON_041 "Einbringen der Endpunktinformationen während der Bootup-Phase “

OPB1	OPB2.1
Eingangsdaten	
Die Eingangsdaten sind gemäß dem Ausgabeschema „Serviceinformation.xsd“ zu formatieren. Eine Beschreibung des Schemas befindet sich in TAB_KON_518.	• serviceInformation (Ein XML-Dokument mit dem Wurzelelement „ServiceInformation“ gemäß dem Schema „Serviceinformation.xsd“. Eine Beschreibung des Schemas befindet sich in TAB_KON_518.)
Ausgangsdaten	
Keine	• Keine

4.1.4 Kartenterminaldienst

4.1.4.3.1 TUC_KON_050 „Beginne Kartenterminalsitzung

TAB_KON_039 - TUC_KON_050 „Beginne Kartenterminalsitzung“

OPB1	OPB2.1
Eingangsdaten	
• CtID	• ctId
• Benutzerrolle (gültig sind: User und Admin)	• role (Benutzerrolle; gültig sind: User und Admin)
Ausgangsdaten	

Keine	keine

4.1.4.4.1 TUC_KON_051 „Mit Anwender über Kartenterminal interagieren“

TAB_KON_112 - TUC_KON_051 „Mit Anwender über Kartenterminal interagieren“

OPB1	**OPB2.1**
Eingangsdaten	
• CtID	• ctId (Kartenterminalidentifikator)
• Data (Text zur Darstellung am KT, Länge durch KT begrenzt); optional bei Mode = OutputErase, sonst mandatory	• displayMessage – *optional/nicht erforderlich bei opmode= OutputErase, sonst mandatory* (Text zur Darstellung am KT, Länge durch KT begrenzt)
• Mode (Input, OutputWait, OutputConfirm, OutputKeep, OutputErase)	• opMode [KtOutputMode] (Kommando-Modus)
• InputLength (nur bei Mode=Input, 00 für „beliebig“ lang)	• inputLength – *optional/nur bei opMode=Input* (erwartete Eingabelänge, 0 für „beliebig“ lang)
• WaitTimer (in Sekunden, nur bei Mode=OutputWait)	• waitTimer – *optional/nur bei opMode=OutputWait* (Wartezeit bis zur ersten Eingabe in Sekunden)
Ausgangsdaten	
• Bei Input und OutputConfirm: Nutzertastendruck OK/ABBRUCH	• opResult [OK | ABBRUCH] – *optional/verpflichtend, wenn opMode=Input oder opMode=OutputConfirm* (Nutzertastendruck)
• Bei Input: Zifferneingabe des Benutzers	• inputData – *optional/nur bei opMode = Input* (Zifferneingabe des Benutzers)

4.1.4.4.2 TUC_KON_056 „Karte anfordern“

TAB_KON_723 - TUC_KON_056 „Karte anfordern“

OPB1	OPB2.1
Eingangsdaten	
• CtID	• ctId (Kartenterminalidentifikator)
• SlotID	• slotId (Nummer des Kartenslots)
• CardType (optional)	• cardType – *optional*
• DisplayMsg (optional, Text zur Darstellung am KT, Länge durch KT begrenzt)	• displayMessage – *optional* (Text zur Darstellung am Kartenterminal, Länge durch KT begrenzt)
• TimeOut (in Sekunden)	• timeOut (Wartezeit in Sekunden)
Ausgangsdaten	
• Informationsobjekt der Karte	• cardObject (Informationsobjekt der Karte)

4.1.4.4.3 TUC_KON_057 „Karte auswerfen“

TAB_KON_725 - TUC_KON_057 „Karte auswerfen“

OPB1	OPB2.1
Eingangsdaten	
• CtID	• ctId (Kartenterminalidentifikator)
• SlotID	• slotId (Nummer des Kartenslots)
• DisplayMsg (optional, Text zur Darstellung am KT, Länge durch KT begrenzt)	• displayMessage – *optional* (Text zur Darstellung am Kartenterminal, Länge durch KT begrenzt)
• TimeOut (in Sekunden)	• timeOut (Wartezeit in Sekunden)
Ausgangsdaten	
keine	keine

4.1.5 Kartendienst

4.1.5.3.1 TUC_KON_001 „Karte öffnen“

TAB_KON_734 - TUC_KON_001 „Karte öffnen“

OPB1	OPB2.1
Eingangsdaten	
• CtID	• ctId (Kartenterminalidentifikator)
• SlotNo	• slotId (Nummer des Kartenslots)
Ausgangsdaten	
keine	keine

4.1.5.4.1 TUC_KON_026 „Liefere CardSession“

TAB_KON_735 - TUC_KON_026 „Liefere CardSession“

OPB1	OPB2.1
Eingangsdaten	
• MandantId	• mandantId
• clientSystemId	• clientSystemId
• cardHandle	• cardHandle
• userId (nur für CardType = HBAx, da aber verpflichtend)	• userId – *optional/verpflichtend, wenn cardType = HBAx*
Ausgangsdaten	
• CardSession	• cardSession

4.1.5.4.2 TUC_KON_012 „PIN verifizieren“

TAB_KON_087 - TUC_KON_012 „PIN verifizieren“

OPB1	OPB2.1
Eingangsdaten	
• CardSession (Kartensitzung der Karte, deren PIN verifiziert werden soll)	• cardSession (Kartensitzung der Karte, deren PIN verifiziert werden soll)

OPB1	OPB2.1
• workplaceID	• workplaceID
• PinRef (laut Kartenspec)	• pinRef (Referenz auf die zu verifizierende PIN, Tupel aus {applicationIdentifier, pwdIdentifier} gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps)
• AppName (Name der zugreifenden Fachanwendung, z. B. „VSDM“, max. 9 Zeichen)	• actionName – *optional/verpflichtend, wenn cardType = eGK* (Zeichenkette, max. 32 Zeichen bzw. 22 Zeichen PIN.AMTS_REP, mit dem Namen der zugreifenden Fachanwendung bzw. des zu nutzenden Datenobjekts und der Zugriffsart, die mit dieser PIN freigeschaltet werden soll, z. B. für MRPIN.NFD: actionName = „Notfalldaten schreiben“; Positionen in der Zeichenkette, an denen ein Zeilenumbruch bei der Ausgabe am Kartenterminal erlaubt ist, werden mit `0x0B` gekennzeichnet. `0x0B` zählt bei der Länge der Zeichenkette nicht.)
• VerificationTyp (Art der PIN-Verifikation): o Mandatorisch: PIN wird immer verifiziert. o Sitzung: PIN wird nicht erneut verifiziert, falls dies für die CardSession zuvor bereits geschehen ist und der dadurch erreichte Sicherheitszustand nicht zurückgesetzt wurde.	• verificationType [Mandatorisch | Sitzung] (Art der PIN-Verifikation: o Mandatorisch: PIN wird immer verifiziert. o Sitzung: PIN wird nicht erneut verifiziert, falls dies für die cardSession zuvor bereits geschehen ist und der dadurch erreichte Sicherheitszustand nicht zurückgesetzt wurde.)
Ausgangsdaten	
• Ergebnis der PIN-Verifikation: [OK/REJECTED/BLOCKED/ERROR]	• pinResult [PinResult] (Ergebnis der PIN-Verifikation)
• LeftTries (Anzahl der verbleibenden Versuche für die Verifikation der PIN)	• leftTries – *optional/verpflichtend, wenn pinResult = REJECTED* (Anzahl der verbleibenden Versuche)

4.1.5.4.3 TUC_KON_019 „PIN ändern“

TAB_KON_736 - TUC_KON_019 „PIN ändern“

OPB1	OPB2.1
Eingangsdaten	
• CardSession	• cardSession
• workplaceID	• workplaceID

	(Arbeitsplatz-Identifikator)
• PinRef (laut Kartenspec.)	• pinRef (Referenz auf die zu ändernde PIN, Tupel aus {applicationIdentifier, pwdIdentifier} gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps)
• Sup.CardSession (der Karte, die für die Card-to-Card-Authentisierung bei Änderung der PIN einer eGK der Generation 1+ verwendet werden soll) (optional)	• sourceCardSession – *optional/verpflichtend, wenn C2C erforderlich ist* (CardSession der Karte, die für die Card-to-Card-Authentisierung bei Änderung der PIN einer eGK der Generation 1+ verwendet werden soll.)
Ausgangsdaten	
• Result (pinStatus [OK/REJECTED/BLOCKED/ERROR]) (Ergebnis der PIN-Verifikation)	• pinResult [PinResult] (Ergebnis der PIN-Verifikation)
• leftTries – optional (verpflichtend wenn pinStatus = REJECTED) (verbleibende Versuche)	• leftTries – *optional/verpflichtend, wenn pinStatus = REJECTED* (verbleibende Versuche)

4.1.5.4.4 TUC_KON_021 „PIN entsperren“

TAB_KON_236 - TUC_KON_021 „PIN entsperren“

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession (der Karte, deren PIN entsperrt werden soll)	• cardSession CardSession der Karte, deren PIN entsperrt werden soll)
• workplaceID	• workplaceID
• PinRef (nach Kartenspec)	• pinRef (Referenz auf die zu entsperrende PIN, Tupel aus {applicationIdentifier, pwdIdentifier} gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps)
• setNewPin (true/false) - Angabe, ob eine neue PIN gesetzt oder die aktuelle weiterverwendet werden soll. Default = false	• setNewPin (true/false) - Angabe, ob eine neue PIN gesetzt oder die aktuelle weiterverwendet werden soll. Default = false
• Sup.CardSession (der Karte, die für die Card-to-Card-Authentisierung bei Änderung der PIN einer eGK der Generation 1+ verwendet werden soll) (optional)	• sourceCardSession - *optional // wenn eGK G1+* (CardSession der Karte, die für die Card-to-Card-Authentisierung bei Entsperrung der PIN einer eGK der Generation 1+ verwendet werden soll)
Ausgangsdaten	
• Result (pukStatus [OK/REJECTED/BLOCKED/ERROR])	• result [PukResult]) (Ergebnis der PIN-Entsperrung durch

OPB1	OPB2.1
(Ergebnis der PIN-Entsperrung durch PUK-Eingabe)	PUK-Eingabe)
• • leftTries – optional/wenn pukStatus = REJECTED (verbleibende Versuche des PUKs)	• leftTries – *optional/verpflichtend, wenn pukStatus = REJECTED* (verbleibende Versuche des PUKs)

4.1.5.4.5 TUC_KON_022 „Liefere PIN-Status“

TAB_KON_532 – TUC_KON_022 „Liefere PIN-Status“

OPB1	OPB2.1
Eingangsdaten	
• CardSession	• cardSession
• PinRef (laut Kartenspec.)	• pinRef (Pin-Referenz der angefragten PIN, Tupel aus {applicationIdentifier, pwdIdentifier} gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps.)
Ausgangsdaten	
• PINStatus (verifiziert, verifizierbar, Transport-PIN, Leer-PIN, gesperrt)	• pinStatus [PinStatus]
• LeftTries (Anzahl der verbleibenden Versuche für die Verifikation der PIN)	• leftTries – *optional // verpflichtend, wenn pinStatus = VERIFYABLE* (Anzahl der verbleibenden Versuche für die Verifikation der PIN)

4.1.5.4.7 TUC_KON_023 „Karte reservieren“

TAB_KON_533 - TUC_KON_023 „Karte reservieren“

OPB1	OPB2.1
Eingangsdaten	
• CardSession	• cardSession
• DoLock (Ja/Nein)	• doLock [Boolean] (Zielzustand der Karte; true = reserviert, false = freigegeben)
Ausgangsdaten	
Keine	Keine

4.1.5.4.8 TUC_KON_005 „Card-to-Card authentisieren“

TAB_KON_096 - TUC_KON_005 „Card-to-Card authentisieren“

OPB1	**OPB2.1**
Eingangsdaten	
• Source_CardSession (Quellkarte)	• sourceCardSession (Quellkarte)
• Target_CardSession (Zielkarte)	• targetCardSession (Zielkarte)
AuthMode (gemäß Tabelle 68)	• authMode (gemäß Tabelle 70)
Ausgangsdaten	
Keine	Keine

4.1.5.4.9 TUC_KON_202 „LeseDatei“

TAB_KON_218 – TUC_KON_202 „LeseDatei“

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession	• cardSession
• FileIdentifier	• fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei)
• SFI (Short File Identifier, optional)	• sfid – *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei)
• Verzeichnis der Karte, in dem sich die Datei befindet	• folder (Verzeichnis auf der Karte, in dem sich die Datei befindet)
• Offset- und Längenangaben, um den Zugriff auf Teile einer Datei einzuschränken (optional)	• offset – *optional/nur verwendbar, wenn fileIdentifier angegeben ist* (Startposition innerhalb der Datei)
•	• length – *optional* (Längenangabe, um den Zugriff auf Teile einer Datei einzuschränken)
Ausgangsdaten	
Gelesene Daten	• content (Gelesene Daten)

4.1.5.4.10 TUC_KON_203 „SchreibeDatei“

TAB_KON_219 – TUC_KON_203 „SchreibeDatei“

OPB1	OPB2.1
Eingangsdaten	
• CardSession	• cardSession
• FileIdentifier	• fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei)
• SFI (Short File Identifier, optional)	• sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei)
• Verzeichnis der Karte, in dem sich die Datei befindet	• folder (Verzeichnis auf der Karte, in dem sich die Datei befindet)
• Offset- und Längenangaben, um den Zugriff auf Teile einer Datei einzuschränken (optional)	• offset– *optional* (Startposition innerhalb der Datei, default: 0)
• Zu schreibende Daten	• length – *optional* (Längenangabe, um den Zugriff auf Teile einer Datei einzuschränken; default: alles ab offset)
Ausgangsdaten	
Keine	Keine

4.1.5.4.11 TUC_KON_209 „LeseRecord“

TAB_KON_538 – TUC_KON_209 „LeseRecord“

OPB1	OPB2.1
Eingangsdaten	
• CardSession	• cardSession
• FileIdentifier	• fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei)
• SFI (Short File Identifier, optional)	• sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei)
• Verzeichnis der Karte, in dem sich die Datei befindet	• folder (Verzeichnis auf der Karte, in dem sich die Datei befindet)

• RecordNummer	• recordNumber
Ausgangsdaten	
• Inhalt des Records	• content (Inhalt des Records)

4.1.5.4.12 TUC_KON_210 „SchreibeRecord“

TAB_KON_224 – TUC_KON_210 „SchreibeRecord “

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession	• cardSession
• FileIdentifier	• fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei)
• SFI (Short File Identifier, optional)	• sfid – *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu bearbeitenden Datei)
• Verzeichnis der Karte, in dem sich die Datei befindet	• folder (Verzeichnis auf der Karte, in dem sich die Datei befindet)
• Recordnummer	• recordNumber
• Zu schreibende Daten	• dataToBeWritten (Zu schreibende Daten)
Ausgangsdaten	
keine	keine

4.1.5.4.13 TUC_KON_214 „FügeHinzuRecord“

TAB_KON_228 – TUC_KON_214 „FügeHinzuRecord“

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession	• cardSession
• FileIdentifier	• fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei)
• SFI (Short File Identifier, optional)	• sfid– *optional/verpflichtend, wenn kein*

	fileIdentifier angegeben ist (Short File Identifier der zu bearbeitenden Datei)
• Verzeichnis der Karte, in dem sich die Datei befindet	• folder (Verzeichnis auf der Karte, in dem sich die Datei befindet)
• Zu schreibende Daten	• dataToBeWritten (Zu schreibende Daten)
Ausgangsdaten	
keine	keine

4.1.5.4.16 TUC_KON_215 „SucheRecord“

TAB_KON_229 – TUC_KON_215 „SucheRecord “

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession	• cardSession
• FileIdentifier	• fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu bearbeitenden Datei)
• SFI (Short File Identifier, optional)	• sfid – *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu bearbeitenden Datei)
• Verzeichnis der Karte, in dem sich die Datei befindet	• folder (Verzeichnis auf der Karte, in dem sich die Datei befindet)
• SuchMuster	• pattern (SuchMuster)
• Recordnummer, bei der Suche beginnen soll (optional)	• recordNumber – optional; default = 1 (Recordnummer, bei der Suche beginnen soll)
Ausgangsdaten	
Liste: Nummern der Records, die dem SuchMuster entsprechen	• numbersFound (Liste: Nummern der Records, die dem SuchMuster entsprechen)

4.1.5.4.17 TUC_KON_018 „eGK-Sperrung prüfen“

TAB_KON_110 - TUC_KON_018 „eGK-Sperrung prüfen “

OPB1	OPB2.1
Eingangsdaten	
• CardSession	• cardSession
	• checkHcaOnly [Boolean] - *optional; default = false* (Prüfung auf die Frage beschränken, ob auf DF.HCA zugegriffen werden kann)
Ausgangsdaten	
Karte gesperrt: ja/nein Status: • DF.HCA gesperrt: ja/nein • Ergebnis der Offline-Prüfung des C.CH.AUT-Zertifikats: gültig/ungültig • Sperrstatus des C.CH.AUT-Zertifikats: gut/gesperrt/nicht ermittelbar	• Karte gesperrt: true | false • Status – *optional // wenn checkHcaOnly = false* o DF.HCA gesperrt: true | false o Ergebnis der Offline-Prüfung des C.CH.AUT-Zertifikats: gültig | ungültig o Sperrstatus des C.CH.AUT-Zertifikats: gut | gesperrt | nicht ermittelbar

4.1.5.4.18 TUC_KON_006 „Datenzugriffsaudit eGK schreiben“

TAB_KON_108 - TUC_KON_006 „Datenzugriffsaudit eGK schreiben “

OPB1	OPB2.1
Eingangsdaten	
• CardSession (einer eGK)	• cardSession (CardSession einer eGK)
• Sup.CardSession (HBA/SMC, die für den eGK-Zugriff verwendet wird)	• sourceCardSession (HBA/SMC-B, der/die für den eGK-Zugriff verwendet wird)
• DATA.TYP (siehe gem_Spec_Karten_Fach_TIP#4.1 – Tabelle 11: Tab_Karten_Fach_TIP_010 ab_StrukturEF.Logging – Struktur der Rekords der Datei EF.Logging]	• dataType (zugreifende Anwendung, siehe [gem_Spec_Karten_Fach_TIP#4.1 – Tabelle 11: Tab_Karten_Fach_TIP_010 _StrukturEF.Logging – Struktur der Rekords der Datei EF.Logging]
• Type of Access (siehe [gemSpec_Karten_Fach_TIP#4.1] – Tabelle 11: Tab_Karten_Fach_TIP_010 ab_StrukturEF.Logging – Struktur der Rekords der Datei EF.Logging)	• accessType (Zugriffsart, siehe ebenda)

Ausgangsdaten	
Keine	Keine

4.1.5.4.19 TUC_KON_218 „Signiere“

TAB_KON_231 – TUC_KON_218 „Signiere “

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession	• cardSession
• PinRef	• pinRef (PIN-Referenz, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps)
• KeyRef	• keyRef (Referenz auf den privaten Schlüssel, mit dem signiert werden soll, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps)
• AlgorithmusID	• algorithmusId (einer der laut Objektspezifikation für diesen Schlüssel zulässigen algorithmIdentifier)
• DTBS (Zu signierende Daten)	• dataToBeSigned (Zu signierende Daten, Hashwert)
Ausgangsdaten	
CHIFFRAT (Signierte Daten)	• chiffrat (Signatur)

4.1.5.4.20 TUC_KON_219 „Entschlüssele“

TAB_KON_232 – TUC_KON_219 „Entschlüssele“

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession	• cardSession
• PinRef	• pinRef (Referenz auf die PIN, mit der der Entschlüsselungsschlüssel freigeschaltet werden kann, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps)
• KeyRef	• keyRef

	(Referenz auf den privaten Schlüssel, mit dem entschlüsselt werden soll, gemäß der Objektsystem-Spezifikation des jeweiligen Kartentyps.)
• AlgorithmusID	• algorithmusId (einer der für diesen Schlüssel zulässigen algorithmIdentifier)
• Zu entschlüsselnde Daten (Chiffrat)	• encryptedData (Zu entschlüsselnde Daten, Chiffrat)
Ausgangsdaten	
Entschlüsselte Daten	• plainData (Entschlüsselte Daten)

4.1.5.4.21 TUC_KON_200 „SendeAPDU“

TAB_KON_215 TUC_KON_200 „SendeAPDU“

OPB1	OPB2.1
Eingangsdaten	
• CARDSESSION, alternativ CtID	• cardSession – *optional/verpflichtend, wenn die APDU an die Karte gerichtet ist*
•	• ctId – *optional/verpflichtend, wenn die APDU an das Kartenterminal gerichtet ist* (Kartenterminalidentifikator für Kommandos an das Kartenterminal)
• APDU Parameter{CLA, Ins, P1,P2, Data (optional) Le(optional)	• commandAPDU (versandfertige APDU (Bytefolge), in dem die Parameter {CLA, INS, P1,P2, Data (*optional*) Le (*optional*) } gesetzt sind.)
Ausgangsdaten	
Antwort (Response-APDU) der Chipkarte	• responseAPDU (Antwort der Chipkarte oder des Kartenterminals, Bytefolge)

4.1.5.4.22 TUC_KON_024 „Karte zurücksetzen“

TAB_KON_737 - TUC_KON_024 „Karte zurücksetzen“

OPB1	OPB2.1
Eingangsdaten	
• CtID	• ctId – *optional/verpflichtend, wenn keine cardSession angegeben ist* (Kartenterminal-Identifikator)

• SlotNo	• slotId – *optional/verpflichtend, wenn keine cardSession angegeben ist* (Nummer des Slots, in dem die Karte steckt)
Alternativ: • CardSession	• cardSession – *optional/verpflichtend, wenn ctId und slotId nicht angegeben sind* (Angabe der CardSession alternativ zur Angabe von ctId und slotId)
Ausgangsdaten	
Keine	Keine

4.1.5.4.23 TUC_KON_216 „LeseZertifikat“

TAB_KON_230 – TUC_KON_216 „LeseZertifikat“

OPB1	**OPB2.1**
Eingangsdaten	
• CardSession	• cardSession
• FileIdentifier	• fileIdentifier – *optional/verpflichtend, wenn kein sfid angegeben ist* (FID der zu lesenden Datei)
•	• sfid– *optional/verpflichtend, wenn kein fileIdentifier angegeben ist* (Short File Identifier der zu lesenden Datei)
•	• folder (Verzeichnis auf der Karte, in dem sich das Zertifikat befindet)
Ausgangsdaten	
• Zertifikat	• certificate (gelesenes Zertifikat)

4.1.6 Systeminformationsdienst

4.1.6.4.1 TUC_KON_256 „Systemereignis absetzen“

TAB_KON_556 - TUC_KON_256 „Systemereignis absetzen“

OPB1	**OPB2.1**
Eingangsdaten	
• Event (zu versendendes Ereignis)	Attribute des zu versendenden Ereignisses:

OPB1	OPB2.1
o Topic	• topic (Name des Ereignisses)
o Typ (Op = Operation, Sec = Security, Infra = Infrastructur)	• eventType [EventType] (Wenn statt eines EventType ein ErrorType übergeben wird, so wird der EventType daraus abgeleitet. Typ des Events: Op = Operation, Sec = Security, Infra = Infrastructure)
o Schwere (Info = Information, Warn = Warning, Err = Error, Fatal)	• severity [EventSeverity] (Schwere des Ereignisses: Info = Information, Warn = Warning, Err = Error, Fatal)
o Parameter	• parameters (weitere Parameter als key-value-Paare)
	Arbeitsanweisungen:
• Schalter „Schreibe Protokolleintrag“ (doLog/noLog; optional; default = doLog)	• doLog [Boolean] – *optional; default = true* (Schalter „Schreibe Protokolleintrag“)
• Schalter „An Clientsysteme versenden“ (doDisp/noDisp; optional; default = doDisp)	• doDisp [Boolean] – *optional; default = true* (Schalter „An Clientsysteme versenden“)
Ausgangsdaten	
Keine	Keine

4.1.6.4.2 TUC_KON_252 „Liefere KT_Liste“

TAB_KON_558 - TUC_KON_252 „Liefere KT_Liste“

OPB1	OPB2.1
Eingangsdaten	
• Arbeitsplatz ID (Optional)	• workplaceId – *optional* (Arbeitsplatz ID)
• Clientsystem ID	• clientSystemId (Clientystem ID)
• Mandanten ID	• mandantId (Mandanten ID)
Ausgangsdaten	
• Liste der Kartenterminals, die den angegebe-	• cardTerminals

OPB1	OPB2.1
nen Arbeitsplätzen, Mandanten und Clientsystemen zugeordnet sind bzw. auf die diese zugreifen dürfen (siehe Zugriffsberechtigungsdienst), sowie deren aktuelle Verfügbarkeit. Verfügbarkeit bedeutet im Falle eines eHealth-Kartenterminals, dass der Konnektor eine Verbindung zum Kartenterminal aktuell hält.	(Liste der Kartenterminals, die den angegebenen Arbeitsplätzen, Mandanten und Clientsystemen zugeordnet sind bzw. auf die diese zugreifen dürfen (siehe Zugriffsberechtigungsdienst), sowie deren aktuelle Verfügbarkeit. Verfügbarkeit bedeutet im Falle eines eHealth-Kartenterminals, dass der Konnektor eine Verbindung zum Kartenterminal aktuell hält.)

4.1.6.4.3 TUC_KON_253 „Liefere Karten_Liste“

TAB_KON_559 - TUC_KON_253 „Liefere Karten_Liste“

OPB1	OPB2.1
Eingangsdaten	
• Arbeitsplatz ID (Optional)	• workplaceId – *optional* (Arbeitsplatz ID)
• Clientsystem ID	• clientSystemId (Clientystem ID)
• Kartenterminal-ID (Optional)	• cardTerminalId - *optional; verpflichtend, wenn slotId übergeben wird* (Kartenterminalidentifikator)
• Slot ID	• slotId – *optional* (Nummer des Slots, beginnend bei 1)
• Mandanten ID	• mandantId (Mandanten ID)
• CardType (Optional)	• cardType – *optional* (Kartentyp gemäß Tabelle TAB_KON_500)
Ausgangsdaten	
• Liste der gesteckten Karten, auf die der Mandant und das Clientsystem von dem Arbeitsplatz aus zugreifen dürfen (siehe Zugriffsberechtigungsdienst). Falls Kartenterminal angegeben, nur Karten die im entsprechenden Kartenterminal stecken.	• cards (Liste der gesteckten Karten einschließlich der Informationen für CARD:card, auf die der Mandant und das Clientsystem von dem Arbeitsplatz aus zugreifen dürfen (siehe Zugriffsberechtigungsdienst). Wird workplaceId nicht übergeben, so werden alle vom Clientsystem und dem Mandant erreichbaren Kartenterminals in die Liste aufgenommen. Die Eingangsdaten dienen als Filter, welche Karten in cards zurückgegeben werden.

OPB1	OPB2.1
	Beispiel: Falls cardTerminalId angegeben ist, werden nur Karten in die Liste aufgenommen, die im entsprechenden Kartenterminal stecken.)

4.1.6.4.4 TUC_KON_254 „Liefere Ressourcendetails“

TAB_KON_561 - TUC_KON_254 „Liefere Ressourcendetails“

OPB1	OPB2.1
Eingangsdaten	
• Clientsystem ID	• clientSystemId (Clientsystem ID)
• Mandanten ID	• mandantId (Mandanten ID)
• Arbeitsplatz ID (Optional)	• workplaceId – *optional* (Arbeitsplatz ID)
• Kartenterminal-ID (Optional)	• cardTerminalId – *optional* (Kartenterminal ID)
• Kartenslot-ID (Optional und nur in Kombination mit Kartenterminal-ID)	• slotId – *optional/zulässig nur, wenn auch cardTerminalId angegeben ist* (Kartenslot-Nummer)
• CardHandle (Optional)	• cardHandle – *optional*
• Iccsn (Optional)	• iccsn – *optional*
Ausgangsdaten	
• Informationsobjekt einer Ressource (Kartenterminal, Karte, HSM)	• resource (Informationsobjekt einer Ressource (Kartenterminal, Karte, HSM))

4.1.7 Verschlüsselungsdienst

4.1.7.4.1 TUC_KON_070 "Daten hybrid verschlüsseln"

TAB_KON_739 - TUC_KON_070 „Daten hybrid verschlüsseln"

OPB1	OPB2.1
Eingangsdaten	
• Zu verschlüsselndes Dokument (Document)	• documentToBeEncrypted (Zu verschlüsselndes Dokument)
• X.509v3-Zertifikate oder öffentliche Schlüssel (EncryptionCertificates or EncryptionKeys) Unterstützte Karten sind SM-B, HBAx und eGK.	• encryptionCertificates – *optional/entfällt, wenn encryptionKeys übergeben wird* (X.509v3-Zertifikate)
	• encryptionKeys – *optional/entfällt, wenn encryptionCertificates übergeben wird* (öffentliche Schlüssel; unterstützte Karten sind SM-B, HBAx und eGK)
• Verschlüsselungsverfahren (EncryptionType) Angabe zum einzusetzenden Verschlüsselungsverfahren (CMS, XMLEnc oder S/MIME).	• encryptionType [EncryptionType] (Angaben zum einzusetzenden Verschlüsselungsverfahren (CMS, XMLEnc oder S/MIME)).
• CardSession (Kartensitzung) und Zertifikatsreferenz (falls ein Zertifikat von einer Karte gelesen werden soll) Unterstützte Karten sind SM-B, HBAx und eGK.	• cardSession – *optional/verpflichtend, wenn ein Zertifikat von einer Karte gelesen werden soll* (Kartensitzung; unterstützte Karten sind SM-B, HBAx und eGK.)
	• certificateReference – *optional/verpflichtend, wenn ein Zertifikat von einer Karte gelesen werden soll* (Zertifikatsreferenz; unterstützte Karten sind SM-B, HBAx und eGK).
Bei Verschlüsselung von XML-Dokumenten mit XMLEnc	
• Festlegung der zu verschlüsselnden Teile des Dokumentes durch Spezifikation eines XPath-Ausdruckes (XML-Elements).	• xmlElements – *optional/verpflichtend, wenn encryptionType = XMLEnc* (Festlegung der zu verschlüsselnden Teile des Dokumentes durch Spezifikation eines Xpath-Ausdruckes (XML-Elements).
• Angabe, ob die KeyInfo in das XML-Dokument eingebettet oder separat an den Aufrufer zurückgegeben werden soll	• keyInfoMode [embedded | separate] – *optional/verpflichtend, wenn encryptionType = XMLEnc* (Angabe, ob die KeyInfo in das XML-Dokument eingebettet oder separat an

OPB1	OPB2.1
	den Aufrufer zurückgegeben werden soll)
Ausgangsdaten	
• Verschlüsseltes Dokument	• encryptedDocument (Verschlüsseltes Dokument)
• Verschlüsselte symmetrische Schlüssel (wenn diese nicht im verschlüsselten Dokument enthalten sind)	• encryptedKeys – *optional/verpflichtend, wenn diese nicht im verschlüsselten Dokument enthalten sind* (Verschlüsselte symmetrische Schlüssel)
• OCSP-Checked (True/False, default=True)	• ocspChecked [Boolean] ~~– optional; default = true~~ (Ergebnis des OCSP-Checks)
Bei Verschlüsselung von XML-Dokumenten mit XMLEnc:	
• KeyInfo (falls nicht ins Dokument eingebettet)	• keyInfo – *optional/verpflichtend, wenn encryptionType = XMLEnc und keyInfoMode = separate* (KeyInfo, falls nicht ins Dokument eingebettet)

4.1.7.4.2 TUC_KON_071 "Daten hybrid entschlüsseln"

TAB_KON_140 - TUC_KON_071 „Daten hybrid entschlüsseln"

OPB1	OPB2.1
Eingangsdaten	
• Zu entschlüsselndes Dokument (EncryptedDocument)	• encryptedDocument (Zu entschlüsselndes Dokument)
• CardSession (Kartensitzung) mit Referenz auf den privaten Schlüssel (KeyReference) Unterstützt werden SM-B, HBAx und eGK mit der jeweiligen C.ENC-KeyReference.	• cardSession (Kartensitzung; unterstützt werden SM-B, HBAx und eGK mit der jeweiligen C.ENC-KeyReference.
	• privateKeyReference (Referenz auf den privaten Schlüssel; unterstützt werden SM-B, HBAx und eGK mit der jeweiligen C.ENC-KeyReference).
• Verschlüsselungszertifikat bzw. eine Referenz auf das Zertifikat auf obiger Karte passend zur Schlüsselreferenz (optional).	• encryptionCertificate – *optional* (Verschlüsselungszertifikat passend zur Schlüsselreferenz).
	• encryptionCertificateReference – *optional* (Referenz auf das Zertifikat auf obiger

OPB1	OPB2.1
	Karte passend zur Schlüsselreferenz).
• Hybrid verschlüsselter symmetrischer Schlüssel (optional, falls nicht in EncryptedDocument enthalten)	• encryptedKey – *optional, falls nicht in encryptedDocument enthalten* (asymmetrisch verschlüsselter symmetrischer Schlüssel)
Bei XML-Dokumenten:	•
• Angabe der zu entschlüsselnden Teile des XML-Dokuments (XmlElements)	• xmlElements – *optional/verpflichtend, wenn encryptionType = XMLEnc* (bei XML-Dokumenten Angabe der zu entschlüsselnden Teile des XML-Dokuments)
Ausgangsdaten	
Unverschlüsseltes Dokument Bei XML-Dokumenten: Das EncryptedData-Element ist durch das entschlüsselte ersetzt.	• plainDocument (Unverschlüsseltes Dokument. Bei XML-Dokumenten: Das EncryptedData-Element ist durch das entschlüsselte ersetzt.)

4.1.7.4.3 TUC_KON_072 "Daten symmetrisch verschlüsseln"

TAB_KON_741 - TUC_KON_072 „Daten symmetrisch verschlüsseln"

OPB1	OPB2.1
Eingangsdaten	
• Zu verschlüsselndes Dokument.	• documentToBeEncrypted (Zu verschlüsselndes Dokument.)
• Symmetrischer Schlüssel (optional)	• symmetricKey – *optional* (zu verwendender symmetrischer Schlüssel)
Ausgangsdaten	
• Verschlüsseltes Dokument	• encryptedDocument (Verschlüsseltes Dokument)
• Erzeugter symmetrischer Schlüssel (optional)	• symmetricKey – *optional/verpflichtend, wenn Schlüssel durch den TUC erzeugt wurde* (erzeugter symmetrischer Schlüssel)

4.1.7.4.4 TUC_KON_073 "Daten symmetrisch entschlüsseln"

TAB_KON_ TAB_KON_743 - TUC_KON_073 „Daten symmetrisch entschlüsseln"

OPB1	OPB2.1
Eingangsdaten	
• Verschlüsseltes Dokument	• encryptedDocument (Verschlüsseltes Dokument)
• Symmetrischer Schlüssel	• symmetricKey (zu verwendender symmetrischer Schlüssel)
Ausgangsdaten	
• Entschlüsseltes Dokument	• plainDocument (Entschlüsseltes Dokument)

4.1.8 Signaturdienst

4.1.8.3.1 TUC_KON_155 "Dokumente zur Signatur vorbereiten"

TAB_KON_748 - TUC_KON_155 „Dokumente zur Signatur vorbereiten"

OPB1	OPB2.1
Eingangsdaten	
• Signaturart (QES/nonQES)	• signatureMode (Signaturart: QES | nonQES)
• Zu signierendes Dokument bzw. zu signierende Dokumente und pro Dokument: o Formatangabe für das zu signierende Dokument	• documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente) und pro Dokument: o documentFormat (Formatangabe für das zu signierende Dokument)
• weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur (siehe Operation SignDocument, Parameter dss:OptionalInputs)	• optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur (siehe Operation SignDocument, Parameter dss:OptionalInputs), darin u.a. o signatureType (URI für den Signaturtyp XML-, CMS-, S/MIME-o PDF-Signatur)
• Signaturzertifikat	• certificate (Signaturzertifikat)
• ocspResponses – optional	• ocspResponses – *optional*

OPB1	OPB2.1
(Liste der OCSP-Responses, die bei der Signaturerstellung in die Signatur eingebettet werden.)	(Liste der OCSP-Responses, die bei der Signaturerstellung in die Signatur eingebettet werden.)
Ausgangsdaten	
• Aufbereitetes zu signierendes Dokument bzw. aufbereitete zu signierende Dokumente	• preProcessedDocuments (Aufbereitetes zu signierendes Dokument bzw. aufbereitete zu signierende Dokumente)

4.1.8.4.1 TUC_KON_160 "Dokumente nonQES signieren"

TAB_KON_753 - TUC_KON_160 „Dokumente nonQES signieren"

OPB1	OPB2.1
Eingangsdaten	
• CardSession (SM-B, HBAx oder bei Aufruf durch Fachmodul auch zusätzlich eGK) (Reihenfolge der Parameter abweichend)	• cardSession (Kartensitzung; zulässig sind SM-B, ~~HBAx~~ oder bei Aufruf durch Fachmodul auch zusätzlich eGK)
	• signRequests (Liste von Signaturaufträgen. Jeder Signaturauftrag (SignRequest) kapselt:
• Zu signierendes Dokument (Document) bzw. zu signierende Dokumente	o documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente); darin u.a. documentFormat (Formatangabe für das zu signierende Dokument)
• Weitere optionale Eingabeparameter (siehe Operation SignDocument, Parameter dss:OptionalInputs) (Reihenfolge der Parameter abweichend)	o optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur, siehe Operation SignDocument, Parameter dss:OptionalInputs); darin u.a. documentFormat (Formatangabe für das zu signierende Dokument)
	o includeRevocationInfo: [Boolean] – *optional; default: true* (Dieser optionale Parameter steuert die Einbettung von OCSP-Antworten in die Signatur: nur wirksam bei der Prüfung von enthaltenen Parallelsignaturen, wenn eine Gegensignatur erstellt werden soll. Die OCSP-Antworten werden in die jeweils geprüfte Parallelsignatur eingebettet.)

OPB1	OPB2.1
• WorkplaceId	• workplaceId (Identifikator des Arbeitsplatzes)
Ausgangsdaten	
Signierte Dokumente	• signedDocuments (Liste der signierten Dokumente)

4.1.8.4.2 TUC_KON_161 "nonQES Dokumentsignatur prüfen"

TAB_KON_121 - TUC_KON_161 „nonQES Dokumentsignatur prüfen"

OPB1	OPB2.1
Eingangsdaten	
• Signiertes Document vom Typ `nonQES_DocFormate`	• signedDocument (Signiertes Document vom Typ `nonQES_DocFormate`)
• Signatur (optional, falls detached Signatur). Es werden Parallel- und Gegensignaturen unterstützt.	• signature – *optional/falls detached Signatur* (Signatur. Es werden Parallel- und Gegensignaturen unterstützt.)
• optionale Eingabeparameter (siehe Operation VerifyDocument, Parameter SIG:OptionalInputs)	• optionalInputs (optionale Eingabeparameter, siehe Operation VerifyDocument, Parameter SIG:OptionalInputs)
• X.509-Zertifikat (falls das Zertifikat nicht im signierten Dokument enthalten ist)	• certificate – *optional/verpflichtend, wenn das Zertifikat nicht im signierten Dokument enthalten ist* (X.509-Zertifikat, gegen das sie Signatur geprüft werden soll)
• Grace Period	• ocspGracePeriod (OCSP-Grace Period: maximal zulässiger Zeitraum, den die letzte OCSP-Antwort aus dem Cache bezüglich des Referenzzeitpunkts zurückliegen darf)
Für XML-Dokumente:	
• Liste von XML-Schemata (optional)	• xmlSchemas – *optional/nur für XML-Dokumente* (XMLSchema und ggf. weitere vom Hauptschema benutzte Schemata)
• includeRevocationInfo [Boolean] – optional; Default = false (Dieser Parameter steuert die Einbettung von OCSP-Responses in die Signatur.)	• includeRevocationInfo: [Boolean] – *optional; Default = false* (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur)

OPB1	OPB2.1
Ausgangsdaten	
• VerificationResult	• verificationResult [VerificationResult] (Ergebnis der Signaturprüfung)
• SIG:OptionalOutput (optional)	• optionalOutput – *optional* (weitere Ausgabedaten gemäß SIG:OptionalOutput)

4.1.8.4.4 TUC_KON_150 "Dokumente QES signieren"

TAB_KON_755 - TUC_KON_150 „Dokumente QES signieren"

OPB1	OPB2.1
Eingangsdaten	
• SignRequests. Jeder SignRequest kapselt:	• signRequests (Liste von Signaturaufträgen. Jeder Signaturauftrag (SignRequest) kapselt:
o Zu signierendes Dokument (Document) bzw. zu signierende Dokumente	o documentsToBeSigned (Zu signierendes Dokument bzw. zu signierende Dokumente)
o weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur (siehe Operation SignDocument, Parameter dss:OptionalInputs)	o optionalInputs (weitere optionale Eingabeparameter zur Steuerung der Details bei der zu erstellenden Signatur, siehe Operation SignDocument, Parameter dss:OptionalInputs)
o includeRevocationInfo [Boolean] - optional; Default: true Dieser optionale Parameter steuert die Einbettung von OCSP-Responses in die Signatur (siehe Operation SignDocument, Parameter SIG:IncludeRevocationInfo)	o includeRevocationInfo [Boolean] – optional; Default: true (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur; siehe Operation SignDocument, Parameter SIG:IncludeRevocationInfo)
• CardSession (HBAx)	• cardSession (Kartensitzung. Unterstützte Kartentypen: HBAx)
• WorkplaceId	• workplaceId
Ausgangsdaten	
Signierte Dokumente	• signedDocuments (Liste der signierten Dokumente)

4.1.8.4.7 TUC_KON_151 "QES Dokumentensignatur prüfen"

TAB_KON_591 - TUC_KON_151 „QES-Dokumentensignatur prüfen“

OPB1	**OPB2.1**
Eingangsdaten	
• QES-signiertes Dokument vom Typ QES_DocFormate	• signedDocument – *optional* (QES-signiertes Dokument vom Typ `QES_DocFormate` -> siehe Definition in Operation `VerifyDocument` mit `SIG:Document`)
• QES Signatur(en). Es werden Parallel- und Gegensignaturen unterstützt.	• signatureObject – *optional* (-> siehe Definition in Operation VerifyDocument mit dss:SignatureObject). Es werden Parallel- und Gegensignaturen unterstützt.)
• optionale Eingabeparameter (siehe Operation VerifyDocument, Parameter SIG:OptionalInputs)	• optionalInputParams (optionale Eingabeparameter, siehe Operation VerifyDocument, Parameter SIG:OptionalInputs)
• X.509-Zertifikate (falls diese nicht im signierten Dokument enthalten sind, sondern nur referenziert werden).	• certificates – *optional/falls diese nicht im signierten Dokument enthalten sind, sondern nur referenziert werden* (X.509-Zertifikate).
• Kurztext (vom Clientsystem übergeben)	• message (Kurztext, vom Clientsystem übergeben)
• WorkplaceId	• workplaceId (Arbeitsplatz)
Für XML-Dokumente:	
• Liste von XML-Schemata (optional)	• xmlSchemas – *optional/nur für XML-Dokumente* (XMLSchema und ggf. weitere vom Hauptschema benutzte Schemata)
• includeRevocationInfo [Boolean] - *optional; Default: false* (Dieser optionale Parameter steuert die Einbettung von OCSP-Responses in die Signatur)	• includeRevocationInfo [Boolean]: – *optional; Default: false* (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur)
Ausgangsdaten	
• VerificationResult	• verificationResult [VerificationResult] (Ergebnis der Signaturprüfung)
• SIG:OptionalOutput (optional)	• optionalOutput – *optional* (weitere Ausgabedaten gemäß SIG:OptionalOutput)

4.1.9 Zertifikatsdienst

4.1.9.3.1 TUC_KON_032 „TSL aktualisieren“

TAB_KON_766 TUC_KON_032 „TSL aktualisieren“

OPB1	OPB2.1
Eingangsdaten	
• TSL aus manuellem Import (Optional)	• importedTSL – *optional* (TSL aus aus manuellem Import) (Optional)
• Referenzzeitpunkt (Default: aktuelles Datum)	• baseTime – *optional; default: aktuelles Datum* (Referenzzeitpunkt) ()
• Flag „MGM_LU_ONLINE“ für Offline/Online-Modus	• onlineMode [ENABLED/DISABLED] (Flag „MGM_LU_ONLINE“ für Offline/Online-Modus)
Ausgangsdaten	
• Status der Prüfung	• result (Status der Prüfung)

4.1.9.3.2 TUC_KON_040 „CRL aktualisieren“

TAB_KON_767 TUC_KON_040 „CRL aktualisieren“

OPB1	OPB2.1
Eingangsdaten	
• Manuell importierte CRL (Optional)	• importedCRL – *optional* (Manuell importierte CRL)
Ausgangsdaten	
• Keine	• Keine

4.1.9.3.3 TUC_KON_033 „Zertifikatsablauf prüfen“

TAB_KON_768 TUC_KON_033 „Zertifikatsablauf prüfen“

OPB1	OPB2.1
Eingangsdaten	
• CardSession oder checkSMCK	• cardSession – *optional*/für eGK, HBA, SM-B, gSMC-KT
•	• checkSMCK [Boolean] – *optional/für*

OPB1	OPB2.1
	gSMC-K; (Referenz auf eine/die gSMC-K, alternativ zu cardSession)
• doInformClients	• doInformClients [Boolean] (Angabe, ob ein Event an die Clients gesendet werden soll)
Ausgangsdaten	
• Ablaufdatum	• expirationDate (Ablaufdatum des untersuchten Zertifikats)

4.1.9.4.1 TUC_KON_037 „Zertifikat prüfen"

TAB_KON_769 TUC_KON_037 „Zertifikat prüfen“

OPB1 (Teil: X.509-Zertifikate	OPB2.1
Eingangsdaten)	
• CV-Zertifikat und der öffentliche Schlüssel der zugehörigen ausstellenden CVC-CA oder X.509-Zertifikat	• certificate (ein X.509-Zertifikat (nonQES- oder QES-X.509-Zertifikat))
o QUALIFIED={not_required|required|if_QC_present}	• qualifiedCheck [not_required | required | if_QC_present] – (Art der Zertifikatsprüfung)
o Referenzzeitpunkt: Zeitpunkt, für den das Zertifikat geprüft werden soll (optional; bei Nichtangabe Verwendung der Systemzeit des Konnektors)	• baseTime – *optional/verpflichtend, wenn ein Zeitpunkt zur Prüfung vorgegeben werden soll* (Referenzzeitpunkt: Zeitpunkt, für den das Zertifikat geprüft werden soll; bei Nichtangabe Verwendung der Systemzeit des Konnektors)
o OFFLINE_ALLOW_NOCHECK. (true/false; Default: false)	• offlineAllowNoCheck [Boolean] – *optional; default: false* (Angabe, ob im Offlinefall der Check entfallen darf.)
~~o PolicyList: Zugelassene Zertifikatstyp-OID gemäß [gemSpec_OID#GS-A_4445]~~	~~• policyList (Liste der zugelassenen Zertifikatstyp-OIDs gemäß [gemSpec_OID#GS-A_4445])~~
nur für nonQes-Zertifikate	
~~o PolicyList: Zugelassene Zertifikatstyp-OID gemäß [gemSpec_OID#GS-A_4445]~~	~~• policyList (Liste der zugelassenen Zertifikatstyp-OIDs gemäß [gemSpec_OID#GS-A_4445])~~
o Vorgesehene KeyUsage (intendedKeyU-	• intendedKeyUsage – *optio-*

OPB1 (Teil: X.509-Zertifikate	OPB2.1
sage)	*nal/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist; wird bei QES nicht ausgewertet* (Vorgesehene KeyUsage)
o Vorgesehene ExtendedKeyUsage (intendedExtendedKeyUsage)	o intendedExtendedKeyUsage – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist; wird bei QES nicht ausgewertet* (Vorgesehene ExtendedKeyUsage)
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 gracePeriod – *optional/nur für nonQES-X.509-Zertifikat und wenn vom Standard abgewichen werden soll; wird bei QES nicht ausgewertet; default: CERT_OCSP_DEFAULT_GRACE_PERIOD_NONQES* (OCSP-GracePeriod: maximal zulässiger Zeitraum, den letzte OCSP-Antwort aus dem Cache bezüglich des Referenzzeitpunkts zurückliegen darf;)
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 validationMode [OCSP | CRL | NONE] – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist* (Prüfmodus: ▪ OCSP: Es wird mittels OCSP geprüft. Dabei wird, falls die OCSP-GracePeriod noch nicht abgelaufen ist, die OCSP-Antwort aus dem Cache des Konnektors verwendet. Für QES einzig erlaubter validationMode. ▪ CRL: Es wird gegen die aktuelle CRL auf dem Konnektor geprüft. ▪ NONE: Keine Prüfung von Statusinformationen)
Alle Zertifikate	
o OCSP-Response (nonQES)/Liste von OCSP-Responses (QES)	• ocspResponses – *optional* (Liste von OCSP-Responses bei QES, einzelne OCSP-Response bei nonQES)
o getOCSPResponses [Boolean] - *optional; Default: false* (liefert die Information, ob eine Liste von OCSP-Responses (QES) oder einzelne OCSP-Antwort (nonQES) des geprüften Zertifikats an den Aufrufer zurückzugegeben ist)	• getOCSPResponses [Boolean] – *optional; default: false* (true – eine Liste von OCSP-Antworten (QES) oder eine einzelne OCSP-Antwort (nonQES) des geprüften Zertifikats soll an den Aufrufer zurückgegeben werden)

OPB1 (Teil: X.509-Zertifikate	OPB2.1
o Liste von Attributzertifikaten (optional, QES)	• attribCertificates – *optional* (Liste von Attributzertifikaten)
Ausgangsdaten	
• Status und Liste von Warnungen/Fehlern bei der Zertifikatsprüfung	• Status und Liste von Warnungen/Fehlern bei der Zertifikatsprüfung
• Ermittelte Rolle (X.509-Zertifikate) Werte ermittelt aus dem Zertifikat aus „Tab_PKI_406 OID-Festlegung technische Rolle in X.509-Zertifikaten“ oder „Tab_PKI_402 OID-Festlegung Rolle im X.509-Zertifikat für Berufsgruppen“ oder Tab_PKI_403 OID-Festlegung Institutionen im X.509-Zertifikat der SMC-B [gemSpec_OID]	• role (aus dem Zertifikate ermittelte Rolle oder Berufsgruppe; siehe „Tab_PKI_406 OID-Festlegung technische Rolle in X.509-Zertifikaten“ oder „Tab_PKI_402 OID-Festlegung Rolle im X.509-Zertifikat für Berufsgruppen“ oder Tab_PKI_403 OID-Festlegung Institutionen im X.509-Zertifikat der SMC-B [gemSpec_OID])
• QCStatements des Zertifikats	• qcStatement – *optional/verpflichtend, wenn certificate ein QES-X.509-Zertifikat ist*, (QCStatements des Zertifikats)
• ocspResponsesRenewed – *optional/verpflichtend, wenn Eingabeparameter getOCSPResponses = true* (eine Liste von OCSP-Responses (QES) oder einzelne OCSP-Antwort (nonQES) des geprüften Zertifikats)	• ocspResponsesRenewed – *optional/verpflichtend, wenn Eingabeparameter getOCSPResponses = true* (eine Liste von OCSP-Responses (QES) oder einzelne OCSP-Antwort (nonQES) des geprüften Zertifikats)

4.1.9.4.2 TUC_KON_042 „CV-Zertifikat prüfen“

Hinweis: Die CV-Zertifikatsprüfung wurde aus TUC_KON_037 entfernt und in einen eigenen TUC ausgelagert. Dies führte zu dem neuen TUC_KON_042 „CV-Zertifikat prüfen“. Die Abläufe wurden dadurch übersichtlicher und die Komplexität konnte vereinfacht werden.

TAB_KON_818 TUC_KON_042 „CV-Zertifikat prüfen“

OPB1 (nur CVC-Teil)	OPB2.1
Eingangsdaten	
• CV-Zertifikat und der öffentliche Schlüssel der zugehörigen ausstellenden CVC-CA oder X.509-Zertifikat	• eeCertificate (zu prüfendes kartenindividuelles CV-Zertifikat)
	• caCertificate (das CVC-CA-Zertifikat mit dem öffentlichen Schlüssel der zugehörigen ausstellenden CA)
Ausgangsdaten	

• Status und Liste von Warnungen/Fehlern bei der Zertifikatsprüfung	• status [Boolean] (Ergebnis der Prüfung; true: CV-Zertifikat ist gültig false: CV-Zertifikat ist ungültig)

4.1.9.4.3 TUC_KON_034 „Zertifikatsinformationen extrahieren“

TAB_KON_770 TUC_KON_034 „Zertifikatsinformationen extrahieren“

OPB1	**OPB2.1**
Eingangsdaten	
• Aufrufkontext (Mandant)	• Aufrufkontext (Mandant)
• CardHandle oder checkSMCK	• cardHandle – *optional*/für eGK, HBA, SM-B, gSMC-KT
	• checkSMCK [Boolean] – *optional/für gSMC-K;* (Referenz auf eine/die gSMC-K, alternativ zu cardSession)
• QES (true/false; Default: false) – Angabe, ob die QES-Identität oder die nonQES-Identität der Karte interessiert	• checkQES [Boolean] - *optional; Default: false* – (Angabe, ob die QES-Identität oder die nonQES-Identität der Karte interessiert)
Ausgangsdaten	
• Zertifikatstyp	• certType [C.CH.AUT | C.HP.AUT | C.HCI.AUT | C.HP.QES] (Zertifikatstyp)
• Zertifikatsinformationen (s. Standardablauf)	• certInfo (Zertifikatsinformationen, bestehend aus SerialNumber, Issuer, Subject, Rollen, registrationNumber und ggf. id-etsi-qcs-QcCompliance, siehe Standardablauf)
• ggf. QCStatements	• qcStatments – *optional/nur wenn certType = C.HP.*QES (QCStatements)

4.1.10 Protokollierungsdienst

4.1.10.4.1 TUC_KON_271 „Schreibe Protokolleintrag“

TAB_KON_607 - TUC_KON_271 „Schreibe Protokolleintrag“

OPB1	OPB2.1
Eingangsdaten	
• Zu protokollierendes Ereignis	• Zu protokollierendes Ereignis
o fmName – optional/verpflichtend für Aufruf durch Fachmodule (Name des aufrufenden Fachmoduls; Default: „: das Ereignis wird in das entsprechende Konnektor-Protokoll geschrieben)	o fmName – *optional/verpflichtend für Aufruf durch Fachmodule; default = „“* (Name des aufrufenden Fachmoduls; das Ereignis wird in das entsprechende Konnektor-Protokoll geschrieben)
o Typ (Sec, Op, Perf) definiert den Protokolltyp, in welchen das Ereignis geschrieben wird; Sec = Security: Ereignis wird in das Securityprotokoll geschrieben Op = Operation: Wenn fmName =““ wird das Ereignis in das Systemprotokoll geschrieben. Wenn fmName gesetzt ist, wird das Ereignis in das durch fmName definierte Fachmodul-Protokoll geschrieben. Perf = Performance: Wenn fmName =““ wird das Ereignis in das Konnektor-Performanceprotokoll geschrieben. Wenn fmName gesetzt ist, wird das Ereignis in das durch fmName definierte Fachmodul-Performanceprotokoll geschrieben.	o eventType [EventType] definiert den Protokolltyp, in welchen das Ereignis geschrieben wird; Sec = Security: Ereignis wird in das Securityprotokoll geschrieben Op = Operation: Wenn fmName =““ wird das Ereignis in das Systemprotokoll geschrieben. Wenn fmName gesetzt ist, wird das Ereignis in das durch fmName definierte Fachmodul-Protokoll geschrieben. Perf = Performance: Wenn fmName =““ wird das Ereignis in das Konnektor-Performanceprotokoll geschrieben. Wenn fmName gesetzt ist, wird das Ereignis in das durch fmName definierte Fachmodul-Performance protokoll geschrieben.
o Schwere (Debug = Debug Information, Info = Information, Warn = Warning, Err = Error, Fatal)	o severity { [EventSeverity] , Debug} (Schwere mit: Debug = Debug Information, Info = Information, Warn = Warning, Err = Error, Fatal
• Parameter beinhaltet die Daten des Ereignisses, die im Protokolleintrag geschrieben werden	• parameters beinhaltet die Daten des Ereignisses, die im Protokolleintrag geschrieben werden
Ausgangsdaten	
• Keine	• Keine

4.1.11 TLS-Dienst

4.1.11.4.1 TUC_KON_110 "Kartenbasierte TLS-Verbindung aufbauen"

TAB_KON_773 - TUC_KON_110 "Kartenbasierte TLS-Verbindung aufbauen“

OPB1	OPB2.1
Eingangsdaten	
• Role – *optional/wenn Rollenprüfung durchgeführt werden soll*	• role – *optional/verpflichtend, wenn Rollenprüfung durchgeführt werden soll*
• CardSession (SM-B) (optional)	• cardSession – *optional/verpflichtend, wenn Clientauthentisierung durchgeführt werden soll* (CardSession einer SM-B)
• URI des Verbindungsziels	• targetUri (URI des Verbindungsziels)
Ausgangsdaten	
• TLSConnectionIdentifier	• tlsConnectionId (ConnectionIdentifier der TLS-Verbindung)

4.1.11.4.2 TUC_KON_111 "Kartenbasierte TLS-Verbindung abbauen"

TAB_KON_774 - TUC_KON_111 "Kartenbasierte TLS-Verbindung abbauen“

OPB1	OPB2.1
Eingangsdaten	
• TLSConnectionIdentifier	• tlsConnectionId (ConnectionIdentifier der TLS-Verbindung)
Ausgangsdaten	
• Keine	• Keine

4.1.12 LDAP-Proxy

Keine Änderungen

Anhänge

Anhang A – Datentypen von Eingangs- und Ausgangsdaten

Die nachfolgende Tabelle enthält die Datentypen, die in den Schnittstellenbeschreibungen in Kapitel 2 als Textmarken referenziert werden.

Tabelle 396: Aufzähltypen

Typname	Werteliste
[Boolean]	{true | false}
[EncryptionType]	{CMS | XMLEnc | S/MIME}
[EventType]	{Op | Sec | Perf}
[EventSeverity]	{Debug | Info | Warn | Err | Fatal}
[KtOutputMode]	{Input | OutputWait | OutputConfirm | OutputKeep | OutputErase}
[PinStatus]	{VERIFIED | VERIFYABLE | BLOCKED | TRANSPORT_PIN | EMPTY_PIN | DISABLED}
[PinResult]	{OK | REJECTED | BLOCKED | ERROR}
[PukResult]	{OK | REJECTED | WASBLOCKED | NOWBLOCKED | ERROR}
[VerificationResult]	{VALID | INVALID | INCONCLUSIVE}

Anhang B – Referenzierte Dokumente

Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument referenzierten Dokumente der gematik zur Telematikinfrastruktur. Der mit der vorliegenden Version korrelierende Entwicklungsstand dieser Konzepte und Spezifikationen wird pro Release in einer Dokumentenlandkarte definiert, Version und Stand der referenzierten Dokumente sind daher in der nachfolgenden Tabelle nicht aufgeführt. Deren zu diesem Dokument passende jeweils gültige Versionsnummer entnehmen Sie bitte der aktuellsten, auf der Internetseite der gematik veröffentlichten Dokumentenlandkarte, in der die vorliegende Version aufgeführt wird.

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
gemSpec_Kon	gematik: Spezifikation Konnektor, Version 4.11.1
gemSpec_Kon_OPB2.1	gematik, Spezifikation Konnektor, Version 5.2.0