

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.

## Elektronische Gesundheitskarte und Telematikinfrastruktur

# Spezifikation Autorisierung ePA

Version: 1.45.0 [CC](#)  
Revision: [198766230679](#)  
Stand: [02.0330.04.2020](#)  
Status: [zur Abstimmung](#) freigegeben  
Klassifizierung: öffentlich [Entwurf](#)  
Referenzierung: gemSpec\_Autorisierung

22

## Dokumentinformationen

### Änderungen zur Vorversion

Anpassungen des vorliegenden Dokumentes im Vergleich zur Vorversion können Sie der nachfolgenden Tabelle entnehmen.

26

### Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	18.12.18		freigegeben	gematik
1.1.0	15.05.19		Einarbeitung Änderungsliste P18.1	gematik
1.2.0	28.06.19		Einarbeitung Änderungsliste P19.1	gematik
1.3.0	02.10.19		Einarbeitung Änderungsliste P20.1/2	gematik
			Einarbeitung Änderungsliste P21.1	gematik
1.4.0	02.03.20		freigegeben	gematik
<a href="#">1.5.0 CC</a>	<a href="#">30.04.20</a>		<a href="#">Anpassungen gemäß Änderungsliste P22.1 und Scope-Themen aus Systemdesign R4.0.0</a>	<a href="#">gematik</a>

## Inhaltsverzeichnis

29	<b>1 Einordnung des Dokumentes .....</b>	<b>7</b>
30	<b>1.1 Zielsetzung .....</b>	<b>7</b>
31	<b>1.2 Zielgruppe .....</b>	<b>7</b>
32	<b>1.3 Geltungsbereich .....</b>	<b>7</b>
33	<b>1.4 Abgrenzungen .....</b>	<b>7</b>
34	<b>1.5 Methodik .....</b>	<b>8</b>
35	<b>2 Systemüberblick .....</b>	<b>9</b>
36	<b>3 Systemkontext .....</b>	<b>10</b>
37	<b>3.1 Akteure und Rollen .....</b>	<b>10</b>
38	<b>3.2 Nachbarsysteme .....</b>	<b>13</b>
39	<b>3.3 Tokenbasierte Autorisierung .....</b>	<b>14</b>
40	<b>4 Zerlegung der Komponente Autorisierung .....</b>	<b>15</b>
41	<b>5 Übergreifende Festlegungen .....</b>	<b>16</b>
42	<b>5.1 Datenschutz und Datensicherheit .....</b>	<b>16</b>
43	<b>5.2 Verwendete Standards .....</b>	<b>20</b>
44	<b>5.3 Protokollierung .....</b>	<b>21</b>
45	<b>5.4 Fehlerbehandlung in Schnittstellenoperationen .....</b>	<b>23</b>
46	<b>5.5 Nicht-funktionale Anforderungen .....</b>	<b>25</b>
47	5.5.1 Skalierbarkeit .....	25
48	5.5.2 Performance .....	25
49	5.5.3 Mengengerüst .....	25
50	<b>6 Funktionsmerkmale .....</b>	<b>26</b>
51	<b>6.1 Übergreifende Festlegungen .....</b>	<b>26</b>
52	<b>6.2 Schnittstellen der Komponente Autorisierung .....</b>	<b>28</b>
53	6.2.1 Schnittstelle I_Authorization .....	31
54	6.2.1.1 Operationsdefinition I_Authorization::getAuthorizationKey .....	31
55	6.2.1.2 Umsetzung I_Authorization::getAuthorizationKey .....	33
56	6.2.2 Schnittstelle I_Authorization_Insurant .....	34
57	6.2.2.1 Operationsdefinition I_Authorization_Insurant::getAuthorizationKey .....	35
58	6.2.2.2 Umsetzung I_Authorization_Insurant::getAuthorizationKey .....	37
59	6.2.3 Schnittstelle I_Authorization_Management .....	38
60	6.2.3.1 Operationsdefinition I_Authorization_Management::putAuthorizationKey .....	38
61	6.2.3.2 Umsetzung I_Authorization_Management::putAuthorizationKey .....	40
62	6.2.3.3 Operationsdefinition I_Authorization_Management::checkRecordExists .....	41
63	6.2.3.4 Umsetzung I_Authorization_Management::checkRecordExists .....	42
64	6.2.3.5 Operationsdefinition I_Authorization_Management::getAuthorizationList .....	42
65	6.2.3.6 Umsetzung I_Authorization_Management::getAuthorizationList .....	43

66	6.2.4 Schnittstelle I_Authorization_Management_Insurant .....	44
67	6.2.4.1 Operationsdefinition	
68	I_Authorization_Management_Insurant::putAuthorizationKey .....	44
69	6.2.4.2 Umsetzung I_Authorization_Management_Insurant::putAuthorizationKey	
70	.....	46
71	6.2.4.3 Operationsdefinition	
72	I_Authorization_Management_Insurant::deleteAuthorizationKey .....	48
73	6.2.4.4 Umsetzung	
74	I_Authorization_Management_Insurant::deleteAuthorizationKey .....	50
75	6.2.4.5 Operationsdefinition	
76	I_Authorization_Management_Insurant::replaceAuthorizationKey .....	50
77	6.2.4.6 Umsetzung	
78	I_Authorization_Management_Insurant::replaceAuthorizationKey .....	52
79	6.2.4.7 Operationsdefinition	
80	I_Authorization_Management_Insurant::getAuditEvents .....	53
81	6.2.4.8 Umsetzung I_Authorization_Management_Insurant::getAuditEvents .....	54
82	6.2.4.9 Operationsdefinition	
83	I_Authorization_Management_Insurant::putNotificationInfo .....	55
84	6.2.4.10 Umsetzung I_Authorization_Management_Insurant::putNotificationInfo .....	57
85	6.2.4.11 Operationsdefinition	
86	I_Authorization_Management_Insurant::getAuthorizationList .....	57
87	6.2.4.12 Umsetzung I_Authorization_Management_Insurant::getAuthorizationList	
88	.....	59
89	<b>6.3 Berechtigungstypen der Autorisierung .....</b>	<b>60</b>
90	<b>6.4 Hardware-Merkmal der Komponente Autorisierung .....</b>	<b>60</b>
91	<b>6.5 Geräteverwaltung .....</b>	<b>61</b>
92	6.5.1 Freischaltprozess neuer Geräte .....	61
93	6.5.2 Geräteadministration .....	64
94	<b>6.6 Freischaltprozess Vertretereinrichtung .....</b>	<b>65</b>
95	<b>7 Informationsmodell .....</b>	<b>68</b>
96	7.1 Namensräume .....	69
97	7.2 SAML-Profil und Tokeninhalte .....	69
98	<b>8 Verteilungssicht .....</b>	<b>73</b>
99	<b>9 Anhang A Verzeichnisse .....</b>	<b>74</b>
100	9.1 Abkürzungen .....	74
101	9.2 Glossar .....	74
102	9.3 Abbildungsverzeichnis .....	74
103	9.4 Tabellenverzeichnis .....	75
104	9.5 Referenzierte Dokumente .....	76
105	9.5.1 Dokumente der gematik .....	76
106	9.5.2 Weitere Dokumente .....	77
107	<b>1 Einordnung des Dokumentes .....</b>	<b>7</b>
108	1.1 Zielsetzung .....	7

109	<b>1.2 Zielgruppe .....</b>	<b>7</b>
110	<b>1.3 Geltungsbereich .....</b>	<b>7</b>
111	<b>1.4 Abgrenzungen .....</b>	<b>7</b>
112	<b>1.5 Methodik .....</b>	<b>8</b>
113	<b>2 Systemüberblick .....</b>	<b>9</b>
114	<b>3 Systemkontext.....</b>	<b>10</b>
115	<b>3.1 Akteure und Rollen .....</b>	<b>10</b>
116	<b>3.2 Nachbarsysteme .....</b>	<b>13</b>
117	<b>3.3 Tokenbasierte Autorisierung .....</b>	<b>14</b>
118	<b>4 Zerlegung der Komponente Autorisierung .....</b>	<b>15</b>
119	<b>5 Übergreifende Festlegungen .....</b>	<b>16</b>
120	<b>5.1 Datenschutz und Datensicherheit .....</b>	<b>16</b>
121	<b>5.2 Verwendete Standards .....</b>	<b>20</b>
122	<b>5.3 Protokollierung.....</b>	<b>21</b>
123	<b>5.4 Fehlerbehandlung in Schnittstellenoperationen .....</b>	<b>23</b>
124	<b>5.5 Nicht-Funktionale Anforderungen.....</b>	<b>25</b>
125	5.5.1 Skalierbarkeit .....	25
126	5.5.2 Performance .....	25
127	5.5.3 Mengengerüst.....	25
128	<b>6 Funktionsmerkmale .....</b>	<b>26</b>
129	<b>6.1 Übergreifende Festlegungen.....</b>	<b>26</b>
130	<b>6.2 Schnittstellen der Komponente Autorisierung .....</b>	<b>28</b>
131	6.2.1 Schnittstelle I Authorization .....	31
132	6.2.1.1 Operationsdefinition I Authorization::getAuthorizationKey .....	31
133	6.2.1.2 Umsetzung I Authorization::getAuthorizationKey .....	33
134	6.2.2 Schnittstelle I Authorization Insurant .....	34
135	6.2.2.1 Operationsdefinition I Authorization Insurant::getAuthorizationKey .....	35
136	6.2.2.2 Umsetzung I Authorization Insurant::getAuthorizationKey .....	37
137	6.2.3 Schnittstelle I Authorization Management .....	38
138	6.2.3.1 Operationsdefinition I Authorization Management::putAuthorizationKey .....	38
139	6.2.3.2 Umsetzung I Authorization Management::putAuthorizationKey .....	40
140	6.2.3.3 Operationsdefinition I Authorization Management::checkRecordExists .....	41
141	6.2.3.4 Umsetzung I Authorization Management::checkRecordExists .....	42
142	6.2.3.5 Operationsdefinition I Authorization Management::getAuthorizationList .....	42
143	6.2.3.6 Umsetzung I Authorization Management::getAuthorizationList .....	43
144	6.2.4 Schnittstelle I Authorization Management Insurant .....	44
145	6.2.4.1 Operationsdefinition .....	
146	I Authorization Management Insurant::putAuthorizationKey .....	44
147	6.2.4.2 Umsetzung I Authorization Management Insurant::putAuthorizationKey .....	
148	.....	46
149	6.2.4.3 Operationsdefinition .....	
150	I Authorization Management Insurant::deleteAuthorizationKey .....	48

151	6.2.4.4 Umsetzung	
152	I Authorization Management Insurant::deleteAuthorizationKey .....	50
153	6.2.4.5 Operationsdefinition	
154	I Authorization Management Insurant::replaceAuthorizationKey .....	50
155	6.2.4.6 Umsetzung	
156	I Authorization Management Insurant::replaceAuthorizationKey .....	52
157	6.2.4.7 Operationsdefinition	
158	I Authorization Management Insurant::getAuditEvents .....	53
159	6.2.4.8 Umsetzung I Authorization Management Insurant::getAuditEvents .....	54
160	6.2.4.9 Operationsdefinition	
161	I Authorization Management Insurant::putNotificationInfo .....	55
162	6.2.4.10 Umsetzung I Authorization Management Insurant::putNotificationInfo .....	57
163	6.2.4.11 Operationsdefinition	
164	I Authorization Management Insurant::getAuthorizationList .....	57
165	6.2.4.12 Umsetzung I Authorization Management Insurant::getAuthorizationList .....	59
166		
167	<b>6.3 Berechtigungstypen der Autorisierung .....</b>	<b>60</b>
168	<b>6.4 Hardware-Merkmal der Komponente Autorisierung .....</b>	<b>60</b>
169	<b>6.5 Geräteverwaltung .....</b>	<b>61</b>
170	6.5.1 Freischaltprozess neuer Geräte .....	61
171	6.5.2 Geräteadministration .....	64
172	<b>6.6 Freischaltprozess Vertretereinrichtung .....</b>	<b>65</b>
173	<b>7 Informationsmodell .....</b>	<b>68</b>
174	7.1 Namensräume .....	69
175	7.2 SAML-Profil und Tokeninhalte .....	69
176	<b>8 Verteilungssicht .....</b>	<b>73</b>
177	<b>9 Anhang A – Verzeichnisse .....</b>	<b>74</b>
178	9.1 Abkürzungen .....	74
179	9.2 Glossar .....	74
180	9.3 Abbildungsverzeichnis .....	74
181	9.4 Tabellenverzeichnis .....	75
182	9.5 Referenzierte Dokumente .....	76
183	9.5.1 Dokumente der gematik .....	76
184	9.5.2 Weitere Dokumente .....	77
185		

---

## 1 Einordnung des Dokumentes

---

### 1.1 Zielsetzung

Das vorliegende Dokument spezifiziert die Anforderungen an die Komponente "Autorisierung" des Produkttyps ePA-Aktensystem. Die Komponente Autorisierung ist verantwortlich für die zentrale Verwaltung des empfängerbezogenen verschlüsselten Schlüsselmaterials.

### 1.2 Zielgruppe

Das Dokument richtet sich an Hersteller und Anbieter der Komponente "Autorisierung" für die Nutzung in einem ePA-Aktensystem sowie an Hersteller und Anbieter von Produkttypen ePA, die Schnittstellen der Komponente "Autorisierung" nutzen.

### 1.3 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.*

### 1.4 Abgrenzungen

Spezifiziert werden in dem Dokument die von der Komponente bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert (siehe auch Anhang A5).

Die vollständige Anforderungslage für den Produkttyp ergibt sich aus weiteren Konzept- und Spezifikationsdokumenten. Diese sind in dem Produkttypsteckbrief des Produkttyps <ePA-Aktensystem> verzeichnet.

Nicht Bestandteil des vorliegenden Dokumentes sind die Festlegungen zum Themenbereich Betrieb.

## 221 1.5 Methodik

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

226 Da in dem Beispielsatz „Eine leere Liste DARF NICHT ein Element besitzen.“ die Phrase  
227 „DARF NICHT“ semantisch irreführend wäre (wenn nicht ein, dann vielleicht zwei?), wird  
228 in diesem Dokument stattdessen „Eine leere Liste DARF KEIN Element besitzen.“  
229 verwendet. Die Schlüsselworte werden außerdem um Pronomen in Großbuchstaben  
230 ergänzt, wenn dies den Sprachfluss verbessert oder die Semantik verdeutlicht.

231 Anforderungen werden im Dokument wie folgt dargestellt:

232 **<AFO-ID> - <Titel der Afo>**

233 Text / Beschreibung

234 [ $\leq$ ]

235 Dabei umfasst die Anforderung sämtliche zwischen Afo-ID und Textmarke [ $\leq$ ]  
236 angeführten Inhalte.



237

---

## 2 Systemüberblick

---

238 Der Autorisierungsdienst ePA ist eine Komponente des Produkttyps ePA-Aktensystem.  
239 Die Systemzerlegung der Fachanwendung ePA in Komponenten und Produkttypen sowie  
240 die Verteilung der Komponenten auf Produkttypen der Telematikinfrastruktur (TI) ist in  
241 [gemSysL\_ePA#2.1] und in [gemSysL\_ePA#4.1] definiert.

242 Die Komponente Autorisierungsdienst ePA verwaltet das empfängerverschlüsselte  
243 Schlüsselmaterial der Nutzer eines Aktenkontos eines Versicherten (kryptografische  
244 Autorisierung). Mit dem Vorhandensein einer kryptografischen Berechtigung ist ein  
245 Nutzer in der Lage, auf den symmetrischen Aktenschlüssel sowie den Kontextschlüssel  
246 zuzugreifen. Um dieses Schlüsselmaterial für den Zugriff auf medizinische Daten und  
247 Dokumente eines Versicherten zu nutzen, benötigt ein Nutzer ggfs. zusätzlich  
248 Berechtigungen auf Objektebene in anderen Komponente und Produkttypen, die die  
249 Daten und Dokumente des Versicherten verwalten.

---

## 3 Systemkontext

---

Der folgende Abschnitt setzt die Komponente Autorisierung in den Systemkontext der Fachanwendung ePA.

### 3.1 Akteure und Rollen

Die Komponente Autorisierung wird als Provider technischer Schnittstellen von weiteren technischen Komponenten und Produkttypen der Fachanwendung ePA aufgerufen. Diese weiteren Komponenten und Produkttypen nutzen die Schnittstellen der Komponente Autorisierung im Zusammenhang von fachlichen Anwendungsfällen der Nutzer der Fachanwendung ePA.

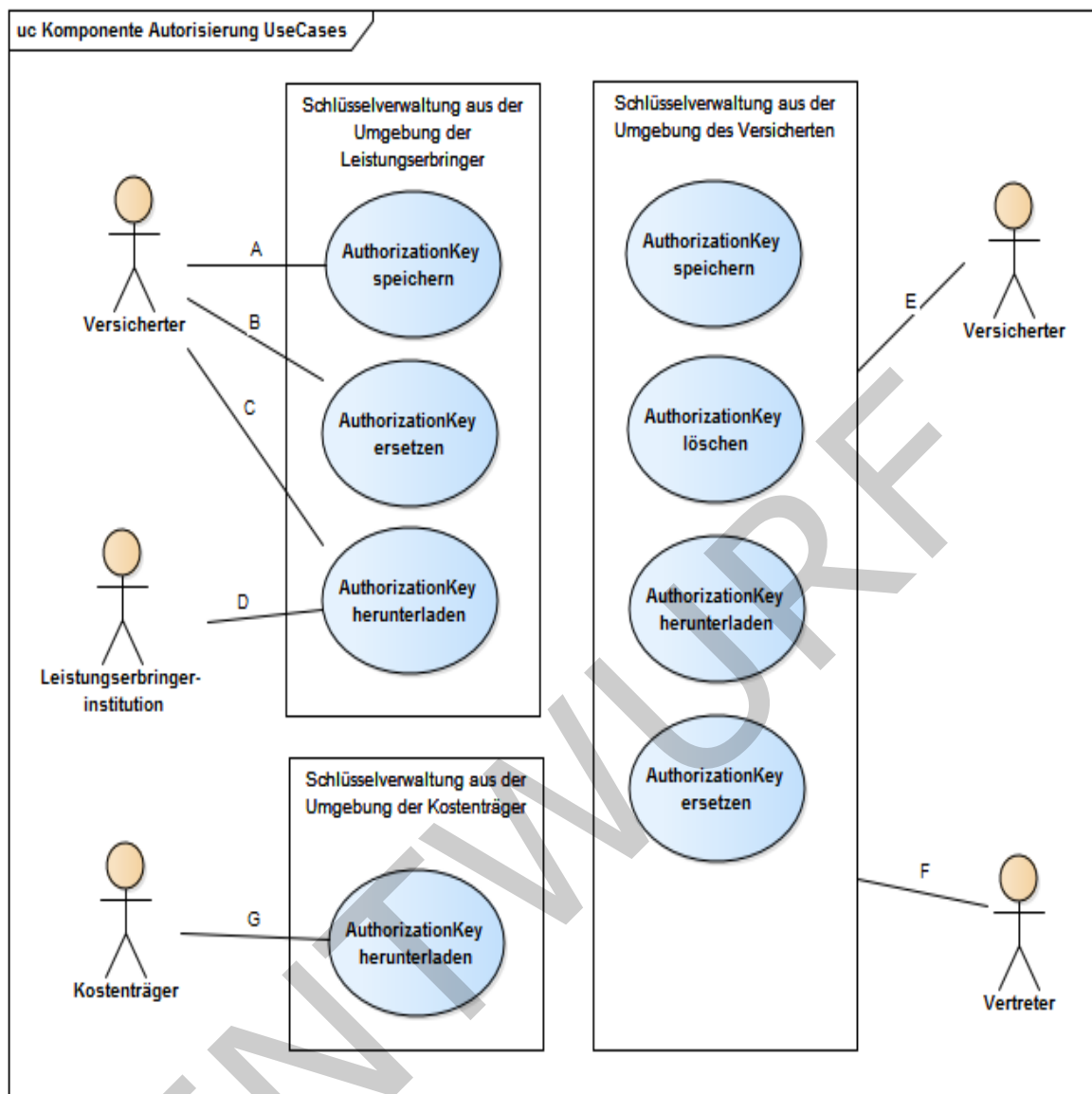
Die Nutzer sind dabei gesetzlich Versicherte, Leistungserbringerinstitutionen und Kostenträger, welche durch ihre jeweilige Karte der TI repräsentiert werden. Über eine kartenbasierte Authentifizierungsbestätigung authentisieren sie sich gegenüber der Komponente Autorisierung. Ein Spezialfall des gesetzlichen Versicherten ist der berechnigte Vertreter.

Für die oben genannten Nutzer verwaltet die Komponente Autorisierung empfängerbezogen verschlüsseltes Schlüsselmaterial

- für Versicherte, plus den Spezialfall des Vertreters - verschlüsselt für die individuelle KVN
- für Leistungserbringerinstitutionen und Kostenträger - verschlüsselt für die individuelle Telematik-ID

Die Komponente Autorisierung wird je nach Erfordernis zur Laufzeit von einem Administrator administriert. Gemäß der Festlegungen des Rollenmodells "Personenkreise der Telematikinfrastruktur" in [gemKPT\_Arch\_TIP] haben Anbieter, Betreiber und Administratoren keinen Zugriff auf medizinische Daten der Anwendungen des §291a SGB V [SGB V]. Die Komponente Autorisierung speichert personenbezogene Informationen, jedoch keine medizinischen Daten im Sinne des § 291a SGB V [SGB V].

Das folgende Bild gibt eine Übersicht der durch die Schnittstellen realisierten Anwendungsfälle zur Schlüsselverwaltung der Komponente Autorisierung. Zur Vereinfachung sind die Anwendungsfälle der Protokollierung und Geräteverwaltung nicht dargestellt.



**Abbildung 1: Anwendungsfälle der Schlüsselverwaltung nach Umgebung**

Die Berechtigung für Anwendungsfälle der Schlüsselverwaltung durch einen Nutzer unterscheidet sich nach Umgebung. Dem Versicherten stehen in der Umgebung der Leistungserbringer keine Anwendungsfälle zum Löschen bestehender Berechtigungen zur Verfügung, da ihm dort kein geeignetes Benutzerinterface zur Verfügung steht. Ein Ersetzen des Schlüsselmaterials erfolgt bei Vergabe einer Änderungsberechtigung für eine Leistungserbringerinstitution, wenn bspw. die Gültigkeitsdauer der Berechtigung angepasst wird.

Eine Leistungserbringerinstitution kann auf das für sie hinterlegte Schlüsselmaterial lesend zugreifen. Analog kann ein Kostenträger nur auf das für ihn hinterlegte Schlüsselmaterial lesend zugreifen.

In der Umgebung des Versicherten hat ein Versicherter vollen Zugriff auf das hinterlegte Schlüsselmaterial mit folgender Ausnahme - ein Versicherter darf das eigene Schlüsselmaterial für die eGK des Versicherten nicht löschen. Ein Vertreter führt Anwendungsfälle der Vertretung ausschließlich in der Umgebung eines Versicherten aus. Ebenso darf der Vertreter nicht das Schlüsselmaterial des Versicherten löschen und auch

nicht Schlüsselmaterial für andere eGK-Inhaber hinzufügen (kein Einrichten weiterer Vertretungen durch einen Vertreter).

Ergänzende Informationen zu Bezeichnern und Datentypen finden sich im Informationsmodell in Abschnitt 7.

**Tabelle 1: Anwendungsfälle der Schlüsselverwaltung nach Umgebung**

Assoziation	Actor	Regel zur Identifikation des Nutzers*
A	Versicherter	subject-id == OwnerKVNR == ActorID
B		
C		
D	Leistungserbringer-institution	subject-id == ActorID != OwnerKVNR (für HBA – erst in Folgestufe) organization-id == ActorID != OwnerKVNR (für SMC-B)
E	Versicherter	subject-id == OwnerKVNR
F	Vertreter	subject-id == ActorID != OwnerKVNR (beim Verwalten des Vertretungsschlüssels) subject-id != ActorID != OwnerKVNR (beim Verwalten aller übrigen Schlüssel)
G	Kostenträger	organization-id == ActorID != OwnerKVNR (für SMC-B KTR)

\* subject-id/organization-id ist Teil der Authentication- bzw. AuthorizationAssertion (als Behauptung gemäß [gemSpec\_TBAuth#TAB\_TBAuth\_02\_1/2]), OwnerKVNR ist ein Attribut der KeyChain (vgl. Kap. 7 Informationsmodell), der mehrere AuthorizationKeys untergeordnet werden, ActorID meint hier den Teil des AuthorizationKeys der dessen Besitzer identifiziert, (einige Schnittstellenoperationen verfügen über einen Parameter ActorID, dieser ist hier jedoch nicht Gegenstand der Betrachtung)

Der Versicherte wird beim Einsatz der eGK in der Umgebung der Leistungserbringer (Anwendungsfälle A und B) und in Anwendungsfällen aus der Umgebung des Versicherten (Anwendungsfälle zu E) anhand der KVNR als subject-id eines AuthenticationTokens erkannt. Diese stimmt gleichzeitig mit der OwnerKVNR des Eigentümers der Akte überein. Im Regelfall existiert für den Versicherten ein AuthorizationKey mit der KVNR des Versicherten als ActorID. Im Zustand der Kontoeröffnung und bei Anbieterwechsel wird das Schlüsselmaterial für den Versicherten extern erzeugt. Ein Nicht-Vorhandensein eines AuthorizationKeys für den Versicherten wird nicht als Fehler behandelt, sondern als Autorisierung im Zusammenhang mit Anwendungsfällen der Kontoverwaltung.

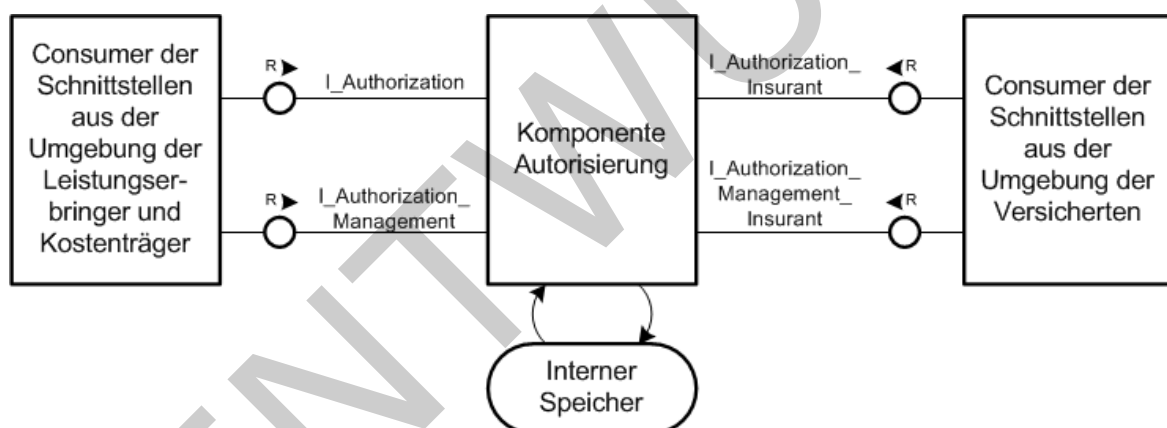
Eine Leistungserbringerinstitution wird bei Einsatz einer SMC-B (Anwendungsfälle C und D) anhand ihrer Telematik-ID aus der organization-id eines AuthenticationTokens erkannt. Für diese Telematik-ID muss ein AuthorizationKey mit gleichlautender ActorID vorhanden sein, andernfalls ist diese Leistungserbringerinstitution nicht autorisiert. Das gleiche gilt für die Kostenträger (Anwendungsfälle G und H).

Der Vertreter wird zunächst als Versicherter mit eigener eGK anhand der KVNR als subject-id eines AuthenticationTokens erkannt. In der Wahrnehmung einer Vertretung (Anwendungsfälle F) ist seine KVNR ungleich der OwnerKVNR des Eigentümers der Akte. Für seine KVNR muss ein AuthorizationKey mit gleichlautender ActorID vorhanden sein, andernfalls ist der Vertreter für den Zugriff nicht autorisiert.

## 3.2 Nachbarsysteme

Der folgende Abschnitt beschreibt die Positionierung der Komponente Autorisierung im Kontext der Fachanwendung ePA.

Die folgende Abbildung zeigt die Beziehung zu benachbarten Produkttypen innerhalb der Fachanwendung mit den von der Komponente Autorisierung bereitgestellten Schnittstellen.



**Abbildung 2: Komponente Autorisierung, benachbarte Komponenten und Produkttypen**

Die Komponente Autorisierung stellt die Schnittstellen `I_Authorization` und `I_Authorization_Management` zur Nutzung aus der Umgebung der Leistungserbringer und Kostenträger bereit. Von dort werden sie aus der Secure Consumer Zone aufgerufen.

Die Schnittstellen `I_Authorization_Insurant` und `I_Authorization_Management_Insurant` werden aus der Personal Zone in der Umgebung des Versicherten aufgerufen. In dieser Umgebung nutzt der Versicherte das ePA-Frontend des Versicherten auf einem Gerät des Versicherten.

Die Komponente Autorisierung wird als Teil des Produkttyps ePA-Aktensystem in der Provider Zone der Telematikinfrastruktur betrieben. Sie verfügt über einen logischen, internen Speicher, an den in diesem Dokument keine Umsetzungsanforderungen gestellt werden. Er dient der Persistierung der im Informationsmodell (siehe [7.2 Informationsmodell](#)) strukturierten Inhalte.

**A\_13956 - Komponente Autorisierung -Separierung der Schnittstellen für verschiedene Umgebungen**

Die Komponente Autorisierung MUSS die Bereitstellungspunkte der Schnittstellen für die Nutzung durch benachbarte Komponenten und Produkttypen aus verschiedenen Einsatzumgebungen voneinander separieren. [ $\leq$ ]

Diese Separierung kann beispielsweise umgesetzt werden durch die Erreichbarkeit der Schnittstellen über verschiedene Netzwerkadressen.

**3.3 Tokenbasierte Autorisierung**

Die Komponente Autorisierung bietet eine Single-Sign-On (SSO)-Lösung an, um einem zuvor authentifizierten Nutzer den Zugriff auf weitere Ressourcen zu ermöglichen. Hierbei wird nach einer erfolgreichen Autorisierung eine Autorisierungsbestätigung (AuthorizationAssertion gemäß SAML 2.0 Assertions [SAML2.0]) ausgestellt.

Für die Initialisierung sowie für den Zugriff auf den Aktenkontext eines Versicherten erwartet die Komponente Dokumentenverwaltung eine gültige Assertion von der Komponente Autorisierung. Die Assertion wird ungültig, wenn der Aktenkontext eines Versicherten geschlossen wird oder der Gültigkeitszeitraum der Assertion abgelaufen ist.

368

---

## 4 Zerlegung der Komponente Autorisierung

---

369 Eine detaillierte Zerlegung der Komponente Autorisierung wird nicht vorgegeben.  
370 Gleichwohl muss die Komponente Autorisierung privates Schlüsselmaterial in einem HSM  
371 speichern, das den Anforderungen einer bestimmten Prüftiefe entspricht. Auf eine  
372 grafische Darstellung wird an dieser Stelle verzichtet.

ENTWURF

373

## 5 Übergreifende Festlegungen

### 5.1 Datenschutz und Datensicherheit

Im folgenden Abschnitt werden die für die Komponente Autorisierung notwendigen Anforderungen für den Schutz personenbezogener Daten bzw. Anforderungen für den Schutz von Daten beschrieben, um beispielsweise vor Datenmanipulation oder Datenverlust zu schützen.

#### **A\_14417 - Komponente Autorisierung - Akzeptieren von Identitätsbestätigungen**

Die Komponente Autorisierung MUSS Identitätsbestätigungen (AuthenticationAssertion) als ungültig mit dem Fehler ASSERTION\_INVALID ablehnen, wenn die Identität des Ausstellers (Issuer) nicht als vertrauenswürdiger Dienst für die Durchführung einer Authentifizierung konfiguriert ist oder dessen X.509-Signatur-Zertifikat nicht zu der Signatur der Identitätsbestätigung passt.

[<=]

#### **A\_13990 - Komponente Autorisierung - Vorgaben für Identitätsbestätigung**

Die Komponente Autorisierung MUSS eine übergebene Identitätsbestätigung (AuthenticationAssertion) als ungültig mit dem Fehler ASSERTION\_INVALID ablehnen, wenn diese nicht konform zu den Vorgaben der Tabelle

[gemSpec\_TBAuth#TAB\_TBAuth\_03 Identitätsbestätigung] ist. [<=]

#### **A\_14688-01 - Komponente Autorisierung - Prüfung einer Identitätsbestätigung**

Die Komponente Autorisierung MUSS eine übergebene Identitätsbestätigung (AuthenticationAssertion) als ungültig mit dem Fehler ASSERTION\_INVALID ablehnen, die nach einer Prüfung gemäß [gemSpec\_TBAuth#A\_15557] (vgl. auch gemSpec\_TBAuth#3.2 Prüfen von Identitätsbestätigungen) als nicht gültig betrachtet wird. Insbesondere MUSS die Komponente Autorisierung das Signaturzertifikat der Ausstelleridentität eines Vertrauensraums außerhalb des Vertrauensraums der Komponente Autorisierung mittels [gemSpec\_PKI#TUC\_PKI\_018] mit den folgenden Parametern prüfen:

Parameter	Belegung für SAML 2.0 Assertions des Fachmoduls ePA	
Zertifikat	Signaturzertifikat (eingebettet in Identitätsbestätigung) C.HCI.OSIG	
PolicyList	oid_smc_b_osig	
intendedKeyUsage	nonRepudiation	
intendedExtendedKeyUsage	(leer)	
OCSP-Graceperiod	60 Minuten	
Offline-Modus	nein	



Prüfmodus	OCSF	
-----------	------	--

Das Signaturzertifikat muss anhand der Zertifikatsprüfung für [mathematisch gültig UND zeitlich gültig UND online gültig ] befunden werden. Die Telematik-ID im Signaturzertifikat muss identisch mit der Telematik-ID in der Identitätsbestätigung sein. [ <= ]

#### **A\_18989 - Komponente Autorisierung – Beschränkung gültiger Identitätsbestätigungen**

Die Komponente Autorisierung DARF in Aufrufen aus Richtung der Komponente Zugangsgateway KEINE Identitätsbestätigung akzeptieren, die nicht durch die Komponente Authentisierung (Versicherter) erstellt wurde. [ <= ]

#### **A\_17839-01A\_17839 - Komponente Autorisierung - Prüfung der Empfänger-Rolle**

Die Komponente Autorisierung MUSS beim Aufruf einer der Operation

- I\_Authorization::getAuthorizationKey

den übergebenen Parameter `AuthenticationAssertion` dahingehend prüfen, ob mindestens eine `ProfessionOID` der ZertifikatsExtension `Admission` gemäß [gemSpec\_PKI#Tab\_PKI\_226] im Signaturzertifikat C.HCI.OSIG `/saml2:Assertion/ds:Signature/ds:KeyInfo/ds:X509Data/ds:X509Certificate` in der Liste der zulässigen Autorisierungsempfänger-Rollen gemäß [gemSpec\_OID#Tab\_PKI\_402] und [gemSpec\_OID#Tab\_PKI\_403]

- `oid_praxis_arzt`
- `oid_zahnarztpraxis`
- `oid_praxis_psychotherapeut`
- `oid_krankenhaus`
- `oid_oeffentliche_apotheke`
- `oid_krankenhausapotheke`
- `oid_bundeswehraphotheke`
- `oid_mobile_einrichtung_rettungsdienst`
- `oid_kostentraeger`
- `oid_institution_pflege`
- `oid_geburtshilfe`
- `oid_praxis_physiotherapeut`
- `oid_gesundheitsdienst`
- `oid_arbeitsmedizin`
- `oid_vorsorge_reha`
- `oid_sanitaetsdienst_bundeswehr`

enthalten ist und sofern nicht, die Operation mit dem Fehler `AUTHORIZATION_ERROR` abbrechen.

[ <= ]

442 Ist die AuthenticationAssertion vom Aktensystem selbst erstellt worden  
 443 (/saml2:Assertion/ds:Signature/ds:KeyInfo/ds:X509Data/ds:X509Certificate  
 444 enthält das Signaturzertifikat C.FD.SIG des Aktensystems), entfällt die Rollenprüfung, da  
 445 die Rolle des Versicherten bereits durch Komponente Authentisierung Versicherter  
 446 geprüft wurde.

#### 447 **A\_17840 - Komponente Autorisierung Vers. - Prüfung Identitätswechsel des** 448 **Versicherten**

449 Die Komponente Autorisierung MUSS eine übergebene AuthenticationAssertion für  
 450 einen Versicherten (Das  
 451 SAML:Assertion/SAML:AttributeStatement/SAML:Attribute urn:gematik:subject:subject-  
 452 id enthält eine KVNR) dahingehend prüfen, ob die in der  
 453 Behauptung urn:gematik:subject:authreference mit der serialNumber des zur  
 454 Authentifizierung verwendeten AUT- bzw. AUT\_ALT-Zertifikats in der Liste der bekannten  
 455 AUT-Referenzen an der KeyChain des im RecordIdentifier benannten Aktenkontos ist und  
 456 falls nicht,  
 457 MUSS die Komponente Autorisierung den Versicherten sowie im Vertretungsfall zusätzlich  
 458 den Vertreter über die Nutzung eines neuen Authentisierungsmittels in einer E-Mail-  
 459 Nachricht an die hinterlegte E-Mailadresse NotificationInfo des Versicherten bzw. des  
 460 Vertreters informieren. Anschließend MUSS die benannte serialNumber in die WhiteList  
 461 der AUT-Referenzen an der KeyChain des im RecordIdentifier benannten  
 462 Aktenkontos übernommen werden.

463  
 464 [ $\leq$ ]

465 Nutzt der Versicherte ein im Aktensystem bisher unbekanntes Authentisierungsmittel  
 466 (z.B. eine Folge-eGK) erhält er eine E-Mailbenachrichtigung, der Anwendungsfall wird  
 467 nicht unterbrochen. Es obliegt dem Versicherten die Legitimität des Zugriffs bzw. des  
 468 Authentisierungsmittels zu prüfen und sich gegebenenfalls mit dem ePA-Aktenanbieter  
 469 und seiner Kasse in Verbindung zu setzen.

470 Nutzt der Vertreter des Versicherten ein bisher unbekanntes Authentisierungsmittel,  
 471 erhalten sowohl der Versicherte als auch der Vertreter eine Benachrichtigung.

#### 472 **A\_17655 - Komponente Autorisierung – Prüfung von Identitätsbestätigungen** 473 **des Aktensystems**

474 Die Komponente Autorisierung MUSS sicherstellen, dass Identitätsbestätigungen für  
 475 Versicherte nur akzeptiert werden, wenn das zugehörige Signaturzertifikat zeitlich gültig  
 476 ist, nicht gesperrt wurde und nach dem Zertifikatsprofil C.FD.SIG auf die Identität der  
 477 Komponente Authentisierung Versicherter ausgestellt wurde.

478 [ $\leq$ ]

479 Dies kann durch eine aktuell gehaltene Konfiguration vertrauenswürdiger Zertifikate  
 480 umgesetzt werden und ersetzt eine detaillierte Prüfung des Signaturzertifikats gemäß  
 481 [gemSpec\_TBAuth#A\_15557], um die Prüfung solcher vom ePA-Aktensystem selbst  
 482 ausgestellten Identitätsbestätigungen zu vereinfachen.

483 Eine Prüfung von Identitätsbestätigungen gemäß den Festlegungen für TBAuth bezieht  
 484 sich auf Identitätsbestätigungen für Leistungserbringerinstitutionen und Kostenträger. . .

#### 485 **A\_14270 - Komponente Autorisierung - Zugriff aus der Umgebung des** 486 **Versicherten**

487 Die Komponente Autorisierung MUSS Zugriffe auf Daten eines Versicherten aus der  
 488 Personal Zone heraus verhindern, wenn das verwendete Gerät des Versicherten nicht in  
 489 der Liste der bekannten/freigeschalteten Geräte vorhanden ist. [ $\leq$ ]

490 Bei Zugriffen aus der Umgebung des Versicherten wird ein Identitätsmerkmal des  
 491 verwendeten Geräts abgefragt (DeviceID). Bei Zugriffen aus der Umgebung der

Leistungserbringer erfolgt dies nicht, da hier als zugreifende Geräte ausschließlich zugelassene Konnektoren mit geprüfter Fachlogik zum Einsatz kommen. Ebenso wird keine Geräteidentität für den Zugang der Kostenträger über ihr jeweiliges Rechenzentrum geprüft, da auch hier ausschließlich zugelassene Produkttypen in einer kontrollierten Betriebsumgebung zum Einsatz kommen.

#### **A\_14402 - Komponente Autorisierung - Integritätsschutz für Autorisierungsbestätigungen**

Die Komponente Autorisierung MUSS jede ausgestellte Autorisierungsbestätigung mit dem privaten Schlüssel der Ausstelleridentität C.FD.SIG in seiner fachlichen Rolle oid\_epa\_authz gemäß [gemSpec\_OID] signieren.[<=]

#### **A\_14740 - Komponente Autorisierung - TLS-Identität innerhalb der TI**

Die Komponente Autorisierung MUSS sich beim TLS-Verbindungsaufbau an den Schnittstellen innerhalb der TI mit der technischen Rolle oid\_epa\_authz der TLS-Identität C.FD.TLS-S authentisieren.[<=]

#### **A\_14529 - Komponente Autorisierung - Absicherung gegenüber dem Internet**

Die Komponente Autorisierung MUSS alle Operationsaufrufe der Schnittstellen I\_Authorization\_Insurant und I\_Authorization\_Management\_Insurant auf Wohlgeformtheit und Zulässigkeit gemäß Protokoll SOAP 1.2 prüfen und bei Schema-, Semantik- oder Protokollverletzungen eine aufgerufene Operation mit dem HTTP-Statuscode 400 gemäß [RFC-7231] abbrechen.[<=]

Die Prüfung der eingehenden Nachrichten auf Syntax-, Semantik- und Protokollverletzungen soll insbesondere den Angriffstypen *XML Injection*, *XPath Query Tampering* und *XML External Entity Injection* entgegenwirken.

Im Fall der Sperrung der Signaturidentität der Komponente Autorisierung, darf diese nicht für die Ausstellung einer Autorisierungsbestätigung genutzt werden. Da diese Identität aus dem gleichen Vertrauensraum stammt wie die Signaturidentität der Identitätsbestätigung eines Authentisierungsdienstes im gleichen Aktensystem, dürfen in diesem Fall auch keine Identitätsbestätigungen des gleichen Vertrauensraums mehr akzeptiert werden.

#### **A\_16260 - Komponente Autorisierung - Periodische Prüfung Signaturidentität**

Die Komponente Autorisierung MUSS den Sperrstatus der eigenen Signaturidentität C.FD.SIG mittels [gemSpec\_PKI#TUC\_PKI\_018] periodisch (einmal täglich) prüfen:

Parameter	Belegung
Zertifikat	Signaturzertifikat C.FD.SIG der Komponente Autorisierung
PolicyList	oid_fd_sig
intendedKeyUsage	digitalSignature
intendedExtendedKeyUsage	(leer)
OCSP-Graceperiod	60 Minuten
Offline-Modus	nein
Prüfmodus	OCSP

Das Signaturzertifikat muss anhand der Zertifikatsprüfung für [mathematisch gültig UND zeitlich gültig UND online gültig] befunden werden. [≤]

**A\_16261 - Komponente Autorisierung - Keine Autorisierung bei gesperrter Signaturidentität**

Die Komponente Autorisierung MUSS das Ausstellen einer Autorisierungsbestätigung mit dem Fehler INTERNAL\_ERROR abbrechen, wenn das Signaturzertifikat der Komponente Autorisierung gemäß einer Statusprüfung nach [A\_16260] nicht gültig ist. [≤]

**A\_16262 - Komponente Autorisierung - Keine Identitätsbestätigung bei gesperrter Signaturidentität**

Die Komponente Autorisierung MUSS alle Identitätsbestätigungen aller Issuer des gleichen Vertrauensraums der Signaturidentität C.FD.SIG der Komponente Autorisierung mit dem Fehler INTERNAL\_ERROR als ungültig ablehnen, wenn das Signaturzertifikat der Komponente Autorisierung gemäß einer Statusprüfung nach [A\_16260] nicht gültig ist. [≤]

## 5.2 Verwendete Standards

Für die Sicherstellung der Interoperabilität wird auf verwendete Standards zurückgegriffen.

Durch die Verwendung des IHE-Frameworks (Integrating the Healthcare Enterprise) zum einheitlichen Datenaustausch im Gesundheitssystem ist die Verwendung von SAML zum Austausch von Authentisierungsinformationen notwendig.

Für die Übertragung von Nachrichten zwischen dem Fachmodul und den Teilkomponenten von ePA wird das vom W3C standardisierte Protokoll SOAP 1.2 in Verbindung mit HTTP verwendet.

**A\_13801 - Komponente Autorisierung - Verwendung von SAML 2.0**

Die Komponente Autorisierung MUSS Authentisierungsbestätigung im Format SAML 2.0 Assertions [SAML2.0] unterstützen.

[≤]

**A\_13802 - Komponente Autorisierung - Ausstellung im Format SAML 2.0**

Die Komponente Autorisierung MUSS Autorisierungsbestätigungen im Format SAML 2.0 Assertions [SAML2.0] ausstellen. [≤]

**A\_14969 - Komponente Autorisierung - Kodierung in UTF-8**

Die Komponente Autorisierung MUSS bei der Erstellung von XML-Fragmenten das Encoding UTF-8 verwenden.

[≤]

**A\_17760 - Komponente Autorisierung - AuthenticationAssertion im SOAP-Header**

Die Komponente Autorisierung MUSS die Identitätsbestätigungen eines Nutzers (AuthenticationAssertion) im Header eines eingehenden SOAP-Requests akzeptieren.

[≤]

**A\_17761 - Komponente Autorisierung - Verwendung des SAML Token Profile 1.1 für Web Services Security bei SAML 2.0 Assertions**

Die Komponente Autorisierung MUSS die Anforderungen aus [WSS-SAML] umsetzen, wenn eine SAML 2.0 Assertion Teil einer SOAP 1.2-Eingangsnachricht ist.

[≤]

## **A\_17762 - Komponente Autorisierung - Verwendung von SOAP Message Security 1.1**

Die Komponente Autorisierung MUSS die Sicherheitsanforderungen aus SOAP Message Security 1.1 [WSS] für die Verarbeitung von SOAP 1.2-Nachrichten umsetzen. [ $\leq$ ]

## **A\_17763 - Komponente Autorisierung - Unterstützung von Profilen der Web Services Interoperability Organization (WS-I)**

Die Komponente Autorisierung MUSS das WS-I Basic Profile V2.0 [WSIBP], das WS-I Basic Security Profile Version V1.1 [WSIBSP] sowie das WS-I Attachment Profile V1.0 [WSIAP] für die Kommunikation über Web Services berücksichtigen. [ $\leq$ ]

## **5.3 Protokollierung**

Die Anforderungen an die Protokollierung für die Komponente Autorisierung leiten sich aus dem Konzept der Protokollierung aus [gemSysL\_ePA#2.5.5] ab.

## **A\_14403 - Komponente Autorisierung - Verwaltungsprotokollierung Autorisierung**

Die Komponente Autorisierung MUSS beim Aufruf einer der folgenden Operationen:

- I\_Authorization\_Insurant::getAuthorizationKey
- I\_Authorization\_Management::putAuthorizationKey
- I\_Authorization\_ManagementI\_Authorization\_Management\_Insurant::putAuthorizationKey
- I\_Authorization\_Management\_Insurant::deleteAuthorizationKey
- I\_Authorization\_Management\_Insurant::replaceAuthorizationKey
- I\_Authorization\_Management\_Insurant::getAuditEvents
- I\_Authorization\_Management\_Insurant::putNotificationInfo
- I\_Authorization\_Management\_Insurant::getAuthorizationList

je einen Eintrag im Verwaltungsprotokoll für den Versicherten gemäß [\[gemSpec\\_DM\\_ePA#A\\_14471\]](#) mit folgenden vom Operationsaufruf abhängigen Parameterwerten vornehmen: UserID, Username, ObjectID, ObjectName, DeviceID.

[ $\leq$ ]

Der Aufruf der Operation I\_Authorization::getAuthorizationKey aus der Umgebung der Leistungserbringer und der Kostenträger wird nicht protokolliert.

## **[A\\_15753-01A\\_15753](#) - Komponente Autorisierung - Verwaltungsprotokollierung E-Mail-Adresse ändern**

Die Komponente Autorisierung MUSS das manuelle Ändern der Benachrichtigungsadresse (z.B. durch den Anbieter im Supportfall) im Verwaltungsprotokoll des Versicherten [mit PHR-451](#) protokollieren. [ $\leq$ ]

## **A\_14427-01 - Komponente Autorisierung - Verwaltungsprotokollierung Gerät hinzufügen**

Die Komponente Autorisierung MUSS beim Hinzufügen eines Geräts in die Liste der registrierten Geräte einen Eintrag im Verwaltungsprotokoll für den Versicherten mit PHR-470 vornehmen. [ $\leq$ ]

**A\_14188-01 - Komponente Autorisierung - Umfang Verwaltungsprotokoll**

Die Komponente Autorisierung MUSS dem Versicherten oder berechtigten Vertreter die Einträge des Verwaltungsprotokolls gemäß der Festlegung in [\[gemSpec\\_DM\\_ePA#A\\_14471\]](#) übergeben:

**Tabelle 2: Parameter des Verwaltungsprotokolls**

Protokollparameter	Parameterwerte gemäß aufgerufener Operation
UserID	Wert des AttributeStatements der übergebenen übergebenen AuthenticationAssertion in SAML:Assertion/SAML:AttributeStatement/SAML:Attribute/@Name (unveränderbare Anteil der KVNR des aufrufenden Versicherten bzw. Vertreters)
UserName	Wert aus SAML:Assertion/SAML:Subject/SAML:NameID der im Operationsaufruf übergebenen AuthenticationAssertion
ObjectID	ActorID des bearbeiteten AuthorizationKey (KVNR für Vertreter bzw. TelematikID für berechnigte Leistungserbringerorganisation) <i>Hinweis: Bei Aufruf von Operationen ohne diesen Parameter wird der Wert im Protokolleintrag nicht belegt.</i>
ObjectName	ActorID des im Operationsaufruf gelesenen, gespeicherten oder geänderten AuthorizationKey <i>Hinweis: Bei Aufruf von Operationen ohne Bezug zu einem AuthorizationKey wird der Wert im Protokolleintrag nicht belegt (z.B. getAuditEvents).</i>
DeviceID	DeviceID-Parameter DeviceIdType::Displayname des Operationsaufrufs <i>Hinweis: Bei Aufruf der Operationen der Schnittstelle I_Authorization_Management gibt es den Parameter nicht, DeviceID wird im Protokolleintrag demzufolge nicht belegt.</i>

[&lt;=]

**A\_14189 - Komponente Autorisierung - Protokollierung Schutz vor Manipulation**

Die Komponente Autorisierung MUSS sicherstellen, dass die Verwaltungsprotokolldaten gegen Veränderung und unberechtigtes Löschen geschützt sind.

[&lt;=]



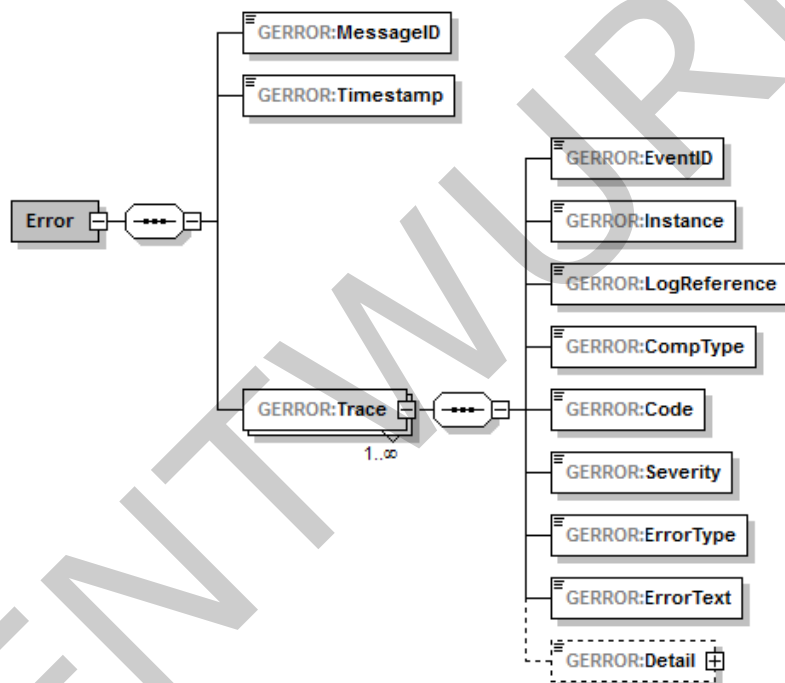
## 5.4 Fehlerbehandlung in Schnittstellenoperationen

Bei Fehlern in der internen Verarbeitung oder fachlichen Fehlern in der Nutzung der von der Komponente Autorisierung bereitgestellten Schnittstellen werden Operationsaufrufe mit gematik-Fehlermeldungen gemäß der Definition in [gemSpec\_OM] beantwortet. Die Fehlermeldungen werden als SOAP-Fault gemäß [TelematikError.xsd] strukturiert. Abweichend von den Festlegungen in [gemSpec\_OM] sind zu meldende Fehler wie folgt mit Informationen zu füllen.

### A\_15068 - Komponente Autorisierung - Fehlername

Die Komponente Autorisierung MUSS in einer GERROR-Fehlermeldung gemäß [TelematikError.xsd] den in der Operationsdefinition festgelegten Fehlernamen Name im Feld `tel:Error/tel:Trace/tel:EventID` verwenden. [ $\leq$ ]

Die folgende Abbildung illustriert das Schema der GERROR-Struktur in TelematikError.xsd:



**Abbildung 3: GERROR-Struktur zur Rückgabe einer Fehlermeldung**

### A\_15069 - Komponente Autorisierung - Fehlertext

Die Komponente Autorisierung MUSS in einer GERROR-Fehlermeldung gemäß [TelematikError.xsd] den in der Operationsdefinition festgelegten Fehlerdetailtext Fehlertext im Feld `tel:Error/tel:Trace/tel:ErrorText` verwenden. [ $\leq$ ]

### A\_15101 - Komponente Autorisierung - Fehlernummer

Die Komponente Autorisierung MUSS in einer GERROR-Fehlermeldung gemäß [TelematikError.xsd] die folgenden Fehlercodes im Feld `tel:Error/tel:Trace/tel:Code` verwenden:

**Tabelle 3: Fehlercodes zu Fehlern gemäß Operationsdefinition**

Name	Fehlercode
TECHNICAL_ERROR	7900

KEY_ERROR	7910
SYNTAX_ERROR	7930
ASSERTION_INVALID	7940
DEVICE_UNKNOWN	7950
ACCESS_DENIED	7960
AUTHORIZATION_ERROR	7970
REPRESENTATIVE_PENDING	7980

652  
653  
654 **[<=]**

655 Die Operationsdefinitionen der Schnittstellen der Komponente Autorisierung beschränken  
656 die Liste möglicher Fehler auf fachliche Fehler. Daneben sind weitere, technische Gründe  
657 für Fehler anderer Art denkbar. Für diese kann der Hersteller der Komponente einen  
658 generischen Fehler für den Transport geeigneter Fehlerinformationen (z.B. für  
659 Supportzwecke) verwenden.

#### 660 **A\_15102 - Komponente Autorisierung - Herstellerspezifische Fehlermeldungen**

661 Die Komponente Autorisierung MUSS komponenteninterne und herstellerspezifische  
662 Fehlermeldungen in einer GERROR-Fehlermeldung gemäß [TelematikError.xsd] mit  
663 folgender Festlegung transportieren:

664 **Tabelle 4: Herstellerspezifische Fehlerdefinition**

GERROR-Element	Herstellerspezifisch zu belegen
tel:Error/tel:Trace/tel:Code	Fester Wert: "7900"
tel:Error/tel:Trace/tel:EventID	Fester Wert: "TECHNICAL_ERROR"
tel:Error/tel:Trace/tel:ErrorText	Je Fehlerfall zufällig gewählte Fehlernummer

665  
666 **[<=]**

#### 667 **A\_15249 - Komponente Autorisierung - Herstellerspezifische Fehlermeldungen** 668 **Detailtext**

669 Die Komponente Autorisierung MUSS Details zu herstellerspezifischen Fehlermeldungen  
670 ausschließlich in einem internen Fehlerprotokoll und zusammen mit der zum Zeitpunkt  
671 des Fehlers gewählten zufälligen Fehlernummer speichern. **[<=]**

672 Die herstellerspezifische und je Fehlerfall zufällig gewählte Fehlernummer dient der  
673 Kapselung von Implementierungs- und Fehlerbehebungsdetails und zum Auffinden der  
674 Fehlermeldungsdetails in einem internen Fehlerprotokoll im Supportfall.



675 **5.5 Nicht-Funktionale Anforderungen**

676 **5.5.1 Skalierbarkeit**

677 Die für die Komponente Autorisierung relevanten Informationen zur Skalierbarkeit sind in  
678 [gemSpec\_Perf] zu entnehmen.

679 **5.5.2 Performance**

680 Die durch die Komponente Autorisierung zu erfüllende Performance-Anforderung  
681 befinden sich in [gemSpec\_Perf].

682 **5.5.3 Mengengerüst**

683 Das für die Komponente Autorisierung relevante Mengengerüst befindet sich in  
684 [gemSpec\_Perf].

685

## 6 Funktionsmerkmale

686 Die Komponente Autorisierung realisiert die Funktionsmerkmale der kryptografischen  
687 Autorisierung und eine Geräteverwaltung. Das Funktionsmerkmal der Autorisierung wird  
688 über die Implementierung der  
689 Schnittstellen `I_Authorization`, `I_Authorization_Management`, `I_Authorization_Insu`  
690 `rant` und `I_Authorization_Management_Insurant` realisiert.

691 Die Nutzung des Funktionsmerkmals der Geräteverwaltung durch den Versicherten  
692 erfolgt über einen separaten Verwaltungszugang abseits der `I_Authorization*`-  
693 Schnittstellen. Dieser Zugang ist für den Versicherten über das Internet erreichbar.

### 6.1 Übergreifende Festlegungen

695 Im Folgenden werden übergreifende Festlegungen formuliert, die in allen Operationen  
696 umgesetzt werden.

697 Wenn im Folgenden die KVNR als ActorID, OwnerKVNR oder subject-id referenziert wird  
698 ist immer der unveränderliche Anteil als 10-stellige Kennung gemeint.

#### **A\_14469 - Komponente Autorisierung - Identifizierung des Versicherten anhand einer AuthenticationAssertion**

701 Die Komponente Autorisierung MUSS jeden Versicherten anhand des unveränderlichen  
702 Teils der KVNR als `urn:gematik:subject:subject-id` in  
703 `SAML:Assertion/SAML:AttributeStatement/SAML:Attribute/@Name` einer  
704 übergebenen, gültigen AuthenticationAssertion eindeutig identifizieren, wenn  
705 die subject-id mit der OwnerKVNR zu einem im Operationsaufruf angegebenen  
706 RecordIdentifier übereinstimmt.

707  
708 [ $\leq$ ]

#### **A\_14499 - Komponente Autorisierung - Identifizierung einer Institution anhand einer AuthenticationAssertion**

711 Die Komponente Autorisierung MUSS jede Leistungserbringerinstitution und jeden  
712 Kostenträger anhand der Telematik-ID als `urn:gematik:subject:organization-id` in  
713 `SAML:Assertion/SAML:AttributeStatement/SAML:Attribute/@Name` einer  
714 übergebenen, gültigen AuthenticationAssertion eindeutig identifizieren, wenn für diese  
715 ein AuthorizationKey zu einem im Operationsaufruf angegebenen RecordIdentifier  
716 existiert.

717  
718 [ $\leq$ ]

#### **A\_14500 - Komponente Autorisierung - Identifizierung eines Vertreters anhand einer AuthenticationAssertion**

721 Die Komponente Autorisierung MUSS einen berechtigten Vertreter anhand seiner KVNR  
722 als `urn:gematik:subject:subject-id` in  
723 `SAML:Assertion/SAML:AttributeStatement/SAML:Attribute/@Name` einer  
724 übergebenen, gültigen AuthenticationAssertion eindeutig identifizieren, wenn  
725 die subject-id ungleich der OwnerKVNR zu einem im Operationsaufruf angegebenen  
726 RecordIdentifier ist und für die KVNR der AuthenticationAssertion ein AuthorizationKey zu  
727 der im Operationsaufruf angegebenen RecordIdentifier existiert.

[<=]

#### **A\_14434 - Komponente Autorisierung - Prüfung der Schnittstellenparameter**

Die Komponente Autorisierung MUSS in jeder Operation alle übergebenen Eingangsparameter auf Konformität zum Schema AuthorizationService.xsd prüfen und bei Nichtkonformität die jeweilige Operation mit dem Fehler TECHNICAL\_ERROR gemäß den Festlegungen zur [Fehlerbehandlung](#) abbrechen.

[<=]

#### **A\_14369 - Komponente Autorisierung - Prüfung des Geräts des Versicherten**

Die Komponente Autorisierung MUSS in allen Operationen der Schnittstellen `I_Authorization_Insurant` und `I_Authorization_Management_Insurant` anhand des Wertes `DeviceID::Device` prüfen, ob das vom Nutzer verwendete Gerät in der Geräteliste des `AuthorizationKeys` des Nutzers bekannt/freigeschaltet ist und andernfalls die Operation mit dem Fehler `DEVICE_UNKNOWN` abbrechen, in dessen SOAP-Error in `tel:Error/tel:Trace/tel:ErrorText` eine gemäß [\[gemSpec Autorisierung#A\\_17866\]](#) generierte `phr:DeviceID::Device` einfügen und den Freischaltprozess neuer Geräte auslösen.

[<=]

Greift ein Nutzer mit einem Gerät erstmalig auf die in A\_14369 genannten Schnittstellen zu, sind die Elemente `phr:DeviceID@` und `phr:DeviceID::Device` in den aufgerufenen Operationen ggfs. leer bzw. enthalten eine Zeichenkette der Länge 0 ("").

#### **A\_14634 - Komponente Autorisierung - Prüfung auf vorhandenen AuthorizationKey**

Die Komponente Autorisierung MUSS eine aufgerufene Operationen mit dem Standardfehler `KEY_ERROR` abbrechen, wenn es zu fachlichen Fehlern in Lese- oder Schreiboperationen eines `AuthorizationKey` kommt oder dieser für einen in der `ActorID` benannten Nutzer in der `KeyChain` eines benannten `RecordIdentifier` nicht vorhanden ist. [<=]

#### **A\_14768 - Komponente Autorisierung - Prüfung auf Berechtigung**

Die Komponente Autorisierung MUSS eine aufgerufene Operation mit dem Standardfehler `ACCESS_DENIED` abbrechen, wenn ein über die `subject-id` bzw. `organization-id` einer `AuthenticationAssertion` identifizierter Nutzer eine Operation auf einem im `RecordIdentifier` benannten Datensatz aufruft, für den kein `AuthorizationKey` hinterlegt und er nicht der Eigentümer ist, d.h. `OwnerKVNDR != subject-id` bzw. `organization-id` und es existiert kein `AuthorizationKey` mit `ActorID == subject-id` bzw. `organization-id`. [<=]

Der Fehler `ACCESS_DENIED` wird ebenso erwartet, wenn im jeweiligen Aufrufparameter ein `RecordIdentifier` mit einer falschen `HomeCommunityID` übergeben wird. Eine leere `HomeCommunityID` führt hingegen nicht zu einem Fehler.

#### **A\_16487 - Komponente Autorisierung - Prüfung auf Tokenherkunft**

Die Komponente Autorisierung MUSS jeden Aufruf an den Schnittstellen `I_Authorization_Insurant` und `I_Authorization_Management_Insurant` mit dem Fehler `ACCESS_DENIED` ablehnen, der mittels einer `AuthenticationAssertion` erfolgt, die nicht aus dem Vertrauensraum der Komponente Autorisierung erfolgt. [<=]

#### **A\_15620 - Komponente Autorisierung - Read-only bei suspendiertem Konto**

Die Komponente Autorisierung MUSS die folgenden Operationen mit dem Standardfehler `ACCESS_DENIED` abbrechen,

776 wenn der RecordState der KeyChain des im Aufrufparameter der Operation benannten  
777 RecordIdentifier den Zustand SUSPENDED ausweist:

- 778 • I\_Authorization\_Management::putAuthorizationKey
- 779 • I\_Authorization\_ManagementI\_Authorization\_Management\_Insurant::putAuthoriz  
780 ationKey
- 781 • I\_Authorization\_Management\_Insurant::deleteAuthorizationKey
- 782 • I\_Authorization\_Management\_Insurant::replaceAuthorizationKey
- 783 • I\_Authorization\_Management\_Insurant::putNotificationInfo

784  
785  
786 [ $\leq$ ]

## 787 **A\_17102 - Komponente Autorisierung - Maximale Berechtigungsstufe für Konto-** 788 **Eigentümer**

789 Die Komponente Autorisierung MUSS sicherstellen, dass der AuthorizationType am  
790 hinterlegten AuthorizationKey des Versicherten immer "DOCUMENT\_AUTHORIZATION"  
791 lautet.

792 [ $\leq$ ]

793 Damit soll verhindert werden, dass ein zur Umschlüsselung berechtigter Vertreter  
794 fälschlich einen ungültigen oder einschränkenden AuthorizationKey für den Versicherten  
795 hinterlegt. Dies berührt nicht die Ausstellung einer AuthorizationAssertion mit  
796 ACCOUNT\_AUTHORIZATION für den Fall eines nicht vorhandenen AuthorizationKey bei  
797 Kontoaktivierung/-umzug.

798

## 799 **6.2 Schnittstellen der Komponente Autorisierung**

800 Das Funktionsmerkmal 'Autorisierung' der Komponente Autorisierung wird durch die in  
801 der folgenden Tabelle beschriebenen Schnittstellen mit den jeweiligen Operationen  
802 umgesetzt.

803 **Tabelle 5: Schnittstellen der Komponente Autorisierung**

ENTWURF

Schnittstellen der Komponente Autorisierung	
<b>I_Authorization</b>	
getAuthorizationKey	Mit der Operation <code>getAuthorizationKey</code> wird das für einen Berechtigten verschlüsselte Schlüsselmaterial für ein konkretes Aktenkonto eines Versicherten in der Leistungserbringer-Umgebung und durch den Kostenträger heruntergeladen.
<b>I_Authorization_Management</b>	
putAuthorizationKey	Mit der Operation <code>putAuthorizationKey</code> wird das für einen Berechtigten verschlüsselte Schlüsselmaterial für ein konkretes Aktenkonto eines Versicherten im Aktensystem ePA gespeichert.
checkRecordExists	Mit der Operation <code>checkRecordExists</code> kann ein anderer Anbieter bei einem Anbieter einer Aktenlösung den Status und die Existenz eines Aktenkontos über die KVNR eines Versicherten abfragen.
getAuthorizationList	Die Operation <code>getAuthorizationList</code> liefert die Liste aller OwnerKVNRs des Aktensystems, in denen für die anfragende Institution ein AuthorizationKey hinterlegt ist. (horizontale Abfrage)
<b>I_Authorization_Insurant</b>	
getAuthorizationKey	Mit der Operation <code>getAuthorizationKey</code> wird das für einen Berechtigten verschlüsselte Schlüsselmaterial (kryptografische Berechtigung) für ein konkretes Aktenkonto eines Versicherten in der Personal-Zone heruntergeladen.
<b>I_Authorization_Management_Insurant</b>	
putAuthorizationKey	Mit der Operation <code>putAuthorizationKey</code> wird das für einen Berechtigten verschlüsselte Schlüsselmaterial AuthorizationKey für ein konkretes Aktenkonto eines Versicherten im ePA-Aktensystem gespeichert.

deleteAuthorizationKey	Mit der Operation deleteAuthorizationKey kann ein authentifizierter Versicherter bzw. ein berechtigter Vertreter die kryptografische Berechtigung für einen Nutzer innerhalb seines Aktenkontos löschen.
replaceAuthorizationKey	Mit der Operation replaceAuthorizationKey kann ein authentifizierter Versicherter bzw. ein berechtigter Vertreter das für eine alte eGK verschlüsselte Schlüsselmaterial durch neues für eine Folgekarte verschlüsseltes Schlüsselmaterial ersetzen.
getAuditEvents	Mit der Operation getAuditEvents kann ein authentifizierter Versicherter bzw. ein berechtigter Vertreter das Verwaltungsprotokoll der Komponente Autorisierung auslesen.
putNotificationInfo	Mit der Operation putNotificationInfo kann ein authentifizierter Versicherter bzw. ein berechtigter Vertreter die eigene, im Benachrichtigungskanal hinterlegten Daten aktualisieren.
getAuthorizationList	Die Operation getAuthorizationList liefert die Liste aller AuthorizationKeys zu einer angefragten Akte eines Versicherten. (vertikale Abfrage)

804

## 805 6.2.1 Schnittstelle I\_Authorization

806 Diese Schnittstelle setzt die in [gemSysL\_Fachanwendung\_ePA#4.2.2.2] definierte  
807 Schnittstelle I\_Authorization technisch um.

808 Die Schnittstelle stellt dem Fachmodul eine Operation zum Bezug eines Autorisierungs-  
809 Tokens für bereits authentifizierte Leistungserbringer und Kostenträger bereit, um die  
810 ePA-Komponente Dokumentenverwaltung verwenden zu können.

### 811 6.2.1.1 Operationsdefinition I\_Authorization::getAuthorizationKey

812 **A\_14045-01 - Komponente Autorisierung -**

813 **I\_Authorization::getAuthorizationKey**

814 Die Komponente Autorisierung MUSS die Operation  
815 I\_Authorization::getAuthorizationKey gemäß der folgenden Signatur  
816 implementieren:

817 **Tabelle 6: I\_Authorization::getAuthorizationKey Definition**

Operation	I_Authorization::getAuthorizationKey
-----------	--------------------------------------

Beschreibung	Mit dieser Operation wird für einen authentifizierten Nutzer eine Autorisierung des Zugriffs auf Daten eines Versicherten geprüft.		
Formatvorgaben	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
Eingangsparameter			
Name	Beschreibung	Typ	opt.
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-
RecordIdentifier	Der RecordIdentifier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der kryptografischen Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifierType	-
Ausgangsparameter			
Name	Beschreibung	Typ	opt.
AuthorizationAssertion	Die AuthorizationAssertion ist eine signierte Autorisierungsbestätigung für einen Nutzer und enthält Informationen über die Art und den Umfang der in der Komponente Autorisierung hinterlegten Autorisierung.	SAML Assertion base64-codiert	-
AuthorizationKey	Die kryptografische Autorisierung eines Nutzers.	AuthorizationKeyType	ja
Fehlermeldungen			
Name	Fehlertext	Details	



<b>TECHNICAL_ERROR</b>	Zufallszahl	
<b>ASSERTION_INVALID</b>	Authentifizierungsbestätigung ungültig	Die Authentifizierungsbestätigung des aufrufenden Nutzers wird nicht akzeptiert.
<b>ACCESS_DENIED</b>	Zugriff verweigert	Die Operation ist mit den angegebenen Parametern nicht zulässig.
<b>KEY_ERROR</b>	Fehler im Schlüsseldatensatz	Kein Datensatz für diesen Nutzer für den benannten RecordIdentifier vorhanden.
<b>REPRESENTATIVE_PENDING</b>	Vertretungsberechtigung erfordert Freischaltung	Die Vertretung kann erst wahrgenommen werden, wenn diese über den Freischaltprozess autorisiert wurde.
<b>AUTHORIZATION_ERROR</b>	Autorisierung nicht zulässig	Die zu hinterlegte Berechtigtenrolle ist nicht zulässig.

Sollte bei der Autorisierung für einen Versicherten kein zugehöriger Datensatz gefunden werden, darf dies nicht mit einer technischen Fehlermeldung behandelt werden. Hierfür MUSS eine sprechende Information (fachliches Ereignis) geliefert werden.

[<=]

#### 6.2.1.2 Umsetzung I\_Authorization::getAuthorizationKey

Die folgenden Anforderungen beschreiben die Umsetzung der Operation I\_Authorization::getAuthorizationKey. Dabei gelten die übergreifenden Festlegungen zur Prüfung der Eingangsparameter.

#### A\_17790 - Komponente Autorisierung LE - Vertretung wahrnehmen Freischaltprüfung

Die Komponente Autorisierung MUSS bei Wahrnehmung einer Vertretung für einen Versicherten mittels I\_Authorization::getAuthorizationKey (subject-id der AuthenticationAssertion != OwnerKVNR) vor der Herausgabe prüfen, ob ein wartender Vertreter-Freischaltprozess für [OwnerKVNR des benannten RecordIdentifiers, subject-id als ActorID] aktiv ist und falls ja, die Operation mit dem Fehler REPRESENTATIVE\_PENDING abbrechen.

[<=]

**A\_13917 - Komponente Autorisierung LE - Ausstellen einer Autorisierungsbestätigung**

Die Komponente Autorisierung MUSS in der Operation `I_Authorization::getAuthorizationKey` bei Vorhandensein eines *AuthorizationKey* in der *KeyChain* des benannten *RecordIdentifier* für den mittels *AuthenticationAssertion* authentifizierten Nutzer (`subject-ID` bzw. `organization-id == ActorID`) eine *AuthorizationAssertion* gemäß der Festlegung in [\[A\\_14491\]](#) ausstellen und diese in der Ausgangsnachricht der Operation zurückgeben. Der Wert für `[AuthorizationType]` in der *AuthorizationAssertion* MUSS dem Wert des hinterlegten *AuthorizationKey* genau dieses authentifizierten Nutzers entsprechen. [`<=`]

**A\_17662 - Komponente Autorisierung LE - Codierung der Autorisierungsbestätigung**

Die Komponente Autorisierung MUSS die erstellte und signierte Autorisierungsbestätigung in der Response der Operation `I_Authorization::getAuthorizationKey` Base64-codiert zurückgeben. [`<=`]

**A\_13692 - Komponente Autorisierung LE - Herausgabe kryptografischer Berechtigung des Nutzers**

Die Komponente Autorisierung MUSS in der Operation `I_Authorization::getAuthorizationKey` bei Vorhandensein eines *AuthorizationKey* in der *KeyChain* des benannten *RecordIdentifier* für den mittels *AuthenticationAssertion* authentifizierten Nutzer (`subject-ID` bzw. `organization-id == ActorID`) den *AuthorizationKey* in der Ausgangsnachricht der Operation zurückgeben. [`<=`]

**A\_14643 - Komponente Autorisierung LE - Aktivierung bei Kontoeröffnung in der Umgebung der Leistungserbringer**

Die Komponente Autorisierung MUSS dem authentifizierten Versicherten als Eigentümer der Akte (`subject-ID == OwnerKVNR` für den benannten *RecordIdentifier*) eine Autorisierungsbestätigung mit `AuthorizationType = ACCOUNT_AUTHORIZATION` gemäß [\[A\\_14491\]](#) ausstellen, wenn für seine *OwnerKVNR* kein Schlüsseldatensatz *AuthorizationKey* in der *KeyChain* vorhanden ist. [`<=`]

**A\_15618 - Komponente Autorisierung LE - Autorisierung bei suspendiertem Konto**

Die Komponente Autorisierung MUSS bei Aufruf der Operation `I_Authorization::getAuthorizationKey` bei Vorhandensein eines *AuthorizationKey* in der *KeyChain* des benannten *RecordIdentifier* für den mittels *AuthenticationAssertion* authentifizierten Nutzer (`subject-id = ActorID` des *AuthorizationKey*) eine Autorisierungsbestätigung mit `AuthorizationType = ACCOUNT_AUTHORIZATION` gemäß [\[A\\_14491\]](#) ausstellen, wenn der *RecordState* der *KeyChain* des benannten *RecordIdentifier* den Zustand *SUSPENDED* ausweist. [`<=`]

**6.2.2 Schnittstelle I\_Authorization\_Insurant**

Diese Schnittstelle setzt die in `[gemSysL_ePA]` definierte Schnittstelle `I_Authorization_Insurant` technisch um.

Die Schnittstelle `I_Authorization_Insurant` stellt Operationen zur Autorisierungsprüfung auf das Vorhandensein von kryptografischem Schlüsselmaterial für einen Nutzer des Aktenkontos eines Versicherten bereit. Sie stellt dem Frontend des

888 Versicherten eine Schnittstelle zum Abruf eines Autorisierungs-Tokens für bereits  
889 authentifizierte Versicherte bereit.

890

### 891 6.2.2.1 Operationsdefinition

#### 892 I\_Authorization\_Insurant::getAuthorizationKey

#### 893 A\_14042-01 - Komponente Autorisierung -

#### 894 I\_Authorization\_Insurant::getAuthorizationKey

895 Die Komponente Autorisierung MUSS die Operation

896 I\_Authorization\_Insurant::getAuthorizationKey gemäß der folgenden Signatur  
897 implementieren:

#### 898 Tabelle 7: I\_Authorization\_Insurant::getAuthorizationKey Definition

Operation	I_Authorization_Insurant::getAuthorizationKey		
Beschreibung	Mit dieser Operation wird für einen authentifizierten Nutzer eine Autorisierung des Zugriffs auf Daten eines Versicherten geprüft.		
Formatvorgaben	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
Eingangsparameter			
Name	Beschreibung	Typ	opt.
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-
RecordIdentifier	Der RecordIdentifier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifierType	-

DeviceID	Die DeviceID enthält die Geräteerkennung eines vom Nutzer verwendeten Geräts.	DeviceIdType	-
<b>Ausgangsparameter</b>			
<b>Name</b>	<b>Beschreibung</b>	<b>Typ</b>	<b>opt.</b>
AuthorizationAssertion	Die AuthorizationAssertion ist eine signierte Autorisierungsbestätigung für einen Nutzer und enthält Informationen über die Art und den Umfang der in der Komponente Autorisierung hinterlegten Autorisierung.	SAML Assertion mit AuthorizationDecision Statement base 64-codiert	-
AuthorizationKey	Die kryptografische Autorisierung eines Nutzers.	AuthorizationKeyType	ja
<b>Fehlermeldungen</b>			
<b>Name</b>	<b>Fehlertext</b>	<b>Details</b>	
TECHNICAL_ERROR	Zufallszahl		
ASSERTION_INVALID	Authentifizierungsbestätigung ungültig	Die Authentifizierungsbestätigung des aufrufenden Nutzers wird nicht akzeptiert.	
ACCESS_DENIED	Zugriff verweigert	Die Operation ist mit den angegebenen Parametern nicht zulässig.	
KEY_ERROR	Fehler im Schlüsseldatensatz	Kein Datensatz für diesen Nutzer für den benannten RecordIdentifier vorhanden.	

<b>DEVICE_UNKOWN</b>	generierte phr:DeviceID::Device	Das vom Nutzer verwendete Gerät des Versicherten ist nicht bekannt und muss freigeschaltet werden.
<b>REPRESENTATIVE_PENDING</b>	Vertretungsberechtigung erfordert Freischaltung	Die Vertretung kann erst wahrgenommen werden, wenn diese über den Freischaltprozess autorisiert wurde.

899  
900  
901  
902  
903  
904

Sollte bei der Autorisierung für einen Versicherten kein zugehöriger Datensatz gefunden werden, darf dies nicht mit einer technischen Fehlermeldung behandelt werden. Hierfür MUSS eine sprechende Information (fachliches Ereignis) geliefert werden.

[<=]

#### 905 **6.2.2.2 Umsetzung I\_Authorization\_Insurant::getAuthorizationKey**

906 Die folgenden Anforderungen beschreiben die Umsetzung der Operation  
907 I\_Authorization\_Insurant::getAuthorizationKey. Dabei gelten die übergreifenden  
908 Festlegungen zur Prüfung der Eingangsparameter.

#### 909 **A\_17789 - Komponente Autorisierung Vers. - Vertretung wahrnehmen** 910 **Freischaltprüfung**

911 Die Komponente Autorisierung MUSS bei Wahrnehmung einer Vertretung für einen  
912 Versicherten mittels I\_Authorization\_Insurant::getAuthorizationKey (subject-id  
913 der AuthenticationAssertion != OwnerKVNR) vor der Herausgabe prüfen, ob ein wartender  
914 Vertreter-Freischaltprozess für [OwnerKVNR des benannten RecordIdentifiers, subject-id  
915 als ActorID] aktiv ist und falls ja, die Operation mit dem  
916 Fehler REPRESENTATIVE\_PENDING abbrechen.

917  
918

[<=]

#### 919 **A\_14436 - Komponente Autorisierung Vers. - Ausstellen einer** 920 **Autorisierungsbestätigung**

921 Die Komponente Autorisierung MUSS in der Operation  
922 I\_Authorization\_Insurant::getAuthorizationKey bei Vorhandensein eines  
923 AuthorizationKey in der KeyChain des benannten RecordIdentifier für den mittels  
924 AuthenticationAssertion authentifizierten Nutzer [subject-id der  
925 AuthenticationAssertion == ActorID des vorhandenen AuthorizationKey] eine  
926 AuthorizationAssertion gemäß der Festlegung in [\[A 14491\]](#) ausstellen und diese in der  
927 Ausgangsnachricht der Operation zurückgeben.

928 Der Wert für [AuthorizationType] in der AuthorizationAssertion MUSS dem Wert des  
929 hinterlegten AuthorizationKey genau dieses authentifizierten Nutzers entsprechen.

930

[<=]

931

#### 932 **A\_17663 - Komponente Autorisierung Vers. - Codierung der** 933 **Autorisierungsbestätigung**

934 Die Komponente Autorisierung MUSS die erstellte und signierte  
935 Autorisierungsbestätigung in der Response der

936 Operation `I_Authorization_Insurant::getAuthorizationKey` Base64-codiert  
 937 zurückgeben.  
 938 [`<=`]

#### 939 **A\_14439 - Komponente Autorisierung Vers. - Herausgabe kryptografischer** 940 **Berechtigung des Nutzers**

941 Die Komponente Autorisierung MUSS in der Operation  
 942 `I_Authorization_Insurant::getAuthorizationKey` bei Vorhandensein eines  
 943 `AuthorizationKey` in der `KeyChain` des benannten `RecordIdentifier` für den mittels  
 944 `AuthenticationAssertion` authentifizierten Versicherten oder Vertreter (`subject-id ==`  
 945 `ActorID`) den `AuthorizationKey` des authentifizierten Nutzers in der Ausgangsnachricht  
 946 der Operation zurückgeben.  
 947 [`<=`]

#### 948 **A\_14644 - Komponente Autorisierung Vers. - Aktivierung bei Kontoeröffnung in** 949 **der Umgebung des Versicherten**

950 Die Komponente Autorisierung MUSS bei Aufruf der Operation  
 951 `I_Authorization_Insurant::getAuthorizationKey` dem authentifizierten Versicherten  
 952 als Eigentümer der Akte (`subject-ID == OwnerKVNR` für den benannten  
 953 `RecordIdentifier`) eine Autorisierungsbestätigung mit `AuthorizationType =`  
 954 `ACCOUNT_AUTHORIZATION` gemäß [\[A 14491\]](#) ausstellen, wenn für seine `OwnerKVNR`  
 955 kein Schlüsseldatensatz `AuthorizationKey` in der `KeyChain` vorhanden ist.  
 956 [`<=`]

#### 957 **A\_15619 - Komponente Autorisierung Vers. - Autorisierung bei suspendiertem** 958 **Konto**

959 Die Komponente Autorisierung MUSS bei Aufruf der Operation  
 960 `I_Authorization_Insurant::getAuthorizationKey` bei Vorhandensein eines  
 961 `AuthorizationKey` in der `KeyChain` des benannten `RecordIdentifier` für den mittels  
 962 `AuthenticationAssertion` authentifizierten Nutzer (`subject-id = ActorID` des  
 963 `AuthorizationKey`) eine Autorisierungsbestätigung mit `AuthorizationType =`  
 964 `ACCOUNT_AUTHORIZATION` gemäß [\[A 14491\]](#) ausstellen, wenn der `RecordState` der  
 965 `KeyChain` des benannten `RecordIdentifier` den Zustand `SUSPENDED` ausweist.[`<=`]

### 966 **6.2.3 Schnittstelle `I_Authorization_Management`**

967 Diese Schnittstelle setzt die in `[gemSysL_ePA]` definierte Schnittstelle  
 968 `I_Authorization_Management` technisch um.

969 Die Schnittstelle `I_Authorization_Management` dient dazu, kryptografische  
 970 Berechtigungen im Autorisierungsdienst eines Aktensystems zu verwalten.

#### 971 **6.2.3.1 Operationsdefinition**

#### 972 **`I_Authorization_Management::putAuthorizationKey`**

#### 973 **A\_14180-01 - Komponente Autorisierung -**

#### 974 **`I_Authorization_Management::putAuthorizationKey`**

975 Die Komponente Autorisierung MUSS die Operation  
 976 `I_Authorization_Management::putAuthorizationKey` gemäß der folgenden Signatur  
 977 implementieren:

#### 978 **Tabelle 8: `I_Authorization_Management::putAuthorizationKey` - Definition**

Operation	<code>I_Authorization_Management::putAuthorizationKey</code>
-----------	--------------------------------------------------------------

<b>Beschreibung</b>	Mit der Operation wird das für einen Berechtigten verschlüsselte Schlüsselmaterial für ein konkretes Aktenkonto eines Versicherten im ePA-Aktensystem gespeichert.		
<b>Formatvorgaben</b>	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
<b>Eingangsparameter</b>			
<b>Name</b>	<b>Beschreibung</b>	<b>Typ</b>	<b>opt.</b>
<b>AuthenticationAssertion</b>	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-
<b>RecordIdentifier</b>	Der RecordIdentifier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifierType	-
<b>AuthorizationKey</b>	Die kryptografische Autorisierung eines Nutzers, bestehend aus Listen von verschlüsselten Schlüsseln. Details zur Struktur finden sich im Kapitel 7 zum Informationsmodell.	AuthorizationKeyType	-
<b>Fehlermeldungen</b>			
<b>Name</b>	<b>Fehlertext</b>	<b>Details</b>	
<b>TECHNICAL_ERROR</b>	Zufallszahl	.	



<b>ASSERTION_INVALID</b>	Authentifizierungsbestätigung ungültig	Die Authentifizierungsbestätigung des aufrufenden Nutzers wird nicht akzeptiert.
ACCESS_DENIED	Zugriff verweigert	Die Operation ist mit den angegebenen Parametern nicht zulässig.

[<=]

### 6.2.3.2 Umsetzung `I_Authorization_Management::putAuthorizationKey`

Die folgenden Anforderungen beschreiben die Umsetzung der Operation `I_Authorization_Management::putAuthorizationKey`. Dabei gelten die übergreifenden Festlegungen zur Prüfung der Eingangsparameter.

#### **A\_14212 - Komponente Autorisierung LE - Speicherung kryptografische Berechtigung des Nutzers**

Die Komponente Autorisierung MUSS in der Operation `I_Authorization_Management::putAuthorizationKey` den im Eingangsparameter übergebenen `AuthorizationKey` als *AuthorizationKey* der KeyChain des im Eingangsparameter benannten `RecordIdentifizier` speichern bzw. ersetzen, falls für die im `AuthorizationKey` benannte `ActorID` bereits ein *AuthorizationKey* in der KeyChain des benannten `RecordIdentifizier` existiert.[<=]

#### **A\_14441 - Komponente Autorisierung LE - Berechtigungsprüfung Schlüssel hinterlegung**

Die Komponente Autorisierung MUSS beim Aufruf der Operation `I_Authorization_Management::putAuthorizationKey` anhand der KVNRR der `AuthenticationAssertion` und des `RecordIdentifizier` prüfen, ob für den aufrufenden Nutzer ein `AuthorizationKey` mit `ActorID == subject-ID` hinterlegt ist, und falls nicht, die Operation mit dem Fehler `ACCESS_DENIED` abbrechen.[<=]

Mit dieser Prüfung wird sichergestellt, dass nur Versicherte bzw. Vertreter einen Schlüssel für einen Berechtigten hinterlegen können. Eine Berechtigung wird nicht von einer Leistungserbringerinstitution oder von einem Kostenträger hinterlegt.

#### **A\_14587 - Komponente Autorisierung LE - Initiale Schlüssel hinterlegung Kontoeröffnung**

Die Komponente Autorisierung MUSS die Operation `I_Authorization_Management::putAuthorizationKey` mit dem Fehler `ACCESS_DENIED` abbrechen, sofern für den Eigentümer der Akte noch kein `AuthorizationKey` vorhanden ist und der zu speichernde `AuthorizationKey` des Aufrufparameters für einen anderen Nutzer als den Eigentümer des `RecordIdentifizier` (`ActorID != OwnerKVNRR`) gespeichert werden soll.[<=]

Mit dieser Anforderung soll verhindert werden, dass die Akte genutzt wird, bevor das Schlüsselmaterial für den Versicherten erzeugt und hinterlegt wurde. Die benannte Konstellation liegt im Rahmen der Kontoeröffnung und bei einem Aktenumzug vor. Das



1017 Schlüsselmaterial für den Versicherten wird im Schritt der Kontoaktivierung erzeugt,  
1018 welcher auf den Schritt der Kontoinitialisierung folgt.

#### 1019 **A\_14737 - Komponente Autorisierung LE - Initiale Schlüsselhinterlegung für** 1020 **den Versicherten**

1021 Die Komponente Autorisierung MUSS bei Aufruf der  
1022 Operation `I_Authorization_Management::putAuthorizationKey` durch den  
1023 Versicherten (subject-id (KVNR) der `AuthenticationAssertion == OwnerKVNR`) im  
1024 Rahmen der initialen Schlüsselhinterlegung während der Kontoaktivierung das `validTo`-  
1025 Datum des übergebenen `AuthorizationKey` vor der Speicherung mit einem technischen  
1026 Datum gleichbedeutend mit "unendlich" (z.B. 31.12.9999) ersetzen. [`<=`]

#### 1027 **A\_14999 - Komponente Autorisierung LE - Zustandswechsel bei** 1028 **Schlüsselhinterlegung für den Versicherten**

1029 Die Komponente Autorisierung MUSS bei Aufruf der  
1030 Operation `I_Authorization_Management::putAuthorizationKey` durch den  
1031 Versicherten (subject-id (KVNR) der `AuthenticationAssertion == OwnerKVNR`) bei  
1032 erfolgreichem Abschluss der initialen Schlüsselhinterlegung für den Versicherten während  
1033 der Kontoaktivierung den Zustand `RecordState` der `KeyChain` des Versicherten von  
1034 `REGISTERED` auf den Wert `ACTIVATED` setzen.  
1035 [`<=`]

### 1036 **6.2.3.3 Operationsdefinition**

#### 1037 **I\_Authorization\_Management::checkRecordExists**

#### 1038 **A\_14965 - Komponente Autorisierung -**

#### 1039 **I\_Authorization\_Management::checkRecordExists**

1040 Die Komponente Autorisierung MUSS die  
1041 Operation `I_Authorization_Management::checkRecordExists` gemäß der folgenden  
1042 Signatur implementieren:

1043 **Tabelle 9: I\_Authorization\_Management::checkRecordExists - Definition**

<b>Operation</b>	<b>I_Authorization_Management::checkRecordExists</b>		
<b>Beschreibung</b>	Die Operation liefert den Status eines Aktenkontos eines via KVNR benannten Versicherten.		
<b>Formatvorgaben</b>	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
<b>Eingangsparameter</b>			
<b>Name</b>	<b>Beschreibung</b>	<b>Typ</b>	<b>opt.</b>
KVNR	Der unveränderliche Teil der Krankenversicherungsnummer eines gesetzlich Versicherten	String	-
<b>Ausgangsparameter</b>			

Name	Beschreibung	Typ	opt.
RecordState	Statuswert zur Existenz eines Aktenkontos in der Komponente Autorisierung zu einer angefragten KVNR	RecordStateType	-
<b>Fehlermeldungen</b>			
Name	Fehlertext	Details	
TECHNICAL_ERROR	Zufallszahl		

1044  
1045

[<=]

#### 1046 **6.2.3.4 Umsetzung I\_Authorization\_Management::checkRecordExists**

1047 Die folgenden Anforderungen beschreiben die Umsetzung der Operation  
1048 I\_Authorization\_Management::checkRecordExists. Dabei gelten die übergreifenden  
1049 Festlegungen zur Prüfung der Eingangsparameter.

#### 1050 **A\_14966 - Komponente Autorisierung LE - Abfrage Aktenexistenz**

1051 Die Komponente Autorisierung MUSS bei Aufruf der Operation  
1052 I\_Authorization\_Management::checkRecordExists den Wert des RecordState  
1053 des Datensatzes KeyChain eines Konto zurückliefern, wenn zu einer angefragten KVNR ein  
1054 Datensatz KeyChain mit OwnerKVNR == KVNR existiert und andernfalls den Statuswert  
1055 UNKNOWN zurückgeben.[<=]

#### 1056 **6.2.3.5 Operationsdefinition**

#### 1057 **I\_Authorization\_Management::getAuthorizationList**

#### 1058 **A\_17110 - Komponente Autorisierung -**

#### 1059 **I\_Authorization\_Management::getAuthorizationList**

1060 Die Komponente Autorisierung MUSS die  
1061 Operation I\_Authorization\_Management::getAuthorizationList gemäß der  
1062 folgenden Signatur implementieren:

1063 **Tabelle 10: I\_Authorization\_Management::getAuthorizationList - Definition**

Operation	I_Authorization_Management::getAuthorizationList
<b>Beschreibung</b>	Die Operation liefert eine Liste der OwnerKVNRs von Konten im Aktensystem, in denen die anfragende Identität berechtigt ist.
<b>Formatvorgaben</b>	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.
<b>Eingangsparameter</b>	

Name	Beschreibung	Typ	opt.
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-
<b>Ausgangsparameter</b>			
Name	Beschreibung	Typ	opt.
AuthorizationInfoList	Liste der OwnerKVNRS von Konten im Aktensystem, in denen für die Telematik-ID der anfragenden Leistungserbringerinstitution bzw. der Kostenträger ein AuthorizationKey aktuell vorhanden ist.	AuthorizationInfo[0..*]	-
<b>Fehlermeldungen</b>			
Name	Fehlertext	Details	
ASSERTION_INVALID	Die übergebene AuthenticationAssertion ist ungültig.	z.B. abgelaufen oder Misstrauen in Signatur des Tokens	
TECHNICAL_ERROR	Zufallszahl		

[&lt;=]

### 6.2.3.6 Umsetzung I\_Authorization\_Management::getAuthorizationList

Die folgenden Anforderungen beschreiben die Umsetzung der Operation I\_Authorization\_Management::getAuthorizationList. Dabei gelten die übergreifenden Festlegungen zur Prüfung der Eingangsparameter.

#### A\_17111 - Komponente Autorisierung LE - Abfrage Berechtigungsliste

Die Komponente Autorisierung MUSS bei Aufruf der Operation I\_Authorization\_Management::getAuthorizationList die Liste aller OwnerKVNRS ermitteln, in deren KeyChain für die organization-id der gültigen AuthenticationAssertion ein AuthorizationKey vorhanden ist (organization-id == ActorID) und diese Liste als AuthorizationInformation [OwnerKVNRS + validTo am jeweiligen AuthorizationKey der ActorID je KeyChain] zurückgeben.

[&lt;=]

## **A\_19007 - Komponente Autorisierung - Einschränkung der Häufigkeit der Abfrage getAuthorizationList**

Das Aktensystem KANN getAuthorizationList-Anfragen mit dem Fehler TOO\_MANY\_REQUESTS zurückweisen, wenn sie von derselben LEI (bei Gleichheit der organization-id) innerhalb eines Zeitraumes von 10 Minuten wiederholt gestellt werden.

[<=]

## **6.2.4 Schnittstelle I\_Authorization\_Management\_Insurant**

Diese Schnittstelle setzt die in [gemSysL\_ePA] definierte Schnittstelle I\_Authorization\_Management\_Insurant technisch um.

Die Schnittstelle I\_Authorization\_Management\_Insurant stellt Operationen zur Verwaltung von kryptografischen Berechtigungen im Autorisierungsdienst eines Aktensystems bereit.

### **6.2.4.1 Operationsdefinition**

#### **I\_Authorization\_Management\_Insurant::putAuthorizationKey**

#### **A\_14672-01 - Komponente Autorisierung -**

#### **I\_Authorization\_Management\_Insurant::putAuthorizationKey**

Die Komponente Autorisierung MUSS die Operation I\_Authorization\_Management\_Insurant::putAuthorizationKey gemäß der folgenden Signatur implementieren:

**Tabelle 11: I\_Authorization\_Management\_Insurant::putAuthorizationKey - Definition**

<b>Operation</b>	<b>I_Authorization_Management_Insurant::putAuthorizationKey</b>		
<b>Beschreibung</b>	Mit dieser Operation wird für einen Berechtigten verschlüsseltes Schlüsselmaterial für ein konkretes Aktenkonto eines Versicherten im Aktensystem gespeichert.		
<b>Formatvorgaben</b>	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
<b>Eingangsparameter</b>			
<b>Name</b>	<b>Beschreibung</b>	<b>Typ</b>	<b>opt.</b>
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-

RecordIdentifier	Der RecordIdentifier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifierType	-
AuthorizationKey	Die kryptografische Autorisierung eines Nutzers, bestehend aus Listen von verschlüsselten Schlüsseln. Details zur Struktur finden sich im Kapitel 7 zum Informationsmodell.	AuthorizationKeyType	-
DeviceID	Die DeviceID enthält die Gerätekennung eines vom Nutzer verwendeten Gerätes.	DeviceIdType	-
NotificationInfoRepresentative	Mit diesem Parameter hinterlegt der Versicherte eine Benachrichtigungsadresse der Geräteverwaltung des mittels AuthorizationKey berechtigten Vertreters.	String	ja
<b>Fehlermeldungen</b>			
<b>Name</b>	<b>Fehlertext</b>	<b>Details</b>	
TECHNICAL_ERROR	Zufallszahl		
ASSERTION_INVALID	Authentifizierungsbestätigung ungültig	Die Authentifizierungsbestätigung des aufrufenden Nutzers wird nicht akzeptiert.	
KEY_ERROR	Fehler im Schlüsseldatensatz	Es ist bereits ein Datensatz vorhanden.	
SYNTAX_ERROR	Fehlerhafte Aufrufparameter	Es wurde ein fehlerhafter Aufrufparameter übergeben.	

DEVICE_UNKNOWN	generierte phr:DeviceID::Device	Das vom Nutzer verwendete Gerät des Versicherten ist nicht bekannt und muss freigeschaltet werden.
----------------	------------------------------------	----------------------------------------------------------------------------------------------------

1100

1101 [ $\leq$ ]1102 **6.2.4.2 Umsetzung**1103 **I\_Authorization\_Management\_Insurant::putAuthorizationKey**

1104 Die folgenden Anforderungen beschreiben die Umsetzung der Operation  
 1105 I\_Authorization\_Management\_Insurant::putAuthorizationKey. Dabei gelten die  
 1106 übergreifenden Festlegungen zur Prüfung der Eingangsparameter.

1107 **A\_14446 - Komponente Autorisierung Vers. - Speicherung kryptografische**  
 1108 **Berechtigung des Nutzers**

1109 Die Komponente Autorisierung MUSS in der Operation  
 1110 I\_Authorization\_Management\_Insurant::putAuthorizationKey den im  
 1111 Eingangsparameter übergebenen AuthorizationKey als AuthorizationKey der KeyChain  
 1112 des im Eingangsparameter benannten RecordIdentifier speichern, sofern kein  
 1113 AuthorizationKey für die ActorID zu diesem RecordIdentifier bereits vorhanden ist, und  
 1114 andernfalls die Operation mit der Fehlermeldung KEY\_ERROR abbrechen.

1115 [ $\leq$ ]

1116 **A\_14447 - Komponente Autorisierung Vers. - Berechtigungsprüfung**  
 1117 **Schlüssel hinterlegung**

1118 Die Komponente Autorisierung MUSS beim Aufruf der  
 1119 Operation I\_Authorization\_Management\_Insurant::putAuthorizationKey anhand der  
 1120 subject-id (KVNR) der AuthenticationAssertion und des RecordIdentifier prüfen, ob  
 1121 für den aufrufenden Nutzer ein AuthorizationKey mit ActorID = KVNR hinterlegt ist und  
 1122 falls nicht, die Operation mit dem Fehler ACCESS\_DENIED abbrechen. [ $\leq$ ]

1123 Mit dieser Prüfung wird sichergestellt, dass nur Versicherte sowie berechtigte Vertreter  
 1124 Schlüsselmaterial für Versicherte, Leistungserbringerinstitutionen und Kostenträger  
 1125 hinterlegen können, die selbst bereits über einen AuthorizationKey verfügen.

1126 **A\_18184 - Komponente Autorisierung Vers. - Prüfung auf**  
 1127 **Vertretungsberechtigung für Prüffidentität**

1128 Die Komponente Autorisierung MUSS bei Hinterlegung einer Vertretungsberechtigung  
 1129 durch Aufruf der  
 1130 Operation I\_Authorization\_Management\_Insurant::putAuthorizationKey mit  
 1131 (subject-id der AuthenticationAssertion != ActorID des Übergabeparameters  
 1132 AuthorizationKey und ActorID des Übergabeparameters AuthorizationKey !=  
 1133 OwnerKVNR) prüfen, ob die Hinterlegung für eine Prüffidentität gemäß  
 1134 [gemSpec\_PK\_eGK#Card-G2-A\_3820] erfolgen soll und falls ja, den Anwendungsfall mit  
 1135 dem Fehler TECHNICAL\_ERROR abbrechen. [ $\leq$ ]

1136 Die Erkennung auf eine Prüffidentität kann über die Auswertung der ActorID des zu  
 1137 berechtigenden Vertreters erfolgen, wobei diese als Prüf-KVNR anhand der Bildungsregel  
 1138 "4 oder mehr gleiche aufeinander folgende Ziffern" eindeutig zu erkennen ist.

**A\_17670 - Komponente Autorisierung Vers. - Freischaltprozess****Vertreterberechtigung**

Die Komponente Autorisierung MUSS bei Hinterlegung einer Vertretungsberechtigung durch Aufruf der

Operation `I_Authorization_Management_Insurant::putAuthorizationKey` mit (subject-id der `AuthenticationAssertion` != ActorID des Übergabeparameters `AuthorizationKey` und ActorID des Übergabeparameters `AuthorizationKey` != OwnerKVNR) die Operation abschließen, sofern kein technischer oder fachlicher Fehler dies verhindert und anschließend den Freischaltprozess für Vertreter Einrichtung starten (6.6- Freischaltprozess Vertreter Einrichtung ), sofern für die im Übergabeparameter `AuthorizationKey` benannte ActorID noch kein `AuthorizationKey` in der Komponente Autorisierung für die im `RecordIdentifier` benannte OwnerKVNR vorhanden ist.

[<=]

**A\_18750 - Komponente Autorisierung Vers. - Begrenzung zu registrierender Vertreter**

Die Komponente Autorisierung MUSS bei Hinterlegung einer Vertretungsberechtigung durch Aufruf der Operation

`I_Authorization_Management_Insurant::putAuthorizationKey` (vgl. A\_17670) prüfen, ob die maximale Anzahl von fünf Vertretern erreicht wurde. Trifft dies zu, MUSS der Anwendungsfall mit dem Fehler `TECHNICAL_ERROR` abgebrochen werden. Eine Prüfung MUSS berücksichtigen, ob zum Zeitpunkt der Vertretungsregistrierung Freischaltprozesse gestartet wurden bzw. im Gange sind. Diese Prozesse sind in der maximalen Anzahl an Vertretern zu berücksichtigen.

[<=]

**A\_15752 - Komponente Autorisierung Vers. - Benachrichtigungskanal für Geräteverwaltung E-Mail-Format**

Die Komponente Autorisierung MUSS die Operation

`I_Authorization_Management_Insurant::putAuthorizationKey` mit dem Fehler `SYNTAX_ERROR` abbrechen, wenn der Parameter `NotificationInfoRepresentative` nicht leer und nicht gemäß [\[RFC-5322\]](#) formatiert ist.[<=]

**A\_14318 - Komponente Autorisierung Vers. - Benachrichtigungskanal für Geräteverwaltung**

Die Komponente Autorisierung MUSS einen in der Operation

`I_Authorization_Management_Insurant::putAuthorizationKey` übergebenen optionalen Parameter `NotificationInfoRepresentative` als Benachrichtigungsadresse der Geräteverwaltung für den im Parameter `AuthorizationKey` durch ActorID benannten Nutzer übernehmen.[<=]

**A\_14615 - Komponente Autorisierung Vers. - Initiale Schlüssel hinterlegung Kontoeröffnung**

Die Komponente Autorisierung MUSS die Operation

`I_Authorization_Management_Insurant::putAuthorizationKey` mit dem Fehler `ACCESS_DENIED` abbrechen, sofern für den Eigentümer der Akte noch kein `AuthorizationKey` vorhanden ist, und der zu speichernde `AuthorizationKey` des Aufrufparameters für einen anderen Nutzer als den Eigentümer des `RecordIdentifier` (ActorID != OwnerKVNR) gespeichert werden soll.[<=]

Mit dieser Anforderung soll verhindert werden, dass die Akte genutzt wird, bevor das Schlüsselmaterial für den Versicherten erzeugt und hinterlegt wurde. Die benannte Konstellation liegt im Rahmen der Kontoeröffnung und bei einem Aktenumzug vor. Das Schlüsselmaterial für den Versicherten wird im Schritt der Kontoaktivierung erzeugt, welcher auf den Schritt der Kontoinitialisierung folgt.



### **A\_14736 - Komponente Autorisierung Vers. - Initiale Schlüssel hinterlegung für den Versicherten**

Die Komponente Autorisierung MUSS bei Aufruf der Operation `I_Authorization_Management_Insurant::putAuthorizationKey` durch den Versicherten (subject-id (KVNR) der `AuthenticationAssertion` == `OwnerKVNR`) im Rahmen der initialen Schlüssel hinterlegung während der Kontoaktivierung das `validTo`-Datum des übergebenen `AuthorizationKey` vor der Speicherung mit einem technischen Datum gleichbedeutend mit "unendlich" (z.B. 31.12.9999) ersetzen. [`<=`]

### **A\_15000 - Komponente Autorisierung Vers. - Zustandswechsel bei Schlüssel hinterlegung für den Versicherten**

Die Komponente Autorisierung MUSS bei Aufruf der Operation `I_Authorization_Management_Insurant::putAuthorizationKey` durch den Versicherten (subject-id (KVNR) der `AuthenticationAssertion` == `OwnerKVNR`) bei erfolgreichem Abschluss der initialen Schlüssel hinterlegung für den Versicherten während der Kontoaktivierung den Zustand `RecordState` der `KeyChain` des Versicherten von `REGISTERED` bzw. `REGISTERED_FOR_MIGRATION` auf den Wert `ACTIVATED` setzen. [`<=`]

## **6.2.4.3 Operationsdefinition**

### **I\_Authorization\_Management\_Insurant::deleteAuthorizationKey**

#### **A\_14674-01 - Komponente Autorisierung -**

#### **I\_Authorization\_Management\_Insurant::deleteAuthorizationKey**

Die Komponente Autorisierung MUSS die Operation `I_Authorization_Management_Insurant::deleteAuthorizationKey` gemäß der folgenden Signatur implementieren:

**Tabelle 12: I\_Authorization\_Management\_Insurant::deleteAuthorizationKey - Definition**

<b>Operation</b>	<b>I_Authorization_Management_Insurant::deleteAuthorizationKey</b>		
<b>Beschreibung</b>	Mit dieser Operation kann ein authentifizierter Nutzer bzw. ein berechtigter Vertreter das im Aktenkonto hinterlegte kryptografische Schlüsselmaterial für einen benannten Nutzer löschen.		
<b>Formatvorgaben</b>	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
<b>Eingangsparameter</b>			
<b>Name</b>	<b>Beschreibung</b>	<b>Typ</b>	<b>opt.</b>
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte	SAML Assertion im SOAP-Header des Requests	-



	Authentifizierungsbestätigung für einen Nutzer.		
RecordIdentifizier	Der RecordIdentifizier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifizierType	-
ActorID	Identifikator des Nutzers, für den der hinterlegte Datensatz AuthorizationKey gelöscht werden soll.	String	-
DeviceID	Die DeviceID enthält die Geräteerkennung eines vom Nutzer verwendeten Gerätes.	DeviceIdType	-
<b>Fehlermeldungen</b>			
<b>Name</b>	<b>Fehlertext</b>	<b>Details</b>	
TECHNICAL_ERROR	Zufallszahl		
ASSERTION_INVALID	Authentifizierungsbestätigung ungültig	Die Authentifizierungsbestätigung des aufrufenden Nutzers wird nicht akzeptiert.	
KEY_ERROR	Fehler im Schlüsseldatensatz	Kein Datensatz vorhanden	
ACCESS_DENIED	Zugriff verweigert	Die Operation ist mit den angegebenen Parametern nicht zulässig.	
DEVICE_UNKNOWN	generierte phr:DeviceID::Device	Das vom Nutzer verwendete Gerät des Versicherten ist nicht bekannt und muss freigeschaltet werden.	

1214

1215 [**<=**]

1216

1217

1218 **6.2.4.4 Umsetzung**1219 **I\_Authorization\_Management\_Insurant::deleteAuthorizationKey**

1220 Die folgenden Anforderungen beschreiben die Umsetzung der Operation

1221 I\_Authorization\_Management\_Insurant::deleteAuthorizationKey. Dabei gelten die  
1222 übergreifenden Festlegungen zur Prüfung der Eingangsparameter.1223 **A\_14451 - Komponente Autorisierung Vers. - Prüfen Löschberechtigung**

1224 Die Komponente Autorisierung MUSS bei Aufruf der Operation

1225 I\_Authorization\_Management\_Insurant::deleteAuthorizationKey prüfen, ob der in  
1226 der AuthenticationAssertion benannte Nutzer über einen AuthorizationKey mit  
1227 AuthorizationType = DOCUMENT\_AUTHORIZATION für den benannten RecordIdentifizier  
1228 verfügt, und andernfalls die Operation mit der Fehlermeldung ACCESS\_DENIED  
1229 abbrechen.

1230

1231 [**<=**]1232 **A\_14452 - Komponente Autorisierung Vers. - Löschen des AuthorizationKeys**

1233 Die Komponente Autorisierung MUSS bei Aufruf der Operation

1234 I\_Authorization\_Management\_Insurant::deleteAuthorizationKey  
1235 den Datensatz AuthorizationKey des Nutzers löschen, der im Aufrufparameter als  
1236 ActorID (Telematik-ID oder KVNR für Vertreter) benannt wurde. [**<=**]1237 **A\_14453 - Komponente Autorisierung Vers. - Löschverbot für**1238 **Versichertenschlüssel**

1239 Die Komponente Autorisierung MUSS bei Aufruf der Operation

1240 I\_Authorization\_Management\_Insurant::deleteAuthorizationKey  
1241 das Löschen verhindern, wenn der im Aufrufparameter als ActorID benannte Datensatz  
1242 gleich der OwnerKVNR des Versicherten als Eigentümer der Akte ist, und die Operation mit  
1243 der Fehlermeldung ACCESS\_DENIED abbrechen. [**<=**]1244 **A\_14552-01 - Komponente Autorisierung Vers. - Löschen veralteter Schlüssel**1245 Die Komponente Autorisierung MUSS alle AuthorizationKey löschen, deren validTo-  
1246 Datum älter als die aktuelle Systemzeit der Komponente Autorisierung sind und das  
1247 Löschen mit den folgenden Parametern als PHR-421 protokollieren:

- 1248
- UserID = interner, systemseitig wählbarer Identifikator
  - UserName = Automatische Löschung nach Ablauf der Berechtigungsdauer
  - ObjectID = RecordIdentifizier des betroffenen Kontos
  - ObjectName = ActorID des gelöschten AuthorizationKey.

1252 [**<=**]1253 **6.2.4.5 Operationsdefinition**1254 **I\_Authorization\_Management\_Insurant::replaceAuthorizationKey**1255 **A\_14325-01 - Komponente Autorisierung -**1256 **I\_Authorization\_Management\_Insurant::replaceAuthorizationKey**

1257 Die Komponente Autorisierung MUSS die Operation

1258 I\_Authorization\_Management\_Insurant::replaceAuthorizationKey gemäß der  
1259 folgenden Signatur implementieren:

1260 **Tabelle 13: I\_Authorization\_Management\_Insurant::replaceAuthorizationKey - Definition**

Operation	I_Authorization_Management_Insurant::replaceAuthorizationKey		
Beschreibung	Mit dieser Operation kann ein authentifizierter Nutzer bzw. ein berechtigter Vertreter das für eine alte eGK verschlüsselte Schlüsselmaterial durch neues für eine Folgekarte verschlüsseltes Schlüsselmaterial ersetzen.		
Formatvorgaben	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
Eingangsparameter			
Name	Beschreibung	Typ	opt.
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-
RecordIdentifier	Der RecordIdentifier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifierType	-
NewAuthorizationKey	Die kryptografische Autorisierung eines Nutzers, bestehend aus Listen von verschlüsselten Schlüsseln. Details zur Struktur finden sich im Kapitel 7 zum Informationsmodell.	AuthorizationKeyType	-
DeviceID	Die DeviceID enthält die Gerätekennung eines vom Nutzer verwendeten Gerätes.	DeviceIdType	-
Fehlermeldungen			
Name	Fehlertext	Details	
TECHNICAL_ERROR	Zufallszahl		

KEY_ERROR	Fehler im Schlüsseldatensatz	Kein Datensatz vorhanden.
<b>ASSERTION_INVALID</b>	Authentifizierungsbestätigung ungültig	Die Authentifizierungsbestätigung des aufrufenden Nutzers wird nicht akzeptiert.
<b>DEVICE_UNKNOWN</b>	generierte <code>phr:DeviceID::Device</code>	Das vom Nutzer verwendete Gerät des Versicherten ist nicht bekannt und muss freigeschaltet werden.
ACCESS_DENIED	Zugriff verweigert	Die Operation ist mit den angegebenen Parametern nicht zulässig.

1261  
1262  
1263 [**<=**]

#### 6.2.4.6 Umsetzung

##### **I\_Authorization\_Management\_Insurant::replaceAuthorizationKey**

Die folgenden Anforderungen beschreiben die Umsetzung der Operation

`I_Authorization_Management_Insurant::replaceAuthorizationKey`. Dabei gelten die übergreifenden Festlegungen zur Prüfung der Eingangsparameter.

##### **A\_14454 - Komponente Autorisierung Vers. - Prüfung Datensatz für bestehenden AuthorizationKey**

Die Komponente Autorisierung MUSS für die Operation

`I_Authorization_Management_Insurant::replaceAuthorizationKey` prüfen, ob ein *AuthorizationKey* für den benannten *RecordIdentifier* und den in der *AuthenticationAssertion* benannten Nutzer (`subject-id == ActorID` des vorhandenen *AuthorizationKey*) hinterlegt ist, und andernfalls die Operation mit der Fehlermeldung `ACCESS_DENIED` abbrechen.[**<=**]

##### **A\_14455 - Komponente Autorisierung Vers. - Ersetzen des AuthorizationKeys**

Die Komponente Autorisierung MUSS bei Aufruf der Operation

`I_Authorization_Management_Insurant::replaceAuthorizationKey` den Datensatz *AuthorizationKey* desjenigen Nutzers durch den übergebenen *NewAuthorizationKey* ersetzen, der im Aufrufparameter als *ActorID* (Telematik-ID oder KVNR) benannt wurde und für den ein *AuthorizationKey* vorhanden ist.[**<=**]

##### **A\_15120 - Komponente Autorisierung Vers. - Fixierung des AuthorizationType für Vertreter**

Die Komponente Autorisierung MUSS bei Aufruf der Operation

`I_Authorization_Management_Insurant::replaceAuthorizationKey` prüfen, ob ein Vertreter seinen eigenen Schlüssel ersetzt (`OwnerKVNR != subject-id == ActorID` des vorhandenen *AuthorizationKey* `== ActorID` in *NewAuthorizationKey*) und in diesem Fall den *AuthorizationType* des vorhandenen *AuthorizationKey* in den zu speichernden *NewAuthorizationKey* übernehmen.

Die Komponente Autorisierung MUSS die Operation mit dem Fehler ACCESS\_DENIED abbrechen, wenn ein lediglich zur Umschlüsselung berechtigter Vertreter (RECOVERY\_AUTHORIZATION im hinterlegten AuthorizationKey des Vertreters) versucht einen anderen AuthorizationKey zu ersetzen als den eigenen oder den des Versicherten.

[<=]

#### **A\_15889 - Komponente Autorisierung Vers. - Prüfung KVNR bei Schlüsselwechsel für den Versicherten**

Die Komponente Autorisierung MUSS den Aufruf der Operation `I_Authorization_Management_Insurant::replaceAuthorizationKey` durch den Versicherten als Eigentümer der Akte (ActorId des übergebenen AuthorizationKey == OwnerKVNR für den benannten RecordIdentifier) mit der Fehlermeldung ACCESS\_DENIED abbrechen, wenn der unveränderliche Teil der KVNR des Versicherten im übergebenen AuthorizationKey nicht übereinstimmt mit dem unveränderlichen Teil der KVNR des Versicherten im bereits gespeicherten AuthorizationKey.

[<=]

#### **6.2.4.7 Operationsdefinition**

##### **I\_Authorization\_Management\_Insurant::getAuditEvents**

##### **A\_14676-01 - Komponente Autorisierung -**

##### **I\_Authorization\_Management\_Insurant::getAuditEvents**

Die Komponente Autorisierung MUSS die Operation `I_Authorization_Management_Insurant::getAuditEvents` gemäß der folgenden Signatur implementieren:

**Tabelle 14: I\_Authorization\_Management\_Insurant::getAuditEvents - Definition**

Operation	I_Authorization_Management_Insurant::getAuditEvents		
Beschreibung	Mit dieser Operation kann ein authentifizierter Versicherter bzw. ein berechtigter Vertreter das Verwaltungsprotokoll der Autorisierungskomponente auslesen.		
Formatvorgaben	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
Eingangsparameter			
Name	Beschreibung	Typ	opt.
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-

RecordIdentifier	Der RecordIdentifier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifierType	-
DeviceID	Die DeviceID enthält die Geräteerkennung eines vom Nutzer verwendeten Gerätes.	DeviceIdType	-
<b>Ausgangsparameter</b>			
<b>Name</b>	<b>Beschreibung</b>	<b>Typ</b>	<b>opt.</b>
AuditEventList	Liste der Verwaltungsprotokolleinträge des im RecordIdentifier referenzierten Aktenkontos	AuditMessage [0..*]	-
<b>Fehlermeldungen</b>			
<b>Name</b>	<b>Fehlertext</b>	<b>Details</b>	
TECHNICAL_ERROR	Zufallszahl		
ASSERTION_INVALID	Authentifizierungsbestätigung ungültig	Die Authentifizierungsbestätigung des aufrufenden Nutzers wird nicht akzeptiert.	
DEVICE_UNKNOWN	generierte phr:DeviceID::Device	Das vom Nutzer verwendete Gerät des Versicherten ist nicht bekannt und muss freigeschaltet werden.	

[&lt;=]

#### 6.2.4.8 Umsetzung

##### I\_Authorization\_Management\_Insurant::getAuditEvents

Die folgenden Anforderungen beschreiben die Umsetzung der Operation I\_Authorization\_Management\_Insurant::getAuditEvents. Dabei gelten die übergreifenden Festlegungen zur Prüfung der Eingangsparameter.

**A\_14394-01 - Komponente Autorisierung Vers. - Auslesen Verwaltungsprotokoll**

Die Komponente Autorisierung MUSS beim Aufruf der Operation

`I_Authorization_Management_Insurant::getAuditEvents` dem anhand einer `AuthenticationAssertion` authentifizierten Nutzer die Liste aller zum angefragten

`RecordIdentifier` verfügbaren Verwaltungsprotokolleinträge

gemäß [\[gemSpec\\_DM\\_ePA#A\\_14471\]](#) zurückliefern, wenn der Wert

von `DeviceID::Device` des Aufrufparameters gleich dem Wert

"urn:gematik:fa:phr:1.0:device:device-id" einer für diesen Nutzer

ausgestellten Autorisierungsbestätigung ist. [`<=`]

Damit wird sichergestellt, dass das Auslesen des Verwaltungsprotokolls nur gestattet wird, wenn zuvor eine Autorisierungsbestätigung für diesen Nutzer ausgestellt wurde.

**6.2.4.9 Operationsdefinition****I\_Authorization\_Management\_Insurant::putNotificationInfo****A\_14344-01 - Komponente Autorisierung -****I\_Authorization\_Management\_Insurant::putNotificationInfo**

Die Komponente Autorisierung MUSS die

Operation `I_Authorization_Management_Insurant::putNotificationInfo` gemäß der folgenden Signatur implementieren:

**Tabelle 15: I\_Authorization\_Management\_Insurant::putNotificationInfo - Definition**

<b>Operation</b>	<b>I_Authorization_Management_Insurant::putNotificationInfo</b>		
<b>Beschreibung</b>	Mit dieser Operation kann ein authentifizierter Versicherter bzw. ein berechtigter Vertreter seine im Benachrichtigungskanal hinterlegte Adresse aktualisieren.		
<b>Formatvorgaben</b>	Die Definition der Ein- und Ausgabeparameter erfolgt in [AuthorizationService.xsd]. Diejenigen Parameter, die im XSD-Schema optional gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.		
<b>Eingangsparameter</b>			
<b>Name</b>	<b>Beschreibung</b>	<b>Typ</b>	<b>opt.</b>
<b>AuthenticationAssertion</b>	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-

<b>RecordIdentifier</b>	Der <code>RecordIdentifier</code> referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	<code>RecordIdentifierType</code>	-
<b>DeviceID</b>	Die <code>DeviceID</code> enthält die Geräteerkennung eines vom Nutzer verwendeten Gerätes.	<code>DeviceIdType</code>	-
<b>NewNotificationInfo</b>	<code>NewNotificationInfo</code> beinhaltet die neue Benachrichtigungsadresse, die für den authentifizierten Nutzer gespeichert werden soll.	<code>String</code>	-
<b>Fehlermeldungen</b>			
<b>Name</b>	<b>Fehlertext</b>	<b>Details</b>	
<b>TECHNICAL_ERROR</b>	Zufallszahl		
<b>SYNTAX_ERROR</b>	Fehlerhafte Aufrufparameter	Es wurde ein fehlerhafter Aufrufparameter übergeben.	
<b>DEVICE_UNKNOWN</b>	generierte <code>phr:DeviceID::Device</code>	Das vom Nutzer verwendete Gerät des Versicherten ist nicht bekannt und muss freigeschaltet werden.	
<b>ACCESS_DENIED</b>	Zugriff verweigert	Die Operation ist mit den angegebenen Parametern nicht zulässig.	

1345

1346 [**<=**]

1347

1348



#### 6.2.4.10 Umsetzung

##### **I\_Authorization\_Management\_Insurant::putNotificationInfo**

Die folgenden Anforderungen beschreiben die Umsetzung der Operation `I_Authorization_Management_Insurant::putNotificationInfo`. Dabei gelten die übergreifenden Festlegungen zur Prüfung der Eingangsparameter.

##### **A\_14715 - Komponente Autorisierung Vers. - Aktualisierung Benachrichtigungsadresse**

Die Komponente Autorisierung MUSS bei Aufruf der Operation `I_Authorization_Management_Insurant::putNotificationInfo` den Wert des Parameters `NotificationInfoRepresentative` als Benachrichtigungsadresse des in der `AuthenticationAssertion` benannten Nutzers für den hinterlegten `AuthorizationKey` des Nutzers (`subject-id` der `AuthenticationAssertion` == `ActorID` des `AuthorizationKey`) speichern. [`<=`]

##### **A\_14716 - Komponente Autorisierung Vers. - E-Mail-Format**

Die Komponente Autorisierung MUSS die Operation `I_Authorization_Management_Insurant::putNotificationInfo` mit dem Fehler `SYNTAX_ERROR` abbrechen, wenn der Parameter `NewNotificationInfo` nicht gemäß [\[RFC-5322\]](#) formatiert ist. [`<=`]

Mit dieser Funktion kann ein Versicherter oder ein berechtigter Vertreter seine persönliche Benachrichtigungsadresse zur Gerätefreischaltung ändern. Sowohl für Versicherte als auch deren berechnigte Vertreter sind vor deren jeweiligem Zugriff Benachrichtigungsadressen vorhanden, da diese Operation ohne Gerätefreischaltung über ihre Adresse nicht aufrufbar ist.

Für Versicherte wird die Benachrichtigungsadresse initial im Rahmen der Kontoeröffnung hinterlegt. Für Vertreter erfolgt die initiale Hinterlegung der Benachrichtigungsadresse durch den Versicherten mittels `I_Authorization_Management_Insurant::putAuthorizationKey` während der Vergabe der Zugriffsberechtigung.

#### 6.2.4.11 Operationsdefinition

##### **I\_Authorization\_Management\_Insurant::getAuthorizationList**

##### **A\_17113-01 - Komponente Autorisierung -**

##### **I\_Authorization\_Management\_Insurant::getAuthorizationList**

Die Komponente Autorisierung MUSS die Operation `I_Authorization_Management_Insurant::getAuthorizationList` gemäß der folgenden Signatur implementieren:

**Tabelle 16: I\_Authorization\_Management\_Insurant::getAuthorizationList - Definition**

Operation	<code>I_Authorization_Management_Insurant::getAuthorizationList</code>
<b>Beschreibung</b>	Die Operation liefert eine Liste aller <code>AuthorizationKeys</code> eines Kontos im Aktensystems, als Liste aller Berechtigten in einem Aktenkonto.
<b>Formatvorgaben</b>	Die Definition der Ein- und Ausgabeparameter erfolgt in <code>[AuthorizationService.xsd]</code> . Diejenigen Parameter, die im XSD-Schema optional

gekennzeichnet, aber hier nicht aufgeführt sind, werden in der Operation an dieser Schnittstelle nicht verwendet.			
<b>Eingangsparameter</b>			
Name	Beschreibung	Typ	opt.
AuthenticationAssertion	Die AuthenticationAssertion ist eine von einem Identity Provider ausgestellte Authentifizierungsbestätigung für einen Nutzer.	SAML Assertion im SOAP-Header des Requests	-
RecordIdentifier	Der RecordIdentifier referenziert ein konkretes Aktenkonto eines Versicherten bei einem Anbieter. Mit diesem wird der Datensatz der Autorisierung in der Komponente Autorisierung für den anfragenden Nutzer lokalisiert.	RecordIdentifierType	-
DeviceID	Die DeviceID enthält die Geräteerkennung eines vom Nutzer verwendeten Gerätes.	DeviceIdType	-
<b>Ausgangsparameter</b>			
Name	Beschreibung	Typ	opt.
AuthorizationKeyList	Liste der AuthorizationKeys des per RecordIdentifier identifizierten Kontos.	AuthorizationKeyType[0..*]	-
<b>Fehlermeldungen</b>			
Name	Fehlertext	Details	
TECHNICAL_ERROR	Zufallszahl		
DEVICE_UNKNOWN	generierte phr:DeviceID::Device	Das vom Nutzer verwendete Gerät des Versicherten ist nicht bekannt und muss freigeschaltet werden.	

ACCESS_DENIED	Zugriff verweigert	Die Operation ist mit den angegebenen Parametern nicht zulässig.
---------------	--------------------	------------------------------------------------------------------

1386  
1387  
1388

[<=]

#### 6.2.4.12 Umsetzung

##### I\_Authorization\_Management\_Insurant::getAuthorizationList

##### A\_17115 - Komponente Autorisierung Vers. - Berechtigung für Berechtigungsliste

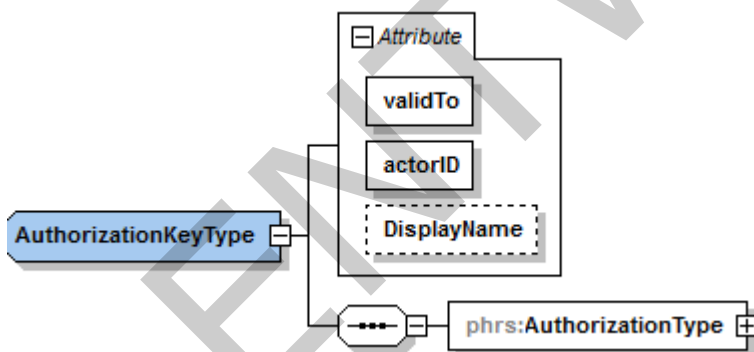
Die Komponente Autorisierung MUSS bei Aufruf der Operation I\_Authorization\_Management\_Insurant::getAuthorizationList prüfen, ob für den in der AuthenticationAssertion benannten User ein AuthorizationKey in der Keychain der mittels RecordIdentifier benannten Akte vorhanden ist (subject-id == ActorID) und andernfalls die Operation mit ACCESS\_DENIED abbrechen.

[<=]

##### A\_17114 - Komponente Autorisierung Vers. - Abfrage Berechtigungsliste

Die Komponente Autorisierung MUSS bei Aufruf der Operation I\_Authorization\_Management\_Insurant::getAuthorizationList die Liste aller AuthorizationKey in der KeyChain der im RecordIdentifier benannten Akte mit Ausnahme des AuthorizationKey des Eigentümers der Akte (für alle zurückgegebenen AuthorizationKey MUSS gelten: ActorID != OwnerKVNR) in der folgenden Struktur zurückgeben

1406



1407  
1408  
1409  
1410

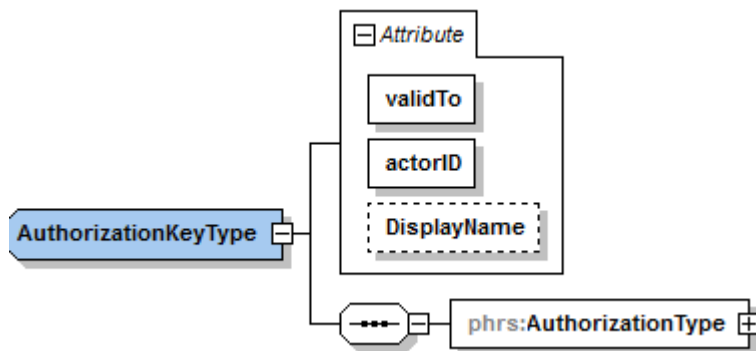
Das heißt, in der Rückgabe an den Aufrufenden werden alle relevanten AuthorizationKeys jeweils ohne das Element EncryptedKeyContainer zurückgegeben.

[<=]

##### A\_17114-01 - Komponente Autorisierung Vers. - Abfrage Berechtigungsliste

Die Komponente Autorisierung MUSS bei Aufruf der Operation I\_Authorization\_Management\_Insurant::getAuthorizationList die Liste aller AuthorizationKey in der KeyChain der im RecordIdentifier benannten Akte mit Ausnahme des AuthorizationKey des Eigentümers der Akte (für alle zurückgegebenen AuthorizationKey MUSS gelten: ActorID != OwnerKVNR) in der folgenden Struktur zurückgeben

1418



Die Elemente Ciphertext und AssociatedData innerhalb des Elements EncryptedKeyContainer MÜSSEN mit einem Leer-String belegt werden. [ <= ]

### 6.3 Berechtigungstypen der Autorisierung

Der Berechtigungstyp (*AuthorizationType*) steuert den Zugriff auf weitere Ressourcen für einen authentisierten Nutzer. Der Berechtigungstyp wird beim Hinzufügen des Schlüsselmaterials für einen Nutzer in der Autorisierungskomponente hinterlegt.

Es wird zwischen drei Typen unterschieden, die in der folgenden Tabelle beschrieben sind:

**Tabelle 17: Berechtigungstypen für AuthorizationType**

<i>AuthorizationType</i>	Beschreibung
DOCUMENT_AUTHORIZATION (Dokumentenautorisierung)	Es wird für einen authentisierten Nutzer eine Autorisierungsbestätigung ausgestellt, die für den Zugang zur Dokumentenverwaltung notwendig ist.
RECOVERY_AUTHORIZATION (Umschlüsselungsberechtigung)	Es wird einem authentisierten Nutzer die Verwendung des hinterlegten Schlüssels zur lokalen Umschlüsselung gestattet. Mit dieser Autorisierungsbestätigung ist kein Zugriff auf die Komponente Dokumentenverwaltung möglich
ACCOUNT_AUTHORIZATION (Betreiberwechselautorisierung)	Es wird dem authentisierten Nutzer eine Autorisierungsbestätigung ausgestellt, mit dem in der Komponente Dokumentenverwaltung nur ein eingeschränkter Zugriff auf Daten des Versicherten möglich ist.

### 6.4 Hardware-Merkmal der Komponente Autorisierung

Es müssen die privaten Schlüssel der Ausstelleridentität für Autorisierungsbestätigungen sowie der TLS-Server-Identität sicher gespeichert werden.

1435 **A\_14366 - Komponente Autorisierung - Verwendung eines HSM**  
 1436 Die Komponente Autorisierung MUSS das private Schlüsselmaterial der Ausstelleridentität  
 1437 C.FD.SIG und der TLS-Server-Identität C.FD.TLS-S in einem HSM speichern.[<=]

## 1438 6.5 Geräteverwaltung

1439 Die Komponente Autorisierung setzt zusätzlich zur kryptografischen Autorisierung eine  
 1440 Geräteautorisierung um. Dazu wird bei Zugriffen aus der Umgebung des Versicherten  
 1441 (über das Internet) geprüft, ob ein Versicherter bzw. berechtigter Vertreter ein  
 1442 bekanntes Gerät für den Zugriff nutzt. Ist das Gerät unbekannt, wird ein  
 1443 Freischaltprozess über einen separaten Benachrichtigungskanal gestartet. Die Erkennung  
 1444 erfolgt auf Basis einer im Gerät des Versicherten gebildeten DeviceID, welche in den  
 1445 Operationsaufrufen mitgeschickt werden muss. Die DeviceId als DeviceIdType gemäß  
 1446 [PHR\_Common.xsd] enthält neben der eigentlichen Geräteerkennung Device, welche für  
 1447 den Abgleich bekannter Geräte verwendet wird, einen DisplayName, der dem Nutzer die  
 1448 Verwaltung seiner genutzten Geräte erleichtert.

1449 Die Umsetzung erfolgt in der Komponente Autorisierung, da eine vorgelagerte  
 1450 zustandslose Komponente der Authentifizierung von Nutzern, ggfs. nicht über einen  
 1451 Speicher zur Verwaltung von Gerätekennungen je Benutzerkonto verfügt bzw. dieser für  
 1452 diesen Zweck erst geschaffen werden müsste.

1453 Die Prüfung auf ein autorisiertes Gerät erfolgt vor der Herausgabe des in der  
 1454 Komponente Autorisierung gespeicherten Schlüsselmaterials.

1455 Für die Benachrichtigung mit anschließender Freischaltung werden E-Mails mit  
 1456 generierten URLs auf generierte HTML-Webseiten verwendet, da E-Mail aus Usability-  
 1457 Sicht am komfortabelsten erscheint und diese Methoden in verschiedensten Diensten im  
 1458 Internet etabliert und den Versicherten sehr wahrscheinlich bekannt sind.

### 1459 6.5.1 Freischaltprozess neuer Geräte

1460 Der Freischaltprozess dient dazu, ein Endgerät des Versicherten in der  
 1461 Komponente Autorisierung zu registrieren. Der folgende Ablauf zeigt informativ einen  
 1462 möglichen Ablauf des Freischaltprozesses.

1463

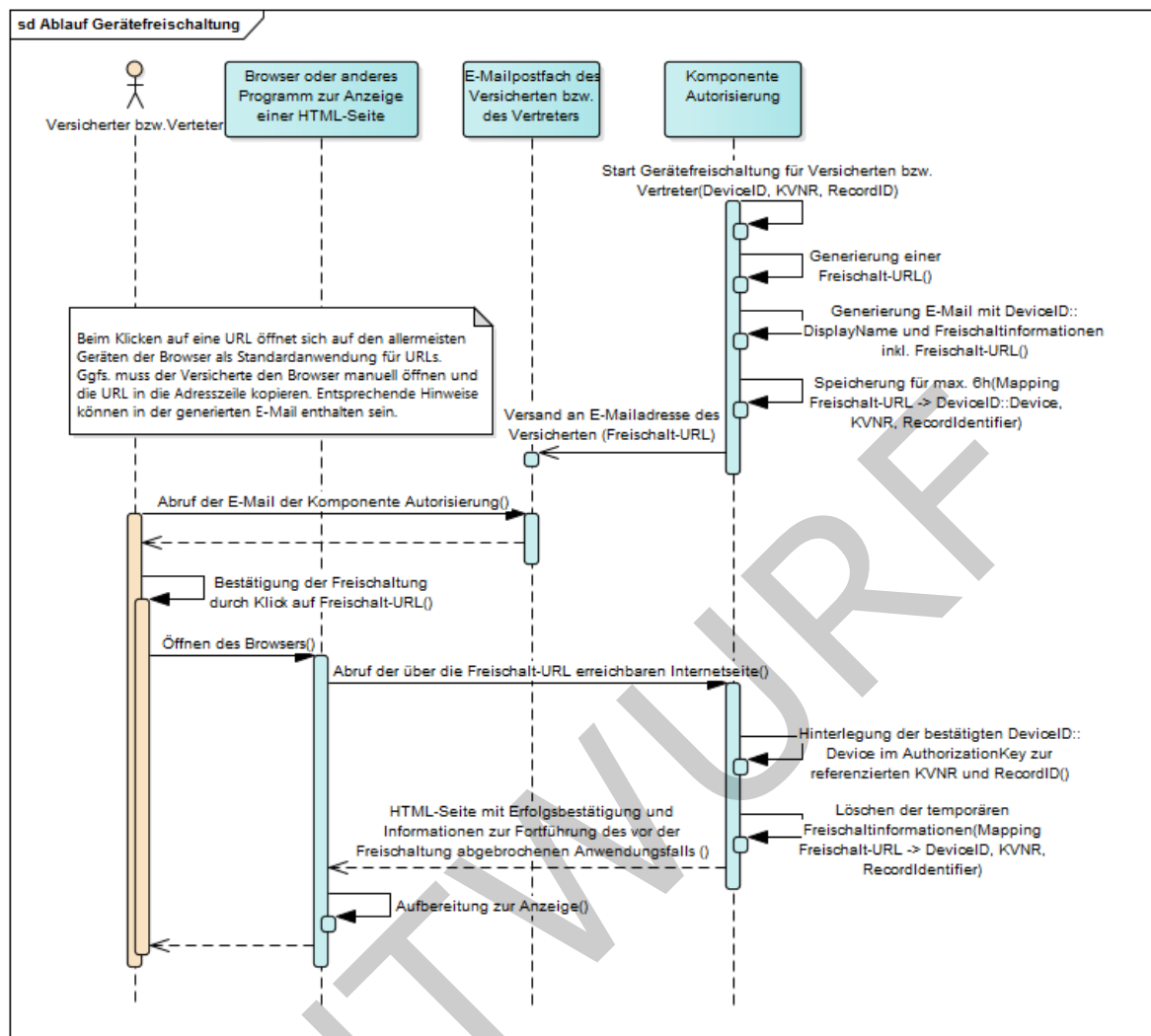


Abbildung 4: Informativer Ablauf des Geräte-Freischaltprozesses

Die Komponente Autorisierung startet den Freischaltprozess für jedes über DeviceID::Device identifizierte Gerät, das für den AuthorizationKey eines per KVNR identifizierten Versicherten bzw. Vertreter zu einer genannten RecordID als unbekannt gilt. D.h. ein vom Vertreter im eigenen Aktenkonto verwendetes Gerät kann dort bereits registriert sein, im Rahmen der Vertretung eines anderen Versicherten kann das gleiche Gerät am Vertretungsschlüssel unbekannt sein. In diesem Fall ist der Freischaltprozess für die Wahrnehmung der Vertretung erforderlich.

Die Komponente Autorisierung generiert zu einem Freischaltprozess einen eindeutigen Link auf Basis von Zufallszahlen und verschickt ihn an die vom Nutzer hinterlegte Benachrichtigungs-E-Mail-Adresse. Durch Klicken auf diesen Link erhält der Versicherte bzw. Vertreter eine Webseite, mit der Bitte um Bestätigung der Freischaltung des genutzten Geräts. Nach Erhalt der Freischaltbestätigung fügt die Komponente Autorisierung das per DeviceID identifizierte Gerät zum AuthorizationKey des Versicherten bzw. Vertreters hinzu.

#### A\_17866 - Komponente Autorisierung - Generierung Device-Kennung für unbekanntes Gerät des Versicherten

Die Komponente Autorisierung MUSS bei Aufruf einer Operation der Schnittstellen I\_Authorization\_Insurant und I\_Authorization\_Management\_Insurant mit einem für den aufrufenden Nutzer im benannten RecordIdentifier unbekanntem Parameter

1485 `phr:DeviceID::Device` eine 256 Bit Zufallszahl (base64-kodiert) mit einer  
1486 Mindestentropie von 120 Bit und Erzeugung gemäß [gemSpec\_Krypt#GS-A\_4367]  
1487 erzeugen, diese als `phr:DeviceID::Device` für den aufrufenden Nutzer im benannten  
1488 RecordIdentifier konfigurieren und den Freischaltprozess gemäß  
1489 [\[gemSpec\\_Autorisierung#A\\_14515\]](#) starten.

1490  
1491 [**<=**]

1492 Mit der Generierung der Device-Kennung auf Basis einer Zufallszahl je Konto ergibt sich,  
1493 dass die Verwendung eines Geräts in verschiedenen Konten (z.B. eigenes Konto +  
1494 Vertretungsberechtigung in einem anderen Konto) zur Erzeugung zweier verschiedener  
1495 Device-IDs führt, die im jeweiligen Aufrufkontext zu verwenden sind.

#### 1496 **A\_17947 - Komponente Autorisierung - Gültigkeitszeitraum und Löschung der** 1497 **Devicekennung**

1498 Die Komponente Autorisierung MUSS jede generierte und in einem Aktenkonto  
1499 gespeicherte Device-Kennung `phr:DeviceID::Device` nach 2 Jahren löschen und darf  
1500 Nutzeranfragen mit dieser Device-Kennung nach diesem Zeitpunkt nicht mehr  
1501 akzeptieren.

1502 [**<=**]

1503 Daraus folgt, dass nach zwei Jahren eine Neuregistrierung des verwendeten Geräts  
1504 erforderlich ist. Ein möglicher Zeitraum der Inaktivität des Geräts ist dabei irrelevant

#### 1505 **A\_14515 - Komponente Autorisierung - Freischaltprozess Freischalt-URL**

1506 Die Komponente Autorisierung MUSS im Freischaltprozess eine Freischalt-URL erzeugen,  
1507 die einzig aus dem FQDN der Komponente Autorisierung und einer Zufallszahl (base64-  
1508 kodiert) mit mindestens 120 Bit Entropie und Erzeugung gemäß [gemSpec\_Krypt#GS-  
1509 A\_4367] besteht und diese Freischalt-URL an die E-Mail-Adresse am `AuthorizationKey`  
1510 des via KVNRR einer `AuthenticationAssertion` referenzierten Nutzers zum angefragten  
1511 `RecordIdentifier` verschicken. [**<=**]

#### 1512 **A\_14518 - Komponente Autorisierung - Freischaltprozess Freischalt-URL** 1513 **Transportsicherheit**

1514 Die Komponente Autorisierung MUSS in der generierten Freischalt-URL das https-  
1515 Protokoll verwenden.

1516 [**<=**]

#### 1517 **A\_14520 - Komponente Autorisierung - Freischaltprozess Webseite zu** 1518 **Freischalt-URL**

1519 Die Komponente Autorisierung MUSS bei Aufruf einer generierten Freischalt-URL durch  
1520 einen Versicherten bzw. Vertreter mit einer HTML-Seite mit folgendem Inhalt über den  
1521 transportverschlüsselten Kanal der https-Freischalt-URL antworten:

- 1522 • `DeviceID::DisplayName` des freizuschaltenden Geräts
- 1523 • Zeitpunkt des Starts des Freischaltprozesses
- 1524 • `RecordIdentifier`
- 1525 • Bestätigungslink (`submit`) zur endgültigen Freischaltung des Geräts

1526 [**<=**]

#### 1527 **A\_14521 - Komponente Autorisierung - Freischaltprozess DeviceID hinzufügen**

1528 Die Komponente Autorisierung MUSS bei Abruf des Bestätigungslinks eines aktiven  
1529 Freischaltprozesses die generierte `phr:DeviceID::Device` zum `AuthorizationKey` eines  
1530 `RecordIdentifier`s des über KVNRR einer `AuthenticationAssertion` identifizierten  
1531 Versicherten bzw. Vertreters hinzufügen und den Freischaltprozess für den Vorgang zu



1532 DeviceID, KVNR und RecordIdentifier beenden.  
1533 [ $\leq$ ]

1534 **A\_14522 - Komponente Autorisierung - Freischaltprozess beenden**

1535 Die Komponente Autorisierung MUSS den Vorgang eines Freischaltprozesses zu  
1536 DeviceID, KVNR und RecordIdentifier nach 6 Stunden Wartezeit beenden. [ $\leq$ ]

1537 **A\_14523 - Komponente Autorisierung - Freischaltprozess Löschen nach**  
1538 **Beendigung**

1539 Die Komponente Autorisierung MUSS beim Beenden des Vorgangs eines  
1540 Freischaltprozesses die generierte Freischalt-URL und alle dazugehörigen temporären  
1541 Daten löschen. [ $\leq$ ]  
1542

1543 **6.5.2 Geräteadministration**

1544 Mit der Geräteadministration wird dem Nutzer die Möglichkeit gegeben, seine Endgeräte  
1545 zu verwalten.

1546 **A\_14364 - Komponente Autorisierung - Geräteverwaltung**

1547 Die Komponente Autorisierung MUSS dem authentifizierten Versicherten über eine Web-  
1548 Schnittstelle folgende Funktionen zur Verfügung stellen:

- 1549 • Sperren von registrierten Geräten, so dass ein Zugriff über diese Geräte bis zur  
1550 Entsperrung nicht möglich ist,
- 1551 • Entsperrn von gesperrten Geräten, so dass ein Zugriff über diese Geräte möglich  
1552 ist,
- 1553 • Deregistrieren von Geräten, so dass ein Zugriff über diese Geräte erst nach  
1554 erneuter erfolgreicher Freischaltung möglich ist sowie
- 1555 • das Vergeben einer alternativen Bezeichnung für ein registriertes Gerät.

1556 [ $\leq$ ]

1557 **A\_15438 - Komponente Autorisierung - Keine negative Beeinflussung des**  
1558 **Aktensystems durch die Geräteverwaltung**

1559 Die Komponente Autorisierung MUSS sicherstellen, dass das Web-Frontend zur  
1560 Geräteverwaltung der Komponente Autorisierung so geschützt wird, dass keine negative  
1561 Beeinflussung des Aktensystems über diese Schnittstelle möglich ist. [ $\leq$ ]

1562 **A\_14595 - Komponente Autorisierung - Pflegeprozess Geräteverwaltung**

1563 Die Komponente Autorisierung MUSS die interne Liste aller bekannten Geräte derart  
1564 pflegen, dass ein Gerät nach spätestens einem Jahr nach der letzten Nutzung des  
1565 Gerätes automatisch aus der Liste der registrierten Geräte gelöscht wird, und bei  
1566 anschließender Verwendung durch einen Versicherten als unbekanntes Gerät über den  
1567 Freischaltprozess neu freizuschalten ist. [ $\leq$ ]

1568 **A\_15551 - Komponente Autorisierung - Deregistrierung in fremden Konten**

1569 Die Komponente Autorisierung MUSS sicherstellen, dass der Versicherte nur diejenigen  
1570 registrierten Geräte verwalten kann, die der Versicherte oder ein Vertreter in seinem  
1571 Konto verwendet. Eine Deregistrierung eines Gerätes in einem Konto DARF NICHT  
1572 automatisch zu einer Deregistrierung in einem anderen Konto führen (z.B. im Konto  
1573 eines anderen Versicherten, für das der Versicherte Vertretungsrechte besitzt). [ $\leq$ ]

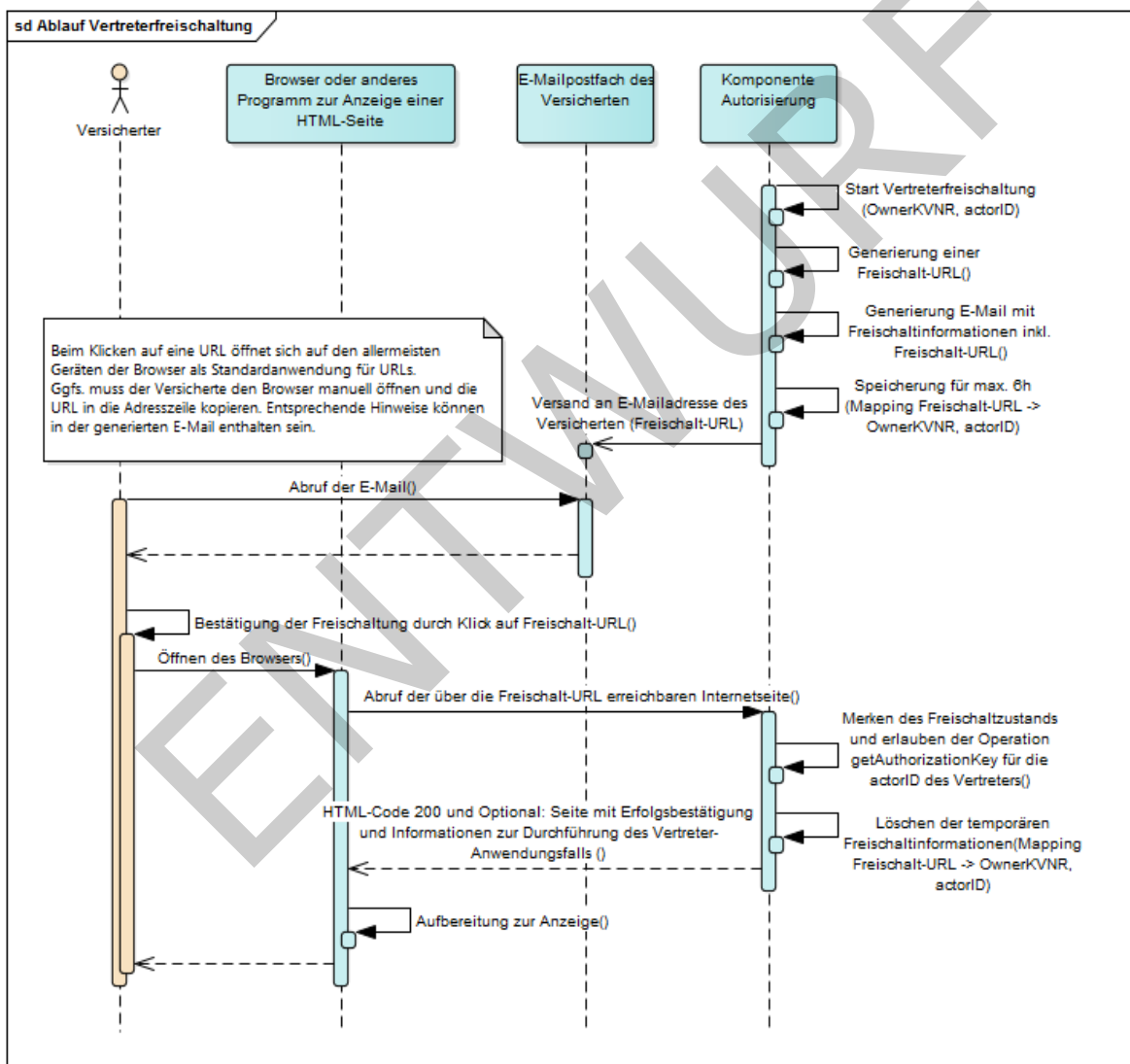
1574 **A\_15755-01 - Komponente Autorisierung - Protokollierung Geräteverwaltung**

1575 Die Komponente Autorisierung MUSS alle Vorgänge der Geräteverwaltung im  
1576 Verwaltungsprotokoll des Versicherten mit PHR-475 protokollieren. [ $\leq$ ]



## 6.6 Freischaltprozess Vertretereinrichtung

Die Komponente Autorisierung führt eine zusätzliche Autorisierung durch den Versicherten bei Einrichtung einer Vertretung für einen Vertreter durch. Der Versicherte wird aufgefordert, auf einen Link in einer E-Mail zu klicken, um die Speicherung eines AuthorizationKey für einen Vertreter zu autorisieren, den er über `I_Authorization_Management_Insurant::putAuthorizationKey` für diesen Vertreter hinterlegt. Die E-Mail mit dem Link zur Freischaltung wird an die E-Mail-Adresse des Versicherten geschickt, die auch für die Gerätefreischaltung des Versicherten verwendet wurde. Der folgende Ablauf zeigt informativ einen möglichen Ablauf des Freischaltprozesses.



**Abbildung 5: Informativer Ablauf des Freischaltprozesses für Vertretung**

Die Komponente Autorisierung startet den Freischaltprozess wenn der Versicherte mittels `I_Authorization_Management_Insurant::putAuthorizationKey` für einen konkreten mittels KVN identifizierten Vertreter (als ActorID am AuthorizationKey) erstmalig eine Berechtigung hinterlegen möchte. Die Operation wird zunächst erfolgreich abgeschlossen, sofern kein fachlicher oder technischer Fehler dies verhindert. Dem

1595 Vertreter wird der Zugriff auf diesen Schlüssel jedoch solange verwehrt, wie der  
1596 Versicherte noch nicht auf einen Freischaltlink in einer generierten Freischalt-E-Mail  
1597 klickt. Die Komponente Autorisierung generiert zum Freischaltprozess der Vertretung  
1598 einen eindeutigen Link auf Basis von Zufallszahlen und verschickt ihn an die vom  
1599 Versicherten hinterlegte Benachrichtigungs-E-Mail-Adresse.

1600 Durch Klicken auf diesen Link signalisiert der Versicherte der Komponente Autorisierung,  
1601 dass die Hinterlegung eines AuthorizationKey für die KVNR d.h. ActorID des Vertreters  
1602 rechtmäßig ist. Die Komponente Autorisierung speichert diesen Freischaltzustand für die  
1603 ActorID des Vertreters und teilt dem Versicherten über die mittels Freischaltlink  
1604 abgerufene Webseite mit, dass der UseCase des Schlüsselabrufs mittels  
1605 I\_Authorization\_Insurant::getAuthorizationKey durch den Vertreter nun autorisiert  
1606 ist. Der Vertreter kann nun den hinterlegten Schlüssel abrufen und eine Vertretung  
1607 wahrnehmen.

1608

#### 1609 **A\_17672 - Komponente Autorisierung - Freischaltprozess Vertretung Freischalt-** 1610 **URL**

1611 Die Komponente Autorisierung MUSS im Freischaltprozess Vertreter Einrichtung eine  
1612 Freischalt-URL erzeugen, die einzig aus dem FQDN der Komponente Autorisierung und  
1613 einer Zufallszahl (base64-kodiert) mit mindestens 120 Bit Entropie und Erzeugung  
1614 gemäß [gemSpec\_Krypt#GS-A\_4367] besteht und diese Freischalt-URL an die E-Mail-  
1615 Adresse des via OwnerKVNR referenzierten Versicherten verschicken.

1616 [**<=**]

#### 1617 **A\_17673 - Komponente Autorisierung - Freischaltprozess Vertretung Freischalt-** 1618 **URL Transportsicherheit**

1619 Die Komponente Autorisierung MUSS in der generierten Freischalt-URL das https-  
1620 Protokoll verwenden.

1621 [**<=**]

#### 1622 **A\_17674 - Komponente Autorisierung - Freischaltprozess Vertretung** 1623 **getAuthorizationKey erlauben**

1624 Die Komponente Autorisierung MUSS bei Abruf des Bestätigungslinks eines aktiven  
1625 Freischaltprozesses zur OwnerKVNR und ActorId des zukünftigen Vertreters die  
1626 Operation I\_Authorization\_Insurant::getAuthorizationKey für das Abrufen eines  
1627 AuthorizationKey durch den Vertreter (ActorId = KVNR des zukünftigen Vertreters)  
1628 erlauben und den Freischaltprozess für den Vorgang zu OwnerKVNR und ActorID  
1629 beenden.

1630 [**<=**]

1631 Damit wird die Operation I\_Authorization\_Insurant::getAuthorizationKey bei  
1632 zukünftigen Aufrufen durch den Vertreter für die freigeschaltete ActorID nicht mehr mit  
1633 Fehler REPRESENTATIVE\_PENDING abgebrochen.

#### 1634 **A\_17677 - Komponente Autorisierung - Freischaltprozess Vertretung** 1635 **Information**

1636 Die Komponente Autorisierung KANN in der HTTP-Response zum URL-Aufruf der  
1637 Vertreterfreischaltung eine Meldung über die erfolgreiche Freischaltung an den  
1638 aufrufenden Versicherten zurückgeben.

1639 [**<=**]

#### 1640 **A\_17675 - Komponente Autorisierung - Freischaltprozess Vertretung beenden**

1641 Die Komponente Autorisierung MUSS den Vorgang eines Freischaltprozesses Vertretung  
1642 zur OwnerKVNR und ActorID nach 6 Stunden Wartezeit beenden.

1643 [**<=**]

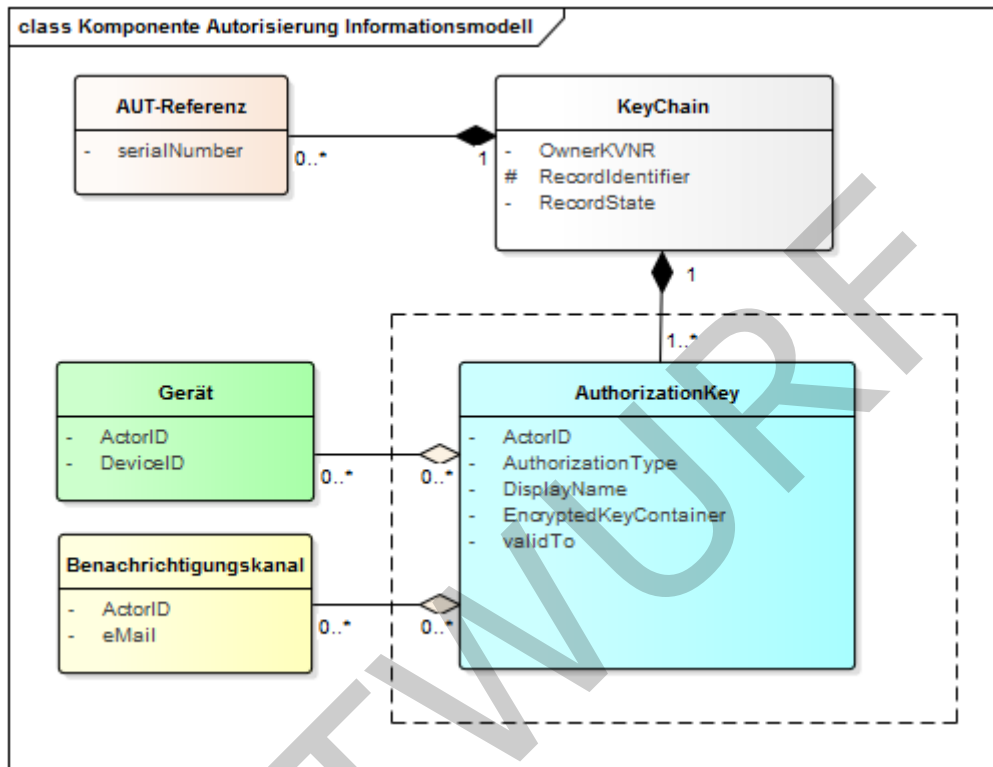
1644 **A\_17676 - Komponente Autorisierung - Freischaltprozess Vertretung Löschen**  
1645 **nach Beendigung**  
1646 Die Komponente Autorisierung MUSS beim Beenden des Vorgangs eines  
1647 Freischaltprozesses die generierte Freischalt-URL und alle dazugehörigen temporären  
1648 Daten löschen.  
1649 [**<=**]

ENTWURF

1650

## 7 Informationsmodell

1651 Das folgende Informationsmodell der Autorisierung gibt eine Übersicht über die  
 1652 verwendeten Objekte mit ihren Eigenschaften und Beziehungen zueinander.



1653

1654

**Abbildung 6: Informationsmodell der intern verwalteten Daten**

1655 Das blau dargestellte Element bildet den verwalteten `AuthorizationKey`, der vom  
 1656 Versicherten für jeden berechtigten Nutzer in der Komponente Autorisierung hinterlegt  
 1657 wird, das Element `EncryptedKeyContainer` enthält dabei das mit dem  
 1658 Empfängerschlüssel individuell verschlüsselte Schlüsselmaterial der Akte (Akten- und  
 1659 Kontextschlüssel). Die Summe aller `AuthorizationKeys` zu einem über den  
 1660 `RecordIdentifier` identifizierten Konto eines über die `OwnerKVNR` identifizierten  
 1661 Versicherten bildet das logische Element des "Schlüsselrings" `KeyChain`. Zu einem über  
 1662 `ActorID` identifizierten Nutzer wird eine Liste autorisierter Geräte (grün dargestellt)  
 1663 geführt, die bei Zugriffen aus der Umgebung des Versicherten die Zulässigkeit des  
 1664 genutzten Geräts prüfen lässt. Für den Fall eines unbekannten und somit nicht in der  
 1665 Liste zulässiger Geräte enthaltenen Geräts wird ein Freischaltprozess über einen  
 1666 Benachrichtigungskanal gestartet. Die Zuordnung der Benachrichtigungsadressen zum  
 1667 jeweiligen Nutzer ist im Bild gelb dargestellt.

1668 Für Versicherte und deren Vertreter wird der unveränderliche Teil der `KVNR`  
 1669 (VersichertenID) der eGK als `ActorID` verwendet. Für den Versicherten wird genau diese  
 1670 ID auch als `OwnerKVNR` genutzt, um den jeweiligen Versicherten als Eigentümer einer  
 1671 Akte zu identifizieren. Für Leistungserbringerinstitutionen und Kostenträger wird die  
 1672 Telematik-ID als `ActorID` verwendet. Für Leistungserbringerinstitutionen sowie für die  
 1673 Kostenträger wird keine Liste autorisierter Geräte und keine Liste von  
 1674 Benachrichtigungskanälen geführt. Die Eigenschaft `validTo` bezeichnet ein  
 1675 Gültigkeitsende-Datum, an welchem ein `AuthorizationKey` systemseitig automatisch

gelöscht wird. Für den Versicherten als Eigentümer der Akte wird ein technisches Ende-Datum gleichbedeutend mit "unendlich" automatisch gesetzt. Für alle anderen AuthorizationKeys wird das Datum clientseitig belegt und definiert das Ende der vom Versicherten vergebenen Berechtigung. Mit dem optionalen `Displayname` je AuthorizationKey kann vom Versicherten ein lesbarer Name für eine Berechtigung vergeben werden, auf LE-Seite und den Abruf durch Kostenträger wird das Feld vollständig ignoriert.

Mittels der Angabe des `RecordIdentifiers` und der `ActorID` (*Telematik-ID/KVNR*) kann der zugehörige `AuthorizationKey` eines Berechtigten gefunden werden. Der `AuthorizationKey` enthält eine Liste verschlüsselter Akten- und Kontextschlüssel.

Das Element AUT-Referenz speichert in einer `WhiteList` die `serialNumber` der zur Authentisierung durch Versicherte in einer Akte verwendeten AUT- bzw. AUT\_ALT-Zertifikate. Über diese Liste wird die Verwendung einer bisher unbekannten kryptografischen Identität erkannt und der Versicherte bzw. der Vertreter über den Benachrichtigungskanal informiert.

## 7.1 Namensräume

Für die Schnittstellen der Komponente Autorisierung werden die in der folgenden Tabelle definierten XML-Präfixe verwendet, um den Namensraum des XML-Dokumentes zu beschreiben.

**Tabelle 18: Namensräume**

Präfix	Namensraum
<code>xmlns:phrs</code>	<code>http://ws.gematik.de/fd/phrs/AuthorizationService/v1.0</code>
<code>xmlns:SAML</code>	<code>urn:oasis:names:tc:SAML:2.0:assertion</code>
<code>xmlns:ds</code>	<code>http://www.w3.org/2000/09/xmldsig#</code>
<code>xmlns:xenc</code>	<code>http://www.w3.org/2001/04/xmlenc#</code>

## 7.2 SAML-Profil und Tokeninhalte

In diesem Abschnitt werden die Inhalte der auszustellenden `AuthorizationAssertion` festgelegt. Eine `AuthorizationAssertion` wird für einen mittels `AuthenticationAssertion` authentifizierten Nutzer ausgestellt. Aus dessen `AuthenticationAssertion` werden identifizierende Attribute in die `AuthorizationAssertion` übernommen.

### A\_14491-01 - Komponente Autorisierung - Inhalte `AuthorizationAssertion`

Die Komponente Autorisierung MUSS Autorisierungsbestätigungen als SAML2-Assertion gemäß den Festlegungen der folgenden Tabelle ausstellen:

1705 **Tabelle 19: Inhalte Autorisierungsbestätigung**

Assertion Element	Usage Convention	Beschreibung
Issuer	[FQDN des ePA-Aktensystems der TI] + "/authz"	Aussteller des Tokens
Signature	[nonQES-Signatur des SAML-Tokens]	nonQES-Signatur des SAML-Tokens gemäß [SAML 2.0], die mit dem privaten Schlüssel der Ausstelleridentität C.FD.SIG der Komponente Autorisierung gemäß [ gemSpec_Krypt#A_17206] erstellt wird. Das Element <code>ds:Signature/ds:KeyInfo/ds:X509Data/ds:X509Certificate</code> muss das zugehörige C.FD.SIG Zertifikat enthalten
Subject		
NameID	[SubjectDN der SMC-B] oder [SubjectDN der eGK]	wird übernommen aus der übergebenen <i>AuthenticationAssertion</i>
SubjectConfirmation		
@Method	urn:oasis:names:tc:SAML:2.0:cm:bearer	Protokoll zur Authentisierung
Conditions		
@NotBefore	[Systemzeit der Komponente Autorisierung]	Zeitpunkt, ab wann die Assertion nutzbar ist.
@NotOnOrAfter	[Systemzeit der Komponente Autorisierung + 15 Minuten]	Zeitpunkt, zu dem die Gültigkeit der Assertion endet.
AudienceRestriction		Liste der Server, für die das Token ausgestellt wird.
Audience	[FQDN des ePA-Aktensystems gemäß gemSpec_Aktensystem A_14128]	<ul style="list-style-type: none"> <li>TI-seitiger FQDN für Aufrufe an den Schnittstellen I_Authorization und</li> </ul>

			I_Authorization_Management <ul style="list-style-type: none"> <li>Internet-seitiger FQDN für Aufrufe der Schnittstellen I_Authorization_Insurant und I_Authorization_Management_Insurant</li> </ul>
AuthnStatement			
	@AuthnInstant	[Systemzeit der Komponente Autorisierung]	Systemzeitpunkt bei Erstellung des Tokens
AuthzDecisionStatement			
	@Ressource	[Telematik-ID] oder [10-stelliger, unveränderlicher Teil der KVNR]	wird übernommen aus der AuthenticationAssertion
	@Decision	Permit	
	Action	[AuthorizationType]	String gemäß der Autorisierungsentscheidung über den authentifizierten Nutzer
	@Namespace	"http://ws.gematik.de/fa/phr/v1.0"	
AttributeStatement			
	Attribute		
	@Name	Resource ID "urn:oasis:names:tc:xacml:1.0:resource:resource-id"	
	AttributeValue	[RecordIdentifier]	RecordIdentifier der Akte, für die eine Autorisierungsbestätigung für den Nutzer ausgestellt wird.
	Attribute		

	@Name	Gerätekennung "urn:gematik:fa:phr:1.0:device:device-id"	Nur bei mittels ActorID authentifizierten Versicherten, bei Abruf durch Leistungserbringer und Kostenträger entfällt dieses Attribut.
	AttributeValue	[DeviceID::Device]	Die DeviceID::Device ist über die ActorID des AuthorizationKey referenziert, der über die KVN der Versicherten einer übergebenen AuthenticationAssertion gefunden wird.
	Attribute		
	@Name	Zustand des Kontos "urn:gematik:fa:phr:1.0:status:status-id"	
	AttributeValue	[RecordState]	Wert der Eigenschaft RecordState der KeyChain des via RecordIdentifier benannten Kontos.
	Attribute		
	@Name	<b>VersichertenID</b> "urn:gematik:subject:subject-id" oder <b>Telematik-ID</b> "urn:gematik:subject:organization-id"	Benutzerkennung für den die AuthorizationAssertion ausgestellt wird.
	AttributeValue	[Telematik-ID] oder [10-stelliger, unveränderlicher Teil der KVN]	wird übernommen aus der AuthenticationAssertion

1706

1707 [ $\leq$ ]



1708

---

## 8 Verteilungssicht

---

1709

Eine Darstellung der hardwareseitigen Verteilung des Produkttyps bzw. seiner Teilsysteme und der Einbettung in die physikalische Umgebung wird nicht benötigt.

1710

ENTWURF

## 9 Anhang A – Verzeichnisse

### 9.1 Abkürzungen

Kürzel	Erläuterung
SAML	Security Assertion Markup Language
WS	Web Services
PKCS	Public-Key Cryptography Standards
ePA-FdV	ePA-Frontend des Versicherten, welches das ePA-Modul FdV inkludiert
IHE	Integrating the Healthcare Enterprise
WSDL	Web Services Description Language
KVNR	Krankenversichertennummer

### 9.2 Glossar

Begriff	Erläuterung
HSM	Hardware Security Module, Gerät zur sicheren Speicherung kryptografischen Schlüsselmaterials
ePA-Modul FdV	Modul der dezentralen ePA-Fachlogik zur Nutzung durch den Versicherten in einem ePA-Frontend des Versicherten

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

### 9.3 Abbildungsverzeichnis

Abbildung 1: Anwendungsfälle der Schlüsselverwaltung nach Umgebung .....	11
Abbildung 2: Komponente Autorisierung, benachbarte Komponenten und Produkttypen .....	13
Abbildung 3: GERROR-Struktur zur Rückgabe einer Fehlermeldung .....	23
Abbildung 4: Informativer Ablauf des Geräte-Freischaltprozesses .....	62

1722	<a href="#">Abbildung 5: Informativer Ablauf des Freischaltprozesses für Vertretung .....</a>	65
1723	<a href="#">Abbildung 6: Informationsmodell der intern verwalteten Daten .....</a>	68
1724	<a href="#">Abbildung 1: Anwendungsfälle der Schlüsselverwaltung nach Umgebung .....</a>	11
1725	<a href="#">Abbildung 2: Komponente Autorisierung, benachbarte Komponenten und Produkttypen .....</a>	13
1726	<a href="#">Abbildung 3: GERROR-Struktur zur Rückgabe einer Fehlermeldung .....</a>	23
1727	<a href="#">Abbildung 4: Informativer Ablauf des Geräte-Freischaltprozesses .....</a>	62
1728	<a href="#">Abbildung 5: Informativer Ablauf des Freischaltprozesses für Vertretung .....</a>	65
1729	<a href="#">Abbildung 6: Informationsmodell der intern verwalteten Daten .....</a>	68
1730		

## 1731 9.4 Tabellenverzeichnis

1732	<a href="#">Tabelle 1: Anwendungsfälle der Schlüsselverwaltung nach Umgebung .....</a>	12
1733	<a href="#">Tabelle 2: Parameter des Verwaltungsprotokolls .....</a>	22
1734	<a href="#">Tabelle 3: Fehlercodes zu Fehlern gemäß Operationsdefinition .....</a>	23
1735	<a href="#">Tabelle 4: Herstellerspezifische Fehlerdefinition .....</a>	24
1736	<a href="#">Tabelle 5: Schnittstellen der Komponente Autorisierung .....</a>	29
1737	<a href="#">Tabelle 6: I_Authorization::getAuthorizationKey Definition .....</a>	31
1738	<a href="#">Tabelle 7: I_Authorization_Insurant::getAuthorizationKey Definition .....</a>	35
1739	<a href="#">Tabelle 8: I_Authorization_Management::putAuthorizationKey Definition .....</a>	38
1740	<a href="#">Tabelle 9: I_Authorization_Management::checkRecordExists Definition .....</a>	41
1741	<a href="#">Tabelle 10: I_Authorization_Management::getAuthorizationList Definition .....</a>	42
1742	<a href="#">Tabelle 11: I_Authorization_Management_Insurant::putAuthorizationKey Definition .....</a>	44
1743	<a href="#">Tabelle 12: I_Authorization_Management_Insurant::deleteAuthorizationKey Definition .....</a>	48
1744	<a href="#">Tabelle 13: I_Authorization_Management_Insurant::replaceAuthorizationKey Definition .....</a>	51
1745	<a href="#">Tabelle 14: I_Authorization_Management_Insurant::getAuditEvents Definition .....</a>	53
1746	<a href="#">Tabelle 15: I_Authorization_Management_Insurant::putNotificationInfo Definition .....</a>	55
1747	<a href="#">Tabelle 16: I_Authorization_Management_Insurant::getAuthorizationList Definition .....</a>	57
1748	<a href="#">Tabelle 17: Berechtigungstypen für AuthorizationType .....</a>	60
1749	<a href="#">Tabelle 18: Namensräume .....</a>	69
1750	<a href="#">Tabelle 19: Inhalte Autorisierungsbestätigung .....</a>	70
1751	<a href="#">Tabelle 20: Referenzierte Dokumente der gematik .....</a>	76
1752	<a href="#">Tabelle 21: Referenzierte externe Dokumente .....</a>	77
1753	<a href="#">Tabelle 1: Anwendungsfälle der Schlüsselverwaltung nach Umgebung .....</a>	12
1754	<a href="#">Tabelle 2: Parameter des Verwaltungsprotokolls .....</a>	22
1755		
1756		

1757	<a href="#">Tabelle 3: Fehlercodes zu Fehlern gemäß Operationsdefinition .....</a>	23
1758	<a href="#">Tabelle 4: Herstellerspezifische Fehlerdefinition .....</a>	24
1759	<a href="#">Tabelle 5: Schnittstellen der Komponente Autorisierung .....</a>	29
1760	<a href="#">Tabelle 6: I Authorization::getAuthorizationKey Definition .....</a>	31
1761	<a href="#">Tabelle 7: I Authorization Insurant::getAuthorizationKey Definition.....</a>	35
1762	<a href="#">Tabelle 8: I Authorization Management::putAuthorizationKey - Definition .....</a>	38
1763	<a href="#">Tabelle 9: I Authorization Management::checkRecordExists - Definition.....</a>	41
1764	<a href="#">Tabelle 10: I Authorization Management::getAuthorizationList - Definition.....</a>	42
1765	<a href="#">Tabelle 11: I Authorization Management Insurant::putAuthorizationKey - Definition...</a>	44
1766	<a href="#">Tabelle 12: I Authorization Management Insurant::deleteAuthorizationKey - Definition .....</a>	48
1767	<a href="#">.....</a>	
1768	<a href="#">Tabelle 13: I Authorization Management Insurant::replaceAuthorizationKey - Definition .....</a>	51
1769	<a href="#">.....</a>	
1770	<a href="#">Tabelle 14: I Authorization Management Insurant::getAuditEvents - Definition .....</a>	53
1771	<a href="#">Tabelle 15: I Authorization Management Insurant::putNotificationInfo - Definition .....</a>	55
1772	<a href="#">Tabelle 16: I Authorization Management Insurant::getAuthorizationList - Definition .....</a>	57
1773	<a href="#">Tabelle 17: Berechtigungstypen für AuthorizationType .....</a>	60
1774	<a href="#">Tabelle 18: Namensräume .....</a>	69
1775	<a href="#">Tabelle 19: Inhalte Autorisierungsbestätigung .....</a>	70
1776	<a href="#">Tabelle 20: Referenzierte Dokumente der gematik.....</a>	76
1777	<a href="#">Tabelle 21: Referenzierte externe Dokumente .....</a>	77
1778		

## 1779 9.5 Referenzierte Dokumente

### 1780 9.5.1 Dokumente der gematik

1781 Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument  
 1782 referenzierten Dokumente der gematik zur Telematikinfrastruktur. Der mit der  
 1783 vorliegenden Version korrelierende Entwicklungsstand dieser Konzepte und  
 1784 Spezifikationen wird pro Release in einer Dokumentenlandkarte definiert. Version und  
 1785 Stand der referenzierten Dokumente sind daher in der nachfolgenden Tabelle nicht  
 1786 aufgeführt. Deren zu diesem Dokument jeweils gültige Versionsnummer ist in der  
 1787 aktuellen, von der gematik veröffentlichten Dokumentenlandkarte enthalten, in der die  
 1788 vorliegende Version aufgeführt wird.

#### 1790 Tabelle 20: Referenzierte Dokumente der gematik

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Glossar der Telematikinfrastruktur

[gemSysL_ePA]	gematik. Systemspezifisches Konzept ePA
[AuthorizationService.wsdl]	Schnittstellendefinition Komponente Autorisierung
[AuthorizationService.xsd]	Schemadefinition der Schnittstellen der Komponente Autorisierung
[TelematikError.xsd]	Schemadefinition Fehlermeldungen TelematikError
[PHR_Common.xsd]	Schemadefinition für übergreifende ePA-Datentypen
[gemKPT_Arch_TIP]	Konzept Architektur der TI-Plattform
[gemSpec_Perf]	Spezifikation Performancevorgaben und Mengengerüst
[gemSpec_Krypt]	Spezifikation der in der TI zulässigen kryptografischen Verfahren
[gemSpec_OID]	Spezifikation Festlegung von OIDs
[gemSpec_OM]	Spezifikation Operation und Maintenance
[gemSpec_PKI]	Übergreifende Spezifikation PKI
[gemSpec_TB_Auth]	Übergreifende Spezifikation Tokenbasierte Authentisierung
[gemSpec_TSL]	Spezifikation der Schnittstelle des TSL-Dienstes

## 9.5.2 Weitere Dokumente

**Tabelle 21: Referenzierte externe Dokumente**

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[SAML2.0]	Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 <a href="http://docs.oasis-open.org/security/saml/v2.0/">http://docs.oasis-open.org/security/saml/v2.0/</a>
[SOAP]	W3C (2007): SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), <a href="https://www.w3.org/TR/soap12-part1/">https://www.w3.org/TR/soap12-part1/</a>
[WSDL]	W3C: Web Services Description Language (WSDL) 1.1 <a href="https://www.w3.org/TR/wsdl.html">https://www.w3.org/TR/wsdl.html</a>

[WSDL11SOAP12]	W3C (2006): WSDL 1.1 Binding Extension for SOAP 1.2, <a href="https://www.w3.org/Submission/wsd11soap12/">https://www.w3.org/Submission/wsd11soap12/</a>
[WSIBP]	Web-Services Interoperability Consortium (2010): WS-I Basic Profile V2.0 (final material), <a href="http://ws-i.org/Profiles/BasicProfile-2.0-2010-11-09.html">http://ws-i.org/Profiles/BasicProfile-2.0-2010-11-09.html</a>
[WS-Trust1.4]	WS-Trust 1.4 <a href="http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/errata01/os/ws-trust-1.4-errata01-os-complete.pdf">http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/errata01/os/ws-trust-1.4-errata01-os-complete.pdf</a>
[WSS]	OASIS (2006): Web Services Security: SOAP Message Security 1.1 (WS-Security 2004), <a href="http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf">http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf</a>
[WSS-SAML]	OASIS (2006): Web Services Security: SAML Token Profile 1.1, <a href="https://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTOKENProfile.pdf">https://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTOKENProfile.pdf</a>
[XSPA]	OASIS: Cross-Enterprise Security and Privacy Authorization (XSPA) Profile of Security Assertion Markup Language (SAML) for Healthcare Version 2.0 <a href="http://docs.oasis-open.org/xspa/saml-xspa/v2.0/saml-xspa-v2.0.html">http://docs.oasis-open.org/xspa/saml-xspa/v2.0/saml-xspa-v2.0.html</a>
[SGB V]	BGBI. I S.2477 (20.12.1988): Sozialgesetzbuch, Fünftes Buch Zuletzt geändert durch Art. 4 G v. 14.4.2010 I 410 Gesetzliche Krankenversicherung
[RFC-5322]	Internet Message Format - Format für E-Mail-Adressen <a href="https://tools.ietf.org/html/rfc5322">https://tools.ietf.org/html/rfc5322</a>
[RFC5280]	Internet X.509 Public Key Infrastructure Certificate Prüfung von Zertifikaten entlang einer Zertifikatskette (inkl. Cross-Zertifikaten) bis zu einem Vertrauensanker (Root-CA) <a href="https://tools.ietf.org/html/rfc5280#page-71">https://tools.ietf.org/html/rfc5280#page-71</a>