Einführung der Gesundheitskarte

Schnittstellen- und Prozessspezifikation Komponenten-PKI

Version:	2.0.0
Stand:	30.08.2017
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemSpec_SST_Komponenten-PKI]

Dokumentinformationen

Dokumentenhistorie

Vers.	Stand	Kap./ Seite	Grund der Änderung, Hinweise	Bearbeitung
0.0.1	01.12.13		Initiale Erstellung	ARV
1.0.0	06.02.14		Freigegeben durch Release Board	ARV
1.0.1	18.03.14		Einarbeitung Kommentare der gem.	ARV
1.1.0	18.03.14		Freigabe durch Release-Mngt.	gematik
1.1.1	31.03.14		Einarbeitung Kommentare der gematik	ARV
1.2.0	31.03.14		Freigabe durch Release-Mngt.	gematik
1.3.0	11.04.14		freigegeben	gematik
1.3.1	02.06.14	5.2.5	Anpassung der URI für SOAP-Schnittstelle	ARV
1.3.2	16.06.14	5.3.5, 6.3.3, Anh. D	Extrakt der XSD und WSDL Definitionen als separate Dateien	ARV
1.3.3	16.06.14		Freigabe durch Release-Mngt.	gematik
1.4.0	24.06.14		freigegeben	gematik
1.4.1	27.11.14	5, 6, WSDL	Konkretisierung der SOAP-Schnittstelle	ARV
1.4.2	20.11.14	5,6	Review	ARV
1.5.0	21.11.14		Freigabe durch Release-Mngt.	gematik
1.5.1	08.01.15	5,6, WSDL	Konkretisierung SOAP- und CMP-SST	ARV
1.6.0	09.01.15		Freigabe durch Release-Mngt.	gematik
1.6.1	30.01.15	Kap. 2, 5, 6, WSDL, XSD	Einarbeitung Kommentare der gematik	ARV
1.7.0	30.01.15		Freigabe durch Release-Mngt.	gematik
1.7.1	10.02.15		Ergänzung gematik-Release und LG	gematik
1.7.2	12.02.15	List. 4, 6 Kap. 2	Einarbeitung Kommentare der gematik	ARV
1.8.0	12.01.15		Freigabe durch Release-Mngt.	gematik
1.8.1	16.03.15	Kap. 2 List. 3, 4, 5, 19, 20	Einarbeitung Kommentare der gematik	ARV
1.9.0	19.03.15		Freigabe durch Release-Mngt.	gematik
1.9.1	21.07.15 25.09.15	Kap. 2, 5.3, XSD	Anpassungen CR030 Störungsampel Freigabe	ARV
1.9.2	09.10.15	Anhang D,	Einpflegen eines Kommentartags mit der	ARV

Vers.	Stand	Kap./ Seite	Grund der Änderung, Hinweise	Bearbeitung
		XSD, WSDL	verknüpften Version der Schnittstellen-Spezifikation	
1.9.3	15.10.15	XSD, WSDL	Korrektur der Platzierung des Kommentartags	ARV
1.9.4	14.12.15 05.01.16	Kap. 2, 5.3.1, 5.3.3, 5.3.4	Anpassungen zu CR035 und CR072 Freigabe	ARV
1.9.5	19.01.16 26.01.16 27.01.16		Überarbeitungen / Korrekturen Überarbeitungen / Korrekturen Freigabe	ARV
1.9.6	19.02.2016 31.03.2016	Kap. 1.4, Tab. 5	Korrektur Zugriffsprofil gSMC-K Anpassungen/Korrekturen lt. Mängelprotokoll	ARV
1.9.7	10.03.2017	Kap. 2, 5.2.5	Ergänzung Registrierungsserver-Profil, Einschränkung Anzahl Zertifikatsbeantragung pro CMP/SOAP-Request pro Aufruf.	ARV
1.9.8	26.07.2017		Anpassungen CR095	ARV
1.9.9	15.08.2017		Korrektur gemäß Mängelliste	ARV
2.0.0	30.08.2017	Anhang D, Kap. 2	Korrektur gemäß Mängelliste & QS	ARV

Inhaltsverzeichnis

1 Einordnung des Dokumentes

1.1 Zielsetzung

Die vorliegende Spezifikation definiert die notwendigen Schnittstellen- und Prozesse, damit Anbieter, Hersteller, TSP-X.509 nonQES und Betreiber des TSL-Dienstes unter Verwendung der Komponenten-PKI Zertifikate beantragen und sperren können.

Dazu sind folgende Schnittstellen- und Prozesse definiert:

- Verwaltung und Übermittlung der Berechtigungsinformationen zugelassener Hersteller, Anbieter und TSPs durch die gematik über das bereitgestellte Zulassungsmanagement.
- Übermittlung der Zugangscredentials.
- Beantragung und Ausstellung von Komponenten-Zertifikaten über die bereitgestellten Schnittstellen.
- Sperrung von Komponenten-Zertifikaten über die bereitgestellten Schnittstellen.

1.2 Zielgruppe

Das Dokument ist maßgeblich für die Anbieter der Lose 1, 2 und 3 des Vorhabens „Erprobung Online-Rollout (Stufe 1)“ sowie für Hersteller und Anbieter von weiteren Produkten zum Online-Rollout (Stufe 1).

1.3 Geltungsbereich

Dieses Dokument spezifiziert die Schnittstellen und Prozesse der Komponenten-PKI.

1.4 Abgrenzungen

Antragsberechtigte für die Komponenten-PKI sind Akteure gemäß [gemSpec_TSP_X.509#Tab_PKI_511].

Keine Antragsberechtigten der Komponenten-PKI sind TSP-X.509 QES und Kartenherausgeber für folgende Zertifikatstypen:

- C.HP.AUT / C.HP.ENC (TSP-X.509 QES)
- C.HCI.AUT, C.HCI.ENC, C.HCI.OSIG (Kartenherausgeber)

Diese Zertifikate werden durch das G2-Vorhaben Los 3 /4 herausgegeben.

Die Ausgabepolicy der Komponenten-PKI kann [ARV_706.3_RL_Komponenten-PKI_CP] entnommen werden.

2 Systemüberblick

Die folgende Abbildung stellt die Gesamtübersicht der PKI der Telematikinfrastruktur mit Einordnung der Komponenten-PKI sowie Zertifikatsprofilen dar:

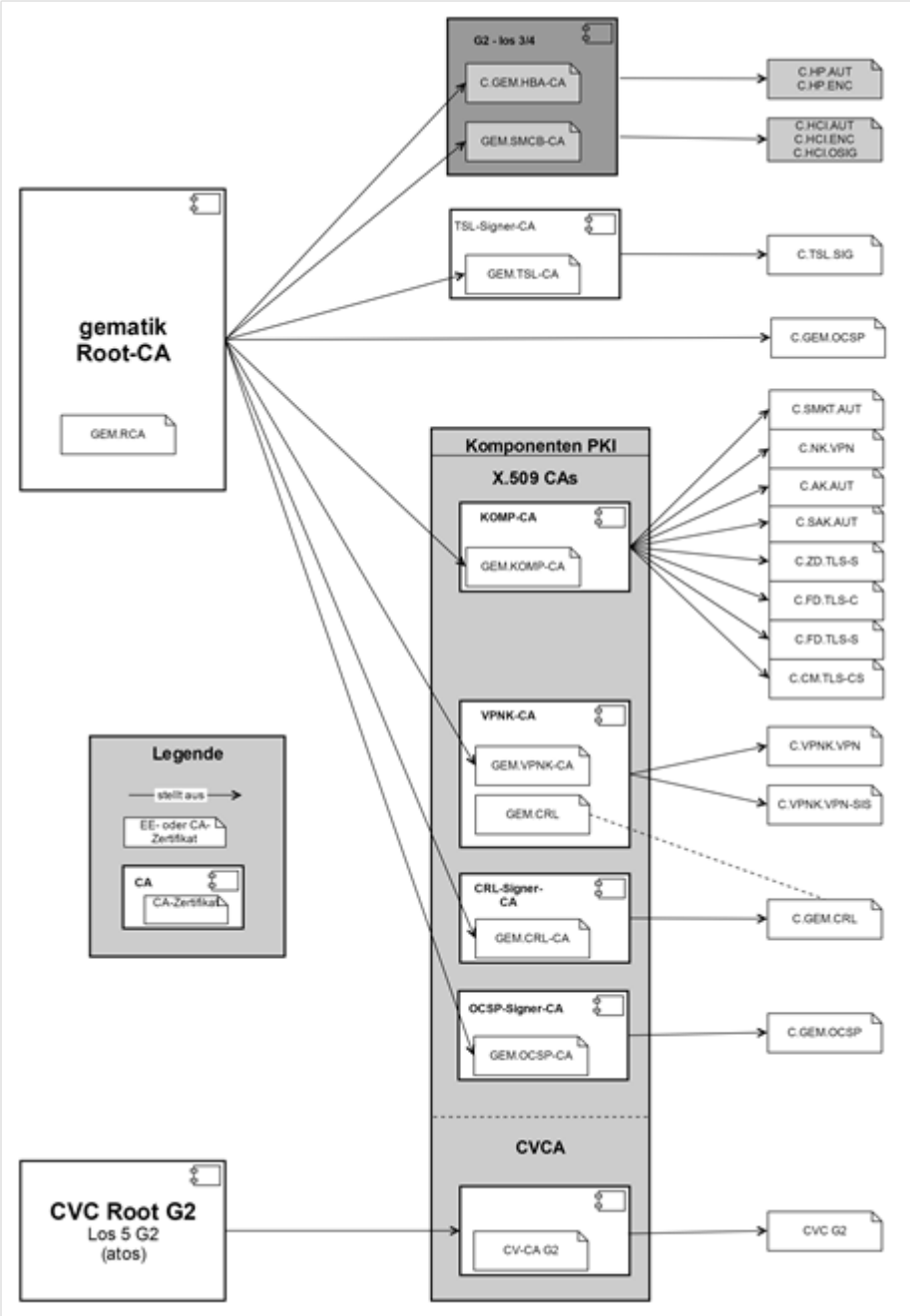

Abbildung 1: Gesamtübersicht der PKI ~~des ORS 1 Vorhabens~~ mit Einordnung der Komponenten PKI

Die CA-Einsatzbereiche der gematik Root-CA entsprechen denen aus Kapitel "3.4 Spezifische Aussteller-CA in der TI", gemSpec_PKI#Tab_PKI_213".

Ausnahme ist die Komponenten-PKI (Los 3) für nonQES-Zertifikate. Entsprechend stellt für diesen Anbieter die gematik Root-CA ein CA-Zertifikat gemäß gemSpec_PKI#GS-A_4702 aus (C.GEM.KOMP-CA), welches folgende Komponenten-Zertifikatsprofile ausstellen darf.

Tabelle 1: Zertifikatsprofile der C.GEM.KOMP-CA

Produkttyp	Zertifikatsprofil
Konnektor	• C.AK.AUT • C.SAK.AUT • C.NK.VPN
Kartenterminal	• C.SMKT.AUT
VSDM	• C.FD.TLS-S
VSDM Intermediär	• C.FD.TLS-S • C.FD.TLS-C
Konfigurationsdienst	• C.ZD.TLS-S
Störungsampel	• C.ZD.TLS-S
Verzeichnisdienst	• C.ZD.TLS-S
FD-KOM-LE	• C.FD.TLS-S • C.FD.TLS-C
CM-KOM-LE	• C.CM.TLS-CS
Registrierungsserver (als Teil vom VPN-Zugangsdienst)	• C.ZD.TLS-S
TSL-Dienst-TI	• C.ZD.TLS-S

Der Zertifikatsstatus der Zertifikate für VPN-Zugangsdienste muss im Internet über CRL prüfbar sein. Auf Grund dieser Anforderung wird für dieses Zertifikatsprofil eine spezifische CA bereitgestellt. Die VPNK-CA (C.GEM.VPNK-CA) der Komponenten-PKI stellt folgende Zertifikatsprofile aus:

Tabelle 2: Zertifikatsprofil C.GEM.VPNK-CA

Produkttyp	Zertifikatsprofil
VPN-Zugangsdienst	• C.VPNK.VPN • C.VPNK.VPN-SIS

Die OCSP-Signer-CA (C.GEM.OCSP-CA) der Komponenten-PKI stellt folgendes Zertifikatsprofil aus:

Tabelle 3: Zertifikatsprofil C.GEM.OCSP-CA

Produkttyp	Zertifikatsprofil
TSP-X.509 nonQES (OCSP-Signer)	• C.GEM.OCSP

Die CRL-Signer-CA (C.GEM.CRL-CA) der Komponenten-PKI stellt folgendes Zertifikatsprofil aus:

Tabelle 4: Zertifikatsprofil C.GEM.CRL-CA

Produkttyp	Zertifikatsprofil
TSP-X.509 nonQES (CRL-Signer)	• C.GEM.CRL

Die Komponenten-PKI stellt über diese Schnittstellen ebenfalls CV-Zertifikate aus und betreibt zu diesem Zweck eine CVCA (G2). Die CVCA stellt folgende Zugriffsprofile zur Verfügung:

Tabelle 5: Zugriffsprofile CVCA (G2)

Produkttyp	Zugriffsprofil
gSMC-K	• 51 • 0
gSMC-KT	• 54

Die Komponenten-PKI gewährleistet eine sichere Registrierung, Erstellung und Sperrung von X.509-Zertifikaten spezifisch für Komponenten und Dienste in der TI. Die Komponenten nutzen dabei Smartcards, deren CV-Zertifikate ebenfalls durch die Komponenten-PKI bereitzustellen sind. Diese PKI und die dazugehörigen Dienste werden durch Teilnehmer der TI und durch zugelassene sowie autorisierte Hersteller, Anbieter und TSPs genutzt.

Die Ausführungen in den nachfolgenden Kapiteln beschreiben die Schnittstellen und Prozesse zur Verwaltung und Übermittlung der Berechtigungsinformation und Zertifikatsausstellung (X.509/CV) sowie Zertifikatssperrung (X.509).

Es wird eine zentrale Komponente "Trust Management System (kurz „TMS“) bereitgestellt, welche von den Akteuren als Schnittstelle für die Registrierung und Verwaltung Berechtigter Antragsteller (I_Cert_Provisioning_Registration), die Zertifikatsausstellung (I_Cert_Provisioning_Erstellung / P_CVC_Provisioning) sowie die Zertifikatssperrung (I_Cert_Revocation) verwendet wird.

Die Registrierung und Verwaltung von Antragsberechtigten und deren Berechtigte Zertifikatsantragsteller wird durch die gematik über die vom TMS bereitgestellte Anwendung "Zulassungsmanagement" durchgeführt.

Das TMS stellt zudem eine Web-Anwendung "Zertifikatsausstellung" bereit, mit der berechtigte Zertifikatsantragsteller X.509/CV-Komponenten- sowie OCSP-/CRL-Signer-Zertifikate beantragen können. Weiterhin wird die Web-Anwendung "Zertifikatssperrung", mit der berechtigte Zertifikatsantragsteller X.509-Komponenten- und OCSP/CRL-Signer-Zertifikate sperren können, bereitgestellt.

Für die Zertifikatsbeantragung und -sperrung werden den Antrags- bzw. Sperrberechtigten eine Web-Anwendung und zusätzlich CMP- und SOAP-Schnittstellen zur Verfügung gestellt. Der Webserver des TMS nutzt das Protokoll „HTTP/1.1“ und unterstützt für die serverseitige Authentisierung TLS in den Versionen 1.1 und 1.2 mit der TLS-Cipher-Suite „TLS_DHE_RSA_WITH_AES_128_CBC_SHA“ gemäß [gemSpec_Krypt#Tab_KRYPT_015].

Auf Basis der von der gematik übermittelten Berechtigungsinformationen werden den berechtigten Akteuren die notwendigen Authentisierungsmechanismen zur Verfügung gestellt, mittels dieser sie sich gegenüber den beschriebenen Anwendungen autorisiert.

Auf Seiten der gematik wird zur Signaturerstellung ein lokaler QES-Signatur-Client (im Folgenden als QSC bezeichnet) installiert.

Als Mechanismus zur Authentisierung an Webanwendungen und am QSC werden RSA-Token eingesetzt. Die Authentisierung an den SOAP/CMP-Schnittstellen erfordert eine HTTP-Basic-Authentisierung und zusätzlich werden Berechtigungs-X.509-Zertifikate (CryptIDs) für die Kommunikation verwendet (siehe Kapitel 5.2.3 CryptID ausstellen).

Nachfolgende Tabelle gibt einen Überblick über die zu realisierenden Schnittstellen und Prozesse der Komponenten-PKI:

Tabelle 6: Überblick über die zu realisierenden Schnittstellen der Komponenten-PKI

Schnittstellen	Kurzbeschreibung
I_Cert_Provisioning	Die technische Schnittstelle zur Veranlassung der Erzeugung eines X.509-Komponenten-, OCSP- oder CRL-Signer-Zertifikats durch den berechtigten Akteur mit anschließender Bereitstellung des Zertifikats durch die Komponenten-PKI.
I_Cert_Revocation	Die technische Schnittstelle zur Veranlassung der Sperrung eines X.509-Komponenten-, OCSP- oder CRL-Signer-Zertifikats durch den berechtigten Akteur mit anschließender Bereitstellung des Zertifikats durch die Komponenten-PKI.
I_OCSP_Status_Information	Die technische Schnittstelle zur Bereitstellung der Zertifikatsstatusinformation von Komponenten-, OCSP- und CRL-Signer-Zertifikaten.
I_CRL_Download	Die technische Schnittstelle zur Bereitstellung der CRL für VPN-Zugangsdienst-Zertifikate.
P_CVC_Provisioning	Die Schnittstelle zur Veranlassung der

	Erzeugung eines CV-Zertifikates durch den berechtigten Akteur mit anschließender Bereitstellung des Zertifikats durch die Komponenten-PKI.

Die gematik muss Hersteller, Anbieter und TSP-X.509 nonQES zulassen und diesen die Berechtigung erteilen für deren zugelassene Produkttypen X.509-/CV-Komponenten- und OCSP/CRL-Signer-Zertifikate bei der Komponenten-PKI zu beantragen. Die gematik übermittelt dem Anbieter der Komponenten-PKI alle notwendigen Berechtigungsinformationen der Hersteller und Anbieter von zugelassenen Produkten, die berechtigt sind Zertifikate bei der Komponenten-PKI zu beantragen oder zu sperren.

Nach erfolgter Zulassung, Erfassung und Übermittlung der Berechtigten Zertifikatsantragsteller bzw. Sperrantragsteller wird den Akteuren ein RSA-Token zur Authentisierung bereitgestellt, so dass diese unter Verwendung der Komponenten-PKI Zertifikate beantragen und sperren können.

Die erforderlichen Schnittstellen und Prozesse sind zu diesem Zweck in folgende Use Cases untergliedert und spezifiziert:

Tabelle 7: Übersicht aller Use Cases

Schnittstellen	Kurzbeschreibung
UC-ZM-001	Antragsberechtigte auflisten
UC-ZM-002	Antragsberechtigten hinzufügen
UC-ZM-003	Berechtigte(n) Zertifikatsantragsteller zu bestehendem Antragsberechtigten hinzufügen
UC-ZM-004	Antragsberechtigten bzw. berechtigten Zertifikatsantragsteller ändern
UC-ZM-005	Antragsberechtigten bzw. berechtigten Zertifikatsantragsteller löschen
UC-ZM-006	Zulassungseinträge verifizieren
UC-ZMB-001	Benutzerkennung und RSA Token versenden
UC-ZMB-002	Benutzerkennung und RSA Token freischalten
UC-ZA-001	Zertifikate über Web-Anwendung beziehen
UC-ZA-002	Crypt-ID ausstellen
UC-ZA-003	Zertifikate über CMP/SOAP beziehen
UC-ZS-001	Zertifikate über Web-Anwendung sperren
UC-ZS-002	Zertifikate über SOAP/CMP sperren
UC-ZS-003	Zertifikate durch die gematik sperren

Der Anbieter der Komponenten-PKI stellt sowohl produktive CAs in der Produktionsumgebung (PU) als auch Test-CAs in der Referenz- und Testumgebung (RU/TU) zur Verfügung.

Die Umsetzung der Schnittstellen und Prozesse der Komponenten-PKI ist für die Betriebsumgebungen RU/TU und PU identisch.

Abbildung 2 stellt die bereitgestellten Schnittstellen und spezifizierten Uses Cases und deren Zusammenhang dar.

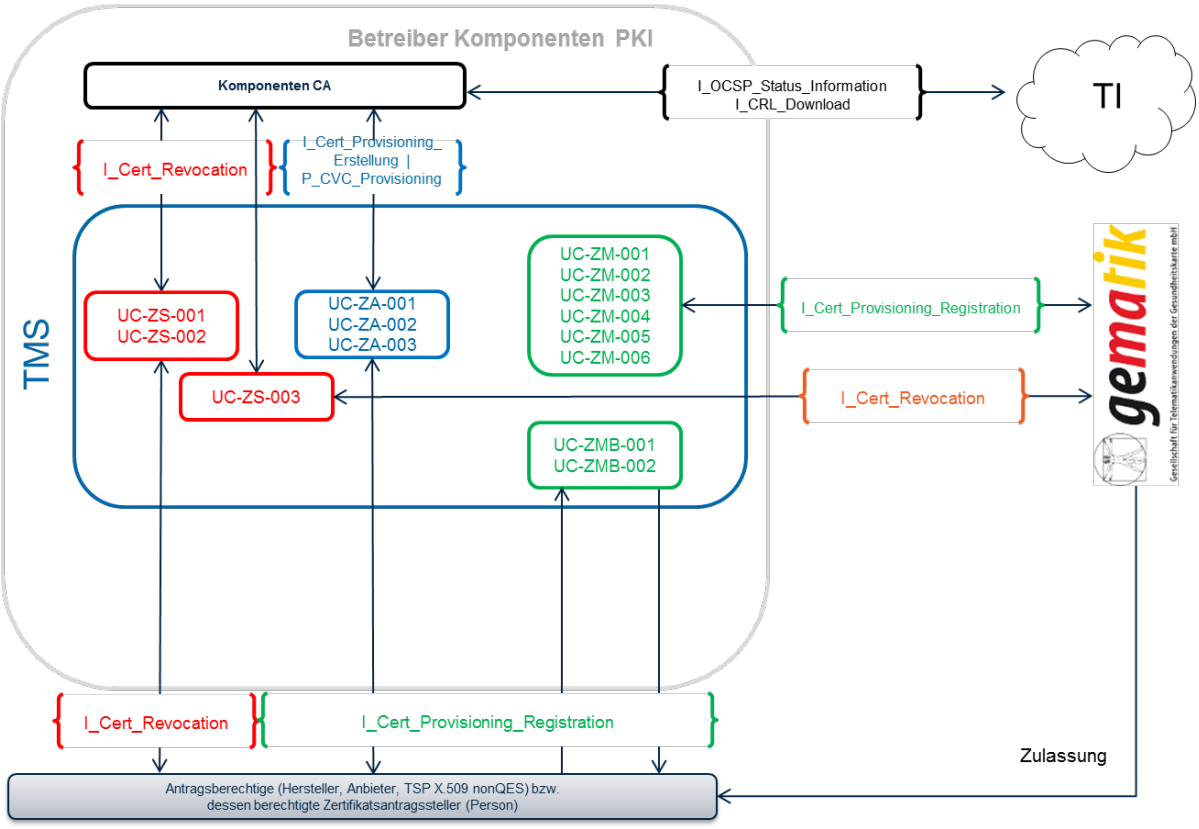

Abbildung 2: Darstellung Schnittstellen und Uses Cases und deren Zusammenhang

2.1 Infrastruktur-CA

Der Betreiber stellt zusätzlich zu Abbildung 1 dargestellten Komponenten-PKI eine separate so genannte „Infrastruktur-CA“ in der gesicherten Betriebsumgebung bereit. Diese wird dazu eingesetzt, um die Sicherheitsanforderungen der Datenintegrität, der Authentizität und der Vertraulichkeit der internen Prozesse zu gewährleisten.

Die Infrastruktur-CA wird für folgende Zwecke eingesetzt:

- Ausstellung von Berechtigungs-X.509-Zertifikaten (Crypt-ID), welche zur Authentisierung an den SOAP/CMP-Schnittstellen eingesetzt werden.
- Ausstellung der internen CMP-Zertifikate
- Ausstellung der notwendigen internen TLS-Zertifikate

Zur Abfrage der Statusinformation aller von der Infrastruktur-CA ausgestellten Zertifikate wird ein (interner) OCSP-Responder (OCSP-ICA) betrieben.

3 Systemkontext

3.1 Akteure und Rollen

3.1.1 Akteure

3.1.1.1 Gematik

Die gematik ist verantwortlich für die Gestaltung und Zulassung der Komponenten-PKI. Sie übernimmt unter anderem die folgenden Aufgaben:

- Zulassung Betreiber der Komponenten-PKI (als TSP-X.509 nonQES_Komp)
- Bereitstellung Berechtigungsinformationen der Berechtigten Zertifikatsantragsteller bzw. Sperrantragsteller
- Sperrberechtigter aller ausgestellten Zertifikate der Komponenten-PKI

3.1.1.2 gematik Root-CA

Die gematik als Verantwortlicher Anbieter der gematik Root-CA beauftragt einen Dienstleister, der diese im Auftrag der gematik betreibt.

Zur Etablierung einer einheitlich geregelten PKI für nonQES-Zertifikate stellt die gematik als Policy-Authority eine zentrale Root-CA für alle zertifikatsausgebenden TSP-X.509 nonQES bereit. Entsprechend werden nonQES-X.509 Aussteller-CA-Zertifikate in der TI durch die „gematik Root-CA“ signiert.

3.1.1.3 TSP-X.509 nonQES

TSP-X.509 nonQES können für ihren Statusinformationsdienst die notwendigen X.509-nonQES OCSP- und CRL-Signerzertifikate bei der Komponenten-PKI (Betreiber OCSP-Signer-CA bzw. CRL-Signer-CA). beantragen.

3.1.1.4 Zertifikatsnehmer

Zertifikatsnehmer der Komponenten-PKI können auch technische Komponenten (z. B. Konnektor, fachanwendungsspezifischer Dienst) sein. Diese Zertifikate werden als Komponentenzertifikate bezeichnet.

3.1.1.5 Hersteller

Die Hersteller von Konnektoren und Kartenterminals beantragen bei der Komponenten-PKI TSP-X.509 nonQES die entsprechenden X.509-Zertifikate und statten ihre Geräte damit aus.

Der Hersteller beauftragt einen Kartenherausgeber mit der Bereitstellung des für das Kartenterminal bzw. für den Konnektor benötigten gerätespezifischen Sicherheitsmoduls gSMC.

Der Hersteller tritt als Sperrberechtigter auf und nutzt dafür die vorgesehenen Schnittstellen der Komponenten-PKI.

3.1.1.6 Anbieter

Anbieter zentraler Dienste (ZD) und fachanwendungsspezifischer Dienste (FD) beantragen bei der Komponenten-PKI für jede Komponente bzw. jeden in der TI etablierten Dienst die notwendigen X.509-Zertifikate.

Der Anbieter tritt als Sperrberechtigter auf und nutzt dafür die vorgesehenen Schnittstellen der Komponenten-PKI.

3.1.1.7 Antragsberechtigter

Antragsberechtigte sind durch die gematik zugelassene TSP X.509 nonQES, Hersteller und Anbieter, die an der gematik Root-CA oder Komponenten-PKI Zertifikate beantragen dürfen.

3.1.1.8 Berechtigter Zertifikatsantragsteller

Berechtigte Zertifikatsantragsteller sind Personen, die im Auftrag zugelassener Antragsberechtigter Zertifikate bei der gematik Root-CA oder Komponenten-PKI Zertifikate beantragen dürfen.

3.1.1.9 Betreiber

Der Betreiber betreibt die Komponenten-PKI im Auftrag der gematik.

3.1.2 Rollen

Für die Umsetzung der im Folgenden beschriebenen Prozesse sind unterschiedliche Rollen notwendig. Die jeweiligen Rollen und deren Berechtigungen sind in den jeweiligen Kapiteln zu den Prozessen Zulassungsmanagement (Kapitel 4.2), Zertifikatsaustellung (Kapitel 5.1) sowie Zertifikatssperrung (Kapitel 6.1) beschrieben.

Die Darstellung der übergeordneten Benutzer- und Rollenadministration ist nicht Bestandteil dieses Dokumentes, da dieses die weiteren PKI-Dienste (gematik Root CA und TSL-Dienst) betrifft. Dieses wird in der Betriebsdokumentation der PKI-Dienste dargestellt.

4 Zulassungsmanagement

Zur Beantragung von Komponenten- und/oder CRL-/OCSP-Signer-Zertifikaten muss der Antragsberechtigte einen Zulassungsantrag bei der Zulassungsstelle der gematik stellen.

Zur Registrierung und Verwaltung Antragsberechtigter und derer Berechtigter Zertifikatsantragsteller wird der gematik eine Applikation "Zulassungsmanagement" (kurz „ZM“) bereitgestellt.

Das Zulassungsmanagement dient dabei als Schnittstelle zwischen gematik und dem Betreiber der Komponenten-PKI sowie dem Betreiber der gematik Root-CA zur Übermittlung der Berechtigungsinformationen der Antragsberechtigten. Antragsberechtigte sind Anbieter, Hersteller, TSP-X.509 nonQES und Betreiber des TSL-Dienstes.

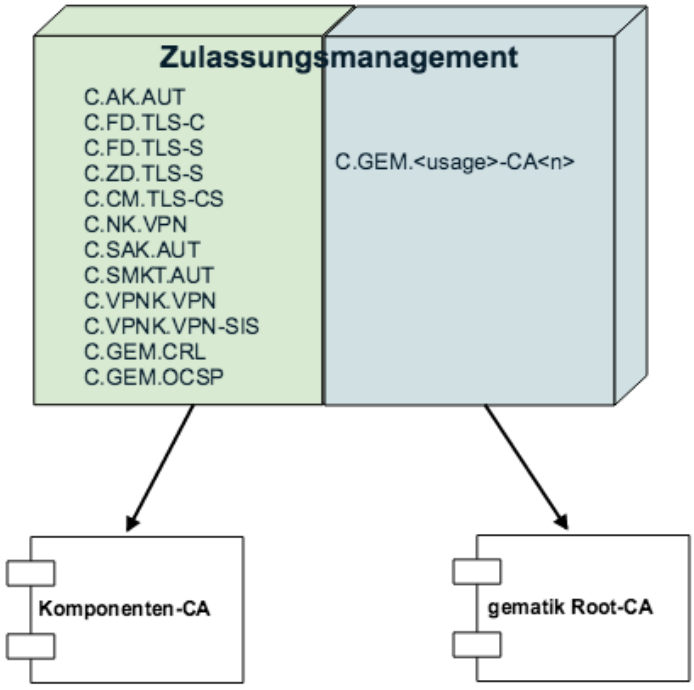

Abbildung 3: Zuordnung der PKI-Zuständigkeiten bzgl. Zertifikatstypen

Durch die gematik zugelassene TSP X.509 nonQES bzw. der zugelassene Betreiber TSL-Dienst ist darüber hinaus berechtigt ein CA-Zertifikat bei der gematik Root-CA zu beantragen. Hierfür erfasst die gematik zu jeder Zulassungsinformation ein spezifisches CA-Einsatzbereich, welcher zur Ausstellung von EE-Zertifikate berechtigt.

4.1 Use Cases Zulassungsmanagement (gematik)

Die Anwendung stellt die folgenden Use Cases zur Verfügung:

Tabelle 8: Use Cases Zulassungsmanagement

Use Case	Beschreibung
UC-ZM-001	Antragsberechtigte auflisten
UC-ZM-002	Antragsberechtigten hinzufügen
UC-ZM-003	Berechtigte(n) Zertifikatsantragsteller zu bestehendem Antragsberechtigten hinzufügen
UC-ZM-004	Antragsberechtigten bzw. berechtigten Zertifikatsantragsteller ändern
UC-ZM-005	Antragsberechtigten bzw. berechtigten Zertifikatsantragsteller löschen
UC-ZM-006	Zulassungseinträge verifizieren

4.2 Rollen und Berechtigungen

Die nachfolgende Tabelle stellt die notwendigen Rollen und Berechtigungen für das Zulassungsmanagement bei gematik und beim Betreiber dar:

Tabelle 9: Rollen und Berechtigungen Zulassungsmanagement

Rolle	Kürzel	Besitzer	Berechtigung
Zulassungs-management Administrator	ZMA	gematik	Administration Zulassungsmanagement (Einsicht, Erstellung, Änderung und Löschung) Antragsberechtigter und Berechtigter Zertifikatsantragsteller.
Zulassungs-management Verifikator	ZMV	gematik	Prüfung und Freigabe der durch den Administrator Zulassungsmanagement durchgeführten Änderungen (Erstellung, Änderung und Löschung) Antragsberechtigter und Berechtigter Zertifikatsantragsteller.
Zulassungs-management Sender	ZMS	Betreiber	Erstellung und Versand der RSA Token an berechtigte Zertifikatsantragsteller
Zulassungs-management Empfänger	ZME	Betreiber	Empfang der Empfangsbestätigung und Aktivierung des Accounts des berechtigten Zertifikatsantragstellers
Anmerkung	Für die Rollen ZMA und ZMV, sowie ZMS und ZME gilt ein Rollenausschluss.		

4.3 Vier-Augen-Prinzip

Um insbesondere dem sehr hohen Schutzziel der Nicht-Abstreitbarkeit und der Rollentrennung des Prozesses gerecht zu werden, müssen alle Zulassungsanträge (Erstellung, Änderung, Löschung) von der berechtigten gematik-Rolle qualifiziert signiert sowie von einer weiteren gematik-Rolle verifiziert und durch erneute qualifizierte Signatur in der Anwendung Zulassungsmanagement bestätigt und freigegeben werden. Erst danach werden alle Signaturen der Zulassungseinträge verarbeitet und in einer zentralen TMS-Datenbank auf Seiten des Betreibers gespeichert.

4.4 Anwendung Zulassungsmanagement (gematik-Sicht)

Die Ausführungen in den nachfolgenden Anwendungsfällen schildern die Aktionen zur Registrierung und Verwaltung berechtigter Hersteller, Anbieter und TSP-X.509 nonQES (Antragsberechtigte) eines zugelassenen Produktes über das Zulassungsmanagement.

Tabelle 10 stellt die URIs des Zulassungsmanagements für die Produktionsumgebung sowie für die Referenz- und Testumgebung dar. Die URI für die Referenz- und Testumgebung ist identisch.

Tabelle 10: URIs des Zulassungsmanagements

Umgebung	URI
PU	https://www.tms.ti-dienste.de/zm
RU/TU	https://www-testref.tms.ti-dienste.de/zm

Alle Änderungen werden im TMS QES-signiert gespeichert. Das Vier-Augen-Prinzip wird anschließend durch den Use Case Zulassungseinträge verifizieren realisiert.

Der Prozess zur Auslieferung der QES-Karten wird vom akkreditierten ZDA definiert.

4.4.1 Antragsberechtigte auflisten

Tabelle 11: UC-ZM-001

Nummer: Name:	UC-ZM-001 Antragsberechtigte auflisten
Kurzbeschreibung	Dieser Anwendungsfall ermöglicht dem Akteur einen einfachen und schnellen Überblick über alle Antragsberechtigten, die im Zulassungsmanagement geführt werden. Darüber hinaus kann der Akteur auch nach einem Antragsberechtigten durch Auswahl geeigneter Kriterien suchen.
Auslösender Akteur	ZMA/ZMV
Vorbedingungen	Dem Akteur werden der RSA-Token sowie die QES-Karte

	zur Verfügung gestellt. Der Akteur ist in der Applikation Zulassungsmanagement angemeldet und autorisiert.
Eingangsdaten	
Ergebnisse	Liste der von der Applikation geführten Antragsberechtigten mit ihren wesentlichen Merkmalen.
Anmerkungen	

Tabelle 12: Prozessschritte UC-ZM-001

Nr.	Akteur	Prozessschritt
1.	ZMA/ZMV	Der Akteur löst in der Applikation die Funktion "Antragsberechtigte auflisten" aus.
2.	Applikation	Es wird eine Übersicht der im System vorhandenen Antragsberechtigten angezeigt.
3.	Applikation	Die Applikation ermöglicht auch die Angabe von Selektionskriterien zur Suche nach Antragsberechtigten. Selektionskriterien sind mindestens: • Name der Organisation • Mandanten-ID der Organisation • Zugelassene(s) Produkt(e)
4.	ZMA	Ggf. erfasst der Akteur Selektionskriterien und löst die Funktion "Antragsberechtigte suchen" aus.
5.	Applikation	Die Applikation listet die Antragsberechtigten, die den Kriterien genügen, mit ihren wesentlichen Merkmalen auf. Zu diesen Merkmalen gehören mindestens: • Name der Organisation • Mandanten-ID der Organisation • zugelassene(s) Produkt(e) • Liste zugeordneter Domains (FQDN) (nur für Produkttyp VSDM, VSDM Intermediär, Konfigurationsdienst, Verzeichnisdienst, KOM-LE Fachdienst oder VPN-Zugangsdienst) • spezifische CA-Einsatzbereiche gemäß gemSpec_TSP_X509#Tab_PKI_213 (nur TSP-X.509 non QES und TSL-Dienst für gematik Root

		CA)
6.	ZMA	Der Akteur wählt eine Funktion aus: • Erstellen eines Antragsberechtigten • Hinzufügen eines Berechtigten Zertifikatsantragstellers • Ändern eines Antragsberechtigten • Löschen eines Antragsberechtigten • Schließen
7.	ZMA	Die Applikation führt die gewählte Funktion aus. Fallunterscheidung: • Akteur wählt die Funktion "Erstellen eines Antragsberechtigten": Die Applikation verzweigt in den entsprechenden Anwendungsfall zur Erstellung eines Antragsberechtigten. • Akteur selektiert einen Antragsberechtigten und wählt die Funktion "Hinzufügen eines Berechtigten Zertifikatsantragstellers": Die Applikation verzweigt in den entsprechenden Anwendungsfall zum Hinzufügen eines Berechtigten Zertifikatsantragstellers. • Akteur selektiert einen Antragsberechtigten und wählt die Funktion "Ändern eines Antragsberechtigten": Die Applikation verzweigt in den entsprechenden Anwendungsfall zum Ändern eines Antragsberechtigten bzw. Berechtigten Zertifikatsantragstellers. • Akteur selektiert einen Antragsberechtigten und wählt die Funktion "Löschen eines Antragsberechtigten": Die Applikation verzweigt in den entsprechenden Anwendungsfall zum Löschen des Antragsberechtigten bzw. Berechtigten Zertifikatsantragstellers. • Akteur wählt eine andere Funktion: Die Applikation führt die entsprechende Aktion aus.

4.4.2 Antragsberechtigten hinzufügen

Tabelle 13: UC-ZM-002

Nummer: Name:	UC-ZM-002 Antragsberechtigten hinzufügen
Kurzbeschreibung	Dieser Anwendungsfall beschreibt die Einrichtung eines neuen Antragsberechtigten in der Anwendung Zulassungsmanagement.
Auslösender Akteur	ZMA
Vorbedingungen	Dem Akteur wurde der RSA-Token sowie QES-Karte zur Verfügung gestellt. Der Akteur ist in der Applikation und im QSC angemeldet und autorisiert.
Eingangsdaten	• Daten gemäß Tabelle 28: Daten des Antragsberechtigten (Organisation) • Daten gemäß Tabelle 29: Daten Berechtigter Zertifikatsantragsteller • aktuelle Zeit
Ergebnisse	Ein neuer Antragsberechtigter (Anbieter, Hersteller, TSP oder Betreiber des TSL-Dienstes) sowie namentlich genannte berechtigte Zertifikatsantragsteller wurden erstellt. Die erfassten Daten wurden (QES signiert) in der TMS-Datenbank zur Freigabe gespeichert. Rolleninhaber ZMV erhalten Information (per E-Mail) über vorliegenden Zulassungseintrag zur Freigabe.
Anmerkungen	Eine Organisation kann mehrere Produkttypen gemäß Tabelle 45 (für die die Organisation durch die gematik zugelassen wurde) sowie mehrere spezifische CA-Einsatzbereiche und mehrere Berechtigte Zertifikatsantragsteller besitzen. Zu jedem Produkttyp gehören verschiedene Zertifikatstypen (z. B. weist der Produkttyp "Konnektor" die Zertifikate vom Typ Netzkonnektor, Anwendungskonnektor und Signatur-anwendungskomponente auf). Einem Berechtigten Zertifikatsantragsteller können ein oder mehrere Produkttypen (nur die Produkttypen, für die die Organisation zugelassen ist) zugewiesen werden. Jedem "Berechtigten Zertifikatsantragsteller-Produkttyp" können nun die Berechtigungen "Erstellen", "Sperren" oder "Erstellen und Sperren" zugeordnet werden.

	Beispiel: Einem Berechtigten Zertifikatsantragsteller, dem der Produkttyp Konnektor mit der Berechtigung "Erstellen" zugewiesen wurde, kann über die Anwendung "Zertifikatsaustellung / Zertifikatssperrung" Zertifikate vom Typ Netzkonnektor, Anwendungskonnektor und Signaturanwendungskomponente beziehen. Einem Berechtigten Zertifikatsantragsteller, dem der Produkttyp TSP X.509nonQES mit der Berechtigung "Erstellen" zugewiesen wurde, kann über die Anwendung "Sub-CA-Zertifikatbeantragung (P_Sub_CA_Certification-_X.509)" der gematik Root CA Zertifikate vom Typ C.GEM.<USAGE>-CA<n>, beziehen. Einem Berechtigten Zertifikatsantragsteller, dem der Produkttyp TSL-Dienst mit der Berechtigung "Erstellen" zugewiesen wurde, kann über die Anwendung Sub-CA-Zertifikatbeantragung (P_Sub_CA_Certification_X.509)" der gematik Root CA Zertifikate vom Typ C.GEM.TSL-CA<n>, beziehen. Ein Berechtigter Zertifikatsantragsteller, dem der Produkttyp VSDM, VSDM Intermediär, Konfigurationsdienst, Verzeichnisdienst, KOM-LE Fachdienst und VPN-Zugangsdienst mit der Berechtigung "Erstellen" zugewiesen wurde, darf nur Werte aus dem im Rahmen ihrer Anbieterzulassung zugewiesenen FQDNs aus dem Namensraum der TI beziehen. Die Werte werden von der gematik als Zulassungsinformation mitgeliefert.

Tabelle 14: Prozesschritte UC-ZM-002

Nr.	Akteur	Prozessschritt
1.	ZMA	Der Akteur wählt in der Applikation die Funktion "Erstellen eines Antragsberechtigten" aus.
2.	Applikation	Die Applikation stellt eine Eingabemaske mit allen Eingabefeldern für die notwendigen Antragsberechtigten-Daten zur Verfügung.
3.	ZMA	Der Akteur gibt die erforderlichen Daten ein und speichert diese.
4.	Applikation	Die Applikation prüft die Eingaben auf Vollständigkeit und Plausibilität. Fehlerfall: • Unvollständige oder fehlende Daten: Die Applikation kehrt zur Eingabemaske zurück und zeigt die

		festgestellten Plausibilitätsverletzungen, z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese daraufhin korrigieren.
5.	Applikation	Die Applikation fordert den Akteur zur Bestätigung der Erfassung auf.
6.	ZMA	Der Akteur bestätigt die Erfassung.
7.	Applikation	Die Applikation meldet die erfolgreiche Erfassung und kehrt zur aufrufenden Funktion zurück.
8.	ZMA	Der Akteur signiert mittels des QSC die erfassten Daten.
9.	TMS	Das TMS prüft die mit dem QSC erzeugte Signatur. Fehlerfälle: a) Fehlerhafte Signatur: Die Applikation stellt die fehlgeschlagene Signaturprüfung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles. b) Nicht zugelassene Signatur: Die Applikation stellt die fehlende Berechtigung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles.
10.	QSC	Der QSC meldet die erfolgreiche Signatur des Eintrags und kehrt zur aufrufenden Funktion zurück.
11.	TMS	Das TMS speichert die erfassten Daten in der TMS Datenbank und protokolliert den Vorgang.
12.	TMS	Das TMS meldet den Eingang erfasster Daten per E-Mail an alle Rolleninhaber „ZMV“

4.4.3 Berechtigten Zertifikatsantragsteller zu bestehendem Antragsberechtigten hinzufügen

Tabelle 15: UC-ZM-003

Nummer: Name:	UC-ZM-003 Berechtigten Zertifikatsantragsteller zu bestehendem Antragsberechtigten hinzufügen
Kurzbeschreibung	Dieser Anwendungsfall beschreibt das Hinzufügen eines Berechtigten Zertifikatsantragsteller zu einem bestehenden

	Antragsberechtigten.
Auslösender Akteur	ZMA
Vorbedingungen	Dem Akteur wurden der RSA-Token sowie die QES-Karte zur Verfügung gestellt. Der Akteur ist in der Applikation und im QSC angemeldet und autorisiert.
Eingangsdaten	• Daten gemäß Tabelle 29: Daten Berechtigter Zertifikatsantragsteller • aktuelle Zeit
Ergebnisse	Ein neuer zur Freigabe Berechtigter Zertifikatsantragsteller wurde (QES signierte Daten) im TMS zur Freigabe eingetragen. Rolleninhaber ZMV erhalten Information (per E-Mail) über vorliegenden Zulassungseintrag zur Freigabe
Anmerkungen	

Tabelle 16: Prozessschritte UC-ZM-003

Nr.	Akteur	Prozessschritt
1.	ZMA	Der Akteur ruft die Funktion "Antragsberechtigten auflisten" auf.
2.	Applikation	Die Applikation zeigt dem Akteur eine Liste der Antragsberechtigten inkl. der Berechtigten Zertifikatsantragsteller an.
3.	ZMA	Der Akteur wählt einen Antragsberechtigten aus und löst die Funktion "Hinzufügen eines Berechtigten Zertifikatsantragsteller" aus.
4.	Applikation	Die Applikation stellt eine Eingabemaske mit allen Eingabefeldern für die notwendigen Daten des Berechtigten Zertifikatsantragstellers zur Verfügung.
5.	ZMA	Der Akteur gibt die erforderlichen Daten ein und speichert diese.
6.	Applikation	Die Applikation prüft die Eingaben auf Vollständigkeit und Plausibilität. Fehlerfälle: Unvollständige oder fehlende Daten: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten

		Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung. Der Akteur kann diese anschließend korrigieren.
7.	Applikation	Die Applikation fordert den Akteur zur Bestätigung der Erfassung auf.
8.	ZMA	Der Akteur bestätigt die Erfassung.
9.	Applikation	Die Applikation meldet die erfolgreiche Erfassung und kehrt zur aufrufenden Funktion zurück.
10.	ZMA	Der Akteur signiert mittels des QSCs die erfassten Daten.
11.	TMS	Das TMS prüft die mit dem QSC erzeugte Signatur. Fehlerfälle: a) Fehlerhafte Signatur: Die Applikation stellt die fehlgeschlagene Signaturprüfung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles. b) Nicht zugelassene Signatur: Die Applikation stellt die fehlende Berechtigung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles.
12.	QSC	Der QSC meldet die erfolgreiche Signatur des Eintrags und kehrt zur aufrufenden Funktion zurück
13.	TMS	Das TMS speichert die erfassten Daten in der TMS Datenbank und protokolliert den Vorgang.
14.	TMS	Das TMS meldet den Eingang erfasster Daten per E-Mail an alle Rolleninhaber „ZMV“

4.4.4 Antragsberechtigten bzw. Berechtigten Zertifikatsantragsteller ändern

Tabelle 17: UC-ZM-004

Nummer: Name:	UC-ZM-004 Antragsberechtigten bzw. Berechtigten Zertifikatsantragsteller ändern
Kurzbeschreibung	Dieser Anwendungsfall ermöglicht das Ändern der Daten eines Antragsberechtigten.

Auslösender Akteur	ZMA
Vorbedingungen	Dem Akteur wurden die ein RSA-Token sowie ein QES-Karte zur Verfügung gestellt. Der Akteur ist in der Applikation und im QSC angemeldet und autorisiert.
Eingangsdaten	Aktuelle Antragsberechtigten-Daten
Ergebnisse	Geänderte Daten o Tabelle 29: Daten Berechtigter Zertifikatsantragsteller und/oder o Daten gemäß Tabelle 28: Daten des Antragsberechtigten (Organisation) wurden (QES signierte Daten) im TMS zur Freigabe eingetragen. Rolleninhaber ZMV erhalten Information (per E-Mail) über vorliegenden Zulassungseintrag zur Freigabe.
Anmerkungen	

Tabelle 18: Prozessschritte UC-ZM-004

Nr.	Akteur	Prozessschritt
1.	ZMA	Der Akteur ruft die Funktion "Antragsberechtigte auflisten" auf.
2.	Applikation	Die Applikation zeigt dem Akteur eine Liste der Antragsberechtigten inkl. deren Berechtigten Zertifikatsantragsteller an.
3.	ZMA	Der Akteur selektiert einen Antragsberechtigten und löst die Funktion "Ändern eines Antragsberechtigten" aus.
4.	Applikation	Die Applikation stellt eine Eingabemaske mit allen Eingabefeldern zum Ändern der Antragsberechtigten-Daten dar. Die Felder sind mit denen in der Datenbank gespeicherten Daten vorbelegt und können einzeln geändert werden.
5.	ZMA	Fallunterscheidung: a) Der Akteur ändert die Organisations-Daten. Fallunterscheidung:

		o Der Status des Antragsberechtigten wird von "aktiviert" auf "gesperrt" gesetzt: Der Status aller Berechtigter Zertifikatsantragsteller des Antragsberechtigten wird auf deaktiviert gesetzt ("eingefroren"). Die Berechtigten Zertifikatsantragsteller können sich nicht an der Web-Anwendung Zertifikatsausstellung / Zertifikatssperrung bzw. CA Zertifikatsbeantragung anmelden. Die SOAP- bzw. CMP-Zertifikate werden im OCSP-Responder der Infrastruktur CA gesperrt. Weiter mit Schritt 8. o Der Status des Antragsberechtigten wird von "gesperrt" auf "aktiviert" gesetzt: Der Status aller Berechtigter Zertifikatsantragsteller des Antragsberechtigten wird auf aktiviert gesetzt. Die Berechtigten Zertifikatsantragsteller können sich wieder an der Web-Anwendung Zertifikatsausstellung / Zertifikatssperrung bzw. CA Zertifikatsbeantragung anmelden. Das SOAP- bzw. CMP-Zertifikat bleibt im OCSP-Responder der Infrastruktur-CA gesperrt. Zur Beantragung von Zertifikaten über CMP/SOAP muss der Berechtigte Zertifikatsantragsteller eine neue CryptID beantragen. Weiter mit Schritt 8. b) Der Akteur selektiert einen Berechtigten Zertifikatsantragsteller. Weiter mit Schritt 6.
6.	Applikation	Die Applikation stellt eine Eingabemaske mit allen Eingabefeldern zum Ändern des Berechtigten Zertifikatsantragstellers dar. Die Felder sind mit denen in der Datenbank gespeicherten Daten vorbelegt und können einzeln geändert werden
7.	ZMA	Der Akteur ändert die Daten des Berechtigten Zertifikatsantragstellers. Fallunterscheidung:

		a) Der Status des Berechtigten Zertifikatsantragstellers wird von "aktiviert" auf "deaktiviert" gesetzt: Der Berechtigte Zertifikatsantragsteller kann sich nicht an der Web-Anwendung Zertifikatsausstellung / Zertifikatssperrung bzw. CA Zertifikatsbeantragung anmelden. Das SOAP- bzw. CMP-Zertifikat wird im OCSP- Responder der Infrastruktur-CA gesperrt. b) Der Status des Berechtigten Zertifikatsantragsteller wird auf aktiviert gesetzt: Der Berechtigte Zertifikatsantragsteller kann sich wieder an der Web-Anwendung Zertifikatsausstellung / Zertifikatssperrung bzw. CA Zertifikatsbeantragung anmelden. Das SOAP- bzw. CMP-Zertifikat bleibt im OCSP-Responder der Infrastruktur-CA gesperrt. Zur Beantragung von Zertifikaten über CMP/SOAP muss der Berechtigte Zertifikatsantragsteller eine neue CryptID beantragen.
8.	ZMA	Der Akteur ändert die Daten des Antragsberechtigten.
9.	Applikation	Die Applikation prüft die Eingaben auf Vollständigkeit und Plausibilität. Fehlerfälle: • Unvollständige oder fehlende Daten: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung an. Der Akteur kann diese korrigieren.
10.	Applikation	Die Applikation fordert den Akteur zur Bestätigung der Änderung auf.
11.	ZMA	Der Akteur bestätigt die Erfassung.
12.	TMS	Das TMS speichert die erfassten Daten in der TMS Datenbank und protokolliert den Vorgang.
13.	Applikation	Die Applikation meldet die erfolgreiche Erfassung und kehrt zur aufrufenden Funktion zurück.
14.	ZMA	Der Akteur signiert mittels des QSCs die erfassten Daten.
15.	TMS	Das TMS prüft die mit dem QSC erzeugte Signatur. Fehlerfälle: a) Fehlerhafte Signatur: Die Applikation stellt

		die fehlgeschlagene Signaturprüfung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles. b) Nicht zugelassene Signatur: Die Applikation stellt die fehlende Berechtigung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles.
16.	QSC	Der QSC meldet die erfolgreiche Signatur des Eintrags und kehrt zur aufrufenden Funktion zurück.
17.	TMS	Die Applikation meldet den Eingang erfasster Daten per E-Mail an alle Rolleninhaber „ZMV“

4.4.5 Antragsberechtigten bzw. Berechtigten Zertifikatsantragsteller löschen

Tabelle 19: UC-ZM-005

Nummer: Name:	UC-ZM-005 Antragsberechtigten bzw. Berechtigten Zertifikatsantragsteller löschen
Kurzbeschreibung	Dieser Anwendungsfall beschreibt das Löschen eines Antragsberechtigten. Der Akteur hat die Möglichkeit einen Antragsberechtigten mit allen zugehörigen Daten oder einzelne Berechtigte Zertifikatsantragsteller zu löschen.
Auslösender Akteur	ZMA
Vorbedingungen	Dem Akteur wurden ein RSA-Token sowie ein QES-Karte zur Verfügung gestellt. Der Akteur ist in der Applikation und im QSC angemeldet und autorisiert.
Eingangsdaten	
Ergebnisse	Ein Antragsberechtigter oder Berechtigter Zertifikatsantragsteller wurde gelöscht.
Anmerkungen	

Tabelle 20: Prozessschritte UC-ZM-005

Nr.	Akteur	Prozessschritt
1.	ZMA	Der Akteur ruft die Funktion "Antragsberechtigte auflisten" auf.
2.	Applikation	Die Applikation zeigt dem Akteur eine Liste der Antragsberechtigten inkl. deren Berechtigten Zertifikatsantragsteller an.
3.	ZMA	Der Akteur selektiert einen Antragsberechtigten und wählt die Funktion "Löschen eines Antragsberechtigten " aus.
4.	Applikation	Die Applikation stellt einen Dialog zum Löschen des Antragsberechtigten dar und listet alle zum Antragsberechtigten gehörenden Berechtigten Zertifikatsantragsteller auf.
5.	ZMA	Fallunterscheidung: a) Der Akteur löscht den selektierten Antragsberechtigten mit allen zugehörigen Berechtigten Zertifikatsantragstellern. Das SOAP- bzw. CMP-Zertifikat wird im OCSP-Responder gesperrt. Weiter mit Schritt 8. b) Der Akteur selektiert einen Berechtigten Zertifikatsantragsteller. Weiter mit Schritt 6.
6.	Applikation	Die Applikation stellt einen Dialog zum Löschen des selektierten Berechtigten Zertifikatsantragstellers dar.
7.	ZMA	Der Akteur löscht den Berechtigten Zertifikatsantragsteller.
8.	ZMA	Der Akteur löscht den Antragsberechtigten.
9.	Applikation	Die Applikation fordert den Akteur zur Bestätigung der Löschung auf.
10.	ZMA	Der Akteur bestätigt die Erfassung.
11.	TMS	Das TMS speichert die erfassten Daten in der TMS Datenbank und protokolliert den Vorgang.
12.	Applikation	Die Applikation meldet die erfolgreiche Erfassung und kehrt zur aufrufenden Funktion zurück.
13.	ZMA	Der Akteur signiert mittels des QSCs die erfassten Daten.
14.	TMS	Das TMS prüft die mit dem QSC erzeugte Signatur. Fehlerfälle:

		a) Fehlerhafte Signatur: Die Applikation stellt die fehlgeschlagene Signaturprüfung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles. b) Nicht zugelassene Signatur: Die Applikation stellt die fehlende Berechtigung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles.
15.	TMS	Das TMS markiert die zu löschenden Daten als gelöscht in der Datenbank und protokolliert den Vorgang.
16.	QSC	Der QSC meldet die erfolgreiche Signatur des Eintrags und kehrt zur aufrufenden Funktion zurück
17.	TMS	Das TMS meldet den Eingang der Löschung per E-Mail an alle Rolleninhaber „ZMV“.

4.4.6 Zulassungseinträge verifizieren

Tabelle 21: UC-ZM-006

Nummer: Name:	UC-ZMS-006 Zulassungseinträge verifizieren
Kurzbeschreibung	Dieser Anwendungsfall ermöglicht die Freigabe der zuvor von einem Akteur ZMA im Zulassungsmanagement erfassten, geänderten oder gelöschten Daten eines Antragsberechtigten bzw. Berechtigten Zertifikatsantragstellers.
Auslösender Akteur	ZMV
Vorbedingungen	Neuer Zulassungseintrag, -löschung oder -änderung im Zulassungsmanagement durch den ZM-Administrator. Dem Akteur wurden die RSA-Token sowie eine QES-Karte zur Verfügung gestellt. Der Akteur ist im QSC angemeldet und autorisiert. Auf Seiten der gematik wird zur Anbindung an das TMS eine lokale Applikation installiert, mit der die Eintragsverwaltung vorgenommen wird. Diese Anwendung verwaltet zudem das lokale Lesegerät zur Authentisierung und Signaturerstellung.
Eingangsdaten	• Vom ZMA erfasste, geänderte oder gelöschte

	Daten eines Antragsberechtigten bzw. Berechtigten Zertifikatsantragstellers. • Signatur des ZMA. • Aktuelle Zeit.
Ergebnisse	Freigegebene oder Abgewiesene (QES signierte) Daten in der TMS-Datenbank.
Anmerkungen	

Tabelle 22: Prozessschritte UC-ZM-006

Nr.	Akteur	Prozessschritt
1.	ZMV	Der Akteur ruft den QSC auf.
2.	QSC	Der QSC zeigt dem Akteur eine Liste der freizugebenden Zulassungseinträge an.
3.	ZMV	Der Akteur selektiert einen Zulassungseintrag.
4.	QSC	Der QSC stellt die ausgewählten Zulassungsinformationen zur Validierung der Daten dar.
5.	ZMV	Fallunterscheidung: a) Der Akteur gibt die TSP Daten frei: Weiter mit Schritt 6. b) Der Akteur widerspricht der Freigabe: Dem Akteur wird eine Eingabemaske mit einem Eingabefeld zum Erfassen einer Begründung dargestellt. Der Akteur erfasst die Begründung. Das TMS erzeugt eine E-Mail mit der Begründung sowie dem Hinweis, dass der Zulassungseintrag abgelehnt wurde und sendet diese an alle Benutzer mit der Rolle "ZMA". Der zur Verifikation vorliegende Zulassungseintrag wird vollständig verworfen. Der Vorgang wird protokolliert. Der Use Case ist beendet.
6.	QSC	Der QSC fordert den Akteur zur Bestätigung der Freigabe auf.
7.	ZMA	Der Akteur bestätigt die Daten und signiert die zu bestätigenden Daten.
8.	TMS	Das TMS prüft die mit dem QSC erzeugte Signatur. Fehlerfälle: a) Fehlerhafte Signatur: Die Applikation stellt die fehlgeschlagene Signaturprüfung durch eine

		eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles. b) Nicht zugelassene Signatur: Die Applikation stellt die fehlende Berechtigung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles.
9.	TMS	Das TMS speichert die freigegebenen Daten in der TMS Datenbank und protokolliert den Vorgang.
10.	TMS	Das TMS meldet den Eingang der freigegebenen Daten per E-Mail an alle Rolleninhaber "ZMS".
11.	QSC	Der QSC meldet die erfolgreiche Freigabe und kehrt zur aufrufenden Funktion zurück

4.4.7 Prozessdarstellung Zulassungsmanagement (gematik)

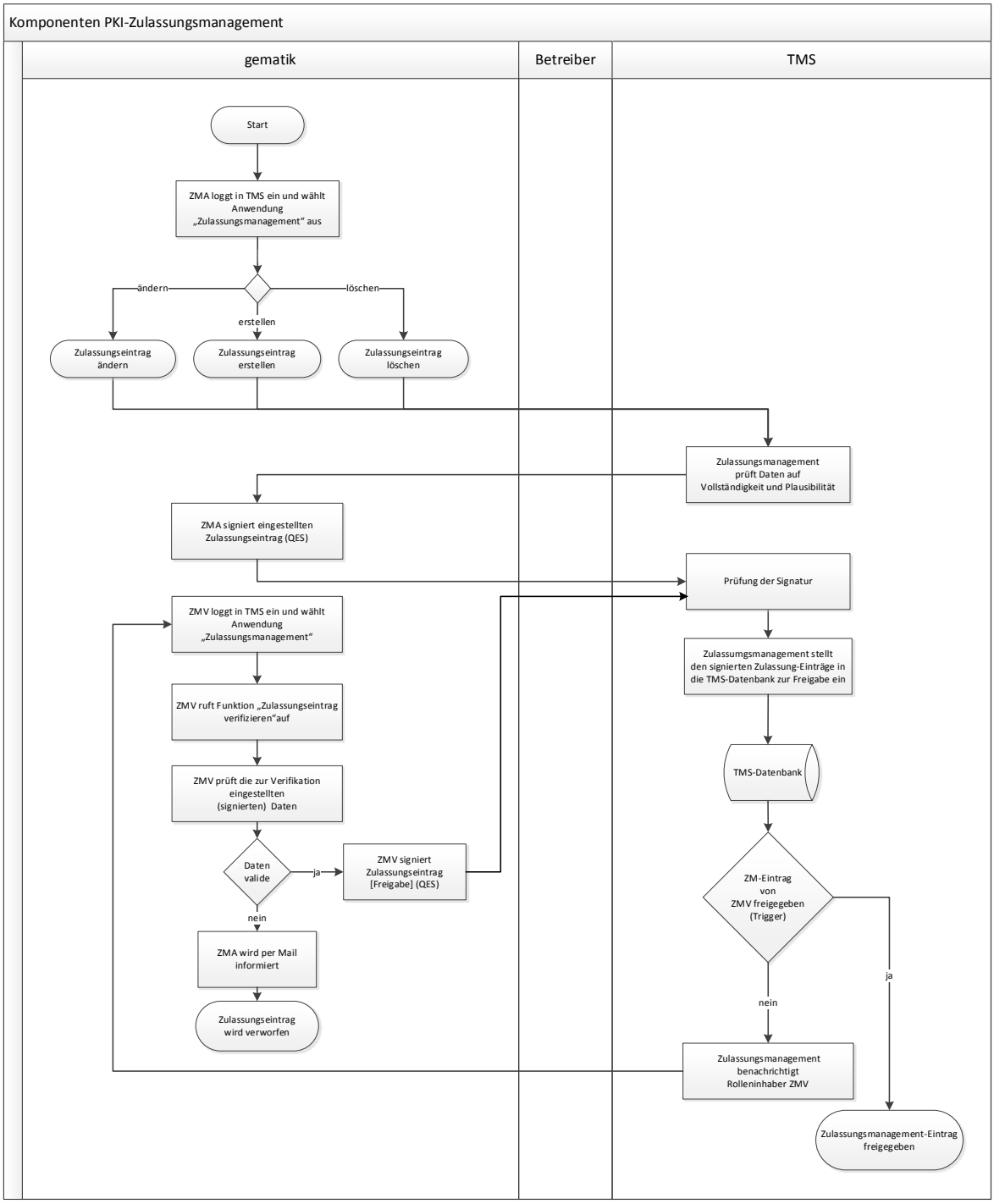

Abbildung 4: Prozessdarstellung Zulassungsmanagement (gematik Sicht)

4.5 Use Cases Zulassungsmanagement (Betreiber)

Zur Verwaltung berechtigter Hersteller, Anbieter und TSP-X.509 nonQES (Berechtigte Antragsteller) auf Seiten des Betreibers der Komponenten PKI wird ebenfalls die Anwendung "Zulassungsmanagement" bereitgestellt.

Die Anwendung stellt dem Betreiber die folgenden Anwendungsfälle (Use Cases) zur Verfügung:

Tabelle 23: Use Cases Zulassungsmanagement Betreiber-Sicht

Use Case	Beschreibung
UC-ZMB-001	Benutzerkennung und RSA Token versenden
UC-ZMB-002	Benutzerkennung und RSA Token freischalten

Zur internen Verwaltung der Berechtigungen von Zertifikatsantragstellern wird ein vom Betreiber bereit gestelltes Active Directory (kurz: „AD“) verwendet. Die Integration des eingesetzten Authentisierungsmechanismus RSA-Token zur Authentisierung an der Applikation erfolgt somit über das AD. Unabhängig vom Status des Benutzerkontos im AD erfolgt die Sperrung bzw. Entsperrung des Antragsberechtigten bzw. der Berechtigten Zertifikatsantragsteller im Zulassungsmanagement durch Änderung dessen Status. Ist der der Antragsberechtigten bzw. des Berechtigten Zertifikatsantragstellers. Sofern der Status des Antragsberechtigten gesperrt wird, gilt diese Sperrung auch für alle zugehörigen Berechtigten Zertifikatsantragsteller.

4.5.1 Benutzerkennung und RSA Token versenden

Tabelle 24: UC-ZMB-001

Nummer: **Name:**	**UC-ZMB-001** **Benutzerkennung und RSA Token versenden**
Kurzbeschreibung	Dieser Use Case beschreibt, wie einem Berechtigten Zertifikatsantragsteller die Benutzerkennung und der RSA-Token für die Anwendungen Zertifikatsausstellung und Zertifikatssperrung bereitgestellt werden.
Auslösender Akteur	ZMS
Vorbedingungen	Der Akteur ist in der Anwendung "Zulassungsmanagement" angemeldet und autorisiert. ZMV hat Berechtigungsinformationen (Berechtigte Zertifikatsantragsteller) freigegeben.
Eingangsdaten	Berechtigte Zertifikatsantragsteller

Ergebnisse	Für die Berechtigten Zertifikatsantragsteller wird ein Benutzerkonto im AD eingerichtet. Die zugehörige Benutzerkennung und der dazu ausgestellte RSA-Token werden an den Berechtigten Zertifikatsantragsteller per Einschreiben/ Rückschein verschickt.
Anmerkungen	Benutzerkonto wird für den Berechtigten Zertifikatsantragsteller mit Status "deaktiviert" eingerichtet. Der Rückschein ist an die Rolle ZME zu adressieren.

Tabelle 25: Prozessschritte UC-ZMB-001

Nr.	Akteur	Prozessschritt
1.	ZMS	Der Akteur ruft die Funktion "Berechtigte Zertifikatsantragsteller auflisten" auf.
2.	Applikation	Die Applikation zeigt dem Akteur eine Liste der eingestellten berechtigten Zertifikatsantragsteller an.(READ-ONLY)
3.	ZMS	Der Akteur selektiert einen Berechtigten Zertifikatsantragsteller.
4.	ZMS	Der Akteur erstellt ein Benutzerkonto (Status „deaktiviert“) für den Berechtigten Zertifikatsantragsteller mit den Daten des Zertifikatsantragstellers (Name, Vorname, Benutzerkennung etc.) aus dem Zulassungsmanagement im AD.
5.	ZMS	Der Akteur erstellt ein Anschreiben für den Berechtigten Zertifikatsantragsteller. Das Anschreiben enthält mindestens die für Versendung und Verwendung des RSA-Tokens notwendigen Daten gemäß Tabelle 30: Daten Anschreiben.
6.	ZMS	ZM-Sender versendet das Anschreiben inklusive des RSA-Tokens per Einschreiben mit Rückschein.

4.5.2 Benutzerkennung und RSA-Token freischalten

Tabelle 26: UC-ZMB-002

Nummer: **Name:**	**UC-ZMB-002** **Benutzerkennung und RSA-Token freischalten**
Kurzbeschreibung	Dieser Anwendungsfall beschreibt, wie nach Empfang des quittierten Einschreibens einem Berechtigten Zertifikatsantragsteller das Benutzerkonto zur Anmeldung an der Applikation zur Zertifikatsausstellung /

	Zertifikatssperrung freigeschaltet wird.
Auslösender Akteur	ZME
Vorbedingungen	Der Akteur ist in der Anwendung "Zulassungsmanagement" angemeldet und autorisiert. ZMS hat Anschreiben und RSA-Token versendet. Einschreiben Rückschein mit Empfangsbestätigung ist eingegangen.
Eingangsdaten	Einschreiben Rückschein mit Empfangsbestätigung
Ergebnisse	Das Benutzerkonto im AD des Zertifikatsantragstellers ist aktiviert.
Anmerkungen	

Tabelle 27: Prozessschritte UC-ZMB-002

Nr.	Akteur	Prozessschritt
1.	ZME	Der Akteur identifiziert den zum Einschreiben Rückschein zugehörigen Berechtigten Zertifikatsantragsteller (anhand der Benutzerkennung) im AD.
2.	ZME	Der Akteur setzt den Status des Benutzerkontos im AD auf „aktiv“.
3.	ZME	Der Akteur protokolliert den Vorgang.
4.	TMS	Das TMS informiert den Berechtigten Zertifikatsantragsteller per E-Mail über die erfolgreiche Freischaltung.

4.5.3 Prozessdarstellung Zulassungsmanagement (Betreiber)

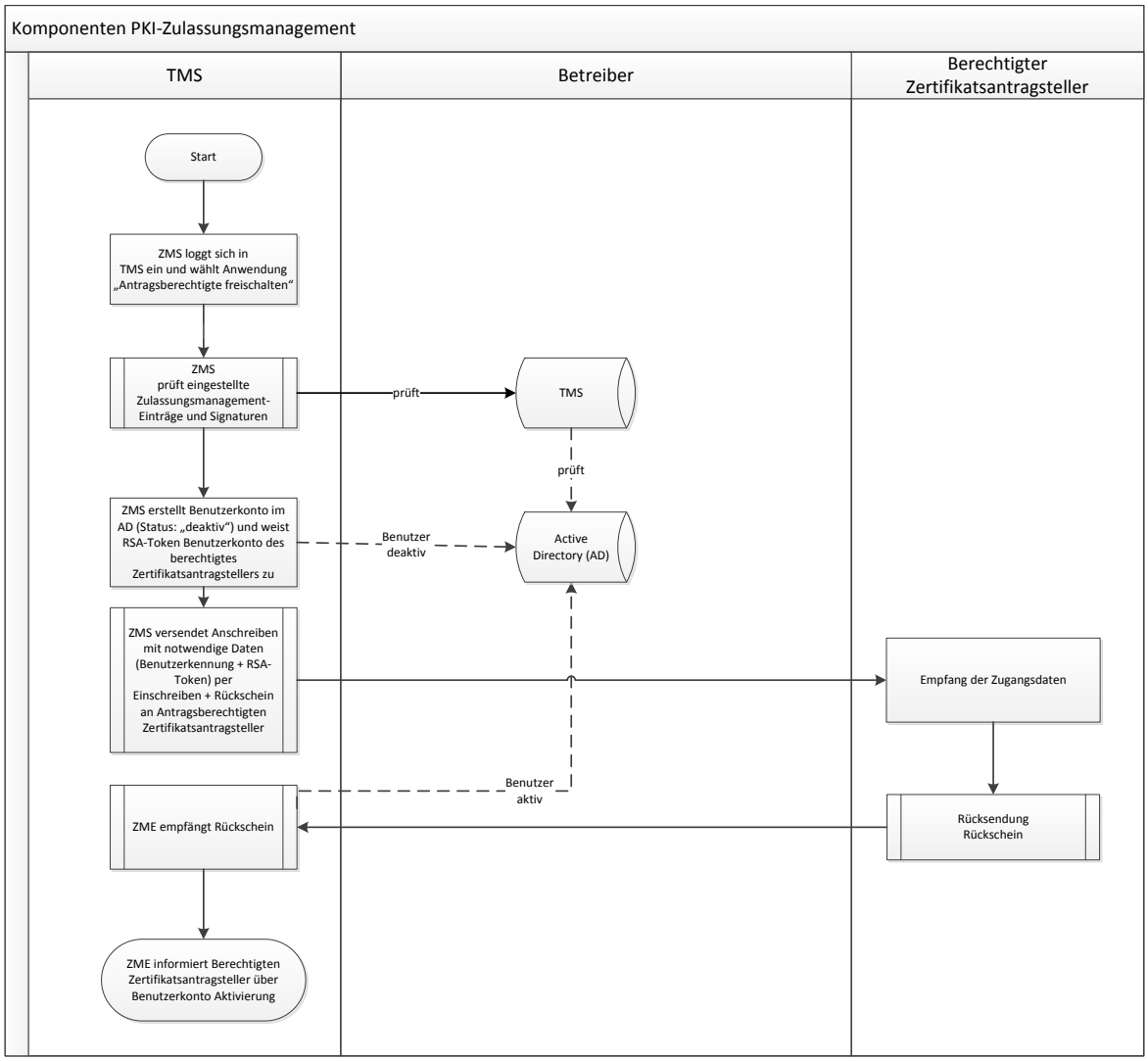

Abbildung 5: Prozessdarstellung Zulassungsmanagement (Betreiber Sicht)

4.6 Artefakte

4.6.1 Eingangsdaten

Die folgenden Tabellen führen die in den Use Cases des Zulassungsmanagement notwendigen Eingabedaten auf:

Tabelle 28: Daten des Antragsberechtigten (Organisation)

Inhalt	Kurzbeschreibung
Organisation:	Name der Organisation
Langname Organisation	Langname der Organisation
Organisationseinheit	Organisationseinheit der zugelassenen Organisation
gematik ID	ID der Organisation (wird durch die gematik vergeben und als eindeutige Mandanten-ID verwendet)
Straße	Straße der Organisation
Hausnummer	Hausnummer der Organisation
Ort	Ort der Organisation
Postleitzahl	Postleitzahl der Organisation
Bundesland	Bundesland der Organisation
Land	Land der Organisation
E-Mail	E-Mailadresse zur Übermittlung von Statusinformationen
Produkttypen	zugelassene Produkttypen (Mehrfachauswahl möglich)
Zulassungsdatum	Datum der Zulassung durch die gematik
Domains	Liste zugewiesener Domainnamen (nur für Zertifikate vom Produkttyp VSDM, VSDM Intermediär, Verzeichnisdienst, KOM-LE Fachdienst, Konfigurationsdienst oder VPN-Zugangsdienst) (Mehrfachangaben möglich)
CA	Liste zugewiesener spezifischer CA-Einsatzbereiche gemäß gemSpec_PKI#Tab_PKI_213 (nur bei TSPs/TSL-Dienst für gematik Root CA) (Mehrfachauswahl möglich)
Zulassungsstatus	Zulassungsstatus (aktiviert / gesperrt)

Tabelle 29: Daten Berechtigter Zertifikatsantragsteller

Inhalt	Kurzbeschreibung
Vorname	Vorname des Mitarbeiters
Name	Name des Mitarbeiters
Organisation	Name der Organisation, falls abweichend vom Antragsberechtigten
Benutzerkennung	Benutzerkennung des Mitarbeiters zur Anmeldung an der Applikation (Sub-CA-)Zertifikatsbeantragung / Zertifikatssperrung
Straße	Straße des Mitarbeiters
Hausnummer	Hausnummer des Mitarbeiters
Ort	Ort des Mitarbeiters
Postleitzahl	Postleitzahl des Mitarbeiters
Telefonnummer	Telefonnummer des Mitarbeiters
E-Mail-Adresse	E-Mail-Adresse des Mitarbeiters
Berechtigungen	Berechtigungen zum • Erstellen und/oder • Sperren von Komponenten-Zertifikaten
Produkttypen	Berechtigte Produkttypen (Mehrfachauswahl möglich)
Benutzerstatus	Zulassungsstatus des Benutzers (aktiviert / deaktiviert)

Tabelle 30: Daten Anschreiben

Inhalt	Kurzbeschreibung
Vorname	Vorname des Mitarbeiters
Name	Name des Mitarbeiters
Straße	Straße des Mitarbeiters
Hausnummer	Hausnummer des Mitarbeiters
Ort	Ort des Mitarbeiters
Postleitzahl	Postleitzahl des Mitarbeiters
Datum	Aktuelles Datum

Benutzerkennung	Benutzerkennung des Mitarbeiters zur Anmeldung an der Applikation Zertifikatsbeantragung / Zertifikatssperrung (gematik Root CA und/oder Komponenten PKI)
RSA Token	Informationen zu der RSA Token

5 Zertifikatsausstellung (X.509/CV)

Zur Zertifikatsbeantragung für X.509 und CV-Zertifikate wird den Antragsberechtigten eine Web-Anwendung "Zertifikatsausstellung Komponenten-PKI" (kurz "ZA") über das TMS bereitgestellt. Darüber hinaus können Antragsberechtigte über die Anwendung ZA "Berechtigungs-X.509-Zertifikate" beziehen (im Folgenden als CryptIDs bezeichnet), um X.509- und CV-Zertifikate über die Schnittstellen CMP und SOAP beantragen zu können.

Es wird zu diesem Zwecke eine Infrastruktur-CA bereitgestellt, welche die Berechtigungs-Zertifikate für die Ausstellung über die Schnittstellen CMP/SOAP, sowie der CMP-Kommunikation zwischen TMS und CA-Systemen, ausstellt.

Die Anwendung ZA stellt die folgenden Use Cases zur Verfügung:

Tabelle 31: Use Cases Zertifikatsausstellung Komponenten PKI

Use Case	Beschreibung
UC-ZA-001	Zertifikate über Webanwendung beziehen
UC-ZA-002	Crypt-ID ausstellen
UC-ZA-003	Zertifikate über CMP/SOAP beziehen

5.1 Rollen und Berechtigungen

Die nachfolgende Tabelle stellt die notwendigen Benutzer-Rollen und Berechtigungen für die Zertifikatsausstellung dar:

Tabelle 32: Rollen und Berechtigungen Zertifikatsausstellung

Rolle	Kürzel	Besitzer	Berechtigung
Berechtigter Zertifikats-antragsteller	BZA	Hersteller, Anbieter, TSP oder TSL-Dienst	Berechtigter Zertifikatsantragsteller von X.509 und CV-Komponentenzertifikaten bzw. OCSP/CRL-Signerzertifikate sowie Berechtigungszertifikaten für die Ausstellung über die Schnittstellen CMP/SOAP.

5.2 Anwendung Zertifikatsausstellung (I_Cert_Provisioning_Registration / P_CVC_Provisioning)

Die Ausführungen in dem nachfolgendem Anwendungsfall schildern die Aktion Ausstellung von X.509/CV-Komponentenzertifikat und OCSP/CRL-Signer-Zertifikate für berechtigte Hersteller, Anbieter und TSP-X.509 nonQES (Antragsberechtigte) über die Webanwendung.

Tabelle 33 stellt die URIs der Webanwendung Zertifikatsausstellung für die Produktionsumgebung sowie für die Referenz- und Testumgebung dar. Die URI für die Referenz- und Testumgebung ist identisch.

Tabelle 33: URIs der Anwendung Zertifikatsausstellung

Umgebung	URI
PU	https://www.tms.ti-dienste.de/zas
RU/TU	https://www-testref.tms.ti-dienste.de/zas

5.2.1 Webanwendung Zertifikatsausstellung

Tabelle 34: UC-ZA-001

Nummer: **Name:**	UC-ZA-001 Zertifikate über Webanwendung beziehen
Kurzbeschreibung	Dieser Anwendungsfall beschreibt, wie Antragsberechtigte über die Web-Anwendung "Zertifikatsausstellung" Zertifikate beantragen können.
Auslösender Akteur	BZA
Vorbedingungen	Ein Antragsberechtigter und zugehöriger Berechtigter Zertifikatsantragsteller wurden im Zulassungsmanagement angelegt. Der Zulassungsstatus des Berechtigten Zertifikatsantragstellers sowie der zugehörigen Organisation ist aktiviert. Der Berechtigte Zertifikatsantragsteller besitzt die Berechtigung zum Beantragen von Zertifikaten und ein AD-Benutzerkonto mit Status „aktiv“. Dem Berechtigten Zertifikatsantragsteller wurde ein RSA-Token zur Verfügung gestellt. Der Akteur ist in der Applikation angemeldet und autorisiert.
Eingangsdaten	Abhängig vom Zertifikatstyp variieren die Zertifikatsantrags-Daten des PKCS#10-Requests gemäß Tabelle 46: Zertifikatsantragdaten.
Ergebnisse	Es wurde ein X.509 oder CV-Zertifikat erstellt und dem Berechtigten Zertifikatsantragsteller bereitgestellt. Das Zertifikat ist dem Antragsberechtigten bzw. der Organisation eindeutig zugeordnet.
Anmerkungen	Abhängig vom zugeordneten Produkttyp gemäß Tabelle 45

	kann der Berechtigte Zertifikatsantragsteller verschiedene Zertifikatstypen beantragen. Ein Berechtigter Zertifikatsantragsteller kann nur Zertifikate für die ihm zugehörige Organisation beantragen. Berechtigte, denen der Produkttyp Konnektor zugeordnet wurde und die die Berechtigung "Erstellen" besitzen, können Zertifikate vom Zertifikatstyp Netzkonnektor (C.NK.VPN), Anwendungskonnektor (C.AK.AUT) und Signatur-anwendungskomponente (C.SAK.AUT) beziehen. Für Berechtigte, denen der Produkttyp VSDM, Intermediär VSDM, Verzeichnisdienst, KOM-LE Fachdienst, Konfigurationsdienst oder VPN-Zugangsdienst zugeordnet wurde und die die Berechtigung "Erstellen" besitzen, müssen die bei der Zertifikatsantragsstellung angegebenen Domainnamen gegen die von der gematik mitgeteilte Liste zugewiesener Domainnamen, geprüft werden.

Tabelle 35: Prozessschritte UC-ZA-001

Nr.	Akteur	Prozessschritt
1.	BZA	Der Akteur ruft die Funktion "Zertifikat ausstellen" auf.
2.	BZA	Der Akteur wählt aus, ob er ein einzelnes oder mehrere Zertifikate beantragen möchte. Fallunterscheidung: • Der Akteur wählt die Zertifikatsausstellung eines einzelnen Zertifikats aus: Die Applikation stellt eine Eingabemaske mit allen Eingabefeldern für die notwendigen Zertifikatantrags-Daten zur Verfügung. Der Akteur gibt die erforderlichen Daten ein und lädt den Zertifikatsantrag (PKCS#10) hoch. • Der Akteur wählt die Zertifikatsausstellung mehrerer Zertifikate aus: Die Applikation stellt eine Eingabemaske mit allen Eingabefeldern für die notwendigen Zertifikatantrags-Daten zur Verfügung. Der Akteur lädt pro Zertifikat einen Zertifikatsantrag (PKCS#10) hoch.
3.	Applikation	Die Applikation prüft die Eingaben auf Vollständigkeit und Plausibilität (die Prüfungen können abhängig vom Zertifikatstyp variieren). Fehlerfälle (Zertifikatsbeantragung über Eingabemaske): • Unvollständige oder fehlende Daten: Die

		Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren. • Signatur-Prüfung des PKCS#10-Requests schlägt fehl: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren. • ICCSN für Produkttyp gSMC-KT im Feld commonName ist nicht eindeutig: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren. • ICCSN für Produkttyp gSMC-K (gilt nur für den gleichen Zertifikatstyp, in unterschiedlichen Zertifikatstypen kann die gleiche ICCSN verwendet werden) im Feld commonName ist nicht eindeutig: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren. • Für ein Zertifikat vom Produkttyp VSDM, VSDM Intermediär, Verzeichnisdienst, KOM-LE Fachdienst Konfigurationsdienst oder VPN-Zugangsdienst wurde im Feld commonName ein nicht explizit erlaubter Host- und Domänenname verwendet: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren.
4.	TMS	Das TMS erzeugt einen CMP-Zertifikatsrequest und ergänzt diesen um die für die Zertifikatserstellung notwendigen Daten des Antragsberechtigten.
5.	TMS	Das TMS signiert den Zertifikatsrequest und sendet diesen an die entsprechende Komponenten-PKI-CA.
6.	Komponenten-PKI	Die Komponenten-PKI nimmt den Zertifikatsrequest entgegen und erzeugt das Zertifikat. Die Antragsdaten und

		das Zertifikat werden in der CA-Datenbank gespeichert.
7.	Komponenten-PKI	Die Komponenten-PKI erzeugt eine interne CMP-Response und sendet diese an das TMS.
8.	TMS	Das TMS nimmt die CMP-Response der Komponenten-PKI-CA entgegen und protokolliert den Vorgang.
9.	TMS	Die Applikation meldet das erfolgreiche Erstellen des Zertifikats und stellt dem Antragsteller das Zertifikat zum Download bereit.
10.	BZA	Der Akteur lädt das Zertifikat herunter.

5.2.2 Prozessdarstellung Zertifikatsausstellung Webportal

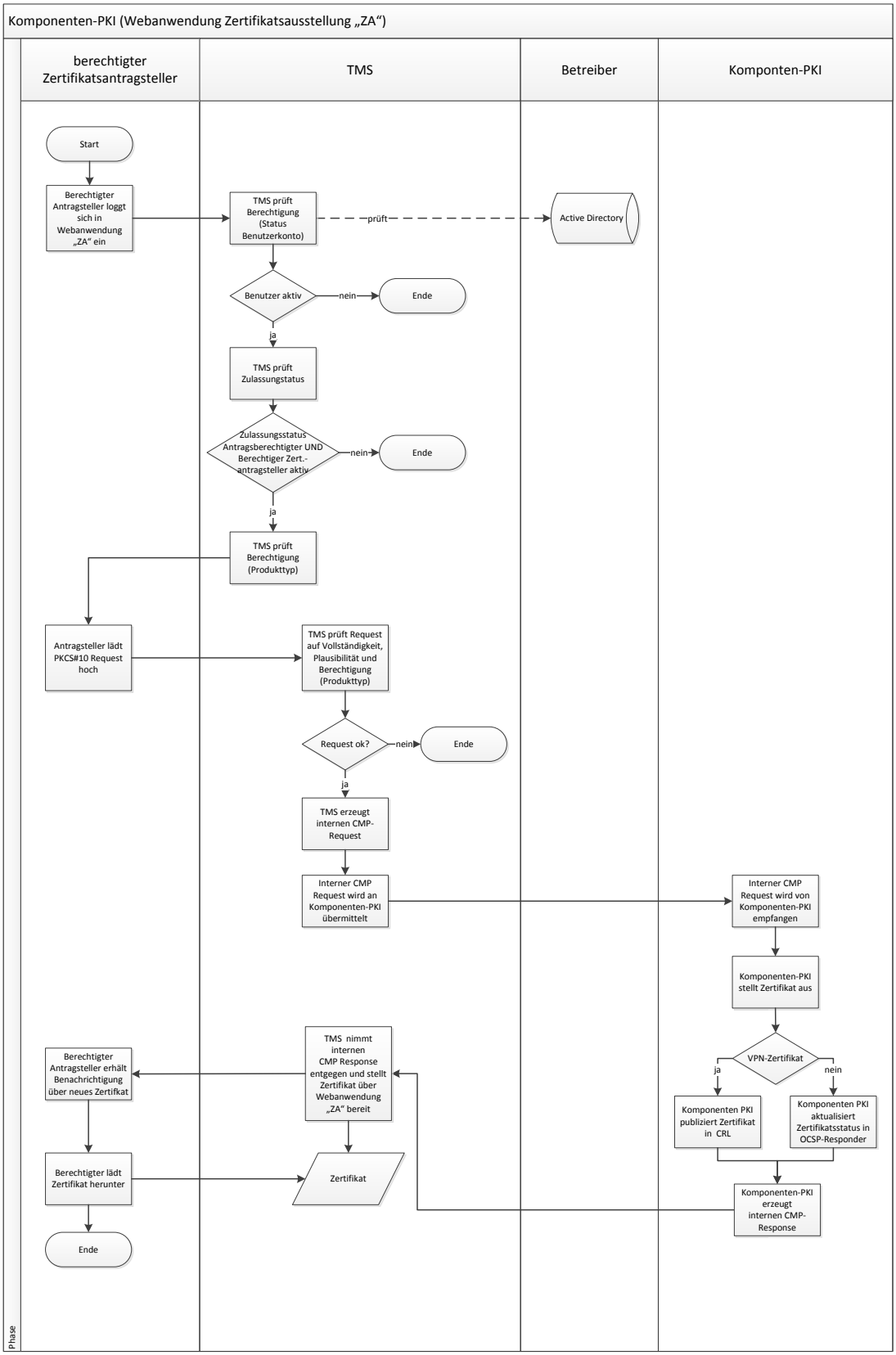

Abbildung 6: Zertifikatsausstellung Komponenten-PKI

5.2.3 CryptID ausstellen

Die Ausführungen in dem nachfolgenden Anwendungsfall schildern die Aktionen zur Ausstellung von Berechtigungs-X.509-Zertifikaten (CryptIDs) durch die Infrastruktur-CA für berechtigter Hersteller, Anbieter und TSP-X.509 nonQES (Antragsberechtigte) zur Antragstellung von X.509/CV-Zertifikate über die Schnittstellen CMP und SOAP

Tabelle 36: UC-ZA-002

Nummer: **Name:**	UC-ZA-002 Ausstellung von CryptIDs
Kurzbeschreibung	Dieser Use Case beschreibt, wie Berechtigte Zertifikatsantragsteller X.509-Zertifikate der Infrastruktur (CryptIDs) über die TMS Anwendung "Zertifikatsausstellung Komponenten-PKI (ZA)" beantragen und erhalten, um Zertifikatsanträge für Komponentenzertifikate über SOAP/CMP an die Komponenten-PKI stellen zu können.
Auslösender Akteur	BZA
Vorbedingungen	Der Berechtigte Zertifikatsantragssteller hat ein Schlüsselpaar gemäß [gemSpec_Krypt] sowie einen daraus abgeleiteten PKCS#10 Request zur Beantragung einer CryptID generiert. Der private Schlüssel muss in einer sicheren Umgebung erstellt und gespeichert werden. Ein Berechtigter Zertifikatsantragsteller wurde in der Web-Anwendung Zulassungsmanagement erzeugt. Der Zulassungsstatus des Berechtigten Zertifikatsantragstellers sowie der zugehörigen Organisation ist aktiviert. Der Berechtigte Zertifikatsantragsteller besitzt die Berechtigung zum Beantragen und/oder Sperren von Zertifikaten. Dem Berechtigten Zertifikatsantragsteller wurde ein RSA-Token zur Verfügung gestellt. Der Akteur ist in der Applikation „ZA“ angemeldet und autorisiert.
Eingangsdaten	PKCS#10 Zertifikatsantrags-Daten gemäß Tabelle 46: Zertifikatsantragdaten.
Ergebnisse	Es wurde ein Berechtigungs-X.509-Zertifikat (CryptID) durch die Infrastruktur-CA erstellt und dem Berechtigten Zertifikatsantragsteller bereitgestellt. Zudem wurde dem Berechtigten Zertifikatsantragsteller das zugehörige CA-Zertifikat zur Verfügung gestellt, das zur Validierung der CMP- bzw. SOAP-Responses benötigt wird.

Anmerkungen	Das ausgestellte X.509-Zertifikat zur Antragsstellung über CMP/SOAP ist dem Antragsberechtigten bzw. der Organisation eindeutig zugeordnet.

Tabelle 37: Prozessschritte UC-ZA-002

Nr.	Akteur	Prozessschritt
1.	BZA	Der Akteur ruft die Funktion "CryptID ausstellen" auf.
2.	BZA	Die Applikation stellt eine Eingabemaske mit allen Eingabefeldern für die notwendigen Zertifikatantrags-Daten zur Verfügung. Der Akteur gibt die erforderlichen Daten ein und lädt den Zertifikatsantrag (PKCS#10) hoch.
3.	Applikation	Die Applikation prüft die Eingaben auf Vollständigkeit und Plausibilität. Fehlerfälle (Zertifikatsbeantragung über Eingabemaske): • Unvollständige oder fehlende Daten: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren. • Signatur-Prüfung des PKCS#10-Requests schlägt fehl: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung. Der Akteur kann diese korrigieren, an.
4.	TMS	Das TMS erzeugt einen internen CMP-Zertifikatsrequest und ergänzt den CMP-Zertifikatsrequest um die notwendigen Zertifikatsdaten des Antragsberechtigten.
5.	TMS	Das TMS signiert den Zertifikatsrequest und sendet diesen an die Infrastruktur-CA.
6.	Infrastruktur-CA	Die Infrastruktur-CA nimmt den Zertifikatsrequest entgegen und erzeugt das Zertifikat. Die Antragsdaten und das Zertifikat bzw. die Zertifikate werden in der CA-Datenbank gespeichert.
7.	Infrastruktur-CA	Die Infrastruktur-CA erzeugt eine interne CMP-Response und sendet diese an das TMS.
8.	TMS	Das TMS nimmt die CMP-Response der Infrastruktur- CA entgegen und protokolliert den Vorgang.
9.	TMS	Das TMS speichert zum erzeugten Zertifikat den

		Zusammenhang mit dem zugehörigen Berechtigten Zertifikatsantragssteller in der TMS Datenbank.
10.	TMS	Die Applikation meldet das erfolgreiche Erstellen des Zertifikats und stellt dem Antragsteller das Zertifikat zum Download bereit.
11.	BZA	Der Akteur lädt das Zertifikat herunter.

5.2.4 Prozessdarstellung CryptID ausstellen

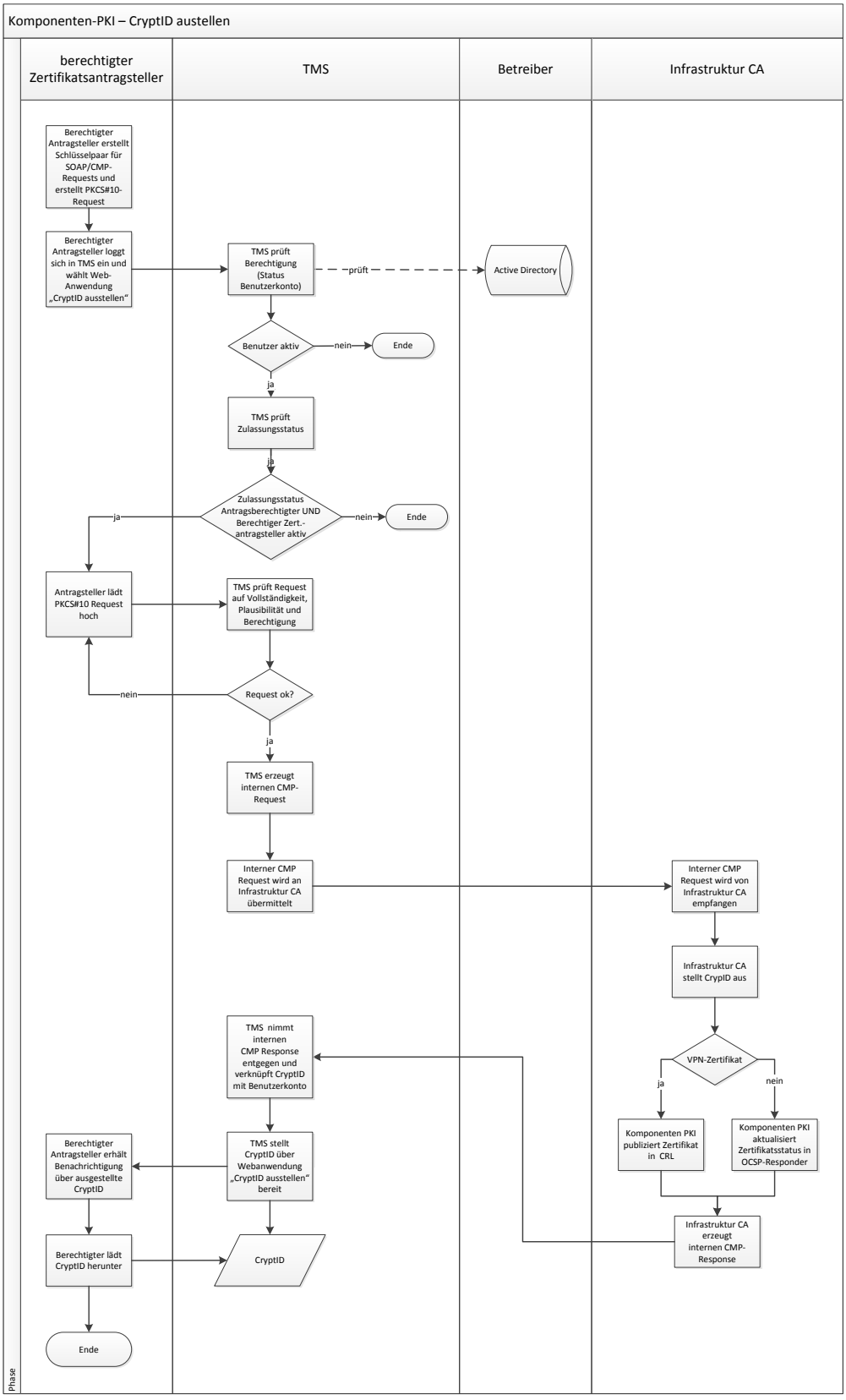

Abbildung 7: CryptID ausstellen

5.2.5 Zertifikate über CMP/SOAP ausstellen

Die Komponenten-PKI bietet SOAP/CMP over HTTP(S) an, um Zertifikate automatisiert zu beantragen. Hierbei ist die Komponenten-PKI maximal für die Anzahl von 100 Zertifikatsbeantragungen pro SOAP/CMP-Request ausgelegt. Der zuständige Webservice verlangt in der Kommunikation die HTTP-Basic-Authentication. Somit muss der entsprechende Header gesetzt werden. Jeder SOAP/CMP-Request wird zudem mit einer CryptID signiert. Entsprechend wird jede SOAP/CMP-Response (bis auf ERROR ...,) mit einem SOAP/CMP-Responder signiert. Bei CMP wird das zugehörige Zertifikat in der Response-Nachricht mitgesendet. Für die Signaturprüfung der SOAP-Response wird das Signer-Zertifikat auf folgenden Downloadpunkten bereitgestellt.

Tabelle 38: Downloadpunkte der SOAP-Responder-Zertifikate

Umgebung	URI
PU	https://download.tsl.ti-dienste.de
RU/TU	https://download-testref.tsl.ti-dienste.de

Tabelle 39 stellt die URIs der Zertifikatsausstellung über SOAP für die Produktionsumgebung sowie für die Referenz- und Testumgebung dar. Die URI für die Referenz- und Testumgebung sind identisch. Entsprechend der Umgebung muss die URI in der beigefügten WSDL-Datei „CertificateManagementService.wsdl“ als „location“ im Element „soap:address“ gesetzt werden.

Tabelle 39: URIs der Zertifikatsausstellung über SOAP

Umgebung	URI
PU	https://www.tms.ti-dienste.de/cxf/certificateManagementService
RU/TU	https://www-testref.tms.ti-dienste.de/cxf/certificateManagementService

Tabelle 40 stellt die URIs der Zertifikatsausstellung über CMP für die Produktionsumgebung sowie für die Referenz- und Testumgebung dar. Die URI für die Referenz- und Testumgebung sind identisch.

Tabelle 40: URIs der Zertifikatsausstellung über CMP

Umgebung	URI
PU	https://www.tms.ti-dienste.de/cxf/cmp
RU/TU	https://www-testref.tms.ti-dienste.de/cxf/cmp

Die angegebenen URIs stellen den CMP-Endpoint im TMS dar. Dort kann über den URI-Query „?_wadl“ (http://<host>/cxf/cmp?_wadl) die Beschreibung der Webanwendung mit der GET-Methode abgerufen werden, welche die zwei Pfade für das Ausstellen und Revozieren von Zertifikaten beschreibt. Die eigentliche Datenübermittlung muss über POST erfolgen. Der Inhalt der WADL ist in folgendem Listing dargestellt:

```
<application>
    <grammars />
    <resources base="http://<host>/cxf/cmp">
```

```
        <resource path="/certificates">
            <method name="POST">
                <request>
                    <representation mediaType="application/pkixcmp" />
                    <representation mediaType="application/x-pkixcmp" />
                </request>
                <response>
                    <representation mediaType="application/pkixcmp" />
                </response>
            </method>
        </resource>
        <resource path="/revocations">
            <method name="POST">
                <request>
                    <representation mediaType="application/pkixcmp" />
                    <representation mediaType="application/x-pkixcmp" />
                </request>
                <response>
                    <representation mediaType="application/pkixcmp" />
                </response>
            </method>
        </resource>
    </resources>
</application>
```

Listing 1: Beschreibung der Webanwendung zur CMP-Schnittstelle

Die Zertifikatsausstellung erfolgt somit über den URI-Path „/cxf/cmp/certificates“ und die Zertifikatssperrung über „/cxf/cmp/revocations“. Daraus ergeben sich folgende URIs für die entsprechenden Umgebungen:

Tabelle 41: Konkrete URIs der Zertifikatsausstellung über CMP

Umgebung	URI
PU	https://www.tms.ti-dienste.de/cxf/cmp/certificates
RU/TU	https://www-testref.tms.ti-dienste.de/cxf/cmp/certificates

Tabelle 42: Konkrete URIs der Zertifikatssperrung über CMP

Umgebung	URI
PU	https://www.tms.ti-dienste.de/cxf/cmp/revocations
RU/TU	https://www-testref.tms.ti-dienste.de/cxf/cmp/revocations

Die Ausführungen in dem nachfolgenden Anwendungsfall schildern die Aktionen zur Ausstellung von X.509/CV-Komponentenzertifikat und OCSP/CRL-Signer-Zertifikate für berechtigter Hersteller, Anbieter und TSP-X.509 nonQES (Antragsberechtigte) über die Schnittstellen CMP und SOAP.

Tabelle 43: UC-ZA-003

Nummer: **Name:**	UC-ZA-003 Zertifikate über SOAP/CMP ausstellen
Kurzbeschreibung	Dieser Anwendungsfall beschreibt, wie Berechtigte Zertifikatsantragsteller unter Verwendung des Protokolls CMP bzw. SOAP Zertifikate beantragen können.

Auslösender Akteur	Berechtigter Zertifikatsantragsteller bzw. technische Komponente des Berechtigten Zertifikatsantragstellers
Vorbedingungen	Der Berechtigte Zertifikatsantragsteller wurde in der Anwendung "Zulassungsmanagement" erzeugt. Der Zulassungsstatus des Berechtigten Zertifikatsantragstellers sowie der zugehörigen Organisation ist aktiviert. Der Berechtigte Zertifikatsantragsteller besitzt die Berechtigung zum Beantragen von Zertifikaten und den Status aktiviert. Der Berechtigte Zertifikatsantragsteller hat über die Anwendung „CryptID ausstellen“ eine CryptID bezogen.
Eingangsdaten	Abhängig vom Zertifikatstyp variieren die Zertifikatsantrags-Daten des PKCS#10-Requests gemäß Tabelle 46: Zertifikatsantragdaten.
Ergebnisse	Es wurde(n) ein oder mehrere Zertifikate erstellt und an den Berechtigten Zertifikatsantragsteller in Form einer SOAP/CMP-Response ausgegeben. Das Zertifikat ist dem Antragsberechtigten bzw. der Organisation eindeutig zugeordnet.
Anmerkungen	Abhängig vom zugeordneten Produkttyp gemäß Tabelle 45 kann der Berechtigte Zertifikatsantragsteller verschiedene Zertifikatstypen beantragen. Ein Berechtigter Zertifikatsantragsteller kann nur Zertifikate für die ihm zugehörige Organisation beantragen. Berechtigte, denen der Produkttyp Konnektor zugeordnet wurde und die die Berechtigung "Erstellen" besitzen, können Zertifikate vom Zertifikatstyp Netzkonnektor (C.NK.VPN), Anwendungskonnektor (C.AK.AUT) und Signatur-anwendungskomponente (C.SAK.AUT) beziehen. Für Berechtigte, denen der Produkttyp VSDM, Intermediär VSDM, Konfigurationsdienst, Verzeichnisdienst, KOM-LE Fachdienst oder VPN-Zugangsdienst zugeordnet wurde und die die Berechtigung "Erstellen" besitzen, müssen die bei der Zertifikatsantragsstellung angegebenen Domainnamen gegen die von der gematik mitgeteilte Liste zugewiesener Domainnamen, geprüft werden.

Tabelle 44: Prozessschritte UC-ZA-003

Nr.	Akteur	Prozessschritt

1.	BZA	Der Akteur sendet einen HTTP-authentisierten CMP/SOAP-Request an das TMS.
2.	TMS	Das TMS prüft die HTTP-Authentisierung + SOAP/CMP-Signatur • Status der Zulassung Berechtigter Zertifikatsantragsteller und Organisation • Status des Benutzers (AD) • Berechtigung zur Beantragung des Zertifikatstyps Fallunterscheidung: a) Keine Berechtigung, weil • Zulassungsstatus deaktiviert • Benutzerstatus deaktiviert (AD) • Berechtigung zur Beantragung des Zertifikatstyps fehlt b) Berechtigung zur Zertifikatsbeantragung erteilt • Zulassungsstatus aktiv • Benutzerkonto (AD) aktiv • Berechtigung für beantragtes Zertifikat vorhanden →Weiter mit Schritt 3
3.	TMS	Die Applikation prüft die Eingangsdaten auf Vollständigkeit und Plausibilität (die Prüfungen können abhängig vom Zertifikatstyp variieren). Überprüfung der Zertifikatstypberechtigung anhand der mitgelieferten Produkttyp-OID im CMP-Request. Fehlerfälle (Zertifikatsbeantragung CMP/SOAP): • Unvollständige oder fehlende Daten: SOAP/CMP-Response mit Fehlermeldung. • Signatur-Prüfung des PKCS#10-Requests schlägt fehl: CMP-Response mit Fehlermeldung • ICCSN für Produkttyp Kartenterminal im Feld commonName ist nicht eindeutig: SOAP/CMP-Response mit Fehlermeldung. • ICCSN für Produkttyp Konnektor im Feld commonName (gilt nur für den gleichen

		Zertifikatstyp, in unterschiedlichen Zertifikatstypen kann die gleiche ICCSN verwendet werden) ist nicht eindeutig: SOAP/CMP-Response mit Fehlermeldung. • Für ein Zertifikat vom Produkttyp VSDM, VSDM Intermediär, Konfigurationsdienst, Verzeichnisdienst, KOM-LE Fachdienst oder VPN-Zugangsdienst wurde im Feld commonName ein nicht erlaubter Host- und Domänenname verwendet: SOAP/CMP-Response mit Fehlermeldung. • Kein Berechtigung für Produkttyp-OID: CMP-Response mit Fehlermeldung.
4.	TMS	Das TMS erzeugt einen CMP-Zertifikatsrequest und ergänzt den CMP-Zertifikatsrequest um die notwendigen Zertifikatsdaten des Antragsberechtigten.
5.	TMS	Das TMS signiert den Zertifikatsrequest und sendet diesen an entsprechende Komponenten-PKI.
6.	Komponenten-PKI	Die Komponenten-PKI nimmt den Zertifikatsrequest entgegen und erzeugt das Zertifikat. Die Antragsdaten und das Zertifikat werden in der CA-Datenbank gespeichert.
7.	Komponenten-PKI	Die Komponenten-PKI erzeugt eine interne CMP-Response und sendet diese an das TMS.
8.	TMS	Das TMS nimmt die interne CMP-Response der Komponenten-PKI entgegen und protokolliert den Vorgang.
9.	TMS	Das TMS konvertiert die interne CMP-Response der Komponenten-PKI-CA und sendet das Zertifikat SOAP/CMP-Response an den berechtigten Antragsteller (technische Komponente).
10.	TMS	Die Applikation meldet das erfolgreiche Erstellen des Zertifikats.
11.	BZA	Der Akteur (technische Komponente) empfängt die SOAP/CMP Response mit dem erstellten Zertifikat.

5.2.6 Prozessdarstellung Zertifikatsausstellung über SOAP/CMP

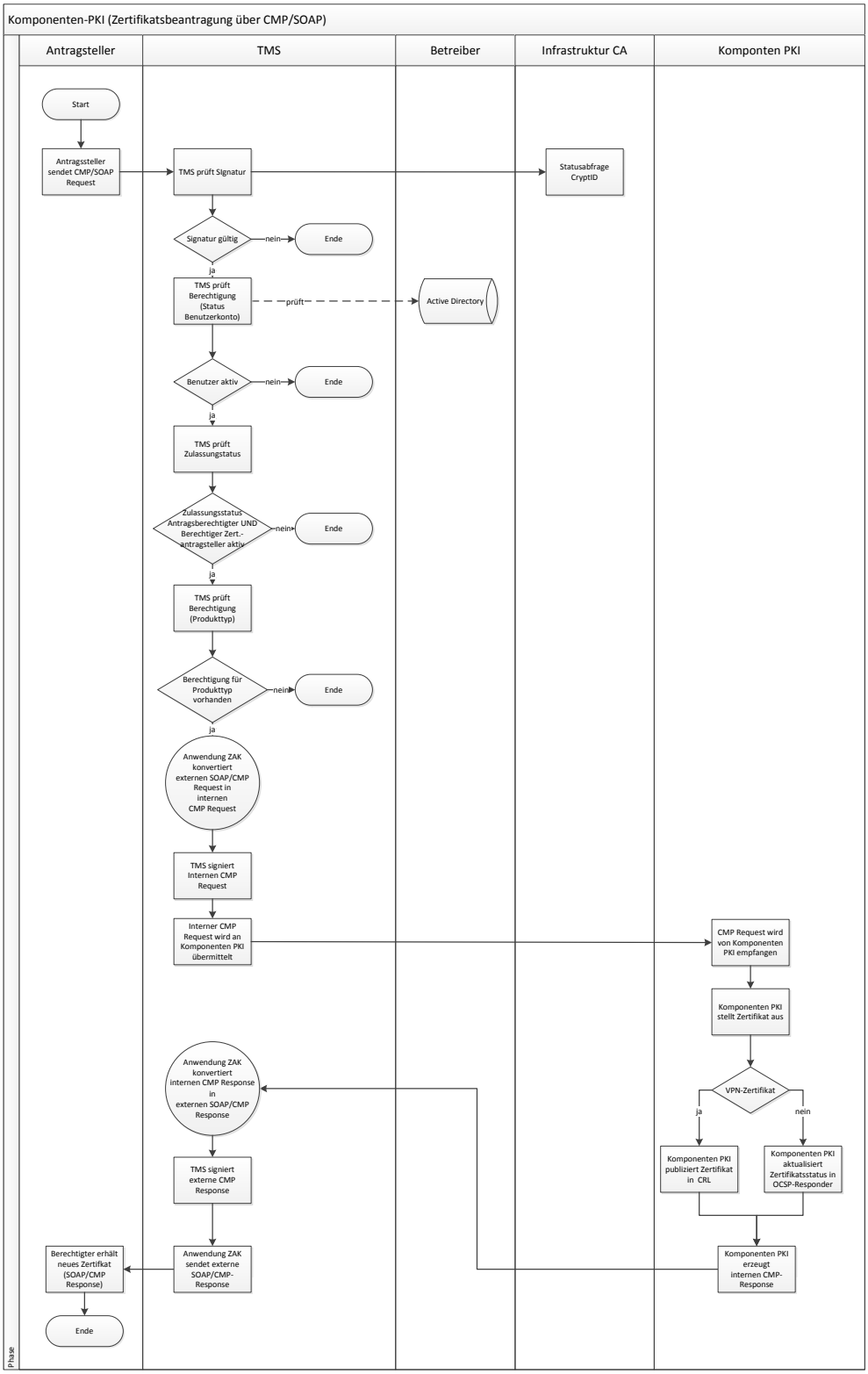

Abbildung 8: Zertifikatsausstellung über SOAP/CMP

5.3 Artefakte

5.3.1 Zertifikatstypen

Nachfolgende Tabelle stellt die Beziehung zwischen Produkttyp, Zertifikatsprofil und Zertifikatstyp dar. Unter Verwendung der Weboberfläche wählt der Berechtigte Zertifikatsantragsteller den Zertifikatstyp aus.

Tabelle 45: Zusammenhang zwischen Produkttyp, Zertifikatsprofil und Zertifikatstyp

Produkttyp	Zertifikatsprofil	Zertifikatstyp (Webanwendung)	Zertifikatstyp (CMP/SOAP)	ProfessionOID
VSDM	C.FD.TLS-S	Versichertenstammdatendienst	C.FD.TLS-S.VSDD	1.2.276.0.76.4.97
		Card Management System	C.FD.TLS-S.CMS	1.2.276.0.76.4.100
		Update Flag Service	C.FD.TLS-S.UFS	1.2.276.0.76.4.101
Intermediär VSDM	C.FD.TLS-S	Intermediär VSDM Server	C.FD.TLS-S.INT_VSDM	1.2.276.0.76.4.159
	C.FD.TLS-C	Intermediär VSDM Client	C.FD.TLS-C.INT_VSDM	1.2.276.0.76.4.159
Konfigurationsdienst	C.ZD.TLS-S	Konfigurationsdienst	C.ZD.TLS-S.KONFIGDIENST	1.2.276.0.76.4.160
Störungsampel	C.ZD.TLS-S	Störungsampel	C.ZD.TLS-S.STAMP	1.2.276.0.76.4.184
Konnektor	C.AK.AUT	Anwendungskonnektor	C.AK.AUT	1.2.276.0.76.4.103
	C.NK.VPN	Netzkonnektor	C.NK.VPN	1.2.276.0.76.4.104
	C.SAK.AUT	Signaturanwendungskomponente	C.SAK.AUT	1.2.276.0.76.4.119
Kartenterminal	C.SMKT.AUT	Kartenterminal	C.SMKT.AUT	1.2.276.0.76.4.105

VPN-Zugangsdienst	C.VPNK.VPN	VPN-Zugangsdienst-TI	C.VPNK.VPN	1.2.276.0.76.4.161
	C.ZD.TLS-S	Registrierungsserver	C.ZD.TLS-S.REGSRV	1.2.276.0.76.4.161
	C.VPNK.VPN-SIS	VPN-Zugangsdienst-SIS	C.VPNK.VPN-SIS	1.2.276.0.76.4.166
CRL	C.GEM.CRL	CRL Signer	C.GEM.CRL	-
OCSP	C.GEM.OCSP	OCSP Signer	C.GEM.OCSP	1.2.276.0.76.4.99
Verzeichnisdienst	C.ZD.TLS-S	Verzeichnisdienst	C.ZD.TLS-S.VZD	1.2.276.0.76.4.171
KOM-LE Fachdienst	C.FD.TLS-S	KOM-LE FD Serverzertifikat	C.FD.TLS-S.KOM-LE	1.2.276.0.76.4.172
	C.FD.TLS-C	KOM-LE FD Clientzertifikat	C.FD.TLS-C.KOM-LE	1.2.276.0.76.4.172
KOM-LE Clientmodul	C.CM.TLS-CS	Clientmodul C/S	C.CM.TLS-CS.KOM-LE	1.2.276.0.76.4.174
TSL-Dienst-TI	C.ZD.TLS-S	TSLD Server	C.ZD.TLS-S.TSLD	1.2.276.0.76.4.189

Im Rahmen der Beantragung von Zertifikaten über SOAP/CMP wird der Zertifikatstyp der Spalte "Zertifikatstyp (CMP/SOAP)" verwendet.

5.3.2 Eingangsdaten für Webanwendung

Die folgende Tabelle führt die in den Use Cases zur Zertifikatserstellung notwendigen und über die Webanwendung zu erfassenden Eingabedaten auf:

Tabelle 46: Zertifikatsantragdaten

Zertifikatsantragsdaten	Kurzbeschreibung
Zertifikatstyp	Auswahl des zu erstellenden Zertifikatstyps (gemäß Tabelle 45 - Spalte Zertifikatstyp (Webanwendung))
PKCS#10-Request	Der Inhalt des PKCS#10-Requests variiert abhängig vom Zertifikatsformat für X.509-Zertifikate (Listing 2) und CV-Zertifikate (siehe Listing 3).

5.3.2.1 PKCS#10 Request für ein X.509 Zertifikat

Es werden folgende Daten im PKCS#10-Request benötigt, um ein X.509 Zertifikat auszustellen.

- SubjectPublicKeyInfo (siehe [RFC3279])
- Subject

```
CertificationRequest ::= SEQUENCE {
    certificationRequestInfo    CertificationRequestInfo,
    signatureAlgorithm          AlgorithmIdentifier,
                                -- sha256WithRSAEncryption (1 2 840 113549 1 1 11) mit parameter NULL
    signature                   BIT STRING
}

CertificationRequestInfo ::= SEQUENCE {
    version             INTEGER { v1(0) },
    subject             Name,
    subjectPKInfo       SubjectPublicKeyInfo,
}

SubjectPublicKeyInfo ::= SEQUENCE{
    algorithm SEQUENCE {
        algorithm   OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
    }
    subjectPublicKey RSAPublicKey SEQUENCE {
        modulus         INTEGER,
        publicExponent  INTEGER
    }
}
```

Listing 2: PKCS#10 Request für ein X.509 Zertifikat

SubjectPublicKeyInfo ist für alle Zertifikattypen gleich gestaltet. Der PKCS#10-Request wird mit dem entsprechenden privaten Schlüssel signiert.

Der Inhalt von Subject wird für die Felder

- Organisation (O)
- Organisationseinheit (OU)
- Ort (L)
- Bundesland (ST)

- Land (C)

vom TMS aus den Informationen vom Zulassungsmanagement gesetzt. Die Seriennummer (SN) im Subject wird von der CA bei der Antragsbearbeitung vergeben.

Werden diese Informationen im PKCS#10-Request angegeben, haben sie für das endgültige Zertifikat keine Bedeutung und werden vom TMS bzw. der CA ignoriert und mit den eigenen Werten überschieben.

Es muss ausschließlich der Common-Name (CN) im PKCS#10-Request gemäß Tabelle 47 für den beantragten Zertifikatstyp angegeben werden. Das Ausstellungsdatum für die Zertifikatstypen

- Netzkonnektor
- Anwendungskonnektor
- Kartenterminal
- Signaturanwendungskomponente

wird von der CA automatisiert der ICCSN im Common-Name angehängt. Das Datum darf NICHT im PKCS#10-Request im Common-Name mit angegeben, da der Zertifikatsrequest ansonsten als ungültig zurückgewiesen wird.

Für Zertifikate aus der Test-PKI werden die Erweiterungen „TEST-ONLY“ und „NOT-VALID“ automatisiert beim Common-Name bzw. den Organisationsnamen angehängt.

Tabelle 47: Inhalt von Subject in Abhängigkeit vom Zertifikatstyp

Zertifikatstyp	Angabe des CN		
	ICCSN	FQDN	vollständiger Name
Intermediär VSDM Server	-	X	-
Intermediär VSDM Client		X	
Versichertenstammdatendienst	-	X	-
Card Management System	-	X	-
Update Flag Service	-	X	-
CRL-Signer	-	-	X
OCSP-Signer	-	-	X
Netzkonnektor	X	-	-
Anwendungskonnektor	X	-	-
Signaturanwendungskomponente	X	-	-
Kartenterminal	X	-	-

VPN-Zugangsdienst-TI	-	X	-
VPN-Zugangsdienst-SIS	-	X	-
Registrierungsserver (als Teil vom VPN-Zugangsdienst)		X	
Konfigurationsdienst	-	X	-
Störungsampel	-	X	-
Verzeichnisdienst	-	X	-
KOM-LE Fachdienst Server	-	X	-
KOM-LE Fachdienst Client	-	X	-
TSL-Dienst-TI	-	X	-
Clientmodul C/S	nicht festgelegt		

5.3.2.2 PKCS#10 Request für ein CV-Zertifikat

Es werden folgende Daten im PKCS#10-Request benötigt, um ein CV-Zertifikat auszustellen (vergleiche [gemSpec_PKI# Kapitel 6.6] und [gemSpec_PKI# Kapitel 6.7]):

- Öffentlicher Schlüssel
- CHR (12 Byte) bestehend aus
  - Key Identifier (2 Bytes).
  - ICCSN (10 Bytes)

```
CertificationRequest ::= SEQUENCE {
    certificationRequestInfo    CertificationRequestInfo,
    signatureAlgorithm          AlgorithmIdentifier,
                        -- ecdsa-with-SHA256 (1 2 840 10045 4 3 2) für G2 EC Key 256 Bit
    signature                   BIT STRING
}

CertificationRequestInfo ::= SEQUENCE {
    version             INTEGER { v1(0) },
    subject             Name,
    subjectPKInfo       SubjectPublicKeyInfo    -- G2:CvECPublicKeyInfo
}

Name ::= SEQUENCE { -- CHR Daten werden in CommonName und SerialNumber Object gekapselt
    RelativeDistinguishedName SET {
        AttributeTypeAndValue SEQUENCE {
            type    OBJECT IDENTIFIER id-at-commonName (2 5 4 3),
            value   UTF8String -- Hexadezimale Darstellung des ICCSN
        }
    }
    RelativeDistinguishedName SET {
        AttributeTypeAndValue SEQUENCE {
            type    OBJECT IDENTIFIER id-at-serialNumber (2 5 4 5),
            value   PrintableString prefix  -- numerischer wert des Prefix
        }
    }
}

CvECPublicKeyInfo ::= SEQUENCE{
    algorithm SEQUENCE {
        algorithm   OBJECT IDENTIFIER   -- authS_gemSpec-COS-G2_ecc-with-sha256 (1 3 36 3 5 3 1)
    }
```

```
    subjectPublicKey ECPoint BIT STRING {
        uncompressed ECPoint            -- brainPoolP256r1: '04' || Q_x (32 Byte) || Q_y (32 Byte)
    }
}
```

Listing 3: PKCS#10 Request für ein CV-Zertifikat

5.3.3 CMP-Request und CMP-Response (X.509)

Die CMP-Schnittstelle zum Beantragen von X.509-Komponenten-Zertifikaten wird über HTTP(S) angeboten.

Während die CMP-Requests mit einem CMP-Requestor (Crypt-ID) signiert werden, werden die CMP-Responses – abgesehen von CMP-Error-Responses - mit einem CMP-Responder signiert.

Gemäß [RFC4210] sind die CMP-Requests und –Responses DER-kodiert im Body des POST-Befehls zu übermitteln. Der Content-Type für die Requests und Responses ist application/pkixcmp.

Das nachfolgende Listing stellt die Struktur der CMP-Nachricht zur Beantragung von X.509-Komponenten-Zertifikaten über die CMP-Schnittstelle gemäß [RFC4211] dar:

```
PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection [0] EXPLICIT PKIProtection,                  -- Signature-Variante mit
                                                               SHA1/224/256/384/512withRSA/DSA/ECDSA/RSAandMGF1
                                                               ist erforderlich (abhängig von der Konfiguration,
                                                               aktuell: SHA256withRSA)
}

PKIHeader ::= SEQUENCE {
    pvno              INTEGER cmp2000(2),                   -- fix
    sender            GeneralName,                          -- directoryName des CMP-Requesters
    recipient         GeneralName,                          -- directoryName des CMP-Responders
    messageTime   [0] EXPLICIT GeneralizedTime,             -- Timestamp der Nachricht, erfoderlich (OPTIONAL nach RFC
4210)
    protectionAlg [1] EXPLICIT AlgorithmIdentifier,         -- Signaturalgorithmus für PKIMessage.protection
erforderlich
    transactionID [4] EXPLICIT OCTET STRING,                -- erforderlich (OPTIONAL nach RFC 4210)
    generalInfo   [8] EXPLICIT SEQUENCE SIZE (1..MAX) OF InfoTypeAndValue
                                                            -- implicitConfirm ist mit dem OID (1 3 6 1 5 5 7 4 13}
                                                               erfoderlich
                                                            -- (OPTIONAL nach RFC 4210)
}

PKIBody ::= CHOICE
{
   ...
   cr [2] EXPLICIT CertReqMessages                  -- Nur dieser Variante ist erlaubt
   ...
}

CertReqMessages ::= SEQUENCE OF CertReqMsg

CertReqMsg ::= SEQUENCE {
    certReq CertRequest,
    popo    ProofOfPossession                       -- erforderlich (OPTIONAL nach RFC 4211)
    regInfo   SEQUENCE SIZE(1..MAX) of AttributeTypeAndValue
                                                    -- hier ist utf8Pairs mit dem OID (1.3.6.1.5.5.7.5.2.1}
                                                    -- erforderlich (OPTIONAL nach RFC 4211)
                                                    -- Mindestens ein UTF8Pair über Zertifikattyp ist erforderlich.
                                                    -- zugelassene Zertifikattypen:
                                                    --Versichertenstammdatendienst: C.FD.TLS-S.VSDD
                                                    --Card Management System: cert_profile?C.FD.TLS-S.CMS
                                                    --Update Flag Service: cert_profile?C.FD.TLS-S.UFS
                                                    --Intermediär VSDM Server: cert_profile?C.FD.TLS-S.INT_VSDM
                                                    --Intermediär VSDM Client: cert_profile?C.FD.TLS-C.INT_VSDM
                                                    --Konfigurationsdienst: cert_profile?C.ZD.TLS-S.KONFIGDIENST
                                                    --Störungsampel: cert_profile?C.ZD.TLS-S.STAMP
                                                    --Anwendungskonnektor: cert_profile?C.AK.AUT
                                                    --Netzkonnektor: cert_profile?C.NK.VPN
                                                    --Signaturanwendungskomponente: cert_profile?C.SAK.AUT
                                                    --Kartenterminal: cert_profile?C.SMKT.AUT
                                                    --VPN-Zugangsdienst-TI: cert_profile?C.VPNK.VPN
                                                    --VPN-Zugangsdienst-SIS: cert_profile?C.VPNK.VPN-SIS
                                                    --CRL Signer: cert_profile?C.GEM.CRL
                                                    --OCSP Signer: cert_profile?GEM.OCSP
                                                    --Verzeichnisdienst: C.ZD.TLS-S.VZD
                                                    --KOM-LE Fachdienst Serverzertifikat: C.FD.TLS-S.KOM-LE
                                                    --KOM-LE Fachdienst Clientzertifikat: C.FD.TLS-C.KOM-LE
                                                    --KOM-LE Clientmodul: C.CM.TLS-CS.KOM-LE
                                                    --Registrierungsserve: C.ZD.TLS-S.REGSRV
```

```
                                                --TSL-Dienst-TI: C.ZD.TLS-S.TSLD
}

ProofOfPossession ::= CHOICE {
   ...
   signature         [1] POPOSigningKey,           -- Nur die Variante POPOSigningKey ist erlaubt
   ...
}

POPOSigningKey ::= SEQUENCE {
   poposkInput         [0] POPOSigningKeyInput OPTIONAL,  --wird nicht unterstützt/nicht erlaubt
   algorithmIdentifier     AlgorithmIdentifier,
   signature               BIT STRING
}

CertRequest ::= SEQUENCE {
    certReqId       INTEGER,
    certTemplate    CertTemplate,
}

CertTemplate ::= SEQUENCE {
    version         [0] EXPLICIT 2 (v3) OPTIONAL,
    subject         [5] EXPLICIT Name,                      -- Gleich wie in PKCS#10 Request (5.3.2.1)
    publicKey       [6] EXPLICIT SubjectPublicKeyInfo       -- Gleich wie in PKCS#10 Request (5.3.2.1)
}
```

Listing 4: Struktur eines CMP-Requests zur Beantragung von X.509-Komponenten-Zertifikaten

Der CMP-Responder prüft den CMP-Request. Im negativen Fall antwortet der Responder mit einer CMP-Response in der Variante "error [23] ErrorMsgContent" im PKIBody.

Andererseits antwortet der Responder mit einer Nachricht auf Basis der im nachfolgenden Listing dargestellten Struktur:

```
PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection [0] EXPLICIT PKIProtection,                  -- Signature-Variante mit
                                                               SHA1/224/256/384/512withRSA/DSA/ECDSA/RSAandMGF1
                                                               ist erforderlich (abhängig von der Konfiguration,
                                                               aktuell: SHA256withRSA)
}

PKIHeader ::= SEQUENCE {
    pvno              INTEGER cmp2000(2),                   -- fix
    sender            GeneralName,                          -- directoryName des CMP-Responders
    recipient         GeneralName,                          -- directoryName des CMP-Requesters
    messageTime   [0] EXPLICIT GeneralizedTime OPTIONAL,    -- Timestamp der Nachricht
    protectionAlg [1] EXPLICIT AlgorithmIdentifier,         -- Signaturalgorithm für PKIMessage.protection, hier
erforderlich
    transactionID [4] EXPLICIT OCTET STRING,                -- transactionID aus dem Request
    generalInfo   [8] EXPLICIT SEQUENCE SIZE (1..MAX) OF InfoTypeAndValue
                                                            -- hier implicitConfirm mit dem OID (1 3 6 1 5 5 7 4 12}
erfoderlich
}

PPKIBody ::= CHOICE
{
   ...
   cp [3] EXPLICIT CertRepMessage,                          -- Nur diese Variante ist erlaubt
   ...
}

CertRepMessage ::= SEQUENCE {
    caPubs  [1] EXPLICIT SEQUENCE SIZE (1..MAX) OF CMPCertificate OPTIONAL,
                                                            -- enthält das CA Zertifikat, falls vorhanden
    response    SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
    certReqId           INTEGER,                    -- Wert aus CMP-Request
    status              PKIStatusInfo,              -- mit dem PKIStatus accepted (0) der grantedWithMods im positiven
                                                       Fall,
                                                    -- ansonsten rejection (2)
    certifiedKeyPair    CertifiedKeyPair    OPTIONAL -- nur vorhanden, wenn das Zertifikat erfolgreich ausgestellt wurde
}

CertifiedKeyPair ::= SEQUENCE {
         certOrEncCert       CertOrEncCert,             -- wird verwendet
         privateKey      [0] EncryptedValue      OPTIONAL, -- nicht genutzt
         -- see [CRMF] for comment on encoding
         publicationInfo [1] PKIPublicationInfo  OPTIONAL  -- nicht genutzt
}

CertOrEncCert ::= CHOICE {
         certificate     [0] CMPCertificate,            -- wird verwendet
         encryptedCert   [1] EncryptedValue             -- nicht genutzt
}
```

Listing 5: Struktur der CMP-Response (X.509)

```
   0 1193: SEQUENCE {
   4  251:   SEQUENCE {
   7    1:     INTEGER 2
  10   91:     [4] {
  12   89:       SEQUENCE {
  14   10:         SET {
  16    8:           SEQUENCE {
  18    3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
  23    1:             PrintableString '4'
        :             }
        :           }
  26   30:         SET {
  28   28:           SEQUENCE {
  30    3:             OBJECT IDENTIFIER commonName (2 5 4 3)
  35   21:             UTF8String 'OLGA-777777 TEST-ONLY'
        :             }
        :           }
  58   30:         SET {
  60   28:           SEQUENCE {
  62    3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
  67   21:             UTF8String 'OLGA-777777 NOT-VALID'
        :             }
        :           }
  90   11:         SET {
  92    9:           SEQUENCE {
  94    3:             OBJECT IDENTIFIER countryName (2 5 4 6)
  99    2:             PrintableString 'DE'
        :             }
        :           }
        :         }
        :       }
 103   89:     [4] {
 105   87:       SEQUENCE {
 107   32:         SET {
 109   30:           SEQUENCE {
 111    3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 116   23:             UTF8String 'TMS CryptID 1 TEST-ONLY'
        :             }
        :           }
 141   38:         SET {
 143   36:           SEQUENCE {
 145    3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 150   29:             UTF8String 'arvato Systems GmbH NOT-VALID'
        :             }
        :           }
 181   11:         SET {
 183    9:           SEQUENCE {
 185    3:             OBJECT IDENTIFIER countryName (2 5 4 6)
 190    2:             PrintableString 'DE'
        :             }
        :           }
        :         }
        :       }
 194   17:     [0] {
 196   15:       GeneralizedTime 07/01/2015 09:07:02 GMT
        :       }
 213   13:     [1] {
 215   11:       SEQUENCE {
 217    9:         OBJECT IDENTIFIER
        :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
        :         }
        :       }
 228   10:     [4] {
 230    8:       OCTET STRING F1 50 E7 74 C7 9E 64 09
        :       }
 240   16:     [8] {
 242   14:       SEQUENCE {
 244   12:         SEQUENCE {
 246    8:           OBJECT IDENTIFIER implicitConfirm (1 3 6 1 5 5 7 4 13)
 256    0:           NULL
        :           }
        :         }
        :       }
        :     }
 258  670:   [2] {
 262  666:     SEQUENCE {
 266  662:       SEQUENCE {
 270  339:         SEQUENCE {
 274    1:           INTEGER 1
 277  332:           SEQUENCE {
 281    1:             [0] 02
 284   33:             [5] {
 286   31:               SEQUENCE {
 288   29:                 SET {
 290   27:                   SEQUENCE {
 292    3:                     OBJECT IDENTIFIER commonName (2 5 4 3)
 297   20:                     UTF8String '80276000112233466669'
        :                     }
        :                   }
        :                 }
        :               }
 319  290:             [6] {
 323   13:               SEQUENCE {
 325    9:                 OBJECT IDENTIFIER
        :                   rsaEncryption (1 2 840 113549 1 1 1)
```

```
 336    0:                 NULL
       :                 }
 338  271:               BIT STRING, encapsulates {
 343  266:                 SEQUENCE {
 347  257:                   INTEGER
       :                     23
 608    3:                   INTEGER 65537
       :                   }
       :                 }
       :               }
       :             }
       :           }
 613  276:         [1] {
 617   13:           SEQUENCE {
 619    9:             OBJECT IDENTIFIER
       :               sha256WithRSAEncryption (1 2 840 113549 1 1 11)
 630    0:             NULL
       :             }
 632  257:           BIT STRING
       :           }
 893   37:         SEQUENCE {
 895   35:           SEQUENCE {
 897    9:             OBJECT IDENTIFIER utf8Pairs (1 3 6 1 5 5 7 5 2 1)
 908   22:             UTF8String 'cert_profile?C.AK.AUT%'
       :             }
       :           }
       :         }
       :       }
       :     }
 932  261:   [0] {
 936  257:     BIT STRING
       :     }
       :   }
```

Listing 6: Beispiel-CMP-Request der Beantragung eines X.509-Zertifikats (C.AK.AUT)

```
   0 1695: SEQUENCE {
   4  146:   SEQUENCE {
   7    1:     INTEGER 2
  10   19:     [4] {
  12   17:       SEQUENCE {
  14   15:         SET {
  16   13:           SEQUENCE {
  18    3:             OBJECT IDENTIFIER commonName (2 5 4 3)
  23    6:             UTF8String 'INTCA1'
       :             }
       :           }
       :         }
       :       }
  31   91:     [4] {
  33   89:       SEQUENCE {
  35   10:         SET {
  37    8:           SEQUENCE {
  39    3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
  44    1:             PrintableString '4'
       :             }
       :           }
  47   30:         SET {
  49   28:           SEQUENCE {
  51    3:             OBJECT IDENTIFIER commonName (2 5 4 3)
  56   21:             UTF8String 'OLGA-777777 TEST-ONLY'
       :             }
       :           }
  79   30:         SET {
  81   28:           SEQUENCE {
  83    3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
  88   21:             UTF8String 'OLGA-777777 NOT-VALID'
       :             }
       :           }
 111   11:         SET {
 113    9:           SEQUENCE {
 115    3:             OBJECT IDENTIFIER countryName (2 5 4 6)
 120    2:             PrintableString 'DE'
       :             }
       :           }
       :         }
       :       }
 124   15:     [1] {
 126   13:       SEQUENCE {
 128    9:         OBJECT IDENTIFIER
       :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
 139    0:         NULL
       :         }
       :       }
 141   10:     [4] {
 143    8:       OCTET STRING F1 50 E7 74 C7 9E 64 09
       :       }
       :     }
 153 1277:   [3] {
 157 1273:     SEQUENCE {
 161    2:       [1] {
 163    0:         SEQUENCE {}
       :         }
 165 1265:       SEQUENCE {
```

```
169 1261:         SEQUENCE {
173    1:           INTEGER 1
176    3:           SEQUENCE {
178    1:             INTEGER 0
      :              }
181 1249:           SEQUENCE {
185 1245:             [0] {
189 1241:               SEQUENCE {
193  961:                 SEQUENCE {
197    3:                   [0] {
199    1:                     INTEGER 2
      :                     }
202    3:                   INTEGER 62415
207   13:                   SEQUENCE {
209    9:                     OBJECT IDENTIFIER
      :                       sha256WithRSAEncryption (1 2 840 113549 1 1 11)
220    0:                     NULL
      :                     }
222  131:                   SEQUENCE {
225   11:                     SET {
227    9:                       SEQUENCE {
229    3:                         OBJECT IDENTIFIER countryName (2 5 4 6)
234    2:                         PrintableString 'DE'
      :                         }
      :                       }
238   31:                     SET {
240   29:                       SEQUENCE {
242    3:                         OBJECT IDENTIFIER organizationName (2 5 4 10)
247   22:                         UTF8String 'gematik GmbH NOT-VALID'
      :                         }
      :                       }
271   50:                     SET {
273   48:                       SEQUENCE {
275    3:                         OBJECT IDENTIFIER
      :                           organizationalUnitName (2 5 4 11)
280   41:                         UTF8String
      :                       'Komponenten-CA der Telematikinfrastruktur'
      :                         }
      :                       }
323   31:                     SET {
325   29:                       SEQUENCE {
327    3:                         OBJECT IDENTIFIER commonName (2 5 4 3)
332   22:                         UTF8String 'GEM.KOMP-CA1 TEST-ONLY'
      :                         }
      :                       }
      :                     }
356   30:                   SEQUENCE {
358   13:                     UTCTime 07/01/2015 09:07:03 GMT
373   13:                     UTCTime 06/01/2020 09:07:02 GMT
      :                     }
388  174:                   SEQUENCE {
391   11:                     SET {
393    9:                       SEQUENCE {
395    3:                         OBJECT IDENTIFIER countryName (2 5 4 6)
400    2:                         PrintableString 'DE'
      :                         }
      :                       }
404   15:                     SET {
406   13:                       SEQUENCE {
408    3:                         OBJECT IDENTIFIER
      :                           stateOrProvinceName (2 5 4 8)
413    6:                         UTF8String 'Bayern'
      :                         }
      :                       }
421   16:                     SET {
423   14:                       SEQUENCE {
425    3:                         OBJECT IDENTIFIER localityName (2 5 4 7)
430    7:                         UTF8String 'Testort'
      :                         }
      :                       }
439   42:                     SET {
441   40:                       SEQUENCE {
443    3:                         OBJECT IDENTIFIER organizationName (2 5 4 10)
448   33:                         UTF8String 'OLGA-777777 TEST-ONLY - NOT-VALID'
      :                         }
      :                       }
483   38:                     SET {
485   36:                       SEQUENCE {
487    3:                         OBJECT IDENTIFIER commonName (2 5 4 3)
492   29:                         UTF8String '80276000112233466669-20150107'
      :                         }
      :                       }
523   14:                     SET {
525   12:                       SEQUENCE {
527    3:                         OBJECT IDENTIFIER postalCode (2 5 4 17)
532    5:                         UTF8String '44458'
      :                         }
      :                       }
539   24:                     SET {
541   22:                       SEQUENCE {
543    3:                         OBJECT IDENTIFIER streetAddress (2 5 4 9)
548   15:                         UTF8String 'Hauptstra..e 15'
      :                         }
      :                       }
      :                     }
565  290:                   SEQUENCE {
569   13:                     SEQUENCE {
```

```
 571    9:                       OBJECT IDENTIFIER
        :                         rsaEncryption (1 2 840 113549 1 1 1)
 582    0:                       NULL
        :                       }
 584  271:                     BIT STRING, encapsulates {
 589  266:                       SEQUENCE {
 593  257:                         INTEGER
 854    3:                         INTEGER 65537
        :                         }
        :                       }
        :                     }
 859  295:                   [3] {
 863  291:                     SEQUENCE {
 867   29:                       SEQUENCE {
 869    3:                         OBJECT IDENTIFIER
        :                           subjectKeyIdentifier (2 5 29 14)
 874   22:                         OCTET STRING, encapsulates {
 876   20:                           OCTET STRING
        :                           84 E0 91 80 83 24 D5 0A 80 D1 5D 33 34 20 E4 A1
        :                           94 69 93 06
        :                           }
        :                         }
 898   31:                       SEQUENCE {
 900    3:                         OBJECT IDENTIFIER
        :                           authorityKeyIdentifier (2 5 29 35)
 905   24:                         OCTET STRING, encapsulates {
 907   22:                           SEQUENCE {
 909   20:                             [0]
        :                           CF 19 C1 69 FB 70 D3 8D 48 B9 11 FE 1D 99 65 DF
        :                           D8 E5 C8 F2
        :                             }
        :                           }
        :                         }
 931   75:                       SEQUENCE {
 933    8:                         OBJECT IDENTIFIER
        :                           authorityInfoAccess (1 3 6 1 5 5 7 1 1)
 943   63:                         OCTET STRING, encapsulates {
 945   61:                           SEQUENCE {
 947   59:                             SEQUENCE {
 949    8:                               OBJECT IDENTIFIER
        :                                 ocsp (1 3 6 1 5 5 7 48 1)
 959   47:                               [6]
        :                           'http://ocsp-testref.komp-ca.telematik-test/ocsp'
        :                             }
        :                           }
        :                         }
        :                       }
1008   12:                       SEQUENCE {
1010    3:                         OBJECT IDENTIFIER
        :                           basicConstraints (2 5 29 19)
1015    1:                         BOOLEAN TRUE
1018    2:                         OCTET STRING, encapsulates {
1020    0:                           SEQUENCE {}
        :                           }
        :                         }
1022   14:                       SEQUENCE {
1024    3:                         OBJECT IDENTIFIER keyUsage (2 5 29 15)
1029    1:                         BOOLEAN TRUE
1032    4:                         OCTET STRING, encapsulates {
1034    2:                           BIT STRING 5 unused bits
        :                             '101'B
        :                           }
        :                         }
1038   29:                       SEQUENCE {
1040    3:                         OBJECT IDENTIFIER extKeyUsage (2 5 29 37)
1045   22:                         OCTET STRING, encapsulates {
1047   20:                           SEQUENCE {
1049    8:                             OBJECT IDENTIFIER
        :                               serverAuth (1 3 6 1 5 5 7 3 1)
1059    8:                             OBJECT IDENTIFIER
        :                               clientAuth (1 3 6 1 5 5 7 3 2)
        :                             }
        :                           }
        :                         }
1069   32:                       SEQUENCE {
1071    3:                         OBJECT IDENTIFIER
        :                           certificatePolicies (2 5 29 32)
1076   25:                         OCTET STRING, encapsulates {
1078   23:                           SEQUENCE {
1080   10:                             SEQUENCE {
1082    8:                               OBJECT IDENTIFIER '1 2 276 0 76 4 163'
        :                               }
1092    9:                             SEQUENCE {
1094    7:                               OBJECT IDENTIFIER '1 2 276 0 76 4 79'
        :                               }
        :                             }
        :                           }
        :                         }
1103   53:                       SEQUENCE {
1105    5:                         OBJECT IDENTIFIER admission (1 3 36 8 3 3)
1112   44:                         OCTET STRING, encapsulates {
1114   42:                           SEQUENCE {
1116   40:                             SEQUENCE {
1118   38:                               SEQUENCE {
1120   36:                                 SEQUENCE {
1122   34:                                   SEQUENCE {
1124   21:                                     SEQUENCE {
```

```
1126   19:                                      UTF8String 'Anwendungskonnektor'
         :                                      }
1147    9:                                    SEQUENCE {
1149    7:                                      OBJECT IDENTIFIER '1 2 276 0 76 4 103'
         :                                      }
         :                                    }
         :                                  }
         :                                }
         :                              }
         :                            }
         :                          }
         :                        }
         :                      }
         :                    }
         :                  }
1158   13:                SEQUENCE {
1160    9:                  OBJECT IDENTIFIER
         :                    sha256WithRSAEncryption (1 2 840 113549 1 1 11)
1171    0:                  NULL
         :                  }
1173  257:                BIT STRING
         :                }
         :              }
         :            }
         :          }
         :        }
         :      }
         :    }
1434  261:    [0] {
1438  257:      BIT STRING
         :      }
         :    }
```

Listing 7: Beispiel-CMP-Response der Beantragung eines X.509-Zertifikats (C.AK.AUT)

```
   0 650: SEQUENCE {
   4 146:   SEQUENCE {
   7   1:     INTEGER 2
  10  19:     [4] {
  12  17:       SEQUENCE {
  14  15:         SET {
  16  13:           SEQUENCE {
  18   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
  23   6:             UTF8String 'INTCA1'
         :             }
         :           }
         :         }
         :       }
  31  91:     [4] {
  33  89:       SEQUENCE {
  35  10:         SET {
  37   8:           SEQUENCE {
  39   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
  44   1:             PrintableString '5'
         :             }
         :           }
  47  30:         SET {
  49  28:           SEQUENCE {
  51   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
  56  21:             UTF8String 'OLGA-777777 TEST-ONLY'
         :             }
         :           }
  79  30:         SET {
  81  28:           SEQUENCE {
  83   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
  88  21:             UTF8String 'OLGA-777777 NOT-VALID'
         :             }
         :           }
 111  11:         SET {
 113   9:           SEQUENCE {
 115   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
 120   2:             PrintableString 'DE'
         :             }
         :           }
         :         }
         :       }
 124  15:     [1] {
 126  13:       SEQUENCE {
 128   9:         OBJECT IDENTIFIER
         :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
 139   0:         NULL
         :         }
         :       }
 141  10:     [4] {
 143   8:       OCTET STRING 7D FF 27 B0 80 87 15 7E
         :       }
         :     }
 153 233:   [3] {
 156 230:     SEQUENCE {
 159   2:       [1] {
 161   0:         SEQUENCE {}
         :         }
 163 223:       SEQUENCE {
```

```
166 220:         SEQUENCE {
169   1:           INTEGER 1
172 214:           SEQUENCE {
175   1:             INTEGER 2
178 204:             SEQUENCE {
181 201:               UTF8String
       :                 'ALREADY_ISSUED: Certificate for the given subjec'
       :                 't cn=test.vsdd.opxkcat777777.com,serialNumber=11'
       :                 '111-B,o=OLGA-777777 TEST-ONLY - NOT-VALID,c=DE a'
       :                 'nd profile C.FD.TLS-S.CMS already issued: CertRe'
       :                 'qId: 4019'
       :               }
385   2:             BIT STRING 5 unused bits
       :               '100'B (bit 2)
       :             }
       :           }
       :         }
       :       }
       :     }
389 261:     [0] {
393 257:       BIT STRING
       :       }
       :     }
```

Listing 8: Beispiel-CMP-Response Fehlerfall X.509-Zertifikat bereits ausgestellt (C.FD.TLS-S)

5.3.4 CMP-Request und CMP-Response (CV)

Die CMP-Schnittstelle zum Beantragen von CV-Komponenten-Zertifikaten wird über HTTP(S) angeboten.

Während die CMP-Requests mit einem CMP-Requestor (Crypt-ID) signiert werden, werden die CMP-Responses – abgesehen von CMP-Error-Responses - mit einem CMP-Responder signiert.

Gemäß [RFC4210] sind die CMP-Requests und –Responses DER-kodiert im Body des POST-Befehls zu übermitteln. Der Content-Type für die Requests und Responses ist application/pkixcmp.

Das nachfolgende Listing stellt die Struktur der CMP-Nachricht zur Beantragung von Komponenten-Zertifikaten über die CMP-Schnittstelle dar:

```
PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection [0] EXPLICIT PKIProtection,                  -- erforderlich
}

PKIHeader ::= SEQUENCE {
    pvno              INTEGER cmp2000(2),                   -- cmp1999 (1) ist nicht erlaubt
    sender            GeneralName,                          -- directoryName des CMP-Requesters
    recipient         GeneralName,                          -- directoryName der CMP-Reponders
    messageTime   [0] EXPLICIT GeneralizedTime,             -- Timestamp der Nachricht, erforderlich (OPTIONAL nach RFC
4210)
    protectionAlg [1] EXPLICIT AlgorithmIdentifier,         -- Signaturalgorithm für PKIMessage.protection, hier
erforderlich
    transactionID [4] EXPLICIT OCTET STRING,                -- erforderlich (OPTIONAL nach RFC 4210)
    generalInfo   [8] EXPLICIT SEQUENCE SIZE (1..MAX) OF InfoTypeAndValue
                                                            -- implicitConfirm ist mit dem OID (1 3 6 1 5 5 7 4 13}
                                                               erforderlich
                                                            -- (OPTIONAL nach RFC 4210)
}

PKIBody ::= CHOICE
{
   ...
   cr [2] EXPLICIT CertReqMessages,                 -- Nur diese Variante ist erlaubt
   ...
}
CertReqMessages ::= SEQUENCE OF CertReqMsg

CertReqMsg ::= SEQUENCE {
    certReq CertRequest,
    popo    ProofOfPossession                       -- erforderlich (OPTIONAL nach RFC 4211)
}

ProofOfPossession ::= CHOICE {
   ...
   signature         [1] POPOSigningKey,            -- Nur diese Variante ist erlaubt
   ...
}
CertRequest ::= SEQUENCE {
    certReqId       INTEGER,
    certTemplate    SEQUENCE {},                    -- leere SEQUENCE, da kein X.509-Zertifikat; CV-Zertifikatstemplate
```

```
ist in controls enthalten
    controls        SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue
                                                -- hier control AltCertTemplate ist erfoderlich
}

AltCertTemplate ::= AttributeTypeAndValue
          -- type id-regCtrl-altCertTemplate (1 3 6 1 5 5 7 5 1 7)
          -- value G2CvCertTemplate


G2CvCertTemplate ::= SEQUENCE {
    type    OBJECT IDENTIFIER id-gematik-g2CvCertTemplate (1.2.276.0.76.3.1.91.44.3.1)
    value   SEQUENCE {
        chr             OCTET STRING chr,                   -- CHR gemäß gemSpec_PKI: 12 Bytes binär/BCD codiert
                                                            -- zugelassene CHR-Werte  {KeyID]:
                                                            -- C.SMC.AUT_CVC.E256:       '00 05' || ICCSN    [05]
                                                            -- C.SAK.AUTD_CVC.E256:      '00 0A' || ICCSN    [0A]
                                                            -- C.SMC.AUTD_RPS_CVC.E256:  '00 0A' || ICCSN    [0A]

        publicKey       CvECPublicKeyInfo,
        autorisation    G2CvAuthorisation
    }
}

CvECPublicKeyInfo ::= SEQUENCE{
    algorithm SEQUENCE {
        algorithm   OBJECT IDENTIFIER   -- brainPoolP256r1:  authS_gemSpec-COS-G2_ecc-with-sha256 (1 3 36 3 5 3 1)
        parameters  NULL
    }
    subjectPublicKey ECPoint BIT STRING {
        uncompressed ECPoint            -- brainPoolP256r1: '04' || Q_x (32 Byte) || Q_y (32 Byte)
    }
}


G2CvAuthorisation ::= CHOICE {
    role INTEGER,
    chat CHAT
}
CHAT ::= SEQUENCE {
    type        OBJECT IDENTIFIER,
                -- oid_cvc_fl_ti (1 2 276 0 76 4 152)
                -- oid_cvc_fl_cms (1 2 276 0 76 4 153)
    flaglist    OCTECT STRING

}
```

Listing 9: Struktur eines CMP-Requests zur Beantragung von CV-Komponenten-Zertifikaten

Der CMP-Responder prüft den CMP-Request. Im negativen Fall antwortet der Responder mit einer CMP-Response in der Variante "error [23] ErrorMsgContent" im PKIBody.

Andererseits antwortet der Responder mit einer Nachricht auf Basis der im nachfolgenden Listing dargestellten Struktur:

```
PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection [0] EXPLICIT PKIProtection,          -- erforderlich
}

PKIHeader ::= SEQUENCE {
    pvno              INTEGER cmp2000(2),           -- fix
    sender            GeneralName,                  -- directoryName der CA
    recipient         GeneralName,                  -- directoryName des Reqesters
    messageTime   [0] EXPLICIT GeneralizedTime,     -- Timestamp der Nachricht
    protectionAlg [1] EXPLICIT AlgorithmIdentifier, -- Signaturalgorithmus für PKIMessage.protection erforderlich
    transactionID [4] EXPLICIT OCTET STRING,        -- transactionID aus dem Request
    generalInfo   [8] EXPLICIT SEQUENCE SIZE (1..MAX) OF InfoTypeAndValue
                                                    -- implicitConfirm ist mit dem OID (1 3 6 1 5 5 7 4 13} erfoderlich
}

PKIBody ::= CHOICE
{
   ...
   cp [3] EXPLICIT CertRepMessage,                  -- Nur diese Variante ist erlaubt
   ...
}

CertRepMessage ::= SEQUENCE {
   caPubs          [1] EXPLICIT SEQUENCE SIZE (1..MAX) OF Certificate
                                                    -- enthältet das CA-CVZertifikat, erforderlich
                                                       (OPTIONAL nach RFC 4210)
   response            SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
   certReqId           INTEGER,                      -- Wert aus CMP-Request
   status              PKIStatusInfo,                -- mit dem PKIStatus accepted (0) der grantedWithMods im positiven
Fall,
                                                     -- ansonsten rejection (2)
   certifiedKeyPair    CertifiedKeyPair    OPTIONAL  -- nur vorhanden, wenn das Zertifikat erfolgreich ausgestellt wurde
```

```
}


CertifiedKeyPair ::= SEQUENCE {
   certOrEncCert       CertOrEncCert
}

CertOrEncCert ::= CHOICE {
   certificate     [0] Certificate,         -- Nur diese Variante ist erlaubt
   ...
}

CMPCertificate ::= CHOICE {
   x509v3PKCert                Certificate, -- wird nicht benutzt
   g2CvCert        [4] EXPLICIT  G2CvCert    -- wird verwendet
}

G2CvCert ::= OCTET STRING
```

Listing 10: Struktur der CMP-Response (CV)

```
  0 777: SEQUENCE {
  4 251:   SEQUENCE {
  7   1:     INTEGER 2
 10  91:     [4] {
 12  89:       SEQUENCE {
 14  10:         SET {
 16   8:           SEQUENCE {
 18   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
 23   1:             PrintableString '4'
      :             }
      :           }
 26  30:         SET {
 28  28:           SEQUENCE {
 30   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 35  21:             UTF8String 'OLGA-777777 TEST-ONLY'
      :             }
      :           }
 58  30:         SET {
 60  28:           SEQUENCE {
 62   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 67  21:             UTF8String 'OLGA-777777 NOT-VALID'
      :             }
      :           }
 90  11:         SET {
 92   9:           SEQUENCE {
 94   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
 99   2:             PrintableString 'DE'
      :             }
      :           }
      :         }
      :       }
103  89:     [4] {
105  87:       SEQUENCE {
107  32:         SET {
109  30:           SEQUENCE {
111   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
116  23:             UTF8String 'TMS CryptID 1 TEST-ONLY'
      :             }
      :           }
141  38:         SET {
143  36:           SEQUENCE {
145   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
150  29:             UTF8String 'arvato Systems GmbH NOT-VALID'
      :             }
      :           }
181  11:         SET {
183   9:           SEQUENCE {
185   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
190   2:             PrintableString 'DE'
      :             }
      :           }
      :         }
      :       }
194  17:     [0] {
196  15:       GeneralizedTime 07/01/2015 09:12:52 GMT
      :       }
213  13:     [1] {
215  11:       SEQUENCE {
217   9:         OBJECT IDENTIFIER
      :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
      :         }
      :       }
228  10:     [4] {
230   8:       OCTET STRING 19 0F 27 26 D2 D8 95 3E
      :       }
240  16:     [8] {
242  14:       SEQUENCE {
244  12:         SEQUENCE {
246   8:           OBJECT IDENTIFIER implicitConfirm (1 3 6 1 5 5 7 4 13)
256   0:           NULL
      :           }
      :         }
      :       }
      :     }
258 255:   [2] {
```

```
261 252:     SEQUENCE {
264 249:       SEQUENCE {
267 159:         SEQUENCE {
270   1:           INTEGER 1
273   0:           SEQUENCE {}
275 151:           SEQUENCE {
278 148:             SEQUENCE {
281   9:               OBJECT IDENTIFIER altCertTemplate (1 3 6 1 5 5 7 5 1 7)
292 134:               SEQUENCE {
295  11:                 OBJECT IDENTIFIER '1 2 276 0 76 3 1 91 44 3 1'
308 119:                 SEQUENCE {
310  12:                   OCTET STRING 00 0A 80 27 60 00 11 22 33 46 66 71
324  80:                   SEQUENCE {
326  10:                     SEQUENCE {
328   6:                       OBJECT IDENTIFIER '1 3 36 3 5 3 1'
336   0:                       NULL
       :                       }
338  66:                     BIT STRING
       :                     }
406   1:                   INTEGER 54
409   8:                   [1] {
411   6:                     PrintableString '150107'
       :                     }
419   8:                   [2] {
421   6:                     PrintableString '150121'
       :                     }
       :                   }
       :                 }
       :               }
       :             }
       :           }
429  85:         [1] {
431  10:           SEQUENCE {
433   8:             OBJECT IDENTIFIER ecdsaWithSHA256 (1 2 840 10045 4 3 2)
       :             }
443  71:           BIT STRING, encapsulates {
446  68:             SEQUENCE {
448  32:               INTEGER
482  32:               INTEGER
       :               }
       :             }
       :           }
       :         }
       :       }
       :     }
516 261:   [0] {
520 257:     BIT STRING
       :     }
       :   }
```

Listing 11: Beispiel-CMP-Request der Beantragung eines CV-Zertifikats (54)

```
  0 902: SEQUENCE {
  4 146:   SEQUENCE {
  7   1:     INTEGER 2
 10  19:     [4] {
 12  17:       SEQUENCE {
 14  15:         SET {
 16  13:           SEQUENCE {
 18   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 23   6:             UTF8String 'INTCA1'
       :             }
       :           }
       :         }
       :       }
 31  91:     [4] {
 33  89:       SEQUENCE {
 35  10:         SET {
 37   8:           SEQUENCE {
 39   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
 44   1:             PrintableString '4'
       :             }
       :           }
 47  30:         SET {
 49  28:           SEQUENCE {
 51   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 56  21:             UTF8String 'OLGA-777777 TEST-ONLY'
       :             }
       :           }
 79  30:         SET {
 81  28:           SEQUENCE {
 83   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 88  21:             UTF8String 'OLGA-777777 NOT-VALID'
       :             }
       :           }
111  11:         SET {
113   9:           SEQUENCE {
115   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
120   2:             PrintableString 'DE'
       :             }
       :           }
       :         }
       :       }
124  15:     [1] {
```

```
126  13:       SEQUENCE {
128   9:         OBJECT IDENTIFIER
       :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
139   0:         NULL
       :         }
       :       }
141  10:     [4] {
143   8:       OCTET STRING 19 0F 27 26 D2 D8 95 3E
       :       }
       :     }
153 484:   [3] {
157 480:     SEQUENCE {
161 229:       [1] {
164 226:         SEQUENCE {
167 223:           [4] {
170 220:             OCTET STRING
       :             }
       :           }
       :         }
393 245:       SEQUENCE {
396 242:         SEQUENCE {
399   1:           INTEGER 1
402   3:           SEQUENCE {
404   1:             INTEGER 0
       :             }
407 231:           SEQUENCE {
410 228:             [0] {
413 225:               [4] {
416 222:                 OCTET STRING
       :                 }
       :               }
       :             }
       :           }
       :         }
       :       }
       :     }
641 261:   [0] {
645 257:     BIT STRING
       :     }
       :   }
```

Listing 12: Beispiel-CMP-Response der Beantragung eines CV-Zertifikats (54)

```
  0 478: SEQUENCE {
  4 146:   SEQUENCE {
  7   1:     INTEGER 2
 10  19:     [4] {
 12  17:       SEQUENCE {
 14  15:         SET {
 16  13:           SEQUENCE {
 18   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 23   6:             UTF8String 'INTCA1'
       :             }
       :           }
       :         }
       :       }
 31  91:     [4] {
 33  89:       SEQUENCE {
 35  10:         SET {
 37   8:           SEQUENCE {
 39   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
 44   1:             PrintableString '5'
       :             }
       :           }
 47  30:         SET {
 49  28:           SEQUENCE {
 51   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 56  21:             UTF8String 'OLGA-777777 TEST-ONLY'
       :             }
       :           }
 79  30:         SET {
 81  28:           SEQUENCE {
 83   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 88  21:             UTF8String 'OLGA-777777 NOT-VALID'
       :             }
       :           }
111  11:         SET {
113   9:           SEQUENCE {
115   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
120   2:             PrintableString 'DE'
       :             }
       :           }
       :         }
       :       }
124  15:     [1] {
126  13:       SEQUENCE {
128   9:         OBJECT IDENTIFIER
       :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
139   0:         NULL
       :         }
       :       }
141  10:     [4] {
143   8:       OCTET STRING 40 CD A2 D4 A3 9F 0B 1C
```

```
      :         }
      :       }
153  62:     [3] {
155  60:       SEQUENCE {
157   2:         [1] {
159   0:           SEQUENCE {}
      :           }
161  54:         SEQUENCE {
163  52:           SEQUENCE {
165   1:             INTEGER 1
168  47:             SEQUENCE {
170   1:               INTEGER 2
173  36:               SEQUENCE {
175  34:                 UTF8String 'BAD_CERT_TEMPLATE: CertReqId: 4023'
      :                 }
211   4:               BIT STRING 4 unused bits
      :                 '1000000000000000000'B (bit 19)
      :               }
      :             }
      :           }
      :         }
      :       }
      :     }
217 261:     [0] {
221 257:       BIT STRING
      :       }
      :     }
```

Listing 13: Beispiel-CMP-Response Beantragung eines CV-Zertifikats Fehlerfall Falscher PublicKey

5.3.5 SOAP-Request und -Response

Die SOAP-Schnittstelle zum Beantragen von X.509- und CV-Komponenten-Zertifikaten wird unter Verwendung des Standards Webservice-Security (WSS) realisiert und über HTTP(S) angeboten.

Alle SOAP-Requests müssen mit der Crypt-ID signiert werden, wobei folgende Objekte zu signieren sind (falls vorhanden):

- {Element}{http://www.w3.org/2005/08/addressing}FaultTo
- {Element}{http://www.w3.org/2005/08/addressing}ReplyTo
- {Element}{http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd}Timestamp
- {Content}{http://www.w3.org/2003/05/soap-envelope}Body

Für die XML-Signatur ist der Algorithmus RSASSA-PSS mit Mask Generation Function (MGF) SHA-256 mit einer Salt-Länge von 32 Byte einzusetzen:

- http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1

Alle SOAP-Responses werden vom TMS ebenfalls mit einer Crypt-ID signiert. Somit erfolgt die SOAP-Kommunikation gegenseitig authentisiert und die Integrität wird sichergestellt.

Ein Beispiel für eine XML-Signatur ist im folgenden Listing dargestellt.

```
<ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="soap"/>
        </ds:CanonicalizationMethod>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-
MGF1"/>
        <ds:Reference URI="#TS-Timestamp Id">
                <ds:Transforms>
                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                <ec:InclusiveNamespaces
```

```
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
                                                        PrefixList="wsse
soap"/>
                    </ds:Transform>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue>VEoZhYEFTd9ExObbFXxK+1mIS2c=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#id-BodyId">
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <ec:InclusiveNamespaces
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
                                                        PrefixList=""/>
                    </ds:Transform>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue>mW0p33+w8BgYzgVD8Yp0Qz0FzQ8=</ds:DigestValue>
        </ds:Reference>
</ds:SignedInfo>
```

Listing 14: Beispiel SignedInfo für eine XML-Signatur

Das dazu verwendbare WSDL-Schema für SOAP-Nachrichten „CertificateManagementService.wsdl“ zur Beantragung von X.509-Komponenten- und CV-Zertifikaten einschließlich der zugehörigen SOAP-Responses ist als Anlage beigefügt.

Zum Senden eines Zertifikatantrags per SOAP muss die in der WSDL-Datei beschriebene Methode (operation) „requestCertificate“ genutzt werden. Diese Operation besitzt die input-Message „tns:requestCertificate“, welche auf den Typ „requestCertificate“ verweist. Als Inhalt des „requestCertificate“-Elements muss hierbei als Kindelement genau ein „CertReqMessagesType“ als „arg0“-Element der Methode übergeben werden. Als Kindelement können nun mehrere konkrete Zertifikatsanträge vom Typ „CertReqMessageType“ übergeben werden.

Die Rückmeldung erfolgt hierzu analog, jedoch wird als Kindelement der Typ „CertRepMessageType“ ebenfalls genau einmal zurückgeliefert. Die Typen sind in der WSDL beschrieben.

In den folgenden Listings (Listing 15 bis Listing 18) sind Beispiel-SOAP-Request- und –Responses für die Zertifikatsausstellung dargestellt. Für PKCS#10-Requests, Zertifikaten und Signaturen wurden Platzhalter eingesetzt.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" SOAP-
ENV:mustUnderstand="1">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-abe4f563-18e4-4837-9b3d-31f8c6abed17">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
                    <ds:Reference URI="#id-9e83e745-0ae9-4281-a176-d1c1b8e88253">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <ds:DigestValue>l0XsL4Pq6Va49O1aD5D4jRVvZxwRLVqCUAokIZbo2eE=</ds:DigestValue>
                    </ds:Reference>
                    <ds:Reference URI="#TS-41eeb52f-167b-4a47-aa43-84d193231c36">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <ds:DigestValue>bQcNlM3RdCAHW3wX1Zes27wR81ydUTJS1h79OKedEeQ=</ds:DigestValue>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue><!-- BASE64-Codierte Signatur --></ds:SignatureValue>
                <ds:KeyInfo Id="KI-b2dfc079-7327-4f18-97b1-1daf566a1472">
                    <wsse:SecurityTokenReference wsu:Id="STR-e3e37140-b140-49ad-b176-3d9bdf794396">
                        <ds:X509Data>
                            <ds:X509IssuerSerial>
                                <ds:X509IssuerName>C=DE,O=arvato Systems GmbH,OU=Infrastruktur-CA,CN=Infrastruktur-
CA1</ds:X509IssuerName>
```

```
                            <ds:X509SerialNumber>14677</ds:X509SerialNumber>
                        </ds:X509IssuerSerial>
                    </ds:X509Data>
                </wsse:SecurityTokenReference>
            </ds:KeyInfo>
        </ds:Signature>
        <wsu:Timestamp wsu:Id="TS-41eeb52f-167b-4a47-aa43-84d193231c36">
            <wsu:Created>2015-01-06T12:14:28.614Z</wsu:Created>
            <wsu:Expires>2015-01-06T13:04:28.614Z</wsu:Expires>
        </wsu:Timestamp>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="id-9e83e745-0ae9-4281-a176-d1c1b8e88253">
    <ns2:requestCertificate xmlns="http://ws.gematik.de/pki/ComponentCertificateService/v1.0"
xmlns:ns2="http://ws.gematik.de/pki/WSDL/ComponentCertificateService/v1.0">
        <arg0 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CertReqMessagesType">
            <certReqMessage>
                <certReqId>TEST123</certReqId>
                <p10Request><!-- BASE64-Codierter PKCS#10-Request --></p10Request>
                <certProfile>
                    <x509CertProfile>C.AK.AUT</x509CertProfile>
                </certProfile>
            </certReqMessage>
        </arg0>
    </ns2:requestCertificate>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 15: Beispiel-SOAP-Request für die Beantragung eines X.509-Zertifikats

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-B6B31C2C47C59F1EF9142054647958972">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="soap"/>
                    </ds:CanonicalizationMethod>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
                    <ds:Reference URI="#TS-B6B31C2C47C59F1EF9142054647958867">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="wsse soap"/>
                            </ds:Transform>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>7Ah3hZ/VdDKcY7RCEUIJKHvAmk0=</ds:DigestValue>
                    </ds:Reference>
                    <ds:Reference URI="#id-B6B31C2C47C59F1EF9142054647958971">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList=""/>
                            </ds:Transform>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>kG7jVJS/e5TR7gO4DURs+nQS3s0=</ds:DigestValue>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue><!-- BASE64-Codierte Signatur --></ds:SignatureValue>
                <ds:KeyInfo Id="KI-B6B31C2C47C59F1EF9142054647958869">
                    <wsse:SecurityTokenReference wsu:Id="STR-B6B31C2C47C59F1EF9142054647958870">
                        <ds:X509Data>
                            <ds:X509IssuerSerial>
                                <ds:X509IssuerName>C=DE,O=arvato Systems GmbH,OU=Infrastruktur-CA,CN=Infrastruktur-
CA1</ds:X509IssuerName>
                                <ds:X509SerialNumber>2</ds:X509SerialNumber>
                            </ds:X509IssuerSerial>
                        </ds:X509Data>
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
            <wsu:Timestamp wsu:Id="TS-B6B31C2C47C59F1EF9142054647958867">
                <wsu:Created>2015-01-06T12:14:39.587Z</wsu:Created>
                <wsu:Expires>2015-01-06T12:19:39.587Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </SOAP-ENV:Header>
    <soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="id-B6B31C2C47C59F1EF9142054647958971">
        <ns2:requestCertificateResponse xmlns="http://ws.gematik.de/pki/ComponentCertificateService/v1.0"
xmlns:ns2="http://ws.gematik.de/pki/WSDL/ComponentCertificateService/v1.0">
            <return>
                <caPubs/>
                <response>
                    <CertResponse>
                        <certReqId>TEST123</certReqId>
                        <status>
                            <status>0</status>
                            <failureInfo>0</failureInfo>
                        </status>
```

```
                        <statusMessage>OK</statusMessage>
                        <cert>
                            <x509v3PKCert><!-- BASE64-Codiertes Zertifikat --></x509v3PKCert>
                        </cert>
                    </CertResponse>
                </response>
            </return>
        </ns2:requestCertificateResponse>
    </soap:Body>
</soap:Envelope>
```

Listing 16: Beispiel-SOAP-Response der Beantragung eines X.509-Zertifikats

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" SOAP-
ENV:mustUnderstand="1">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-f20dd7a4-cbb7-4113-8cb7-4f4a0ea65362">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
                    <ds:Reference URI="#id-cd83e14e-5167-40f6-874f-281be2e385d8">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <ds:DigestValue>g3W4JGgzHRVgpYTqYYtwNhD5fUJDuh8PM3tJKFhlTOM=</ds:DigestValue>
                    </ds:Reference>
                    <ds:Reference URI="#TS-084afeb6-9fd9-4d39-9833-5beafa45e65b">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <ds:DigestValue>lLdOlZ5075ORJ2tYtM6yp9SFZuWhopo1sX9nAUO/r7s=</ds:DigestValue>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue><!-- BASE64-Codierte Signatur --></ds:SignatureValue>
                <ds:KeyInfo Id="KI-54ed959a-9036-4f12-91e4-7a291ac05de3">
                    <wsse:SecurityTokenReference wsu:Id="STR-ff04363c-9fd8-4cb5-9a2a-969e4c84219b">
                        <ds:X509Data>
                            <ds:X509IssuerSerial>
                                <ds:X509IssuerName>C=DE,O=arvato Systems GmbH,OU=Infrastruktur-CA,CN=Infrastruktur-CA1
TEST-ONLY</ds:X509IssuerName>
                                <ds:X509SerialNumber>14677</ds:X509SerialNumber>
                            </ds:X509IssuerSerial>
                        </ds:X509Data>
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
            <wsu:Timestamp wsu:Id="TS-084afeb6-9fd9-4d39-9833-5beafa45e65b">
                <wsu:Created>2015-01-01T11:21:29.922Z</wsu:Created>
                <wsu:Expires>2015-01-06T12:11:29.922Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="id-cd83e14e-5167-40f6-874f-281be2e385d8">
        <ns2:requestCertificate xmlns="http://ws.gematik.de/pki/ComponentCertificateService/v1.0"
xmlns:ns2="http://ws.gematik.de/pki/WSDL/ComponentCertificateService/v1.0">
            <arg0 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CertReqMessagesType">
                <certReqMessage>
                    <certReqId>TEST123</certReqId>
                    <p10Request><!-- BASE64-Codierter PKCS#10-Request --></p10Request>
                    <certProfile>
                        <cvRole>54</cvRole>
                    </certProfile>
                </certReqMessage>
            </arg0>
        </ns2:requestCertificate>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 17: Beispiel-SOAP-Request für die Beantragung eines CV-Zertifikats

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-B6B31C2C47C59F1EF9142054329088860">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="soap"/>
                    </ds:CanonicalizationMethod>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
                    <ds:Reference URI="#TS-B6B31C2C47C59F1EF9142054329088755">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="wsse soap"/>
```

```
                        </ds:Transform>
                    </ds:Transforms>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                    <ds:DigestValue>Jyhof/qfGaZEomx4px673F5JFzQ=</ds:DigestValue>
                </ds:Reference>
                <ds:Reference URI="#id-B6B31C2C47C59F1EF9142054329088859">
                    <ds:Transforms>
                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                            <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList=""/>
                        </ds:Transform>
                    </ds:Transforms>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                    <ds:DigestValue>F16yM74ZDy1FgWm3w1kLIZrx8Aw=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue><!-- BASE64-Codierte Signatur --></ds:SignatureValue>
            <ds:KeyInfo Id="KI-B6B31C2C47C59F1EF9142054329088757">
                <wsse:SecurityTokenReference wsu:Id="STR-B6B31C2C47C59F1EF9142054329088758">
                    <ds:X509Data>
                        <ds:X509IssuerSerial>
                            <ds:X509IssuerName>C=DE,O=arvato Systems GmbH,OU=Infrastruktur-CA,CN=Infrastruktur-
CA1</ds:X509IssuerName>
                            <ds:X509SerialNumber>2</ds:X509SerialNumber>
                        </ds:X509IssuerSerial>
                    </ds:X509Data>
                </wsse:SecurityTokenReference>
            </ds:KeyInfo>
        </ds:Signature>
        <wsu:Timestamp wsu:Id="TS-B6B31C2C47C59F1EF9142054329088755">
            <wsu:Created>2015-01-06T11:21:30.887Z</wsu:Created>
            <wsu:Expires>2015-01-06T11:26:30.887Z</wsu:Expires>
        </wsu:Timestamp>
    </wsse:Security>
</SOAP-ENV:Header>
<soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="id-B6B31C2C47C59F1EF9142054329088859">
    <ns2:requestCertificateResponse xmlns="http://ws.gematik.de/pki/ComponentCertificateService/v1.0"
xmlns:ns2="http://ws.gematik.de/pki/WSDL/ComponentCertificateService/v1.0">
        <return>
            <caPubs>
                <caPub>
                    <g2CvCert><!-- BASE64-Codierter öffentlicher Schlüssel der CA --></g2CvCert>
                </caPub>
            </caPubs>
            <response>
                <CertResponse>
                    <certReqId>TEST123</certReqId>
                    <status>
                        <status>0</status>
                        <failureInfo>0</failureInfo>
                    </status>
                    <statusMessage>OK</statusMessage>
                    <cert>
                        <g2CvCert><!-- BASE64-Codiertes CVC --></g2CvCert>
                    </cert>
                </CertResponse>
            </response>
        </return>
    </ns2:requestCertificateResponse>
</soap:Body>
</soap:Envelope>
```

Listing 18: Beispiel-SOAP-Response der Beantragung eines CV-Zertifikats

6 Zertifikatssperrung (X.509)

Zur Zertifikatssperrung von X.509 Zertifikaten wird Sperrberechtigten eine Web-Anwendung "Zertifikatssperrung Komponenten-PKI" (kurz ZS) über das TMS bereitgestellt. Darüber hinaus können Sperrberechtigte über CMP/SOAP Zertifikate sperren.

Der Prozess zur Zertifikatssperrung stellt die folgenden Anwendungsfälle (Use Cases) zur Verfügung:

Tabelle 48: Use Cases Zertifikatssperrung Komponenten PKI

Use Case	Beschreibung
UC-ZS-001	Zertifikate über Web-Anwendung sperren
UC-ZS-002	Zertifikate über SOAP/CMP sperren
UC-ZS-003	Zertifikate durch die gematik sperren

6.1 Rollen und Berechtigungen

Die nachfolgende Tabelle stellt die notwendigen Rollen und Berechtigungen für Zertifikatssperrungen dar:

Tabelle 49: Rollen und Berechtigungen Zertifikatssperrung

Rolle	Kürzel	Besitzer	Berechtigung
Sperrantragsteller	SAS	Hersteller, Anbieter, TSP	Durch gematik (Zulassungsmanagement) berechtigte Sperrantragsteller zur Sperrantragsstellung von X.509 Komponentenzertifikaten.
gematik Sperrberechtigter	GSB	gematik	Berechtigter Sperrantragsteller der gematik zur Sperrantragsstellung von X.509 Komponentenzertifikaten über alle Organisationen.
gematik Sperrverifikator	GSV	gematik	Prüfung und Freigabe der durch den gematik Sperrberechtigten eingestellten Sperranträge.
Anmerkung	Für die Rollen GSB und GSV gilt ein Rollenausschluss.		

6.2 Anwendung Zertifikatssperrung (I_Cert_ Revocation)

Die Ausführungen in dem nachfolgendem Anwendungsfall schildern die Aktion Sperrung von X.509-Komponentenzertifikaten für berechtigte Hersteller, Anbieter und TSP-X.509 nonQES (Antragsberechtigte) über das Webportal.

Tabelle 50 stellt die URIs der Webanwendung Zertifikatssperrung Komponenten-PKI für die Produktionsumgebung sowie für die Referenz- und Testumgebung dar. Die URI für die Referenz- und Testumgebung ist identisch.

Tabelle 50: URIs der Anwendung Zertifikatssperrung Komponenten-PKI

Umgebung	URI
PU	https://www.tms.ti-dienste.de/zas
RU/TU	https://www-testref.tms.ti-dienste.de/zas

Für die Zertifikatssperrung unter Verwendung der Webanwendung Zertifikatssperrung Komponenten-PKI durch die gematik (vgl. Kapitel 6.2.6) wird eine eigenständige URL zur Verfügung gestellt.

Tabelle 51 stellt die URIs der Webanwendung Zertifikatssperrung Komponenten-PKI durch die gematik für die Produktionsumgebung sowie für die Referenz- und Testumgebung dar. Die URI für die Referenz- und Testumgebung ist identisch.

Tabelle 51: URIs der Anwendung Zertifikatssperrung Komponenten-PKI (gematik)

Umgebung	URI
PU	https://www.tms.ti-dienste.de/zasGematik
RU/TU	https://www-testref.tms.ti-dienste.de/zasGematik

6.2.1 Zertifikatssperrung über Webanwendung

Tabelle 52: UC-ZS-001

Nummer: **Name:**	UC-ZS-001 Zertifikatssperrung über Web-Anwendung
Kurzbeschreibung	Dieser Anwendungsfall beschreibt, wie Antragsberechtigte über die Web-Anwendung "Zertifikatssperrung" (ZS) X.509-Zertifikate sperren können.
Auslösender Akteur	SAS
Vorbedingungen	Ein Berechtigter Zertifikatsantragsteller wurde in der Web-Anwendung Zulassungsmanagement erzeugt. Der Zulassungsstatus des Berechtigten Zertifikatsantragstellers sowie der zugehörigen Organisation

	ist aktiviert. Der Berechtigte Zertifikatsantragsteller besitzt die Berechtigung zum Sperren von Zertifikaten und den Benutzerkonto-Status „aktiv“. Dem Berechtigten Zertifikatsantragsteller wurde ein RSA-Token zur Verfügung gestellt. Der Akteur ist in der Applikation „Zertifikatssperrung“ angemeldet und autorisiert.
Eingangsdaten	Abhängig vom Zertifikatstyp variieren die Zertifikatsantrags-Daten des PKCS#10-Requests gemäß Tabelle 58: Zertifikatssperrdaten.
Ergebnisse	Es wurden ein oder mehrere X.509-Zertifikate gesperrt.
Anmerkungen	Ein Berechtigter Zertifikatsantragsteller kann nur durch seinen Antragsberechtigten bzw. seine Organisation erstellte Zertifikate sperren.

Tabelle 53: Prozessschritte UC-ZS-001

Nr.	Akteur	Prozessschritt
1.	SAS	Der Akteur ruft die Funktion "Zertifikat sperren" auf.
2.	Applikation	Die Applikation zeigt dem Akteur eine Liste der vom Antragsberechtigten beantragten und erhaltenen Zertifikate an, sowie deren Status.
3.	Applikation	Die Applikation ermöglicht auch die Angabe von Selektionskriterien zur Suche nach Zertifikaten. Selektionskriterien sind mindestens: • Seriennummer • Ausstellende X.509-CA • Gültigkeit o Von o Bis • FQDN bzw. ICCSN
4.	SAS	Ggf. erfasst der Akteur Selektionskriterien und löst die Funktion "Zertifikat suchen" aus.
5.	Applikation	Die Applikation listet die Zertifikate, die den Kriterien genügen, mit ihren wesentlichen Merkmalen auf. Zu

		diesen Merkmalen gehören mindestens: • Version • Seriennummer • Signaturalgorithmus • Signaturhashalgorithmus • Antragsteller (DN) • Gültigkeit o Von o Bis • Öffentlicher Schlüssel • Zertifikatsstatus
6.	SAS	Der Akteur selektiert ein oder mehrere Zertifikate und löst die Funktion "Sperren von Zertifikaten" aus.
7.	Applikation	Die Applikation prüft die Eingaben auf Vollständigkeit und Plausibilität. Fehlerfälle (Zertifikatssperrung über Eingabemaske): o Unvollständige oder fehlende Daten: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren.
8.	TMS	Das TMS erzeugt einen internen CMP-Zertifikatsrequest (Sperrung) und sendet diesen an die Komponenten-PKI.
9.	Komponenten-PKI	Die Komponenten-PKI nimmt den Request zur Sperrung entgegen und sperrt das Zertifikat im OCSP-Responder der Komponenten-PKI. Die CA erzeugt eine interne CMP-Response und sendet diese an das TMS. Fallunterscheidung: • Sperrung eines VPN-Zugangsdienst-Zertifikats: Aufnahme der Sperrinformationen in die CRL für VPN-Zugangsdienst-Zertifikate. • Sperrung sonstiger Komponentenzertifikate: Aufnahme der Sperrinformation in OCSP-Responder der Komponenten-CA
10.	TMS	Das TMS nimmt die CMP-Response der Komponenten-PKI-CA entgegen und protokolliert den Vorgang.

11.	Applikation	Die Applikation meldet das erfolgreiche Sperren des Zertifikats und kehrt zur aufrufenden Funktion zurück.
12.	TMS	Das TMS informiert den Antragsberechtigten (Organisation) zusätzlich über die Sperrung des Zertifikats per E-Mail.

6.2.2 Prozessdarstellung Zertifikatssperrung über Webanwendung

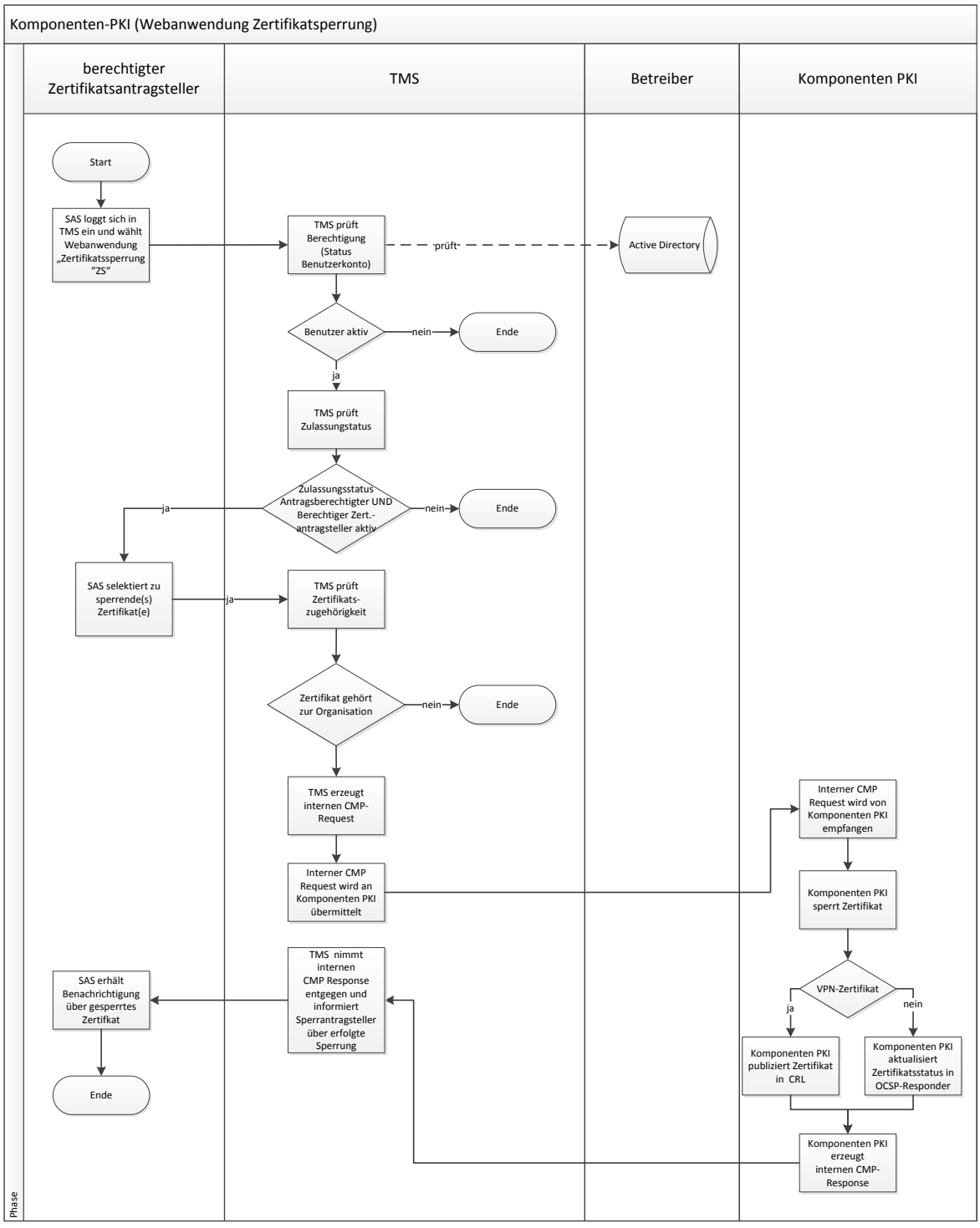

Abbildung 9: Zertifikatssperrung über Webanwendung

6.2.3 Zertifikatssperrung über SOAP/CMP

Die Ausführungen in dem nachfolgenden Anwendungsfall schildern die Aktion Sperrung von X.509-Komponentenzertifikat für berechtigter Hersteller, Anbieter und TSP-X.509 nonQES (Antragsberechtigte) über die Schnittstellen SOAP/CMP.

Tabelle 54: UC-ZS-002

Nummer: **Name:**	UC-ZS-002 Zertifikatssperrung über SOAP/CMP
Kurzbeschreibung	Dieser Anwendungsfall beschreibt, wie Antragsberechtigte über SOAP/CMP für sie ausgestellte X.509-Zertifikate (für sie ausgestellte) sperren können.
Auslösender Akteur	SAS (technische Komponente)
Vorbedingungen	Ein Berechtigter Zertifikatsantragsteller wurde durch die gematik im Zulassungsmanagement angelegt. Der Zulassungsstatus des Berechtigten Zertifikatsantragstellers sowie der zugehörigen Organisation ist aktiviert. Der Berechtigte Zertifikatsantragsteller besitzt die Berechtigung zum Sperren von Zertifikaten und ein Benutzerkonto mit dem Status „aktiviert“. Der Benutzer hat über die Web-Anwendung „CryptID ausstellen“ eine CryptID bezogen.
Eingangsdaten	Abhängig vom Zertifikatstyp variieren die Zertifikatssperrantrags-Daten gemäß Tabelle 58: Zertifikatssperrdaten.
Ergebnisse	Es wurden ein oder mehrere X.509-Zertifikate gesperrt.
Anmerkungen	Ein Berechtigter Zertifikatsantragsteller kann nur durch seinen Antragsberechtigten bzw. seine Organisation erstellte Zertifikate sperren.

Tabelle 55: Prozessschritte UC-ZS-002

Nr.	Akteur	Prozessschritt
1.	SAS	Eine Komponente des Berechtigten Zertifikatsantragstellers erzeugt einen SOAP/CMP-Request und übermittelt diese an das TMS zur Sperrung eines oder mehrerer Zertifikate.
2.	TMS	Das TMS nimmt den SOAP/CMP-Request entgegen und überprüft diesen. Fehlerfall:

		• Die SOAP/CMP-Authentifizierung schlägt fehl: Es wird ein entsprechender SOAP/CMP-Fault erzeugt und an die aufrufende Komponente zurückgegeben. Der Prozess wird abgebrochen. • Es ist ein syntaktischer und / oder semantischer Fehler aufgetreten: Es wird ein entsprechender SOAP/CMP-Fault erzeugt und an die aufrufende Komponente zurückgegeben. Der Prozess wird abgebrochen. • Der Benutzer besitzt nicht die Berechtigung zur Sperrung eines Zertifikats vom angefragten Zertifikatstyp oder das Zertifikat ist nicht der Organisation des Berechtigten Antragstellers zugeordnet: Es wird ein entsprechender SOAP/CMP-Fault erzeugt und an die aufrufende Komponente zurückgegeben. Der Prozess wird abgebrochen.
3.	TMS	Das TMS extrahiert die erforderlichen Daten aus dem SOAP/CMP-Request, erzeugt einen internen CMP-Request (Sperrung) und sendet diesen an die Komponenten-PKI.
4.	Komponenten-PKI	Die Komponenten-PKI nimmt den Sperrantrag entgegen und sperrt das Zertifikat im OCSP-Responder der Komponenten-PKI-CA. Die CA erzeugt eine interne CMP-Response und sendet diese an die Applikation. Fallunterscheidung: • Sperrung eines VPN-Zugangsdienst-Zertifikats: Aufnahme der Sperrinformationen in die CRL für VPN-Zugangsdienst-Zertifikate. • Sperrung aller sonstige Komponenten-Zertifikate: Aufnahme der Sperrinformationen in OCSP-Responder der Komponenten-PKI
5.	TMS	Das TMS nimmt die interne CMP-Response der Komponenten-PKI entgegen, konvertiert diese in eine externe SOAP/CMP Response und sendet diese an den Berechtigten Zertifikatsantragsteller zurück.
6.	SAS	Die Komponente des Berechtigten Zertifikatsantragsteller nimmt die SOAP/CMP-Response entgegen.
7.	TMS	TMS informiert den Antragsberechtigten (Organisation) über die Sperrung des Zertifikats per E-Mail.

6.2.4 Prozessdarstellung Zertifikatssperrung über SOAP/CMP

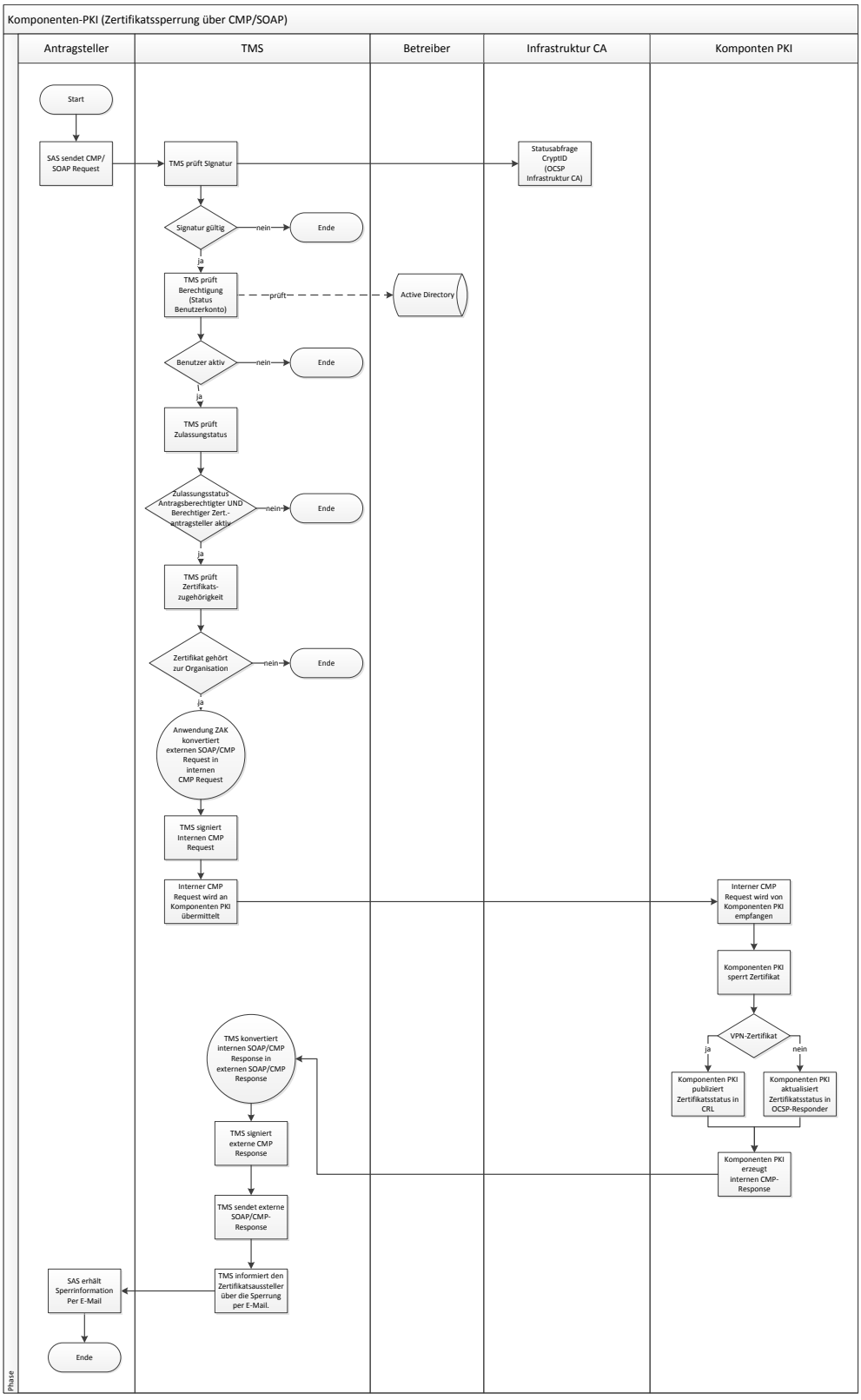

Abbildung 10: Zertifikatssperrung über SOAP/CMP

6.2.5 Zertifikatssperrung über organisatorische Schnittstelle

In der Regel werden X.509-Komponenten-Zertifikate über das Webportal (siehe Kapitel 6.2.1) oder unter Verwendung der Protokolle CMP/SOAP (siehe Kapitel 6.2.4) vom Sperrantragsteller (SAS) gesperrt. Darüber hinaus können Sperrantragsteller Sperranträge über eine organisatorische Schnittstelle an den Betreiber der Komponenten-PKI übermitteln. Der Betreiber der Komponenten-PKI nimmt nur Sperranträge von berechtigten Zertifikatsantragstellern entgegen, die von der gematik über das Zulassungsmanagement freigegeben und an der Betreiber übermittelt wurden.

6.2.6 Zertifikatssperrung durch die gematik

Die Ausführungen in dem nachfolgenden Anwendungsfall schildern die Aktion Sperrung von X.509-Komponentenzertifikaten durch die gematik. Die Aktion ist nur über das Webportal möglich.

Tabelle 56: UC-ZS-003

Nummer: **Name:**	UC-ZS-003 Zertifikatssperrung durch gematik
Kurzbeschreibung	Dieser Anwendungsfall beschreibt, wie zwei gematik Rollen (GSB und GSV) über die Web-Anwendung "Zertifikatssperrung" Zertifikate der Antragsberechtigten sperren können.
Auslösender Akteur	GSB
Vorbedingungen	Der erste Akteur ist ein berechtigter Mitarbeiter der gematik und besitzt die Rolle „GSB“. Der zweite Akteur ist ein berechtigter Mitarbeiter der gematik und besitzt die Rolle „GSV“. Den berechtigten Mitarbeitern wurden je ein RSA-Token und eine QES-Karte zur Verfügung gestellt. Der erste Akteur ist in der Applikation „Zertifikatssperrung“ und im QSC angemeldet und autorisiert. Der zweite Akteur ist im QSC angemeldet und autorisiert.
Eingangsdaten	Abhängig vom Zertifikatstyp variieren die Zertifikatssperrantrags-Daten gemäß Tabelle 58: Zertifikatssperrdaten.
Ergebnisse	Es wurde(n) ein oder mehrere X.509-Zertifikate gesperrt.
Anmerkungen	

Tabelle 57: Prozessschritte UC-ZS-003

Nr.	Akteur	Prozessschritt
1.	GSB	Der Akteur ruft die Funktion "Zertifikat sperren" auf.
2.	Applikation	Die Applikation zeigt dem Akteur eine Liste der von den Antragsberechtigten beantragten und erhaltenen Zertifikate an.
3.	Applikation	Die Applikation ermöglicht auch die Angabe von Selektionskriterien zur Suche nach Zertifikaten. Selektionskriterien sind mindestens: • Name der Organisation • ID der Organisation • Ausstellende X.509-CA • FQDN bzw. ICCSN • Seriennummer • Gültigkeit o Von o Bis
4.	GSB	Ggf. erfasst der Akteur Selektionskriterien und löst die Funktion "Zertifikat suchen" aus.
5.	Applikation	Die Applikation listet die Zertifikate, die den Kriterien genügen, mit ihren wesentlichen Merkmalen auf. Zu diesen Merkmalen gehören mindestens: • Version • Seriennummer • Signaturalgorithmus • Signaturhashalgorithmus • Antragsteller (DN) • Gültigkeit o Von o Bis • Öffentlicher Schlüssel • Zertifikatsstatus
6.	GSB	Der Akteur selektiert ein oder mehrere Zertifikate und löst die Funktion "Sperren von Zertifikaten" aus.

7.	Applikation	Die Applikation prüft die Eingaben auf Vollständigkeit und Plausibilität. Fehlerfälle (Zertifikatssperrung über Eingabemaske): o Unvollständige oder fehlende Daten: Die Applikation kehrt zur Eingabemaske zurück und zeigt die festgestellten Plausibilitätsverletzungen an z. B. durch eine farbliche Kennzeichnung, an. Der Akteur kann diese korrigieren.
8.	Applikation	Der QSC fordert den Akteur zur Bestätigung der Freigabe auf.
9.	GSB	Der Akteur bestätigt die Erfassung und signiert die erfassten Daten.
10.	TMS	Das TMS prüft die mit dem QSC erzeugte Signatur. Fehlerfälle: a) Fehlerhafte Signatur: Das TMS stellt die fehlgeschlagene Signaturprüfung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles. b) Nicht zugelassene Signatur: Das TMS stellt die fehlende Berechtigung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles.
11.	TMS	Das TMS speichert die erfassten Daten in der TMS Datenbank, protokolliert den Vorgang und informiert die Rolleninhaber „GSV“ über den eingestellten Sperrantrag.
12.	GSV	Der Akteur ruft den QSC auf.
13.	QSC	Der QSC zeigt dem Akteur eine Liste der Sperranträge an.
14.	ZMV	Der Akteur selektiert einen oder mehrere Sperranträge.
15.	QSC	Der QSC stellt die ausgewählten Sperranträge zur Verifikation dar.
16.	ZMV	Fallunterscheidung: c) Der Akteur gibt den Sperrantrag frei: Weiter mit Schritt 17. d) Der Akteur widerspricht der Freigabe: Dem Akteur

		wird eine Eingabemaske mit einem Eingabefeld zum Erfassen einer Begründung dargestellt. Der Akteur erfasst die Begründung. Die Applikation erzeugt eine E-Mail mit der Begründung sowie dem Hinweis, dass der Sperrantrag abgelehnt wurde und sendet diese an alle Benutzer mit der Rolle "GSB". Der zur Verifikation vorliegende Sperrantrag wird vollständig verworfen. Der Vorgang wird protokolliert. Der Use Case ist beendet.
17.	QSC	Der QSC fordert den Akteur zur Bestätigung der Freigabe auf.
18.	ZMA	Der Akteur bestätigt den Sperrantrag und signiert die zu bestätigenden Daten.
19.	TMS	Das TMS prüft die mit dem QSC erzeugte Signatur. Fehlerfälle: a) Fehlerhafte Signatur: Die Applikation stellt die fehlgeschlagene Signaturprüfung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles. b) Nicht zugelassene Signatur: Die Applikation stellt die fehlende Berechtigung durch eine eindeutige und aussagekräftige Meldung dar. Die Applikation ermöglicht die Neuerstellung der Signatur oder den Abbruch des Anwendungsfalles.
20.	TMS	Das TMS speichert den freigegebenen Sperrantrag in der TMS Datenbank und protokolliert den Vorgang.
21.	QSC	Der QSC meldet die erfolgreiche Freigabe und kehrt zur aufrufenden Funktion zurück
22.	TMS	Das TMS erzeugt einen internen CMP-Zertifikatsrequest (Sperrung) und sendet diesen an die Komponenten-PKI.
23.	Komponenten-PKI	Die Komponenten-PKI nimmt den Request zur Sperrung entgegen und sperrt das Zertifikat im OCSP-Responder der Komponenten-PKI-CA. Die CA erzeugt eine interne CMP-Response und sendet diese an die Applikation. Fallunterscheidung: • Sperrung eines VPN-Zugangsdienst-Zertifikats: Aufnahme der Sperrinformationen in die CRL für VPN-Zugangsdienst-Zertifikate. • Sperrung sonstiger Komponentenzertifikate:

		Aufnahme der Sperrinformation in OCSP-Responder der Komponenten-CA.
24.	TMS	Das TMS nimmt die CMP-Response der Komponenten-PKI-CA entgegen und protokolliert den Vorgang.
25.	Applikation	Die Applikation meldet das erfolgreiche Sperren des Zertifikats / der Zertifikate und kehrt zur aufrufenden Funktion zurück.
26.	TMS	Das TMS informiert den Antragsberechtigten (Organisation) sowie Rolleninhaber „GSB“ und „GSV“ über die Sperrung des Zertifikats bzw. der Zertifikate per E-Mail.

6.2.7 Prozessdarstellung Zertifikatssperrung durch gematik

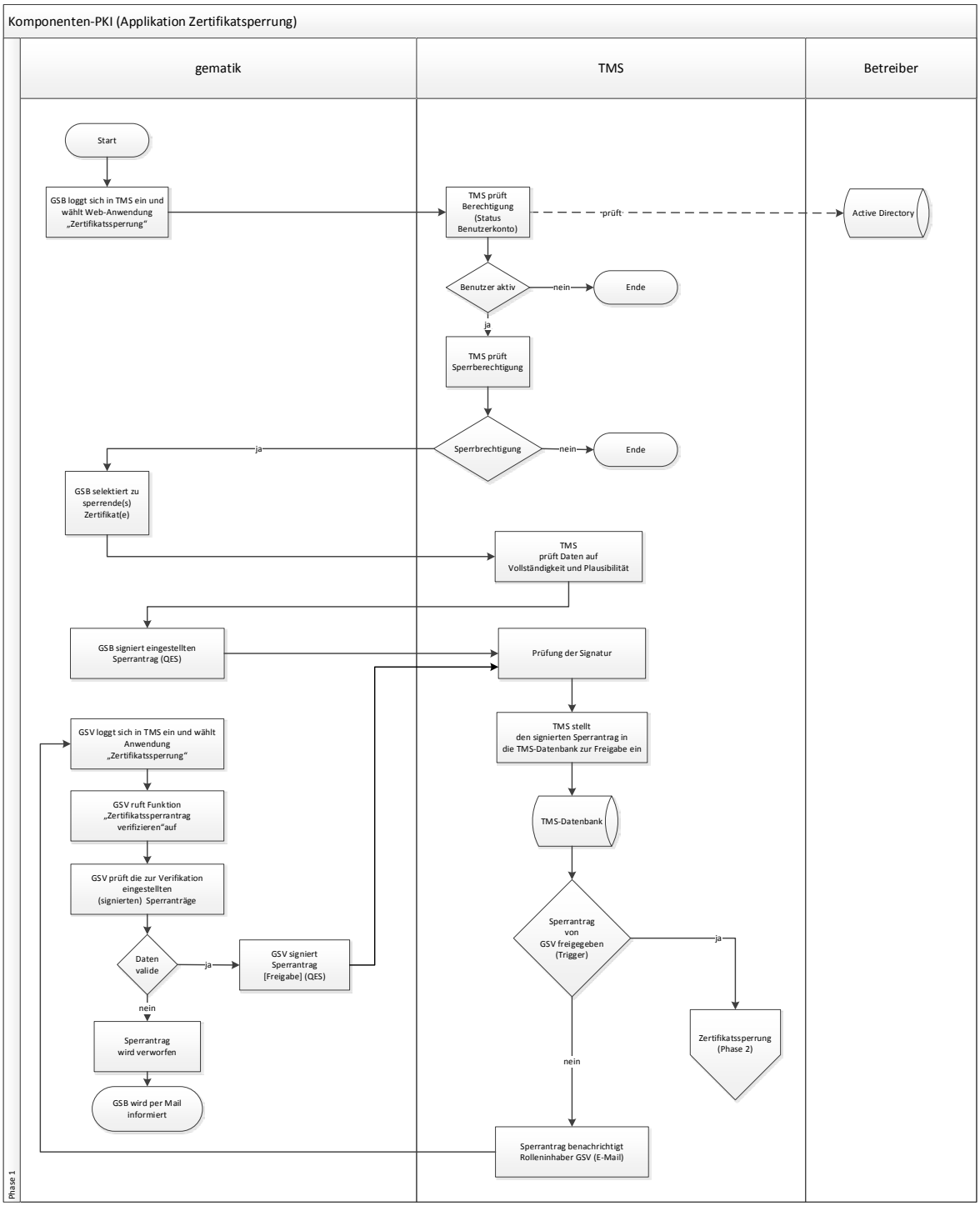

Abbildung 11: Zertifikatssperrung durch gematik - Phase 1: Sperrantragstellung

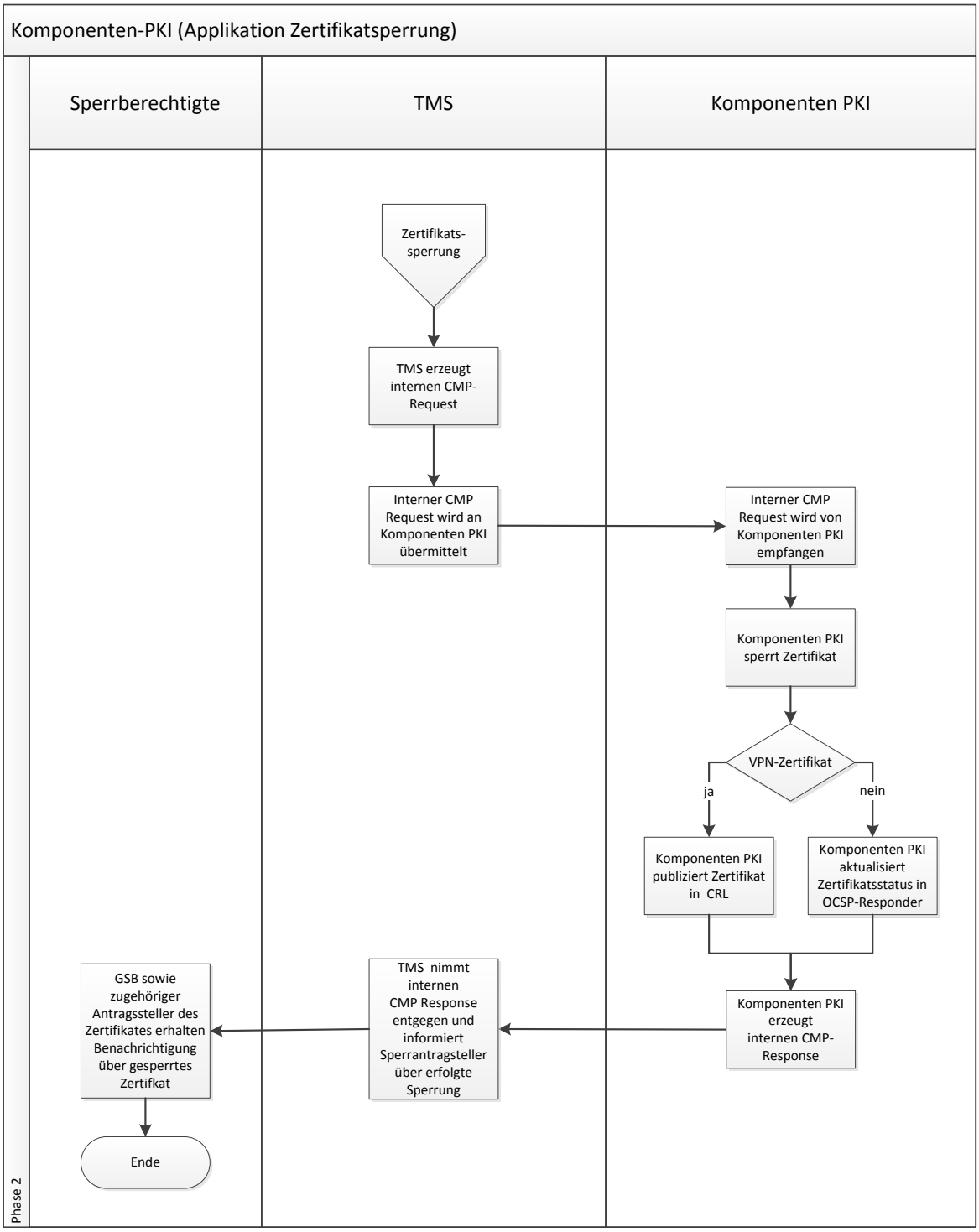

Abbildung 12: Zertifikatssperrung durch gematik - Phase 2: Zertifikatssperrung

6.3 Artefakte

6.3.1 Eingangsdaten für Webanwendung

Die folgende Tabelle führt die in den Use Cases zur Zertifikatssperrung notwendigen Eingabedaten auf:

Tabelle 58: Zertifikatssperrdaten

Zertifikatssperrantragsdaten	Kurzbeschreibung
Seriennummer	Seriennummer des zu sperrenden X.509-Zertifikats
CA	Ausstellende CA des zu sperrenden X.509-Zerifikats
Organisations-Name des Antragstellers	Name der zugehörigen Organisation, der das Zertifikat zugeordnet ist
Sperrgrund	Grund der Sperrung
Common Name	FQDN / ISSCN / Vollständiger Name (Abhängig vom Zertifikatstyp gemäß Tabelle 45)

6.3.2 CMP-Request und CMP-Response (X.509)

Die CMP-Schnittstelle zum Sperren von X.509-Komponenten-Zertifikaten wird über HTTP(S) angeboten.

Während die CMP-Requests mit einem CMP-Requestor signiert werden, werden die CMP-Responses – abgesehen von CMP-Error-Responses - mit einem CMP-Responder signiert.

Gemäß [RFC4210] sind die CMP-Requests und –Responses DER-kodiert im Body des POST-Befehls zu übermitteln. Der Content-Type für die Requests und Responses ist application/pkixcmp.

Das nachfolgende Listing stellt die Struktur der CMP-Nachricht zur Sperrung von X.509-Komponenten-Zertifikaten über die CMP-Schnittstelle dar:

```
PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection [0] EXPLICIT PKIProtection,                    -- Signature-Variante mit
                                                              SHA1/224/256/384/512withRSA/DSA/ECDSA/RSAandMGF1
                                                              ist erforderlich (abhängig von der Konfiguration,
                                                              aktuell: SHA256withRSA)
}

PKIHeader ::= SEQUENCE {
    pvno              INTEGER cmp2000(2),                 -- fix
    sender            GeneralName,                        -- directoryName des CMP-Requesters
    recipient         GeneralName,                        -- directoryName des CMP-Responders
    messageTime   [0] EXPLICIT GeneralizedTime,           -- Timestamp der Nachricht, erfoderlich (OPTIONAL nach RFC
4210)
    protectionAlg [1] EXPLICIT AlgorithmIdentifier,       -- Signaturalgorithm für PKIMessage.protection, hier
erforderlich
    transactionID [4] EXPLICIT OCTET STRING,              -- erforderlich (OPTIONAL nach RFC 4210)
    generalInfo   [8] EXPLICIT SEQUENCE SIZE (1..MAX) OF InfoTypeAndValue OPTIONAL
}

PKIBody ::= CHOICE
{
   ...
   rr [11] EXPLICIT RevReqContent                    -- Nur diese Variante ist erlaubt
   ..
```

```
}

RevReqContent ::= SEQUENCE OF RevDetails

RevDetails ::= SEQUENCE {
    certDetails     CertTemplate,
    crlEntryDetails Extensions OPTIONAL              -- nur Extension CRLReasons [RFC5280§5.3.1] wird unterstützt
                                                        (wenn nicht gesetzt, wird vom System “unspecified (0)”
                                                        verwendet)
}

 CRLReason ::= ENUMERATED {
        unspecified             (0),
        keyCompromise           (1),
        cACompromise            (2),                 -- nicht zugelassen/wird nicht genutzt
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        certificateHold         (6),
             -- value 7 is not used
        removeFromCRL           (8),                 -- nicht zugelassen/wird nicht genutzt
        privilegeWithdrawn      (9),
        aACompromise           (10) }                -- nicht zugelassen/wird nicht genutzt


CertTemplate ::= SEQUENCE {                          -- vgl. Tab_PKI_517 und Tab_PKI_518
    serialNumber    [1] EXPLICIT INTEGER,
    issuer          [3] EXPLICIT Name,
    subject         [5] EXPLICIT Name OPTIONAL

}
```

Listing 19: CMP-Request - Zertifikatsperrung (X.509)

Der CMP-Responder prüft den CMP-Request. Im negativen Fall antwortet der Responder mit einer CMP-Response in der Variante "error [23] ErrorMsgContent" im PKIBody.

Andererseits antwortet der Responder mit einer Nachricht auf Basis der im nachfolgenden Listing dargestellten Struktur:

```
PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection [0] EXPLICIT PKIProtection,                -- Signature-Variante mit
                                                             SHA1/224/256/384/512withRSA/DSA/ECDSA/RSAandMGF1
                                                             ist erforderlich (abhängig von der Konfiguration, aktuell:
                                                             SHA256withRSA)
}

PKIHeader ::= SEQUENCE {
    pvno              INTEGER cmp2000(2),             -- fix
    sender            GeneralName,                    -- directoryName des CMP-Responders
    recipient         GeneralName,                    -- directoryName des CMP-Requesters
    messageTime   [0] EXPLICIT GeneralizedTime OPTIONAL,  -- Timestamp der Nachricht
    protectionAlg [1] EXPLICIT AlgorithmIdentifier,   -- Signaturalgorithmus für PKIMessage.protection erforderlich
    transactionID [4] EXPLICIT OCTET STRING           -- transactionID aus dem Request oder neu generiert,
                                                         falls nicht vorhanden
}

PKIBody :: CHOICE {
   ...
   rp [12] EXPLICIT RevRepContent                     -- Nur dieser Variant ist erlaubt
   ...
}

RevRepContent ::= SEQUENCE {
    status          SEQUENCE OF PKIStatusInfo,
    revCerts    [0] SEQUENCE OF CertId OPTIONAL       -- Erforderlich, wenn mindestens ein Zertifikat erfolgreich
                                                         zurückgezogen wurde.

}


PKIStatusInfo ::= SEQUENCE {
         status        PKIStatus,
         statusString  PKIFreeText     OPTIONAL,     -- wird nur bei konkreten Fehlermeldungen genutzt
         failInfo      PKIFailureInfo  OPTIONAL      -- wird im Fehlerfall immer genutzt
}

PKIStatus ::= INTEGER {
         accepted               (0),
         grantedWithMods        (1),                 -- wird nicht unterstützt
         rejection              (2),
         waiting                (3),                 -- wird nicht unterstützt
         revocationWarning      (4),                 -- wird nicht unterstützt
         revocationNotification (5),                 -- wird nicht unterstützt
         keyUpdateWarning       (6)                  -- wird nicht unterstützt
}

CertId ::= SEQUENCE {
          issuer           GeneralName,
          serialNumber     INTEGER
```

```
}
```

Listing 20: CMP-Response – Zertifikatsperrung (X.509)

```
  0 871: SEQUENCE {
  4 251:   SEQUENCE {
  7   1:     INTEGER 2
 10  91:     [4] {
 12  89:       SEQUENCE {
 14  10:         SET {
 16   8:           SEQUENCE {
 18   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
 23   1:             PrintableString '4'
       :             }
       :           }
 26  30:         SET {
 28  28:           SEQUENCE {
 30   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 35  21:             UTF8String 'OLGA-777777 TEST-ONLY'
       :             }
       :           }
 58  30:         SET {
 60  28:           SEQUENCE {
 62   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 67  21:             UTF8String 'OLGA-777777 NOT-VALID'
       :             }
       :           }
 90  11:         SET {
 92   9:           SEQUENCE {
 94   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
 99   2:             PrintableString 'DE'
       :             }
       :           }
       :         }
       :       }
103  89:     [4] {
105  87:       SEQUENCE {
107  32:         SET {
109  30:           SEQUENCE {
111   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
116  23:             UTF8String 'TMS CryptID 1 TEST-ONLY'
       :             }
       :           }
141  38:         SET {
143  36:           SEQUENCE {
145   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
150  29:             UTF8String 'arvato Systems GmbH NOT-VALID'
       :             }
       :           }
181  11:         SET {
183   9:           SEQUENCE {
185   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
190   2:             PrintableString 'DE'
       :             }
       :           }
       :         }
       :       }
194  17:     [0] {
196  15:       GeneralizedTime 07/01/2015 09:26:36 GMT
       :       }
213  13:     [1] {
215  11:       SEQUENCE {
217   9:         OBJECT IDENTIFIER
       :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
       :         }
       :       }
228  10:     [4] {
230   8:       OCTET STRING 1B F4 94 8E B8 0D 4A 44
       :       }
240  16:     [8] {
242  14:       SEQUENCE {
244  12:         SEQUENCE {
246   8:           OBJECT IDENTIFIER implicitConfirm (1 3 6 1 5 5 7 4 13)
256   0:           NULL
       :           }
       :         }
       :       }
       :     }
258 348:   [11] {
262 344:     SEQUENCE {
266 340:       SEQUENCE {
270 322:         SEQUENCE {
274   3:           [1] 00 F3 CF
279 134:           [3] {
282 131:             SEQUENCE {
285  31:               SET {
287  29:                 SEQUENCE {
289   3:                   OBJECT IDENTIFIER commonName (2 5 4 3)
294  22:                   UTF8String 'GEM.KOMP-CA1 TEST-ONLY'
       :                   }
       :                 }
318  50:               SET {
320  48:                 SEQUENCE {
322   3:                   OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
327  41:                   UTF8String
       :                   'Komponenten-CA der Telematikinfrastruktur'
       :                   }
```

```
      :                 }
370  31:               SET {
372  29:                 SEQUENCE {
374   3:                   OBJECT IDENTIFIER organizationName (2 5 4 10)
379  22:                   UTF8String 'gematik GmbH NOT-VALID'
      :                   }
      :                 }
403  11:               SET {
405   9:                 SEQUENCE {
407   3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
412   2:                   PrintableString 'DE'
      :                   }
      :                 }
      :               }
      :             }
416 177:           [5] {
419 174:             SEQUENCE {
422  24:               SET {
424  22:                 SEQUENCE {
426   3:                   OBJECT IDENTIFIER streetAddress (2 5 4 9)
431  15:                   UTF8String 'Hauptstra..e 15'
      :                   }
      :                 }
448  14:               SET {
450  12:                 SEQUENCE {
452   3:                   OBJECT IDENTIFIER postalCode (2 5 4 17)
457   5:                   UTF8String '44458'
      :                   }
      :                 }
464  38:               SET {
466  36:                 SEQUENCE {
468   3:                   OBJECT IDENTIFIER commonName (2 5 4 3)
473  29:                   UTF8String '8027600011223346666 9-20150107'
      :                   }
      :                 }
504  42:               SET {
506  40:                 SEQUENCE {
508   3:                   OBJECT IDENTIFIER organizationName (2 5 4 10)
513  33:                   UTF8String 'OLGA-777777 TEST-ONLY - NOT-VALID'
      :                   }
      :                 }
548  16:               SET {
550  14:                 SEQUENCE {
552   3:                   OBJECT IDENTIFIER localityName (2 5 4 7)
557   7:                   UTF8String 'Testort'
      :                   }
      :                 }
566  15:               SET {
568  13:                 SEQUENCE {
570   3:                   OBJECT IDENTIFIER stateOrProvinceName (2 5 4 8)
575   6:                   UTF8String 'Bayern'
      :                   }
      :                 }
583  11:               SET {
585   9:                 SEQUENCE {
587   3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
592   2:                   PrintableString 'DE'
      :                   }
      :                 }
      :               }
      :             }
      :           }
596  12:         SEQUENCE {
598  10:           SEQUENCE {
600   3:             OBJECT IDENTIFIER cRLReason (2 5 29 21)
605   3:             OCTET STRING, encapsulates {
607   1:               ENUMERATED 1
      :               }
      :             }
      :           }
      :         }
      :       }
      :     }
610 261:   [0] {
614 257:     BIT STRING
      :     }
      :   }
```

Listing 21: Beispiel-CMP-Request zur Sperrung eines X.509-Zertifikats

```
  0 578: SEQUENCE {
  4 146:   SEQUENCE {
  7   1:     INTEGER 2
 10  19:     [4] {
 12  17:       SEQUENCE {
 14  15:         SET {
 16  13:           SEQUENCE {
 18   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 23   6:             UTF8String 'INTCA1'
      :             }
      :           }
      :         }
      :       }
 31  91:     [4] {
```

```
 33  89:       SEQUENCE {
 35  10:         SET {
 37   8:           SEQUENCE {
 39   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
 44   1:             PrintableString '4'
      :             }
      :           }
 47  30:         SET {
 49  28:           SEQUENCE {
 51   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 56  21:             UTF8String 'OLGA-777777 TEST-ONLY'
      :             }
      :           }
 79  30:         SET {
 81  28:           SEQUENCE {
 83   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 88  21:             UTF8String 'OLGA-777777 NOT-VALID'
      :             }
      :           }
111  11:         SET {
113   9:           SEQUENCE {
115   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
120   2:             PrintableString 'DE'
      :             }
      :           }
      :         }
      :       }
124  15:     [1] {
126  13:       SEQUENCE {
128   9:         OBJECT IDENTIFIER
      :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
139   0:         NULL
      :         }
      :       }
141  10:     [4] {
143   8:       OCTET STRING 1B F4 94 8E B8 0D 4A 44
      :       }
      :     }
153 161:   [12] {
156 158:     SEQUENCE {
159   5:       SEQUENCE {
161   3:         SEQUENCE {
163   1:           INTEGER 0
      :           }
      :         }
166 148:       [0] {
169 145:         SEQUENCE {
172 142:           SEQUENCE {
175 134:             [4] {
178 131:               SEQUENCE {
181  31:                 SET {
183  29:                   SEQUENCE {
185   3:                     OBJECT IDENTIFIER commonName (2 5 4 3)
190  22:                     UTF8String 'GEM.KOMP-CA1 TEST-ONLY'
      :                     }
      :                   }
214  50:                 SET {
216  48:                   SEQUENCE {
218   3:                     OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
223  41:                     UTF8String
      :                     'Komponenten-CA der Telematikinfrastruktur'
      :                     }
      :                   }
266  31:                 SET {
268  29:                   SEQUENCE {
270   3:                     OBJECT IDENTIFIER organizationName (2 5 4 10)
275  22:                     UTF8String 'gematik GmbH NOT-VALID'
      :                     }
      :                   }
299  11:                 SET {
301   9:                   SEQUENCE {
303   3:                     OBJECT IDENTIFIER countryName (2 5 4 6)
308   2:                     PrintableString 'DE'
      :                     }
      :                   }
      :                 }
      :               }
312   3:             INTEGER 62415
      :             }
      :           }
      :         }
      :       }
      :     }
317 261:   [0] {
321 257:     BIT STRING
      :     }
      :   }
```

Listing 22: Beispiel-CMP-Response zur Sperrung eines X.509-Zertifikats

```
  0 578: SEQUENCE {
  4 146:   SEQUENCE {
  7   1:     INTEGER 2
 10  19:     [4] {
```

```
 12  17:       SEQUENCE {
 14  15:         SET {
 16  13:           SEQUENCE {
 18   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 23   6:             UTF8String 'INTCA1'
      :             }
      :           }
      :         }
      :       }
 31  91:     [4] {
 33  89:       SEQUENCE {
 35  10:         SET {
 37   8:           SEQUENCE {
 39   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
 44   1:             PrintableString '5'
      :             }
      :           }
 47  30:         SET {
 49  28:           SEQUENCE {
 51   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 56  21:             UTF8String 'OLGA-777777 TEST-ONLY'
      :             }
      :           }
 79  30:         SET {
 81  28:           SEQUENCE {
 83   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 88  21:             UTF8String 'OLGA-777777 NOT-VALID'
      :             }
      :           }
111  11:         SET {
113   9:           SEQUENCE {
115   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
120   2:             PrintableString 'DE'
      :             }
      :           }
      :         }
      :       }
124  15:     [1] {
126  13:       SEQUENCE {
128   9:         OBJECT IDENTIFIER
      :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
139   0:         NULL
      :         }
      :       }
141  10:     [4] {
143   8:       OCTET STRING 24 0D E9 8F E2 0C 7E 60
      :       }
      :     }
153 161:   [12] {
156 158:     SEQUENCE {
159   5:       SEQUENCE {
161   3:         SEQUENCE {
163   1:           INTEGER 2
      :           }
      :         }
166 148:       [0] {
169 145:         SEQUENCE {
172 142:           SEQUENCE {
175 134:             [4] {
178 131:               SEQUENCE {
181  31:                 SET {
183  29:                   SEQUENCE {
185   3:                     OBJECT IDENTIFIER commonName (2 5 4 3)
190  22:                     UTF8String 'GEM.KOMP-CA1 TEST-ONLY'
      :                     }
      :                   }
214  50:                 SET {
216  48:                   SEQUENCE {
218   3:                     OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
223  41:                     UTF8String
      :                     'Komponenten-CA der Telematikinfrastruktur'
      :                     }
      :                   }
266  31:                 SET {
268  29:                   SEQUENCE {
270   3:                     OBJECT IDENTIFIER organizationName (2 5 4 10)
275  22:                     UTF8String 'gematik GmbH NOT-VALID'
      :                     }
      :                   }
299  11:                 SET {
301   9:                   SEQUENCE {
303   3:                     OBJECT IDENTIFIER countryName (2 5 4 6)
308   2:                     PrintableString 'DE'
      :                     }
      :                   }
      :                 }
      :               }
312   3:             INTEGER 68835
      :             }
      :           }
      :         }
      :       }
      :     }
317 261:   [0] {
321 257:     BIT STRING
      :     }
      :   }
```

Listing 23: Beispiel-CMP-Response Zertifikatssperrung X.509 Fehlerfall Zertifikat bereits gesperrt

```
  0 578: SEQUENCE {
  4 146:   SEQUENCE {
  7   1:     INTEGER 2
 10  19:     [4] {
 12  17:       SEQUENCE {
 14  15:         SET {
 16  13:           SEQUENCE {
 18   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 23   6:             UTF8String 'INTCA1'
      :             }
      :           }
      :         }
      :       }
 31  91:     [4] {
 33  89:       SEQUENCE {
 35  10:         SET {
 37   8:           SEQUENCE {
 39   3:             OBJECT IDENTIFIER serialNumber (2 5 4 5)
 44   1:             PrintableString '5'
      :             }
      :           }
 47  30:         SET {
 49  28:           SEQUENCE {
 51   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
 56  21:             UTF8String 'OLGA-777777 TEST-ONLY'
      :             }
      :           }
 79  30:         SET {
 81  28:           SEQUENCE {
 83   3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
 88  21:             UTF8String 'OLGA-777777 NOT-VALID'
      :             }
      :           }
111  11:         SET {
113   9:           SEQUENCE {
115   3:             OBJECT IDENTIFIER countryName (2 5 4 6)
120   2:             PrintableString 'DE'
      :             }
      :           }
      :         }
      :       }
124  15:     [1] {
126  13:       SEQUENCE {
128   9:         OBJECT IDENTIFIER
      :           sha256WithRSAEncryption (1 2 840 113549 1 1 11)
139   0:         NULL
      :         }
      :       }
141  10:     [4] {
143   8:       OCTET STRING 4F 80 06 C7 F0 44 C6 A6
      :       }
      :     }
153 161:   [12] {
156 158:     SEQUENCE {
159   5:       SEQUENCE {
161   3:         SEQUENCE {
163   1:           INTEGER 2
      :           }
      :         }
166 148:       [0] {
169 145:         SEQUENCE {
172 142:           SEQUENCE {
175 134:             [4] {
178 131:               SEQUENCE {
181  31:                 SET {
183  29:                   SEQUENCE {
185   3:                     OBJECT IDENTIFIER commonName (2 5 4 3)
190  22:                     UTF8String 'GEM.KOMP-CA1 TEST-ONLY'
      :                     }
      :                   }
214  50:                 SET {
216  48:                   SEQUENCE {
218   3:                     OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
223  41:                     UTF8String
      :                     'Komponenten-CA der Telematikinfrastruktur'
      :                     }
      :                   }
266  31:                 SET {
268  29:                   SEQUENCE {
270   3:                     OBJECT IDENTIFIER organizationName (2 5 4 10)
275  22:                     UTF8String 'gematik GmbH NOT-VALID'
      :                     }
      :                   }
299  11:                 SET {
301   9:                   SEQUENCE {
303   3:                     OBJECT IDENTIFIER countryName (2 5 4 6)
308   2:                     PrintableString 'DE'
      :                     }
      :                   }
      :                 }
```

```
:                 }
312   3:            INTEGER 99999
:               }
:             }
:           }
:         }
:       }
317 261:  [0] {
321 257:    BIT STRING
:         }
:       }
```

Listing 24: Beispiel-CMP-Response Zertifikatssperrung X.509 Fehlerfall Zertifikat in CA nicht bekannt

6.3.3 SOAP-Request und SOAP Response

Die SOAP-Schnittstelle zum Sperren von X.509-Komponenten-Zertifikaten wird unter Verwendung des Standards Webservice-Security realisiert und über HTTP(S) angeboten.

Alle SOAP-Requests müssen dabei mit der Crypt-ID signiert werden, wobei folgende Objekte zu signieren sind (falls vorhanden):

- {Element}{http://www.w3.org/2005/08/addressing}FaultTo
- {Element}{http://www.w3.org/2005/08/addressing}ReplyTo
- {Element}{http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd}Timestamp
- {Content}{http://www.w3.org/2003/05/soap-envelope}Body

Für die XML-Signatur ist der Algorithmus RSASSA-PSS mit Mask Generation Function (MGF) SHA-256 mit einer Salt-Länge von 32 Byte einzusetzen:

- http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1

Alle SOAP-Responses werden vom TMS ebenfalls mit einer Crypt-ID signiert. Somit erfolgt die SOAP-Kommunikation gegenseitig authentisiert und die Integrität wird sichergestellt.

Ein Beispiel für eine XML-Signatur ist im Listing 14 dargestellt.

Das dazu verwendbare WSDL-Schema für SOAP-Nachrichten „CertificateManagementService.wsdl“ zur Sperrung von X.509-Komponenten sowie der zugehörigen Response über die SOAP-Schnittstelle ist als Anlage beigefügt.

Zum Senden eines Zertifikatsperrantrags per SOAP muss die in der WSDL-Datei beschriebene Methode (operation) „revokeCertificate“ genutzt werden. Diese Operation besitzt die input-Message „tns:revokeCertificate“, welche auf den Typ „revokeCertificate“ verweist. Als Inhalt des „revokeCertificate“-Elements muss hierbei als Kindelement genau ein „RevReqContentType“ als „arg0“-Element der Methode übergeben werden. Als Kindelement können nun mehrere konkrete Sperranträge vom Typ „RevDetailsType“ übergeben werden.

Die Rückmeldung erfolgt hierzu analog, jedoch wird als Kindelement der Typ „RevRepContentType“ ebenfalls genau einmal zurückgeliefert. Die Typen sind in der WSDL beschrieben.

Eine Zertifikatssperrung erfordert gemäß [gemSpec_X.509_TSP#TIP1-A_3651] einen Sperrgrund. Dieser muss mit genau einem XML-Element „reason“ angegeben werden. Hierbei werden folgende Werte unterstützt: (Alle weiteren sowie keine Angabe eines Sperrgrunds werden mit einer Fehlernachricht abgelehnt)

- 0: unspecified
- 1: keyCompromise
- 3: affiliationChanged
- 4: superseded
- 5: cessationOfOperation
- 6: certificateHold
- 9: privilegeWithdrawn

In den folgenden Listings (Listing 25 und Listing 26) sind ein Beispiel-SOAP-Request und die zugehörige -Response für die Zertifikatssperrung dargestellt.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" SOAP-
ENV:mustUnderstand="1">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-78376263-f0ff-40a4-9286-84b9dab63ab3">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
                    <ds:Reference URI="#id-b70910c8-8d48-408e-bacb-93f4703eba34">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <ds:DigestValue>ekQthESjCP6wmUX9DV944iYpRPEV3ypFMmR1++D+ICc=</ds:DigestValue>
                    </ds:Reference>
                    <ds:Reference URI="#TS-f4adc244-bd89-4f09-87dc-a74e442e0fb0">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <ds:DigestValue>MeDueVHLbECPYdcywYC9aIGdkhDhNY7dDMIkwFGaVB0=</ds:DigestValue>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue><!-- BASE64-Codierte Signatur --></ds:SignatureValue>
                <ds:KeyInfo Id="KI-0d376223-53af-4cb9-bba9-0fe53e649a72">
                    <wsse:SecurityTokenReference wsu:Id="STR-5d1bf624-bcf6-4442-9d91-25a85b061a60">
                        <ds:X509Data>
                            <ds:X509IssuerSerial>
                                <ds:X509IssuerName>C=DE,O=arvato Systems GmbH,OU=Infrastruktur-CA,CN=Infrastruktur-CA1
TEST-ONLY</ds:X509IssuerName>
                                <ds:X509SerialNumber>14677</ds:X509SerialNumber>
                            </ds:X509IssuerSerial>
                        </ds:X509Data>
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
            <wsu:Timestamp wsu:Id="TS-f4adc244-bd89-4f09-87dc-a74e442e0fb0">
                <wsu:Created>2015-01-06T13:48:52.939Z</wsu:Created>
                <wsu:Expires>2015-01-06T14:38:52.939Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="id-b70910c8-8d48-408e-bacb-93f4703eba34">
        <ns2:revokeCertificate xmlns="http://ws.gematik.de/pki/ComponentCertificateService/v1.0"
xmlns:ns2="http://ws.gematik.de/pki/WSDL/ComponentCertificateService/v1.0">
            <arg0 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RevReqContentType">
                <revDetails>
                    <X509IssuerSerial>
                        <X509IssuerName>CN=GEM.KOMP-CA1 TEST-ONLY,OU=Komponenten-CA der Telematikinfrastruktur,O=gematik
GmbH NOT-VALID,C=DE</X509IssuerName>
                        <X509SerialNumber>62399</X509SerialNumber>
                    </X509IssuerSerial>
                    <X509Subject>STREET=Hauptstraße 15,2.5.4.17=#0c053434343538,CN=80276000112233466668-20150106,O=OLGA-
777777 TEST-ONLY - NOT-VALID,L=Testort,ST=Bayern,C=DE</X509Subject>
                    <reason>1</reason>
                </revDetails>
            </arg0>
        </ns2:revokeCertificate>
```

```
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 25: Beispiel-SOAP-Request für die Sperrung eines X.509-Zertifikats

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-B6B31C2C47C59F1EF9142055213372490">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="soap"/>
                    </ds:CanonicalizationMethod>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"/>
                    <ds:Reference URI="#TS-B6B31C2C47C59F1EF9142055213372385">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="wsse soap"/>
                            </ds:Transform>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>XAnMwI987rACRx84zHwynLESShw=</ds:DigestValue>
                    </ds:Reference>
                    <ds:Reference URI="#id-B6B31C2C47C59F1EF9142055213372489">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList=""/>
                            </ds:Transform>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>Fphkfnb9ptci/UiavLEtAbSEzeY=</ds:DigestValue>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue><!-- BASE64-Codierte Signatur --></ds:SignatureValue>
                <ds:KeyInfo Id="KI-B6B31C2C47C59F1EF9142055213372387">
                    <wsse:SecurityTokenReference wsu:Id="STR-B6B31C2C47C59F1EF9142055213372388">
                        <ds:X509Data>
                            <ds:X509IssuerSerial>
                                <ds:X509IssuerName>C=DE,O=arvato Systems GmbH,OU=Infrastruktur-CA,CN=Infrastruktur-
CA1</ds:X509IssuerName>
                                <ds:X509SerialNumber>2</ds:X509SerialNumber>
                            </ds:X509IssuerSerial>
                        </ds:X509Data>
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
            <wsu:Timestamp wsu:Id="TS-B6B31C2C47C59F1EF9142055213372385">
                <wsu:Created>2015-01-06T13:48:53.723Z</wsu:Created>
                <wsu:Expires>2015-01-06T13:53:53.723Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </SOAP-ENV:Header>
    <soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="id-B6B31C2C47C59F1EF9142055213372489">
        <ns2:revokeCertificateResponse xmlns="http://ws.gematik.de/pki/ComponentCertificateService/v1.0"
xmlns:ns2="http://ws.gematik.de/pki/WSDL/ComponentCertificateService/v1.0">
            <return>
                <status>
                    <status>0</status>
                    <failureInfo>0</failureInfo>
                </status>
                <revCert>
                    <X509IssuerName>CN=GEM.KOMP-CA1 TEST-ONLY,OU=Komponenten-CA der Telematikinfrastruktur,O=gematik
GmbH,C=DE</X509IssuerName>
                    <X509SerialNumber>62399</X509SerialNumber>
                </revCert>
            </return>
        </ns2:revokeCertificateResponse>
    </soap:Body>
</soap:Envelope>
```

Listing 26: Beispiel-SOAP-Response der Sperrung eines X.509-Zertifikats

7 I_OCSP_Status_Information

Es wird ein OCSP-Responder gemäß [gemSpec_PKI#9] konzeptioniert und implementiert. Dieser ermöglicht das Abfragen der Statusinformationen für ausgestellte Zertifikate der Komponenten-PKI. Der OCSP-Responder implementiert die Schnittstelle I_OCSP_Status_Information gemäß [Common-PKI] und wird in der Telematikinfrastruktur bereitgestellt.

Anhang A – Verzeichnisse

A1 – Abkürzungen

Kürzel	Erläuterung
AD	Active Directory
AUT	Authentisierung (Authentication)
AUTN	Technisches Authentisierungszertifikat für Nachrichten
BSI	Bundesamt für Sicherheit in der Informationstechnik
BZA	Berechtigter Zertifikatsantragsteller
CA	certification authority
CP	Certificate Policy
CPI	Certificate Profile Identifier
CPS	Certification Practice Statement
CRL	Certificate Revocation List
DN	Distinguished Name
DNs	Distinguished Names
EE	End Entity
eGK	Elektronische Gesundheitskarte
ENC	Verschlüsselung (Encryption)
ENCV	Technisches Verschlüsselungszertifikat für Verordnungen
FQDN	Fully Qualified Domain Name
GBSM	Gerätebezogenes Sicherheitsmodul
GKV	Gesetzliche Krankenversicherung
gSMC	Gerätebezogene Security Module Card
HBA	Heilberufsausweis
HCI	Health Care Institution
HP	Health Professional
HPC	Health Professional Card
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
ICCSN	ICC Serial Number
ID	Identität (Identity)

Kürzel	Erläuterung
IK	Individual Key
IPSec	Internet Protocol Security
ISM	Information Security Management
KT	Kartenterminal
KTR	Kostenträger
KV	Kassenärztliche Vereinigung
KVNR	Krankenversichertennummer
LEO	Leistungserbringer-Organisation
OCSP	Online Certificate Status Protocol
OCSP-R	OCSP-Responder
OID	Object Identifier
OSIG	Organizational Signature
PKI	Public Key Infrastructure
PKIX	PKI nach X.509 Standard der IETF
PrK	Private Key
PuK	Public Key
QES	Qualifizierte elektronische Signatur
RCA	Root-CA
RFC	Request For Comment
SAS	Sperrantragsteller
SGB	Sozialgesetzbuch
SHA	Secure Hash Algorithm
SIG	Elektronische Signatur
SM	Security Module
SMC-B	Sicherheitsmodul vom Typ B <medizinische Institution>
SMC	Security Module Card
gSMC-K	Security Module Card Konnektor als <holder>
SM-K	Sicherheitsmodul für Konnektoren
SM-KT	Security Module Kartenterminal als <holder>
SM-KT-Zertifikat	X.509-Komponentenzertifikat zu einem SM-KT
SubjectDN	Subject Distinguished Name

Kürzel	Erläuterung
TI	Telematikinfrastruktur
TLS	Transport Layer Security
TMS	Trust Management System
TSL	Trust-service Status List
TSP	Trust Service Provider
VPN	Virtual Private Network
XML	Extensible Markup Language
ZDA	Zertifizierungsdiensteanbieter
ZM	Zulassungsmanagement

A2 – 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.

A3 – Abbildungsverzeichnis

A4 – Tabellenverzeichnis

A5 – Listings

A6 – Referenzierte Dokumente

A6.1 – 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
[gemKPT_Arch_TIP]	gematik: Einführung der Gesundheitskarte – Konzept Architektur der TI-Plattform
[gemKPT_PKI_TIP]	gematik: Einführung der Gesundheitskarte – Konzept PKI der TI-Plattform
[gemSpec_Krypt]	gematik: Einführung der Gesundheitskarte – Übergreifende Spezifikation Verwendung kryptographischer Algorithmen in der Telematikinfrastrukur
[gemSpec_PKI]	gematik: Einführung der Gesundheitskarte – PKI Spezifikation
[gemSpec_X.509_TSP]	gematik: Einführung der Gesundheitskarte – Spezifikation Trust Service Provider X.509

A6.2 – Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
ARV_706.3_KPT_Betriebskonzept	Providerspezifisches Betriebskonzept
[ARV_706.3_RL_Komponenten-PKI_CP]	Zertifizierungsrichtlinie der Komponenten-PKI
[COMMON-PKI]	T7 & TeleTrust (20.01.2009): Common PKI Spezifikation, Version 2.0 http://www.t7ev.org/themen/entwickler/common-pki-v20-spezifikation.html
[DIN5008]	DIN 5008 (2005): Schreib- und Gestaltungsregeln für die Textverarbeitung
[RFC2986]	RFC 2986 (November 2000): PKCS #10: Certification Request Syntax Specification, Version 1.7 Nystrom, M.; Kaliski, B.
[RFC4210]	RFC 4210 (September 2005): Internet X.509 Public Key Infrastructure, Certificate Management Protocol (CMP) C. Adams, S. Farrell, T. Kause, T. Mononen
[RFC4211]	RFC 4211 (September 2005): Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF) J. Schaad, http://www.ietf.org/rfc/rfc4211.txt
[RFC2119]	RFC 2119 (März 1997): Key words for use in RFCs to Indicate Requirement Levels S. Bradner, http://www.ietf.org/rfc/rfc2119.txt (zuletzt geprüft am 28.05.2008)
[RFC3279]	RFC 3279 (April 2002): Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile W. Polk, R. Housley, L. Bassham, https://tools.ietf.org/html/rfc3279

Anhang B – Anforderungsregister

Das Anforderungsregister dient der internen Anforderungsverfolgung durch arvato. Die im Anforderungsregister aufgeführten Anforderungen stellen die Anforderungen dar, durch die die Inhalte der Schnittstellen- und Prozessspezifikation Komponenten-PKI motiviert sind.

Eingangsanforderung	Quelle	Umgesetzt durch
TIP1-A_3597	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3598	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3599	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3889	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3603	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3605	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3606	[gemSpec_X.509_TSP]	Kapitel 4 und 5
TIP1-A_3607	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3608	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3609	[gemSpec_X.509_TSP]	Kapitel 4 und 5
TIP1-A_3611	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_4240	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3612	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3613	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3614	[gemSpec_X.509_TSP]	Kapitel 4
TIP1-A_3615	[gemSpec_X.509_TSP]	Kapitel 4 und 5
TIP1-A_3616	[gemSpec_X.509_TSP]	Kapitel 4 und 5
TIP1-A_5095	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3618	[gemSpec_X.509_TSP]	Kapitel 4 und 5
TIP1-A_3619	[gemSpec_X.509_TSP]	Kapitel 4 und 5
TIP1-A_3620	[gemSpec_X.509_TSP]	Kapitel 4 und 5
TIP1-A_5097	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_5098	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3621	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3622	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3623	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3624	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3626	[gemSpec_X.509_TSP]	Kapitel 5
TIP1-A_3627	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_3629	[gemSpec_X.509_TSP]	Kapitel 5 und 6
TIP1-A_3644	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_3645	[gemSpec_X.509_TSP]	Kapitel 6

TIP1-A_3648	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_3650	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_3651	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_3653	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_3646	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_4244	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_4246	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_4247	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_3654	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_4469	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_5101	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_5102	[gemSpec_X.509_TSP]	Kapitel 6
TIP1-A_4470	[gemSpec_X.509_TSP]	Kapitel 6

Anhang C – Events

C1 – Übersicht der TMS-Events

Zur besseren Übersicht werden die durch das TMS automatisiert generierten Events tabellarisch dargestellt:

Event-ID	Auslöser	Aktion
1.	4.4.2#Prozesschritt 12	Das TMS meldet den Eingang erfasster Daten per E-Mail an alle Rolleninhaber „ZMV“
2.	4.4.3#Prozesschritt 13	Das TMS meldet den Eingang erfasster Daten per E-Mail an alle Rolleninhaber „ZMV“.
3.	4.4.4#Prozesschritt 17	Das TMS meldet den Eingang erfasster Daten per E-Mail an alle Rolleninhaber „ZMV“.
4.	4.4.5#Prozesschritt 17	Das TMS meldet den Eingang der Löschung per E-Mail an alle Rolleninhaber „ZMV“
5.	4.4.6#Prozessschritt 5	Bei Ablehnung eines Zulassungsantrags: Das TMS erzeugt eine E-Mail an alle Rolleninhaber „ZMA“ mit der Begründung sowie dem Hinweis, dass der Zulassungseintrag abgelehnt wurde.
6.	4.4.6#Prozesschritt 10	Bei Freigabe eines Zulassungsantrags: Das TMS meldet den Eingang der freigegebenen Daten per E-Mail an alle Rolleninhaber "ZMS".
7.	5.2.1#Prozessschritt 9	Das TMS meldet das erfolgreiche Erstellen des Zertifikats per E-Mail an den Antragsberechtigten.
8.	5.2.3#Prozesschritt 10	Das TMS meldet das erfolgreiche Erstellen der CryptID per E-Mail an den Antragsberechtigten.
9.	5.2.5#Prozesschritt 10	Das TMS meldet das erfolgreiche Erstellen des Zertifikats per E-Mail an den Antragsberechtigten
10.	6.2.1#Prozesschritt 12	Das TMS informiert den Antragsberechtigten (Organisation) über die Sperrung des Zertifikats per E-Mail.
11.	6.2.3#Prozesschritt 7	TMS informiert den Antragsberechtigten (Organisation) über die Sperrung des Zertifikats per E-Mail.
12.	6.2.6#Prozesschritt 11	Das TMS meldet den Eingang erfasster Sperrdaten per E-Mail an alle Rolleninhaber „GSV“
13.	6.2.6#Prozesschritt 16	Bei Ablehnung eines Sperrantrags: Das TMS erzeugt eine E-Mail an alle Rolleninhaber „GSB“ mit der Begründung sowie dem Hinweis, dass der Sperrantrag abgelehnt wurde.
14.	6.2.6#Prozesschritt 26	Das TMS informiert den Antragsberechtigten (Organisation) sowie alle Rolleninhaber der gematik über die Sperrung des Zertifikats / der Zertifikate per E-Mail.

Anhang D – Übersicht über die verwendeten Versionen

Die folgenden Dateien haben eine Versionsinformation in Form eines Kommentartags am Anfang der jeweiligen Datei erhalten, welche nun direkt mit der Version der Schnittstellenspezifikation (dieses Dokument) verknüpft ist.

~~**Tabelle 59: Verwendete Schemas (XSD)**~~

~~Zertifikatssperrung und Beantragung~~		
	~~XSD Name~~	~~CertificateManagementTypes.xsd~~
	~~Version~~	~~1.0~~

Tabelle 60: Verwendete Definitionen (WSDL)

Zertifikatssperrung und Beantragung		
	WDSL Name	CertificateManagementService.wsdl
	Version	1.0

- Ende des Dokuments -