Einführung der Gesundheitskarte

Implementierungsleitfaden zur Einbindung der eGK in die Primärsysteme der Leistungserbringer

Version:	1.6.0
Revision:	
Stand:	21.10.2013
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemLF_Impl_eGK]

Dokumentinformationen

Änderungen zur Version 1.5.0

Festlegungen zur Verwendung der VSD-Versionen.

Hinweis zur XML-Verarbeitung.

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
0.0.1	26.04.11		initial	sko
0.0.3	29.04.11	1,2, 3-		SiSCH
0.0.5	03.05.11	Allg.	Überarbeitung	sko
0.0.7	13.05.11	3.7	Verarbeitungshinweise /	SiSCH
0.0.8	18.05.11		Mobiles Kartenterminal	SiSCH
0.0.9	25.05.11		Reviewkommentare eingearbeitet	SiSCH
0.9.0	30.05.11		zur Freigabe empfohlen	QM
	21.06.11	2.4	ext. Kommentar eingearbeitet (Hinweis SRQs verdeutlicht, Mappingdokument der DKG für KIS ergänzt)	TST
1.0.0	22.06.11		freigegeben	gematik
1.0.1	04.07.11	3.2	Versionsnummer der G1 u. G1plus Karten	SiSCH
1.1.0	05.07.11		freigegeben	gematik
1.1.1	21.07.11	3.2	Verdeutlichung der Versionsnummern Unterschied G1 zu G1 plus	SiSCH
1.2.0	21.07.11		freigegeben	gematik
1.2.1	17.11.11	3.2 – 3.7	Hinweise für den Umgang mit veralteten Gesund-heitskarten (keine G1-Karten bzw. VSD-Schema < 5.1.0) Hinweise auf bekanntgewordene Problemfälle	SiSCH
1.3.0	21.11.11		freigegeben	gematik
1.3.1	24.4.12		Read Record Extented Length	SiSCH
1.4.0	26.04.12		freigegeben	gematik
1.4.1			eGK Generation 2 VSD-Schema 5.2.0 besonders zu beachtender Datenfelder	SiSCH

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.4.9	08.03.13		zur Abstimmung freigegeben	gematik
1.4.10	09.04.13	A5.2	Link zur eGK->KVK Mappingtabelle	SiSCH
1.5.0	06.09.13		freigegeben	gematik
1.5.1	8.10.13	3.4 4.3	Hinweis zur XML-Verarbeitung Anpassung an GSV-Beschluss	SiSCH
1.6.0	21.10.13		freigegeben	gematik

Inhaltsverzeichnis

1 Zusammenfassung

Mit dem Basis-Rollout der elektronischen Gesundheitskarte (eGK) entsteht für die Anbieter von Primärsystemen (PS) die Aufgabe, neben dem Einlesen der Krankenversichertenkarten (KVK) auch das Einlesen der eGK zu ermöglichen. Zwischen den beiden Kartentypen bestehen gravierende Unterschiede, sowohl, was die Hardware betrifft (Speicher- versus Prozessorkarte) als auch bezüglich der Datenstruktur. Während die Daten auf der KVK ASN.1-kodiert in einem einzigen linearen File abgelegt sind, besitzt die eGK ein hierarchisches Filesystem und verwendet für die Fachdaten, welche auf der eGK gzip-komprimiert abgelegt sind, das XML-Format. Der Einlesevorgang für die eGK muss daher anderen Algorithmen folgen als der für die KVK.

Das vorliegende Dokument liefert eine Zusammenfassung der Arbeitsvorgänge, die zum Lesen der Versichertenstammdaten von einer eGK nötig sind, und gibt Hinweise, wo man detaillierte Informationen finden kann.

Elektronische Gesundheitskarten, die im Basis-Rollout ausgegeben werden, sind vollständig ausgestattet, um für die später geplanten Online-Verfahren verwendet zu werden. Sie enthalten neben den Versichertenstammdaten weitere (noch leere) Dateien und verschiedene Zertifikate und private Schlüssel, die für die Authentisierung und Aktualisierung der eGK nötig werden. Diese Dateien und Features sind nicht Gegenstand des vorliegenden Dokuments, sie werden erst im Online-Rollout relevant.

2 Einführung

2.1 Zielsetzung und Einordnung des Dokuments

Das vorliegende Dokument dient als Leitfaden zur Implementierung von Routinen zum Lesen der Versichertenstammdaten von der eGK in die Software für Praxisverwaltungs- (PVS) und Krankenhausinformationssysteme (KIS), im Weiteren als Primärsystem (PS) bezeichnet.

Zunächst wird erläutert, wie ein PS erkennen kann, ob es eine KVK oder eine eGK vor sich hat. Dann werden die Arbeitsschritte zum Lesen der Versichertenstammdaten erläutert. Es folgen Übersichten zur Dateistruktur der eGK und die erforderlichen Kommandos.

Dieses Dokument soll Entwicklern die Möglichkeit eröffnen, schnell und ohne Detailkenntnis der Spezifikationen zur Einführung der eGK ein Lesen der Versichertenstammdaten von der eGK zu realisieren. Für weiterführende Informationen verweisen wir auf die von der gematik herausgegebenen Spezifikationen.

2.2 Zielgruppe

Dieses Dokument sollte von Personen gelesen werden, die mit der Entwicklung von Software für Primärsysteme befasst sind und diese um die Fähigkeit zum Einlesen der Versichertenstammdaten einer eGK erweitern wollen. Es wird vorausgesetzt, dass bei den Entwicklern Basiskenntnisse beim Umgang mit Smartcards vorhanden sind, z. B. Erfahrung mit dem Einlesen von KVK.

2.3 Geltungsbereich

Das vorliegende Dokument ist maßgeblich für den oben genannten Personenkreis bestimmt. Es enthält keine neuen normativen Aussagen, sondern eine Beschreibung dessen, was die Spezifikationen zur eGK fordern bzw. eine eGK liefern kann und worauf sich die Entwickler einzustellen haben. Es macht keine Angaben darüber, wie diese Daten anschließend im PS weiterverarbeitet werden.

Der Implementierungsleitfaden bezieht sich auf die Releases 0.5.2 und 0.5.3 für den Basis-Rollout. Normative Festlegungen zu den technischen Details der Schnittstelle sind den im Anhang A5 aufgeführten Dokumenten zu entnehmen.

2.4 Arbeitsgrundlagen

Grundlagen dieses Leitfadens sind die Spezifikationen der gematik zur elektronischen Gesundheitskarte. Im Basis-Rollout sind zwei Releases – 0.5.2 und 0.5.3 – zu beachten. Für eGK der Generation 1 gilt Release 0.5.2, für Generation 1plus das Release 0.5.3.

Beide Releases unterscheiden sich in der Dateistruktur auf der eGK, allerdings – abgesehen von EF.Version - bei solchen Dateien, die für den Basis-Rollout noch keine Rolle spielen. Das Format und der Inhalt der Versichertenstammdatendateien sind in beiden Releases gleich.

Es gelten folgende Spezifikationen, einschließlich der jeweils zugehörigen SRQs:

- Fachkonzept u. Facharchitektur Versichertenstammdatenmanagement (VSDM) ([gemFK_VSDM], [gemFA_VSDM]) beschreiben Inhalt und Struktur der drei Stammdatendateien mit den Daten des Versicherten
- Speicherstrukturen der eGK für Gesundheitsanwendungen [gemeGK_Fach] legt den Aufbau der nicht-fachgebundenen Datenstrukturen fest
- Mappingtabelle eGK → KVK [eGK-KVK]
- Mapping für den KIS-Bereich auf die § 301-Daten: "Mapping von § 301-Aufnahmesatz und eGK-VSD Version 5.1.0", [tk_r0_profil_vsdm_2009-12-17]
- MKT-Spezifikation inkl. CT-API [MKT]
- eGK-Spezifikationen [gemSpec_eGK_P1], [gemSpec_eGK_P2]

Im Teil 1 der eGK-Spezifikation [gemSpec_eGK_P1] werden die Basiskommandos, die Grundfunktionen des Betriebssystems sowie die grundlegenden Sicherheitsfunktionen und -algorithmen (hard facts) detailliert beschrieben.

Im Teil 2 der eGK-Spezifikation [gemSpec_eGK_P2] werden die anwendungsspezifischen Strukturen der eGK beschrieben. Dieser Teil enthält die Spezifikationen für die Dateistrukturen der Pflichtanwendungen und der zugehörigen Datenelemente, die bei der Initialisierung und Personalisierung in die eGK geladen werden.

Es wird explizit auf die Notwendigkeit der Beachtung der von der gematik zu den Spezifikationen veröffentlichten SRQ/FAQ hingewiesen.

Die Spezifikationen für Karten der Generation 2 sind in mehrere Teile aufgeteilt:

- Spezifikation des Kartenbetriebssystems [gemSpec_COS]
- Spezifikation der Anwendungsstrukturen [gemSpec_ObjSys]
- Speicherstrukturen der eGK für die TI-Plattform [gemSpec_eGK_Fach_TIP]
- Speicherstrukturen der eGK für die Fachanwendung VSDM [gemSpec_eGK_Fach_VSDM]

Da alle im Basis-Rollout notwendigen Befehle wie auch die verwendeten Dateistrukturen in der Kartengeneration 2 identisch zu denen der Generation 1 umgesetzt wurden, wird im folgenden Text auf eine Verlinkung zur Spezifikation der Kartengeneration 2 verzichtet.

2.5 Abgrenzung des Dokuments

Es wird nur auf das Einlesen der Versichertenstammdaten eingegangen, die im ungeschützten Bereich der eGK liegen. Darüber hinaus gehende Vorgehensweisen, die erst beim Online-Rollout relevant werden, sind nicht Gegenstand dieses Dokumentes. Es

macht auch keine Angaben darüber, wie diese Daten anschließend im PS weiterverarbeitet werden.

2.6 Methodik

Die Abläufe beim Umgang mit einer eGK werden im Vergleich zu denen im Umgang mit einer KVK dargestellt, um den Entwicklern, welche Erfahrung mit einer KVK haben, den Umstieg zu erleichtern.

Die wesentlichen Abläufe werden neben der verbalen Beschreibung auch durch Aktivitätsdiagramme dargestellt. Technische Use Cases und Klassendiagramme erübrigen sich hier. Für die XML-Strukturen und Schnittstellenbeschreibungen wird auf die VSDM-Facharchitektur [gemFA_VSDM] und auf die Schemadefinitionen [gem_Schemadateien] gemäß Dokumentenlandkarten [gemDokLK] verwiesen.

3 Einlesealgorithmus

In diesem Kapitel wird dargestellt, wie beim Einlesen der Versichertenstammdaten von der eGK vorzugehen ist. Die Ablaufdiagramme sind in Kapitel 3.5 zu finden.

3.1 Kartenterminaltyp ermitteln

Da sowohl die Prozessschritte als auch die Kommandofolgen bei der Verwendung von mobilen und stationären Kartenterminals unterschiedlich sind, ist es empfehlenswert zu Beginn einer Leseroutine den Kartenterminaltyp zu ermitteln.

Mobile und stationäre Kartenterminals liefern unterschiedliche Antwortdaten auf das Kommando „Reset CT“ im Erfolgsfall zurück. Wird als Antwort '9000' zurückgegeben, handelt es sich um ein stationäres Kartenterminal. Wird hingegen '9500' zurückgegeben, handelt es sich um ein mobiles Kartenterminal.

Nach der Ermittlung des Kartenterminaltyps muss dann der entsprechende Ablauf für das Einlesen der Daten gewählt werden, da die Kartensimulation eines mobilen Kartenterminals nur einen reduzierten Befehlsvorrat unterstützt, unterscheiden sich die Befehlssequenzen beim Einlesen vom mobilen und stationären Kartenterminal.

3.2 Versichertenkarte Lesen (mobil)

Da ein PS über einen längeren Zeitraum hinweg sowohl eGK als auch KVK verarbeiten wird, muss eine Leseroutine zunächst den Kartentyp prüfen (eGK, KVK).

3.2.1 Kartentyp ermitteln

Als erstes Unterscheidungskriterium bietet sich hier die unterschiedliche Technologie der KVK und der eGK an. Die KVKn sind reine Speicherkarten; sie verwenden ein synchrones Übertragungsprotokoll. Die eGKn hingegen sind Prozessorkarten und verwenden ein asynchrones Übertragungsprotokoll.

Das zur eingesteckten Karte gehörige Übertragungsprotokoll wird in den Antwortdaten des Kommandos “Request ICC” von der CT-API angezeigt. Wird als Antwort '9000' zurückgegeben, handelt es sich um eine synchrone Karte. Wird hingegen '9001' zurückgegeben, ist es eine asynchrone Karte. Das Kartenterminal gibt als Antwort '6200' zurück, wenn in dem mobilen Kartenterminal keine Karte (eGK oder KVK) gespeichert ist.

3.2.2 Daten lesen

Für das Lesen von einem mobilen Kartenterminal stellt das Kartenterminal eine eingeschränkte Simulation einer elektronischen Gesundheitskarte zur Verfügung, daher ist der Ablauf weniger komplex als bei der direkten Kommunikation mit einer eGK in einem stationären Kartenterminal. Die Kommandofolge kann als statische Abfolge betrachtet werden (siehe [gemSpec_mobKT#7.2]).

Die folgende Kommandosequenz für das Auslesen der Daten einer eGK muss verwendet werden.

Kommando	APDU	Bemerkung
SELECT FILE (HCA)	00 a4 04 0c 06 d2 76 00 00 01 02	eGK-Anwendung selektieren
READ BINARY (StatusVD)	00 b0 8c 00 00 oder 00 b0 8c 00 00 00 00	Statusdaten, Erfassungsdatum und Zulassungsnummer lesen
READ BINARY (Personal Data)	00 b0 81 00 00 00 00	Personendaten lesen
READ BINARY (Insurance Data)	00 b0 82 00 00 00 00	Allgemeine Versicherungsdaten lesen
ERASE BINARY	00 0e 00 00	unmittelbar zuvor übertragenen Datensatz löschen (Empfang quittieren)
EJECT ICC	20 15 01 00 01 01	Deaktivieren des elektrischen Interface Chipkarte auswerfen

Tabelle 1: Kommandosequenz für das Auslesen der Daten einer eGK

Wurde das mobile Kartenterminal für die Übertragung zum PS noch nicht aufgeschlossen (PIN-Eingabe erforderlich), so wird vom Kommando READ BINARY und ERASE BINARY '6900' (Kommando nicht erlaubt) zurück gegeben.

Für das Auslesen der eGK sind drei READ-BINARY-Befehle nötig. Zusätzlich muss das Kommando READ BINARY mit erweiterter Längenangabe (extended Length) gesendet werden. Die Methode ein READ BINARY mehrfach mit fortschreitendem Offset zu senden, wird nicht unterstützt. Das Lesen des Status kann auch mit einfacher Länge erfolgen, da die Antwort geeignet kurz ist.

3.2.3 Erweiterungen der Datentypen bei der Übertragung

Bei den gelesenen Daten handelt es sich um gezippte XML-Dateien, wie sie in der eGK-Spezifikation [gemeGK_Fach], [gemSpec_eGK_Fach_VSDM] definiert sind. Lediglich bei den Daten aus StatusVD werden diese um die Zulassungsnummer und den Erfassungszeitpunkt ergänzt. Hiefür wird die folgende TLV-Struktur definiert, welche die Tag-Kodierung aus der mobile-KVT-Spezifikation übernimmt. Die Reihenfolge der Tags soll eingehalten werden:

Pos.	Herkunft	Tag	Länge	Inhalt
1	eGK	A0	xx	StatusVD, wie aus der eGK ausgelesen.
2	Term.	91	xx	Einlesedatum im Format TTMMJJJJ (ASCII)
3	Term.	92	xx	Zulassungsnummer der mobilen Kartenterminals (ASCII)
4	Term.	93	01	XOR-Prüfsumme über die Tags 91,92 und Tag und Länge von Tag 93

Tabelle 2: TLV-Struktur EF.StatusVD (mobil)

3.3 Versichertenkarte Lesen (stationär)

Da ein PS über einen längeren Zeitraum hinweg sowohl eGK als auch KVK verarbeiten wird, muss eine Leseroutine zunächst den Kartentyp prüfen (eGK, KVK).

3.3.1 Kartentyp ermitteln

Als erstes Unterscheidungskriterium bietet sich hier die unterschiedliche Technologie der KVK und der eGK an. Die KVKn sind reine Speicherkarten; sie verwenden ein synchrones Übertragungsprotokoll. Die eGKs hingegen sind Prozessorkarten und verwenden ein asynchrones Übertragungsprotokoll.

Das zur eingesteckten Karte gehörige Übertragungsprotokoll wird in den Antwortdaten des Kommandos "Request ICC" von der CT-API angezeigt. Wird als Antwort '9000' zurückgegeben, handelt es sich um eine synchrone Karte. Wird hingegen '9001' zurückgegeben, ist es eine asynchrone Karte.

3.3.2 Applikation selektieren

Ein zweites Unterscheidungskriterium sind die Applikationen, die auf der Karte vorhanden sind. Im Falle einer KVK wird hierzu ein "Select" mit der Application-ID (AID) der KVK ('D27600000101') durchgeführt. Nach erfolgreicher Selektion können die Daten gelesen werden, und die Karte kann anschließend dem Kartenleser entnommen werden.

Bei einer eGK ist der Ablauf komplexer. Zunächst wird auch hier mit dem "Select"-Kommando geprüft, ob die erwartete Applikation vorhanden ist. Seit der eGK-Generation 1 besitzt die Root-Applikation die AID='D2760001448000'. Ist sie nicht vorhanden, so sollte die Leseroutine mit einer Fehlermeldung beendet werden. Da es sich in einem Fehlerfall nicht um eine Gesundheitskarte der Generation 1 oder 2 handelt, muss davon ausgegangen werden dass es sich bei dieser Karte um eine veraltete Gesundheitskarte handelt. Der Anwender sollte auf diesen Sachverhalt hingewiesen werden, damit der Versicherte einen Austausch seiner Karte veranlassen kann.

Beispielhafte Fehlermeldung:

> „Keine gültige Gesundheitskarte. Bitte überprüfen und ggf. austauschen lassen."

Im Erfolgsfall hingegen muss überprüft werden, ob dem PS die Version der vorliegenden eGK bekannt ist und eine weitere Verarbeitung möglich ist. Die Version der eGK ist in der Datei EF.Version definiert. Die Records dieser Datei geben Auskunft über die Spezifikationen der eGK, welche sich aus mehreren Teilen zusammensetzt. In Kapitel 0 finden sie eine Aufstellung der unterschiedlichen eGKs. Aus der Kombination der Versionsangaben ist das Verhalten der Karte eindeutig bestimmbar.

Nachdem die Applikation erfolgreich selektiert wurde, sind die Dateninhalte zweier Dateien auszulesen. Dies wird in Kapitel 3.3.6 beschrieben.

3.3.3 Pufferlänge ermitteln

Die eGK ist in der Lage, auch Datenblöcke mit mehr als 256 Byte auszulesen (extended length). Aus Performancegründen empfiehlt es sich, dies auch zu tun. Wie viele Bytes genau mit einem Kommando gelesen oder geschrieben werden können, steht in der Da-

tei EF.ATR, siehe [gemSpec_eGK_P2#6.2.1], nicht im ATR; es gibt keine standardmäßig vorgegebene Länge. Daher muss für eine Leseroutine, die das “Read-Kommando” mit Längen über 256 Byte nutzen will, zuerst die maximal mögliche Anzahl der Antwortdaten der Karte ermittelt werden.

Hierzu muss die Datei EF.ATR gelesen und ausgewertet werden. In dieser Datei befindet sich am Anfang ein DER-TLV kodiertes Datenobjekt (Tag='E0'), welches wiederum vier Datenobjekte (Tag='02') mit Angaben für die maximal zulässigen Kommando- bzw. Antwortlängen der Karte enthält, und zwar:

(1) maximale Kommandolänge (Header plus Daten)

(2) maximale Antwortlänge, incl. Statuswort

(3) maximale Länge gesicherter Kommandos (Secure Messaging)

(4) maximale Länge einer gesicherten Antwort

Demnach enthält das zweite Datenobjekt die für “extended”-Kommandos wichtige Datenfeldlänge für ungesicherte Antworten. Leseoperationen dürfen nicht mehr als die angegebene Länge - 2 (für das Statuswort) anfordern. Es ist selbst als TLV kodiert, d. h. Tag = '02', dem folgt ein Byte für die Länge des Datenobjekts (Wert = 2) und zwei Bytes für die eigentliche Leselänge.

Beispiel:

Die Datei EF.ATR hat den Inhalt:
'E0 10 02 02 01 23 02 02 **02 34** 02 02 04 56 02 02 07 89 ....'

Hier ist die maximale Länge der Antwortdaten mit '0234' = 564 Byte angegeben. Ein “Read” darf demzufolge maximal 562 Byte als erwartete Datenlänge anfordern.

Hinweis:

Eine Nichtbeachtung der Maximallängen führt zu Fehlern bei der Kommandoabarbeitung. Bei der Verwendung von WildCardExtended bei READ BINARY muss auch die mögliche Antwortdatenlänge berücksichtigt werden, denn überschreitet diese die verfügbare Pufferlänge, kann die Karte mit einem Fehler ('6700') antworten.

3.3.4 Sperrstatus ermitteln

Anders als bei einer KVK ist es möglich, auf der eGK die Gesundheitskartenapplikation (HCA=Health Care Application) zu deaktivieren. Dies wird mit der Bereitstellung von Aktualisierungsmechanismen, unabhängig vom Start des Online-Rollouts, möglich. Deshalb kann das Selektieren des DF.HCA, auch bei einer zuvor erkannten eGK, mit dem Statuswort '6283' (FileDeactivated) enden. Diese Karte wurde offensichtlich durch eine Aktualisierung für nicht mehr gültig erklärt. Ein Lesen der Versichertendaten ist nicht möglich; das PS muss den Einlesevorgang mit einer aussagekräftigen Fehlermeldung abbrechen.

3.3.5 Transaktionsstatus ermitteln

Die eGK bietet die Möglichkeit, Versichertenstammdaten zu aktualisieren. Dies geschieht online durch eine Client-Applikation in Verbindung mit dem VSDD-Server einer Krankenkasse. Das PS hat damit zunächst nichts zu tun, kann allerdings auf Karten stoßen, bei denen eine Aktualisierung der Versichertenstammdaten misslungen ist. Diesen Zustand muss das PS erkennen, daher sei hier kurz darauf eingegangen.

Beim Start einer Aktualisierung setzt die steuernde Applikation ein Kennzeichen auf der Karte, den Transaktionsstatus, in der Datei EF.StatusVD (siehe [gemeGK_Fach#3.5], [gemSpec_eGK_Fach_VSDM#2.5] Tabelle 5 in diesem Dokument). Nach erfolgreichem Abschluss wird dieser Status wieder zurückgesetzt. Trifft man auf eine Karte mit „offenem“ Transaktionsstatus, so kann man davon ausgehen, dass die Aktualisierung der Versichertenstammdaten nicht beendet wurde (Karte vorzeitig gezogen, Verbindung zum Server zusammengebrochen, etc.) und die Versichertenstammdaten demzufolge nicht valide sein können. In diesem Fall ist es sinnlos, die Daten zu verarbeiten; die Karte muss zuerst erfolgreich aktualisiert werden.

In EF.StatusVD ist auch die Schemaversion der auf dieser Karte vorhandenen Versichertenstammdaten hinterlegt. Eine Prüfung auf Gültigkeit (alte Versionen dürfen für die Abrechnung nicht verwendet werden) kann schon an dieser Stelle, also vor dem eigentlichen Lesen der Daten, erfolgen. Auch in diesem Fall sollte der Anwender auf eine nicht verarbeitbare Version hingewiesen werden.

Beispielhafte Fehlermeldung:

> Die Versichertenstammdaten der Version x.y.z können nicht verarbeitet werden.

3.3.6 Daten lesen

Erst nachdem alle diese Prüfungen erfolgreich verlaufen sind, ist es ratsam, die Versichertendaten aus den Dateien EF.PD (persönliche Versichertendaten) (siehe [gemeGK_Fach#3.4], [gemSpec_eGK_Fach_VSDM#2.4]) und EF.VD (Versicherungsdaten) (siehe [gemeGK_Fach#3.2], [gemSpec_eGK_Fach_VSDM#2.2]) zu lesen. Für eine Übergangsphase werden die in der zugriffsgeschützten Datei EF.GVD abgelegten schützenswerten Versichertendaten als Duplikat ebenfalls in EF.VD abgelegt. Ob dieses Duplikat vorhanden ist, kann über die Versionsinformation aus Record 3 von EF.Version ermittelt werden sowie aus den Offsets in der Datei EF.VD selbst.

Beim Lesen der Daten aus EF.PD und EF.VD sollten die Längenangaben am Anfang der Dateien berücksichtigt werden, damit nicht benötigte Daten erst gar nicht gelesen werden. Ob die Daten aus EF.VD zunächst komplett gelesen und dann in die Teile VD und GVD zerlegt werden oder ob diese Aufteilung der Daten bereits beim Lesen erfolgen soll, muss man abwägen. Hier sei lediglich darauf hingewiesen, dass es nicht vorgeschrieben ist, die Inhalte direkt hintereinander in dieser Datei zu speichern.

Hinweis:

> Ein Lesen von Daten über den benutzten Datenbereich hinaus (EOF) wird nicht von allen Karten unterstützt. Es empfiehlt sich daher, nur die notwendigen Daten zu lesen.

3.4 Daten dekomprimieren und verarbeiten

Die Versichertenstammdaten liegen in Form von drei gzip-komprimierten XML-Dateien unverschlüsselt vor. Somit müssen die gelesenen Dateien nach dem Auslesen zuerst dekomprimiert werden. Danach können die XML-Daten verarbeitet werden. Da für die XML-Dateien keine Restriktionen gemacht wurden, ist es empfehlenswert XML-Leseroutinen zu verwenden welche die XML-Spezifikation vollständig unterstützten. Hierbei sollte besonderes die Verarbeitung mit unterschiedlichen bzw. mehrfachen Namespacedeklarationen beachtet werden. Die Schemadefinition der Versichertenstammdaten

ermöglicht es XML-Dokumente mit und ohne Namespaceprefix zu erstellen, auch in einer gemischten Form. Bei der Verarbeitung muss die Version der Daten auf Gültigkeit überprüft werden, da bei Vorliegen einer neueren Version der Stammdaten eine fehlerfreie Verarbeitung nicht gewährleistet werden kann. Ungültig gewordene Versionen dürfen für die Abrechnung nicht verwendet werden. Die Version kann aus dem Attribut „CDM_VERSION“ des Rootelementes oder dem Namespace ermittelt werden. Eine Verarbeitung unbekannter oder ungültiger Versionen der Stammdaten sollte mit einem Hinweis auf eine unbekannte Version abgebrochen werden.

Beispielhafte Fehlermeldung:

> Die Versichertenstammdaten der Version x.y.z können nicht verarbeitet werden.

3.4.1 Besonders zu beachtende Datenfelder

In den Versichertenstammdaten der eGK sind Datenfelder enthalten, welche erst nach Bereitstellung von Aktualisierungsverfahren nutzbar sind. Hierzu gehören die Felder zur Kostenerstattung und zum Zuzahlungsstatus der Versicherten. Eine Zuzahlungsbefreiung wird, wie bisher, durch ein zusätzliches Dokument nachgewiesen. Die Felder werden mit den laut Spezifikation definierten Standardwerten vorbelegt, dürfen aber derzeit nicht ausgewertet werden. Ab wann eine direkte Verarbeitung dieser Felder in den Stammdaten der eGK möglich ist, wird durch die Vertragspartner rechtzeitig bekannt gegeben. Die folgende Tabelle enthält die im Basis-Rollout nicht zu berücksichtigenden Datenfelder.

VSD	Klasse/Element	VSD-Version
VD	Kostenerstattung	≥ 5.1.0
GVD	Zuzahlungsstatus	≥ 5.1.0
GVD	Selektivverträge	≥ 5.2.0
GVD	Ruhender Leistungsanspruch	≥ 5.2.0

Tabelle 3: Im Basis-Rollout nicht zu berücksichtigende Datenfelder

Die fachliche Verarbeitung der Stammdaten kann es notwendig machen, diese Daten für die Weiterverarbeitung aufzubereiten. Hierfür haben einzelne Organisationen entsprechende Vorgaben gemacht. Die notwendigen Dokumente (Mappingtabellen) sind im Kapitel 2.4 Arbeitsgrundlagen aufgeführt.

3.5 Aktivitätsdiagramme

Mit den folgenden beiden Aktivitätsdiagrammen wird der Ablauf beim Einlesen der Daten von beiden Kartentypen noch einmal gegenübergestellt.

3.5.1 KVK lesen

Das folgende Diagramm stellt den in 3.3.2 genannten Ablauf grafisch dar:

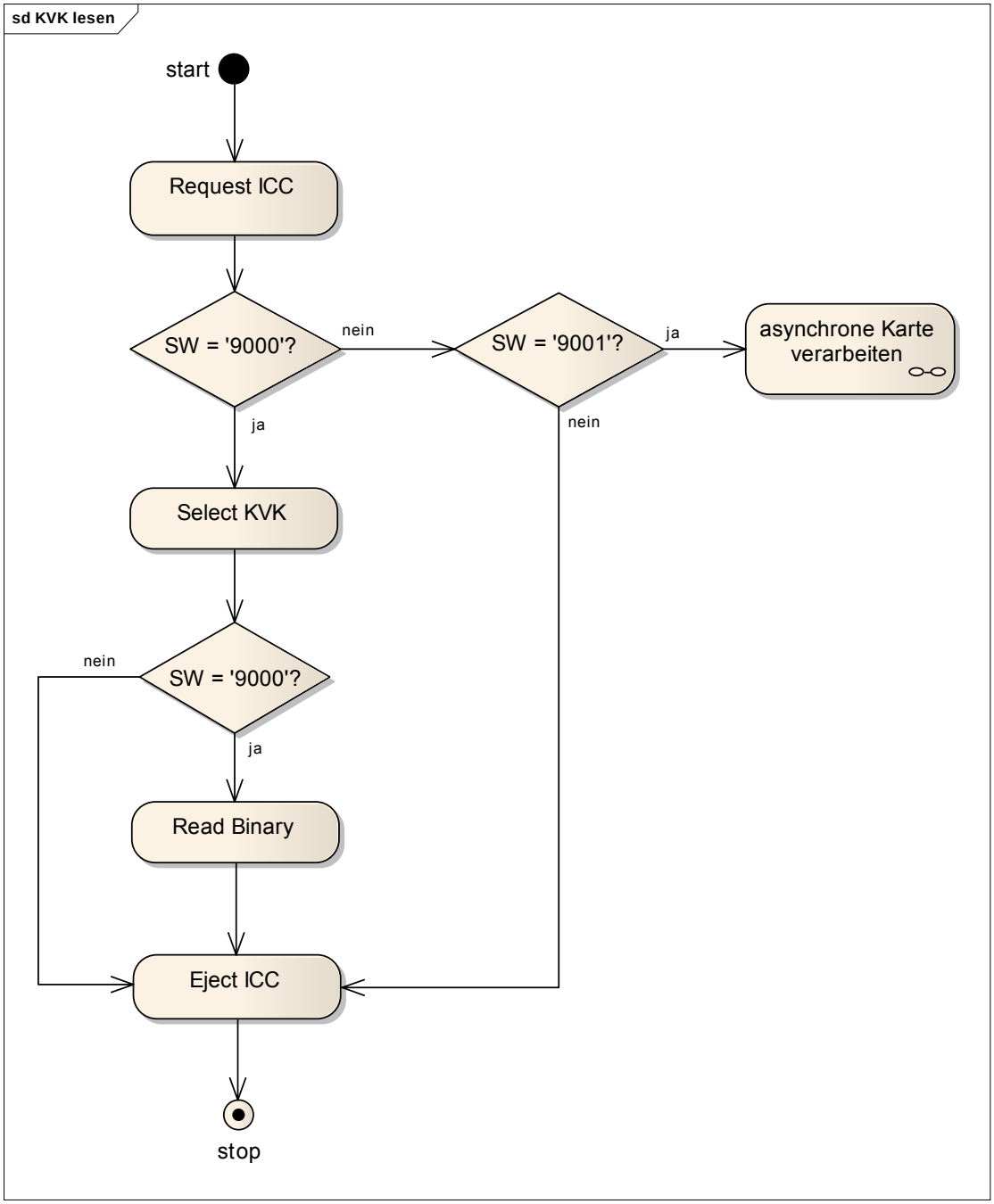

Abbildung 1: Einlesen einer KVK

3.5.2 eGK lesen

Dieses Diagramm stellt die Kartenzugriffe beim Einlesen einer eGK dar.

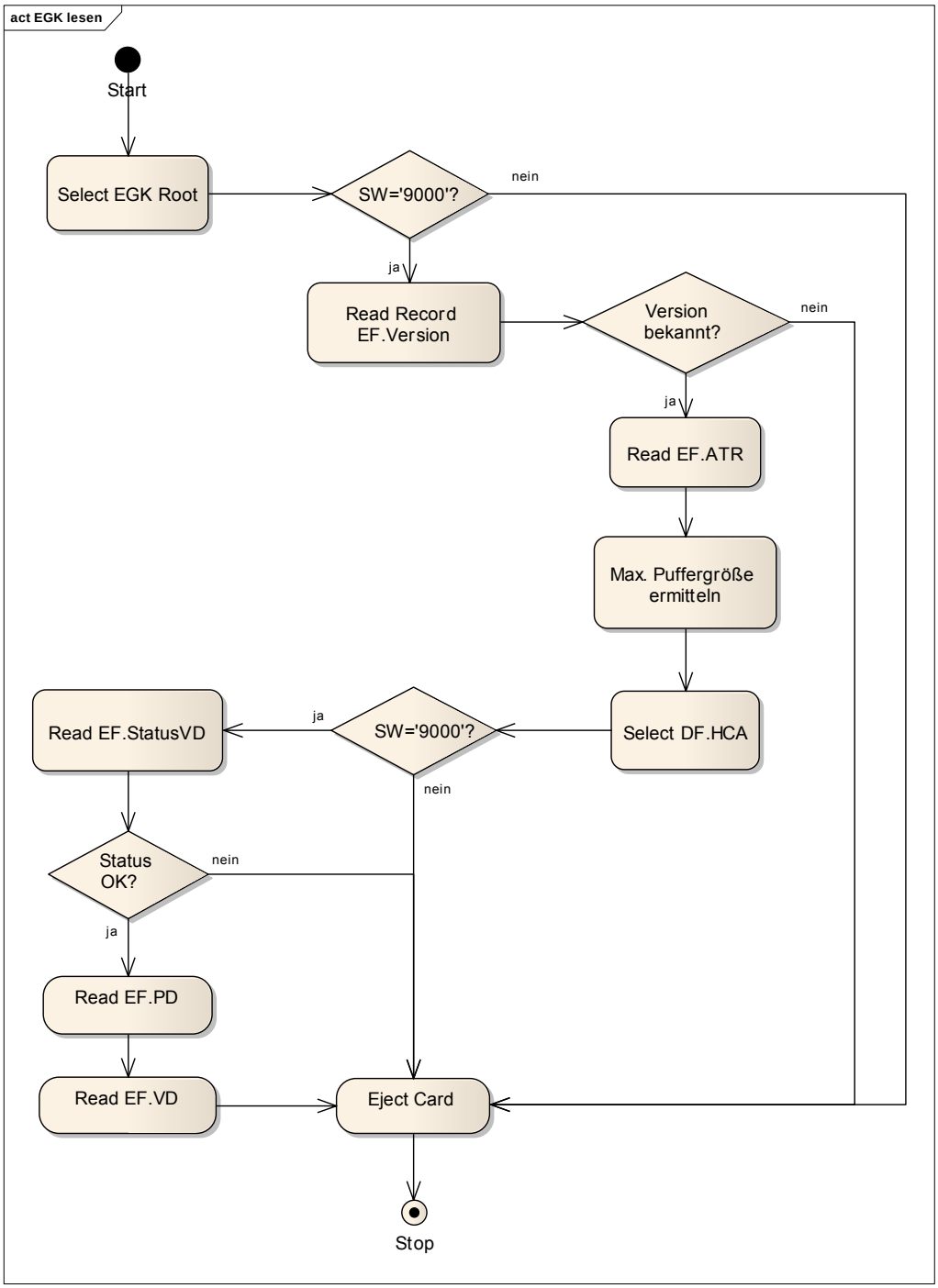

Abbildung 2: Einlesen einer eGK

4 Dateistrukturen

4.1 Übersicht

Die eGK besitzt eine hierarchische Datenstruktur. Das Wurzelverzeichnis (Masterfile, MF) enthält neben dem Verzeichnis DF.HCA (Health Care Application) für die Fachanwendungen der Gesundheitskarte weitere Verzeichnisse für Schlüssel und Zertifikate und Einzeldateien für Verwaltungszwecke, wie EF.ATR, EF.DIR, EF.GDO, EF.Version u. a.

Die genannten Datei- und Verzeichnisnamen dienen nur der menschlichen Kommunikation; die Kommandos an die Karte beinhalten hexadezimale Datei-Identifikatoren, die File IDs (FID). Nach Selektion eines Verzeichnisses können die lokal darin liegenden Dateien auch mit ihren Short File IDs (SFID) adressiert werden.

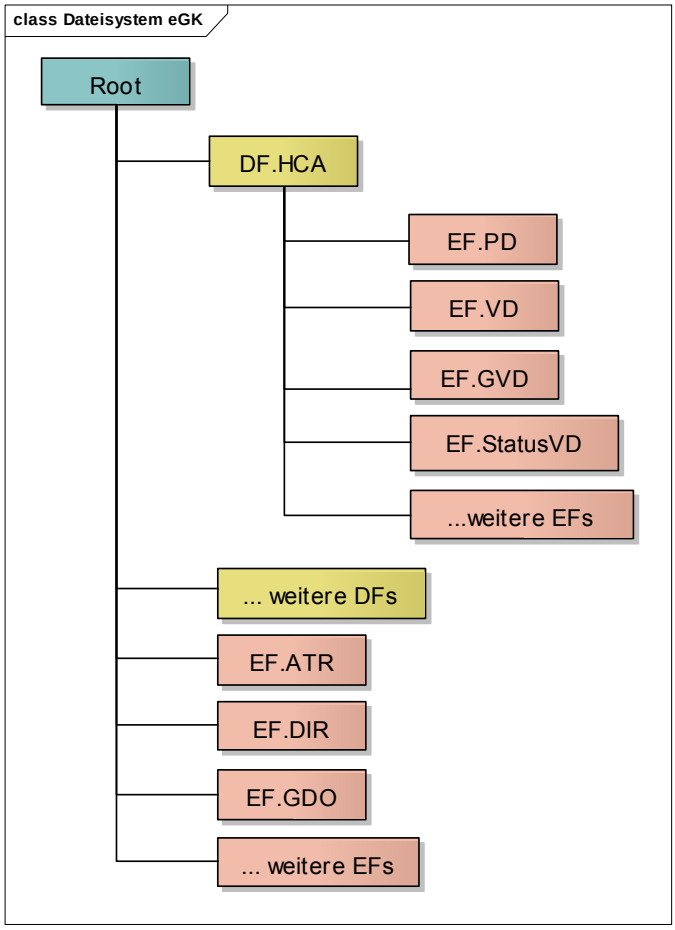

Abbildung 3: eGK Dateistruktur

4.2 Auszüge aus der Spezifikation

Dieser Abschnitt ist informativ und soll einen Überblick über die für die Implementierung relevanten Dateien geben. Die normativen Festlegungen müssen der Spezifikation [gemeGK_Fach], [gemSpec_eGK_Fach_TIP], [gemSpec_eGK_Fach_VSDM] unter Berücksichtigung aller dazu geltenden SRQ entnommen werden. Dabei sind die Ausprägungen der eGK Generation 1, 1plus und 2 zu unterscheiden.

4.2.1 Datei EF.Version

Informations-element	Länge in Byte	Typ	Initial-wert	Bemerkung
Version der unterstützten eGK Spec. Teil 1	5	BCD	siehe 4.	Siehe 1,3 und 5.
Version der unterstützten eGK Spec. Teil 2	5	BCD	siehe 4.	Siehe 1,3 und 5.
Version der unterstützten Speicherstrukturen	5	BCD	siehe 4.	Siehe 2.
Reserviert	5		0	

1. Versionsnummer des Betriebssystems und der Dateisystemkonfiguration, zu dem die Karte kompatibel ist. Dabei sind die Spezifikationen [gemSpec_eGK_P1] und [gemSpec_eGK_P2] sowie alle SRQs zu berücksichtigen, die gemäß Dokumentenlandkarte den vorgenannten Spezifikationen zugeordnet sind.
2. Versionsnummer der in dieser Spezifikation beschriebenen Strukturen, zu der die Karte kompatibel ist.
3. Die Versionsangaben in den Rekords von EF.Version sind entkoppelt von den Versionsnummern der Spezifikationsdokumente, weil neben den Dokumenten auch schnittstellenrelevante SRQs berücksichtigt werden müssen.
4. Das Informationselement wird mit der aktuellen Versionsnummer zum Personalisierungszeitpunkt vorbelegt.
5. Die Versionsinformation wird im Format XXXYYYZZZZ BCD-gepackt gespeichert. Der Wert 0x0030010002 ist als Version 3.1.2 zu interpretieren.

Tabelle 4 - Struktur der Datei EF.Version

4.2.2 Datei EF.StatusVD

Informations-element	Länge in Byte	Typ	Initial-wert	Bemerkung
Status	1	ALPHA	„0“	„1“ = Transaktionen offen
				„0“ = keine Transaktionen offen
Timestamp	14	ALPHA	siehe 1.	Timestamp der letzten Aktualisierung der VSD durch den VSDD im Format YYYYMMDDhhmmss
Version	5	BCD	siehe 2.	Schemaversionsnummer der in EF.VD,

Informations-element	Länge in Byte	Typ	Initial-wert	Bemerkung
				EF.PD und EF.GVD befindlichen Versicher-tenstammdaten. Dazu wird der Wert des Attri-buts CDM_VERSION (ist für alle drei Sche-mata immer gleich) im Format XXXYYYZZZZ eingetragen. Z. B. ist für die CDM_VERSION 7.3.1 der Wert 0x0070030001 einzutragen.
Reserviert	5	BINÄR	0	
1. Das Informationselement Timestamp wird mit dem Zeitstempel des Personalisierungszeit-punktes vorbelegt. 2. Das Informationselement Version wird mit der aktuellen Versionsnummer der XSD-Schemata zum Personalisierungszeitpunkt vorbelegt.				

Tabelle 5 - Struktur der Datei EF.StatusVD

Besonderheit bei mobilen Kartenterminals:

Vom mobilen Kartenterminal werden die Daten von EF.StatusVD um die Zulassungs-nummer und den Erfassungszeitpunkt ergänzt. Hiefür wird die folgende TLV-Struktur de-finiert, welche die Tag-Kodierung aus der mobile-KVT-Spezifikation übernimmt. Die Rei-henfolge der Tags SOLL eingehalten werden (siehe [gemSpec_mobKT#7.3]).

Pos.	Herkunft	Tag	Länge	Inhalt
1	eGK	A0	xx	StatusVD, wie aus der eGK ausgelesen.
2	Term.	91	xx	Einlesedatum im Format *TTMMJJJJ* (ASCII)
3	Term.	92	xx	Zulassungsnummer der mobilen Kartenterminals (ASCII)
4	Term.	93	01	XOR-Prüfsumme über die Tags 91, 92 und Tag und Län-ge von Tag 93

Tabelle 6 - Struktur der Datei EF.StatusVD (mobiles Kartenterminal)

4.2.3 Datei EF.PD

Informationselement	Länge in Byte	Typ	Initialwert	Bemerkung
Länge PD	2	BINÄR		Siehe 1.
PD	variabel			Siehe 2.
1. Gibt die Länge des Feldes PD in Byte an. Die maximale Länge ergibt sich aus der in [gemSpec_eGK_P2] definierten Dateigröße minus des Speicherplatzbedarfes für das Längenfeld (2 Byte). 2. Die persönlichen Versichertendaten PD selbst, werden als XML-Daten gemäß vorge-gebenem XML-Schema, gzip-komprimiert und nicht verschlüsselt innerhalb der Datei abgelegt. Der zu verwendende Zeichensatz für die fachlichen Inhalte ist ISO8859-15.				

Tabelle 7 - Struktur der Datei EF.PD

4.2.4 Datei EF.VD

Solange noch keine Authentisierung der eGK mit einem Heilberufsausweis oder einer Institutionskarte möglich ist, befindet sich eine Kopie der geschützten Daten (GVD) im ungeschützten Bereich, d. h. in der Datei EF.VD befinden sich zwei gepackte XML-Dokumente. Sie müssen nicht notwendig unmittelbar hintereinander liegen. Start- und Ende-Offsets für beide Dokumente sind in den ersten 8 Bytes der Datei EF.VD eingetragen.

Informationselement	Länge in Byte	Typ	Initialwert	Bemerkung
Offset Start VD	2	BINÄR		siehe 1.
Offset Ende VD	2	BINÄR		siehe 1.
Offset Start GVD	2	BINÄR		siehe 2.
Offset Ende GVD	2	BINÄR		siehe 2.
VD	variabel			siehe 3., siehe 5.
GVD	variabel			siehe 4., siehe 5.

1. Der Offset berechnet sich ab Start der Datei und wird in Byte angegeben. Durch die Länge des Offset-Bereiches ist der kleinstmögliche Offset-Wert 8.
2. Liegen keine geschützten Versichertendaten GVD vor, zum Beispiel im Falle einer privaten Krankenversicherung, so werden die Felder Offset Start GVD und Offset Ende GVD jeweils mit hexadezimal ‚FFFF' belegt, um das Feld GVD als leer zu markieren. Der Offset berechnet sich ab Start der Datei und wird in Byte angegeben. Durch die Länge des Offset-Bereiches ist der kleinstmögliche Offset-Wert 8.
3. Die allgemeinen Versicherungsdaten VD selbst, werden als XML-Daten gemäß vorgegebenem XML-Schema, gzip-komprimiert und nicht verschlüsselt innerhalb der Datei abgelegt. Der zu verwendende Zeichensatz für die fachlichen Inhalte ist ISO8859-15.
4. Die allgemeinen geschützten Versichertendaten GVD selbst werden als XML-Daten gemäß vorgegebenem XML-Schema gzip-komprimiert und nicht verschlüsselt innerhalb der Datei abgelegt. Der zu verwendende Zeichensatz für die fachlichen Inhalte ist ISO8859-15.
5. Die Reihenfolge der Felder VD und GVD ist durch die entsprechenden Offsets definiert und nur exemplarisch wie oben abgebildet. Die maximale Länge ergibt sich aus der in [gemSpec_eGK_P2] definierten Dateigröße für Versicherungsdaten minus dem Speicherplatzbedarf für den Offset-Bereich (8 Byte).

Tabelle 8 - Struktur der Datei EF.VD

4.3 Unterschiede der elektronischen Gesundheitskarten

Im Zuge der Einführung der elektronischen Gesundheitskarte wurden die Spezifikationen hierzu weiterentwickelt. Aktuell wird zwischen drei Kartengenerationen unterschieden. Eine Übersicht über die Unterschiedsmerkmale bietet die folgende Tabelle, aus der auch ersichtlich ist, wie zwischen den Kartengenerationen unterschieden werden kann. Der entscheidende Unterschied für den Basis-Rollout bildet das Objektsystem, welches durch den Rekord 2 in EF.Version definiert wird. Obwohl sich die Kartenbetriebssysteme der einzelnen Generationen unterscheiden, hat dies auf die Verwendung im Basis-Rollout keine Auswirkungen, da die notwendigen Befehle von allen Karten unterstützt werden.

Eine Auflistung der notwendigen Spezifikationen finden sie im Kapitel 2.4 Arbeitsgrundlagen.

	AID des MF	EF.Version	VSD-Version
G0	Nicht definiert	Nicht vorhanden	< 5.1.0
G1	'D2760001448000'	R1 ≤ 3.0.0 R2 ≤ 3.0.0 R3 ≤ 3.0.2	= 5.1.0
G1plus	'D2760001448000'	R1 = 3.0.0 R2 = 3.0.1 R3 = 3.0.3	≥ 5.1.0
G2	'D2760001448000'	R1 = 4.0.0 R2 = 4.0.0 R3 = 4.0.0	≥ 5.2.0

Tabelle 9 – Merkmale der Kartengenerationen

4.4 Unterschiede der VSD-Schemaversionen

Nachfolgend werden die Änderungen des Schemas der Version 5.2.0 im Vergleich zum bisher einzigen im Basis-Rollout verwendeten Schema in der Version 5.1.0 dargestellt (siehe [gem_Schemadateien]).

Klasse	Änderung
Adresse	Änderung der Kardinalität des Elementes „Postleitzahl“ -> Optional
Zusatzinfos GKV	Wegfall der Elemente Rechtskreis und Versichertenstatus RSA
Zusatzinfos_Abrechnung_GKV	Änderung der Kardinalität WOP
Kostenerstattung	Umbenennung der Elemente für ambulante und stationäre Kostenerstattung Aufnahme der Elemente für zahnärztliche Versorgung und veranlasste Leistungen Änderung der Kardinalität der Klasse „Kostenerstattung“ ->Optional
Zusatzinfos PKV	Wegfall aller Klassen zur PKV
Ruhender Leistungsanspruch	Aufnahme einer neuen Klasse mit den Elementen Beginn, Ende und Art des Ruhens
Selektivverträge	Aufnahme einer neuen Klasse mit den Elementen ärztliche, zahnärztliche und Art der Selektivverträge

Tabelle 10: Übersicht Änderungen der Attribute in den Klassen

Attribut / Element	Änderung
Vorname	Anpassung der Befüllungsvorschrift bei Vornamen, die nicht dem deutschen Namensrecht unterliegen
Geburtsdatum	Anpassung an die Vorgaben der DEÜV

Tabelle 11: Übersicht der Änderungen von Befüllungsvorschriften

5 Karten- und Terminalkommandos und Konstanten

5.1 Kommandos an den Kartenleser

Im Basis-Rollout werden die zugelassenen eHealth-BCS-Kartenterminals üblicherweise über die CT-API-Schnittstelle angesprochen. Diese wird von den Kartenterminalherstellern für unterschiedliche Betriebssysteme angeboten.

Der entsprechende Kommandosatz basiert auf dem CT-BCS-Befehlssatz der MKT [MKT]. Die Struktur der Kommandos folgt derjenigen für ISO 7816-4 [7816-4] APDUs.

Mobile Kartenterminals ermöglichen das Einlesen und Abspeichern von Inhalten der Versichertenkarten (eGK und KVK). Die gespeicherten Daten werden nach Anschluss an das Primärsystem ausgelesen. Die Kartenterminals emulieren hierbei den jeweiligen Kartentyp. Die mobilen Kartenterminals unterstützen nicht den vollen Umfang an Kommandos der Karten, sondern nur den für die Übertragung zum Primärsystem notwendigen Umfang. Hier ist hervorzuheben dass alle eGK-Lesekommandos zum mobilen Kartenterminal mit extended length durchzuführen sind und weder die Datei EF.ATR noch EF.Version vom mobilen Kartenterminal der Ausbaustufe 1 gespeichert werden. Des Weiteren müssen die Datensätze nach der Übertragung explizit gelöscht werden. Unterschiede bei den Kommandos werden in den nachfolgenden Beschreibungen hervorgehoben.

Zum Lesen der eGK wird nur ein Teil des Befehlssatzes benötigt. Zum Anfordern und Auswerfen der Karten werden folgende Kommandos genutzt:

Kommando	Sequenz (APDU)	Bemerkung
Reset CT	'20 11 00 00 00'	Kartenleser zurücksetzen
Request ICC1	'20 12 01 00 01 xx'	Kartenanforderung (siehe 1)
Eject ICC1	'20 15 01 00 01 xx'	Karte auswerfen (siehe 1)
xx ist durch die gewünschte Wartezeit des Kommandos zu ersetzen. Bei mobilen Kartenterminals muss als Wartezeit '01' angegeben werden.		

Tabelle 12: Kommandos an den Kartenleser

5.2 Kommandos zum Lesen einer KVK

Kommando	Sequenz (APDU)	Bemerkung
Select File (KVK)	'00 A4 04 00 06 D2 76 00 00 01 01'	KVK-Applikation selektieren
Read Binary	'00 B0 00 00 00'	KVK-Template lesen

Tabelle 13: Kommandos zum Lesen einer KVK

5.3 Konstanten einer eGK

Für die im vorangegangenen Text genannten Dateien und Verzeichnisse der eGK sind die Identifikatoren, die in den Kommandos zu verwenden sind, als Konstanten festgelegt. Dazu gehören Application Identifier (AID) und File Identifier (FID).

Applikation	ID (hex)	Bemerkung
Root (MF)	'D2760001448000'	Masterfile
HCA	'D27600000102'	Health Care Application

Tabelle 14: Application Identifier AID

Short File Identifier (SFID) gelten nur relativ zu ihrem Parent-Directory (DF). Die Verwendung eines SFID setzt voraus, dass zuvor die entsprechende Applikation oder das DF selektiert wurde. SFID für EF.GDO gilt z. B. nur innerhalb der Root, und SFID für EF.PD gilt nur nach der Selektion der HCA. Bei Selektion eines falschen Parent-Verzeichnisses wird die SFID von der eGK fehlinterpretiert und kann zu überraschenden Ergebnissen führen.

Datei	Short File ID (hex)	Gültig in	Bemerkung
EF.ATR	'1D'	Root	Kenndaten der Karte
EF.GDO	'02'	Root	ICCSN
EF.Version	'10'	Root	Versionierung von Kartenfeatures und Dateien
EF.StatusVD	'0C'	HCA	Status der Stammdaten VSD
EF.PD	'01'	HCA	Persönliche Daten des Versicherten
EF.VD	'02'	HCA	Allgemeine Versicherungsdaten

Tabelle 15: ausgewählte Dateien einer eGK (Generation 1, 1plus und 2)

5.4 Kommandos an die eGK

In diesem Kapitel sind die Kommandos zusammengetragen, die für das Lesen der elektronischen Gesundheitskarte (eGK) benötigt werden. Alle diese Kommandos sind im Teil 1 der Spezifikation der eGK ausführlich beschrieben. Hier werden sie nur soweit dargestellt, wie sie in den folgenden Beispielen benutzt werden.

Den Bezeichnungen in [gemSpec_eGK_P1] folgend, werden die Antwort-Codes der Karte in den folgenden Tabellen als Trailer bezeichnet, da es sich immer um die letzten beiden Bytes der Antwort handelt.

5.4.1 Select

Mit dem Select-Kommando (siehe [gemSpec_eGK_P1#15.2.6]) kann eine Applikation, ein Ordner oder eine Datei durch ihren/seinen Bezeichner (AID / FID) selektiert werden.

In den folgenden Tabellen werden nur die Ausprägungen des Select-Kommandos genannt, mit denen die Root bzw. die Gesundheitskarten-Applikation selektiert werden kön-

nen. Die Selektion von Dateien ist in den hier beschriebenen Anwendungsfällen unnötig, da sie implizit mit dem Read-Kommando erfolgen kann.

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [7816–4]
INS	'A4'	Instruction Byte gemäß [7816–4]
P1	'04'	selectionMode = Ordnerselektion mit applicationIdentifier
P2	'0C'	fileOccurrence + responseType = first occurrence, keine Antwortdaten
Data	'XX…YY'	AID, Oktettstring, Anzahl Oktette aus dem Intervall [1, 16]

Tabelle 16: SELECT, AID, first occurrence, keine Antwortdaten

5.4.1.1 Antwort der Karte auf Selektieren eines Files

Trailer	Inhalt	Beschreibung
'90 00'	NoError	erfolgreiche Selektion eines Files
'62 83'	FileDeactivated	selektiertes File ist logisch oder physikalisch deaktiviert

Tabelle 17: SELECT Antwort APDU im Erfolgsfall

Trailer	Inhalte	Beschreibung
'6A 82'	FileNotFound	zu selektierendes File wurde nicht gefunden
'69 00'	Command not allowed	Mobiles Kartenterminal: Autorisierung fehlt

Tabelle 18: SELECT Antwort APDU im Fehlerfall

Beispiel: Select EGK-Root

'00 A4 04 0C 07 D2 76 00 01 44 80 00'

Beispiel: Select HCA

'00 A4 04 0C 06 D2 76 00 00 01 02'

Die Dateien EF.PD und EF.VD müssen nach der Selektion der HCA nicht explizit selektiert werden, da es eine Variante des Lesekommandos gibt, mit der eine Datei implizit anhand der SFID ausgewählt wird.

5.4.2 Read Binary

Read Binary (siehe [gemSpec_eGK_P1#15.3.2]) liest die gewünschte Anzahl von Bytes aus einer zuvor selektierten Datei. Die Anzahl der Datenbytes, die von diesem Komman-

do zurückgeliefert werden soll, wird in dem Parameter Ne (siehe [gemSpec_eGK_P1#12.5.6]) angegeben und muss aus dem Intervall [1,65535] oder WildcardShort ['00'] bzw. WildcardExtended ['0000'] stammen. Bei der Verwendung von WildCard-Werten werden maximal 256 bzw. 65536 Bytes zurückgeliefert, sofern das Ende der Daten noch nicht erreicht ist. Der Offset in diesem Befehl darf 32767 nicht überschreiten.

Read Binary mit Short File Identifier (SFID) kombiniert die Selektion einer Datei mit dem Lesen aus dieser Datei. Eventuell notwendige weitere Lesevorgänge werden mit Read Binary ohne die gleichzeitige Selektion einer Datei durchgeführt. In diesem Fall enthalten die Parameter P1 und P2 den Offset, ab dem gelesen werden soll.

Alle Längenangaben (Le bzw. Lc) können in ‘short’ (8Bit) oder ‘extended’ (16Bit) kodiert werden. Hierbei ist zu beachten, dass bei extended Längenangaben nach dem Parameter 2 zwingend ein Null-Byte folgen muss und in einem Kommando keine Vermischung von ‘short’ und ‘extended’ verwendet werden darf (siehe [gemSpec_eGK_P1#12.7]).

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [7816–4]
INS	'B0'	Instruction Byte gemäß [7816–4]
P1	'XX'	Offset (most significant Byte) max. '7F'
P2	'XX'	Offset (least significant Byte)
Ne	length	Anzahl der erwarteten Oktette in den Antwortdaten

Tabelle 19: READ BINARY ohne shortFileIdentifier

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [7816–4]
INS	'B0'	Instruction Byte gemäß [7816–4]
P1	'XX'	128 + shortFileIdentifier, das heißt ´80´ + shortFileIdentifier
P2	'XX'	offset
Ne	length	Anzahl der erwarteten Oktette in den Antwortdaten

Tabelle 20: READ BINARY mit shortFileIdentifier

5.4.2.1 Antwort der Karte auf Lesen in transparenten EF

Daten	Inhalt	Beschreibung
'XX…YY'		ausgelesene Daten
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	erfolgreiche Leseoperation
'62 82'	EndOfFileWarning	weniger Daten vorhanden, als mittels Ne angefordert
'62 81'	CorruptDataWarning	möglicherweise sind die Antwortdaten korrupt

Tabelle 21: READ BINARY Antwort APDU im Erfolgsfall

Beispiele Trailer	Inhalt	Beschreibung
'67 00'	WrongLength	Die Anzahl der angeforderten Daten übersteigt die maximale Puffergröße.
'6A 82'	FileNotFound	per shortFileIdentifier adressiertes EF nicht gefunden
'69 86'	NoCurrentEF	es ist kein EF ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	ausgewähltes EF ist nicht transparent
'6B 00'	OffsetTooBig	Parameter offset in Kommando APDU ist zu groß
'69 00'	Command not allowed	Mobiles Kartenterminal: Autorisierung fehlt

Tabelle 22: READ BINARY Antwort APDU im Fehlerfall

Beispiel:

Read EF.PD bei ausreichendem Puffer (nach Selektion von MF und HCA). Zunächst ist die Länge der Daten zu ermitteln (SFID von EF.PD ist 1, ergibt P1='81'; die Länge steht in den ersten zwei Bytes im File):

'00 B0 81 00 02

Dann die ermittelte Länge (in diesem Beispiel 0x016F) im nächsten Lesekommando angeben. Da sie länger als 1 Byte ist, muss zur Kennzeichnung ein Nullbyte vorangestellt werden. In P1, P2 steht jetzt der Offset, ab dem gelesen werden soll. In gleicher Art werden gegebenenfalls weitere Fortsetzungskommandos aufgebaut.

'00 B0 00 02 00 01 6F'

Beim mobilen Kartenterminal braucht keine Puffergröße berücksichtigt zu werden, die einzige unterstützte Befehlsform ist das Lesen des kompletten Inhaltes mit einem Kommando.

'00 B0 81 00 00 00 00'

Hinweis:

Bei der Verwendung von WildCard im Zusammenhang mit Extended Length bei READ BINARY muss auch die mögliche Antwortdatenlänge berücksichtigt werden, denn überschreitet diese die verfügbare Pufferlänge, kann die Karte mit einem Fehler ('6700') antworten.

5.4.3 Read Record

Das Kommando Read Record (siehe [gemSpec_eGK_P1#15.4.5]) liest einen Record [1,n] aus einer zuvor selektierten Datei. Die Anzahl der Datenbytes, die von diesem Kommando zurück geliefert werden soll, wird in dem Parameter Ne (siehe [gemSpec_eGK_P1#12.5.6]) angegeben und muss aus dem Intervall [1,255] oder WildcardShort ['00'] stammen. Bei der Verwendung von WildCardShort werden maximal 256 Bytes zurückgeliefert, sofern das Ende der Daten noch nicht erreicht ist.

Read Record mit shortFileIdentifier kombiniert die Selektion einer Datei mit dem Lesen eines Records aus dieser Datei. Eventuell notwendige weitere Lesevorgänge können mit Read Record ohne die gleichzeitige Selektion einer Datei durchgeführt werden.

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [7816–4]
INS	'B2'	Instruction Byte [7816–4]
P1	'XX'	recordNumber
P2	'04'	Kodierung für „nutze Listenelement P1“
Ne	length	Anzahl der erwarteten Oktette in den Antwortdaten

Tabelle 23: READ RECORD ohne shortFileIdentifier

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [7816–4]
INS	'B2'	Instruction Byte [7816–4]
P1	'XX'	recordNumber
P2	'XX'	ShortFileIdentifier * 8 + '04'
Ne	length	Anzahl der erwarteten Oktette in den Antwortdaten

Tabelle 24: READ RECORD mit shortFileIdentifier

5.4.3.1 Antwort der Karte auf Lesen in strukturierten EF

Daten	Inhalt	Beschreibung
	'XX…YY'	Ausgelesene Daten
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	erfolgreiche Leseoperation
'62 82'	EndOfRecordWarning	mittels Ne mehr Daten angefordert, als vorhanden sind
'62 81'	CorruptDataWarning	möglicherweise sind Antwortdaten korrupt

Tabelle 25: READ RECORD Antwort APDU im Erfolgsfall

Trailer	Inhalt	Beschreibung
'67 00'	WrongLength	Falsche Rekordlänge
'6A 82'	FileNotFound	per shortFileIdentifier adressiertes EF nicht gefunden
'69 86'	NoCurrentEF	es ist kein EF ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	ausgewähltes EF ist nicht strukturiert
'6A 83'	RecordNotFound	Listenelement recordNumber existiert nicht

Tabelle 26: READ RECORD Antwort APDU im Fehlerfall

Anhang A

A1 – Abkürzungen

Kürzel	Erläuterung
AID	Application Identifier
ATR	Answer to Reset
BCD	Binary Coded Decimal
CLA	Class Byte
DER-TLV	TLV nach Distinguished Encoding Rules
FCP	File Control Parameter
FID	File Identifier
GVD	Geschützte Versichertendaten
HCA	Health Care Application
ICCSN	Integrated Circuit Card Serial Number
INS	Instruction Byte
MF	Maste File
MKT	Multifunktionales Kartenterminal
PD	Persönliche Versichertendaten
SFID	Short File Identifier
SICCT	Secure Interoperable ChipCard Terminal
SRQ	Specification Related Question (d.i. Change Requests)
TLV	Type-Length-Value-Format
VD	Allgemeine Versicherungsdaten
VSD	Versichertenstammdaten
XML	Extensible Markup Language

A2 – Glossar

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

A3 – Abbildungsverzeichnis

A4 – Tabellenverzeichnis

A5 - Referenzierte Dokumente

A5.1 – Dokumente der gematik

Die nachfolgende Tabelle enthält die Bezeichnung der im vorliegenden Dokument referenzierten Dokumente der gematik zur Telematikinfrastruktur. Es werden die Versionen genannt, die zum Entstehungszeitpunkt dieses Dokuments (Mai 2011) verbindlich sind. Spätere Versionen können Abweichungen aufweisen, die sich auf die Implementierung der hier beschriebenen Abläufe auswirken. Die jeweils gültigen Dokumentenstände sind in den Dokumentenlandkarten (Rel. 0.5.2 für die eGK der Generation 1, Rel. 0.5.3 für eGK der Generation 1plus) aufgelistet; sie werden auf der Internetseite der gematik veröf-

fentlicht und aktualisiert. Als wichtige Ergänzung müssen die SRQs berücksichtigt werden, die die Änderungen seit der Veröffentlichung einer Dokumentenversion festlegen. Sie sind ebenfalls in den Dokumentenlandkarten genannt.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Einführung der Gesundheitskarte – Glossar gematik_glossar
[gemSpec_eGK_P1]	gematik (20.03.2008): Einführung der Gesundheitskarte – Die Spezifikation elektronische Gesundheitskarte; Teil 1 – Spezifikation der elektrischen Schnittstelle Version 2.2.0 sowie SRQ release_0_5_2_spezifikationen_der_egk_teil1 release_0_5_3_spezifikationen_der_egk_teil1
[gemSpec_eGK_P2]	gematik (25.03.2008): Einführung der Gesundheitskarte – Die Spezifikation elektronische Gesundheitskarte ; Teil 2 – Grundlegende Applikationen Version 2.2.0 sowie SRQ release_0_5_2_spezifikationen_der_egk_teil2 release_0_5_3_spezifikationen_der_egk_teil2
[gemeGK_Fach]	gematik (18.03.2008): Einführung der Gesundheitskarte - Speicherstrukturen der eGK für Gesundheitsanwendungen Version 1.6.0 release_0_5_2_speicherstrukturen release_0_5_3_speicherstrukturen
[gemFK_VSDM]	gematik (26.02.2008): Einführung der Gesundheitskarte – Fachkonzept Versichertenstammdatenmanagement (VSDM), Version 2.7.0 sowie SRQ release_0_5_2_fachkonzeptvsdm release_0_5_3_fachkonzeptvsdm
[gemFA_VSDM]	gematik (14.03.2008): Einführung der Gesundheitskarte - Facharchitektur Versichertenstammdatenmanagement Version 2.5.0 sowie SRQ (für G1 und G1 plus) release_0_5_2_facharchitektur_vsdm release_0_5_3_facharchitektur_vsdm
[gemDokLK]	gematik: Einführung der Gesundheitskarte – Dokumentenlandkarte Festlegung der Versionsstände Dokumentenlandkarte_Rel0_5_2 Dokumentenlandkarte_Rel0_5_3 Dokumentenlandkarte G2
[gem_Schemadateien]	gematik: Einführung der Gesundheitskarte - Schemadefinitionen Schemadateien Rel. 0.5.2 Schemadateien Rel. 0.5.3

[Quelle]	Herausgeber: Titel
	Schemadateien_ORS1
[gemSpec_mobKT]	gematik (02.04.2008): Einführung der Gesundheitskarte - Spezifikation Mobiles Kartenterminal Ausbaustufe 1 Version 1.1.1 sowie SRQ release_0_5_2_mobiles_kartenterminal_stufe1 release_0_5_3_mobiles_kartenterminal_stufe1
[gemSpec_COS]	gematik (20.09.2012): Spezifikation des Card Operating System (COS) Version: 3.1.0 gematik_CARD_Spezifikation_COS
[gemSpec_eGK_ObjSys]	gematik (20.09.2012): Spezifikation der elektronischen Gesundheitskarte eGK-Objektsystem Version: 3.1.1 gemSpec_eGK_ObjSys
gemSpec_eGK_Fach_TIP	Speicherstrukturen der eGK für die TI-Plattform Version: 1.1.0 ORS1_Dokumente Basis-TI
gemSpec_eGK_Fach_VSDM	Speicherstrukturen der eGK für die Fachanwendung VSDM Version: 1.1.0 ORS1_Dokumente VSDM

A5.2 – Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[eGK-KVK]	Mappingtabelle eGK -> KVK ftp://ftp.kbv.de/ita-update/Abrechnung/KBV_ITA_VGEX_Mapping_eGK.pdf
[MKT]	MKT Spezifikation inkl. CT-API http://www.teletrust.de/publikationen/spezifikationen/mkt
[SICCT]	TeleTrusT, SICCT Secure Interoperable ChipCard Terminal, SICCT-Spezifikation V-1.21 vom 17.12.2010 http://www.teletrust.de/projekte/sicct
[7816–4]	ISO/IEC 7816-4: 2004 (2nd edition) Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange
[tk_r0_profil_vsdm_2009-12-17]	Deutsche Krankenhausgesellschaft–DKG Verband der Hersteller von IT-Lösungen für das Gesundheitswesen–VHitG Telematikkonformität

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
	Release 0 Profil Versichertenstammdaten vom 17. Dezember 2009 Version: 1.2.3 http://www.dkgev.de/.../Rs-470_2009-12-17_A.pdf
XML	Extensible Markup Language (XML) http://www.w3.org/XML/

Anhang B

B1 - CT-Api-Protokoll einer Leseroutine

```
*** Start ***16:32:26
PerformanceCounterTest (50 msec) 0.060749010 sec
DLL C:\windows\system32\CTAPI\CT-api.dll geladen.
CT-Funktionen gefunden.
CT_init 0.064 sec
CT_init ctn: 1 pn: 1 Ergebnis: 0
CT_data(out) ctn: 1 dad: 1 sad: 2 Commandlänge: 5
20 11 00 00 00                                          Leser zurücksetzen
CT_data 0.007 sec
CT_data(in) dad: 2 sad: 1 Ergebnis: 0 Responselänge: 2
90 00
CT_data(out) ctn: 1 dad: 1 sad: 2 Commandlänge: 6
20 12 01 00 01 05                                       Karte anfordern
CT_data 1.146 sec
CT_data(in) dad: 2 sad: 1 Ergebnis: 0 Responselänge: 2
90 01
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 12
00 A4 04 0C 07 D2 76 00 01 44 80 00                     Selektiere eGK Root
CT_data 0.111 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 2
90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 9D 00 00                                          Lese EF.ATR
CT_data 0.059 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 36
E0 10 02 02 02 48 02 02 02 48 02 02 02 48 02 02
02 48 66 0E 46 0C 05 44 45 47 2B 44 1A B4 00 01
03 00 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 82 00 00                                          Lese EF.GDO
CT_data 0.057 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 14
5A 0A 80 27 68 81 03 00 00 01 00 33 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B2 01 84 00                                          Lese EF.Version Rec.1
CT_data 0.060 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 7
00 20 02 00 01 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B2 02 04 00                                          Lese EF.Version Rec.2
CT_data 0.059 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 7
00 20 02 00 01 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B2 03 04 00                                          Lese EF.Version Rec.3
CT_data 0.059 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 7
00 10 07 00 00 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 11
00 A4 04 0C 06 D2 76 00 00 01 02                        Selektiere DF.HCA
CT_data 0.068 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 2
90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 8C 00 19                                          Lese EF.StatusVD
CT_data 0.062 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 27
30 32 30 30 39 30 37 31 30 31 34 34 34 34 31 00
50 01 00 00 00 00 00 00 00 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 81 00 02                                          Lese Länge PD aus EF.PD
CT_data 0.060 sec
```

```
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 4
01 6F 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 7
00 B0 00 02 00 01 6F                                    Lese PD aus EF.PD
CT_data 0.098 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 369
1F 8B 08 00 00 00 00 00 00 8D 91 4D 4F C3 30
0C 86 FF CA 94 FB EA 66 6C B0 21 37 08 31 3E 26
AD 0C 31 31 B8 4D A1 35 6B 45 9B A2 24 1D 1F BF
9D 03 EE D8 C6 86 38 70 49 F2 DA CE F3 5A 36 9E
BC 95 45 6B 49 D6 E5 95 89 84 0C 42 D1 22 93 54
69 6E 16 91 18 4D 27 ED 7E BF 37 68 CB 9E 68 39
AF 4D AA 8B CA 50 24 DE C9 89 13 85 77 67 F3 1B
FE 5A 91 29 F2 24 A3 59 83 E1 DB 7A E2 4A 3E 1E
E2 71 EB 6C 18 CF 67 E7 B7 D3 D1 E4 3A 12 BD 60
E5 C0 9E C6 45 22 F3 FE E5 18 E0 D5 05 0B 2A B5
CF 9F 83 94 E0 49 C3 D2 A5 0E FE C1 86 25 F3 84
C2 9D 9C DD 53 66 3E 1A AA A1 94 A1 0C BB 87 72
80 F0 3B 87 2B 0B A3 F0 92 1E 6B EB 1D 93 EB 52
C9 41 3F 0C A5 EC 20 EC 85 71 56 59 A3 4B 52 63
5F 33 6A 2D F0 5A 27 D9 EA 15 D7 8E 1B 78 D6 0C
6F 37 7E 08 DB 14 F3 5D 92 15 94 64 5E C5 0D 76
AB 70 EA AD 76 8E CC 69 6A 89 6F 6E A9 72 BE A0
DC 7F E8 AC 50 9D EE E0 E0 08 61 2F 86 13 EB D5
45 41 C6 71 73 0B 84 46 E2 98 97 A3 F0 BE CA 8C
E3 B2 42 F3 94 C8 F2 1E 49 0D 11 FE 0A 23 7C 7F
59 FB AB F8 33 6B 90 DE 06 08 9B 18 5E E9 DA 99
BA 2C 79 AE DD 43 84 1D B9 2D FA 69 1C 36 C3 84
FD 85 FC 67 95 EA 0B 6B 3E 2A 8C 89 02 00 00 90
00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 82 00 08                                          Lese Zeiger aus EF.VD
CT_data 0.069 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 10
00 08 01 A3 01 A4 02 7D 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 7
00 B0 00 08 00 02 46                              Lese VD+GVD Teil 1 aus EF.VD
CT_data 0.128 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 584
1F 8B 08 00 00 00 00 00 00 AD 53 EF 4F C2 30
10 FD 57 96 7E 97 6E 26 A8 98 AE 04 D1 18 82 88
81 88 C6 2F 4B 61 E7 58 18 37 D3 EB F0 C7 5F EF
6D 1A B3 21 41 4D DC 97 EE EE BD 7B 7D BD 6B 55
F7 65 9D 79 1B B0 94 E6 18 8A A0 E5 0B 0F 70 91
C7 29 26 A1 18 4C C7 07 27 27 ED CE 41 D0 16 1E
39 83 B1 C9 72 84 50 BC 02 89 AE 56 B7 FD A8 97
65 09 AC 21 45 98 95 22 8B 25 D8 02 13 8A 8D 03
BC 1F 5D 79 FD F3 51 34 BB 98 4C 07 E3 EB 50 B4
5B D5 06 BC 25 52 28 96 CE 3D 9D 4A F9 4C 2D 56
30 2E 5D B5 62 90 8F 46 6E 28 26 F9 B3 B4 DC B0
9C D0 EA 0B 74 60 6B 51 49 A5 C5 B2 70 6F 5A 9D
41 92 22 EA A0 D3 39 F6 3B 7E A0 E4 67 42 0D 73
62 31 67 0D 24 65 71 23 5C 01 22 8B E8 C0 0F DA
47 3E 7F 4A EE C4 9B 55 99 01 8C C1 72 07 41 9F
5F 6C 95 D4 41 75 6D D6 4C 81 C2 B1 4B F0 7A 98
00 33 B3 8C D9 DE D0 1A 64 F9 95 21 02 25 2B A2
EA CD 2D 2C 96 58 09 FC 8B ED 3F EE 2F F7 18 90
DB F1 CE 29 3C 14 64 DC 5B 8A 8F 39 35 82 CB E1
4C AB 09 6B 3B 5A 59 48 49 F3 80 EA 61 7D C2 68
AC 2B F1 ED 54 83 C3 37 D5 15 14 4D A6 3D 7D D8
A4 D6 90 BA 83 E8 F3 6C EC 36 AA DC 7C 9C 87 0B
99 EF CA AC 59 CF 8B CC A0 D3 5F ED DC 05 7E AF
2B 7F F8 65 19 EE CA AE CA 1A AC EE C6 37 BA BC
9B E5 AA E4 3E 77 72 BB 79 B2 D1 5A D9 7C 11 BF
78 4A FA 1D EA F5 AA D9 07 04 00 00 1F 8B 08 00
00 00 00 00 00 00 8D 8E 4F 6B C2 40 10 C5 BF 4A
98 BB 19 45 02 5A 76 D7 83 4A 11 4D 95 86 4A F1
12 96 64 4C 42 E3 44 9C 4D D4 7C FA 6E FF DC 3C
B4 97 99 E1 CD E3 F7 9E 9A DD 4E 75 D0 D1 45 AA
86 35 8C C2 21 04 C4 59 93 57 5C 68 58 25 DB C1
64 12 4D 07 A3 08 02 71 96 73 5B 37 4C 1A EE 24
30 33 EA 6D 9E 3E 93 64 65 4B AE 77 B4 FF A2 64
25 5D 1C 79 A3 1F EF F1 26 98 2F E2 74 BF 7C 4D
56 DB 17 0D 51 F8 1D E0 23 59 34 94 CE 9D 9F 10
AF 12 16 74 B2 AE FA 08 73 C2 A3 C5 4E 72 C1 BF
```

```
D1 D8 79 1C 18 75 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 02 4E 30                                  Lese VD+GVD Rest aus EF.VD
CT_data 0.054 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 50
68 7B 5B D6 2D 17 E2 1B BA 56 8C 4A 7E F6 50 E1
EF A5 F0 D1 B4 88 77 E9 9A 98 7B F2 60 F6 1F 33
56 F8 28 AA 7F 34 31 9F FC DF 13 73 46 01 00 00
90 00
CT_data(out) ctn: 1 dad: 1 sad: 2 Commandlänge: 6
20 15 01 00 01 05                               Karte auswerfen
CT_data 0.570 sec
CT_data(in) dad: 2 sad: 1 Ergebnis: 0 Responselänge: 2
90 01
CT_close 0.000 sec
CT_close ctn: 1 Ergebnis: 0
*** Ende *** 16:32:29
```

B2 - Variante mit Lesen von VD und GVD in getrennten Aufrufen

```
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 82 00 08                                          Lese Zeiger aus EF.VD
CT_data 0.057 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 10
00 08 01 B5 01 B6 02 84 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 7
00 B0 00 08 00 01 AE                                    Lese VD aus EF.VD
CT_data 0.111 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 432
1F 8B 08 00 00 00 00 00 00 B5 53 D1 6E 9B 40
10 FC 15 74 EF F1 01 09 92 6D 99 8B 9C 38 4A 2D
37 71 65 2B 6E 95 17 74 31 6B 40 C6 4B 75 77 B8
6D BE BE 8B B9 24 24 E0 B8 2F 7D 3B 66 67 66 F7
66 B9 D1 E5 EF 5D EE EC 41 E9 AC C0 90 79 3D 97
39 80 EB 22 CE 30 09 D9 74 39 3F EB F7 83 C1 99
17 30 47 1B 89 B1 CC 0B 84 90 FD 01 CD 2E C5 68
AF 63 39 7C B8 8E C6 79 9E C0 0E 32 84 55 E5 B4
4E 41 95 98 50 D1 00 FE B8 FB EA 5C 4F EE A2 D5
CD 62 39 9D DF 87 2C E8 1D BA 50 5F D4 C3 CA 21
64 A9 31 3F 87 9C FF D2 3D B2 91 26 DB F6 62 E0
1B C9 A9 AA F9 69 7F BE 27 4F 66 C7 79 65 18 50
1F A1 4A A4 D7 69 69 9E 6D E5 0A 92 0C 51 F8 AE
3B 70 CF 3D 7F C4 9B 68 4D 99 15 9A BA 18 25 21
79 35 7C 87 6D 01 91 8C 85 E7 5E F8 9E 1F B8 81
75 E9 24 75 E8 73 09 18 83 A2 CC 41 4C BA B4 4D
42 AD BF 97 3B 10 E3 F9 CC 59 A4 14 4A 4E 7B E1
5F E4 EE A9 54 89 D5 1F 08 35 77 FC A4 60 9D E2
C1 E2 7F DC E5 1F 67 E1 A7 86 E9 68 F4 02 1E DF
DF 63 A9 A5 79 CE 70 53 E8 36 72 3B 5B 59 70 41
4D 8D DE 2A C8 B4 F0 AC 69 13 6B FD 39 28 95 11
C1 C7 F6 16 6F B3 E9 69 98 52 47 8B E5 58 F8 1D
A2 46 B9 35 64 64 23 A1 AB 45 6F 03 D7 31 90 05
29 4D 55 AA 12 A5 6C 8D 70 DF 05 D5 C5 38 E2 50
1D E8 8D 4B 8A F5 A8 47 83 53 BB 7C 9F 7F 13 E7
7D 4B AF 3E EC F1 B3 0B B4 18 DD 68 6B B9 E6 6D
E1 A7 9F BC F8 0B F1 F0 7E 48 B9 04 00 00 90 00
CT_data(out) ctn: 1 dad: 0 sad: 2 Commandlänge: 5
00 B0 01 B6 CF                                          Lese GVD aus EF.VD
CT_data 0.079 sec
CT_data(in) dad: 2 sad: 0 Ergebnis: 0 Responselänge: 209
1F 8B 08 00 00 00 00 00 00 8D 8F 4F 0B 82 40
10 C5 BF 8A CC 3D 37 0F 42 89 5B 07 8B 08 2A 21
29 A2 8B 2C EE A4 92 AE D1 AC 56 7E FA D6 F2 10
74 A8 CB C0 9B 3F BF F7 C6 9F DE CB C2 6A F0 4A
79 A5 38 38 F6 10 2C 54 49 25 73 95 72 58 46 E1
60 34 72 C7 03 C7 05 8B B4 50 52 14 95 42 0E 0F
24 98 4E FC 86 64 EA ED 82 78 81 94 64 35 EA 56
E3 BE 43 25 19 5E 35 9A 6D 53 0E EB 95 15 CC D6
F1 7E BE 8D 96 E1 86 83 6B BF 5C 8C AF 22 AF 23
70 C8 B4 BE 78 8C DD C8 4E B1 14 3A 3F DB 12 D9
49 30 33 25 F6 9B CF 1A C3 84 3E CE B1 6E 45 56
D4 2A 25 13 58 D7 D4 B7 A3 B7 18 FA EC 53 F6 EA
FB 86 FD F9 DA E4 09 0B C5 49 5F 41 01 00 00 90
00
```