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

Änderungsbedarf:

Die Technische Richtlinie zur Zertifizierung des Fachmoduls NFDM sieht vor, dass die Zugriffsprotokollierung auf der eGK in jedem Fall gewährleistet ist. Dies kann nicht sichergestellt werden, wenn die Protokollierung erst nach dem eigentlichen eGK-Zugriff erfolgt, da Fehlersituationen vorstellbar sind, in denen nach dem Zugriff auf die eGK eine Protokollierung auf dieser nicht mehr möglich ist.

Änderungen in gemSpec_FLA_NFDM

6 Funktionsmerkmale

Die Unterkapitel „Umsetzung“ beschreiben die Umsetzung der Operationsabläufe der jeweiligen Schnittstelle in der KTR-AdV Umgebung mittels der aufzurufenden Plattformbausteine, die KTR-AdV zur Verfügung stellt, oder internen Operationen von FLA-NFDM. Tabellarisch wird jeder Aktion der Aktivitätsdiagramme entweder ein bzw. mehrere Plattformbausteine des KTR-AdV zugeordnet oder – falls keine aktionsrelevante Dienstfunktionalität bereitgestellt wird – eine interne Funktion benannt und deren Aufgabe beschrieben. Werden nicht explizit im Fehlerfalle zurückzugebende Fehlermeldungen genannt, werden die Fehlermeldungen der aufgerufenen Plattformbausteine zurückgegeben. Der in KTR-AdV realisierte Card Proxy gemäß [gemSpec_KTR-AdV] hat das Wissen über die fachspezifischen Anteile der jeweiligen Smartcard und setzt die erforderliche Funktionalität um. Deshalb muss FLA-NFDM in KTR-AdV folgende Aspekte nicht spezifizieren:

- wo die Artefakte auf einer Karte liegen,
- welche Zugriffsrechte zum Ausführen von Aktionen erforderlich (bspw. Verifikation einer PIN, externe Authentisierung eines CV-Zertifikates mittels Card-2-Card) sind,
- wie erforderliche Zugriffsrechte erworben werden können und
- wie auf Daten zugegriffen oder gewisse Aktionen ausgeführt werden

6.1 Schnittstelle I_DPE_Management

6.1.1 Schnittstellendefinition

6.1.1.1 ReadDPE

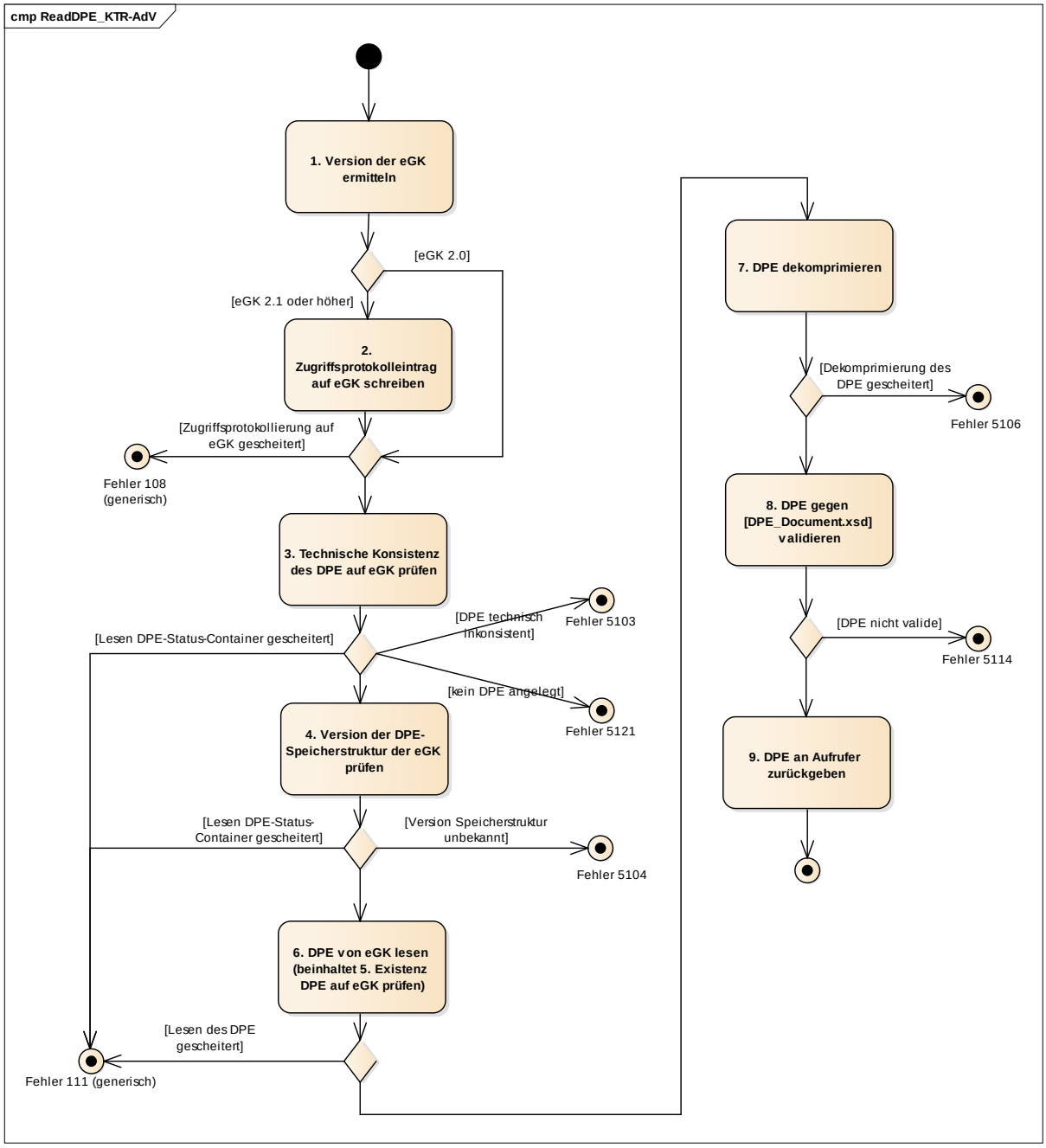

Abbildung 1: Abb_FLA_NFDM_001 – Ablauf ReadDPE

6.1.1.2 WriteDPE

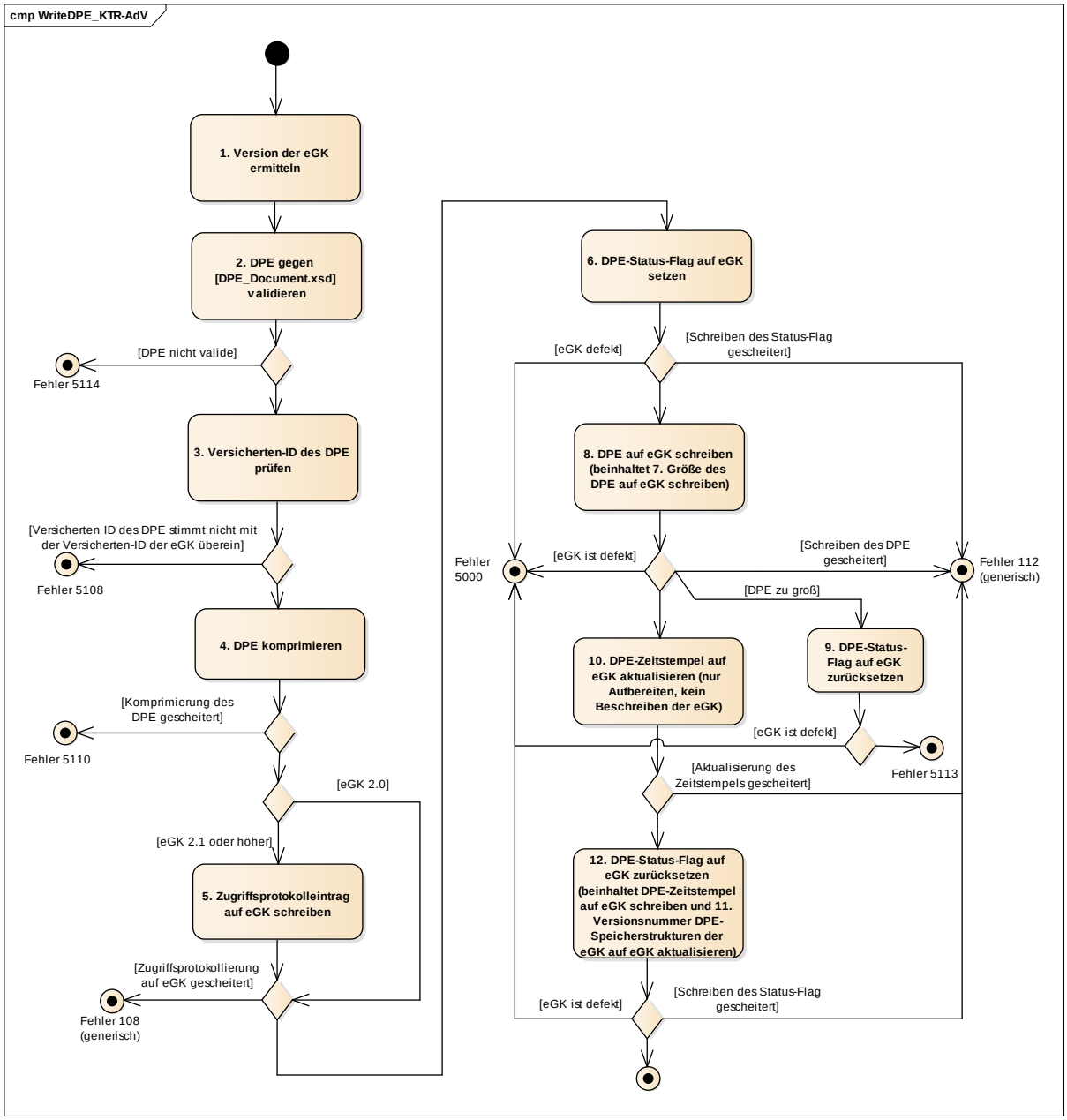

Abbildung 2: Abb_FLA_NFDM_002 – Ablauf `WriteDPE`

6.1.1.3 EraseDPE

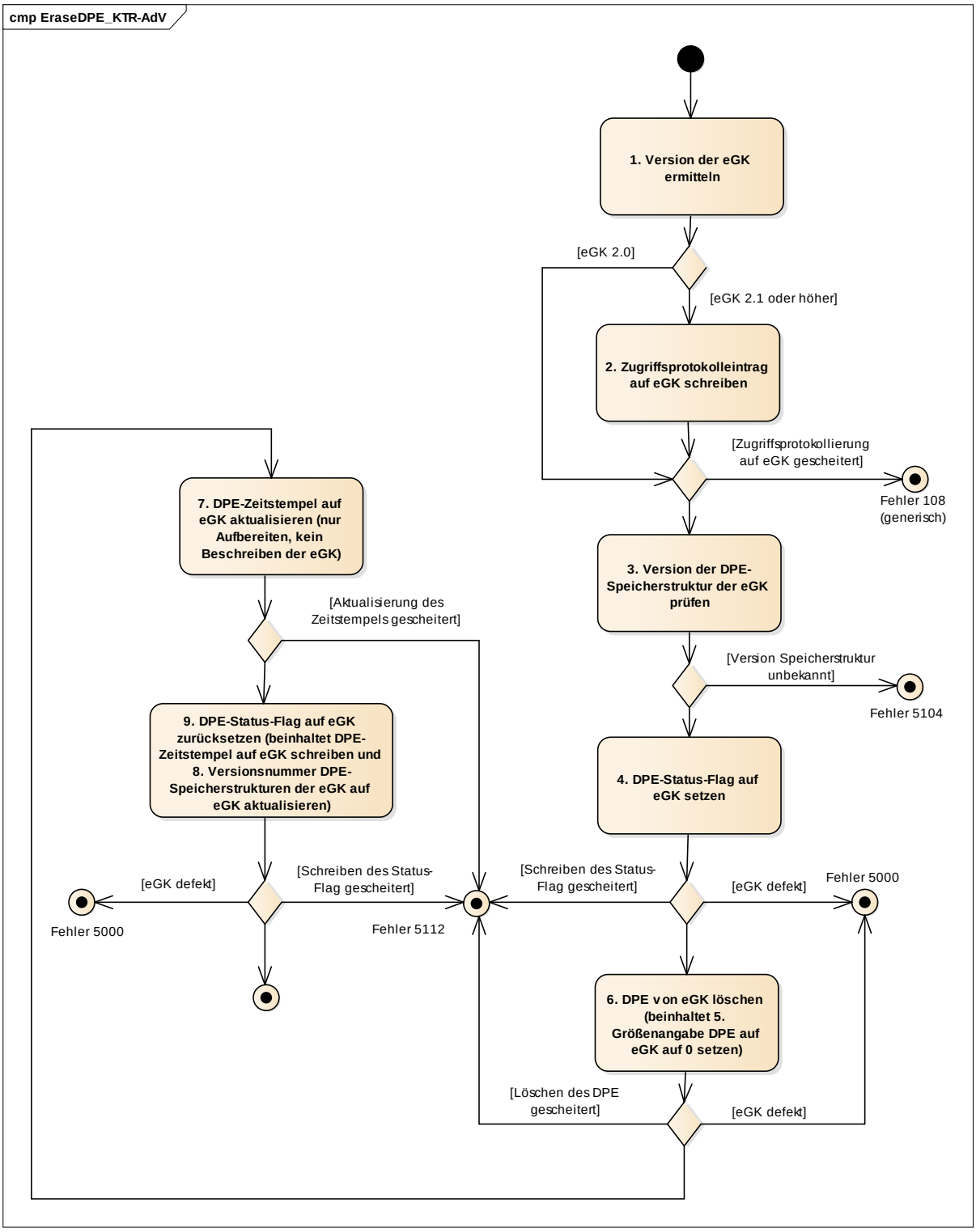

Abbildung 3: Abb_FLA_NFDM_003 – Ablauf EraseDPE

6.1.2 Umsetzung

6.1.2.1 ReadDPE

Tabelle 11: Tab_FLA_NFDM_010 – Umsetzung Ablaufaktivitäten ReadDPE

1	Version der eGK ermitteln	
	Information lt. Plattformbaustein PL_TUC_CARD_INFORMATION	
	Beschreibung	
	Für die eGK wird die Produkttypversion des COS ermittelt.	
2	Zugriffsprotokolleintrag auf eGK schreiben	
	Plattformbaustein PL_TUC_EGK_APPEND_PROTOCOL	
	Eingangsdaten	
	`cardVersion`	In 1 für eGK ermittelte Version
	`dataType`	gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“
	`accessType`	gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“
	Beschreibung	
	Ist die ermittelte Version eGK `cardVersion` G2.1 oder höher dann erfolgt das Schreiben des Zugriffsprotokolleintrags, andernfalls wird kein Protokolleintrag geschrieben (erfolgt durch KTR-AdV). Ein Zugriffsprotokolleintrag wird als Record in die Datei `EF.Logging` auf der eGK geschrieben. Tritt bei der Zugriffsprotokollierung ein Fehler auf, bricht die Operation mit Fehler 108 ab.	
23	Technische Konsistenz des DPE auf eGK prüfen	

~~23~~.1	Aufruf Plattformbaustein PL_TUC_CARD_READ_FILE	
	Eingangsdaten	
	`IDENTIFIKATOR`	`EF.StatusDPE`
	Beschreibung	
	Es werden alle Informationselemente der Datei EF.StatusDPE der eGK gelesen. Tritt dabei ein Lesefehler auf, bricht die Operation mit Fehler 111 ab.	
~~23~~.2	`NFDM:checkConsistency`	
	Eingangsdaten	
	In ~~23~~.1 aus der Datei `EF.StatusDPE` gelesene Daten (= alle Informationselemente)	
	Beschreibung	
	Es wird überprüft, ob der Wert des Informationselements `Status` der Eingangsdaten „0“ ist. Ist der Status „0“, ist der DPE technisch konsistent und die Verarbeitung läuft weiter. Ist der Status „1“, ist der Fehler 5103 mit dem Wert des Informationselements `Timestamp` der Eingangsdaten im Detailtext zurückzugeben. Ist der Wert weder „0“ noch „1“ gemäß [gemSpec_eGK_Fach_NFDM#2.2], so wurde noch kein DPE angelegt. Die Operation bricht daher mit dem Fehler 5121 ab.	
~~34~~	Version der DPE-Speicherstruktur der eGK prüfen	
	`NFDM:checkContainerVersion`	
	Eingangsdaten	
	In ~~23~~.1 aus dem DPE-Statuscontainer der eGK gelesene Daten (= alle Informationselemente)	
	Beschreibung	
	Es wird überprüft, ob FLA-NFDM die im Informationselement `Version_Speicherstruktur` der Eingangsdaten gespeicherte Versionsnummer der DPE-Speicherstruktur der eGK bekannt ist. Ist dies nicht der Fall, bricht die Operation mit Fehler 5104 ab.	
~~45~~	Existenz DPE auf eGK prüfen	
	Die Prüfung, ob ein DPE auf der eGK existiert, erfolgt implizit in ~~56~~. In ~~56~~ werden alle Daten aus der Datei `EF.DPE` der eGK gelesen. Das Informationselement `Länge DPE` enthält die Größe (Anzahl Oktette) des im Informationselement `DPE` gespeicherten DPE. Nur wenn ein Wert für dieses Informationselement gesetzt ist und dieser ungleich ´00 00´ ist, existiert ein DPE auf der eGK.	

~~5~~6	DPE von eGK lesen	
~~5~~6.1	Aufruf Plattformbaustein PL_TUC_CARD_READ_FILE	
	Eingangsdaten	
	`IDENTIFIKATOR`	`EF.DPE`
	Beschreibung	
	Es werden die Daten aus beiden Informationselementen (`Länge DPE` und `DPE`) der Datei `EF.DPE` der eGK gelesen. Tritt dabei ein Fehler auf, bricht die Operation mit Fehler 111 ab.	
~~5~~6.2	`NFDM:checkSize`	
	Eingangsdaten	
	In ~~5~~6.1 aus der Datei `EF.DPE` der eGK gelesene Daten (die beiden Informationselemente `Länge DPE` und `DPE`)	
	Beschreibung	
	Es wird das Informationselement `Länge DPE` der Eingangsdaten ausgewertet. Das Informationselement `Länge DPE` enthält die Größe (Anzahl Oktette) des im Informationselement `DPE` gespeicherten DPE. Ist die Größe ´00 00´, bedeutet dies, es existiert kein DPE auf der eGK. In diesem Fall bricht die Operation mit Fehler 5121 ab.	
~~5~~6.3	`NFDM:extractDpe`	
	Eingangsdaten	
	In 5.1 aus der Datei `EF.DPE` der eGK gelesene Daten (die beiden Informationselemente `Länge DPE` und `DPE`)	
	Beschreibung	
	Die dem Informationselement `Länge DPE` folgende und der in ~~5~~6.2 ausgewerteten Größe entsprechende Anzahl Oktette wird als DPE zur Weiterverarbeitung extrahiert. Tritt dabei ein Fehler auf, bricht die Operation mit Fehler 111 ab.	
~~6~~	~~Zugriffsprotokolleintrag auf eGK schreiben~~	
	~~Plattformbaustein PL_TUC_EGK_APPEND_PROTOCOL~~	
	~~**Eingangsdaten**~~	
	~~`cardVersion`~~	~~In 1.1 für eGK ermittelte Version~~
	~~`dataType`~~	~~gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“~~
	~~`accessType`~~	~~gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“~~
	~~**Beschreibung**~~	
	~~Ist die ermittelte Version eGK `cardVersion` G2.1 oder höher dann erfolgt das Schreiben des Zugriffsprotokolleintrags, andernfalls wird kein Protokolleintrag geschrieben (erfolgt durch KTR-AdV).~~ ~~Ein Zugriffsprotokolleintrag wird als Record in die Datei `EF.Logging` auf der eGK geschrieben. Tritt bei der Zugriffsprotokollierung ein Fehler auf, bricht die Operation mit Fehler 108 ab.~~	
7	DPE dekomprimieren	
	`NFDM:decompress`	
	Eingangsdaten	
	Der in ~~5~~6.3 extrahierte DPE	
	Beschreibung	

	Der DPE wird dekomprimiert. Tritt dabei ein Fehler auf, bricht die Operation mit Fehler 5106 ab.
8	DPE gegen [DPE_Document.xsd] validieren
	`NFDM:validateDocument`
	Beschreibung
	Der DPE wird gegen das XML-Schema [DPE_Document.xsd] validiert. Ist der DPE nicht valide gegen das Schema, so bricht die Operation mit Fehler 5114 ab.
9	DPE an Aufrufer zurückgeben
	Rückgabe `DPEDocument`

6.1.2.2 WriteDPE

Tabelle 12: Tab_FLA_NFDM_011 – Umsetzung Ablaufaktivitäten `WriteDPE`

1	Version der eGK ermitteln
	Information lt. Plattformbaustein PL_TUC_CARD_INFORMATION
	Beschreibung
	Für die eGK wird die Betriebssystemversion ermittelt und das Zertifikat `C.CH.AUTN` gelesen.
2	`DPEDocument` gegen [DPE_Document.xsd] validieren
	`NFDM: validateDocument`
	Beschreibung
	Der DPE wird gegen das XML-Schema [DPE_Document.xsd] validiert. Ist der DPE nicht valide gegen das Schema, so bricht die Operation mit Fehler 5114 ab.
3	Versicherten-ID des DPE prüfen
3.1	`NFDM:extractDpeInsurantId`
	Eingangsdaten
	`DPEDocument`
	Beschreibung
	Die Versicherten-ID (Wert des Elements `/DPEDocument/Persönliche_Erklärungen/DPE_Versicherter/Versicherter/Versicherten_ID`) wird aus `DPEDocument` extrahiert.
3.2	`NFDM:extractSubjectDN`
	Beschreibung
	Aus dem in 1.~~1~~ ermittelten Zertifikat `C.CH.AUTN` wird die SubjectDN extrahiert.
3.3	`NFDM:extractCertInsurantId`
	Beschreibung

	Aus der in 3.2 ermittelten SubjectDN wird die Versicherten-ID im Feld mit dem Namen `organizationalUnitName` extrahiert. Die Struktur des SubjectDN ist in [gemSpec_PKI#5.1.2] und [gemSpec_PKI#5.1.3.1] definiert. *Anmerkung: Es gibt zwei `organizationalUnitName`-Felder. Das zehnstellige, alphanumerische Feld beinhaltet die Versicherten-ID (unveränderbarer Teil der Krankenversichertennummer), das andere, neunstellige numerische Feld, das Institutionskennzeichen der Krankenversichertennummer (s. [gemSpec_PKI#5.1.3.1]).*
3.4	`NFDM:checkInsurantIdsForEquality`
	Beschreibung
	Die in 3.3 aus dem DPE extrahierte Versicherten ID und die in 3.2 aus SubjectDN von `C.CH.AUTN` extrahierte Versicherten-ID werden auf Gleichheit getestet. Sind die beiden IDs nicht gleich, bricht die Operation mit Fehler 5108 ab.
4	DPE komprimieren
	`NFDM:compress`
	Beschreibung
	Das in 2 validierte `DPEDocument` wird komprimiert. Tritt in diesem Schritt ein Fehler auf, bricht die Operation mit Fehler 5110 ab.
5	Zugriffsprotokolleintrag auf eGK schreiben
	Plattformbaustein PL_TUC_EGK_APPEND_PROTOCOL
	Eingangsdaten
	`cardVersion` — In 1 für eGK ermittelte Version
	`dataType` — gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“
	`accessType` — gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“
	Beschreibung
	Ist die ermittelte Version eGK `cardVersion` G2.1 oder höher dann erfolgt das Schreiben des Zugriffsprotokolleintrags, andernfalls wird kein Protokolleintrag geschrieben (erfolgt durch KTR-AdV). Ein Zugriffsprotokolleintrag wird als Record in die Datei `EF.Logging` auf der eGK geschrieben. Tritt bei der Zugriffsprotokollierung ein Fehler auf, bricht die Operation mit Fehler 108 ab.
~~5~~6	DPE-Status-Flag auf eGK setzen
	Plattformbaustein PL_TUC_CARD_UPDATE_FILE
	Eingangsdaten
	`IDENTIFIKATOR` — `EF.StatusDPE`
	`OFFSET` — 0
	`NEWDATA` — 1
	Beschreibung
	Zur „Transaktionssicherung“ wird der Wert des Informationselements `Status` der Datei `EF.StatusDPE` der eGK vor Beginn des Schreibvorgangs des DPE auf „1“ gesetzt. Liefert PL_TUC_CARD_UPDATE_FILE als Ergebnis `MemoryFailure`, bricht die Operation mit Fehler 5000 ab. In allen anderen Fehlerfällen bricht die Operation mit Fehler 112 ab.
~~6~~7	Größe des DPE auf eGK schreiben
~~6~~7.1	Größe des DPE ermitteln

	`NFDM:determineSize`
	Eingangsdaten
	Der in 4 komprimierte DPE
	Beschreibung
	Die Größe (Anzahl Oktette) des DPE wird bestimmt.
~~6~~7.2	Das Schreiben der in ~~6~~7.1 ermittelten Größe des DPE auf die eGK erfolgt zusammen mit dem Schreiben des DPE in ~~7~~8.
~~7~~8	DPE auf eGK schreiben
~~7~~8.1	`NFDM:concatenate`
	Eingangsdaten
	DPE und die in ~~6~~7.1 ermittelte Größe (Anzahl Oktette) des DPE
	Beschreibung
	Die in ~~6~~7.1 ermittelte Größenangabe wird konkateniert mit dem DPE. Die Struktur entspricht [gemSpec_eGK_Fach_NFDM#3.1]. Tritt hierbei ein Fehler auf, bricht die Operation mit Fehler 112 ab.
~~7~~8.2	Plattformbaustein PL_TUC_CARD_UPDATE_FILE
	Eingangsdaten
	`IDENTIFIKATOR` — `EF.DPE`
	Beschreibung
	Die in ~~7~~8.1 konkatenierte Oktettkette (Größenangabe und DPE) wird auf die eGK in die Datei `EF.DPE` geschrieben. Liefert PL_TUC_CARD_UPDATE_FILE als Ergebnis `MemoryFailure`, bricht die Operation mit Fehler 5000 ab. Liefert PL_TUC_CARD_UPDATE_FILE als Ergebnis `DataTooBig` (Größe des zu schreibenden DPE ist größer als der in der Datei EF.DPE entsprechend des Objektsystems der eGK zur Verfügung stehende Speicherplatz) wird Schritt ~~8~~9 ausgeführt und die Operation mit dem Fehler 5113 abgebrochen. In allen anderen Fehlerfällen bricht die Operation mit Fehler 112 ab.
~~8~~9	DPE-Status-Flag auf eGK zurücksetzen
Falls die Größenprüfung beim Aufruf von PL_TUC_CARD_UPDATE_FILE in ~~7~~8.2 mit Fehler `DataTooBig` endete.	

~~8~~9.1	Plattformbaustein PL_TUC_CARD_UPDATE_FILE	
	Eingangsdaten	
	`IDENTIFIKATOR`	`EF.StatusDPE`
	`OFFSET`	`0`
	`NEWDATA`	`0`
	Beschreibung	
	Mit Beendigung des Schreibvorgangs des DPE auf die eGK wird das DPE-Status-Flag (Wert des Informationselements `Status` der Datei `EF.StatusDPE` der eGK) wieder zurück auf „0“ gesetzt. Liefert PL_TUC_CARD_UPDATE_FILE als Ergebnis `MemoryFailure`, bricht die Operation mit Fehler 5000 ab. In allen anderen Fehlerfällen bricht die Operation mit Fehler 112 ab.	
~~8~~9.2	Die Operation bricht mit dem Fehler 5113 ab.	
~~9~~10	DPE-Zeitstempel auf eGK aktualisieren	
~~9~~10.1	`NFDM:getTime`	
	Beschreibung	
	Liefere Systemzeit, die durch den Plattformbaustein PL_TUC_NET_SYNC_TIME bereitgestellt wird.	
~~9~~10.2	`NFDM:formatSystemTime`	
	Eingangsdaten	
	Im vorherigen Teilschritt ermittelte Systemzeit	
	Beschreibung	
	Die Systemzeit wird für das Schreiben in das Informationselement `Timestamp` der Datei `EF.StatusDPE` der eGK gemäß [gemSpec_eGK_Fach_NFDM#3.2] formatiert (14 Oktette; YYYYMMDDhhmmss). Das eigentliche Schreiben des DPE-Zeitstempels in das Informationselement `Timestamp` der Datei `EF.StatusDPE` der eGK erfolgt aus Performancegründen in 1~~1~~2.2. Dort wird die Oktettkette zur Befüllung der Datei `EF.StatusDPE` als Ganzes in die Datei `EF.StatusDPE` der eGK geschrieben. Tritt hierbei ein Fehler auf, bricht die Operation mit Fehler 112 ab.	
1~~0~~1	Versionsnummer DPE-Speicherstruktur der eGK auf eGK aktualisieren	
	Das Aktualisieren der Versionsnummer erfolgt implizit im nächsten Schritt. Dort wird die Oktettkette zur Befüllung der Datei `EF.StatusDPE` als Ganzes in die Datei `EF.StatusDPE` auf der eGK geschrieben.	
1~~1~~2	DPE-Status-Flag auf eGK zurücksetzen	
1~~1~~2.1	`NFDM:concatenate`	
	Eingangsdaten	
	In ~~9~~10.2 formatierter DPE-Zeitstempel, Versionsnummer der DPE-Speicherstruktur der eGK, gemäß derer der DPE auf die eGK geschrieben wird.	
	Beschreibung	
	Die Oktettkette für den DPE-Status-Container wird gemäß [gemSpec_eGK_Fach_NFDM#3.2] konkateniert, wobei das Informationselement `Version_XML` mit 5 Oktetten des Wertes ´00´ zu befüllen ist und das Informationselement `Status` den Wert „0“ erhält. Tritt hierbei ein Fehler auf, bricht die Operation mit Fehler 112 ab.	
1~~1~~2.2	Plattformbaustein PL_TUC_CARD_UPDATE_FILE	
	Eingangsdaten	

	IDENTIFIKATOR	EF.StatusDPE
	OFFSET	0
	NEWDATA	0
	Beschreibung	
	Mit Beendigung des Schreibvorgangs des DPE auf die eGK wird das DPE-Status-Flag (Wert des Informationselements `Status` der Datei `EF.StatusDPE` der eGK) wieder zurück auf „0“ gesetzt und der DPE-Zeitstempel sowie die Versionsnummer der DPE-Speicherstruktur der eGK gemäß [gemSpec_eGK_Fach_NFDM#3.2] aktualisiert. Liefert PL_TUC_CARD_UPDATE_FILE als Ergebnis `MemoryFailure`, bricht die Operation mit Fehler 5000 ab. In allen anderen Fehlerfällen bricht die Operation mit Fehler 112 ab.	
~~12~~	~~Zugriffsprotokolleintrag auf eGK schreiben~~	
	~~Plattformbaustein PL_TUC_EGK_APPEND_PROTOCOL~~	
	~~**Eingangsdaten**~~	
	~~`cardVersion`~~	~~In 1.1 für eGK ermittelte Version~~
	~~`dataType`~~	~~gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“~~
	~~`accessType`~~	~~gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“~~
	~~**Beschreibung**~~	
	~~Ist die ermittelte Version eGK `cardVersion` G2.1 oder höher dann erfolgt das Schreiben des Zugriffsprotokolleintrags, andernfalls wird kein Protokolleintrag geschrieben (erfolgt durch KTR-AdV).~~ ~~Ein Zugriffsprotokolleintrag wird als Record in die Datei `EF.Logging` auf der eGK geschrieben. Tritt bei der Zugriffsprotokollierung ein Fehler auf, bricht die Operation mit Fehler 108 ab.~~	

6.1.2.3 EraseDPE

Tabelle 13: Tab_FLA_NFDM_012 – Umsetzung Ablaufaktivitäten `EraseDPE`

1	Version der eGK ermitteln	
	Information lt. Plattformbaustein PL_TUC_CARD_INFORMATION	
	Beschreibung	
	Für die eGK wird die Betriebssystemversion ermittelt.	
2	Zugriffsprotokolleintrag auf eGK schreiben	
	Plattformbaustein PL_TUC_EGK_APPEND_PROTOCOL	
	Eingangsdaten	
	`cardVersion`	In 1 für eGK ermittelte Version
	`dataType`	gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“
	`accessType`	gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“
	Beschreibung	

	Ist die ermittelte Version eGK `cardVersion` G2.1 oder höher dann erfolgt das Schreiben des Zugriffsprotokolleintrags, andernfalls wird kein Protokolleintrag geschrieben (erfolgt durch KTR-AdV). Ein Zugriffsprotokolleintrag wird als Record in die Datei `EF.Logging` auf der eGK geschrieben. Tritt bei der Zugriffsprotokollierung ein Fehler auf, bricht die Operation mit Fehler 108 ab.	
~~23~~	Version der DPE-Speicherstruktur der eGK prüfen	
~~23~~.1	Aufruf Plattformbaustein PL_TUC_CARD_READ_FILE	
	Eingangsdaten	
	`IDENTIFIKATOR`	`EF.StatusDPE`
	Beschreibung	
	Mittels PL_TUC_CARD_READ_FILE wird das Informationselement `Version_Speicherstruktur` der Datei `EF.StatusDPE` der eGK ausgelesen.	
~~23~~.2	`NFDM:checkContainerVersion`	
	Eingangsdaten	
	In ~~23~~.1 gelesener Wert des Informationselements `Version_Speicherstruktur` der Datei `EF.StatusDPE` der eGK.	
	Beschreibung	
	Es wird überprüft, ob FLA-NFDM die in ~~23~~.1 gelesene Versionsnummer der DPE-Speicherstruktur der eGK bekannt ist. Ist dies nicht der Fall, bricht die Operation mit Fehler 5104 ab.	
~~34~~	DPE-Status-Flag auf eGK setzen	
	Aufruf Plattformbaustein PL_TUC_CARD_UPDATE_FILE	
	Eingangsdaten	
	`IDENTIFIKATOR`	`EF.StatusDPE`
	`OFFSET`	`0`
	`NEWDATA`	`1`
	Beschreibung	
	Zur „Transaktionssicherung“ wird der Wert des Informationselements `Status` der Datei `EF.StatusDPE` der eGK vor Beginn des Löschvorgangs des DPE auf „1“ gesetzt. Liefert PL_TUC_CARD_UPDATE_FILE als Ergebnis `MemoryFailure`, bricht die Operation mit Fehler 5000 ab. In allen anderen Fehlerfällen bricht die Operation mit Fehler 5112 ab.	
45	Größenangabe DPE auf eGK auf 0 setzen	
	Das Setzen der Größenangabe für den DPE auf der eGK auf 0 erfolgt implizit in ~~56~~.	

~~56~~	DPE von eGK löschen	
	Plattformbaustein PL_TUC_CARD_ERASE_FILE	
	Eingangsdaten	
	`IDENTIFIKATOR`	`EF.DPE`
	Beschreibung	
	Der Inhalt der Datei `EF.DPE` auf der eGK wird durch Oktette mit dem Wert ´00´ (NULL) überschrieben. Dadurch wird einerseits der Inhalt des Informationselements `Länge DPE` mit ´00 00´ überschrieben und zudem die Daten des im Informationselement `DPE` gespeicherten DPE „gelöscht“, d. h. mit ´00´ (NULL) überschrieben. Gibt PL_TUC_CARD_ERASE_FILE den Fehlercode MemoryFailure der eGK zurück, bricht die Operation mit Fehler 5000 ab. In allen anderen Fehlerfällen bricht die Operation mit Fehler 5112 ab.	
~~67~~	DPE-Zeitstempel auf eGK aktualisieren	
~~67~~.1	`NFDM:getTime`	
	Beschreibung	
	Liefere Systemzeit, die durch den Plattformbaustein PL_TUC_NET_SYNC_TIME bereitgestellt wird.	
~~67~~.2	`NFDM:formatSystemTime`	
	Eingangsdaten	
	Im vorherigen Teilschritt ermittelte Systemzeit	
	Beschreibung	
	Die Systemzeit wird für das Schreiben in das Informationselement `Timestamp` der Datei `EF.StatusDPE` der eGK gemäß [gemSpec_eGK_Fach_NFDM#3.2] formatiert (14 Oktette; YYYYMMDDhhmmss). Das eigentliche Schreiben des DPE-Zeitstempels in das Informationselement `Timestamp` der Datei `EF.StatusDPE` der eGK erfolgt aus Performancegründen in ~~89~~.2. Dort wird die Oktettkette zur Befüllung der Datei `EF.StatusDPE` als Ganzes in die Datei `EF.StatusDPE` der eGK geschrieben. Tritt hierbei ein Fehler auf, bricht die Operation mit Fehler 5112 ab.	
~~78~~	Versionsnummer DPE-Speicherstruktur der eGK auf eGK aktualisieren	
	Das Aktualisieren der Versionsnummer der DPE-Speicherstruktur der eGK erfolgt implizit im nächsten Schritt. Dort wird die Oktettkette zur Befüllung des DPE-Status-Containers als Ganzes in die Datei `EF.StatusDPE` der eGK geschrieben.	
~~89~~	DPE-Status-Flag auf eGK zurücksetzen	
~~89~~.1	`NFDM:concatenate`	
	Eingangsdaten	
	In ~~67~~.2 formatierter DPE-Zeitstempel, Versionsnummer der DPE-Speicherstruktur der eGK, gemäß derer der DPE auf die eGK geschrieben wird.	
	Beschreibung	
	Die Oktettkette für den DPE-Status-Container wird gemäß [gemSpec_eGK_Fach_NFDM#3.2] konkateniert, wobei das Informationselement `Version_XML` mit 5 Oktetten des Wertes ´00´ zu befüllen ist und das Informationselement `Status` den Wert „0“ erhält. Tritt hierbei ein Fehler auf, bricht die Operation mit Fehler 5112 ab.	
~~89~~.2	Plattformbaustein PL_TUC_CARD_UPDATE_FILE	
	Eingangsdaten	
	`IDENTIFIKATOR`	`EF.StatusDPE`

	OFFSET	0
	NEWDATA	0
	Beschreibung	
	Mit Beendigung des Löschvorgangs des DPE auf die eGK wird das DPE-Status-Flag (Wert des Informationselements `Status` der Datei `EF.StatusDPE` der eGK) wieder zurück auf „0“ gesetzt und der DPE-Zeitstempel sowie die Versionsnummer der DPE-Speicherstruktur der eGK gemäß [gemSpec_eGK_Fach_NFDM#3.2] aktualisiert. Liefert PL_TUC_CARD_UPDATE_FILE als Ergebnis `MemoryFailure`, bricht die Operation mit Fehler 5000 ab. In allen anderen Fehlerfällen bricht die Operation mit Fehler 5112 ab.	
~~9~~	~~Zugriffsprotokolleintrag auf eGK schreiben~~	
	~~Plattformbaustein PL_TUC_EGK_APPEND_PROTOCOL~~	
	~~**Eingangsdaten**~~	
	~~`cardVersion`~~	~~In 1.1 für eGK ermittelte Version~~
	~~`dataType`~~	~~gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“~~
	~~`accessType`~~	~~gemäß Tabelle „Tab_FLA_NFDM_002 – Werte der Zugriffsprotokolleinträge auf der eGK“~~
	~~**Beschreibung**~~	
	~~Ist die ermittelte Version eGK `cardVersion` G2.1 oder höher dann erfolgt das Schreiben des Zugriffsprotokolleintrags, andernfalls wird kein Protokolleintrag geschrieben (erfolgt durch KTR-AdV).~~ ~~Ein Zugriffsprotokolleintrag wird als Record in die Datei `EF.Logging` auf der eGK geschrieben. Tritt bei der Zugriffsprotokollierung ein Fehler auf, bricht die Operation mit Fehler 108 ab.~~	