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

Elektronische Gesundheitskarte und Telematikinfrastruktur

Spezifikation
Basis- und KTR-Consumer

Version:	1.~~23~~.0 CC
Revision:	~~241913~~269753
Stand:	~~30.06.2020~~17.08.20
Status:	zur Abstimmung freigegeben
Klassifizierung:	öffentlich_Entwurf
Referenzierung:	gemSpec_Basis_KTR_Consumer

Dokumentinformationen

Änderungen zur Vorversion

Anpassungen des vorliegenden Dokumentes im Vergleich zur Vorversion können Sie der nachfolgenden Tabelle entnehmen.

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	15.05.19		initiale Erstellung des Dokuments	gematik
1.1.0	28.06.19		Einarbeitung P19.1	gematik
1.2.0	30.06.~~2020~~20		Einarbeitung P22.1	gematik
1.3.0 CC	17.08.20		Einarbeitung P22.3	gematik

Inhaltsverzeichnis

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die Anforderungen an Herstellung, Test und Betrieb der beiden Produkttypen Basis-Consumer und KTR-Consumer.

Der Basis-Consumer und der KTR-Consumer sind Produkttypen der TI-Plattform, die in der Rolle eines Consumers mit der Telematikinfrastruktur (TI) interagieren und dabei sowohl Anteile der TI-Plattform als auch Anteile des sicheren Übermittlungsverfahrens KOM-LE enthalten. Der KTR-Consumer enthält darüber hinaus auch Fachmodule, um den Nutzerkreis „Krankenkassen" die Teilnahme an den für sie vorgesehenen Fachanwendungen der Telematikinfrastruktur zu ermöglichen.

1.2 Zielgruppe

Das Dokument ist maßgeblich für Anbieter und Hersteller des Produkttyps Basis- und KTR-Consumer sowie für Anbieter und Hersteller von Produkten, die die Schnittstellen des Produkttyps Basis- und KTR-Consumer nutzen.

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) fest-gelegt und bekannt gegeben.

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 den Produkttypen Basis- und KTR-Consumer bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert (siehe auch Anhang A5).

Die vollständige Anforderungslage für die Produkttypen ergibt sich aus weiteren Konzept- und Spezifikationsdokumenten, diese sind in den Produkttypsteckbriefen des Produkttyps Basis- bzw. KTR-Consumer verzeichnet.

1.5 Methodik

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 zwischen der ID und der Textmarke angeführten Inhalte.

2 Systemüberblick

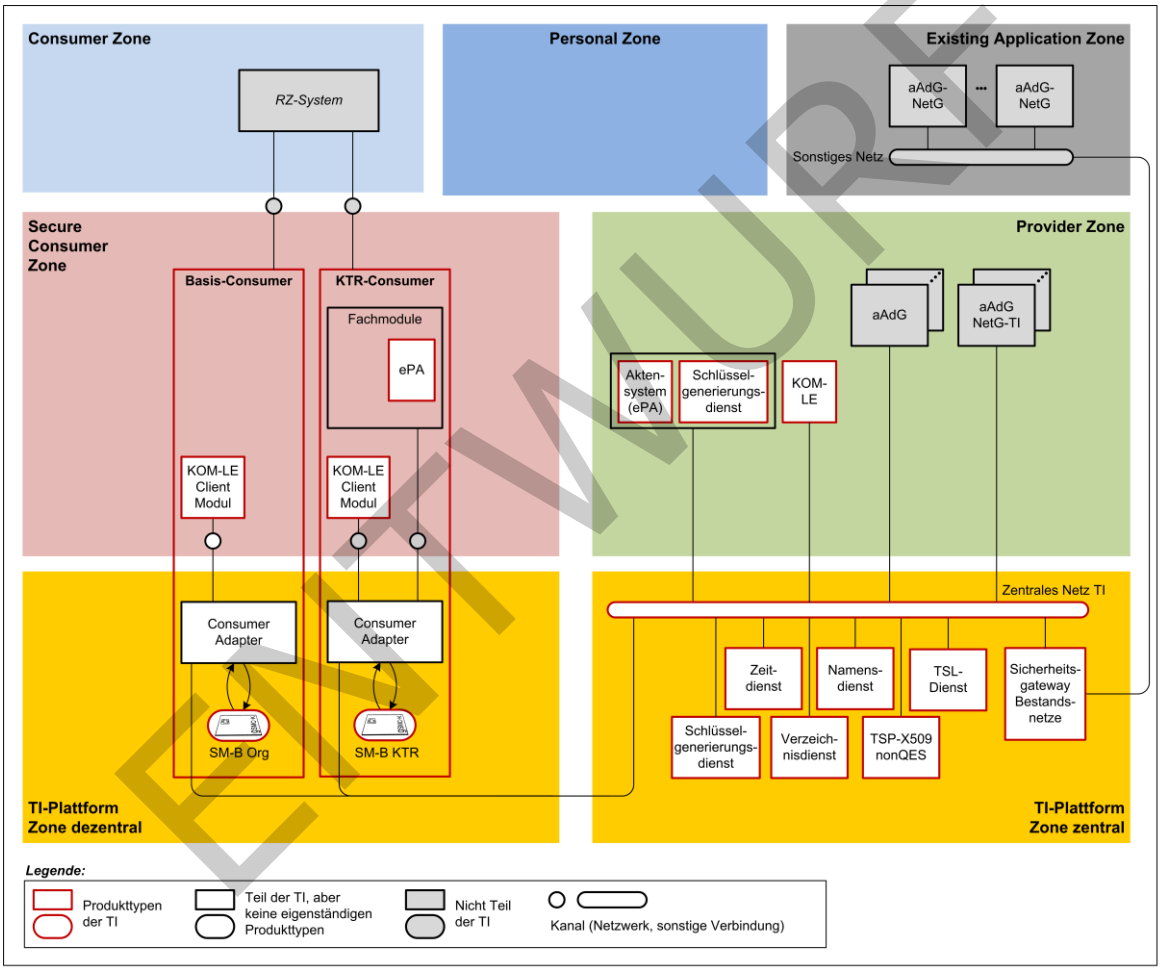
Die Produkttypen Basis- und KTR-Consumer sind beides Realisierungen des konzeptionellen Konstrukts „RZ-Consumer" aus dem [gemKPT_Arch_TIP]. D.h., sie agieren als Consumer in der Telematikinfrastruktur (TI), nutzen dabei zentrale Dienste, die Dienste des sicheren Übermittlungsverfahrens und ggf. fachanwendungsspezifische Dienste und werden in einem Rechenzentrum entsprechend den Vorgaben der TI betrieben. Beide Produkttypen bieten für externe Clients eine Menge von Basisfunktionen (z.B. kryptographische Operationen), ermöglichen den Zugriff auf weitere Anwendungen des Gesundheitswesens und die Nutzung des sicheren Übermittlungsverfahrens KOM-LE.

Der Basis-Consumer ermöglicht es den Gesellschaftern der gematik sowie den durch sie vertretenen Organisationen, als Nutzer an der TI teilzunehmen. Der Zugriff auf Fachanwendungen der TI ist dieser Nutzergruppe nicht gestattet. Der Produkttyp enthält demnach zwar keine Fachmodule, aber ein Clientmodul KOM-LE zur Nutzung des sicheren Übermittlungsverfahrens. Auf technischer Ebene wird die Nutzergruppe durch die kryptographische Identität der SMC-B Org identifiziert, die in einem HSM oder auf einer Karte gespeichert wird.

Der KTR-Consumer ermöglicht es Krankenkassen, als Nutzer an der TI teilzunehmen. Genutzt werden können dabei Fachanwendungen, bei der die Krankenkassen als berechtigte Nutzer festgelegt sind (mit Ausnahme von VSDM), die sicheren Übermittlungsverfahren und die weiteren Anwendungen des Gesundheitswesens. Dieser Produkttyp enthält Fachmodule und ein Clientmodul KOM-LE zur Nutzung des sicheren Übermittlungsverfahrens. Auf technischer Ebene wird die Nutzergruppe durch die kryptographische Identität der SMC-B KTR identifiziert, die in einem HSM gespeichert wird.

3 Systemkontext

Nachfolgend wird angelehnt an den Systemüberblick aus [gemKPT_Arch_TIP] die Einbettung der Produkttypen Basis-Consumer und KTR-Consumer in das System der TI dargestellt. Die Darstellung ist reduziert auf die Produkttypen der TI sowie Clients und Anwendungen außerhalb der TI, mit denen potentiell eine Interaktion stattfindet. Die Festlegungen des vorliegenden Dokuments beziehen sich auf die Produkttypen Basis-Consumer und KTR-Consumer als Ganzes und das logische Konstrukt des Consumer-Adapters aus [gemKPT_Arch_TIP], das den Umfang der Basisfunktionen der Produkttypen festlegt.



FMC Block Diagram
TI Architektur – KTR-Consumer
Project: TI Architekturdarstellung
Author: WOC,PTA TEC/TN | Date: 26.03.2019

Abbildung 1: Systemkontext für Basis-/KTR-Consumer

4 Zerlegung der Produkttypen

Der Produkttyp Basis-Consumer teilt sich in die folgenden Bestandteile auf:

- Basisfunktionen,
- LDAP-Proxy und
- Clientmodul KOM-LE

Der Produkttyp KTR-Consumer teilt sich in die folgenden Bestandteile auf:

- Basisfunktionen,
- LDAP-Proxy und
- Clientmodul KOM-LE
- Fachmodul ePA im KTR-Consumer

Die Festlegungen der vorliegenden Dokuments beziehen sich auf die Produkttypen Basis-Consumer und KTR-Consumer als Ganzes sowie deren oben aufgeführten Bestandteile, mit Ausnahme des Fachmoduls ePA, welches in [gemSpec_FM_ePA_KTR_Consumer] beschrieben wird. Das logische Konstrukt des Consumer-Adapters aus [gemKPT_Arch_TIP], wird durch die Basisfunktionen und den LDAP-Proxy in dem für die Produkttypen benötigten Umfang umgesetzt.

4.1 Basisfunktionen

Die Basisfunktionen enthalten:

- den Verschlüsselungsdienst zum Ver- und Entschlüsseln von Dokumenten
- den Signaturdienst zum Signieren und Signaturprüfen
- den Zertifikatsdienst, um Zertifikate zu überprüfen
- netztechnische Anbindung an die Telematikinfrastruktur (Interface, Firewall und DNS)

4.2 LDAP-Proxy

Der Basis- und KTR-Consumer ermöglicht es Clientsystemen und Clientmodulen durch Nutzung des LDAP-Proxies Daten aus dem Verzeichnisdienst der TI-Plattform (VZD) abzufragen. Die Kommunikation erfolgt über das LDAPv3-Protokoll.

4.3 Clientmodul KOM-LE

Der Basis- und KTR-Consumer enthält ein Clientmodul KOM-LE, um das sichere Übermittlungsverfahren KOM-LE nutzen zu können. Es werden die Anwendungsfälle „Senden und Empfangen von Nachrichten“ unterstützt. Die Spezifikation [gemSpec_CM_KOMLE] gilt in großen Teilen auch für den Basis- und KTR-Consumer. Es gibt aber verschiedene Bereiche, in denen eine Anpassung für den Basis- und KTR-Consumer erforderlich ist. Für diese Bereiche werden neue Anforderungen aufgenommen,

die statt der bestehenden Anforderungen aus [gemSpec_CM_KOMLE] zu verwenden sind. Die Bereiche sind:

- Nutzung des Basis- und KTR-Consumer
  Die Spezifikation des Clientmoduls [gemSpec_CM_KOMLE] schreibt an einigen Stellen die Nutzung des Konnektors für Signatur/Signaturprüfung und Ver-/Entschlüsselung vor. Diese Anforderungen werden ersetzt durch Anforderungen, die die Nutzung der Systemprozesse im Basis-/KTR-Consumer vorschreiben.
- Client-Schnittstelle des Moduls
  Die SMTP/POP3-Schnittstelle des Clientmoduls soll beibehalten werden. Abweichend von [gemSpec_CM_KOMLE] werden die Informationen bzgl. der Adresse und des Ports des Mail Transfer Agents (MTA, KOM-LE Fachdienst) und die Informationen des Aufrufkontext nicht beim Aufruf mitgegeben, sondern im Basis- und KTR-Consumer lokal konfiguriert.

5 Übergreifende Festlegungen

5.1 Anschluss an die TI

5.1.1 Anbindung per LAN/WAN

Unter Anbindung per LAN/WAN werden die Mechanismen beschrieben, mit denen der Basis- und KTR-Consumer auf der einen Seite in das lokale Netz der Einsatzumgebung und auf der anderen Seite in die zentrale TI und die aAdG und aAdG NetG-TI angebunden wird. Diese wesentlichen Aspekte betreffen Routing und Firewall.

5.1.1.1 Funktionsmerkmalweite Aspekte

A_17396 - Verhalten als IPv4-Router
Der Basis- und KTR-Consumer MUSS sich nach den in [RFC1812#1.1.3] definierten Rahmenbedingungen als IP-Version-4-(IPv4)-Router verhalten. Die in [RFC2644] geforderten Aktualisierungen zum [RFC1812] MÜSSEN umgesetzt werden.**[**<=**]**

A_17397 - IP-Pakete mit Source Route Option
Der Basis- und KTR-Consumer DARF NICHT IP-Pakete mit gesetzter Source Route Option gemäß [RFC791] erzeugen oder weiterleiten.**[**<=**]**

A_17400 - NAT-Umsetzung
Der Basis- und KTR-Consumer MUSS für die Kommunikation mit Adressbereichen der TI und aAdG und aAdG NetG-TI eine Network Address Translation (NAT) gemäß [RFC3022#2.2, 3, 4.1-4.3] vornehmen.
Für die Umsetzung der Private Local Address aus den Adressbereichen der Einsatzumgebung MUSS die verwendete IP-Adresse aus dem vom Anbieter Zentrale Plattform Dienste (AZPD) bereitgestellten Adress-Pool entnommen werden und als Global Address genutzt werden.**[**<=**]**

A_17405 - Nur IPv4. IPv6 nur hardwareseitig vorbereitet
Der Basis- und KTR-Consumer MUSS die IP Version 4 (IPv4) für alle seine IP-Schnittstellen unterstützen.
Die Hardware des Basis- und KTR-Consumer MUSS für den Einsatz von IPv4 und IPv6 im Dual-Stack-Mode geeignet sein.
Bis zu einer Migration von IPv4 auf IPv6 MUSS der Basis- und KTR-Consumer sämtliche empfangenen IP-Pakete der Version 6 (IPv6) verwerfen.**[**<=**]**

Die Anbindung des Basis- und KTR-Consumers an die zentrale TI erfolgt über einen Sicheren Zentralen Zugangspunkt (SZZP), siehe gemSpec_Net Kapitel 3.1.1. Dieser Produkttyp unterstützt kein dynamisches Routing.

A_17406 - Kein dynamisches Routing
Basis- und KTR-Consumer DÜRFEN NICHT Dynamische Routing-Protokolle einsetzen.**[**<=**]**

5.1.1.1.1 Netzwerksegmentierung

In Anlehnung an die in der [gemSpec_Net#2.3.3] definierten Netzwerksegmente werden in der Basis- und KTR-Consumerspezifikation die folgenden Bezeichner verwendet:

Tabelle 1 : Mapping der Netzwerksegmente

ReferenzID im Basis- und KTR-Consumer	Adressbereich für die TI-Produktivumgebung	Adressbereich für die TI-Testumgebung	Adressbereich für die TI-Referenzumgebung
NET_TI_ ZENTRAL	TI_Zentral - Zentrale Dienste	TI_Test_Zentral - Zentrale Dienste	Ist durch den Testbetriebsverantwortlichen zu definieren.
NET_TI_ GESICHERTE_ FD	TI_Fachdienste - Gesicherte Fachdienste	TI_Test_Fachdienste - Gesicherte Fachdienste	Ist durch den Testbetriebsverantwortlichen zu definieren.
NET_TI_OFFENE_ FD	TI_Fachdienste - Offene Fachdienste	TI_Test_Fachdienste - Offene Fachdienste	Ist durch den Testbetriebsverantwortlichen zu definieren.
NET_aAdG_aAdG NetG-TI	aAdG und aAdG NetG-TI	aAdG und aAdG NetG-TI	aAdG und aAdG NetG-TI
NET_CONSUMER	Liste der Netzwerke die in der Einsatzumgebung über den Basis- und KTR-Consumer erreichbar sind. Ein Eintrag der Liste enthält die Netzwerkadresse und den Netzwerkpräfix.		

A_17411 - Kommunikation mit NET_TI_Offene_FD
Der Basis- und KTR-Consumer MUSS sicherstellen, dass IP-Pakete mit dem Ziel NET_TI_Offene_FD und NET_aAdG_aAdG NetG-TI weitergeleitet werden.**[<=]**

A_17514 - Kommunikation mit NET_TI_Gesicherte_FD
Der KTR-Consumer MUSS sicherstellen, dass IP-Pakete mit dem Ziel NET_TI_Gesicherte_FD nur durch das im KTR-Consumer vorhandene jeweilige Fachmodul in Richtung TI mit dem Ziel NET_TI_Gesicherte_FD weitergeleitet. werden.**[<=]**

A_17415 - Kommunikation mit NET_TI_ZENTRAL
Der Basis- und KTR-Consumer MUSS sicherstellen, dass IP-Pakete in Richtung NET_TI_ZENTRAL mit dem Ziel TI-Namens- und Zeitdienst nur vom Basis- und KTR-Consumer weitergeleitet werden.**[<=]**

A_17417 - Einschränkung von nicht genehmigten Traffic
Der Basis- und KTR-Consumer MUSS nicht genehmigten Traffic blockieren.**[<=]**

A_17418 - Drop statt Reject
Der Basis- und KTR-Consumer MUSS alle abgelehnten IP-Pakete verwerfen (DROP), ohne ein ICMP-Destination-Unreachable (Type 3) zu schicken.**[<=]**

A_17419 - Abwehr von IP-Spoofing, DoS/DDoS-Angriffe und Martian Packets
Der Basis- und KTR-Consumer MUSS geeignete technische Funktionen zur Abwehr von IP-Spoofing und DoS/DDoS-Angriffen implementieren.
Der Basis- und KTR-Consumer MUSS Martian Packets (Absender- oder Empfängeradressen aus den von der IETF als Special-Purpose definierten Netzbereichen), mindestens jedoch aus folgenden Netzbereichen 0.0.0.0/8, 127.0.0.0/8, 169.254.0.0/16, 192.0.0.0/24, 192.0.2.0/24, 198.18.0.0/15, 198.51.100.0/24, 203.0.113.0/24, 224.0.0.0/4, 240.0.0.0/4, verwerfen. Die in [RFC1918] und [RFC 6598] definierten Netzbereiche sind hiervon ausgenommen.**[<=]**

A_17420 - Eingeschränkte Nutzung von „Ping"
Der Basis- und KTR-Consumer MUSS TCP-Port-7(Echo)-Pakete verwerfen.
Der Basis- und KTR-Consumer MUSS ICMP-Echo-Request (Typ 8) und ICMP-Echo-Response (Typ 0) ausschließlich für, per Anforderung genehmigten, Traffic weiterleiten.**[<=]**

A_17421 - Einschränkungen der IP-Protokolle
Der Basis- und KTR-Consumer MUSS alle IP-Protokolle außer 1 (ICMP), 17 (UDP) und 6 (TCP) für alle ein- oder ausgehenden Pakete an allen seinen Adaptern verwerfen.**[<=]**

A_17423 - Firewall Restart
Der Basis- und KTR-Consumer MUSS gewährleisten, dass unmittelbar nach einer Änderung der Parameter eines Adapters (LAN-Adapter, WAN-Adapter) die Firewall des Basis- und KTR-Consumer neu erstellt und geladen wird.**[<=]**

Umsetzungshinweis für den Hersteller: Es können zwei getrennten Firewall-Regelsets für den LAN- bzw. für den WAN-Adapter verwendet werden.

A_17424 - Firewall-Protokollierung
Der Basis- und KTR-Consumer MUSS bei Konfigurationsänderungen der Firewall einen Protokolleintrag mit der Schwere „Warning" und dem Typ „Operations" sowie mindestens folgenden Informationen generieren:

- Zeitstempel, Aktion (Add/Delete/Change), Details (Beschreibung der Änderung), Auslöser (Prozess/User).

Der Basis- und KTR-Consumer MUSS für alle vom Basis- und KTR-Consumer ausgehenden, nicht zugelassenen Kommunikationsversuche einen Protokolleintrag mit der Schwere „Warning" und dem Typ „Security" sowie mindestens folgenden Informationen generieren:

- Zeitstempel, Aktion (Drop, Reject), Absender-IP-Adresse, Empfänger-IP-Adresse, Protokoll, Absender-Port und Empfänger-Port, Interface, über die das Paket empfangen wurde.

Der Basis- und KTR-Consumer MUSS für alle verworfenen IP-Spoofing- und Martian-Packets einen Protokolleintrag mit der Schwere „Warning" und dem Typ „Security" sowie mindestens folgenden Informationen generieren:

- Zeitstempel, Aktion (Drop, Reject), Absender-IP-Adresse, Empfänger-IP-Adresse, Protokoll, Absender-Port und Empfänger-Port, Interface über das das Paket empfangen wurde.

Der Basis- und KTR-Consumer MUSS für alle weiteren von der Firewall verworfenen IP-Pakete einen Protokolleintrag mit der Schwere „Info" und dem Typ „Security" sowie mindestens folgenden Informationen generieren, wobei Layer 3 Broadcasts von der Protokollierung ausgenommen werden können:

- Zeitstempel, Aktion (Drop, Reject), Absender-IP-Adresse, Empfänger-IP-Adresse, Protokoll, Absender-Port und Empfänger-Port, Interface über das das Paket empfangen wurde.

[<=]

5.1.1.2 Durch Ereignisse ausgelöste Reaktionen

A_17425 - Reagiere auf LAN_IP_Changed
Wurde die IP Adresse des LAN Interfaces geändert oder hat, bei aktiven DHCP Client, ein erfolgreiches DHCP_RENEW stattgefunden MUSS der Basis- und KTR-Consumer den LAN-Adapter initialisieren.**[<=]**

A_17426 - Reagiere auf WAN_IP_Changed
Wurde die IP Adresse des WAN Interfaces geändert oder hat, bei aktiven DHCP Client, ein erfolgreiches DHCP_RENEW stattgefunden MUSS der Basis- und KTR-Consumer den WAN-Adapter initialisieren.**[<=]**

A_17430 - Netzwerk-Routen einrichten
Der Basis- und KTR-Consumer MUSS die Konfiguration aller notwendigen Netzwerk-Routen ermöglichen.**[<=]**

A_17474 - Anzeige IP-Routinginformationen
Der Basis- und KTR-Consumer MUSS über die Managementschnittstelle die konfigurierten IP-Routen und die aktuelle IP-Routingtabelle mit mindestens folgenden Informationen anzeigen:

- Forwarding Status
- Zieladresse/Präfix
- Gateway (Next-Hop)
- Routing Typ
- Routing Preference.

[<=]

Zur Bekanntmachung von Änderungen und Neuanschlüssen zu den, an die TI angeschlossenen, anderen Anwendungen des Gesundheitswesens (aAdG bzw. aAdG NetG-TI) wird tagesaktuell eine Datei mit dem Namen "Bestandsnetze.xml" bereitgestellt (siehe dazu gemSpec_KSR, Kapitel 9 Anhang C). Die Datei liefert für alle angeschlossenen aAdG bzw. aAdG NetG-TI einen Namen/ID, Netzwerkinformationen (IP-Adressen) und den für dieses Netz zu verwendenden DNS Server welcher dem DNS Forwarder des Basis- und KTR-Konsumer übergeben wird.

A_17576 - KSR lokalisieren
Der Basis- und KTR-Consumer MUSS für die Lokalisierung des Konfigurationsdienstes der TI (KSR) die Möglichkeit der Lokalisierung des KSR durch DNS-Anfragen an den DNS-Forwarder DNS_SERVERS_TI zur Auflösung der SRV-RR und TXT-RR mit den Bezeichnern „_ksrkonfig._tcp.ksr.<TOP_LEVEL_DOMAIN_TI>“ vorsehen.
Der Basis- und KTR-Consumer erhält damit URLs der Downloadpunkte des KSR für Konfigurationsdaten (MGM_KSR_KONFIG_URL).**[<=]**

A_17574 - Infrastruktur Konfiguration aktualisieren
Der Basis- und KTR-Consumer MUSS täglich seine Infrastruktur Konfiguration aktualisieren.
Der Basis- und KTR-Consumer MUSS dazu eine TLS-Verbindung zum Konfigurationsdienst der TI aufbauen. Dabei MUSS er das durch den Server präsentierte Zertifikat prüfen.

Das Herunterladen der Konfigurationsdaten erfolgt mittels I_KSRS_Download::get_Ext_Net_Config (MGM_KSR_KONFIG_URL, „Bestandsnetze.xml“.)**[<=]**

5.1.2 Zeitdienst

Der Zeitdienst schafft die Grundlage einer gleichen Systemzeit für alle in der TI einzusetzenden Produkttypen. Innerhalb des Basis-KTR-Consumers ist dafür ein NTP-Client erforderlich, welcher die Zeitangaben des Zeitdienstes der zentralen TI abfragt und verwendet. Die in [gemSpec_Net#6.2.2] „Nutzung“ getroffenen Anforderungen werden durch dieses Kapitel erweitert.

A_17485 - Maximale Zeitabweichung
Der Basis- und KTR-Consumer MUSS sicherstellen, dass der maximale zulässige Fehler von +/- 20ppm (part per million) gegenüber einer Referenzuhr nicht überschritten wird. Dies entspricht einer maximalen Abweichung im Freilauf von +/- 34,56 Sekunden über 20 Tage.**[<=]**

5.1.3 Namensdienst und Dienstlokalisierung

5.1.3.1 Funktionsmerkmalweite Aspekte

A_17498 - Grundlagen des Namensdienstes
Der Basis- und KTR-Consumer MUSS die Funktion eines Recursive Caching Nameservers zur Auflösung von DNS-Anfragen anbieten. (Im Folgenden kurz DNS-Server genannt). Der Caching-Nameserver des Basis- und KTR-Consumer MUSS für Clientsysteme aus dem lokalen Netzwerk der Einsatzumgebung erreichbar sein.
Der Caching Nameserver des Basis- und KTR-Consumer MUSS einen sinnvollen Timeout für die Bearbeitung von DNS-Abfragen beachten. Konnte eine DNS-Abfrage nicht durchgeführt werden, MUSS die Bearbeitung abgebrochen werden. **[<=]**

A_17499 - DNS-Forwards des DNS-Servers
Der DNS-Server des Basis- und KTR-Consumer MUSS die folgenden DNS-Forwards durchführen:

Tabelle 2 : TAB_CONS_687 DNS-Forwards des DNS-Servers

Domain	Forwarders	Bemerkungen
Namensraum TI (*.DNS_TOP_ LEVEL_DOMAIN_TI)	DNS_SERVERS_TI	DNS Forward Rule zur Auflösung aller DNS-Namen innerhalb des Namensraums der TI.
Namensraum angeschlossene Netze des Gesundheitswesens mit aAdG-NetG (Domainnamen von angeschlossenen Netzen des	DNS_SERVERS_BESTANDSNETZE (Je Domainnamen eines angeschlossenen Netzes des Gesundheitswesens mit aAdG-NetG alle zugehörigen DNS-Server IP-Adressen gemäß Bestandsnetze.xml)	Je angeschlossenem Netz des Gesundheitswesens mit aAdG-NetG in NLW_AKTIVE_BESTANDSNETZE wird eine DNS Forward Rule zur Auflösung von DNS-Namen innerhalb dieses Netzes verwendet.

Gesundheitswesens mit aAdG-NetG gemäß Bestandsnetze.xml)		
Namensraum lokale Einsatzumgebung	DNS_SERVERS_CONSUMER	DNS Forward Rule zur Auflösung aller DNS-Namen innerhalb der DNS-Domain im LAN des Consumer

[<=]

A_17500 - DNS Stub-Resolver
Der Basis- und KTR-Consumer MUSS von allen internen Diensten zur Namensauflösung genutzt werden.
Der Stub-Resolver im Basis- und KTR-Consumer MUSS immer den Caching Nameserver im Basis- und KTR-Consumer anfragen.**[<=]**

5.1.3.2 Interne TUCs, auch durch Fachmodule nutzbar

5.1.3.2.1 TUC_CON_362 „Liste der Dienste abrufen"

A_17502 - TUC_CON_362 „Liste der Dienste abrufen"
Der Basis- und KTR-Consumer MUSS den technischen Use Case TUC_CONS_362 „Liste der Dienste abrufen" umsetzen.

Tabelle 3: TAB_CONS_648 – TUC_CONS_362 „Liste der Dienste abrufen"

Element	Beschreibung
Name	TUC „Liste der Dienste abrufen"
Beschreibung	Ermittlung aller zu einer DNS-SD-Gruppe gehörenden DNS-Namen.
Auslöser	interne Anfrage (Basisdienst oder Fachmodul)
Vorbedingungen	Die vom Basis- und KTR-Consumer zu verwendenden DNS-Server müssen konfiguriert sein.
Eingangsdaten	FQDN des PTR Resource Records
Komponenten	Basis- und KTR-Consumer
Ausgangsdaten	LIST_OF_SRV_ENTITIES

Standardablauf	Mit dem FQDN wird eine Typ „PTR" Anfrage an den Stub-Resolver des Basis- und KTR-Consumer gestellt.

[<=]

5.1.3.3 Operationen an der Außenschnittstelle

A_17509 - Basisanwendung Namensdienst
Der Basis- und KTR-Consumer MUSS für Clients in der Einsatzumgebung und den Fachmodulen im jeweiligen Consumer eine Basisanwendung Namensdienst, mit der Funktion Namensauflösung und Dienstlokalisierung anbieten.

Tabelle 4: Basisanwendung Namensdienst

Name	**Namensdienst**	
Version	wird im Produktsteckbrief des Basis- und KTR-Consumer definiert	
Namensraum	Keiner	
Namensraum-Kürzel	Keiner	
Operationen	Name	Kurzbeschreibung
	GetIPAddress	Diese Operation ermöglicht die Auflösung von FQDNs in IP-Adressen
WSDL	Keines	
Schema	Keines	

[<=]

5.1.3.4 Betriebsaspekte

A_17512 - Initialisierung „Namensdienst und Dienstlokalisierung"
Der Basis- und KTR-Consumer MUSS in der Bootup-Phase zur Initialisierung des Funktionsmerkmals „Namensdienst und Dienstlokalisierung":

- den autoritativen Nameserver starten
- den Caching-Nameserver starten.

[<=]

A_17513 - Konfigurationsparameter Namensdienst und Dienstlokalisierung
Der Administrator des Basis- und KTR-Consumer MUSS die aufgelisteten Parameter in Tabelle 5 über die Managementschnittstelle konfigurieren und die aufgelisteten Parameter in Tabelle 6 ausschließlich einsehen können.
Nach jeder Änderung MUSS sichergestellt werden, dass die Änderungen sofort am autoritativen bzw. am Caching Nameserver zur Verfügung stehen.

Tabelle 5: Konfigurationsparameter Namensdienst

ReferenzID	**Belegung**	**Bedeutung und Administrator-Interaktion**

DNS_SERVERS_ CONSUMER	Liste von IP-Adressen der DNS-Server	Liste von DNS-Servern, die zur Namensauflösung von Namensräumen in der Einsatzumgebung verwendet werden. Der Administrator MUSS die Liste von DNS-Servern, die die DNS_DOMAIN_CONSUMER auflösen, bearbeiten können. Die IP-Adressen der DNS-Server KÖNNEN auf den Adressbereich der ANLW_LAN_IP_ADDRESS eingeschränkt sein.
DNS_DOMAIN_ CONSUMER	DNS Domainname	DNS Domainname, der von einem DNS-Server der Einsatzumgebung aufgelöst wird. Der Name DARF NICHT mit einem „.“ beginnen.

Tabelle 6: Einsehbare Konfigurationsparameter Namensdienst

ReferenzID	Belegung	Bedeutung
DNS_SERVERS_TI	Liste von IP-Adressen der DNS-Server	Liste von DNS-Servern, die zur Namensauflösung des Namensraums der TI verwendet werden
DNS_TOP_LEVEL_ DOMAIN_TI	DNS Domainname	Top Level Domain des Namensraumes TI

[<=]

5.2 Sicherheit

Die Sicherheits- und Datenschutzanforderungen sind abgedeckt durch die übergreifenden Sicherheits- und Datenschutzanforderungen an Hersteller und Anbieter [gemSpec_DS_Hersteller], [gemSpec_DS_Anbieter], die spezifischen Sicherheits- und Datenschutzanforderungen des Clientmoduls KOM-LE und des Fachmoduls ePA im KTR-Consumer [gemSpec_FM_ePA_KTR_Consumer] sowie die spezifischen Sicherheits- und Datenschutzanforderungen der Systemprozesse der dezentralen TI [gemSpec_Systemprozesse_dezTI].

5.3 Identitäten

In diesem Dokument werden kryptographische Identitäten entsprechend ihrer Bezeichner im Objektsystem der SMC-B referenziert. Dies dient der Eindeutigkeit der Referenz und bedeutet nicht, dass die Strukturen des Objektsystems der SMC-B in einem HSM nachgebildet werden müssen.

Im KTR-Consumer werden private Schlüssel der SMC-B, aber auch Schlüsselmaterial des KOM-LE-Clientmoduls in einem HSM gespeichert. Im Basis-Consumer werden private Schlüssel der SMC-B in einem HSM oder auf einer SMC-B in Kartenform gespeichert. Das Schlüsselmaterial des KOM-LE-Clientmoduls hingegen wird auch hier in einem HSM gespeichert.

Nachfolgend wird festgelegt, welche Qualitäten dabei erreicht werden müssen und was bei der Personalisierung zu beachten ist.

A_17598 - Qualität des HSM
Die Basis- und KTR-Consumer MÜSSEN privates Schlüsselmaterial zu Zertifikaten der Telematikinfrastruktur in einem HSM, dessen Eignung durch eine erfolgreiche Evaluierung nachgewiesen wurde, integritätsgeschützt und vertraulich speichern. Als Evaluierungsschema kommen dabei Common Criteria oder Federal Information Processing Standard (FIPS) in Frage. Die Prüftiefe MUSS mindestens (a) FIPS 140-2 Level 3, oder (b) Common Criteria EAL 4 entsprechen.**[<=]**

A_18195 - Basis-Consumer mit SMC-B
Der Basis-Consumer KANN privates Schlüsselmaterial einer SMC-B in Kartenform nutzen.**[<=]**

Tabelle 7: Tab_Personalisierung_HSM – Personalisierung des HSM

Aspekt	Beschreibung
Schlüsselmaterial der SMC-B	Das Schlüsselmaterial wird sicher im HSM erzeugt. Das private Schlüsselmaterial verlässt das HSM nicht oder nur zum Zwecke eines Backups auf einem Backup-HSM, wobei die Übertragung hinsichtlich Vertraulichkeit geschützt sein muss.
Zertifikatsrequest	Die benötigten Zertifikatsrequests werden im HSM erzeugt und exportiert. Die Zertifikatsrequests werden unter Wahrung der Authentizität und Integrität dem TSP übermittelt.
Zertifikat	Das Zertifikat wird vom TSP zum Betreiber übermittelt.
TLS-Schlüsselmaterial des KOM-LE-Clientmoduls	Der KOM-LE-Anbieter erzeugt die Schlüsselpaare für die Zertifikate des KOM-LE-Clientmoduls und bezieht aus der Komponenten-PKI der TI die C.CM.TLS-CS-Zertifikate. Das Schlüsselpaar muss zur sicheren Speicherung ins HSM eingebracht werden.

A_17599 - Personalisierung des HSM
Der Anbieter des Basis- oder KTR-Consumers MUSS einen sicheren Prozess zur Personalisierung des HSMs definieren und etablieren, der die in Tab_Personalisierung_HSM genannten Aspekte beinhaltet.**[<=]**

A_18196 - Personalisierung des HSM beim Basis-Consumer
Der Anbieter eines Basis-Consumers, der ausschließlich mit SMC-Bs in Kartenform arbeitet, KANN auf einen Prozess zur Personalisierung der Identitäten der SMC-B im HSM verzichten.**[<=]**

5.4 Schnittstellen

Für den Basis- und KTR-Consumer werden einheitliche Schnittstellen definiert und im Rahmen des Zulassungstests genutzt. Für eine bessere Integrationsfähigkeit ist es aber erlaubt, dass zusätzlich zu den definierten Schnittstellen auch weitere Schnittstellentechnologien genutzt werden können, über welche die festgelegten Operationen angesprochen werden können.

A_17712 - Zusätzlich alternative Schnittstellentechnologien
Der Basis- und KTR-Consumer KANN zusätzlich zu den in den Spezifikationen festgelegten Schnittstellen zusätzlich weitere Schnittstellentechnologien anbieten, über welche die festgelegten Operationen angesprochen werden können.**[<=]**

6 Funktionsmerkmale

6.1 Verschlüsselungsdienst

6.1.1 Durch Module nutzbare TUCs

A_17466 - Systemprozess PL_TUC_HYBRID_ENCIPHER
Der Basis- und KTR-Consumer MUSS den Systemprozess PL_TUC_HYBRID_ENCIPHER implementieren und bereitstellen.**[<=]**

A_17467 - Systemprozess PL_TUC_HYBRID_DECIPHER
Der Basis- und KTR-Consumer MUSS den Systemprozess PL_TUC_HYBRID_DECIPHER implementieren und bereitstellen.**[<=]**

6.1.2 Operationen an der Clientschnittstelle

A_17477 - Basisdienst Verschlüsselungsdienst
Der Basis- und KTR-Consumer MUSS für Clients einen Basisdienst Verschlüsselungsdienst anbieten.

Tabelle 8: Tab_Verschlüsselungsdienst

Name	**EncryptionService**	
Version	Siehe Anhang	
Namensraum	Siehe Anhang	
Namensraum-Kürzel	CRYPT für Schema und CRYPTW für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	EncryptDocument	Dokument hybrid verschlüsseln
	DecryptDocument	Dokument hybrid entschlüsseln
WSDL	EncryptionService.wsdl	
Schema	EncryptionService.xsd	

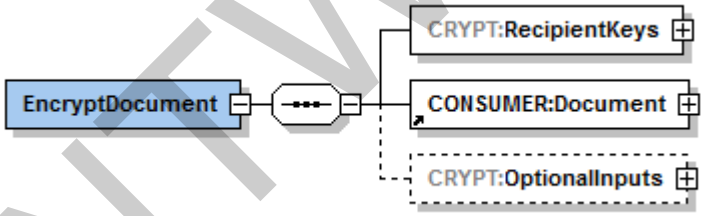
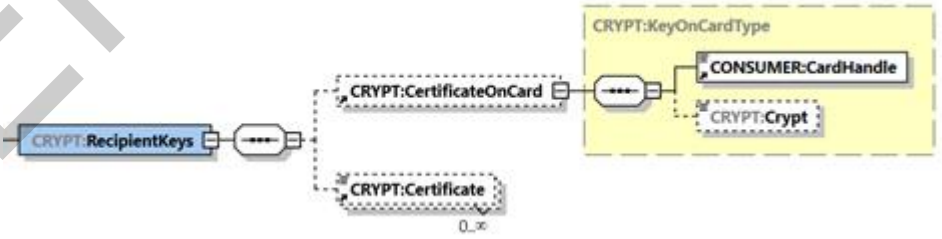
[<=]

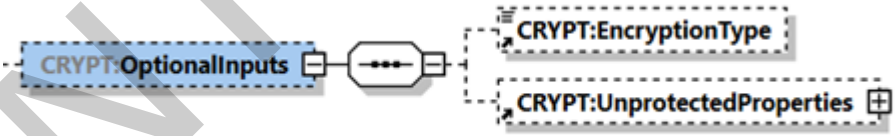
6.1.2.1 EncryptDocument

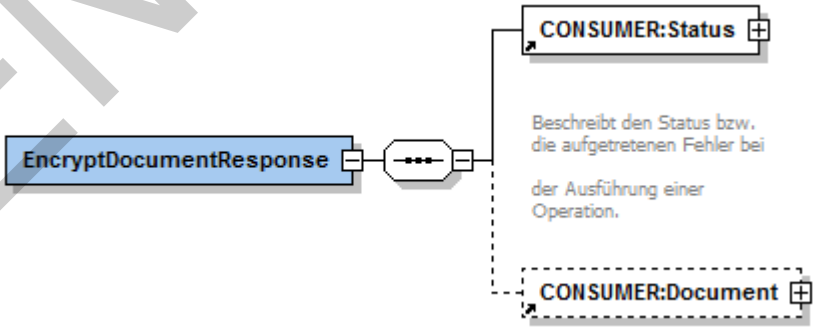
A_17510-03~~A_17510-02~~ - Basis- und KTR-Consumer, Operation EncryptDocument

Der Verschlüsselungsdienst des Basis- und KTR-Consumer MUSS an der Clientschnittstelle eine Operation EncryptDocument anbieten.

Tabelle 9: Tab_Operation_EncryptDocument

Name	`EncryptDocument`	
Beschreibung	Diese Operation verschlüsselt ein übergebenes Dokument hybrid. Der Dokumententyp XML wird gesondert behandelt. Alle anderen Dokumententypen nutzen die binäre Verschlüsselung Für die hybride Verschlüsselung wird ein asymmetrischer Schlüssel aus einem X.509v3-Zertifikat genutzt. Dieses Zertifikat wird als Parameter übergeben oder auf dem HSM referenziert. Pro Operationsaufruf können mehrere Hybridschlüssel erzeugt werden. Durch das Zertifikat wird festgelegt, ob RSA oder ECC basierte Hybridschlüssel erzeugt werden. Bei Angabe der Zertifikate über `CertificateOnCard` (Referenz auf HSM) wird das Verschüsselungsverfahren durch die Angabe in `Crypt` bestimmt. Es können Hybridschlüssel für RSA oder ECC oder beide Verfahren erzeugt werden. Für alle Dokumententypen wird immer das gesamte Dokument verschlüsselt.	
Aufrufparameter		
		
	`RecipientKeys`	Identifiziert die Empfänger der zu verschlüsselnden Nachricht über X.509-Zertifikate (öffentliche Schlüssel). Quelle für die Zertifikate kann eine Karte sein, die per CertificateOnCard-Element referenziert wird, oder der Aufrufer, der X.509-Zertifikate im Certificate-Element übergibt.
	`CardHandle`	Identifiziert die zu verwendende Karte mit dem (öffentlichen) Schlüssel.

		Ist das Element nicht vorhanden, so werden nur Zertifikate per Element `Certificate` übergeben.
	`Crypt`	Der Wert dieses Parameters ist in Tabelle Tab_KeyReference_für_Encrypt/Decrypt spezifiziert und gibt den Typ von Zertifikaten und dadurch das Verfahren für die Erzeugung der Hybridschlüssel vor. (Default-Wert ist RSA)
	`Certificate`	`Certificate` ist ein Base64-kodiertes XML-Element, in dem das Zertifikat, das den asymmetrischen Schlüssel enthält (öffentlicher Schlüssel), DER-kodiert übergeben wird. Es kann eine Liste von Zertifikaten übergeben werden. Dieses Element kann leer sein, wenn ausschließlich Zertifikate verwendet werden sollen, die über `CertificateOnCard` angegeben werden.
	`CONSUMER:Document`	Dieses entsprechend [OASIS-DSS] Section 2.4.2 spezifizierte Element enthält das zu verschlüsselnde Dokument, wobei das Kindelement `dss:Base64Data` ~~und~~ oder `CONSUMER:Base64XML` verwendet wird. Das zugeordnete Verschlüsselungsverfahren ist • XMLEnc: „http://www.w3.org/TR/xmlenc-core/" für `CONSUMER:Base64XML` • CMS: „urn:ietf:rfc:5652" für `dss:Base64Data`
		
	`CRYPT:OptionalInputs`	Enthält ~~eine Auswahl der folgenden unten näher erläuterten (~~die optionalen~~) Eingabeparameter:~~ Parameter `CRYPT:UnprotectedProperties` und `CRYPT:EncryptionType`.
	`EncryptionType`	~~Zu wählendes~~ Dieses optionale Element bestimmt das Verschlüsselungsverfahren~~, wobei folgende URI vorgesehen sind:~~ • ~~.~~ Es MUSS das Verfahren XMLEnc: „http://www.w3.org/TR/xmlenc-core~~/"~~ • ~~/"~~ unterstützt werden, wenn das Dokument in `CONSUMER:Base64XML` übergeben wird und CMS: „urn:ietf:rfc:5652~~"~~

		~~Im Fall XMLEnc wird ein Base64-codiertes XML-~~", wenn das Dokument in ~~CONSUMER:Document/CONSUMER:Base64XML übergeben.~~ ~~Im Fall CMS wird ein Base64-codiertes Binär-Dokument im Element~~ ~~CONSUMER:Document/~~dss:Base64Data übergeben~~. Ist~~ wird. Die Verwendung dieses Elements ist aufgrund der ~~Parameter EncryptionType~~impliziten Zuordnung der Verschlüsselungsverfahren zur Methode der Dokumentübergabe nicht ~~gesetzt, wird anhand des für die Übergabe des Dokuments verwendeten Elements (CONSUMER:Base64XML oder dss:Base64data) das Verfahren XMLEnc, bzw. CMS angewendet.~~erforderlich.
	CRYPT: Unprotected Properties	Dieses optionale Element wird ~~im~~nur für das Verschlüsselungsverfahren CMS ~~Fall (EncryptionType = urn:ietf:rfc:5652)~~ ausgewertet~~.~~ (zu verschlüsselndes Dokument ist in dss:Base64Data vorhanden). Die Elemente ./UnprotectedProperties/Property/Value/CMSAttribute müssen base64/DER-kodiert ein vollständiges ASN.1-Attribute enthalten, definiert in [CMS# 9.1.AuthenticatedData Type]. Es muss bei der Erstellung des CMS-Containers unter "unauthAttrs" aufgenommen werden. Das zugehörige Element ./UnprotectedProperties/Property/Identifier wird nicht ausgewertet.
Rückgabe		
	Status	Enthält den Ausführungsstatus der Operation.
	Document	Enthält das verschlüsselte Dokument in Base64-codierter Form, wenn die Verschlüsselung erfolgreich durchgeführt wurde. Im Fall XMLEnc wird das verschlüsselte XML-Dokument in CONSUMER:Document/CONSUMER:Base64XML

		zurückgegeben. Im Fall CMS wird das verschlüsselte Dokument in `CONSUMER:Document/dss:Base64data` zurückgegeben.
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Vor der Verwendung für die Verschlüsselung MÜSSEN Zertifikate durch den Aufruf von PL_TUC_PKI_VERIFY_CERTIFICATE auf ihre Gültigkeit geprüft werden. Abgelaufene oder gesperrte Zertifikate MÜSSEN von der Verwendung ausgeschlossen werden.

Das Verschlüsseln erfolgt durch Aufruf von PL_TUC_HYBRID_ENCIPHER {
Doc, das zu verschlüsselnde Dokument = `CONSUMER:Document;`
{Cert(i)}, „Menge der Empfänger-/Ziel-Zertifikate" = `RecipientKeys;`
Attribute, optionale, zusätzliche Attribute = `UnprotectedProperties;`
}
Wird ein Zertifikat per CertificateOnCard-Element referenziert, ist dieses vorher durch den HSMProxy zu extrahieren

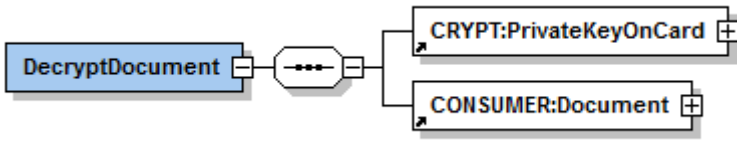
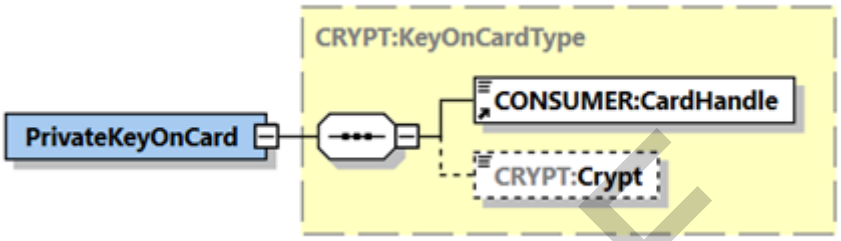
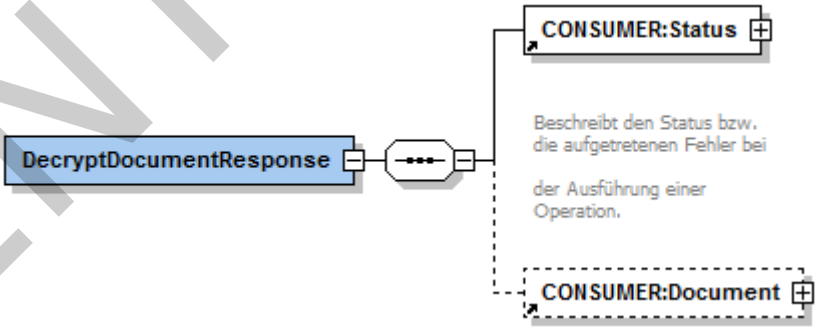
[<=]

6.1.2.2 DecryptDocument

A_17515-02~~A_17515-01~~ - Basis- und KTR-Consumer, Operation DecryptDocument
Der Verschlüsselungsdienst des Basis- und KTR-Consumer MUSS an der Clientschnittstelle eine Operation `DecryptDocument` anbieten.

Tabelle 10: Tab_Operation_DecryptDocument

Name	`DecryptDocument`
Beschreibung	Diese Operation entschlüsselt ein hybrid verschlüsseltes Dokument. Es werden die Dokumententypen XML und Andere (Binär) unterstützt. Für die Entschlüsselung wird ein asymmetrischer Schlüssel zu einem X.509v3-Zertifikat genutzt. Das Kryptoverfahren (RSA oder ECC) wird durch den Hybridschlüssel des verschlüsselten Dokuments bestimmt. Liegt eine Verschlüsselung sowohl für RSA, als auch ECC vor, erfolgt vorrangig eine Entschlüsselung mittels des ECC-Schlüssels. ~~Ist dieses nicht erfolgreich oder nicht anwendbar, erfolgt die Entschlüsselung mittels des RSA-Schlüssels.~~

<table>
<tr><td rowspan="5">Aufrufparameter</td><td colspan="2"> 
</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td>PrivateKeyOnCard</td><td>Identifiziert die zu verwendende Karte mit dem (privaten) Schlüssel.</td></tr>
<tr><td>CardHandle</td><td>Identifiziert die Karte.</td></tr>
<tr><td>Crypt</td><td>Wird nicht verwendet. Die Auswahl des Kryptoverfahrens erfolgt anhand des Hybridschlüssels des verschlüsselten Dokuments..</td></tr>
<tr><td>CONSUMER:Document</td><td>Enthält das base64-codierte Dokument, das entschlüsselt werden soll.</td></tr>
<tr><td rowspan="3">Rückgabe</td><td colspan="2"> 
</td></tr>
<tr><td>Status</td><td>Enthält den Ausführungsstatus der Operation.</td></tr>
<tr><td>Document</td><td>Enthält das entschlüsselte Dokument in Base64-codierter Form.
Im Fall der Verschlüsselung mit XMLEnc wird das entschlüsselte XML-Dokument in CONSUMER:Document/CONSUMER:Base64XML zurückgegeben.
Im Fall der Verschlüsselung mit CMS wird das entschlüsselte Dokument in</td></tr>
</table>

		CONSUMER:Document/dss:Base64data zurückgegeben.
Vorbedingungen	Keine	
Nachbedingungen	Keine	

Das Entschlüsseln erfolgt durch Aufruf von PL_TUC_HYBRID_DECIPHER {
D, "das verschlüsselte Dokument =CONSUMER:Document;
Id, "(Identität des) Empfänger" =PrivateKeyOnCard;
}
[<=]

Tabelle 11: Tab_KeyReference_für_Encrypt/Decrypt

Karte	Crypt (Wert)	**KeyReference (Encrypt)**	**KeyReference (Decrypt)**
		In DF.ESIGN	In DF.ESIGN
SM-B (HSM)	RSA	EF.C.HCI.ENC.R2048	PrK.HCI.ENC.R2048
	ECC	EF.C.HCI.ENC.E256	PrK.HP.ENC.E256
	RSA_ECC	EF.C.HCI.ENC.R2048 EF.C.HCI.ENC.E256	PrK.HCI.ENC.R2048 PrK.HP.ENC.E256

6.2 Signaturdienst

6.2.1 Durch Module nutzbare TUCs

A_17517 - Systemprozess PL_TUC_SIGN_DOCUMENT_nonQES
Der Basis- und KTR-Consumer MUSS den Systemprozess PL_TUC_SIGN_DOCUMENT_nonQES implementieren und bereitstellen.**[<=]**

A_17518 - Systemprozess PL_TUC_SIGN_HASH_nonQES
Der Basis- und KTR-Consumer MUSS den Systemprozess PL_TUC_SIGN_HASH_nonQES implementieren und bereitstellen.**[<=]**

A_17577 - Systemprozess PL_TUC_VERIFY_DOCUMENT_nonQES
Der Basis- und KTR-Consumer MUSS den Systemprozess PL_TUC_VERIFY_DOCUMENT_nonQES implementieren und bereitstellen.
[<=]

6.2.2 Operationen an der Clientschnittstelle

A_17523 - Basisdienst Signaturdienst
Der Basis- und KTR-Consumer MUSS Clientsystemen einen Basisdienst Signaturdienst (nonQES) anbieten.

Tabelle 12: Tab_Signaturdienst

Name	**SignatureService**	
Version	Siehe Anhang	
Namensraum	Siehe Anhang	
Namensraum-Kürzel	SIG für Schema und SIGW für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	SignDocument	Dokument signieren
	VerifyDocument	Signatur verifizieren
	ExternalAuthenticate	Binärstring signieren
WSDL	SignatureService.wsdl	
Schema	SignatureService.xsd	

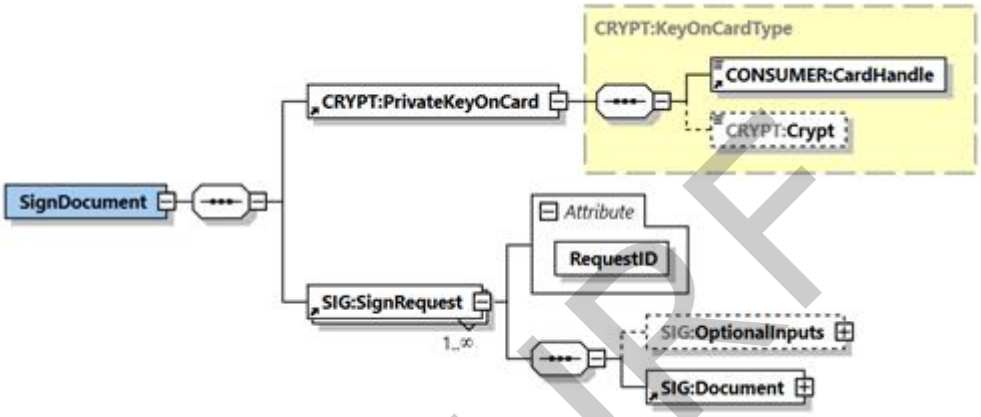
[<=]

6.2.2.1 SignDocument

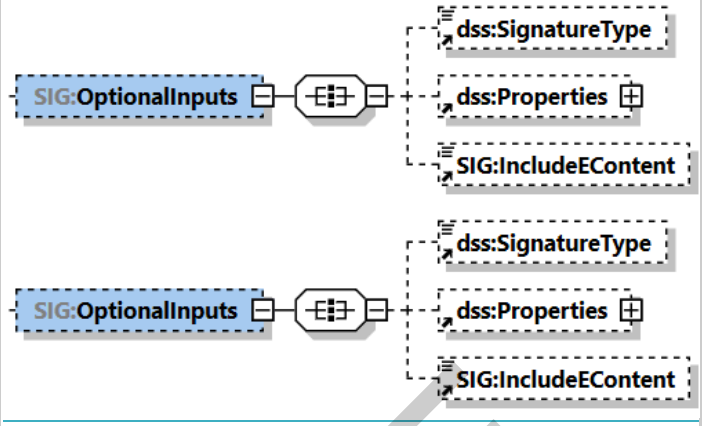
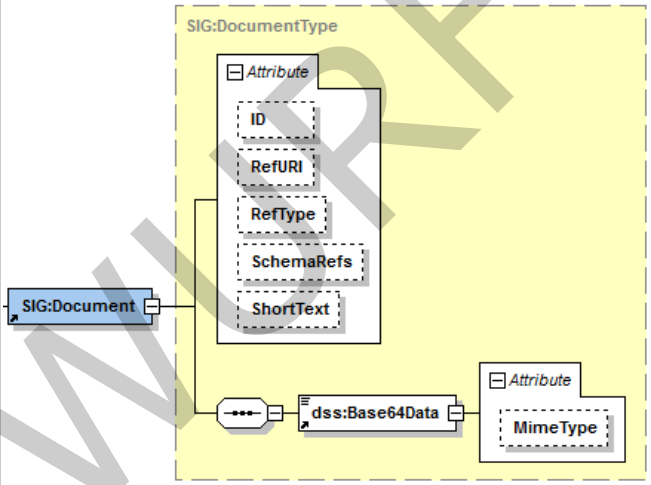
A_17525-02~~A_17525-01~~ - Basis- und KTR-Consumer, Operation SignDocument
Der Signaturdienst des Basis- und KTR-Consumer MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation SignDocument wie in Tabelle Tab_Operation_SignDocument beschrieben anbieten.

Tabelle 13: Tab_Operation_SignDocument

Name	**SignDocument**

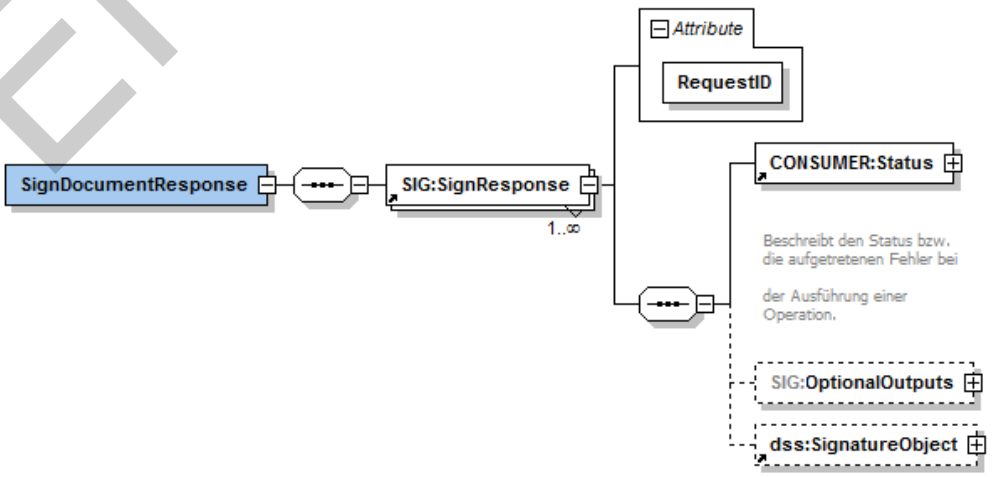
Beschreibung	Diese Operation lehnt sich an [OASIS-DSS] an. Sie enthält voneinander unabhängige SignRequests. Jeder SignRequest erzeugt eine Signatur für ein Dokument. Zur Signaturerzeugung werden Schlüssel und Zertifikate eines HSM benutzt. Es wird ausschließlich der Signaturtyp "CMS-Signatur" ~~verwendet.~~gemäß [RFC 5652] (URIurn:ietf:rfc:5652) und das Profil CAdES-BES gemäß[CAdES] verwendet.
Aufruf-parameter	

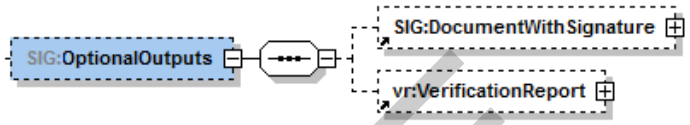
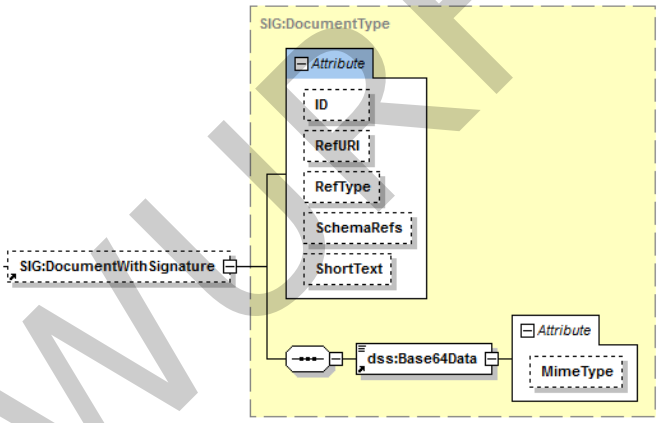
`PrivateKeyOnCard`	Identifiziert die zu verwendende Karte mit dem (privaten) Schlüssel.
`CardHandle`	Identifiziert die zu verwendende Signaturkarte.
`Crypt`	Dieser Parameter steuert die Auswahl der Zertifikate und Schlüssel für die Signaturerstellung. Die Werte sind in der Tabelle Tab_Zertifikate_für_Sign/VerifyDocument vorgegeben. (Default-Wert ist RSA)
`SIG:SignRequest`	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 das Kindelement~~ in dss:Base64Data ~~auftreten kann.~~ .
	dss:SignatureType	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element kann der generelle Typ der zu erzeugenden Signaturen ~~spezifiziert~~angegeben werden. ~~Hierbei MUSS folgender~~ Es muss der Signaturtyp ~~unterstützt werden:~~ 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.~~ ~~(Siehe Tabelle Tab_Default-Signaturverfahren))~~

		unterstützt werden. Fehlt dieses Element, so ~~wird dieser~~muss der Signaturtyp CMS-Signatur (URI urn:ietf:rfc:5652) implizit verwendet werden.

	`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
		.
		Es dürfen genau die folgenden Attribute
		`./SignedProperties/Property/Value/CMSAttribute`
		und
		`./UnsignedProperties/Property/Value/CMSAttribute`
		enthalten sein.
		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:IncludeEContent`	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. Fehlt dieses Element oder ist der Wert = "'false', wird die Signaturvariante "detached" verwendet, ansonsten "enveloping".
Rückgabe		
	`SIG:SignResponse`	Eine `SignResponse` kapselt den ausgeführten Signaturauftrag pro Dokument. Die Zuordnung

		zwischen `SignRequest` und `SignResponse` erfolgt über die RequestID.
	`CONSUMER:Status`	Enthält den Status der ausgeführten Operation pro SignRequest.
	`SIG:OptionalOutputs`	Enthält ~~(angelehnt an `dss:OptionalOutputs`)~~ optionale Ausgangsparameter~~:~~. Dieses Element wird durch den Basis- und KTR-Consumer nicht befüllt. 
	`SIG:DocumentWithSignature`	 ~~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 `Base64Data` zurückgegeben. Die Signatur wird dann im Element `dss:SignatureObject` abgelegt.~~ ~~Wenn die Signatur im Dokument enthalten ist, wird das signierte Dokument im Feld `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.~~Dieses Element wird durch den Basis- und KTR-Consumer nicht befüllt.
	`vr:VerificationReport`	~~Vom~~Dieses Element wird durch den Basis- und KTR-Consumer nicht befüllt.

	dss:SignatureObject	Enthält im Erfolgsfall die erzeugte Signatur in Form eines dss:SignatureObject-Elements gemäß [OASIS-DSS] (Abschnitt 3.2). Der Signaturwert wird im XML-Element dss:SignatureObject/dss:Base64Signature übergeben. Der Signatur-Typ (CMS Signatur) in dss:SignatureObject/dss:Base64Signature/@Type Die XML-Elemente dss:SignatureObject/ds:Signature dss:SignatureObject/dss:Timestamp dss:SignatureObject/dss:SignaturePtr dss:SignatureObject/dss:Other werden nicht verwendet.
Vorbedingungen	Keine	
Nachbedingungen	Keine	

~~Tabelle 14: Tab_Default-Signaturverfahren~~

~~Dokument-Format~~	~~Signaturverfahren (und -variante)~~			
	~~Signaturverfahren~~	~~Signaturvariante~~	~~WAS wird signiert?~~	~~WO wird die Signatur abgelegt?~~
~~Alle~~	~~CAdES~~	~~detached~~	~~gesamtes Binärdokument~~	~~außerhalb des Dokuments in der SignResponse~~

Das Signieren erfolgt durch Aufruf von PL_TUC_SIGN_DOCUMENT_nonQES {
  IDENTIFIKATOR = PrivateKeyOnCard;
  DOKUMENT = SIG:Document;
  DOKUMENTTYPE = dss:SignatureType;
}

Die folgende Tabelle führt die zulässigen Zertifikate und Schlüssel für die nonQES auf:

Tabelle 14: Tab_Zertifikate_für_Sign/VerifyDocument(nonQeS)

Karte	Crypt (Wert)	**KeyReference (Verify)**	**KeyReference (Sign)**
		in DF.ESIGN	in DF.ESIGN
	RSA	EF.C.HCI.OSIG.R2048	PrK.HCI.OSIG.R2048

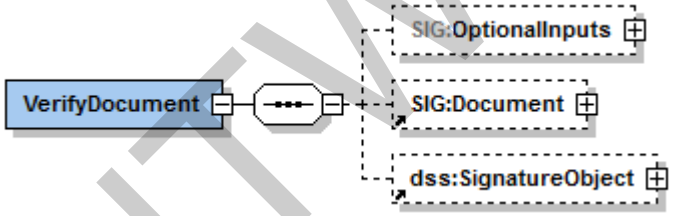
SM-B (KTR/Org) (HSM)	ECC	EF.C.HCI.OSIG.E256	PrK.HCI.OSIG.E256
	RSA_ECC	EF.C.HCI.OSIG.R2048 EF.C.HCI.OSIG.E256	PrK.HCI.OSIG.R2048 PrK.HCI.OSIG.E256

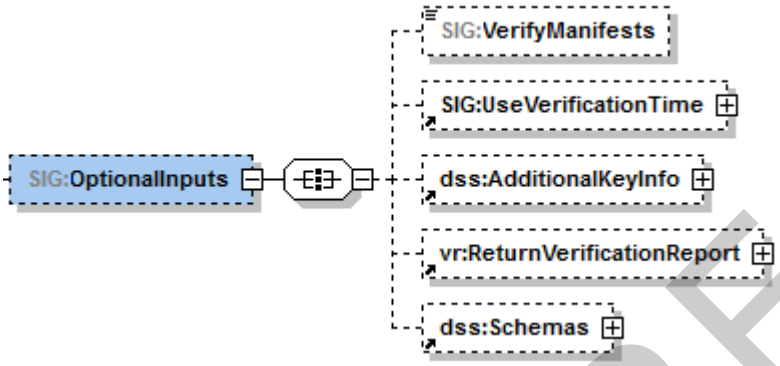
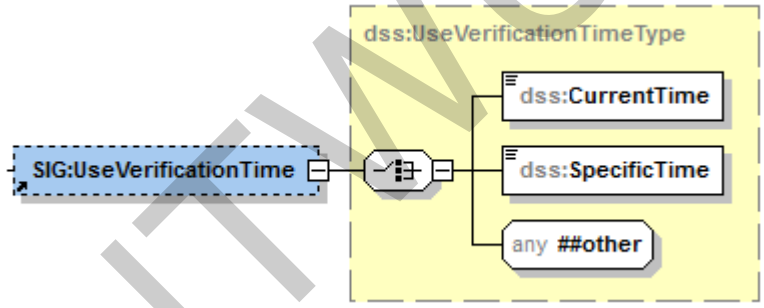
663 **[<=]**

664 ### 6.2.2.2 VerifyDocument

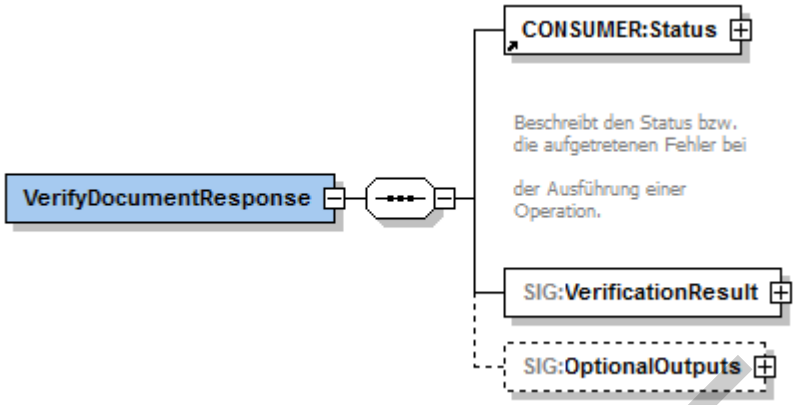
665 **A_17526-02~~A_17526-01~~ - Basis- und KTR-Consumer, Operation VerifyDocument**
666 Der Signaturdienst des Basis- und KTR-Consumer MUSS an der Clientschnittstelle eine
667 Operation `VerifyDocument` wie in Tabelle Tab_Operation_VerifyDocument beschrieben
668 anbieten.

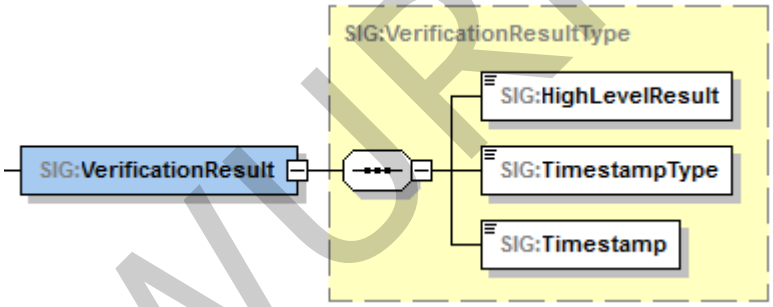
669 **Tabelle 15: Tab_Operation_VerifyDocument**

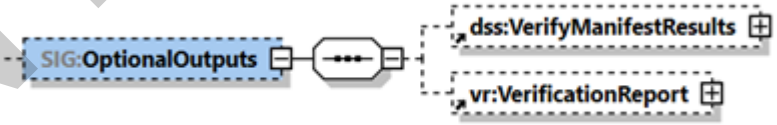
Name	**`VerifyDocument`**	
Beschreibung	Diese Operation verifiziert die Signatur eines Dokumentes. Der Basis- und KTR-Consumer MUSS jede konform zur Clientschnittstelle SignDocument erzeugte Signatur durch VerifyDocument prüfen können. Das Ergebnis der Prüfung wird, wenn gefordert, in Form eines standardisierten Prüfberichts in einer `VerificationReport`-Struktur gemäß [OASIS-VR] zurückgeliefert.	
Aufruf-parameter		
	`SIG: OptionalInputs`	Enthält optionale Eingabeparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7): Die zulässigen optionalen Eingabeparameter sind unten erläutert.
	`SIG: Document`	Enthält im Fall der Prüfung von detached oder enveloped Signaturen das zur Signatur gehörende bzw. das diese umschließende Dokument (siehe [OASIS-DSS] Section 2.4.2 und oben).
	`dss: SignatureObject`	Enthält die zu prüfende Signatur, wenn sie nicht im Dokument selbst eingebettet ist ([OASIS-DSS] Kapitel 4.1). Die Signatur wird in `ss:Base64Signature` mit entsprechend gesetztem `Type`-Attribut (siehe `SignatureType`) übergeben, wobei der nachfolgende Werte unterstützt werden muss:

	• **CMS-Signatur** **urn:ietf:rfc:5652**
	
SIG: VerifyManifests	~~Durch das in [OASIS-DSS] (Abschnitt 4.5.1) definierte Element kann die Prüfung eines ggf. vorhandenen Manifests angefordert werden.~~Dieses Element wird durch den Basis-KTR-Consumer nicht verwendet.
	
SIG: UseVerification Time	Durch das in [OASIS-DSS] (Abschnitt 4.5.2) spezifizierte Element kann die Prüfung der Signatur bezüglich eines durch den Aufrufer bestimmten Zeitpunktes (Benutzerdefinierter_Zeitpunkt) erfolgen.
dss: AdditionalKeyInfo	~~Durch das in [OASIS-DSS] (Abschnitt 4.5.4) spezifizierte Element kann zusätzliches, für die Prüfung benötigtes, Schlüsselmaterial übergeben werden.~~Dieses Element wird durch den Basis-KTR-Consumer nicht verwendet.
vr: Return VerificationReport	Durch dieses in [OASIS-VR] spezifizierte Element kann die Erstellung eines ausführlichen Prüfberichtes angefordert werden.
dss:Schemas	Dieses Element wird durch den Basis-KTR-Consumer nicht verwendet.

Rückgabe



`Status`	Enthält den Ausführungsstatus der Operation.
`SIG: VerificationResult`	 Das Element `Sig:VerificationResult` enthält das Ergebnis der Prüfung als Ampel, den Typ des zugehörigen angenommenen Signaturzeitpunkts und der angenommene Signaturzeitpunkt selbst.
`SIG: HighLevelResult`	Das Ergebnis der Prüfung (Ampelschaltung) mit folgenden Werten: • VALID: alle Signaturen sind gültig • INVALID: mindestens eine der Signaturen ist ungültig • INCONCLUSIVE: in allen anderen Fällen

	SIG: TimestampType	Der Typ des angenommenen Signaturzeitpunkts mit folgenden Werten: • SIGNATURE_EMBEDDED_TIMESTAMP: in der Signatur eingebetter Zeitpunkt Ermittelter_Signaturzeitpunkt_Eingebettet • ~~QUALIFIED_TIMESTAMP: qualifizierter Zeitstempel über die Signatur Ermittelter_Signaturzeitpunkt_Qualifiziert~~ • SYSTEM_TIMESTAMP: Systemzeit des ~~Konnektors~~Consumers bei Signaturprüfung Ermittelter_Signaturzeitpunkt_System • USER_DEFINED_TIMESTAMP: benutzerdefinierter Zeitpunkt Benutzerdefinierter_Zeitpunkt Als Format darf jedes zum XML-Typ "dateTime" konforme Format verwendet werden (<element name="Timestamp" type="dateTime"/>). Wenn mehrere Signaturen im Dokument vorhanden sind, wird hier der angenommene Signaturzeitpunkt der jüngsten Signatur angegeben.
	SIG: Timestamp	Im Element SIG:Timestamp wird der zu SIG:TimestampType gehörende Zeitstempel zurückgegeben.
	SIG: OptionalOutputs	Enthält (angelehnt an dss:OptionalOutputs, wie in Abschnitt 2.7 von [OASIS-DSS] beschrieben) optionale Ausgangselemente: 
	dss: VerifyManifestResults	Dieses in Abschnitt 4.5.1 von [OASIS-DSS] definierte Element enthält Informationen zur Prüfung eines ggf. vorhandenen Signaturmanifests und wird zurückgeliefert, sofern beim Aufruf das dss:VerifyManifest-Element, aber nicht das RequestVerificationReport als optionales Eingabeelement übergeben wurde.
	vr: VerificationReport	Dieses in [OASIS-VR] spezifizierte Element wird zurückgeliefert, falls das ReturnVerificationReport-Element als Eingabeparameter verwendet wurde.
Vorbedingungen	Keine	

Nachbedingungen	Keine

SigningTime ist der zu prüfende Signaturzeitpunkt. Dieser ergibt sich wie folgt:

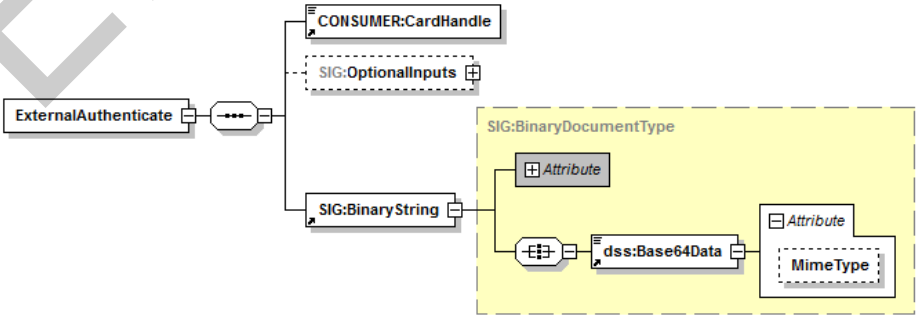
1. SigningTime = Benutzerdefinierter_Zeitpunkt, wenn SIG:UseVerificationTime Angaben enthält, sonst
2. SigningTime = Ermittelter_Signaturzeitpunkt_Eingebettet wenn die Signatur einen Signaturzeitpunkt enthält, sonst
3. SigningTime = Ermittelter_Signaturzeitpunkt_System, die Systemzeit.

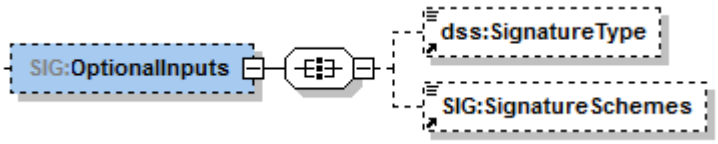
Das Verifizieren erfolgt durch Aufruf von PL_TUC_VERIFY_DOCUMENT_nonQES {
SIGNED_DOCUMENT = SIG:Document;
CERTIFICATE = extrahiert aus SIG:Document;
SIGNATURE = dss: SignatureObject ;
TIME_REFERENCE = ~~extrahierte~~ SigningTime ~~aus SIG:Document;~~;
}.
[<=]

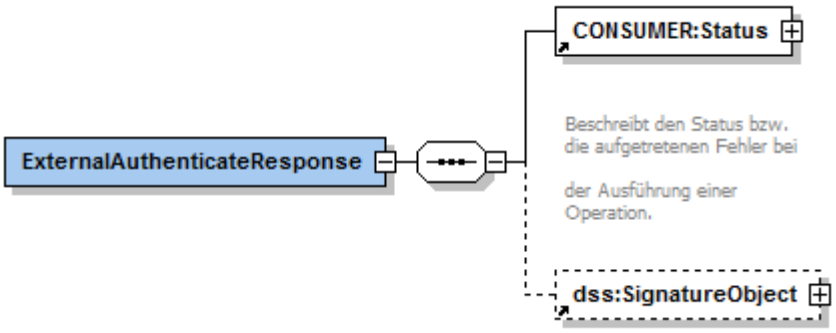
6.2.2.3 ExternalAuthenticate

A_17578 - Basis- und KTR-Consumer, Operation ExternalAuthenticate
Der Signaturdienst des Basis- und KTR-Consumer MUSS an der Clientschnittstelle die Operation ExternalAuthenticate wie in Tabelle Tab_Operation_ExternalAuthenticate beschrieben anbieten.

Tabelle 16: Tab_Operation_ExternalAuthenticate

Name	**ExternalAuthenticate**	
Beschreibung	Diese Operation versieht einen Binärstring der maximalen Länge 512 Bit mit einer nicht-qualifizierten elektronischen Signatur (nonQES). Dazu wird das Signaturverfahren PKCS#1 oder ECDSA verwendet.	
Aufruf parameter		
	Name	**Beschreibung**
	CONSUMER: CardHandle	Identifiziert die zu verwendende Signaturkarte.

	SIG: Optional Inputs	Enthält optionale Eingangsparameter: 
	SIG: Binary String	Dieses Element enthält im Kindelement `dss:Base64Data` den zu signierenden Binärstring. Das XML Attribut `SIG:BinaryString/dss:Base64Data/@MimeType` MUSS den Wert "application/octet-stream" haben. Die maximale Länge des Binärstrings beträgt 512 Bit entsprechend der maximal zu erwartenden Hash-Größe.
	dss: Signature Type	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element wird der Typ der zu erzeugenden Signatur bestimmt. Als Signaturtyp wird unterstützt : • **PKCS#1-Signatur** Durch Übergabe der URI urn:ietf:rfc:3447 wird eine PKCS#1 (Version 2.1) Signatur gemäß [RFC3447] erzeugt, die als `dss:Base64Signature` mit der oben genannten URI zurückgeliefert wird. • **ECDSA-Signatur** Durch Übergabe der URI urn:bsi:tr:03111:ecdsa wird eine ECDSA Signatur gemäß [BSI-TR-03111]#4.2.1 erzeugt, die als `dss:Base64Signature` mit der oben genannten URI zurückgeliefert wird. Andere `SignatureType`-Angaben führen zu einer Fehlermeldung. Fehlt dieses Element, so wird ebenfalls der Signaturtyp PKCS#1-Signatur verwendet.
	SIG: Signature Schemes	Durch dieses Element wird für PKCS#1-Signaturen zwischen den folgenden SignatureScheme-Optionen unterschieden: • RSASSA-PSS • RSASSA-PKCS1-v1_5 Fehlt dieses Element, so wird als Default-SignatureScheme RSASSA-PSS gewählt.

Rückgabe		
	`CONSUMER: Status`	Enthält den Status der ausgeführten Operation.

	`dss: SignatureObject`	Enthält im Erfolgsfall die erzeugte Signatur in Form eines `dss:SignatureObject`-Elements gemäß [OASIS-DSS] (Abschnitt 3.2).
		Der Signaturwert wird im XML-Element `dss:SignatureObject/dss:Base64Signature` übergeben. Das XML-Attribut `dss:SignatureObject/dss:Base64Signature/@Type` kennzeichnet durch den Wert:
		• urn:ietf:rfc:3447 den Signatur-Typ PKCS#1
		bzw.
		• urn:bsi:tr:03111:ecdsa den Signatur-Typ ECDSA.
		Die XML-Elemente
		`dss:SignatureObject/ds:Signature`
		`dss:SignatureObject/dss:Timestamp`
		`dss:SignatureObject/dss:SignaturePtr`
		`dss:SignatureObject/dss:Other`
		werden nicht verwendet.

Vorbedingungen	Keine	
Nachbedingungen	Keine	

Das Signieren erfolgt durch Aufruf von PL_TUC_SIGN_HASH_nonQES {
  IDENTIFIKATOR = `CardHandle`;
  SIGNATURVERFAHREN = `SIG:SignatureSchemes`;
  HASHWERT = `SIG:BinaryString`;
}
[<=]

6.3 Zertifikatsdienst

6.3.1 Durch Module nutzbare TUCs

A_17401 - Systemprozess PL_TUC_PKI_VERIFY_CERTIFICATE
Der Basis- und KTR-Consumer MUSS den Systemprozess PL_TUC_PKI_VERIFY_CERTIFICATE implementieren und bereitstellen.**[<=]**

6.3.2 Operationen an der Clientschnittstelle

A_17408 - Basisdienst Zertifikatsdienst
Der Basis- und KTR-Consumer MUSS Clientsystemen einen Basisdienst Zertifikatsdienst zur Verfügung stellen.

Tabelle 17: Tab_Zertifikatsdienst

Name	**CertificateService**	
Version	Siehe Anhang B	
Namensraum	Siehe Anhang B	
Namensraum-Kürzel	CERT für Schema und CERTW für WSDL	
Operationen	**Name**	**Kurzbeschreibung**
	VerifyCertificate	Prüfung des Status eines Zertifikats
WSDL	CertificateService.wsdl	
Schema	CertificateService.xsd	

[<=]

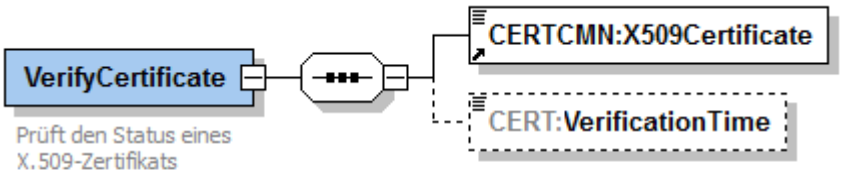
6.3.2.1 VerifyCertificate

A_17429-01 - Basis- und KTR-Consumer, Operation VerifyCertificate
Der Zertifikatsdienst des Basis- und KTR-Consumer MUSS an der Clientschnittstelle eine Operation `VerifyCertificate` wie in Tabelle Tab_Operation_VerifyCertificate beschrieben anbieten.

Tabelle 18: Tab_Operation_VerifyCertificate

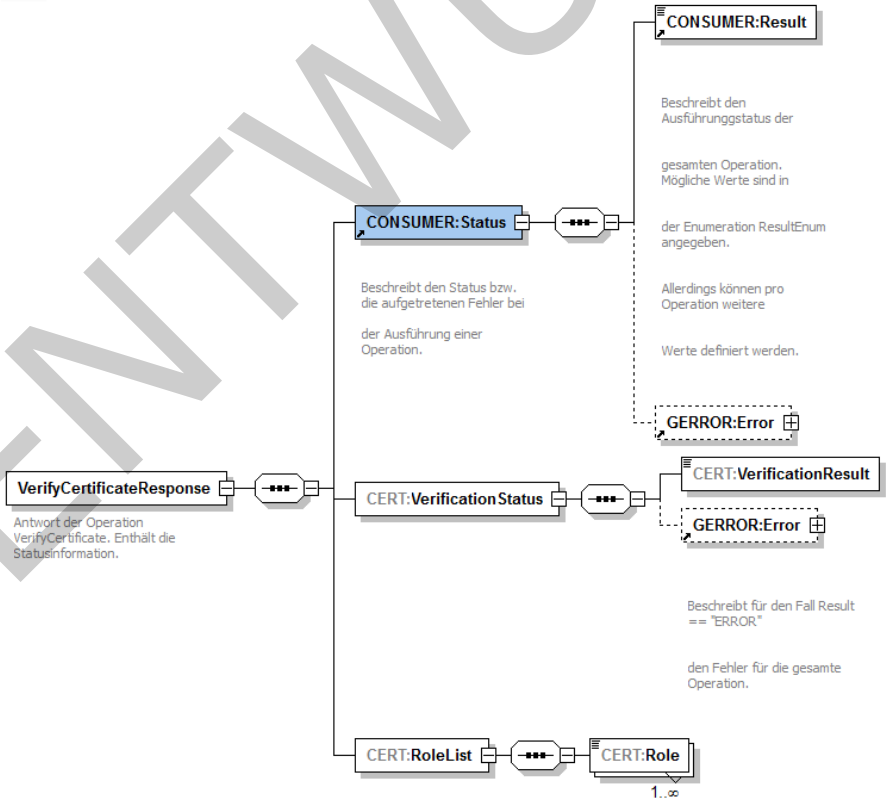
Name	**VerifyCertificate**
Beschreibung	Prüft den Status eines Zertifikats.

Aufruf-parameter



Name	Beschreibung
CERTCMN: X509Certificate	Enthält das base64-codierte Zertifikat, dessen Binärstruktur wiederum ASN.1-codiert (gemäß [gemSpec_PKI]) vorliegt.
CERT: VerificationTime	Der für die Prüfung zu verwendende Referenzzeitpunkt. Falls der Parameter nicht angegeben ist, wird als Referenzzeitpunkt die Systemzeit verwendet.

Rückgabe



Status	Enthält den Ausführungsstatus der Operation.

	CERT:VerificationStatus	Enthält eines der drei möglichen Prüfungsergebnisse in CERT:VerificationResult • VALID • INCONCLUSIVE • INVALID sowie weiter Details zu den Zuständen „INCONCLUSIVE“ und „INVALID“ in GERROR:Error.
	CERT:RoleList	OIDs der im Zertifikat gespeicherten Rollen.
Vorbe-dingungen	Keine	
Nachbe-dingungen	Keine	

Der Ablauf der Operation `VerifyCertificate` ist in Tabelle Tab_Ablauf_VerifyCertificate beschrieben:

Tabelle 19: Tab_Ablauf_VerifyCertificate

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung

1.	PL_TUC_PKI_ VERIFY_CERTIFICATE	Die Zertifikatsprüfung erfolgt durch Aufruf von PL_TUC_PKI_VERIFY_CERTIFICATE {
		Zu prüfendes Zertifikat = CERTCMN:X509Certificate;
		Referenzzeitpunkt = CERT:VerificationTime;
		PolicyList = keine Einschränkung;
		KeyUsage = empty;
		ExtendedKeyUsage = empty;
		OCSP-Graceperiod = empty;
		Offline-Modus = nein;
		OCSP-Response = empty ;
		Timeout = empty;
		TOLERATE_OCSP_FAILURE = ja;
		}

2.		Wenn der Prüfprozess fehlerhaft war und nicht zu einem Ergebnis im Sinne eines VerificationResult führt, wird eine FaultMessage erzeugt. War der Prüfprozess erfolgreich, wird eine VerifyCertificateResponse mit • CONSUMER:Status/CONSUMER:Result=OK, • dem VerificationStatus (als Ergebnis der Zertifikatsprüfung) und • den ermittelten Rollen-OIDs erzeugt. Ein Prüfergebnis „INCONCLUSIVE" bzw. „INVALID" wird in CERT:VerificationStatus/GERROR:Error mit den zugehörigen Fehlermeldungen detailliert (in diesem Fall kann CONSUMER:Status/CONSUMER:Result=OK oder CONSUMER:Status/CONSUMER:Result=Warning gesetzt sein).

Tabelle 20: Tab_Übersicht_VerificationResult_VerifyCertificate

CERT:VerificationResult	Bedeutung
VALID	Wenn Gültigkeit zu Referenzzeitpunkt: "gültig" Mathematische Gültigkeit:"gültig" OCSP-Prüfung: Online gültig
INVALID	Wenn mindestens ein Wert von (Gültigkeit zu Referenzzeitpunkt, Mathematische Gültigkeit, OCSP-Prüfung) „ungültig",„Prüffehler" oder „gesperrt" ist.
INCONCLUSIVE	Wenn OCSP-Prüfung „unbekannt" und die andere Werte „gültig" sind.

[<=]

6.4 LDAP-Proxy

6.4.1 Durch Module nutzbare TUCs

A_17343 - Basis- und KTR-Consumer, LDAPv3 Operationen für interne Module
Der Basis- und KTR-Consumer MUSS für die in Tab_Ldap_TUC_Mapping aufgelisteten Systemprozesse die entsprechenden LDAP-Operationen implementieren und zur Nutzung durch interne Module zur Verfügung stellen.

Tabelle 21: Tab_Ldap_TUC_Mapping

LDAPv3-Operation	Systemprozess

Bind	PL_TUC_VZD_BIND
Unbind	PL_TUC_VZD_UNBIND
Search	PL_TUC_VZD_SEARCH
Abandon	PL_TUC_VZD_ABANDON

[<=]

6.4.2 Unterstützte LDAPv3-Operationen an der Clientschnittstelle

A_17341 - Basis- und KTR-Consumer, LDAPv3-Operationen an der Clientschnittstelle
Der Basis- und KTR-Consumer MUSS an der Client-Schnittstelle die folgenden LDAPv3-Operationen für den Zugriff auf den Verzeichnisdienst der TI gemäß [RFC4511] anbieten.

- Bind Operation
- Unbind Operation
- Search Operation
- Abandon Operation

Andere LDAPv3-Operationen MÜSSEN mit dem LDAP-Fehler unwillingToPerform (53) beantwortet werden.
Fehler MÜSSEN gemäß [RFC4511]#Appendix A behandelt werden.**[<=]**

6.5 Clientmodul KOM-LE

6.5.1 Allgemeine Anforderungen

A_17298 - Synchronisation mit der Systemzeit der zentralen TI-Plattform
Das KOM-LE-Clientmodul MUSS sich unter Verwendung des Systemprozesses PL_TUC_NET_SYNC_TIME mit der Systemzeit des Zeitservers der zentralen TI-Plattform synchronisieren.**[<=]**

A_17299 - Konfigurationsparameter
Das KOM-LE-Clientmodul MUSS die in Tabelle Tab_Konf_Param aufgelisteten Parameter über eine Managementoberfläche oder eine Konfigurationsdatei konfigurierbar gestalten und mit einer Standardkonfiguration entsprechend den Defaultwerten ausliefern.

Tabelle 22: Tab_Konf_Param Standardkonfiguration

Parameter	Beschreibung des Parameters	Defaultwert
ADDRESS_SMTP	URI SMTP-Server	-
ADDRESS POP3	URI POP3-Server	-

PORT_SMTP	SMTP-Port für Clientsysteme	25
PORT_POP3	POP3-Port für Clientsysteme	995
SMTP_TIMEOUT_SERVER	Timeout für Antworten vom SMTP-Server auf SMTP-Kommandos	5 Minuten
SMTP_TIMEOUT_CLIENT	Timeout für das Warten auf neue SMTP-Kommandos vom Clientsystem	5 Minuten
POP3_TIMEOUT_SERVER	Timeout für Antworten vom POP3-Server auf POP3-Kommandos	5 Minuten
POP3_TIMEOUT_CLIENT	Timeout für das Warten auf neue POP3-Kommandos vom Clientsystem	5 Minuten
TTL_ENC_CERT	Time to Live für gecachte Verschlüsselungs-zertifikate	24 Stunden
TTL_EMAIL_ICCSN	Time to Live für gecachte Zuordnungen von E-Mail-Adressen der Sender bzw. Empfänger zu ICCSNs von deren HBAs/SM-Bs	30 Tage
TTL_PROTS	Time to Live für Protokolldateien.	30 Tage
PROT_PERF	Protokolldatei für Performance	JA

[<=]

A_17503 - Prüfung von TLS-Server-Zertifikaten
Das KOM-LE-Clientmodul MUSS für die Prüfung von TLS-Server-Zertifikaten der KOM-LE-Fachdienste den Systemprozess PL_TUC_PKI_VERIFY_CERTIFICATE des Basis- und KTR-Consumer benutzen.
[<=]

6.5.2 Senden von Nachrichten

A_17300 - Initialer SMTP-Dialog
Das KOM-LE-Clientmodul MUSS, nachdem die SMTP-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut wird und bis zum Punkt an dem das Clientsystem die Bestätigung des Erfolgs oder Misserfolgs seiner Authentifizierung erwartet, einen SMTP-Dialog entsprechend der Tabelle Tab_SMTP_Ant_Init mit dem Clientsystem führen.

Tabelle 23: Tab_SMTP_Ant_Init Antworten Clientmodul im CONNECT-Zustand

SMTP-Kommando (Clientsystem -> Clientmodul)	SMTP-Antwortcode (Clientmodul -> Clientsystem)
HELO	"250 OK" Antwortcode
EHLO	"250 OK" Antwortcode mit folgenden EHLO-Kennworten: SIZE <size> AUTH LOGIN PLAIN 8BITMIME ENCHANCEDSTATUSCODES DSN und <size> gleich oder großer als 35882577
AUTH	Anmeldungsdaten erhalten und Verbindungsaufbau mit dem MTA beginnen
RSET, NOOP	„250 OK" Antwortcode
MAIL, RCPT, DATA	„530 5.7.0" Antwortcode (Authentication required)
QUIT	„221 OK" Antwortcode senden und die Verbindung mit dem Clientsystem schließen
Andere Meldungen	„502 5.5.1" Antwortcode (Invalid command)

[<=]

A_17301 - Verbindungsaufbau mit dem SMTP-Servers
Das KOM-LE-Clientmodul MUSS für den Verbindungsaufbau mit dem SMTP-Server die Werte der Konfigurationsparameter ADDRESS_SMTP und PORT_SMTP verwenden.**[<=]**

A_17302 - Authentisierung gegenüber dem SMTP-Server mit Benutzernamen und Passwort
Das KOM-LE-Clientmodul MUSS den Benutzernamen und das Passwort, die es vom Clientsystem erhalten hat, für die Authentisierung gegenüber dem SMTP-Server verwenden.**[<=]**

A_17303 - Ergebnis des Verbindungsaufbaus mit dem SMTP-Server
Das KOM-LE-Clientmodul MUSS das Clientsystem über das Ergebnis des Verbindungsaufbaus mit dem MTA mit den in Tabelle Tab_SMTP_Verbindung beschriebenen SMTP-Antwortcodes informieren.

Tabelle 24: Tab_SMTP_Verbindung SMTP-Antwortcodes für MTA-Verbindungsaufbau

Bedingung	**SMTP-Antwortcode (Clientmodul -> Clientsystem)**
Das Clientmodul hat sich erfolgreich gegenüber dem MTA mit den vom Clientsystem erhaltenen Anmeldungsdaten authentifiziert.	235 2.7.0 (Authentication successful)

Das Clientsystem verwendet für die SMTP-Authentifizierung einen anderen Mechanismus als PLAIN oder LOGIN.	504 5.7.4 (Security features not supported)
Die Verbindung zwischen dem Clientmodul und dem MTA kann nicht aufgebaut werden.	454 4.7.0 (Temporary authentication failure)
Die Authentifizierung gegenüber dem MTA schlägt fehl.	535 5.7.8 (Authentication credentials invalid)

[<=]

A_17305 - Verwenden von PL_TUC_SIGN_DOCUMENT_nonQES und PL_TUC_HYBRID_ENCIPHER
Das KOM-LE-Clientmodul MUSS für das Signieren und Verschlüsseln der Nachrichten entsprechend dem KOM-LE-S/MIME-Profil die Systemprozesse PL_TUC_SIGN_DOCUMENT_nonQES und PL_TUC_HYBRID_ENCIPHER des Basis- und KTR-Consumers verwenden.**[<=]**

A_17306 - Vorgehen bei Signatur und Verschlüsselung einer KOM-LE Nachricht
Das KOM-LE-Clientmodul MUSS zur Signatur und Verschlüsselung von KOM-LE Nachrichten das folgende Vorgehen umsetzen:

1. Unter Verwendung des Systemprozesses PL_TUC_SIGN_DOCUMENT_nonQES des Basis- und KTR-Consumers erzeugt das Clientmodul KOM-LE einen binären Opak-signierten CMS-Container entsprechend dem KOM-LE-S/MIME-Profil.
2. Der binäre CMS-Container mit der signierten Nachricht wird als „application/pkcs7-mime" MIME-Einheit vom smime-type „signed-data" mit dem Content-Transfer-Encoding „binary" verpackt.
3. Zur CMS-Verschlüsselung übergibt das KOM-LE-Clientmodul beim Aufruf des Systemprozesses PL_TUC_HYBRID_ENCIPHER die in Schritt zwei erzeugte Nachricht als binär-Dokument. Als Antwort erhält das KOM-LE-Clientmodul einen binären CMS-Container zurück.
4. Der aus der Verschlüsselung resultierende CMS-Container wird in eine „application/pkcs7-mime" MIME-Einheit vom smime-type „authenticated-enveloped-data" mit dem Content-Transfer-Encoding „base64" verpackt.

[<=]

A_17327 - Signieren der Nachricht mit dem Schlüssel Prk.HCI.OSIG
Das KOM-LE-Clientmodul MUSS für das Signieren einer KOM-LE-Nachricht den privaten Schlüssel PrK.HCI.OSIG.R2048 der SM-B der jeweiligen Organisation (Kostenträger oder Leistungserbringerorganisation) verwenden.
[<=]

6.5.3 Empfangen von Nachrichten

A_17328 - POP3-Dialog zur Authentifizierung
Das KOM-LE-Clientmodul MUSS, nachdem die POP3-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut wurde und bis zu dem Punkt an dem das Clientsystem die Bestätigung des Erfolgs oder Misserfolgs seiner Authentifizierung

erwartet, einen POP3-Dialog entsprechend Tabelle Tab_POP3_Ant_Init mit dem Clientsystem führen.

Tabelle 25: Tab_POP3_Ant_Init Antworten Clientmodul im CONNECT-Zustand

Clientsystem -> Clientmodul	Clientmodul -> Clientsystem
CAPA	"+OK" Antwortcode mit folgenden CAPA Kennworten: TOP USER SASL PLAIN UIDL
USER, AUTH	Anmeldungsdaten erhalten und Verbindungsaufbau mit dem POP3-Server fortsetzen
QUIT	„+ OK" Antwortcode senden und die Verbindung mit dem Clientsystem schließen
Andere Meldungen	„-ERR" Antwortcode

[<=]

A_17329 - Verbindungsaufbau mit dem POP3-Servers
Das KOM-LE-Clientmodul MUSS für den Verbindungsaufbau mit dem POP3-Server die Werte der Konfigurationsparameter ADDRESS_POP3 und PORT_POP3 verwenden.**[<=]**

A_17330 - Authentifizierung gegenüber POP3-Server mit Benutzernamen und Passwort
Das KOM-LE-Clientmodul MUSS den Benutzernamen und das Passwort,die es vom Clientsystem erhalten hat, für die Authentifizierung gegenüber dem POP3-Server verwenden.**[<=]**

A_17331 - Ergebnis des Verbindungsaufbaus mit dem POP3-Server
Das KOM-LE-Clientmodul MUSS das Clientsystem über das Ergebnis des Verbindungsaufbaus mit dem POP3-Server mit den in der Tabelle Tab_POP3_Verbindung beschriebenen POP3-Antwortcodes informieren.

Tabelle 26: Tab_POP3_Verbindung Antwortcodes für POP3-Server-Verbindungsaufbau

Bedingung	**POP3 Antwortcode (Clientmodul -> Clientsystem)**
Das Clientsystem hat sich erfolgreich gegenüber dem POP3-Server mit den vom Clientsystem erhaltenen Anmeldungsdaten authentifiziert.	+OK
Das Clientsystem verwendet für die POP3-Authentifizierung einen anderen Mechanismus als USER/PASS oder PLAIN.	-ERR
Die Verbindung zwischen dem Clientmodul und dem POP3-Server kann nicht aufgebaut werden.	-ERR
Die Authentifizierung gegenüber dem MTA schlägt fehl.	-ERR

[<=]

A_17333 - E-Mail-Adresse des den Abholvorgang auslösenden Nutzers
Das KOM-LE-Clientmodul MUSS den vom Clientsystem erhaltenen POP3-Usernamen als die E-Mail-Adresse des den Abholvorgang auslösenden Nutzers betrachten.**[<=]**

A_17504 - Verwenden von PL_TUC_VERIFY_DOCUMENT_nonQES und PL_TUC_HYBRID_DECIPHER
Das KOM-LE-Clientmodul MUSS für das Entschlüsseln und die Signaturprüfung der Nachrichten die Systemprozesse PL_TUC_VERIFY_DOCUMENT_nonQES und PL_TUC_HYBRID_DECIPHER des Basis- und KTR-Consumers verwenden.
[<=]

A_17337 - Abbrechen des Entschlüsseln, wenn die erforderliche SM-B nicht verfügbar ist
Das KOM-LE-Clientmodul MUSS die Entschlüsselung einer Nachricht abbrechen, wenn die für die Entschlüsselung erforderliche SM-B nicht verfügbar ist.**[<=]**

A_17338 - Abbrechen des Entschlüsseln, wenn Freischaltung der erforderlichen SM-B fehlschlägt
Das KOM-LE-Clientmodul MUSS die Entschlüsselung einer Nachricht abbrechen, wenn die Freischaltung der für die Entschlüsselung erforderlichen SM-B fehlschlägt.**[<=]**

6.6 Realisierung der Leistungen der TI-Plattform

A_18130 - Nutzung von PL_TUC_CARD Systemprozessen
Der Basis-Consumer MUSS für den Zugriff auf Smartcards die in TAB_Systemprozesse mit PL_TUC_CARD_* bezeichneten Systemprozesse benutzen.
[<=]

6.6.1 Transportschnittstelle für Kartenkommandos

Wenn der Basis-Consumer Smartcards unterstützt, muss er eine Schnittstelle zu Karten der TI über ein Kartenterminal herstellen. Diese Schnittstelle muss die von den Plattformprozessen erzeugten, kartenverständlichen APDUs an die Karte übertragen. Neben proprietären Schnittstellentreibern von Kartenterminalherstellern existiert eine Reihe standardisierter Schnittstellen, die auch von verschiedenen Betriebssystemen zur Anbindung handelsüblicher Kartenterminals unterstützt werden.

Die folgenden Anforderungen betreffen die gemäß [gemSpec_Systemprozesse_dezTI#ENV_TUC_CARD_APDU_TRANSPORT] zu beschreibende Transportschnittstelle.

A_18166 - Vertrauliche und integritätsgeschützte Kommunikation mit KT
Wenn der Basis-Consumer Smartcards unterstützt, MUSS der Basis-Consumer mit dem Kartenterminal ausschließlich über eine vertrauliche, integritätsgeschützte Verbindung kommunizieren.**[<=]**

A_18097 - Transportschnittstelle für Kartenkommandos
Wenn der Basis-Consumer Smartcards unterstützt, MUSS er eine sichere Transportschnittstelle für die Übertragung von Smartcard-APDUs gemäß [CT-API] implementieren.**[<=]**

A_18100 - Ergänzende Standards für Transportschnittstelle
Der Basis-Consumer KANN eine Transportschnittstelle für die Übertragung von SmartCard-APDUs auf Basis des SICCT-Protokolls gemäß [CCID] und unter Verwendung der vom Hersteller des Kartenterminals ggf. bereitgestellten Hardwaretreiber implementieren.**[<=]**

A_18163 - Kartenterminal für Basis-Consumer
Wenn der Basis-Consumer Smartcards unterstützt, MUSS er mindestens ein Kartenterminal enthalten.
[<=]

A_18102 - PIN-Eingabe nicht speichern
Der Basis-Consumer DARF ein eingegebenes PIN-Geheimnis NICHT speichern.**[<=]**

A_18103 - PIN-Geheimnis ausschließlich an Karte übermitteln
Der Basis-Consumer MUSS sicherstellen, dass das eingegebene PIN-Geheimnis ausschließlich an die Karte und nicht an andere Adressaten übermittelt wird.
[<=]

6.6.2 Schnittstelle für PIN-Operationen und Anbindung der Karten der TI

Anwendungsfälle zur PIN-Verwaltung, zur Kartenfreischaltung oder weiterer Fachanwendungen können die Eingabe eines PIN- oder PUK-Geheimnisses erfordern. Der Zugriff auf Karten der TI erfolgt über die Systemprozesse PL_TUC_CARD_*. Der Basis-Consumer als Realisierungsumgebung der Systemprozesse muss seinerseits die von der Plattform geforderten Schnittstellen gemäß [gemSpec_Systemprozesse_dezTI#ENV_TUC_CARD_SECRET_INPUT] implementieren, um die Kommunikation der Plattform mit dem Benutzer zu ermöglichen.

Die Kommunikationsschnittstelle ist in Kapitel 6.6.1 Transportschnittstelle für Kartenkommandos beschrieben und umfasst das Kartenterminal, Eingabemedium und Hinweistexte an den Benutzer. Diese kann je nach Konfiguration an einem Gerät als Kartenterminal oder auch eine Kombination aus Bildschirmausgabe, Kartenterminal-PIN-Pad und/oder Tastatureingabe erfolgen.

A_18107 - Übergabeschnittstelle PIN/PUK-Geheimnis
Wenn der Basis-Consumer Smartcards unterstützt, MUSS er eine Operation gemäß [gemSpec_Systemprozesse_dezTI#ENV_TUC_CARD_SECRET_INPUT] zur Eingabe eines PIN/PUK-Geheimnisses und Weiterleitung an eine Smartcard mit folgenden Parametern implementieren:
Eingabeparameter:

- Identifikator
- Aktion
- minLength
- maxLength
- commandApduPart

Rückgabewerte

- responseApdu

[<=]

A_18108 - Umsetzung ENV_TUC_CARD_SECRET_INPUT
Wenn der Basis-Consumer Smartcards unterstützt, MUSS er die Abbildung der Eingabeparameter auf die Rückgabewerte der Operation ENV_TUC_SECRET_INPUT derart umsetzen, dass

- die Eingabeparameter `Identifikator` und `Aktion` für einen Hinweistext an den Benutzer verwendet werden, welche Aktion auf welchem konkreten Kartenobjekt (z.B. Name einer PIN) durchgeführt wird,
- der `commandApduPart` an der Eingabeschnittstelle um das Benutzergeheimnis ergänzt wird,
- der `commandApduPart` über die Transportschnittstelle für Kartenkommandos an die Karte gesendet wird

und die Antwortnachricht der Karte als responseApdu an den Aufrufer zur Auswertung zurückgegeben wird.
[<=]

A_18109 - Minimalprinzip Karteninteraktion
Der Basis-Consumer DARF ein Kartenkommando NICHT an eine angebundene Karte weiterleiten, wenn dies nicht explizit im Kontext eines Anwendungsfalls (intendierte Kartenoperationen und Erhöhen des Sicherheitszustands der Karte, falls erforderlich) erforderlich ist.**[<=]**

953 # 7 Anhang A - Verzeichnisse

954 ## 7.1 Abkürzungen

955 Abkürzungen

Kürzel	Erläuterung
aAdG	Andere Anwendungen des Gesundheitswesens
aAdG NetG-TI	Andere Anwendungen des Gesundheitswesens mit Zugriff auf Dienste der TI aus angeschlossenen Netzen des Gesundheitswesens
AZPD	Anbieter Zentrale Plattform Dienste
CMS	Cryptographic Message Syntax
HSM	Hardware Security Module
IPv4	Internet Protokoll Version 4
IPv6	Internet Protokoll Version 6
KOM-LE	Kommunikation für Leistungserbringer
LDAP	Leightweight Directory Access Protocol
MIME	Multipurpose Internet Mail Extensions
MTA	Mail Transfer Agent
POP3	Post Office Protocol Version 3
S/MIME	Secure/Multipurpose Internet Mail Extensions
SM-B	Security Module Typ B
SMTP	Simple Mail Transfer Protocol
TI	Telematikinfrastruktur

7.2 Glossar

Begriff	Erläuterung
Funktionsmerkmal	Der Begriff beschreibt eine Funktion oder auch einzelne, eine logische Einheit bildende Teilfunktionen der TI im Rahmen der funktionalen Zerlegung des Systems.

Das Glossar wird als eigenständiges Dokument, vgl. [gemGlossar] zur Verfügung gestellt.

7.3 Abbildungsverzeichnis

7.4 Tabellenverzeichnis

7.5 Referenzierte Dokumente

7.5.1 Dokumente der gematik

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 sind in der aktuellsten, von der gematik veröffentlichten Dokumentenlandkarte enthalten, in der die vorliegende Version aufgeführt wird.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Einführung der Gesundheitskarte - Glossar
[gemSMIME_KOMLE]	gematik: S/MIME-Profil Kommunikation Leistungserbringer(KOM-LE)
[gemSpec_CM_KOMLE]	gematic: Spezifikation KOM-LE-Clientmodul
[gemSpec_Systemprozesse_dezTI]	gematik: Spezifikation der Systemprozesse der dezentralen TI
[gemSpec_VZD]	gematik: Spezifikation Verzeichnisdienst
[gemKPT_Arch_TIP]	gematik: Konzept Architektur der TI-Plattform
[gemSpec_FM_ePA_KTR_Consumer]	gematik: Spezifikation Fachmodul ePA im KTR-Consumer
[gemSpec_PKI]	gematik: Übergreifende Spezifikation PKI

7.5.2 Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[RFC1939]	RFC 1939: Post Office Protocol – Version 3, J. Myers, M. Rose, Mai 1996
[RFC2045]	RFC 2045: Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet Message Bodies, N. Freed, N. Borenstein, November 1996

[RFC2119]	RFC 2119 (März 1997): Key words for use in RFCs to Indicate Requirement Levels S. Bradner
~~[RFC3275]~~	~~D. Eastlage, J. Reagle, D. Solo: *(Extensible Markup Language) XMLSignature Syntax and Processing*, IETF RFC 3275, via http://www.ietf.org/rfc/rfc3275.txt~~
[RFC4511]	RFC 4511: Lightweight Directory Access Protocol (LDAP), J. Sermersheim, Juni 2006
[RFC4954]	RFC 4954: SMTP Service Extension for Authentication, R. Siemborski, A. Melnikov, März 2007
[RFC5083]	RFC 5083: Authenticated-Enveloped-Data Content Type, R.Housley, November 2007
[RFC5321]	RFC 5321: Simple Mail Transfer Protocol, J. Klensin, Oktober 2008
[RFC5652]	RFC 5652: Cryptographic Message Syntax (CMS), R. Housley, September 2009
[RFC5751]	RFC 5751: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification, B. Ramsdell, S. Turner, Januar 2010
[OASIS-DSS]	OASIS: Digital Signature Service Core Protocols, Elements, and Bindings, Version 1.0, OASIS Standard, via http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf
[OASIS-SP]	OASIS: Signature Policy Profile of the OASIS Digital Signature Services Version 1.0, Committee Draft 01, 18 May 2009, http://docs.oasis-open.org/dss-x/profiles/sigpolicy/oasis-dssx-1.0-profiles-sigpolicy-cd01.pdf
[OASIS-VR]	OASIS: Profile for comprehensive multi-signature verification reports for OASIS Digital Signature Services Version 1.0, Committee Specification 01, 12 November 2010, http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cs01.pdf
~~[XAdES]~~	~~European Telecommunications Standards Institute (ETSI): Technical Specication XML Advanced Electronic Signatures (XAdES). ETSI Technical Specication TS 101 903, Version 1.4.2, 2010~~
~~[XMLDSig]~~	~~W3C Recommendation (06.2008): XML-Signature Syntax and Processing http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/~~
[XMLEnc]	XML Encryption Syntax and Processing W3C Recommendation 11 April 2013 http://www.w3.org/TR/xmlenc-core1/
[XPATH]	W3C Recommendation (14 December 2010) XML Path Language (XPath) 2.0 (Second Edition) http://www.w3.org/TR/2010/REC-xpath20-20101214/

[CMS]	Cryptographic Message Syntax (CMS), September 2009 http://tools.ietf.org/html/rfc5652
[Canon XML1.1]	Canonical XML Version 1.1 http://www.w3.org/TR/2008/REC-xml-c14n11-20080502/
[CAdES]	ETSI: Electronic Signature Formats, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V2.2.1, 2008-07, via http://www.etsi.org
[CT-API]	https://www.tuvit.de/de/aktuelles/beitraege-white-paper/card-terminal-application-programing-interface-fuer-chipkartenanwendungen//
[CCID]	https://usb.org.10-1-108-210.causewaynow.com/sites/default/files/DWG_Smart-Card_CCID_Rev110.pdf

8 Anhang B – Übersicht über die verwendeten Versionen

Für den Fall, dass Schnittstellenversionen unterstützt werden müssen, die den gleichen TargetNamespace nutzen, kann der Basis- und KTR-Consumer zu diesen Schnittstellenversionen einheitlich einen SOAP-Endpunkt anbieten, der die höchste der Schnittstellenversionen implementiert.

Tabelle 27: Tab_Schema_Versionen Versionen der Schemas aus dem Namensraum des Basis- und KTR-Consumers

Schemas aus dem Namensraum des Basis- und KTR-Consumer „http://ws.gematik.de/consumer“		
Name	**Version**	**TargetNamespace**
CertificateService.wsdl	2.0.0	http://ws.gematik.de/consumer/CertificateService/WSDL/v2.0
CertificateService.xsd	2.0.0	http://ws.gematik.de/consumer/CertificateService/v2.0
CertificateServiceCommon.xsd	1.0.0	http://ws.gematik.de/consumer/CertificateServiceCommon/v1.0
ConsumerCommon.xsd	2.0.0	http://ws.gematik.de/consumer/ConsumerCommon/v2.0
EncryptionService.wsdl	2.0.0	http://ws.gematik.de/consumer/EncryptionService/WSDL/v2.0
EncryptionService.xsd	2.0.0	http://ws.gematik.de/consumer/EncryptionServiceCommon/v2.0
SignatureService.wsdl	2.0.0	http://ws.gematik.de/consumer/SignatureService/WSDL/v2.0
SignatureService.xsd	2.0.0	http://ws.gematik.de/consumer/SignatureServiceCommon/v2.0

9 Anhang C – Übersicht der genutzten Systemprozesse

Der Basis- und KTR-Consumer verwendet u.a. die in Tabelle TAB_Systemprozesse dargestellten Plattformleistungen aus [gemSpec_Systemprozesse_dezTI].

Tabelle 28: TAB_Systemprozesse – Verwendete Plattformleistungen

Kürzel	Bezeichnung
PL_TUC_HYBRID_DECIPHER	Hybrid entschlüsseln
PL_TUC_HYBRID_ENCIPHER	Hybrid verschlüsseln
PL_TUC_SIGN_DOCUMENT_nonQES	Dokument nonQES signieren
PL_TUC_SIGN_HASH_nonQES	mit Karten-Identität signieren
PL_TUC_VERIFY_DOCUMENT_nonQES	nonQES Dokumentensignatur verifizieren
PL_TUC_PKI_VERIFY_CERTIFICATE	Prüfung eines Zertifikats der TI
PL_TUC_VZD_BIND	Verbindung aufbauen
PL_TUC_VZD_UNBIND	Verbindung trennen
PL_TUC_VZD_SEARCH	Verzeichnis abfragen
PL_TUC_VZD_ABANDON	Verzeichnisabfrage abbrechen
PL_TUC_NET_SYNC_TIME	Zeit synchronisieren
PL_TUC_CARD_INFORMATION	gesammelte Statusinformationen zu einer Karte
PL_TUC_CARD_RESET	Rücksetzen einer Karte
PL_TUC_CARD_CHANGE_PIN	PIN ändern
PL_TUC_CARD_ENABLE_PIN	PIN-Schutz einschalten
PL_TUC_CARD_DISABLE_PIN	PIN-Schutz abschalten
PL_TUC_CARD_VERIFY_PIN	Benutzer verifizieren
PL_TUC_CARD_ACTIVATE_APPLICATION	Anwendung aktivieren

PL_TUC_CARD_DEACTIVATE_APPLICATION	Anwendung deaktivieren
PL_TUC_CARD_GET_CHALLENGE	Auslesen einer Zufallszahl

1048