Einführung der Gesundheitskarte

Spezifikation

Fachlogik der Fachanwendung NFDM in KTR-AdV

(FLA-NFDM)

|  |  |
| --- | --- |
| Version: | 1.0.0 |
| Revision: | \main\rel\_ors2\23 |
| Stand: | 02.08.2017 |
| Status: | freigegeben |
| Klassifizierung: | öffentlich |
| Referenzierung: | [gemSpec\_FLA\_NFDM] |

Dokumentinformationen

**Änderungen zur Vorversion**

Es handelt sich um die Erstversion des Dokumentes.

**Dokumentenhistorie**

| **Version** | **Stand** | **Kap./ Seite** | **Grund der Änderung, besondere Hinweise** | **Bearbeitung** |
| --- | --- | --- | --- | --- |
| 1.0.0 | 02.08.17 |  | freigegeben | gematik |

Inhaltsverzeichnis

[1 Einordnung des Dokumentes 5](#_Toc490553299)

[1.1 Zielsetzung 5](#_Toc490553300)

[1.2 Zielgruppe 5](#_Toc490553301)

[1.3 Geltungsbereich 5](#_Toc490553302)

[1.4 Abgrenzungen 6](#_Toc490553303)

[1.5 Methodik 6](#_Toc490553304)

[2 Systemüberblick 8](#_Toc490553305)

[3 Systemkontext 9](#_Toc490553306)

[3.1 Akteure und Rollen 9](#_Toc490553307)

[3.2 Nachbarsysteme 9](#_Toc490553308)

[4 Zerlegung des Produkttyps 10](#_Toc490553309)

[5 Übergreifende Festlegungen 11](#_Toc490553310)

[5.1 Fehlerbehandlung 11](#_Toc490553311)

[5.2 Berechtigungen 11](#_Toc490553312)

[5.2.1 Fachliche Rollen 11](#_Toc490553313)

[5.2.2 Berechtigung 12](#_Toc490553314)

[5.3 Unterstützte Generationen der eGK 12](#_Toc490553315)

[5.4 Versionierung 12](#_Toc490553316)

[5.5 Protokollierung 12](#_Toc490553317)

[5.5.1 Zugriffsprotokolleinträge auf der eGK 13](#_Toc490553318)

[5.6 Datenspeicherung 13](#_Toc490553319)

[5.7 Meldungen am Kartenterminal 14](#_Toc490553320)

[5.8 Übergreifende Vorbedingungen 14](#_Toc490553321)

[5.9 Übergreifende Nachbedingungen 15](#_Toc490553322)

[6 Funktionsmerkmale 16](#_Toc490553323)

[6.1 Schnittstelle I\_DPE\_Management 16](#_Toc490553324)

[6.1.1 Schnittstellendefinition 16](#_Toc490553325)

[6.1.1.1 ReadDPE 16](#_Toc490553326)

[6.1.1.2 WriteDPE 19](#_Toc490553327)

[6.1.1.3 EraseDPE 22](#_Toc490553328)

[6.1.2 Umsetzung 26](#_Toc490553329)

[6.1.2.1 ReadDPE 26](#_Toc490553330)

[6.1.2.2 WriteDPE 28](#_Toc490553331)

[6.1.2.3 EraseDPE 32](#_Toc490553332)

[6.2 Schnittstelle I\_COPY\_DATA 34](#_Toc490553333)

[6.2.1 Schnittstellendefinition 35](#_Toc490553334)

[6.2.1.1 GetData 35](#_Toc490553335)

[6.2.1.2 PutData 37](#_Toc490553336)

[6.2.2 Umsetzung 40](#_Toc490553337)

[6.2.2.1 GetData 40](#_Toc490553338)

[6.2.2.2 PutData 40](#_Toc490553339)

[6.3 Artefakte 42](#_Toc490553340)

[6.3.1 DPE-Speicherstruktur der eGK 42](#_Toc490553341)

[6.3.2 Der DPE auf der eGK 42](#_Toc490553342)

[6.4 Testunterstützung 42](#_Toc490553343)

[6.5 Hardwaremerkmale 42](#_Toc490553344)

[7 Informationsmodell 43](#_Toc490553345)

[8 Verteilungssicht 44](#_Toc490553346)

[9 Anhang A – Verzeichnisse 45](#_Toc490553347)

[9.1 A1 – Abkürzungen 45](#_Toc490553348)

[9.2 A2 – Glossar 45](#_Toc490553349)

[9.3 A3 – Abbildungsverzeichnis 45](#_Toc490553350)

[9.4 A4 – Tabellenverzeichnis 46](#_Toc490553351)

[9.5 A5 – Referenzierte Dokumente 46](#_Toc490553352)

[9.5.1 A5.1 – Dokumente der gematik 46](#_Toc490553353)

[9.5.2 A5.2 – Weitere Dokumente 47](#_Toc490553354)

[10 Anhang B 49](#_Toc490553355)

[10.1 B1 – Fehlermeldungen 49](#_Toc490553356)

# Einordnung des Dokumentes

## Zielsetzung

Die vorliegende Spezifikation definiert die spezifischen Anforderungen für die Fachlogik der Fachanwendung NFDM in KTR-AdV (FLA-NFDM).

FLA-NFDM ist diejenige Komponente der Fachanwendung NFDM, welche die Anwendungsfälle der Fachanwendung NFDM und die Anwendungsfälle des Leistungsmerkmals „Datenübertragung bei Kartentausch“ der Fachanwendung AdV in KTR-AdV auf Seite der AdV-App umsetzt. Der Funktionsumfang wird definiert durch das Leistungsmerkmal NFDM-L\_2 („Persönliche Erklärungen NFDM“) der Fachanwendung NFDM. Dieses wurde in [gemSysL\_NFDM] definiert und stellt KTR-AdV eine Grundfunktionalität zur Verwaltung von Datensätzen für persönliche Erklärungen (DPE) zur Verfügung.

## Zielgruppe

Das Dokument richtet sich an Hersteller der KTR-AdV, die die Fachlogik der Fachanwendung NFDM umsetzen.

## Geltungsbereich

Dieses Dokument enthält normative Festlegungen zur Telematikinfrastruktur des deutschen Gesundheitswesens. Der Gültigkeitszeitraum der vorliegenden Version und deren Anwendung in Zulassungs- oder Abnahmeverfahren wird durch die gematik GmbH in gesonderten Dokumenten (z. B. Dokumentenlandkarte, Produkttypsteckbrief, Leistungsbeschreibung) festgelegt und bekannt gegeben.

**Schutzrechts-/Patentrechtshinweis**

Die nachfolgende Spezifikation ist von der gematik allein unter technischen Gesichtspunkten erstellt worden. Im Einzelfall kann nicht ausgeschlossen werden, dass die Implementierung der Spezifikation in technische Schutzrechte Dritter eingreift. Es ist allein Sache des Anbieters oder Herstellers, durch geeignete Maßnahmen dafür Sorge zu tragen, dass von ihm aufgrund der Spezifikation angebotene Produkte und/oder Leistungen nicht gegen Schutzrechte Dritter verstoßen und sich ggf. die erforderlichen Erlaubnisse/Lizenzen von den betroffenen Schutzrechtsinhabern einzuholen. Die gematik GmbH übernimmt insofern keinerlei Gewährleistungen.

## Abgrenzungen

Spezifiziert werden in dem Dokument die von FLA-NFDM bereitgestellten logischen Schnittstellen. Es werden Plattformbausteine aus KTR-AdV gemäß [gemSpec\_KTR-AdV] verwendet, die Zugriff auf produktübergreifende Plattformleistungen definieren.

Die vollständige Anforderungslage für FLA-NFDM ergibt sich aus weiteren Konzept- und Spezifikationsdokumenten. Diese sind in dem Produkttypsteckbrief der KTR-AdV verzeichnet, da FLA-NFDM integraler Bestandteil der KTR-AdV ist.

## Methodik

Anforderungen als Ausdruck normativer Festlegungen werden durch eine eindeutige ID in eckigen Klammern sowie die dem RFC 2119 [RFC2119] entsprechenden, in Großbuchstaben geschriebenen deutschen Schlüsselworte MUSS, DARF NICHT, SOLL, SOLL NICHT, KANN gekennzeichnet.

Sie werden im Dokument wie folgt dargestellt:

**⌦ NFDM-A\_xxxx <Titel der Afo>**

Text / Beschreibung

**⌫**

Dabei umfasst die Anforderung sämtliche innerhalb der Textmarken angeführten Inhalte.

*Hinweise zur Nomenklatur:*

* Schnittstellen-, Operations-, Parameter- und Dateinamen sowie Bezeichner für Objekte auf der elektronischen Gesundheitskarte (eGK), Namen der referenzierten Technischen Use Cases (TUCs) des Konnektors und Extensible-Markup-Language(XML)-Elemente oder -Attribute werden in diesem Dokument in nicht-proportionaler Schriftart gesetzt.
* Hexadezimale Zahlen und Oktett-Strings werden in Hochkommata eingeschlossen (z. B. ´2F 03´).

*Bezeichner für Objekte der eGK:*

Der Bezeichner der in [gemSpec\_eGK\_ObjSys] definierten Objekte der eGK dient dazu, das Objekt innerhalb der Spezifikation zu identifizieren. Adressieren lassen sich die Objekte auf der eGK über deren Attribut pwdIdentifier (für Passwortobjekte), fileIdentifier (für Elementary Files (EFs)) oder applicationIdentifier (für Dedicated Files (DFs)). Ein File Identifier (FID) /Application Identifier (AID)/Password Identifier (PID) muss nur innerhalb eines Ordners eindeutig sein. Aus Gründen der Lesbarkeit und um Unterschiede zwischen eGK G2 und eKG G2.1 darzustellen, werden in dieser Spezifikation Bezeichner eingeführt, für die in der folgenden Tabelle sowohl der Bezeichner als auch der FID/AID/PID mit dem übergeordneten Ordner (überg. DF) angegeben sind.

Tabelle 1: Tab\_FLA\_NFDM\_001 Verwendete Bezeichner für Objekte der eGK

| **Objekt** | | **G 2.0** | **G 2.1** |
| --- | --- | --- | --- |
| **Bezeichner** | **FID o. AID o. PID** | **Bezeichner des überg. DF** | **Bezeichner des überg. DF** |
| MF | ´D276 0001 4480 00´ |  |  |
| DF.ESIGN | ´A000000167 455349474E´ | MF | MF |
| EF.Version2 | ´2F 10´ | MF | MF |
| EF.C.CH.AUTN.R2048 | ´C5 00´ | DF.ESIGN | DF.ESIGN |
| DF.HCA | ´D276000001 02´ | MF | MF |
| EF.Logging | ´D0 06´ | DF.HCA | DF.HCA |
| DF.NFD | ´D276 0001 4407´ | DF.HCA | DF.HCA |
| EF.NFD | ´D0 10´ | DF.NFD | DF.NFD |
| EF.StatusNFD | ´D0 0E´ | DF.NFD | DF.NFD |
| DF.DPE | ´D276 0001 4408´ | DF.HCA | DF.HCA |
| EF.DPE | ´D0 1B´ | DF.DPE | DF.DPE |
| EF.StatusDPE | ´D0 18´ | DF.DPE | DF.DPE |
| PIN.CH | ´01´ | MF | MF |
| MRPIN.NFD | ´03´ | DF.NFD | MF |
| MRPIN.NFD\_READ | ´07´ | DF.NFD | MF |
| MRPIN.DPE | ´04´ | DF.DPE | MF |
| MRPIN.DPE\_READ | ´08´ | DF.DPE | nicht vorhanden |

# 

# Systemüberblick

FLA-NFDM ist Bestandteil des Produkttyps KTR-AdV der Fachanwendung AdV. Die Systemzerlegung der Fachanwendung AdV ist in [gemSysL\_AdV#2.1] definiert.

Der Produkttyp KTR-AdV wird gebildet aus einem AdV-Server mit zugehöriger AdV-App. Die AdV-App beinhaltet Plattformbausteine, Serverproxy und Fachlogik, zu welcher FLA-NFDM gehört. Die Zerlegung des Produkttyps KTR-AdV wird in [gemSpec\_KTR-AdV#4] definiert.

FLA-NFDM nutzt die dezentrale TI-Plattform in KTR-AdV durch Aufruf der Plattformbausteine gemäß [gemSpec\_KTR-AdV#Anhang B]. FLA-NFDM ist integraler Bestandteil der KTR-AdV Umgebung (App und AdV-Server), d.h. die Spezifikationen KTR-AdV (als Plattform­kom­po­nente) und FLA-NFDM sind zwar getrennt, werden aber von einem Hersteller als eine Funktionalität aus einem AdV-Server mit zugehöriger AdV-App umgesetzt. Die inneren (rein logischen) Schnittstellen zwischen FLA-NFDM und KTR-AdV sind von außen nicht erkennbar.

FLA-NFDM stellt seine Anwendungsfälle über folgende Schnittstellen der KTR-AdV zur Verfügung:

* I\_DPE\_Management für das Leistungsmerkmal NFDM-L\_2 der Fachanwendung NFDM
* I\_COPY\_DATA für das Leistungsmerkmal „Datenübertragung bei Kartentausch“ der Fachanwendung AdV

Die Operationen der Schnittstelle I\_DPE\_Management bilden die Anwendungsfälle des Leistungsmerkmals NFDM-L\_2 gemäß folgender Tabelle ab.

Tabelle 2: Tab\_FLA\_NFDM\_019 Abbildung NFDM-Anwendungsfälle auf Operationen des FLA-NFDM

|  |  |
| --- | --- |
| **Anwendungsfall aus [gemSysL\_NFDM]** | **Operation FLA-NFDM** |
| I\_DPE\_Management | |
| NFDM\_UC\_200 DPE auf eGK schreiben | WriteDPE |
| NFDM\_UC\_201 DPE von eGK lesen | ReadDPE |
| NFDM\_UC\_202 DPE von eGK löschen | EraseDPE |

# 

# Systemkontext

## Akteure und Rollen

FLA-NFDM ist Bestandteil des Produkttyps KTR-AdV und unterliegt damit der Definition für Akteure und Rollen von KTR-AdV gemäß [gemSpec\_KTR-AdV#3.1].

Der Nutzer, als natürliche Person, welche die Anwendungsfälle NFDM und AdV in der KTR-AdV startet, agiert in der Rolle „Versicherter“.

## Nachbarsysteme

FLA-NFDM ist integraler Bestandteil von KTR-AdV. Auf der logischen Ebene in KTR-AdV ist FLA-NFDM Teil der Fachlogik der AdV-App. Die Nachbarsysteme der KTR-AdV sind in [gemSpec\_KTR-AdV#3.2] dargestellt.

# Zerlegung des Produkttyps

Eine weitere Untergliederung der Aufbaustruktur von FLA-NFDM ist nicht erforderlich.

# 

# Übergreifende Festlegungen

## Fehlerbehandlung

Tritt bei der Ausführung einer Operation von FLA-NFDM ein Fehler auf, der zum Abbruch der Operation führt, so wird dieser an die aufrufende AdV-App der KTR-AdV gemeldet.

Für das Fehlermanagement gelten neben den hier aufgeführten spezifischen Anforderungen für das FLA-NFDM die Anforderungen aus Kapitel 3 der übergreifenden Spezifikation [gemSpec\_OM].

**⌦ NFDM-A\_2390 Fehlermeldungen inkl. Trace**

FLA-NFDM MUSS sicherstellen, dass Fehlermeldungen einen Trace beinhalten, der den auftretenden Fehler in der Software von der Fehlerursache zurück bis zum auslösenden Operationsaufruf vollständig nachvollziehbar macht.

**⌫**

**⌦ NFDM-A\_2405 Trace ohne Implementierungsdetails**

FLA-NFDM DARF NICHT im Trace der Fehlermeldungen Informationen übermitteln, die Rückschlüsse über die Art und Weise der Implementierung zulassen (z.B. Teile des Programm-Stack-Traces).

**⌫**

**⌦ NFDM-A\_2391 Vorgaben spezifische Fehlermeldungen**

FLA-NFDM MUSS sicherstellen, dass alle spezifischen Fehlermeldungen den Vorgaben gemäß Tabelle „Tab\_FLA\_NFDM\_018 – Fehlermeldungen “ genügen.

**⌫**

## Berechtigungen

### Fachliche Rollen

Der fachliche Akteur (hier Versicherter) ruft fachliche Anwendungsfälle von NFDM zur Anzeige und Bearbeitung der auf seiner eGK gespeicherten Daten auf. Der Versicherte erhält mittels der fachlichen Rolle, die technisch durch das Zugriffsprofil seiner eGK repräsentiert wird, die Autorisierung zur Ausführung des jeweiligen Anwendungsfalls.

### Berechtigung

FLA-NFDM spezifiziert die Fachlogik NFDM. Die Berechtigungen zum Zugriff auf die eGK, die für die Anwendungsfälle NFDM erforderlich sind werden durch KTR-AdV gesteuert und sind nicht Bestandteil von FLA-NFDM.

## Unterstützte Generationen der eGK

**⌦ NFDM-A\_2392 Unterstützte Generationen der eGK**

FLA-NFDM MUSS alle Versionen der eGK der Generationen G2 und höher unterstützen, die ihm bekannt sind. Das FLA-NFDM DARF NICHT die Generationen G0, G1 und G1+ unterstützen.

Falls die ermittelte Kartenversion kleiner als alle zu unterstützenden Kartenversionen ist oder die ermittelte Kartenversion nicht bekannt und kleiner als die größte zu unterstützende Kartenversion ist, MUSS FLA-NFDM von einer inkompatiblen Karte ausgehen und die weitere Verarbeitung der Karte unmittelbar abbrechen.

Falls die ermittelte Kartenversion größer als alle FLA-NFDM bekannten Kartenversionen ist, MUSS FLA-NFDM von einer kompatiblen Karte ausgehen und versuchen, diese zu verarbeiten.

**⌫**

Die zu unterstützenden Versionen der Karten ergeben sich aus den in der zu dieser Spezifikation gehörigen Dokumentenlandkarte aufgeführten zugelassenen Produkttyp-versionen.

## Versionierung

FLA-NFDM ist integraler Bestandteil des Produkttyps KTR-AdV und unterliegt der Versionierung von KTR-AdV. Deshalb ist für FLA-NFDM keine separate Versionierung vorgesehen.

## Protokollierung

Für die Protokollierung von FLA-NFDM in KTR-AdV gelten die Anforderungen zum Logging aus [gemSpec\_KTR-AdV#5.2]. Darüberhinaus gelten in FLA-NFDM die folgenden Anforderungen.

**⌦ NFDM-A\_2393 Verbot Protokollierung medizinischer Daten**

FLA-NFDM DARF NICHT medizinische Daten des Versicherten protokollieren.

**⌫**

**⌦ NFDM-A\_2394 Verbot Protokollierung Schlüsselmaterial**

FLA-NFDM DARF NICHT geheimes Schlüsselmaterial protokollieren.

**⌫**

**⌦ NFDM-A\_2395 Verbot Protokollierung personenbezogener Daten**

FLA-NFDM DARF NICHT personenbezogene Daten des Versicherten protokollieren.

**⌫**

### Zugriffsprotokolleinträge auf der eGK

**⌦ NFDM-A\_2396 Zugriffsprotokollierung auf der eGK**

FLA-NFDM MUSS die in Tabelle „Tab\_FLA\_NFDM\_002 – Werte der Zugriffsprotokolleinträge auf der eGK“ definierten Werte für die Informationselemente eines Zugriffsprotokolleintrags auf der eGK verwenden.

Tabelle 3: Tab\_FLA\_NFDM\_002 – Werte der Zugriffsprotokolleinträge auf der eGK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **Data Type** | **Type of Access** | | **Timestamp, Actor-ID, Actor-Name** |
| ReadDPE | c | R | Lesen | gemäß Tabelle „Tab\_Karten\_Fach\_TIP\_010 \_Struktur“ in [gemSpec\_Karten\_Fach\_TIP#4.1] |
| WriteDPE | c | W | Schreiben |
| EraseDPE | c | E | Löschen |
| GetData | c | R | Lesen des Datensatzes zum Kopieren |
| PutData | c | C | Kopierter Datensatz der alten eGK |
| *Hinweis:*  *Data Type „c“ bedeutet, dass der Kartenzugriff auf Daten der Anwendung DPE in der KTR-AdV-Umgebung erfolgt.* | | | | |

**⌫**

## Datenspeicherung

**⌦ NFDM-A\_2397 Verbot der persistenten Speicherung medizinischer Daten**

FLA-NFDM DARF NICHT medizinische Daten des Versicherten persistent speichern.

**⌫**

## Meldungen am Kartenterminal

**⌦ NFDM-A\_2399 Terminalanzeigen für PIN-Eingabe**

FLA-NFDM SOLL im Rahmen der PIN-Eingaben bei Aufruf der Umgebungsoperation ENV\_TUC\_SECRET\_INPUT am Kartenterminal bei Operationen der Schnittstelle I\_DPE\_Management und I\_COPY\_DATA die in Tabelle „Tabelle 4: Tab\_FLA\_NFDM\_003 – Terminal-Anzeigen für PIN-Eingabe“ definierten Werte als Parameter für die PIN-Verifizierung verwenden.

**⌫**

Wenn es sich um ein Kartenterminal ohne Anzeige handelt, dann kann die Anforderung NFDM-A\_2399 nicht umgesetzt werden. Die Anzeige wird dann ausschließlich durch die AdV-App in KTR-AdV gemäß „Tabelle 4: Tab\_FLA\_NFDM\_003 – Terminal-Anzeigen für PIN-Eingabe“ realisiert.

Tabelle 4: Tab\_FLA\_NFDM\_003 – Terminal-Anzeigen für PIN-Eingabe

|  |  |
| --- | --- |
| **Operation** | **Terminalanzeige** |
| ReadDPE  WriteDPE  EraseDPE  GetDPE  PutDPE | PIN•**0x0B**für•**0x0B**pers.•Erklärungen  **0x0F**Versicherten-PIN: |
| 0x0B mögliches Trennzeichen: Ein Display eines Kartenterminals hat physikalisch bedingt eine begrenzte Anzahl von Zeichen je Zeile.“0x0B“ bezeichnet die Position im anzuzeigenden Text, an welcher der Zeilenumbruch möglich ist, für den Fall dass das Display nicht den gesamten Text in einer Zeile darstellen kann. Ein Zeilenumbruch an anderen Stellen im Text ist nicht erlaubt.  0x0F Trennzeichen zwischen Nachricht und PIN-Prompt)  Leerzeichen werden als „•“ dargestellt.  Die Zeilenumbrüche in der Spalte „Terminalanzeige“ sind editorisch bedingt. | |

## Übergreifende Vorbedingungen

Vorbedingungen sind Bedingungen, die erfüllt sein müssen, bevor eine Operation von FLA-NFDM aufgerufen werden kann. Die Bedingungen müssen von der aufrufenden Komponente in KTR-AdV durchgesetzt werden.

Die folgenden „Übergreifenden Vorbedingungen“ gelten für für alle Operationen der in Kapitel 6 definierten Services und werden von der aufrufenden Komponente vor dem Aufruf der Operationen der Schnittstelle I\_DPE\_Management und I\_COPY\_DATA in KTR-AdV geprüft:

Tabelle 5: Tab\_FLA\_NFDM\_004 – Übergreifende Vorbedingungen für FLA-NFDM

|  |  |
| --- | --- |
|  | **Bedingung** |
| ÜV1 | Die übergebenen Werte der Parameter sind gültig gemäß der Signatur der jeweiligen Operation. |
| ÜV2 | Die Gesundheitsanwendung der eGK (DF.HCA) ist nicht gesperrt (= ist nicht deaktiviert). |
| ÜV3 | Die beteiligte eGK hat keine ältere Versionsnummer als die der Generation 2. |
| ÜV4 | Die beteiligten Smartcards sind echt. |
| ÜV5 | Der DPE auf der eGK ist sichtbar, d. h. der Ordner DF.DPE der  eGK muss als Wert des Attributs lifeCycleStatus den Wert  „active“ haben |

Darüberhinaus können in den Operationen weitere Vorbedingungen definiert werden.

## Übergreifende Nachbedingungen

Erfolgsbedingungen sind Bedingungen, die erfüllt sein müssen, damit eine Operation von FLA-NFDM erfolgreich durchlaufen werden kann. Jede der Bedingungen wird von den Operationen im Rahmen der Verarbeitung überprüft.

Folgende „Übergreifende Erfolgsbedingungen“ gelten für alle Operationen der in Kapitel 6 definierten Services. Sie werden in allen Operationsspezifikationen des Kapitel 6 referenziert.

Tabelle 6: Tab\_FLA\_NFDM\_005 – Übergreifende Erfolgsbedingungen

|  |  |
| --- | --- |
|  | **Bedingung** |
|  | keine |

Folgende „Übergreifende Nachbedingungen“ gelten für alle Operationen der in Kapitel 6 definierten Services. Sie werden in allen Operationsspezifikationen des Kapitel 6 referenziert.

Tabelle 7: Tab\_FLA\_NFDM\_006 – Übergreifende Nachbedingungen

|  |  |
| --- | --- |
|  | **Bedingung** |
| ÜN1 | Ist die ermittelte Version eGK cardVersion G2.1 oder höher dann wurde ein Zugriffsprotokolleintrag gemäß Tabelle „Tab\_FLA\_NFDM\_002 – Werte der Zugriffsprotokolleinträge auf der eGK“ als Record im EF.Logging auf der eGK geschrieben, andernfalls wurde kein Protokolleintrag geschrieben (sondern erfolgt in KTR-AdV). |

# 

# Funktionsmerkmale

In diesem Kapitel werden die Schnittstellen von FLA-NFDM definiert. Für jede Operation wird das an der jeweiligen Schnittstelle sichtbare und damit testbare Verhalten normativ in tabellarischer Form spezifiziert.

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 sind,
* wie erforderliche Zugriffsrechte erworben werden können, und
* wie auf Daten zugegriffen oder gewisse Aktionen ausgeführt werden

## Schnittstelle I\_DPE\_Management

FLA-NFDM stellt mit der Schnittstelle I\_DPE\_Management Operationen in KTR-AdV zur Verfügung, die die fachlichen Anwendungsfälle von NFDM zum Lesen, Schreiben und Löschen des Datensatzes „Persönliche Erklärungen“ auf der eGK unterstützen.   
Es handelt sich hierbei um eine interne Schnittstelle, deren Signatur und Verhalten definiert wird, jedoch keine Festlegungen hinsichtlich Implementierung getroffen werden.

Die Schnittstelle I\_DPE\_Management umfasst die 3 Operationen:

* ReadDPE: DPE von eGK lesen
* WriteDPE: DPE auf eGK schreiben
* EraseDPE: DPE von eGK löschen

### Schnittstellendefinition

#### ReadDPE

**⌦ NFDM-A\_2400 ReadDPE**

FLA-NFDM MUSS für KTR-AdV die Operation ReadDPE gemäß Tabelle 8: Tab\_FLA\_NFDM\_007 – Operation ReadDPE anbieten.

Tabelle 8: Tab\_FLA\_NFDM\_007 – Operation ReadDPE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | ReadDPE | | | | |
| **Beschreibung** | Die Operation liest den DPE im Informationselement DPE der Datei EF.DPE von der eGK und gibt ihn über den Parameter DPEDocument an den Aufrufer zurück. | | | | |
| **Vorbedingungen** | In KTR-AdV werden vor Aufruf dieser Operation die übergreifenden Vorbedingungen aus Tabelle „Tab\_FLA\_NFDM\_004 – Übergreifende Vorbedingungen für FLA-NFDM“ geprüft. | | | | |
| **Erfolgsbedingungen** | Die Operation MUSS die folgenden Erfolgsbedingungen überprüfen. | | | | |
|  | **Bedingung** | | | |
| E1 | Der auf der eGK gespeicherte DPE ist technisch konsistent, d. h. der Wert des Informationselement Status der Datei EF.StatusDPE der eGK ist „0“. | | | |
| E2 | Die Version der internen Speicherstruktur (s. 6.3.1) der Dateien der Anwendung „Datensatz Persönliche Erklärungen“ der eGK (DPE-Speicherstruktur) wird von FLA-NFDM unterstützt. | | | |
| E3 | Es ist ein gemäß [RFC1952] gzip-komprimierter DPE auf der eGK im Informationselement DPE der Datei EF.DPE gemäß [gemSpec\_eGK\_Fach\_NFDM#3.1] gespeichert, d. h. das Informationselement Länge DPE der Datei EF.DPE der eGK hat einen Wert ungleich ´00 00´. | | | |
| E4 | Der auf der eGK gespeicherte DPE ist valide gegen das XML-Schema für den DPE (s. [gemSpec\_InfoNFDM#5]). | | | |
| **Aufrufparameter** | keine | | | | |
| **Rückgabe** | DPEDocument | | | | Von der eGK des Versicherten gelesener, dekomprimierter und validierter DPE gemäß gemSpec\_KTR-AdV |
| **Nachbedingungen** | Falls alle Erfolgsbedingungen erfüllt sind, MUSS die Operation die übergreifenden Nachbedingungen gemäß Tabelle „Tab\_FLA\_NFDM\_006 – Übergreifende Nachbedingungen“ und die folgenden erfüllen. | | | | |
|  | | **Bedingung** | | |
| N1 | | Der DPE (Ausgabeparameter DPEDocument) ist dekomprimiert. | | |
| **Ablauf** | Der Ablauf der Operation ReadDPE, der das definierte Außenverhalten abbildet, ist im Aktivitätsdiagramm der Abbildung 1: Abb\_FLA\_NFDM\_001 – Ablauf ReadDPE modelliert. Die Umsetzung des Ablaufs mittels Plattformbausteinen von KTR-AdV und interner Operationen von FLA-NFDM spezifiziert Tabelle 11: Tab\_FLA\_NFDM\_010 – Umsetzung Ablaufaktivitäten ReadDPE. Der Hersteller kann von der Umsetzung bzw. den spezifizierten Abläufen (z. B. zum Zwecke der Performanceoptimierung) abweichen, falls dadurch das definierte Außenverhalten der Operation gewährleistet bleibt. | | | | |
| **Fehlermeldungen** | Für die generischen Fehlermeldungen finden sich die Attribute ErrorType, Severity, Fehlertext, Befüllung Details und Auslösende Bedingung in [gemSpec\_OM#3.2.2]. Für die spezifischen Fehlermeldungen sind die Attribute ErrorType, Severity, Fehlertext und Befüllung Details in Tabelle „Tab\_FLA\_NFDM\_018 – Fehlermeldungen “ definiert.  ***Generische Fehlermeldungen*** | | | | |
| **Code** | | | **Befüllung Details** | |
| 108 | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| 111 | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| ***Spezifische Fehlermeldungen*** | | | | |
| **Code** | | | **Auslösende Bedingung** | |
| 5103 | | | E1 ist nicht erfüllt. | |
| 5104 | | | E2 ist nicht erfüllt. | |
| 5106 | | | Die Dekomprimierung des DPE ist gescheitert. | |
| 5114 | | | E4 ist nicht erfüllt. | |
| 5121 | | | E3 ist nicht erfüllt. | |
| 5500 | | | Jegliches fehlerhafte Verhalten, das nicht durch die anderen Fehlermeldungen erfasst wird. | |



Abbildung 1: Abb\_FLA\_NFDM\_001 – Ablauf ReadDPE

**⌫**

#### WriteDPE

**⌦ NFDM-A\_2401 WriteDPE**

FLA-NFDM MUSS für KTR-AdV die Operation WriteDPE gemäß Tabelle 9: Tab\_FLA\_NFDM\_008 – Operation WriteDPE anbieten.

Tabelle 9: Tab\_FLA\_NFDM\_008 – Operation WriteDPE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | WriteDPE | | | | | |
| **Beschreibung** | Die Operation schreibt den ihr im Parameter DPEDocument übergebenen DPE auf die eGK in das Informationselement DPE der Datei EF.DPE | | | | | |
| **Vorbedingungen** | In KTR-AdV werden vor Aufruf dieser Operation die übergreifenden Vorbedingungen aus Tabelle „Tab\_FLA\_NFDM\_004 – Übergreifende Vorbedingungen für FLA-NFDM“ und folgende Vorbedingung geprüft. | | | | | |
|  | | | **Bedingung** | | |
| V1 | | | Das Zertifikat C.CH.AUTN der eGK ist zeitlich gültig und nicht gesperrt. | | |
| **Erfolgsbedingungen** | Die Operation MUSS die folgenden Erfolgsbedingungen überprüfen. | | | | | |
|  | **Bedingung** | | | | |
| E1 | Der übergebene DPE ist valide gegen das XML-Schema für den DPE (s. [gemSpec\_InfoNFDM#5]). | | | | |
| E2 | Der Versicherte der eGK ist mit dem Versicherten des DPE identisch, d. h. die Versicherten-ID (die ersten zehn unveränderbaren Stellen der Krankenversichertennummer (KVNR)) im Feld organizationalUnitName des Subject Distinguished Name (SubjectDN) des Authentisierungszertifikats der eGK (gespeichert in der Datei EF.C.CH.AUTN.R2048) ist die gleiche wie die im DPE im Element Versicherten\_ID gespeicherte.  Anmerkung: Es gibt zwei organizationalUnitName-Felder im SubjectDN. 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]). | | | | |
| E3 | Der DPE ist nicht größer als der auf der eGK im Informationselement DPE der Datei EF.DPE der eGK zur Verfügung stehende Speicherplatz. | | | | |
| **Aufrufparameter** | DPEDocument | | | | | Auf die eGK des Versicherten zu schreibender DPE gemäß gemSpec\_KTR-AdV |
| **Rückgabe** | keine | | | | | |
| **Nachbedingungen** | Falls alle Erfolgsbedingungen erfüllt sind, MUSS die Operation die übergreifenden Nachbedingungen gemäß Tabelle „Tab\_FLA\_NFDM\_006 – Übergreifende Nachbedingungen“ und die folgenden erfüllen. | | | | | |
|  | | **Bedingung** | | | |
| N1 | | Die Größe des DPE in Oktett ist im Informationselement Länge DPE der Datei EF.DPE der eGK gemäß [gemSpec\_eGK\_Fach\_NFDM#3.1] gespeichert. | | | |
| N2 | | Der übergebene DPE ist gemäß [RFC1952] gzip-komprimiert auf der eGK im Informationselement DPE der Datei EF.DPE gemäß [gemSpec\_eGK\_Fach\_NFDM#3.1] gespeichert. | | | |
| N3 | | Der Wert des Informationselements Timestamp der Datei EF.StatusDPE der eGK ist aktualisiert. | | | |
| N4 | | Der Wert des Informationselements Version\_Speicherstruktur der Datei EF.StatusDPE der eGK ist aktualisiert mit einer gemäß [gemSpec\_eGK\_Fach\_NFDM#3.2] gültigen Versionsnummer. | | | |
| N5 | | Der Wert des Informationselement Status der Datei EF.StatusDPE der eGK ist „0“. | | | |
| **Ablauf** | Der Ablauf der Operation WriteDPE, der das definierte Außenverhalten abbildet, ist im Aktivitätsdiagramm der Abbildung „Abb\_FLA\_NFDM\_002 – Ablauf WriteDPE“ modelliert. Die Umsetzung des Ablaufs mittels Plattformbausteinen von KTR-AdV und interner Operationen FLA-NFDM spezifiziert Tabelle 12: Tab\_FLA\_NFDM\_011 – Umsetzung Ablaufaktivitäten WriteDPE. Der Hersteller kann von der Umsetzung bzw. den spezifizierten Abläufen (z. B. zum Zwecke der Performanceoptimierung) abweichen, falls dadurch das definierte Außenverhalten der Operation gewährleistet bleibt. | | | | | |
| **Fehlermeldungen** | Für die generischen Fehlermeldungen finden sich die Attribute ErrorType, Severity, Fehlertext, Befüllung Details und Auslösende Bedingung in [gemSpec\_OM#3.2.2]. Für die spezifischen Fehlermeldungen sind die Attribute ErrorType, Severity, Fehlertext und Befüllung Details in Tabelle „Tab\_FLA\_NFDM\_018 – Fehlermeldungen “ definiert.  ***Generische Fehlermeldungen*** | | | | | |
| **Code** | | | | **Befüllung Details** | |
| 108 | | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| 112 | | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| ***Spezifische Fehlermeldungen*** | | | | | |
| **Code** | | | | **Auslösende Bedingung** | |
| 5000 | | | | Die eGK ist defekt. | |
| 5108 | | | | E2 ist nicht erfüllt. | |
| 5110 | | | | Die Komprimierung des DPE ist gescheitert. | |
| 5113 | | | | E3 ist nicht erfüllt | |
| 5114 | | | | E1 nicht erfüllt. | |
| 5500 | | | | Jegliches fehlerhafte Verhalten, das nicht durch die anderen Fehlermeldungen erfasst wird. | |



Abbildung 2: Abb\_FLA\_NFDM\_002 – Ablauf WriteDPE

**⌫**

#### EraseDPE

**⌦ NFDM-A\_2402 EraseDPE**

FLA-NFDM MUSS für KTR-AdV die Operation EraseDPE gemäß Tabelle 10: Tab\_FLA\_NFDM\_009 – Operation EraseDPE anbieten.

Tabelle 10: Tab\_FLA\_NFDM\_009 – Operation EraseDPE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | EraseDPE | | | |
| **Beschreibung** | Die Operation löscht den DPE im Informationselement DPE der Datei EF.DPE der eGK. | | | |
| **Vorbedingungen** | In KTR-AdV werden vor Aufruf dieser Operation die übergreifenden Vorbedingungen aus Tabelle „Tab\_FLA\_NFDM\_004 – Übergreifende Vorbedingungen für FLA-NFDM“ geprüft. | | | |
| **Erfolgsbedingungen** | Die Operation MUSS die folgenden Erfolgsbedingungen überprüfen. | | | |
|  | **Bedingung** | | |
| E1 | Die Version der internen Speicherstruktur (s. 6.3.1) der Dateien der Anwendung „Datensatz Persönliche Erklärungen“ der eGK (DPE-Speicherstruktur) wird FLA-NFDM unterstützt. | | |
| **Aufrufparameter** | keine | | | |
| **Rückgabe** | keine | | | |
| **Nachbedingungen** | Falls alle Erfolgsbedingungen erfüllt sind, MUSS die Operation die übergreifenden Nachbedingungen gemäß Tabelle „Tab\_FLA\_NFDM\_006 – Übergreifende Nachbedingungen“ und die folgenden erfüllen. | | | |
|  | | **Bedingung** | |
| N1 | | Der Wert des Informationselements Länge DPE der Datei EF.DPE der eGK ist ´00 00´. | |
| N2 | | Der DPE im Informationselement DPE der Datei EF.DPE der eGK ist gelöscht, d.h. alle ursprünglich vom DPE belegten Oktette sind mit ´00´ (NULL) überschrieben worden. | |
| N3 | | Der Wert des Informationselements Timestamp der Datei EF.StatusDPE der eGK ist aktualisiert. | |
| N4 | | Der Wert des Informationselements Version\_Speicherstruktur der Datei EF.StatusDPE der eGK ist aktualisiert mit einer gemäß [gemSpec\_eGK\_Fach\_NFDM#3.2] gültigen Versionsnummer. | |
| N5 | | Der Wert des Informationselements Status der Datei EF.StatusDPE der eGK ist „0“. | |
| **Ablauf** | Der Ablauf der Operation EraseDPE, der das definierte Außenverhalten abbildet, ist im Aktivitätsdiagramm der Abbildung 3: Abb\_FLA\_NFDM\_003 – Ablauf EraseDPE modelliert. Die Umsetzung des Ablaufs mittels Plattformbausteinen von KTR-AdV und interner Operationen von FLA-NFDM spezifiziert Tabelle 13: Tab\_FLA\_NFDM\_012 – Umsetzung Ablaufaktivitäten EraseDPE. Der Hersteller kann von der Umsetzung bzw. den spezifizierten Abläufen (z. B. zum Zwecke der Performanceoptimierung) abweichen, falls dadurch das definierte Außenverhalten der Operation gewährleistet bleibt. | | | |
| **Fehlermeldungen** | Für die generischen Fehlermeldungen finden sich die Attribute ErrorType, Severity, Fehlertext, Befüllung Details und Auslösende Bedingung in [gemSpec\_OM#3.2.2]. Für die spezifischen Fehlermeldungen sind die Attribute ErrorType, Severity, Fehlertext und Befüllung Details in Tabelle „Tab\_FLA\_NFDM\_018 – Fehlermeldungen “ definiert.  ***Generische Fehlermeldungen*** | | | |
| **Code** | | | **Befüllung Details** |
| 108 | | | Der Detailtext KANN den Fehler näher beschreiben. |
| 112 | | | Der Detailtext KANN den Fehler näher beschreiben. |
| ***Spezifische Fehlermeldungen*** | | | |
| **Code** | | | **Auslösende Bedingung** |
| 5000 | | | Die eGK ist defekt. |
| 5104 | | | E1 ist nicht erfüllt. |
| 5112 | | | Löschen des DPE (technisch) gescheitert. |
| 5500 | | | Jegliches fehlerhafte Verhalten, das nicht durch die anderen Fehlermeldungen erfasst wird. |



Abbildung 3: Abb\_FLA\_NFDM\_003 – Ablauf EraseDPE

**⌫**

### Umsetzung

Die folgenden Unterkapitel beschreiben die Umsetzung der Operationsabläufe von I\_DPE\_Management in der KTR-AdV Umgebung.

#### 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** | Technische Konsistenz des DPE auf eGK prüfen | | |
| 2.1 | 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. | | |
| 2.2 | NFDM:checkConsistency | | |
| **Eingangsdaten** | | |
| In 2.1 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. | | |
| **3** | Version der DPE-Speicherstruktur der eGK prüfen | | |
|  | NFDM:checkContainerVersion | | |
| **Eingangsdaten** | | |
| In 2.1 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. | | |
| **4** | Existenz DPE auf eGK prüfen | | |
|  | Die Prüfung, ob ein DPE auf der eGK existiert, erfolgt implizit in 5.  In 5 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** | DPE von eGK lesen | | |
| 5.1 | 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. | | |
| 5.2 | NFDM:checkSize | | |
| **Eingangsdaten** | | |
| In 5.1 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. | | |
| 5.3 | 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 5.2 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 5.3 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 | | |

#### 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** | 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** | Größe des DPE auf eGK schreiben | | | |
| 6.1 | Größe des DPE ermitteln | | | |
| NFDM:determineSize | | | |
| **Eingangsdaten** | | | |
| Der in 4 komprimierte DPE | | | |
| **Beschreibung** | | | |
| Die Größe (Anzahl Oktette) des DPE wird bestimmt. | | | |
| 6.2 | Das Schreiben der in 6.1 ermittelten Größe des DPE auf die eGK erfolgt zusammen mit dem Schreiben des DPE in 7. | | | |
| **7** | DPE auf eGK schreiben | | | |
| 7.1 | NFDM:concatenate | | | |
| **Eingangsdaten** | | | |
| DPE und die in 6.1 ermittelte Größe (Anzahl Oktette) des DPE | | | |
| **Beschreibung** | | | |
| Die in 6.1 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. | | | |
| 7.2 | Plattformbaustein PL\_TUC\_CARD\_UPDATE\_FILE | | | |
| **Eingangsdaten** | | | |
| IDENTIFIKATOR | | | EF.DPE |
| **Beschreibung** | | | |
| Die in 7.1 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 ausgeführt und die Operation mit dem Fehler 5113 abgebrochen.  In allen anderen Fehlerfällen bricht die Operation mit Fehler 112 ab. | | | |
| 8 | DPE-Status-Flag auf eGK zurücksetzen | | | |
| Falls die Größenprüfung beim Aufruf von PL\_TUC\_CARD\_UPDATE\_FILE in 7.2 mit Fehler `DataTooBig` endete. | | | | |
| 8.1 | 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. | | | |
| 8.2 | Die Operation bricht mit dem Fehler 5113 ab. | | | |
| **9** | DPE-Zeitstempel auf eGK aktualisieren | | | |
| 9.1 | NFDM:getTime | | | |
| **Beschreibung** | | | |
| Liefere Systemzeit, die durch den Plattformbaustein PL\_TUC\_NET\_SYNC\_TIME bereitgestellt wird. | | | |
| 9.2 | 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 11.2. 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. | | | |
| **10** | 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. | | | |
| **11** | DPE-Status-Flag auf eGK zurücksetzen | | | |
| 11.1 | NFDM:concatenate | | | |
| **Eingangsdaten** | | | |
| In 9.2 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. | | | |
| 11.2 | 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. | | | |

#### 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** | Version der DPE-Speicherstruktur der eGK prüfen | | | |
| **2.1** | 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. | | | |
| 2.2 | NFDM:checkContainerVersion | | | |
| **Eingangsdaten** | | | |
| In 2.1 gelesener Wert des Informationselements Version\_Speicherstruktur der Datei EF.StatusDPE der eGK. | | | |
| **Beschreibung** | | | |
| Es wird überprüft, ob FLA-NFDM die in 2.1 gelesene Versionsnummer der DPE-Speicherstruktur der eGK bekannt ist. Ist dies nicht der Fall, bricht die Operation mit Fehler 5104 ab. | | | |
| **3** | 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. | | | |
| **4** | Größenangabe DPE auf eGK auf 0 setzen | | | |
|  | Das Setzen der Größenangabe für den DPE auf der eGK auf 0 erfolgt implizit in 5. | | | |
| **5** | 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. | | | |
| **6** | DPE-Zeitstempel auf eGK aktualisieren | | | |
| 6.1 | NFDM:getTime | | | |
| **Beschreibung** | | | |
| Liefere Systemzeit, die durch den Plattformbaustein PL\_TUC\_NET\_SYNC\_TIME bereitgestellt wird. | | | |
| 6.2 | 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 8.2. 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. | | | |
| **7** | 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. | | | |
| **8** | DPE-Status-Flag auf eGK zurücksetzen | | | |
| 8.1 | NFDM:concatenate | | | |
| **Eingangsdaten** | | | |
| In 6.2 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. | | | |
| 8.2 | 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. | | | |

## Schnittstelle I\_COPY\_DATA

FLA-NFDM stellt mit der Schnittstelle I\_COPY\_DATA Operationen in KTR-AdV zur Verfügung, die in NFDM das Funktionsmerkmal „Datenübertragung bei Kartentausch“ unterstützen. Der Datensatz „Persönliche Erklärungen“ soll hierüber im Rahmen der Bereitstellung einer neuen eGK des Versicherten von der alten Karte auf die neue Karte übertragen werden. Es handelt sich hierbei um eine interne Schnittstelle, deren Signatur und Verhalten definiert wird, jedoch keine Festlegungen hinsichtlich Implementierung getroffen werden.

Die Schnittstelle I\_COPY\_DATA umfasst die 2 Operationen:

* GetData
* PutData

### Schnittstellend­­efinition

#### GetData

**⌦ NFDM-A\_2403 Operation GetData**

FLA-NFDM MUSS für KTR-AdV die Operation GetData des Funktionsmerkmals „Datenübertragung bei Kartentausch“ gemäß Tabelle „Tab\_FLA\_NFDM\_014 – Operation GetData“ anbieten.

Tabelle 14: Tab\_FLA\_NFDM\_014 – Operation GetData

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **GetData** | | | | |
| **Beschreibung** | Die Operation liest den im Aufrufparameter application benannten Datensatz DPE von der eGK und gibt ihn über den Parameter Document an den Aufrufer zurück. | | | | |
| **Vorbedingung** | In KTR-AdV werden vor Aufruf dieser Operation die übergreifenden Vorbedingungen aus Tabelle „Tab\_FLA\_NFDM\_004 – Übergreifende Vorbedingungen für FLA-NFDM“ geprüft. | | | | |
| **Erfolgs­bedingungen** | Die Operation MUSS die folgenden Erfolgsbedingungen überprüfen. | | | | |
|  | **Bedingung** | | | |
| E1 | Der auf der eGK gespeicherte Datensatz ist technisch konsistent, d. h. der Wert des Informationselements Status des Datensatzes in der Datei EF.StatusDPE der eGK ist „0“. | | | |
| E2 | Die Version der internen Speicherstruktur (s. 6.3.1) der auf der eGK wird von FLA-NFDM unterstützt. | | | |
| E3 | Es ist ein gemäß [RFC1952] gzip-komprimierter DPE auf der eGK gemäß [gemSpec\_eGK\_Fach\_NFDM#2.1] gespeichert, d. h. das Informationselement Länge DPE der Datei EF.DPE der eGK hat einen Wert ungleich ´00 00´. | | | |
| E4 | Der auf der eGK gespeicherte Datensatz DPE ist valide gegen das entsprechende XML-Schema für den Datensatz DPE (s. [gemSpec\_InfoNFDM#5]). | | | |
| **Aufruf­parameter** | **Name** | | | | **Beschreibung** |
| objsysVersion | | | | Version des Objektsystems der eGK, von der der Datensatz DPE gelesen werden soll |
| application | | | | Identifier der eGK für den Datensatz, dessen Daten gelesen werden sollen. Folgende Werte sind zulässig: DF.DPE |
| **Rückgabe** | **Name** | | | | **Beschreibung** |
| document | | | | Von der eGK des Versicherten gelesener, dekomprimierter Datensatz DPE, der durch Aufrufparameter application benannt wurde.  Folgende XML-Dokumente sind zulässig:  DPE\_Document gemäß gemSpec\_KTR-AdV |
| **Nach­bedingungen** | Falls alle Erfolgsbedingungen erfüllt sind, MUSS die Operation die folgenden Nachbedingungen erfüllen. | | | | |
|  | | **Bedingung** | | |
| N1 | | Der Datensatz DPE (Ausgabeparameter document) ist dekomprimiert. | | |
| **Ablauf** | Der Ablauf der Operation GetData, ist im Aktivitätsdiagramm der Abbildung „Abbildung 4: Abb\_FLA\_NFDM\_004 – Ablauf GetData“ modelliert. Die Umsetzung des Ablaufs spezifiziert Tabelle „Tabelle 16: Tab\_FLA\_NFDM\_016 – Umsetzung Ablaufaktivitäten GetData“. Der Hersteller kann von der Umsetzung bzw. den spezifizierten Abläufen (z. B. zum Zwecke der Performanceoptimierung) abweichen, falls dadurch das definierte Außenverhalten der Operation gewährleistet bleibt. | | | | |
| **Fehler­meldungen** | Für die generischen Fehlermeldungen finden sich die Attribute ErrorType, Severity, Fehlertext, Befüllung Details und Auslösende Bedingung in [gemSpec\_OM#3.2.2]. Für die spezifischen Fehlermeldungen sind die Attribute ErrorType, Severity, Fehlertext und Befüllung Details in Tabelle „Tab\_FLA\_NFDM\_018 – Fehlermeldungen “ definiert.  ***Generische Fehlermeldungen*** | | | | |
|  | **Code** | | | **Befüllung Details** | |
| 108 | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| 111 | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| **Code** | | | **Befüllung Details** | |
| 5103 | | | E1 ist nicht erfüllt. | |
| 5104 | | | E2 ist nicht erfüllt. | |
| 5106 | | | Die Dekomprimierung des DPE ist gescheitert. | |
| 5114 | | | E4 ist nicht erfüllt. | |
| 5121 | | | E3 ist nicht erfüllt. | |
| 5500 | | | Jegliches fehlerhafte Verhalten, das nicht durch die anderen Fehlermeldungen erfasst wird. | |



Abbildung 4: Abb\_FLA\_NFDM\_004 – Ablauf GetData

**⌫**

#### PutData

**⌦ NFDM-A\_2404 Operation PutData**

FLA-NFDM MUSS für KTR-AdV die Operation PutData des Funktionsmerkmals „Datenübertragung bei Kartentausch“ gemäß Tabelle „Tab\_FLA\_NFDM\_015 – Operation PutData“ anbieten.

Tabelle 15: Tab\_FLA\_NFDM\_015 – Operation PutData

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **PutData** | | | | |
| **Beschreibung** | Die Operation schreibt im Aufrufparameter application benannten Datensatz DPE\_Document gemäß gemSpec\_KTR-AdV in das Informationselement DPE der Datei EF.DPE. | | | | |
| **Vorbedingung** | In KTR-AdV werden vor Aufruf dieser Operation die übergreifenden Vorbedingungen aus Tabelle „Tab\_FLA\_NFDM\_004 – Übergreifende Vorbedingungen für FLA-NFDM“ und folgende Vorbedingung geprüft: | | | | |
|  | | **Bedingung** | | |
| V1 | | Das Zertifikat C.CH.AUTN der eGK ist zeitlich gültig und nicht gesperrt. | | |
| V2 | | Die KVNR der eGK, von der der DPE Datensatz stammt (Document) ist gleich der KVNR der eGK für welche die Operation PutData aufgerufen wird (Die ersten zehn unveränderbaren Stellen der Krankenversichertennummer (KVNR) werden für den Vergleich herangezogen.). | | |
| **Erfolgs­bedingungen** | Die Operation MUSS die folgenden Erfolgsbedingungen überprüfen. | | | | |
|  | **Bedingung** | | | |
| E1 | Der übergebene DPE ist valide gegen das XML-Schema für den DPE (s. [gemSpec\_InfoNFDM#5]). | | | |
| E2 | Der Versicherte der eGK ist mit dem Versicherten des DPE identisch, d. h. die Versicherten-ID (die ersten zehn unveränderbaren Stellen der Krankenversichertennummer (KVNR)) im Feld organizationalUnitName des Subject Distinguished Name (SubjectDN) des Authentisierungszertifikats der eGK (gespeichert in der Datei EF.C.CH.AUTN.R2048) ist die gleiche wie die im DPE im Element Versicherten\_ID gespeicherte.  Anmerkung: Es gibt zwei organizationalUnitName-Felder im SubjectDN. 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]). | | | |
| E3 | Der DPE ist nicht größer als der auf der eGK im Informationselement DPE der Datei EF.DPE der eGK zur Verfügung stehende Speicherplatz. | | | |
| E4 | Es ist auf der eGK kein DPE bereits vorhanden, d. h. das Informationselement Länge DPE der Datei EF.DPE der eGK hat einen Wert gleich ´00 00´. | | | |
| **Aufruf­parameter** | **Name** | | | | **Beschreibung** |
| objsysVersionSrc | | | | Version des Objektsystems der eGK, von der der Datensatz DPE ursprünglich gelesen werden soll |
| objsysVersionDest | | | | Version des Objektsystems der eGK, auf die der Datensatz DPE geschrieben werden soll |
| application | | | | Identifier der eGK für den Datensatz, dessen Daten geschrieben werden sollen. Folgende Werte sind zulässig:  DF.DPE |
| document | | | | Der durch Aufrufparameter application benannte und auf die eGK zu schreibende Datensatz DPE.  Folgende XML-Dokumente sind zulässig:  DPE\_Document gemäß gemSpec\_KTR-AdV |
| **Rückgabe** | **Name** | | | | **Beschreibung** |
| **Nach­bedingungen** | Falls alle Erfolgsbedingungen erfüllt sind, MUSS die Operation die folgenden Nachbedingungen erfüllen. | | | | |
|  | | **Bedingung** | | |
| N1 | | Der im Aufufpameter Document übergebene Datensatz DPE ist gemäß [RFC1952] gzip-komprimiert auf der eGK im Informationselement DPE der Datei EF.DPE gemäß [gemSpec\_eGK\_Fach\_NFDM#3.1] gespeichert. | | |
| N2 | | Die Größe des DPE in Oktett ist im Informationselement Länge DPE der Datei EF.DPE der eGK gemäß [gemSpec\_eGK\_Fach\_NFDM#3.1] gespeichert. | | |
| N3 | | Der Wert des Informationselements Timestamp der Datei EF.StatusDPE der eGK ist aktualisiert. | | |
| N4 | | Der Wert des Informationselements Version\_Speicherstruktur der Datei EF.StatusDPE der eGK ist aktualisiert mit einer gemäß [gemSpec\_eGK\_Fach\_NFDM#3.2] gültigen Versionsnummer. | | |
| N5 | | Der Wert des Informationselement Status der Datei EF.StatusDPE der eGK ist „0“. | | |
| **Ablauf** | Der Ablauf der Operation PutData, ist im Aktivitätsdiagramm der Abbildung „Abb\_FLA\_NFDM\_005 – Ablauf PutData“ modelliert. Die Umsetzung des Ablaufs spezifiziert Tabelle „Tabelle 17: Tab\_FLA\_NFDM\_017 – Umsetzung Ablaufaktivitäten PutData“. Der Hersteller kann von der Umsetzung bzw. den spezifizierten Abläufen (z. B. zum Zwecke der Performanceoptimierung) abweichen, falls dadurch das definierte Außenverhalten der Operation gewährleistet bleibt. | | | | |
| **Fehler­meldungen** | Für die generischen Fehlermeldungen finden sich die Attribute ErrorType, Severity, Fehlertext, Befüllung Details und Auslösende Bedingung in [gemSpec\_OM#3.2.2]. Für die spezifischen Fehlermeldungen sind die Attribute ErrorType, Severity, Fehlertext und Befüllung Details in Tabelle „Tab\_FLA\_NFDM\_018 – Fehlermeldungen “ definiert.  ***Generische Fehlermeldungen*** | | | | |
| **Code** | | | **Befüllung Details** | |
| 108 | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| 112 | | | Der Detailtext KANN den Fehler näher beschreiben. | |
| ***Spezifische Fehlermeldungen*** | | | | |
| **Code** | | | **Auslösende Bedingung** | |
| 5000 | | | Die eGK ist defekt. | |
| 5104 | | | Die Version der internen DPE-Speicherstruktur wird von FLA-NFDM nicht unterstützt. Eine Prüfung der Existenz eines DPE kann nicht erfolgen. | |
| 5108 | | | E2 ist nicht erfüllt. | |
| 5110 | | | Die Komprimierung des DPE ist gescheitert. | |
| 5113 | | | E3 ist nicht erfüllt | |
| 5114 | | | E1 nicht erfüllt. | |
| 5122 | | | E4 ist nicht erfüllt. | |
| 5500 | | | Jegliches fehlerhafte Verhalten, das nicht durch die anderen Fehlermeldungen erfasst wird. | |



Abbildung 5: Abb\_FLA\_NFDM\_005 – Ablauf PutData

**⌫**

### Umsetzung

Die folgenden Unterkapitel beschreiben die Umsetzung der Operationsabläufe von I\_ COPY\_DATA in der KTR-AdV Umgebung.

#### GetData

Tabelle 16: Tab\_FLA\_NFDM\_016 – Umsetzung Ablaufaktivitäten GetData

|  |  |
| --- | --- |
| **1** | Version der eGK setzen |
|  | In CardVersion wird die Information aus dem Aufrufparameter objsysVersion übernommen. |
| **2-8** | Identisch zu 2-8 von ReadDPE (s. Tabelle „Tabelle 11: Tab\_FLA\_NFDM\_010 – Umsetzung Ablaufaktivitäten ReadDPE“) |
|  |  |
| **9** | DPEDocument als Document an Aufrufer zurückgeben |
|  | Rückgabe Document |

#### PutData

Tabelle 17: Tab\_FLA\_NFDM\_017 – Umsetzung Ablaufaktivitäten PutData

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | Version der eGK setzen | | |
|  | In CardVersion wird die Information aus dem Aufrufparameter objsysVersionDest übernommen. | | |
| **2** | Existenz eines DPE prüfen | | |
| **2.1** | Aufruf Plattformbaustein PL\_TUC\_CARD\_READ\_FILE | | |
| **Eingangsdaten** | | |
| IDENTIFIKATOR | EF.StatusDPE | |
| **Beschreibung** | | |
| Es werden alle Informationselemente der Datei EF.StatusDPE der eGK gelesen. | | |
| **2.2** | NFDM:checkConsistency | | |
| **Eingangsdaten** | | |
| In 2.1 aus der Datei EF.StatusPDE gelesene Daten (= alle Informationselemente) | | |
| **Beschreibung** | | |
| Ist der Wert des Informationselements Status der Eingangsdaten „0“, dann ist der DPE technisch konsistent.  Ist der Wert „1“, ist der DPE technisch inkonsistent und kann auch dann überschrieben werden, wenn auf der eGK bereits ein DPE angelegt ist.  Ist der Wert weder „0“ noch „1“ gemäß [gemSpec\_eGK\_Fach\_NFDM#2.2], so wurde noch kein DPE angelegt. | | |
| **2.3** | NFDM:checkContainerVersion | | |
| **Eingangsdaten** | | |
| In 2.1 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. | | |
| **2.4** | Aufruf Plattformbaustein PL\_TUC\_CARD\_READ\_FILE | | |
| **Eingangsdaten** | | |
| IDENTIFIKATOR | | EF.DPE |
| LENGTH | | ´02´ |
| **Beschreibung** | | |
| Es wird das Informationselement Länge DPE der Datei EF.DPE der eGK gelesen.  Tritt dabei ein Fehler auf, bricht die Operation mit Fehler 111 ab. | | |
| **2.5** | NFDM:checkSize | | |
| **Eingangsdaten** | | |
| In 2.1 gelesene Daten aus der Datei EF.DPE. | | |
| **Beschreibung** | | |
| Es wird das Informationselement Länge DPE der Eingangsdaten als Größe des DPE (Anzahl Oktette) ausgewertet. Ist die Größe ungleich ´00 00´, bedeutet dies, es existiert ein DPE auf der eGK.  Existiert ein DPE auf der eGK und ist der Datensatz entsprechend der Prüfung in 2.2 konsistent, so bricht die Operation mit Fehler 5122 ab. | | |
| **3-13** | Identisch zu 2-12 von WriteDPE (s. Tabelle „Tabelle 12: Tab\_FLA\_NFDM\_011 – Umsetzung Ablaufaktivitäten WriteDPE“), wobei der Aufrufparameter Document als DPEDocument verarbeitet wird. | | |

## Artefakte

### DPE-Speicherstruktur der eGK

Die DPE-Speicherstruktur der eGK wird in [gemSpec\_eGK\_Fach\_NFDM] definiert. Die Version der DPE-Speicherstruktur wird im Informationselement Version\_Speicherstruktur der Datei EF.StatusDPE der eGK gespeichert.

### Der DPE auf der eGK

Der DPE ist auf der Karte in der Datei EF.DPE gemäß [RFC1952] gzip-komprimiert gespeichert. Die XML-Struktur des DPE ist in [gemSpec\_InfoNFDM#5] definiert.

## Testunterstützung

Zur Unterstützung von Tests im Zusammenhang mit den Funktionsmerkmalen werden keine gesonderten Festlegungen getroffen.

## Hardwaremerkmale

Die Funktionsmerkmale setzen keine besonderen Hardwaremerkmale voraus.

# Informationsmodell

Das (technische) Informationsmodell NFDM ist in [gemSpec\_InfoNFDM] spezifiziert.

Schnittstellenbeschreibungen und Speicherstruktur der eGK sind im jeweiligen Unterkapitel „Artefakte“ der Funktionsmerkmale aufgeführt.

# Verteilungssicht

FLA-NFDM ist integraler Bestandteil von KTR-AdV. Die Hardware-Merkmale von KTR-AdV sind in [gemSpec\_KTR-AdV] spezifiziert.

Eine weitergehende Darstellung der hardwareseitigen Verteilung von FLA-NFDM bzw. seiner Teilsysteme und der Einbettung in die physikalische Umgebung wird daher nicht benötigt.

# 

# Anhang A – Verzeichnisse

## A1 – Abkürzungen

| **Kürzel** | **Erläuterung** |
| --- | --- |
| AID | Application Identifier |
| AdV | Anwendungen des Versicherten |
| C2C | Card-to-Card |
| DF | Dedicated File |
| DPE | Datensatz „Persönliche Erklärungen“ |
| EF | Elementary File |
| eGK | elektronische Gesundheitskarte |
| FID | File Identifier |
| FLA-NFDM | Fachlogik der Fachanwendung Notfalldatenmanagement auf Seite der AdV-App |
| ID | Identifier |
| KVNR | Krankenversichertennummer |
| NFD | Notfalldatensatz |
| NFDM | Notfalldaten-Management |
| PID | Password Identifier |
| PIN | Personal Identification Number |
| RFC | Request for Comments |
| SM-B | Security Module Card |
| SubjectDN | Subject Distinguished Name |
| TI | Telematikinfrastruktur |

## A2 – Glossar

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

## A3 – Abbildungsverzeichnis

[Abbildung 1: Abb\_FLA\_NFDM\_001 – Ablauf ReadDPE 19](#_Toc489629860)

[Abbildung 2: Abb\_FLA\_NFDM\_002 – Ablauf WriteDPE 22](#_Toc489629861)

[Abbildung 3: Abb\_FLA\_NFDM\_003 – Ablauf EraseDPE 25](#_Toc489629862)

[Abbildung 4: Abb\_FLA\_NFDM\_004 – Ablauf GetData 37](#_Toc489629863)

[Abbildung 5: Abb\_FLA\_NFDM\_005 – Ablauf PutData 40](#_Toc489629864)

## A4 – Tabellenverzeichnis

[Tabelle 1: Tab\_FLA\_NFDM\_001 Verwendete Bezeichner für Objekte der eGK 7](#_Toc489629865)

[Tabelle 2: Tab\_FLA\_NFDM\_019 Abbildung NFDM-Anwendungsfälle auf Operationen des FLA-NFDM 8](#_Toc489629866)

[Tabelle 3: Tab\_FLA\_NFDM\_002 – Werte der Zugriffsprotokolleinträge auf der eGK 13](#_Toc489629867)

[Tabelle 4: Tab\_FLA\_NFDM\_003 – Terminal-Anzeigen für PIN-Eingabe 14](#_Toc489629868)

[Tabelle 5: Tab\_FLA\_NFDM\_004 – Übergreifende Vorbedingungen für FLA-NFDM 14](#_Toc489629869)

[Tabelle 6: Tab\_FLA\_NFDM\_005 – Übergreifende Erfolgsbedingungen 15](#_Toc489629870)

[Tabelle 7: Tab\_FLA\_NFDM\_006 – Übergreifende Nachbedingungen 15](#_Toc489629871)

[Tabelle 8: Tab\_FLA\_NFDM\_007 – Operation ReadDPE 17](#_Toc489629872)

[Tabelle 9: Tab\_FLA\_NFDM\_008 – Operation WriteDPE 20](#_Toc489629873)

[Tabelle 10: Tab\_FLA\_NFDM\_009 – Operation EraseDPE 23](#_Toc489629874)

[Tabelle 11: Tab\_FLA\_NFDM\_010 – Umsetzung Ablaufaktivitäten ReadDPE 26](#_Toc489629875)

[Tabelle 12: Tab\_FLA\_NFDM\_011 – Umsetzung Ablaufaktivitäten WriteDPE 28](#_Toc489629876)

[Tabelle 13: Tab\_FLA\_NFDM\_012 – Umsetzung Ablaufaktivitäten EraseDPE 32](#_Toc489629877)

[Tabelle 14: Tab\_FLA\_NFDM\_014 – Operation GetData 35](#_Toc489629878)

[Tabelle 15: Tab\_FLA\_NFDM\_015 – Operation PutData 37](#_Toc489629879)

[Tabelle 16: Tab\_FLA\_NFDM\_016 – Umsetzung Ablaufaktivitäten GetData 40](#_Toc489629880)

[Tabelle 17: Tab\_FLA\_NFDM\_017 – Umsetzung Ablaufaktivitäten PutData 40](#_Toc489629881)

[Tabelle 18: Tab\_FLA\_NFDM\_018 – Fehlermeldungen FLA-NFDM 48](#_Toc489629882)

## A5 – Referenzierte Dokumente

### A5.1 – Dokumente der gematik

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

| **[Quelle]** | **Herausgeber: Titel** |
| --- | --- |
| [DPE\_Document.xsd] | gematik: XML-Schema-Dokument für den DPE |
| [gemGlossar] | gematik: Glossar der Telematikinfrastruktur |
| [gemLH\_NFDM] | Projektteam NFDM: Lastenheft Notfalldaten-Management |
| [gemSysL\_AdV] | gematik: Systemspezifisches Konzept  Anwendungen des Versicherten (AdV) |
| [gemSysL\_NFDM] | gematik: Systemspezifisches Konzept  Notfalldaten-Management (NFDM) |
| [gemSpec\_Karten\_Fach\_TIP] | gematik: Befüllvorschriften für die Plattformanteile der Karten der TI |
| [gemSpec\_eGK\_Fach\_NFDM] | gematik: Speicherstrukturen der eGK für die Fachanwendung NFDM |
| [gemSpec\_eGK\_ObjSys] | gematik: Spezifikation der elektronischen Gesundheitskarte eGK-Objektsystem |
| [gemSpec\_InfoNFDM] | gematik: Informationsmodell Notfalldaten-Management (NFDM) |
| [gemSpec\_OM] | gematik: Übergreifende Spezifikation Operations und Maintenance (Fehlermanagement, Versionierung, Monitoring) |
| [gemSpec\_PKI] | gematik: Spezifikation PKI |
| [gemSpec\_KTR-AdV] | gematik: Spezifikation KTR-AdV |

### A5.2 – Weitere Dokumente

| [Quelle] | Herausgeber (Erscheinungsdatum): Titel |
| --- | --- |
| [RFC1952] | RFC 1952 (Mai 1996): GZIP file format specification version 4.3, <http://tools.ietf.org/html/rfc1952>(zuletzt geprüft am 31.01.2014) |
| [RFC2119] | RFC 2119 (März 1997): Key words for use in RFCs to Indicate Requirement Levels S. Bradner,[http://tools.ietf.org/html/rfc2109](http://tools.ietf.org/html/rfc2109%20) (zuletzt geprüft am 06.06.2012) |

# Anhang B

## B1 – Fehlermeldungen

Die Tabelle richtet sich nach den Vorgaben von [gemSpec\_OM#5.3.1.2].

Tabelle 18: Tab\_FLA\_NFDM\_018 – Fehlermeldungen FLA-NFDM

| **Code** | **ErrorType** | **Severity** | **ErrorText** | **Befüllung Details** |
| --- | --- | --- | --- | --- |
| 5000 | Technical | FATAL | Die eGK ist defekt. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5103 | Technical | ERROR | Datensatz „Persönliche Erklärungen“ nicht konsistent | Der Detailtext MUSS den Zeitstempel des DPE aus dem Informationselement Timestamp der Datei EF.StatusDPE der eGK enthalten. |
| 5104 | Technical | FATAL | Unbekannte Version der Speicherstruktur für den Datensatz „Persönliche Erklärungen“ auf der eGK | Der Detailtext MUSS die Versionsnummer der DPE-Speicherstruktur auf der eGK (s. Kapitel 6.3.1) enthalten. |
| 5106 | Technical | ERROR | Dekomprimierung des Datensatz „Persönliche Erklärungen“ gescheitert | Der Detailtext KANN den Fehler näher beschreiben. |
| 5108 | Security | ERROR | Die Versicherten-ID des Datensatz „Persönliche Erklärungen“ stimmt nicht mit der Versicherten-ID der eGK überein. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5110 | Technical | ERROR | Die Komprimierung des Datensatz „Persönliche Erklärungen“ ist gescheitert. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5112 | Technical | ERROR | Das Löschen des Datensatz „Persönliche Erklärungen“ ist gescheitert. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5113 | Business | ERROR | Der Datensatz „Persönliche Erklärungen“ überschreitet die maximal zulässige Größe. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5114 | Security | ERROR | Der Datensatz „Persönliche Erklärungen“ ist nicht valide. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5121 | Business | ERROR | Es ist kein Datensatz „Persönliche Erklärungen“ auf der eGK gespeichert. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5122 | Business | ERROR | Es ist bereits ein Datensatz „Persönliche Erklärungen“auf der eGK gespeichert. | Der Detailtext KANN den Fehler näher beschreiben. |
| 5500 | Technical | FATAL | Interner Fehler | Der Detailtext KANN den Fehler näher beschreiben. |