

Einführung der Gesundheitskarte

Spezifikation für Testlaborkarten: eGK

Dokument-ID	TEST_TLK_eGK
Version:	1.1.0
Stand:	08.04.2011
ClearCase Version:	\main\8
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemSpec_TLK_eGK]

Dokumentinformationen

Änderungen zur Vorversion

In der vorliegenden Version der Testlaborkartenspezifikation wurden redundante Angaben zu den File-Strukturen der Test-Images entfernt. Sie werden mittels, in diesem Dokument referenzierter, XML-Dateien spezifiziert. Die durch SRQs notwendig gewordenen Anpassungen an den File-Strukturen sind in den XML-Dateien vorgenommen worden und müssen in diesem Dokument nicht mehr durchgeführt werden.

Damit beschränkt sich diese Dokument auf konkrete Vorgaben an den Kartenhersteller zu Mitwirkungspflicht des Antragstellers gemäß [gemZul_eGK] zur Durchführung des funktionalen Abnahmetests eGK.

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
0.0.9			hergeleitet aus Version 2.0.0 vom 15.05.2007 von [gemSpec_TLK]	gematik
0.1.0	01.07.09		Anpassungen an die für das Release 2.3.4 maßgeblichen Spezifikationen	TST/DK, Karten
0.2.0	12.08.09		aktualisiert und QM-Kommentare eingearbeitet	TST/DK, Karten
0.3.0	31.08.09		formale Überarbeitung	QM
0.9.0	04.09.09		zur Freigabe empfohlen	QM
1.0.0	29.09.09		freigegeben	PL
1.0.5	15.03.11	Alle	Redundante Informationen entfernt und durch Verweis auf entsprechende Dokumente ersetzt.	TST/KS, Kartentest
1.1.0	08.04.11		freigegeben	gematik

Inhaltsverzeichnis

Dokumentinformationen	2
Inhaltsverzeichnis	3
1 Zusammenfassung	5
2 Einführung.....	6
2.1 Zielsetzung und Einordnung des Dokuments	6
2.2 Zielgruppe	6
2.3 Geltungsbereich	6
2.4 Arbeitsgrundlagen	8
2.5 Abgrenzung des Dokuments.....	8
3 Anforderungen.....	9
3.1 Layout der Testlaborkarte	9
3.2 Für die Testlaborkarte relevante Prüfgebiete	10
3.3 Grundlagen	10
3.3.1 ATR.....	10
3.3.2 CVC's	11
3.3.3 Vergabe der ICCSN	11
3.3.4 X.509.....	11
3.4 Transport-PIN-Verfahren	11
3.5 QES-Anwendung komplettieren	12
3.6 Unterbrechungstest	12
4 Test Labor Dateien	14
4.1 Aufbau der XML Datei „eGK_TestLaborKartenDaten.xml“	14
4.2 Aufbau der XML-Datei „eGK_TestLaborKartenTemplate.xml“	15
4.2.1 Template <card_capabilities>	16
4.2.2 Template <basistest>.....	17
4.2.3 Template <anwendungstest>.....	18
4.2.4 Template <unterbrechungstest>	20
5 Initialisierungsdateien	22
5.1 Übersicht der Image-Files (d3-Files)	22
5.2 Aufbau und Struktur eines Image-Files (d3-Files)	23

Anhang A	25
A1 – Abkürzungen.....	25
A2 - Glossar	26
A3 - Abbildungsverzeichnis	26
A4 - Tabellenverzeichnis	26
A5 - Referenzierte Dokumente	26

1 Zusammenfassung

Für die Ausgabe der elektronischen Gesundheitskarte (eGK), der Heilberufsausweiskarte (HBA) – oder auch Health Professional Card (HPC) genannt – und der Security Module Card (SMC) sowie deren Einsatz in der Telematikinfrastruktur des Gesundheitswesens ergeben sich besonders hohe Anforderungen an die Zuverlässigkeit, Sicherheit und den Datenschutz. Zur Sicherstellung dieser Anforderungen hat die gematik ein Zulassungsverfahren entwickelt, Prüfprozesse definiert sowie zentrale Testplattformen installiert.

~~In der [RVO2006] werden 4 Teststufen Labortest, Anwendertest, 10.000er Feldtest und 100.000er Feldtest (auch Massentest genannt) vorgegeben.~~

In diesem Dokument wird die für den Labortest einer eGK notwendige Datenstruktur spezifiziert. Ergebnis sind ~~zwei~~ Dateien im XML-Format, die die gematik den Herstellern (Antragstellern) zur Vorbereitung auf die ~~funktionalen Abnahmetests~~ ~~Zulassungstests~~ einer eGK zur Verfügung stellt. ~~Die Beide~~ Dateien enthalten Informationen, die dem Antragsteller zur Erstellung von Image-Files dienen. Diese stellen die Grundlage für den Basis-, Anwendungs- und Unterbrechungstest dar und werden während der Prüfung situationsbezogen in den persistenten Speicher der eGK eingebracht. Darüber hinaus ist eine der beiden XML-Files mit herstellerepezifischen Angaben zu ergänzen. Details zu den Dateien sind den Kapiteln 4 und 5 zu entnehmen. In diesem Zusammenhang wird auf die Mitwirkungspflicht des Antragstellers gemäß [gemZul_eGK] hingewiesen.

2 Einführung

2.1 Zielsetzung und Einordnung des Dokuments

Die eGK muss vor dem produktiven Einsatz ausführlich auf Übereinstimmung mit den Spezifikationen und den Vorgaben zur Funktionalität geprüft werden.

Dieses Dokument beschreibt:

- welche Informationen (XML-Files) die gematik den Herstellern/Antragstellern zur Verfügung stellt,
- welche Chip-ROM-Masken- und herstellerspezifischen Informationen in das Template XML-Files einzubringen sind,
- welche Image-Files (d3-Files), die während des Testablaufes zur Initialisierung des persistenten Speichers der eGK benötigt werden und mit welchen Inhalten zu erstellen sind,
- wie die vom Hersteller zur Verfügung zu stellenden Testlaborkarten optisch zu gestalten sind.

2.2 Zielgruppe

Das Dokument richtet sich an

- Hersteller/Antragsteller die eine eGK für die Telematikinfrastuktur zulassen möchten,
- den Testbereich der gematik

und ermöglicht die Herstellung spezifikationsgerechter eGK-Testlaborkarten für Zulassungstest.

2.3 Geltungsbereich

Dieses Dokument enthält verbindliche Festlegungen für die Abläufe, die Datenformate und ~~Inhalte~~ ~~inhalte~~ sowie die Verantwortlichkeiten für die Erzeugung der verschiedenen zur Prüfung einer eGK (Testlaborkarte) benötigten Daten.

Tabelle 1 ~~Abbildung 1~~ zeigt eine grobe Unterteilung des Labortests und die Einordnung des für die Prüfung einer eGK relevanten Komponententests (KT) und ihren Testphasen Funktionstest (KFT), Interoperabilitätstest (KIT) und Leistungstest (KLT).

Abbildung 1: Einordnung Tests und Testphasen

Tabelle 1 – Überblick über die Testphasen und Verantwortlichkeiten

Testphase		Umfang der Prüfung	Verantwortliche Org.
1	Komponenten-Funktionstest - KFT	Testphase im Komponententest. Testfokus sind die funktionalen Anforderungen einer einzelnen Komponente.	gematik
2	Komponenten-Interoperabilitätstest - KIT	Testphase im Komponententest. Testfokus ist das Zusammenspiel mehrerer Komponenten und die Austauschbarkeit von gleichartigen Komponenten.	gematik
3	Komponenten-Leistungstest - KLT	Testphase im Komponententest. Testfokus ist das Leistungsverhalten einer einzelnen Komponente (oder auch im Zusammenspiel mit anderen Komponenten).	gematik
4	Dienst-Funktionstest - DFT	Testphase im Diensttest. Testfokus sind die funktionalen Anforderungen eines einzelnen Dienstes.	gematik
5	Dienst-Interoperabilitätstest - DIT	Testphase im Diensttest. Testfokus ist das Zusammenspiel mehrerer Dienste (ggf. auch mit Komponenten). Dabei werden in mehreren Stufen immer größere Systeme von integrierten Diensten betrachtet.	gematik
6	Dienst-Leistungstest [einzeln] – DLT-E	Testphase im Diensttest. Testfokus ist dabei das Leistungsverhalten einzelner Dienste.	gematik
7	Dienst-Leistungstest [integriert]– DLT-I	Testphase im Diensttest. Testfokus ist dabei das Leistungsverhalten mehrerer Dienste im Zusammenspiel insbesondere bei der Ausführung von Anwendungsfällen. Der DLT-I kann auch Ausfalltests einzelner Dienste/Komponenten enthalten bei denen das Verhalten des Gesamtsystems überprüft wird.	gematik
8	Dienst-Monitoringtest [einzeln] – DMT-E	Testphase im Diensttest. Beim DMT-E werden die Schnittstellen für das Monitoring und das Systemmanagement der einzelnen Dienste getestet.	gematik
9	Dienst-Monitoringtest [integriert] – DMT-I	Testphase im Diensttest. Beim DMT-I werden die Schnittstellen für das Monitoring und das Systemmanagement im Zusammenspiel der Provider, der Betriebsleitzentrale und dem Leitstand getestet.	gematik
10	Dienst-Sicherheitsrobustheitstest - DST	Testphase im Diensttest. Beim Sicherheitsrobustheitstest erfolgen Angriffsversuche, wie diese auch ein realer Angreifer durchführen würde. Dabei wird versucht, konzeptionelle Schwä-	gematik

Testphase		Umfang der Prüfung	Verantwortliche Org.
		chen und Implementierungsfehler auszunutzen, um so vorhandene Sicherheitsfehler aufzudecken.	
11	Physikalische Tests	Die Materialprüfung beinhaltet die physikalischen Tests der Komponente. Der Nachweis, dass die geforderten Eigenschaften erfüllt werden, kann durch Herstellererklärungen und durch Prüfungsnachweise akkreditierter Testlabore belegt werden.	Externe Prüfstellen
12	Betriebsaudit - AuB	Betriebsaudits sind Prüfungen von kritischen und wahrscheinlichen Betriebsszenarien, auch querschnittlicher Art, mit dem Schwerpunkt Betriebsprozesse. Das Ziel von Betriebsaudits ist, alle wichtigen Anforderungen des Betriebes an die Betreiber und Provider von Komponenten der Telematikinfrastruktur, die sich aus den anforderungsgebenden Dokumenten (wie gemBetr_Qu, gemBetr_SLA und gem_Betr_LB für die jeweiligen Dienste und weitere Zulassungsanforderungen) ergeben, zu auditieren und die Erfüllung dieser Anforderungen damit zu verifizieren.	gematik
13	Sicherheitsaudit - AuS	Es wird in Form einer Auditierung geprüft, ob ein Provider die Maßnahmen umgesetzt hat, die vom gematik-Sicherheitskonzept gefordert sind und die er in seinem individuellen Sicherheitskonzept beschrieben hat.	gematik

2.4 Arbeitsgrundlagen

Die Grundlagen für dieses Dokument stellen die eGK-Prüfvorschrift [gemPVo_eGK], die eGK-Spezifikationen [gemSpec_eGK_P1] und [gemSpec_eGK_P2] sowie die Testspezifikationen [gemTSp_eGK] und [gemTSp_UTB] dar.

2.5 Abgrenzung des Dokuments

Dieses Dokument legt den Focus auf die für die Prüfung einer eGK notwendigen Daten und Strukturen sowie die äußere Form einer für den Labortest notwendigen eGK-Testlaborkarte.

Nicht Gegenstand dieses Dokumentes sind Optik, Struktur und Inhalt der

- eGK-Musterkarten,
- eGK-Testkarten für die weiteren Teststufen.

3 Anforderungen

3.1 Layout der Testlaborkarte

Auf der Testlaborkarte muss neben dem Typ der Karte der Name des Herstellers, die Version und das Erstellungsdatum aufgedruckt werden. Die Anordnung der Beschriftung soll der folgenden Abbildung entsprechen und in Verdana True Type oder gleichartiger Schriftart in 10 pt erfolgen. Wegen der geringen Ansprüche an das Layout der Testlaborkarte wurde auf eine Bemaßung verzichtet.

Optional kann zusätzlich das Logo des Herstellers aufgebracht werden.

Die von der Testlaborkarte unterstützten Kartentypen, hier mindestens eGK, sind in der rechten oberen Ecke aufzubringen.

Der Produktname wird durch den Hersteller bestimmt.

Das Erstellungsdatum dient der Unterscheidung zu unterschiedlichen Zeitpunkten eingereichter Testlaborkarten gleicher Version.

Die Version muss eine ROM-Masken-Version unzweifelhaft wiedergeben. Als Version ist das „Pre-Issuing Data Objekt“ aus dem ATR und die Patchversion anzugeben.

Die Antragsnummer erhält der Kartenhersteller nach Eingang des Zulassungsantrags und erlaubt die eindeutige Zuordnung der Karte zu einem funktionalen Abnahmeprozess.

Die Gestaltung der Kartenrückseite ist dem Hersteller überlassen.

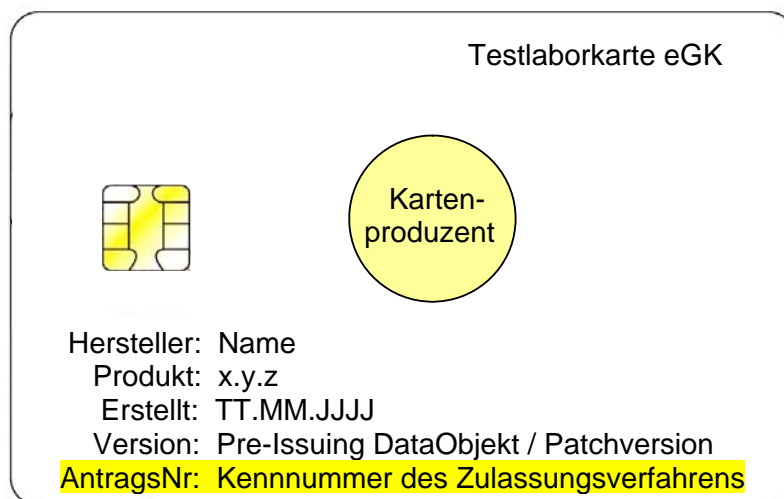


Abbildung 2: Layout einer Testlaborkarte vom Typ eGK

3.2 Für die Testlaborkarte relevante Prüfgebiete

Wie bereits in der Prüfvorschrift [gemPVo_eGK] beschrieben, unterteilt sich der Komponententest in Testphasen und Prüfgebiete:

Testphase Komponenten-Funktionstest mit den Prüfgebieten:

- Basistest
Test der Anforderungen an ein Card Operating System (COS) auf der Basis der für diese Tests notwendigen Zugriffbedingungen und Datenstrukturen inkl. **Transport-PIN-Verfahren und** des Transportprotokolls (T=1).
- Anwendungs- und Strukturtest
Tests der für eine eGK vorgesehenen Anwendungen inklusive QES in den unterschiedlichen Ausprägungen.
- Unterbrechungstest
Test der Kommando- und File-Kandidaten, für die das Roll-Forward- bzw. Roll-Back-Verfahren vorgeschrieben sind.

Testphase Komponenten-Interoperabilitätstest mit dem Prüfgebiet:

- IOP Test eGK/**weitere Komponenten/Kartenleser/Konnektoren**
Im Rahmen der Zulassungstest wird die eGK mit allen relevanten, bereits zugelassenen Komponenten einem Interoperabilitätstest unterzogen.

Testphase Komponenten-Leistungstest mit dem Prüfgebiet:

- Performancetest
Test entsprechend der Vorgaben zum Performanceverhalten (siehe [gemSpec_eGK_P1#AnhB]) einer eGK.

3.3 Grundlagen

Einige durch die gematik vorgegebene allgemeine Informationen gelten für alle Testszenarien und haben Auswirkungen auf die vom Hersteller zu liefernden Informationen.

3.3.1 ATR

Der ATR der Karte **muss für jedes sollte** der zu erstellenden Images identisch sein und **nachfolgenden Tabelle, die auf** den Vorgaben der [gemSpec_eGK_P1#Kap.12.2.1] und [gemSpec_eGK_P2#Kap.6.1.1] **aufbaut** entsprechen. **Diese gelten für den Anwendungstest und für den Basistest.** Die im ATR vermerkten Eigenschaften der Karten werden explizit getestet und mit den globalen Parametern aus [gemXML_eGKTpl] sowie den Werten im EF.ATR abgeglichen.

Hinweis:

Das Pre-Issuing-Dataobject muss zusammen mit der Patchversion ein COS eindeutig charakterisieren und wird als Versionsidentifikation in den Zulassungsdokumenten verwendet.

Da der ATR zulassungsrelevante Eigenschaften der Karte widerspiegelt, darf sich dieser zwischen den Testlaborkarten für den Basis- und Anwendungstest sowie Muster- und Eckkarten nicht unterscheiden.

Tabelle 1: ATR Werte (Empfehlungen)

3.3.2 CVC's

Für die Generation 1 Karten ist, basierend auf der gematik Root-CA ([gemXML_eGKDaten] siehe eGK_TestLaborKartenDatenVx.x.x.xml Hauptknoten ,common'), eine Datenbank mit den auf die gematik-Root-CA basierenden CVC's angelegt. Die CVC-Datenbank kann ROM-Maskenentwicklern zur Verfügung gestellt werden. Die Nutzung der CVC-Datenbank ist jedoch freigestellt und nicht verpflichtend.

3.3.3 Vergabe der ICCSN

Sowohl für den Basistest als auch für den Anwendungstest ist das EF.GDO in der Karte definiert. Die ICCSN kann der Hersteller im Rahmen der Vorgaben in [gemSpec_eGK_P1] und [gemSpec_eGK_P2] frei festlegen. Die ICCSN muss jedoch über alle Testlaborkarten einer Lieferung denselben Wert besitzen, **dieser sollte in allen Images konstant sein** und bei den relevanten CVC's Berücksichtigung finden.

3.3.4 X.509

In den Datenfiles (EFs) im Anwendungstest, die ein X.509-Zertifikat mit dem öffentlichen Schlüssel zu einem privaten Schlüssel in der Karte enthalten sollen, wird von der Testsuite ein syntaktisch korrektes X.509-Zertifikat erwartet, aus dem mit den üblichen Verfahren der öffentliche Schlüssel ausgelesen werden kann. Das X.509-Zertifikat wird keiner formalen Prüfung unterzogen.

3.4 Transport-PIN-Verfahren

Die eGK-Spezifikation [gemSpec_eGK_P1] definiert sieben Transport-PIN-Verfahren, deren Prüfung Bestandteil des Basistests ist. Vom Hersteller muss mindestens ein Verfahren aus der Menge „Leer-PIN_1“, „Leer-PIN_2“ oder „Transport-PIN_0000“ unterstützt werden. Das Verfahren „regularPasswort“ ist von jedem Hersteller zu unterstützen. Darüber hinaus können vom Hersteller weitere Verfahren unterstützt werden, auf die nach positivem Test die Zulassung ausgedehnt wird.

Die vom Hersteller unterstützten Transport-PIN-Verfahren sind in [gemXML_eGKTp], Knoten <transportpin_verfahren_x> zu kennzeichnen und im Image für **Die folgende Tabelle spiegelt den Stand Basistest gemäß [gemXML_eGKDaten] umzusetzen. der gestatteten**

Jedes Transport-PIN Verfahren wird in einem eignen DF getestet. Die Zugriffsrechte sind in jedem DF für das zu testende Transportschutzverfahren zu konfigurieren, so dass nur das für diese Verfahren zulässige Kommando akzeptiert wird. Siehe hierzu [gemXML_eGKDaten], gemäß [gemFKCMS_PIN] zur Zeit der Veröffentlichung dieses Dokumentes wieder.

Tabelle 2: Zugelassene Transport-PIN-Verfahren

3.5 QES-Anwendung komplettieren

Die QES-Anwendung kann sich zum Zeitpunkt der eGK-Ausgabe in unterschiedlichen Zuständen auf einer eGK befinden. Nachfolgend wird nur die Variante betrachtet, bei der die QES-Anwendung mit den benötigten Dateien inklusive der Zugriffsregeln angelegt ist, aber in EF.C.CH.QES kein für Signaturzwecke nutzbares Zertifikat und u. U. keinen QES-Schlüssel enthält. Um diese Anwendung nutzbar zu machen, muss sie komplettiert werden, d. h. fehlende Elemente müssen nachgeladen werden.

Laut [gemSpec_eGK_P2] kann es zum Zeitpunkt der Auslieferung einer eGK maximal sechs unterschiedlich zu komplettierende Varianten, basierend auf unterschiedlichen Authentikationsvarianten und unterschiedlichen Schlüsselvoraussetzungen, geben. Nachfolgend die Komplettierungsvarianten:

Aufbau eines sicheren Kommunikationskanals

- Trusted Channel mittels symmetrischem Schlüssel gemäß [gemSpec_eGK_P2],
- Trusted Channel mittels asymmetrischem Schlüssel gemäß [gemSpec_eGK_P2].

Existenz des Signaturschlüsselpaares

- Signaturschlüsselpaar nicht vorhanden gemäß [gemSpec_eGK_P2],
- Signaturprüf Schlüssel auslesbar gemäß [gemSpec_eGK_P2],
- Signaturprüf Schlüssel im Gütesiegel gemäß [gemSpec_eGK_P2].

Der Komplettierungsteil der QES-Anwendung ist ohne Transportschutz zu personalisieren. Trotzdem ist EF.BVD obligatorisch und enthält eine PUK zu PIN.QES. Da kein Transport-PIN-Verfahren genutzt wird, sollte der entsprechende Wert im EF.ASD (siehe Anforderung N993300) auf '00' gesetzt werden.

Die Komplettierung der QES-Anwendung ist optional. Ein Hersteller kann eine beliebige Teilmenge der Komplettierungsvarianten unterstützen. Die vom Hersteller unterstützten Komplettierungsvarianten sind in [gemXML_eGKTpl], Knoten <completion_of_qes> zu kennzeichnen. Für jede der zu testenden Komplettierungsvarianten ist ein entsprechender Image-File bereitzustellen (siehe Tab.5).

Es ist zu beachten, dass die gesamte Struktur – für alle Anwendungen plus Basisstruktur für die Komplettierungsvariante – auf der eGK vorhanden sein muss. Die Komplettierungstests umfassen auch den für die Komplettierung notwendigen Speicherplatz.

3.6 Unterbrechungstest

Wegen der in der Telematikinfrastuktur zum Einsatz kommenden Steckkartenleser kommt den Unterbrechungstests eine große Bedeutung zu. Durch ein eventuelles Ziehen der Karte kommt es unweigerlich zu einem Spannungsabfall in der eGK. Geschieht dies nun ausgerechnet zu einem Zeitpunkt, während dessen **wichtige Daten z. B. eVerordnungstickets** in der eGK abgelegt werden sollen, kann **dies nicht mehr mit Sicherheit gesagt werden, ob das Ticket komplett oder nur** zu einem **undefinierten Teil oder gar nicht**

im dafür vorgesehen Datenbereich abgelegt worden ist. Es existiert ein undefinierter Zustand im Datenstand der Karte führen.

Mit den Unterbrechungstests werden gezielt durch Spannungsabfälle undefinierte Zustände in der eGK provoziert und anschließend das spezifikationskonforme Verhalten der eGK getestet. Nach jedem Unterbrechungstest, der einen undefinierten Zustand hinterlässt, muss die eGK neu initialisiert werden. Um die Initialisierung zeitlich so kurz wie möglich zu halten, besteht das Image-File (d3-File) für den Unterbrechungstest nur aus dem Anlegen des ROOT-Verzeichnisses (nur MF, kein Datenfile) und eventuell Grundstrukturen. Die für die Unterbrechungstests notwendigen Datenfiles sind in der Template-datei definiert, die vom Hersteller um APDU-Sequenzen zu ergänzen sind (siehe Tab 6).

4 Test Labor Dateien

Die gematik stellt Herstellern Dateien im XML-Format zur Verfügung, die die Basis für den Mitwirkungsprozess des Herstellers darstellen. Es handelt sich um

- eine XML-Datendatei die Vorgaben der gematik enthält (eGK_TestLaborKartenDaten.xml)

und

- eine XML-Templatedatei, die durch den Hersteller zu ergänzen ist (eGK_TestLaborKartenTemplate.xml).
(eGK_TestLaborKartenTemplateVx.x.x.xml) und
- eine XML-Datendatei die Vorgaben der gematik enthält (eGK_TestLaborKartenDatenVx.x.x.xml).

Zur Durchführung der unterschiedlichen funktionalen Tests macht die gematik Vorgaben bzgl. Inhalt und Struktur der Karte. Für den Basistest muss die Testlaborkarte eine spezielle Teststruktur erhalten, die der XML-Datendatei entnommen werden kann. Für den Anwendungs- und Performacetest ist eine Struktur gemäß [gemSpec_eGK_P2] zu erstellen und um größtenteils frei wählbare Daten zu ergänzen. Sind durch die gematik keine Vorgaben erstellt, sind die Werte durch den Hersteller frei wählbar.

Der/Die Dateinamen der XML-Templatedatei ist und der XML-Datendatei sind vom Kartenhersteller entsprechend dem Beispielen/Beispielen auszutauschen:

- eGK_TestLaborKartenTemplate.xml
→ eGK_TLK_<Hersteller>_<Version>_Template_<Datum>.xml
- eGK_TestLaborKartenDatenV2.1.0.xml
→ eGK_TLK_<Hersteller>_<Version>_Daten_<Datum>.xml

<Hersteller> Name des Herstellers

<Version> Die Version muss sich auf eine COS-Version beziehen und dem hexadezimal kodierten Pre-Issuing Data Object im ATR entsprechen (siehe Tabelle 1) und enthält zusätzlich die Kartenversion (Patchversion).

<Datum> Modifikationsdatum der XML-Templatedatei (jjjjmmtt)

4.1 Aufbau der XML Datei „eGK_TestLaborKartenDaten.xml“

Die Datei [gemXML_eGKDaten] enthält Vorgaben der gematik und bedarf keiner Modifikation durch den Hersteller. Diese Datei stellt eine Grundlage für die Erstellung der Image-Files (d3-Files) dar.

Die Datei ist in drei Hauptknoten unterteilt. Es wird dabei zwischen Angaben für den Basistest, den Anwendungstest und den für beide Tests notwendigen Vorgaben unterschieden.

- Knoten <common>
Der für den Basis- und Anwendungstest relevante ROOT_CA_KEY ist hier abgelegt.
- Knoten <basistest>
Struktur, Dateninhalte, Passworte, Schlüssel, Zugriffsrechte und Transport-PIN-Verfahren sind auf die lt. [gemSpec_eGK_P1] notwendigen Tests abgestellt. Diese Tests stellen die Basisfunktionalität einer eGK sicher. Die hier beschriebenen Strukturen sind in einem Image für den Basistest umzusetzen.
- Knoten <anwendungstest>
Die Strukturen für den Anwendungstest müssen entsprechend [gemSpec_eGK_P2] umgesetzt werden. In diesem Knoten befinden sich zu verwendende Vorgaben für die PIN/PUK und Schlüsselwerte. Der Anwendungstest umfasst im Weiteren das Anlegen einer neuen Applikationen im Umfeld der Datenstruktur lt. [gemSpec_eGK_P2]. Auf der Basis der hier definierten Struktur ist eine adpu-Sequenz für das LOAD APPLICATION Kommando in der [gemXML_eGKTpl], Knoten <anwendungstest>/<new_application> zu erstellen.
- Knoten <unterbrechungstest>
Die hier definierten Strukturen sind für das Unterbrechungstest-Image umzusetzen.

Die Datenstruktur und Dateninhalte dieser XML-Datei werden von der gematik vorgegeben und dürfen nicht modifiziert werden!

4.2 Aufbau der XML-Datei „eGK_TestLaborKartenTemplate.xml“

Die Datei [gemXML_eGKTpl] enthält von der gematik zur Verfügung gestellte Rahmeninformationen, die durch den Hersteller zu ergänzen, auszutauschen oder zu selektieren sind.

Die Datei ist in vier Hauptknoten unterteilt. Es wird unterschieden zwischen den Knoten die die Karteneigenschaften beschreiben, den Daten für den Basistest, dem Anwendungstest und dem Unterbrechungstest.

- Knoten <card_capabilities>
Karten- und herstellerspezifische Eigenschaften z. B. die Bezeichnung der Image-Files (d3-Files), Angaben zum EF_ATR oder verwendete Transport-PIN-Verfahren sind zu ergänzen.
- Knoten <basistest>
Für den Test des Basiskommandos LOAD APPLICATION ist die adpu-Sequenz zur Anlage einer Datei (EF_) zu hinterlegen.
- Knoten <anwendungstest>
Im Mittelpunkt dieses Knotens stehen das Anlegen von Dateien und Applikationen sowie die Komplettierungsvarianten der QES-Anwendung.

- Knoten <unterbrechungstest>
Die benötigten apdu-Sequenzen zum Anlegen der für den Unterbrechungstest benötigten Dateien sind hier zu hinterlegen.

In den nachfolgenden Tabellen werden nur die Elemente/Attribute beschrieben, die durch den Hersteller in der Templatedatei zu modifizieren oder zu ergänzen sind.

4.2.1 Template <card_capabilities>

Die Angaben in diesem Knoten sind mit größter Sorgfalt zu erstellen, da ansonsten z. B. der Test aufgrund nicht gefundener D3-Image-Files nicht durchgeführt oder auf die falsche Ausprägung hin getestet wird. Letztere fließen in den Abnahmebericht ein. In der folgenden Tabelle sind die Hauptelemente und ihre Funktion erklärt. Die Tabelle dient als Orientierungshilfe. Weitere Hinweise zu den Elementen finden sich in [gemXML_eGKtp].

Tabelle 3: Hauptknoten <card_capabilities>

Element / Attribut	Attribut	Wert
<OS_APDU_LIST> Name des für den Basistest notwendigen d3-Files.		eGK_TLK_<Hersteller>_<Version>_<Initfile>_<Datum>.d3 (siehe Kapitel 5)
<APPL_APDU_LIST> Name des für den Anwendungstest notwendigen d3-Files.		eGK_TLK_<Hersteller>_<Version>_<Appl>_<Datum>.d3 (siehe Kapitel 5)
<BREAK_APDU_LIST> Name des für den Unterbrechungstest notwendigen d3-Files.		eGK_TLK_<Hersteller>_<Version>_<UbTest>_<Datum>.d3 (siehe Kapitel 5)
<TRANSPORTPIN _VERFAHREN_X> (X = 1 bis 7 + F) Dieser Eintrag steuert den Test der Transport-PIN-Verfahren. Jedes auf ‚YES‘ gesetzte Verfahren wird getestet und nur für diese sind Zulassungen möglich.		Wert aus YES NO YES → wird unterstützt und getestet NO → wird nicht unterstützt Verfahren „reguläres Passwort“ ist zwingend zu unterstützen
<SUPPORTS_LOGICAL_EOF> (Betrifft alle binären Dateien) Mit diesem Eintrag wird die Verwendung Logical End of File für die genannten Dateien aus dem Anwendungs-Image angezeigt.		Wert aus YES NO YES → Verwendung von logical EOF NO → keine Verwendung
<supports_persistent_keys> Die relevanten öffentlichen Schlüssel der Zertifizierungsinstanz und die öffentlichen Authentifikationsschlüssel können persistent gespeichert werden.	<ca_keys>	Wert aus YES NO YES → Schlüssel werden persistent gespeichert NO → keine persistente Speicherung
	<ca_keys . count= >	“x“ wenn YES, hexadezimal kodierte Anzahl der Schlüssel die persistent gespeichert werden können

Element / Attribut	Attribut	Wert
	<aut_keys>	Wert aus YES NO YES → Schlüssel werden persistent gespeichert NO → keine persistente Speicherung
	<aut_keys . Count= >	“x” wenn YES, hexadezimal kodierte Anzahl der Schlüssel die persistent gespeichert werden können
<ef_atr><preissuing_data> Über diesen Eintrag ist eindeutig die Karte identifizierbar. Es handelt sich um das Pre-Issuing Data Object im ATR / Historical Bytes und den Daten des TAG '66'/'46' des EF.ATR. Die hier einzutragenden Werte sind hexadezimal zu kodieren.	<icm>	Angabe der IC-Herstellererkennung
	<ict>	Angabe des IC-Typ
	<osv>	Angabe zur Operation System Version
	<dd>	Discretionary Data
	<iccm>	Kartenherstellererkennung
<ef_atr><buffer_size> Die Angaben zu den max. APDU-Längen muss mit denen im EF.ATR TAG 'E0' identisch sein. Die Längen sind hexadezimal zu kodieren	<req_apdu_ohne_sm >	maximale Länge einer ungesicherte Kommando APDU
	<rep_apdu_ohne_sm >	maximale Länge einer ungesicherte Antwort APDU
	<resq_apdu_mit_sm >	maximale Länge einer Kommando APDU mit Secure Messaging
	<resp_apdu_mit_sm >	maximale Länge einer Antwort APDU mit Secure Messaging

4.2.2 Template <basistest>

In [gemXML_eGKDaten] sind Karten-Files, Verzeichnisse, PINs und Schlüssel für den Basistest definiert, welche per LOAD APPLICATION in der Karte angelegt werden sollen. Die hierfür benötigten APDU-Sequenzen sind hier abzulegen. Weitere Hinweise sind [gemXML_eGKTpl] zu entnehmen.

Tabelle 4: Hauptknoten <basistest>

Element	Attribut	Wert
<SUPPORTS_LOGICAL_EOF> Dieser Eintrag ist genau dann auf "YES" zu setzen, wenn das COS LogicalEndOfFile unterstützt. Zusätzlich ist unter "DF_TESTB" ein "ef_test_leof" anzulegen, mit dem die Korrektheit dieses Mechanismus geprüft werden kann.		Wert aus YES NO YES → Verwendung von logical EOF NO → keine Verwendung
<df_testB><ef_testBn> Die Datei EF.TESTBn ist transparentes EF, das nicht statisch im DF.TESTB anzulegen ist, sondern zum dynamischen Anlegen von Dateien mit frei	<apdu>	Herstellerindividuelle Datensequenz unter Berücksichtigung vorgegebener Werte um das EF mit LOAD APPLICATION anlegen zu können.

Element	Attribut	Wert
wählbaren FID und Größe dient.		

4.2.3 Template <anwendungstest>

In der eGK-Anwendung nach [gemSpec_eGK_P2] sind mehrere Files löscherbar. Um sie wieder anlegen zu können, sind in diesem Knoten entsprechende APDU-Sequenzen anzulegen. Zusätzlich sind für den Test des LOAD APPLICATION-Kommandos und der Anforderung zur Speicherdefragmentierung APDU-Sequenzen nach den Vorgaben in [gemXML_eGKDaten] zu erstellen. In weiteren sind in [gemXML_eGKTpl] Angaben zur QES-Anwendung (vorhanden Ja/Nein, Nachladeverfahren) zu machen. Die Angaben zur QES-Anwendung haben Einfluss auf den Abnahmebericht des funktionalen Tests und sind mit größter Sorgfalt zu erstellen. Weitere Hinweise sind [gemXML_eGKTpl] zu entnehmen.

Tabelle 5: Hauptknoten <anwendungstest>

Element	Attribut	Wert
<df_hca>	<apdu>	Herstellerindividuelle Kommando-sequenz zur Neuanlage gelöschter EFs, sowie ein Template zum Test der Speicherdefragmentierung..
<pin_01>, <pin_02>	<secret>, <puk>	Die vorgegebenen Werte sind für die eGK Anwendung zu verwenden.
<df_hca><ef_pd> <df_hca><ef_vd> <df_hca><ef_gvd> <df_hca><ef_eVerordnungsContainer> <df_hca><ef_TTN> Mehrere Dateien der Health Care Appli-cation werden im Zuge des Anwen-dungstest gelöscht und müssen neu angelegt werden. Die dazu notwendigen Informationen sind zu liefern.	<apdu>	Herstellerindividuelle Kommando-sequenz zur Anlage des EF unter Berücksichtigung vorgegebener Werte gemäß gemSpec_eGK_P2.
<df_hca ><ef_testBn> Die Datei EF.TESTBn ist transparentes EF, das nicht statisch im df.hca anzulegen ist, sondern zum dynamischen Anlegen von Dateien mit frei wählbaren FID und Größe dient.	<apdu>	Herstellerindividuelle Datense-quenz unter Berücksichtigung vorgegebener Werte um das EF mit LOAD APPLICATION anlegen zu können.
<df_qes> Die Anwendung „Qualifizierte Signatur“ kann in der eGK angelegt aber noch nicht nutzbar sein. Zur Komplettierung der QES Anwendung stehen unterschiedliche Wege zur Verfügung die unterschiedliche Anforderungen an die Template Dateien stellen.	<supported>	Wert aus COMPLETE NO complete → QES ist komplett angelegt und kann genutzt werden. Diese Variante beinhaltet auch die Zulassung der eGK ohne QES. no → QES wird nicht unterstützt oder ist zu kompletie-

Element	Attribut	Wert
		ren. Das Anwendungsimage (d3-File) beinhaltet keine QES Fragmente.
<p><df_qes><completion_of_qes> Die QES Anwendung ist in einem „Rohzustand“ in der Karte und muss komplettiert werde. Jede der sechs Komplettierungsvarianten besitzt eine Anzahl Attribute die kartenindividuell angepasst werden müssen.</p>		Für jede zu testende Komplettierungsvariante ist ein zur Variante passendes Anwendungsimage (d3-File) zur Verfügung zu stellen.
<p><df_qes><completion_of_qes> <completion_symm_keynotavailable> TC: symm. Schlüssel Signatur Schlüssel: nicht vorhanden</p>	<supported>	Wert aus YES NO YES → diese Komplettierungsvariante wird unterstützt NO → wird nicht unterstützt
	<apdu_list>	eGK_TLK_<Hersteller>_<Version>_symKeyNotRead_<Datum>.d3 (siehe Kapitel 5)
<p><df_qes><completion_of_qes> <completion_symm_pukreadable> TC: symm. Schlüssel Signatur Schlüssel: vorhanden, öffentliche Schlüssel auslesbar</p>	<supported>	Wert aus YES NO YES → diese Komplettierungsvariante wird unterstützt NO → wird nicht unterstützt
	<apdu_list>	eGK_TLK_<Hersteller>_<Version>_symPukRead_<Datum>.d3 (siehe Kapitel 5)
<p><df_qes><completion_of_qes> <completion_symm_pukcertificate> TC: symm. Schlüssel Signatur Schlüssel: vorhanden, öffentliche Schlüssel im Gütesiegel</p>	<supported>	Wert aus YES NO YES → diese Komplettierungsvariante wird unterstützt NO → wird nicht unterstützt
	<apdu_list>	eGK_TLK_<Hersteller>_<Version>_symPukCert_<Datum>.d3 (siehe Kapitel 5)
<p><df_qes><completion_of_qes><completion_asymm_keynotavailable> TC: asymm. Schlüssel Signatur Schlüssel: nicht vorhanden</p>	<supported>	Wert aus YES NO YES → diese Komplettierungsvariante wird unterstützt NO → wird nicht unterstützt
	<apdu_list>	eGK_TLK_<Hersteller>_<Version>_asymKeyNotRead_<Datum>.d3 (siehe Kapitel 5)
<p><df_qes><completion_of_qes> <completion_asymm_pukreadable> TC: asymm. Schlüssel Signatur Schlüssel: vorhanden, öffentliche Schlüssel auslesbar</p>	<supported>	Wert aus YES NO YES → diese Komplettierungsvariante wird unterstützt NO → wird nicht unterstützt
	<apdu_list>	eGK_TLK_<Hersteller>_<Version>_asymPukRead_<Datum>.d3 (siehe Kapitel 5)

Element	Attribut	Wert
<code><df_qes><completion_of_qes></code> <code><completion_asymm_pukcertificate></code> TC: asymm. Schlüssel Signaturschlüssel: vorhanden, öffentliche Schlüssel im Gütesiegel	<code><supported></code>	Wert aus YES NO YES → diese Komplettierungsvariante wird unterstützt NO → wird nicht unterstützt
	<code><apdu_list></code>	eGK_TLK_<Hersteller>_<Version>_<asymPukCert_<Datum>.d3 (siehe Kapitel 5)
<code><df_qes><completion_of_qes></code> Gefordert ist die Kennung des Zertifizierungsdiensteanbieters der die eGK mit der QES Vorphersonalisierungsvariante ausgestattet hat.	<code><kennung_zda_vp></code>	Kennung des Zertifizierungsdiensteanbieters
<code><df_qes><completion_of_qes></code> <code><tc_symm_DF></code>	<code><enc_key></code>	Ver- und Entschlüsselungsschlüssel
	<code><mac_key></code>	Schlüssel zur MAC Berechnung
<code><df_qes><completion_of_qes></code> <code><tc_asymm_PrKDF_PuKDF></code>	<code><key_reference></code>	Schlüsselreferenz
	<code><n></code>	Modulus
	<code><d></code>	Privater Exponent
	<code><e></code>	Öffentlicher Exponent
<code><new_application></code> <code><load_application_apdus></code> Ausgangsbasis für die Datensequenz der neuen Anwendung ist [gemXML_eGKDaten] der eGK_TestLaborkartenDatenVx.x.x.xml Datei Hauptknoten <code><anwendungstest></code> zu entnehmen.	<code><apdu></code>	Herstellerindividuelle Kommando-sequenz zum Nachladen einer Anwendung via LOAD APPLICATION. Der Einsatz des „Command Chaining“ ist zwingend.

4.2.4 Template `<unterbrechungstest>`

Im Rahmen des Unterbrechungstest müssen eine Reihe von Objekten (Files, Verzeichnis, Schlüssel, PIN) nach dem Löschen neu angelegt werden. Die APDU-Sequenzen sind nach den Vorgaben in [gemXML_eGKDaten] hier einzubringen. Weitere Hinweise sind [gemXML_eGKTpl] zu entnehmen.

Tabelle 6: Hauptknoten `<unterbrechungstest>`

4.2.1 Objekte im Root (MF)

4.2.2 Objekte im DF.TestB

4.2.3 Objekte im DF.TestR

4.2.4 Objekte im DF.TestA1

4.2.5 Objekte im DF.TestAppl_x / DF.TestDF_x

4.2.6 ~~Objekte im DF.TP_1 bis DF.TP_7~~

5 Initialisierungsdateien

Für die Durchführung der Tests ist es zwingend notwendig, dass die Testlaborkarten re-initialisiert werden können, da einige Testscripte die Kartenstruktur ändern bzw. Dateninhalte z. B. im Fehlerfall irreversibel verändert werden. In einem solchen Fall muss das Testsystem in die Lage versetzt werden, die eGK wieder in ihren Ursprungszustand bringen zu können.

Die Durchführung einer Reinitialisierung erfolgt durch Ausführung einer vom Hersteller bereitzustellenden Liste von APDUs, die in einer Textdatei (mit der Endung .d3) abgelegt werden müssen. Nach der Ausführung dieser Kommandosequenz muss sich die Karte wieder im Ausgangszustand befinden.

5.1 Übersicht der Image-Files (d3-Files)

Die d3-Files enthalten ein Image, das vor den Tests abhängig vom Prüfgebiet in die eGK übertragen werden muss. Diese d3-Files müssen und können nur vom Hersteller erstellt werden. Nachfolgend eine Übersicht der d3-Files inklusive der Datenbasis für diese Dateien.

Tabelle 7: Übersicht der d3-Files

Prüfgebiet	d3-File Name* / Datenbasis
Basistest	eGK_TLK_<Hersteller>_<Version>_Initfile_<Datum>.d3 TLK_Daten / <basistest> und TLK_Template / <card_capabilities>
Anwendungstest	eGK_TLK_<Hersteller>_<Version>_Appl_<Datum>.d3 [gemSpec_eGK_P2] und TLK_Template / <card_capabilities>
Anwendungstest / QES Komplettierung	eGK_TLK_<Hersteller>_<Version>_symKeyNot_<Datum>.d3 TLK_Template / <anwendungstest> <mf> <df_qes> <completion_of_qes> <completion_symm_keynotavailable>
Anwendungstest / QES Komplettierung	eGK_TLK_<Hersteller>_<Version>_symPukRead_<Datum>.d3 TLK_Template / <anwendungstest> <mf> <df_qes> <completion_of_qes> <completion_symm_pukreadable>
Anwendungstest / QES Komplettierung	eGK_TLK_<Hersteller>_<Version>_symPukCer_<Datum>.d3 TLK_Template / <anwendungstest> <mf> <df_qes> <completion_of_qes> <completion_symm_pukcertificate>
Anwendungstest / QES Komplettierung	eGK_TLK_<Hersteller>_<Version>_asymKeyNot_<Datum>.d3 TLK_Template / <anwendungstest> <mf> <df_qes> <completion_of_qes> <completion_asymm_keynotavailable>
Anwendungstest / QES Komplettierung	eGK_TLK_<Hersteller>_<Version>_asymPukRead_<Datum>.d3 TLK_Template / <anwendungstest> <mf> <df_qes> <completion_of_qes> <completion_asymm_pukreadable>
Anwendungstest / QES Komplettierung	eGK_TLK_<Hersteller>_<Version>_asymPukCert_<Datum>.d3 TLK_Template / <anwendungstest> <mf> <df_qes> <comple-

Prüfgebiet	d3-File Name* / Datenbasis
	tion_of_ges> <completion_asymm_pukcertificate>
Unterbrechungstest / Nur MF	eGK_TLK_<Hersteller>_<Version>_UbTest_<Datum>.d3 TLK_Template / <unterbrechungstest> <mf>

* Die „Platzhalter“ sind durch folgende Werte zu ersetzen

<Hersteller> Name des Herstellers.

<Version> Die Version muss sich auf eine COS Version beziehen und dem hexadezimal kodierten Pre-Issuing Data Object im ATR entsprechen (siehe Tabelle 1) und enthält zusätzlich die Kartenversion (Patchversion).

<Datum> Erstellungsdatum des d3-Files (jjjjmmtt).

5.2 Aufbau und Struktur eines Image-Files (d3-Files)

d3-Files beinhalten eine Auflistung von Chipkartenkommandos (APDUs), mit deren Hilfe Struktur und Inhalte in eine eGK eingebracht werden können. Für das Einbringen dieser Informationen in eine eGK können auch Chipkartenkommandos benutzt werden, die nicht Bestandteil der [gemSpec_eGK_P1] sind.

Chipkartenkommandos müssen in dem d3-File entsprechend dem folgenden Format gespeichert werden:

```
{
  (LOAD)
  {
    (APDU_1) = 00 01 02 03;
    (APDU_2) = 00 01 02 03 04 05;
    :
    (APDU_n) = 00 01 02 03;
  }
}
```

Diese APDU-Liste wird durch die Testskripte von oben nach unten abgearbeitet, also wird jede APDU an die Karte gesendet. Testkarten sollten mit speziellen Testkommandos ausgestattet sein. Mit diesen soll es möglich sein, die Karte zurückzusetzen, um sie dann wieder neu aufbauen zu können. Alle benötigten Kommandos, um eine Karte zurückzusetzen und sie anschließend wieder aufzubauen, müssen in der korrekten Reihenfolge in dieser APDU-Liste enthalten sein.

Die APDU-Liste kann Sonderkommandos enthalten. Diese sind durch folgende Syntax charakterisiert:

00 00 00 00 xx

Für xx == 00 erfolgt ein COLD RESET der Karte. ~~Weitere Sonderkommandos können falls nötig in Absprache mit der gematik verwendet werden.~~

Für xx == 01 wird die Antwort des nachfolgenden Kommandos nicht durch das Testsystem ausgewertet.

Weitere Sonderkommandos können – falls nötig – in Absprache mit der gematik verwendet werden.

Anhang A

A1 – Abkürzungen

Kürzel	Erläuterung
ATR	Answer to Reset
AUT	Authentifizierung
BÄK	Bundesärztekammer
BCD	Binär kodierte Dezimalzahl
COS	Card Operating System
CVC	Card Verifiable Certificate
DF	Dedicated File
EF	Elementary File
eGK	elektronische Gesundheitskarte
EHIC	Europäische Krankenversichertenkarte
ENC	Verschlüsselung (Encryption)
HBA	Heilberufsausweis (auch HPC)
HPC	Health Professional Card (auch HBA)
ICCSN	Integrated Circuit Card Serial Number
MF	Master File
OCSP	Online Certificate Status Protocol
PIN	Persönliche Identifikationsnummer
PuK	Public Key (öffentlicher Schlüssel)
PUK	Pin Unblocking Key
PrK	Private Key (privater Schlüssel)
QES	Qualifizierte Elektronische Signatur
SMC	Security Module Card
TLV	Tag Length Value
XML	Universelle Datenbeschreibungssprache (Extensible Markup Language)

A2 - Glossar

Das Projektglossar wird als eigenständiges Dokument zur Verfügung gestellt.

A3 - Abbildungsverzeichnis

Abbildung 1: Einordnung Tests und Testphasen	6
Abbildung 2: Layout einer Testlaborkarte vom Typ eGK	9

A4 - Tabellenverzeichnis

Tabelle 1: ATR Werte (Empfehlungen)	11
Tabelle 2: Zugelassene Transport-PIN-Verfahren	11
Tabelle 3: Hauptknoten <card_capabilities>	16
Tabelle 4: Hauptknoten <basistest>	17
Tabelle 5: Hauptknoten <anwendungstest>	18
Tabelle 6: Hauptknoten <unterbrechungstest>	20
Tabelle 7: Übersicht der d3-Files	22

A5 - Referenzierte Dokumente

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

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[gemFKCMS_PIN]	gematik: Einführung der Gesundheitskarte – Beschreibung der PIN- und PUK-Verfahren für die eGK
[gemPVo_eGK]	gematik: Einführung der Gesundheitskarte – Prüfvorschriften zum Test der eGK
[gemSpec_eGK_P1]	gematik: Einführung der Gesundheitskarte – Die Spezifikation der elektronischen Gesundheitskarte, Teil 1: Spezifikation der elektrischen Schnittstelle
[gemSpec_eGK_P2]	gematik: Einführung der Gesundheitskarte – Die Spezifikation der elektronischen Gesundheitskarte, Teil 2: Grundlegende Applikationen
[gemSpec_TLK_eGK], [gemSpec_TLK_HBA], [gemSpec_TLK_SMC]	gematik: Einführung der Gesundheitskarten – Spezifikation für Testlaborkarten: eGK bzw. HBA bzw. SMC
[gemTSp_eGK]	gematik: Einführung der Gesundheitskarte – Testspezifikation eGK

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[gemTSp_UTB]	gematik: Einführung der Gesundheitskarte – Testspezifikation Unterbrechungstests
[gemXML_eGKDaten]	gematik: eGK_TestLaborKartenDaten.xml eGK_TestLaborKartenDatenV2.1.1.xml
[gemXML_eGKTpl]	gematik: eGK_TestLaborKartenTemplate.xml eGK_TestLaborKartenTemplateV2.1.2.xml
[gemZul_eGK]	gematik: Einführung der Gesundheitskarte - Zulassung von dezentralen Komponenten in der Telematikinfrastruktur (eGK)

Weitere Referenzierungen:

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[DIN66291-1]	DIN V66291-1 (2000): Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG/SigV Teil 1: Anwendungsschnittstelle
[HPC-P1]	German Health Professional Card and Security Module Card Part 1: Commands, Algorithms and Functions of the COS Platform
[HPC-P2]	German Health Professional Card and Security Module Card Part 2: HPC Applications and Functions
[HPC-P3]	German Health Professional Card and Security Module Card Part 3: SMC Applications and Functions
[RVO2006]	Bundesgesetzblatt I (2006) vom 10.10.2006, Seite 2199 ff.: Verordnung über Testmaßnahmen für die Einführung der elektronischen Gesundheitskarte in der Fassung vom 5. Oktober 2006