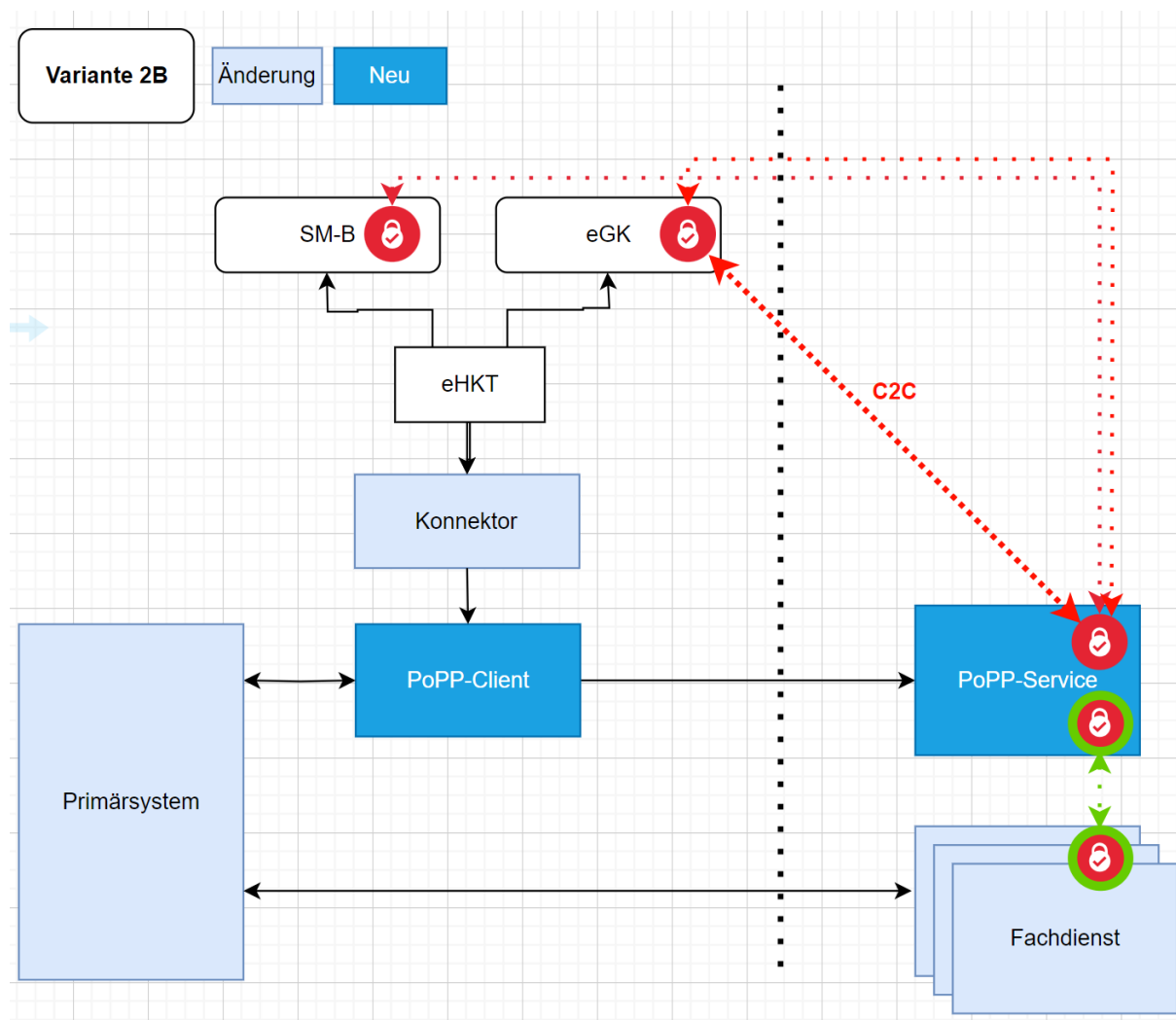


Diese Anlage beschreibt Änderungen am Konnektor, die unten skizzierte PoPP-Lösung unterstützen sollen.

Eine Beschreibung findet sich im Änderungsbedarf des Änderungseintrags.



Inhaltsverzeichnis

1 Änderung in gemSpec_Kon	3
1.1 Kapitel 4.1.5 "Kartendienst"	3
1.2 Anhang F - Übersicht Events	15
1.3 Kapitel 5.5.1 Dokumente der gematik	15
2 Änderungen in Steckbriefen	16
2.1 Änderungen in gemProdT_Kon_PTV6	16
2.2 Sonstige Änderungen	16

1 Änderung in gemSpec_Kon

1.1 Kapitel 4.1.5 "Kartendienst"

In Kapitel 4.1.5.1 wird Anforderung A_25895 am Ende neu aufgenommen:

A_25895 - Exklusive Nutzung des Karten-Kommunikationskanals durch Operation SecureSendAPDU

Wenn eine Karte durch Aufruf der Operation StartCardSession reserviert ist, dann MUSS jeder weitere Aufruf von TUC_KON_023 mit doLock=true, welcher dieselbe Karte zu reservieren versucht, mit Fehlercode 4093 abbrechen. [<=, ,]

In Kapitel 4.1.5.2 "Durch Ereignisse ausgelöste Reaktionen" wird A_25860 unter A_23702 neu aufgenommen:

A_25860 - Reaktion auf abgelaufenen APDU-Szenario-Timer

Der Konnektor MUSS bei Eintritt der Situationen a) oder b) die Aktionen 1) und 2) ausführen:

- a. Der vorhergehende, zu einer cardSession zugehörige Aufruf der Operation SecureSendAPDU wies keinen Fehler auf und der zu derselben cardSession zugehörige Folgeaufruf von SecureSendAPDU ist nicht innerhalb des durch den vorhergehenden Aufruf definierten Zeitraums (TimeSpan) nach Absenden der Response zum vorherigen Aufruf erfolgt.
- b. Der vorhergehende, zu einer cardSession zugehörige Aufruf der Operation StartCardSession wies keinen Fehler auf und der zugehörige erste Aufruf der Operation SecureSendAPDU ist nicht innerhalb des durch \$CARD_SESSION_TIMEOUT definierten Zeitraums erfolgt.

1) Aufruf von TUC_KON_224 { sessionId = sessionId(cardSession) }

2) Ereignis auslösen durch TUC_KON_256 Systemereignis {

topic = „CARD/SESSION/TIMEOUT“;

eventType = Op;

severity = Info;

parameters = (CardType=eGK, SessionID=\$sessionId, Timer=\$Timer)}

[<=, Konnektor Highspeed, Konnektor PTV6, funkt. Eignung: Test Produkt/FA]

Prüfverfahren: Funktionale Eignung

Der Inhalt von Kapitel 4.1.5.4.21 wird auf 4.1.5.4.22 inkrementiert. Die darauf folgenden Kapitel der Dokumentenebene 5.1.5.4.* inkrementieren entsprechend.

[api-popp] entspricht aktuell dem Pull Request unter dem Link

<https://github.com/gematik/api-popp/pull/1>

Die folgenden neuen und geänderten Anforderungen werden in Kapitel 4.1.5.4 aufgenommen.

A_26067 - TUC_KON_223 "Starte Kartensitzung"

Der Konnektor MUSS den technischen Use Case „Starte Kartensitzung“ gemäß TAB_KON_279 umsetzen.

Tabelle 1: TAB_KON_279 – TUC_KON_223 „Starte Kartensitzung“

Element	Beschreibung
Name	TUC_KON_223 „Starte Kartensitzung“
Beschreibung	Der technische Use Case richtet eine Karte für eine Kartensitzung ein. Als Kartentyp wird die eGK unterstützt.
Auslöser	Operation StartCardSession
Vorbedingungen	keine
Eingangsdaten	<ul style="list-style-type: none"> cardSession
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	<ul style="list-style-type: none"> sessionID
Standardablauf	<ol style="list-style-type: none"> Karte reservieren mit TUC_KON_023 { cardSession; doLock = true } Karte zurücksetzen mit TUC_KON_024 { cardSession } Generiere sessionID als UUID gem. [RFC4122] und persistiere diese im Kontext der \$cardSession Setze APDU-Szenario-Timer auf Wert von CARD_SESSION_TIMEOUT
Varianten/ Alternativen	Keine
Fehlerfälle	Keine

Tabelle 2: TAB_KON_280 – Fehlercodes TUC_KON_223 „Starte Kartensitzung“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases treten keine weiteren Fehlercodes auf.			

[<=, ,]

A_26069 - TUC_KON_208 "Sende gesicherte APDU"

Der Konnektor MUSS den technischen Use Case „Sende gesicherte APDU“ gemäß TAB_KON_283 umsetzen.

Tabelle 3: TAB_KON_283 – TUC_KON_208 „Sende gesicherte APDU“

Element	Beschreibung
Name	TUC_KON_208 „Sende gesicherte APDU“
Beschreibung	Der technische Use Case löst Karten-Transaktionen aus. Aus übergebenen integritäts- und authentizitätsgeschützten Transaktionsdaten extrahiert er APDUs, sendet diese zur Ausführung an die Karte und gibt die Ergebnisse an den Aufrufer zurück. Als Kartentyp wird die eGK unterstützt.
Auslöser	Operation SecureSendAPDU
Vorbedingungen	keine
Eingangsdaten	<ul style="list-style-type: none">• transactionData
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	<ul style="list-style-type: none">• transactionResult• timeSpan

Element	Beschreibung
Standardablauf	<ol style="list-style-type: none"> 1. Setze die Liste der erwarteten StatusCodes (\$StatusCodeList) durch Leeren der Liste und Einfügen eines Code-Elements mit Wert 0x9000 zurück. 2. Dekodiere <code>transactionData</code> (siehe Eingangsdaten) und extrahiere als Scenario eine Liste von Elements, die jeweils SequenceCounter und SessionID enthalten. Das Format von Scenario ist in [api-popp] beschrieben. 3. Prüfe, ob eine CardSession mit \$SessionID existiert (d. h. durch Aufruf von TUC_KON_223 persistiert wurde) 4. Prüfe, dass ein Lock für die durch \$SessionID identifizierte CardSession bereits existiert 5. Prüfe ob die laufende Sequenznummer \$SequenceCounter entweder die erste (\$SequenceCounter = 0) oder das Inkrement des vorhergehenden Aufrufes ist 6. Für jedes \$Element aus \$Scenario.Scenario7816: <ol style="list-style-type: none"> a. Falls \$Element eine Listevon erwarteten StatusCodes (ExpectedStatusWords) ist <ul style="list-style-type: none"> • \$StatusCodeList = "\$Element" b. Falls \$Elementeine Kommando-APDU (CommandAPDU) ist <ul style="list-style-type: none"> • Ermittle \$responseAPDU für das \$Elementmittels Aufruf von TUC_KON_200 { <ul style="list-style-type: none"> • cardSession = "\$cardSession"; • ctId nicht übergeben; • commandAPDU = "\$Element" } • Hänge \$responseAPDUan das Ende der ResultList • Falls Status der \$responseAPDUnicht in StatusCodeList <ul style="list-style-type: none"> • nimm die Warnung 4284 in die Antwort auf • verlasse die Schleife c. Falls \$Element eine Logging-Information (LoggingInformation) ist <ul style="list-style-type: none"> • führe für das \$Element keine Aktionen durch d. Falls \$Element ein anderes Element als die in a)-c) aufgezählten ist <ul style="list-style-type: none"> • führe für das \$Element keine Aktionen durch 7. Falls \$Scenario.TimeSpan = 0 (letztes Szenario): stoppe Kartensitzung durch TUC_KON_224 { sessionId = \$SessionID" } Andernfalls: Setze APDU-Szenario-Timer auf Wert von \$Scenario.TimeSpan 8. Setze timeSpan = \$Scenario.TimeSpan

Element	Beschreibung
Varianten/ Alternativen	Keine
Fehlerfälle	<p>(->2) Die dekodierten Eingabeparameter sind nicht nach [api-popp] validierbar: Fehlercode 4286 (->3) \$SessionID existiert nicht: Fehlercode 4288 (->4) Es ist kein Lock für \$SessionID gesetzt: Fehlercode 4289 (->5) Die laufende Sequenznummer \$SequenceCounter ist nicht die erste und nicht das Inkrement des vorhergehenden Aufrufes: Fehlercode 4285</p> <p>In Fehlerfällen ab Schritt 5): TUC_KON_224 { sessionID = \$SessionID }</p>

Tabelle 4: TAB_KON_280 – Fehlercodes TUC_KON_223 „Sende gesicherte APDU“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4284	Technical	Warnung	APDU konnte nicht verarbeitet werden.
4285	Technical	Error	Unerwartetes Sequence-Element
4286	Technical	Error	Inhalt von TransactionData nicht valide
4288	Technical	Error	Unbekannte Session ID
4289	Technical	Error	Karte nicht reserviert

[<=, ,]

A_26068 - TUC_KON_224 "Stoppe Kartensitzung"

Der Konnektor MUSS den technischen Use Case „Stoppe Kartensitzung“ gemäß TAB_KON_281 umsetzen.

Tabelle 5: TAB_KON_281 – TUC_KON_224 „Stoppe Kartensitzung“

Element	Beschreibung
---------	--------------

Element	Beschreibung
Name	TUC_KON_224 „Stoppe Kartensitzung“
Beschreibung	Der technische Use Case beendet eine Kartensitzung.
Auslöser	Operation StopCardSession
Vorbedingungen	keine
Eingangsdaten	<ul style="list-style-type: none"> sessionID
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	Keine
Standardablauf	<ol style="list-style-type: none"> 1. Ermittle durch \$sessionID identifizierte CardSession 2. Entferne die SessionID aus dem persistierten CardSession Kontext 3. Setze die Karte zurück mittels TUC_KON_024 { cardSession } 4. Entferne das Lock von der Karte durch TUC_KON_023 { cardSession; doLock = false }
Varianten/ Alternativen	Keine
Fehlerfälle	(->1) \$SessionID existiert nicht: Fehlercode 4288

Tabelle 6: TAB_KON_282 – Fehlercodes TUC_KON_224 „Stoppe Kartensitzung“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4288	Technical	Error	Unbekannte Session ID

[<=, ,]

TIP1-A_4584 wird durch TIP1-A_4584-02 ersetzt (Achtung: gleiche Änderung wird auch in [F ML-155357 - Komfortsignatur: Integration des Fehlerhandling von Fehler 4019 in die Operation ActivateComfortSignature](#) vorgenommen):

TIP1-A_4584-02 - TUC_KON_024 „Karte zurücksetzen“

Der Konnektor MUSS den technischen Use Case „Karte zurücksetzen“ gemäß TUC_KON_024 umsetzen.

Tabelle 7: TAB_KON_737 – TUC_KON_024 „Karte zurücksetzen“

Element	Beschreibung
Name	TUC_KON_024 „Karte zurücksetzen“
Beschreibung	Der technische Use Case setzt die gewählte Karte zurück (alle erreichten Sicherheitszustände werden auf der Karte und in der Verwaltung des Konnektors zurückgesetzt; auf der Karte wird MF selektiert). Ein eventuell laufendes C2C wird dabei abgebrochen.
Auslöser	Aufruf durch: <ul style="list-style-type: none"> Basisdienst Fachmodul
Vorbedingungen	keine
Eingangsdaten	<ul style="list-style-type: none"> ctId – <i>optional/verpflichtend, wenn keine cardSession angegeben ist</i> (Kartenterminalidentifikator) slotId – <i>optional/verpflichtend, wenn keine cardSession angegeben ist</i> (Nummer des Slots, in dem die Karte steckt) cardSession – <i>optional/verpflichtend, wenn ctId und slotId nicht angegeben sind</i> (Angabe der CardSession alternativ zur Angabe von ctId und slotId)
Komponenten	Karte, Kartenterminal, Konnektor
Ausgangsdaten	Keine
Standardablauf	<ol style="list-style-type: none"> Wenn cardSession gegeben, dann ermittle ctId und slotId Der Konnektor prüft, dass entweder die Karte nicht reserviert ist oder der Aufrufer im Besitz des Karten-Locks ist. Brich eventuell parallel laufenden TUC_KON_005 ab Sende SICCT RESET ICC für slotId an das Kartenterminal CtID, um einen Warm Reset auszulösen Lösche alle Sicherheitszustände aus CARDSESSION.AUTHSTATE und den Inhalt von CARDSESSION.AUTHBY.
Varianten/ Alternativen	Keine

Element	Beschreibung
Fehlerfälle	* Karte antwortet nicht innerhalb von CARD_TIMEOUT_CARD Sekunden, Fehlercode 4094 (→2) Der Aufrufer ist nicht im Besitz des Karten-Locks, Fehlercode 4232 (→4) Karte antwortet mit einer spezifischen Fehlermeldung, Fehlercode <Kartenfehlercode gemäß [gemSpec_COS]>
Nichtfunktionale Anforderungen	Keine
Zugehörige Diagramme	Keine

Tabelle 8: TAB_KON_544 Fehlercodes TUC_KON_024 „Karte zurücksetzen“

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4094	Technical	Error	Timeout beim Kartenzugriff aufgetreten
4232	Technical	Error	der Aufrufer ist nicht im Besitz des Karten-Locks

[<=, Konnektor Highspeed, Konnektor PTV4, Konnektor PTV5, Konnektor PTV5Plus, Konnektor PTV6, Konnektor PTV4Plus, Konnektor eHealth, Sich.techn. Eignung: CC-Evaluierung, funkt. Eignung: Test Produkt/FA, Sich.techn. Eignung: Prüfung durch CC-Prüfstelle]

[Die folgenden neuen Anforderungen werden in Kapitel 4.1.5.5 aufgenommen.](#)

A_25970 - Operation StartCardSession

Der Konnektor MUSS an der Außenschnittstelle eine Operation StartCardSession, wie in Tabelle TAB_KON_273 Operation StartCardSession beschrieben, anbieten.

Tabelle 9: TAB_KON_273 Operation StartCardSession

Name		
Beschreibung	Die Operation nimmt eine operationsübergreifende Reservierung einer Karte vor und erzeugt eine unique Session ID zur Verwendung in Folgeaufrufen von Kartenoperationen.	
Aufrufparameter	Name	Beschreibung
	CCTX:Context	MandantId, CsId, WorkplaceId verpflichtend
	CONN:CardHandle	Adressiert die Karte, mit der in Folgeaufrufen Kartenoperationen ausgeführt werden sollen. Die Operation MUSS die eGK unterstützen. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4209 abbrechen.

Name		
Rückgabe	Name	Beschreibung
	SessionID	UUID gem. [RFC4122]

Der Ablauf der Operation StartCardSession ist in Tabelle TAB_KON_274 Ablauf StartCardSession beschrieben.

Tabelle 10 TAB_KON_274 Ablauf StartCardSession

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“	Prüfung der Zugriffsberechtigung durch den Aufruf TUC_KON_000 { mandantId = Context.mandantId; clientSystemId = Context.clientsystemId; workplaceId = Context.workplaceId; userId = Context.userId; CardHandle } Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.
3.	TUC_KON_026 „Liefere CardSession“	Ermittle CardSession über TUC_KON_026 { mandantId = \$context.mandantId; clientsystemId = \$context.clientsystemId; cardHandle = \$context.cardHandle; userId = \$context.userId }
4.	TUC_KON_223 „Starte Kartensitzung“	TUC_KON_223 { cardSession = CardSession }

Tabelle 11: TAB_KON_277 Fehlercodes StartCardSession

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=, ,]

A_25822 - Operation SecureSendAPDU

Der Konnektor MUSS an der Außenschnittstelle eine Operation SecureSendAPDU, wie in Tabelle TAB_KON_270 Operation SecureSendAPDU beschrieben, anbieten.

Tabelle 12: TAB_KON_270 Operation SecureSendAPDU

Name	SecureSendAPDU	
Beschreibung	<p>Die Operation sendet eine Liste von Kommando-APDUs an eine Karte und liefert die Liste der Rückgabe-APDUs. Die Operation MUSS nur eGK unterstützen.</p> <p>Die Zuordnung der Kommando-APDUs und der Rückgabe-APDUs ergibt sich aus der Reihenfolge in den Listen.</p> <p>In der Liste der Kommando-APDUs kann vor jedem Kommando-APDU eine Liste mit erwarteten StatusCodes zu dem jeweiligen Kommando-APDU mitgeschickt werden.</p> <p>Die Liste der Rückgabe-APDUs enthält ausschließlich Rückgabe-APDUs.</p>	
Aufrufparameter	Name	Beschreibung
	TransactionData	TransactionData enthält ein Scenario wie in [api-popp] beschrieben (base64-codiert). Ein Scenario enthält eine Unterstruktur Scenario7816 und diese wiederum eine Liste bestehend aus Elementen (hier: Elements).
	dss:SignatureObject	Enthält die zu prüfende Signatur über TransactionData. Hierbei wird sie als dss:Base64Signature mit entsprechend gesetztem Type-Attribut (siehe SignatureType, Operation SignDocument) übergeben. Es MUSS CMS-Signatur und der Wert "urn:ietf:rfc:5652" unterstützt werden.
	CERTCMN:X509Certificate	Enthält das Signaturzertifikat (base64-codierte ASN.1/DER-Struktur)
Rückgabe	Name	Beschreibung
	CONN:Status	Enthält den Ausführungsstatus der Operation

Name	SecureSendAPDU	
	TransactionResult	Enthält die bae64-codierte Liste der Rückgabe-APDUs (hier: ResultList). Das Format von ResultList ist in [api-popp] beschrieben.
	TimeSpan	Zeitspanne, in der der nächste Aufruf von SecureSendAPDU mit dem nächsten Szenario der Sequenz erfolgen muss TimeSpan = 0 zeigt das letztes Szenario der Sequenz an.
Vorbedingung	keine	
Nachbedingung	keine	

Der Ablauf der Operation SecureSendAPDU ist in Tabelle TAB_KON_271 Ablauf SecureSendAPDU beschrieben.

Tabelle 13: TAB_KON_271 Ablauf SecureSendAPDU

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_161 „nonQES Dokumentensignatur prüfen“	Die nonQES wird geprüft mittels TUC_KON_161 { certificate = X509Certificate; signature = SignatureObject; signedDocument = TransactionData }. Bei der Zertifikatsprüfung mit TUC_KON_037 muss als policyList die Zertifikatstyp-OID "oid_zd_sig" übergeben werden. Des Weiteren muss geprüft werden, dass die zurückgegebene Rolle ("role") "oid_popp" ist. Tritt hierbei ein Fehler auf, bricht die Operation ab.
3.	TUC_KON_208 „Sende gesicherte APDU“	Die Kommando-APDUs werden an die Karte gesendet und das Ergebnis zurückgegeben mittels TUC_KON_208 { transactionData }

Tabelle 14: TAB_KON_272 Fehlercodes SecureSendAPDU

Fehlercode	ErrorType	Severity	Fehlertext
Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten:			
4000	Technical	Error	Syntaxfehler
4209	Technical	Error	Kartentyp %CardType% wird durch diese Operation nicht unterstützt.

[<=, Konnektor Highspeed, Konnektor PTV6, Sich.techn. Eignung: CC-Evaluierung, funkt. Eignung: Test Produkt/FA, Sich.techn. Eignung: Prüfung durch CC-Prüfstelle]

A_26022 - Operation StopCardSession

Der Konnektor MUSS an der Außenschnittstelle eine Operation StopCardSession, wie in Tabelle TAB_KON_275 Operation StopCardSession beschrieben, anbieten.

Tabelle 15: TAB_KON_275 Operation StopCardSession

Name		
Beschreibung	Die Operation beendet eine operationsübergreifende Reservierung einer Karte.	
Aufrufparameter	Name	Beschreibung
	SessionID	UUID gem. [RFC4122]
Rückgabe	Name	Beschreibung
	CONN:Status	Enthält den Ausführungsstatus der Operation.

Der Ablauf der Operation StopCardSession ist in Tabelle TAB_KON_276 Ablauf StopCardSession beschrieben.

Tabelle 16 TAB_KON_276 Ablauf StopCardSession

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_224 „Stoppe Kartensitzung“	TUC_KON_224 { sessionID = SessionID }

Tabelle 17: TAB_KON_278 Fehlercodes StopCardSession

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

[<=, ,]

Änderungen in Kapitel 4.1.5.6: In TIP1-A_4592* wird in TAB_KON_554 ein neuer Konfigurationswert aufgenommen.

Tabelle 18: TAB_KON_554 Konfiguration des Kartendienstes

ReferenzID	Belegung	Bedeutung
CARD_SESSION_TIMEOUT	[10 - 180 Sekunden]	Timeout für Inaktivität einer CardSession Default-Wert = 120 Sekunden

1.2 Anhang F - Übersicht Events

Tabelle 19 TAB_KON_777 Events Interne Mechanismen

Topic Ebene1 /Topic Ebene2 /Topic Ebene3	Typ	Schwere	Prot	An Cli ents	Parameter	Bedeutung	Auslöser (TUC/Op)
CARD /SecureSendAPD U /TIMEOUT	Op	Info	x	x	CardType=\$; SessionID=\$; Timer="Konnekto r PoPP"	Deadline für Folgeaufruf in cardSession überschritte n	SecureSendAPD U

1.3 Kapitel 5.5.1 Dokumente der gematik

Es wird an die Tabelle in dem Kapitel eine neue Zeile angehängt, in der nach der Veröffentlichung der GitHub-Link zum [api-popp] spezifiziert wird.

2 Änderungen in Steckbriefen

2.1 Änderungen in gemProdT_Kon_PTV6

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

**Tabelle 20: Anforderungen zur funktionalen Eignung
"Produkttest/Produktübergreifender Test"**

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)

2.2 Sonstige Änderungen

Es werden in api-telematik die Dateien `CardService_v8_2_0.xsd` und `CardService_v8_2_0.wsdl` neu aufgenommen.

Aktuell sind sie in Pull Request <https://github.com/gematik/api-telematik/pull/24> zu finden.

Die darin enthaltenen Änderungen müssen vom Konnektor umgesetzt werden.