

---

## C\_12299\_Anlage

---

# Inhaltsverzeichnis

<b>1 Änderungsbeschreibung.....</b>	<b>2</b>
<b>2 Änderung in gemSpec_IDP_Dienst.....</b>	<b>4</b>
<b>3 Änderungen in gemSpec_Perf.....</b>	<b>19</b>
3.1.1 E-Mail-Report IDP-Dienst.....	19
<b>4 Änderungen in Steckbriefen.....</b>	<b>21</b>
4.1 Änderungen in gemProdT_IDP-Dienst_PTV.....	21
4.2 Änderungen in gemAnbT_IDP-Dienst_ATV.....	21

---

## 1 Änderungsbeschreibung

---

### Problemstellung:

Der Schlüsselwechselprozess des IDP-Dienstes ist derzeit so gestaltet, dass der Anbieter des IDP-Dienstes eine "harte" Umstellung (hard cutover) mit Vorankündigung (per IT-ITSM-Change) durchführt.

Beim Schlüsselwechsel kam es in der Vergangenheit immer wieder zu Problemen, wenn die Clientsysteme (PVS, AVS, KVS und FD) das Discovery Document des IDP-Dienstes nicht unmittelbar nach Bereitstellung des neuen Schlüsselmaterials neu geladen haben bzw. das alte Schlüsselmaterial hartcodiert implementiert hatten (auch bekannt als Zertifikatspinning).

So nutzen Clients, welche den Schlüsselwechsel noch nicht erkannt haben, weiterhin das alte (nicht länger gültige) Schlüsselmaterial. Die mit dem alten Schlüssel für den IDP-Dienst verschlüsselten Daten sind dann am IDP-Dienst nicht verarbeitbar, da dort der alte Schlüssel nicht mehr gültig ist. Neue, vom IDP-Dienst ausgestellte Token können vom Fachdienst nicht validiert werden, wenn der Fachdienst das Discovery Document des IDP-Dienstes nicht unmittelbar nach Bereitstellung des neuen Schlüsselmaterials neu geladen hat.

Es kommt aus Sicht der Nutzer zu einer Störung.

Gewünscht ist eine Übergangsphase, in der das neu zu verwendende Schlüsselmaterial bekannt gemacht wird, während das alte Schlüsselmaterial noch genutzt werden kann. Um eine schrittweise Migration der Client-Systeme zu ermöglichen, bleibt das bestehende Verhalten zunächst erhalten und es wird parallel dazu ein neuer Schlüsselwechsel-Mechanismus etabliert. Die Client-Systeme können dann selbstständig entscheiden, wann sie eine Migration umsetzen.

Folgende Eigenschaften sind für die Lösung charakteristisch:

- Parallel zur heute im Discovery Document hinterlegten "jwks\_uri" (die auf eine JSON-Datei zeigt, die alle verwendeten Schlüssel enthält), wird in diesem Dokument der neue Parameter "signed\_jwks\_uri" aufgenommen.
- Als Signaturschlüssel wird das gleiche Schlüsselmaterial verwendet, das auch zum Signieren des Metadaten-Statements (Discovery Document) verwendet wird.
- "signed\_jwks\_uri" referenziert auf ein als JSON Web Signature signiertes JSON Web Key Set (JWKS), das alle derzeit gültigen öffentlichen Schlüssel des IDP-Dienstes auflistet.
- Für jeden "key" in diesem Set wird eine eindeutige Key-ID (Parameter "kid") im UUID7-Format [[RFC9562#name-uuid-version-7](#)] vom Anbieter verwaltet - im Unterschied zu bisher: gab es einen konstanten Wert: "idp\_puk\_sig".
- Der Identifier "kid" wird im Discovery Document an verschiedenen Stellen verwendet und muss für eine konsistente Referenzierung jeweils die passenden Identifier verwenden.
- Während des Übergangszeitraums (üblicherweise mindestens 14 Tage vor dem Entfernen des alten Schlüsselmaterials) enthält das signed\_JWKS dann beide public keys. Nach Ablauf der Übergangszeit kann der alte Schlüssel entfernt werden.
- Die unsignierten JSON Web Key Sets unter jwks\_uri (puk\_idp\_sig, puk\_idp\_enc, puk\_idp\_sig\_sek, uri\_puk\_idp\_enc puk\_idp\_sek) und uri\_puk\_idp\_sig (puk\_idp\_sig) werden befristet.



---

## 2 Änderung in gemSpec\_IDP\_Dienst

---

Änderungen an gemSpec\_IDP\_Dienst zur Umsetzung des optimierten Schlüsselwechselprozesses

In Kapitel 4 Zerlegung des Produkttyps

Befristung: Langfristig sollen die öffentlichen Signatur- und Encryption Schlüssel nur noch über das signed\_jwks\_uri JWKS bereitgestellt werden, daher wird die Bereitstellung unter den einzelnen Schlüsseln als eigene URI (uri\_puk\_idp\_enc uri\_puk\_idp\_sig) befristet.

Durch Auswertung der gelieferten Betriebsdaten (siehe Kapitel [3- Änderungen in gemSpec\_Perf]) wird die gematik die Umstellung der Clients auf die neuen Schnittstellen beobachten und an einem geeigneten Zeitpunkt die Entfernung dieser Schnittstellen über ein weiteres Spezifikationsupdate vorbereiten.

### **A\_20732 -Aufnahme der öffentlichen Schlüssel in das Discovery Document**

Der Authorization Server MUSS zu jedem privaten Schlüssel dessen öffentlichen Teil mit einer eigenen absoluten URI in das Discovery Document aufnehmen. [ $\leq$ , IDP-D, funkt. Eignung: Test Produkt/FA]

Änderung nur in der Überschrift: **A\_20732 - Aufnahme der öffentlichen Schlüssel in das Discovery Document (befristet)**

In Kapitel 5.1.1 Aufbau des Discovery Document

alt:

### **A\_20458-02 -Inhalte des Discovery Document**

Der Discovery-Endpunkt MUSS sowohl im internen als auch im externen Discovery Document gemäß [RFC8414#section-2] mindestens die folgenden Attribute als URI angeben:

- "issuer" (hier ist der IdP-Dienst erreichbar),
- "jwks\_uri" (für den Abruf von "PUK\_IDP\_ENC" sowie des öffentlichen Schlüssels und des Zertifikats von "PUK\_IDP\_SIG" entsprechend TAB\_IDP\_DIENST\_0003 [RFC7517] – identifiziert anhand der "kid"-Parameter (puk\_idp\_enc / puk\_idp\_sig),
- "uri\_disc" (URI, unter welcher das Discovery Document bereitgestellt wird),
- "authorization\_endpoint" (URI des Dienstes und des öffentlichen Verschlüsselungsschlüssels des Authorization-Endpunktes gemäß [RFC6749]),
- "sso\_endpoint" (URI des Authorization-Endpunktes für Requests mit SSO-Token),
- "auth\_pair\_endpoint" (URI des Authorization-Endpunktes für Requests mit Pairing-Daten),
- "token\_endpoint" (URI des Token-Endpunktes gemäß [RFC6749]),
- "uri\_puk\_idp\_enc" und "uri\_puk\_idp\_sig" (URI der JWK-Objekte für die zwei Schlüssel und des Zertifikates).

[ $\leq$ , IDP-D, funkt. Eignung: Test Produkt/FA]

**Geänderte Prüfzuordnung: Die Anforderung richtet sich nun an den Anbieter statt an den Hersteller**

neu:

### A\_20458-03 -Inhalte des Discovery Document

Der Anbieter des IDP-Dienstes MUSS am Discovery-Endpunkt MUSS sowohl im internen als auch im externen Discovery Document gemäß [RFC8414#section-2] mindestens die folgenden Attribute als URI angeben:

- "issuer" (hier ist der IdP-Dienst erreichbar),
- "jwks\_uri" (für den Abruf von „PUK\_IDP\_ENC“ sowie des öffentlichen Schlüssels und des Zertifikats von „PUK\_IDP\_SIG“ entsprechend TAB\_IDP\_DIENST\_0003 [RFC7517] – identifiziert anhand der „kid“-Parameter (puk\_idp\_enc / puk\_idp\_sig)
- "signed\_jwks\_uri" (für den Abruf der öffentlichen Schlüssel und Zertifikate von "PUK\_IDP\_ENC" sowie "PUK\_IDP\_SIG" entsprechend TAB\_IDP\_DIENST\_0003 [RFC7517] – identifiziert anhand der "kid"-Parameter und des "alias" (puk\_idp\_enc / puk\_idp\_sig)),
- "uri\_disc" (URI, unter welcher das Discovery Document bereitgestellt wird),
- "authorization\_endpoint" (URI des Dienstes und des öffentlichen Verschlüsselungsschlüssels des Authorization-Endpunktes gemäß [RFC6749]),
- "sso\_endpoint" (URI des Authorization-Endpunktes für Requests mit SSO Token),
- "auth\_pair\_endpoint" (URI des Authorization-Endpunktes für Requests mit Pairing-Daten),
- "token\_endpoint" (URI des Token-Endpunktes gemäß [RFC6749])
- "uri\_puk\_idp\_enc" und „uri\_puk\_idp\_sig“ (URI der JWK Objekte für die zwei Schlüssel und des Zertifikates).

[<=, IDP-D, funkt. Eignung: Test Produkt/FA]

*Hinweis: Die Anforderung zur Bereitstellung der Attribute "jwks\_uri", "uri\_puk\_idp\_sig" und "uri\_puk\_idp\_enc" wurden inhaltlich unverändert in A\_27962-\* und A\_27963-\* abgetrennt und befristet.*

neu:

### A\_27963 -jwks\_uri im Discovery Document (befristet)

Der Anbieter des IDP-Dienstes MUSS am Discovery-Endpunkt sowohl im internen als auch im externen Discovery Document gemäß [RFC8414#section-2] darüber hinaus die folgenden Attribute als URI angeben:

- "jwks\_uri" (für den Abruf von "PUK\_IDP\_ENC" sowie des öffentlichen Schlüssels und des Zertifikats von "PUK\_IDP\_SIG" entsprechend TAB\_IDP\_DIENST\_0003 [RFC7517] – identifiziert anhand der "kid"-Parameter (puk\_idp\_enc / puk\_idp\_sig).

[<=, Anb\_IDP-D, funkt. Eignung: Test Produkt/FA (Anwendung)]

neu:

### A\_27962 -PUK\_IDP\_SEK und PUK\_IDP\_ENC uri im Discovery Document (befristet)

Der Anbieter des IDP-Dienstes MUSS am Discovery-Endpunkt sowohl im internen als auch im externen Discovery Document gemäß [RFC8414#section-2] zusätzlich die folgenden Attribute als URI angeben:

- "uri\_puk\_idp\_enc" und "uri\_puk\_idp\_sig" (URI der JWK-Objekte für die zwei Schlüssel und das zugehörige Zertifikat).

[<=, IDP-D, funkt. Eignung: Test Produkt/FA]

neu:

### A\_28376 -Inhalte des signierten JWK-Schlüsselsets

Der Anbieter des IDP-Dienstes MUSS sicherstellen, dass das JWK-Schlüsselset, welches im Discovery Document über den Parameter "signed\_jwks\_uri" referenziert ist, die folgenden Schlüssel gemäß TAB\_IDP\_DIENST\_0003 enthält:

1. PuK\_IDP\_SIG,
2. PuK\_IDP\_SIG\_Sek,
3. PuK\_IDP\_ENC,
4. PuK\_DISC\_SIG.

Für Schlüssel vom Typ 1 und 2 gilt, dass neue Schlüssel spätestens 48 Stunden vor Ablauf des aktuell verwendeten Schlüssels gleichen Typs im JWK-Schlüsselset veröffentlicht sein MÜSSEN.

Für Schlüssel vom Typ 3 gilt, dass neue Schlüssel spätestens 72 Stunden vor Ablauf des aktuell verwendeten Schlüssels gleichen Typs im JWK-Schlüsselset veröffentlicht sein MÜSSEN. Entfernte Schlüssel MÜSSEN noch 48 Stunden systemintern zum Entschlüsseln nutzbar sein.

Für Schlüssel vom Typ 4 gilt, dass neue Schlüssel spätestens 2 Wochen vor Ablauf des aktuell verwendeten Schlüssels gleichen Typs im JWK-Schlüsselset veröffentlicht sein MÜSSEN.

[<=, Anb\_IDP-D, Sich.techn. Eignung: Anbietererklärung, organ./betriebl. Eignung: Anbietererklärung]

neu:

#### **A\_28381 -signed\_JWKS IDP\_SIG Schlüssel Verwendung und Nachhaltung**

Der Anbieter des IDP-Dienst SOLL für Schlüssel vom Typ 1 und 2 nach A\_28376-\* sicherstellen, dass die Verwendung eines neuen Schlüssels frühestens 48 Stunden nach dessen Veröffentlichung erfolgt und nicht mehr benutzte Schlüssel noch 48 Stunden im JWK-Schlüsselset vorgehalten werden. [<=, Anb\_IDP-D, organ./betriebl. Eignung: Anbietererklärung]

neu:

#### **A\_28382 -signed\_JWKS DISC\_SIG Schlüssel Verwendung**

Der Anbieter des IDP-Dienst SOLL für neue Schlüssel vom Typ 4 nach A\_28376-\* sicherstellen, dass diese frühestens 2 Wochen nach Veröffentlichung verwendet werden. [<=, Anb\_IDP-D, organ./betriebl. Eignung: Anbietererklärung]

neu:

#### **A\_27936 -Signaturschlüssel des JWK-Schlüsselsets in signiertem Format**

Der Hersteller des IDP-Dienstes MUSS sicherstellen, dass das JWK-Schlüsselset, welches im Discovery Document über den Parameter "signed\_jwks\_uri" referenziert ist, im Format JSON Web Signature bereitgestellt und mit dem "PrK\_DISC\_SIG" Schlüssel gemäß TAB\_IDP\_DIENST\_0003 und [[RFC7517](#)] signiert wird.

*Hinweis: Das Schlüsselset wird mit dem Content-Type: "application/jwk-set+json" bereitgestellt.* [<=, IDP-D, funkt. Eignung: Test Produkt/FA]

neu:

#### **A\_27937 -Konsistente Repräsentation des JWK-Schlüsselsets in zwei Formaten (befristet)**

Der Hersteller des IDP-Dienstes MUSS am Discovery-Endpunkt sowohl im internen als auch im externen Discovery Document gemäß [\[RFC8414#section-2\]](#) sicherstellen, dass die öffentlichen Anteile der Schlüsselpaare in den JWK-Schlüsselsets, die durch "jwks\_uri", "signed\_jwks\_uri", "uri\_puk\_idp\_sig" und "uri\_puk\_idp\_enc" referenziert werden, identisch sind.

Die in dem unter "signed\_jwks\_uri" referenzierten Schlüsselset veröffentlichten Schlüssel MÜSSEN für die anderen Bereitstellungsorte wie folgt angepasst werden:

- Abweichend von A\_27940-\* wird das "alias" als "kid" Wert übernommen,
- Abweichend von A\_27938-\* entfällt das "alias".

Sind in dem unter "signed\_jwks\_uri" referenzierten Schlüsselset gleichzeitig mehrere Schlüssel mit dem gleichen "alias" veröffentlicht, welche jeweils eine UUIDv7 als "kid" verwenden, so MÜSSEN die Schlüssel für die anderen Bereitstellungsorte wie folgt gewählt werden:

- Bei Schlüsseln mit einem auf "\_enc" endenden "alias", ist der jüngste (mit größerem numerischem Wert des "kid") zu wählen,
- Bei Schlüsseln mit einem auf "\_sig" endenden "alias", ist der älteste (mit kleinerem numerischem Wert des "kid") zu wählen.

Sind in dem unter "signed\_jwks\_uri" referenzierten Schlüsselset gleichzeitig mehrere Schlüssel mit dem gleichen "alias" veröffentlicht, von denen einer keine UUIDv7 als "kid" verwendet, so MÜSSEN die Schlüssel für die anderen Bereitstellungsorte wie folgt gewählt werden:

- Bei Schlüsseln mit einem auf "\_enc" endenden "alias", ist der Schlüssel zu wählen, der eine UUIDv7 als "kid" verwendet,
- Bei Schlüsseln mit einem auf "\_sig" endenden "alias", ist der Schlüssel zu wählen, der keine UUIDv7 als "kid" verwendet.

[<=, IDP-D, funkt. Eignung: Test Produkt/FA]

*Hinweis: Sobald alle den IDP-Dienst nutzenden Systeme mit dem neuen Schlüssel-Set unter "signed\_jwks\_uri" umgehen können, wird die Unterstützung für das alte Format A\_27962-\* und A\_27963-\* aus dem IDP-Dienst entfernt.*

neu:

### **A\_27940 -Einheitliche Schlüssel-Identifikatoren**

Der Anbieter des IDP-Dienstes MUSS am Discovery-Endpunkt sowohl im internen als auch im externen Discovery Document gemäß [\[RFC8414#section-2\]](#) sicherstellen, dass die öffentlichen Anteile der Schlüsselpaare in den JWK-Schlüsselsets, die durch "signed\_jwks\_uri" referenziert werden, mit dem Parameter "kid" befüllt sind. Der Wert des jeweiligen "kid"-Parameters MUSS für neu hinzugefügte Schlüssel im UUIDv7-Format gemäß [\[RFC9562#name-uuid-version-7\]](#) angegeben sein, dessen Wert den Zeitpunkt der Schlüsselerzeugung widerspiegelt.

[<=, Anb\_IDP-D, funkt. Eignung: Test Produkt/FA (Anwendung)]

*Hinweis:*

*Für Systeme, die den IDP-Dienst nutzen und vom IDP-Dienst herausgegebene Token validieren, ist bei der Auswahl des Schlüssels in der signed\_JWKS, der zum "kid" im Token-Signaturheader passt, zu beachten, dass während einer Übergangsphase eine Abweichung zu berücksichtigen ist. Entspricht der "kid"-Wert im Signaturheader einem Wert aus TAB\_IDP\_DIENST\_0003 (in Kleinschreibung), so ist der korrelierende Schlüssel in der signed\_JWKS anhand des "alias" zu identifizieren. Dabei können mehrere Schlüssel zu berücksichtigen sein.*

neu:

**A\_27938 -Ergänzung des Parameters "alias" in signierten JWK-Schlüsselsets**

Der Anbieter des IDP-Dienstes MUSS am Discovery-Endpunkt sowohl im internen als auch im externen Discovery Document gemäß [\[RFC8414#section-2\]](#) sicherstellen, dass die öffentlichen Anteile der Schlüsselpaare in den JWK-Schlüsselsets, die durch "signed\_jwks\_uri" referenziert werden, den Parameter "alias" mit den zugehörigen Werten aus TAB\_IDP\_DIENST\_0003 gemäß [\[RFC7517\]](#) in Kleinschreibweise enthalten. [ $\leq$ , IDP-D, funkt. Eignung: Test Produkt/FA]

*Hinweis: Der Parameter „alias“ im JWK dient der leichteren Zuordenbarkeit der Schlüsselverwendung.*

*neu:*

**A\_27941 -Zweifaches Vorkommen von JWK mit gleichem "alias" im Key-Set während des Übergangszeitraums beim Schlüsselwechsel**

Der Anbieter des IDP-Dienstes MUSS während des Übergangszeitraums beim Schlüsselwechsel sicherstellen, dass:

- der aktuell gültige öffentliche Schlüssel mit zugehörigem "alias" im Key Set vorhanden ist,
- der zukünftig gültige öffentliche Schlüssel mit gleichem "alias" im Key Set vorhanden ist,
- beide Schlüsselpaare vom System verwendet werden können.

In der Auflistung innerhalb des signed\_JWKS soll ein jüngerer Schlüssel (mit größerem numerischen Wert des "kid") stets vor dem älteren Schlüssel des gleichen "alias" erfolgen. Der abzulösende Schlüssel soll mit dem Attribut "deprecated" versehen sein, das den gemäß [\[RFC7519#section-2\]](#) als "NumericDate" formatierten Zeitpunkt angibt, ab dem dieser Schlüssel nicht mehr verwendet werden soll.

[ $\leq$ , IDP-D, funkt. Eignung: Test Produkt/FA]

*Hinweis: Den IDP-Dienst nutzende Systeme müssen ab dem Umstellungszeitpunkt  $t_n$  (siehe Kapitel "Schlüsselwechselprozess bei JWK-Schlüsselsets") damit umgehen können, dass die ausgestellten Token mit neuen Schlüsseln signiert sind. Der aktuell verwendete Schlüssel ist stets durch seine im Signatur-Header der JWS angegebene Key-ID „kid“ identifizierbar. Im alten Format bezieht sie sich stets auf die "kid" im JWKS – im neuen signed\_JWKS Format, sofern die Schlüsselauswahl gemäß Hinweis zu A\_27940-\* anhand des „alias“ erfolgt, auf einen dieser Schlüssel mit dem gleichlautenden "alias". Stehen mehrere passende Schlüssel (des gleichen "alias") zur Validierung zur Verfügung, so sind ältere (mit kleinerem numerischem Wert des "kid") Schlüssel zu präferieren, neuere aber auch zu berücksichtigen.*

*Hinweis 2: Den IDP-Dienst nutzende Systeme sollen bei der Auswahl eines Schlüssels, dessen Verwendungszweck das Verschlüsseln ist, den jeweils jüngeren (mit größerem numerischen Wert des "kid") Schlüssel präferieren.*

## Neues Kapitel 5.1.5 Schlüsselwechselprozess

Der bisher etablierte Schlüsselwechsel zu einem Stichtag wird über einen "harten" Wechsel eines Schlüssels durchgeführt. Dies kann bei den nutzenden Systemen zu Problemen führen, die den Arbeitsablauf stören oder unterbrechen. Durch das parallel bereitgestellte signierte Schlüssel-Set wird ein Schlüsselwechselprozess mit Übergangszeitraum ermöglicht, während dem sowohl die aktuell gültigen als auch die zukünftig gültigen Schlüssel im Schlüssel-Set bereitgestellt werden und angepasste

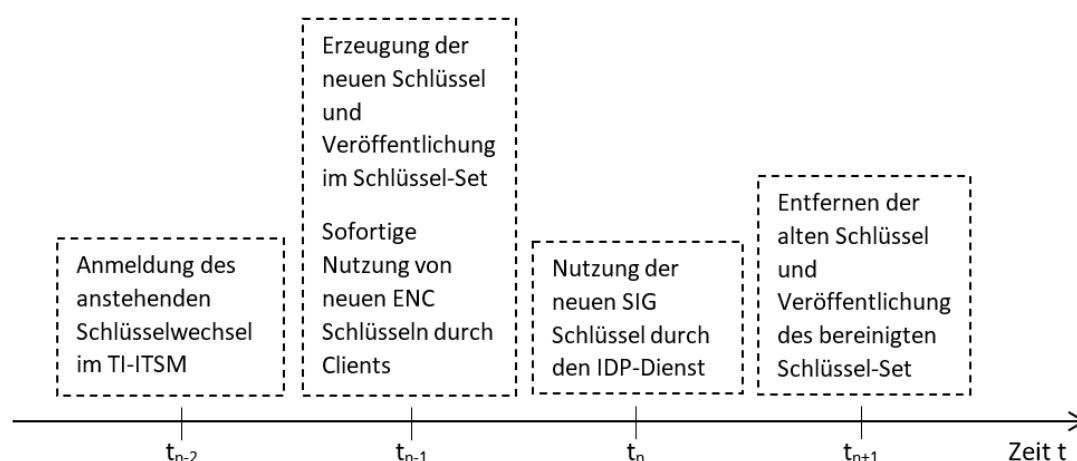


betriebliche Anforderungen an den IDP-Dienst die zuverlässige Entschlüsselung auch während des Wechsel des Verschlüsselungsschlüssels sicherstellen.

Vor Beginn des Übergangszeitraums wird der Schlüsselwechsel über das TI-ITSM beantragt ( $t_{n-2}$ ). Während des Übergangszeitraums ( $t_{n-1}$  bis  $t_n$ ) haben Clientsysteme die Aufgabe, den zukünftig gültigen Signaturschlüssel in ihre Systeme (deren Key- bzw. Trust-Store) zu importieren, so dass diese Schlüssel zum Schlüsselwechsel-Termin ( $t_n$ ) vom IDP-Dienst genutzt werden können. Für Verschlüsselungsschlüssel gilt abweichend hiervon, dass diese unverzüglich von den Clientsysteme ab erster Sichtung ( $t_{n-1}$ ) zu benutzen sind, spätestens jedoch nach dem Übergangszeitraum.

Nach Ablauf des Übergangszeitraums werden die ungültigen/abgelaufenen Schlüssel aus dem Key-Set entfernt ( $t_{n+1}$ ).

Die folgende Abbildung "Schematische Übersicht zum Ablauf des Schlüsselwechselprozess beim IDP-Dienst" zeigt die zeitliche Abfolge des Schlüsselwechselprozesses.



**Abbildung 1: Schematische Übersicht zum Ablauf des Schlüsselwechselprozess beim IDP-Dienst**

Wenn alle den IDP-Dienst nutzenden Systeme auf die Verarbeitung des signed\_JWKS umgestellt sind, kann die Umstellung der "kid" im ID\_TOKEN Header erfolgen, dass dieser die tatsächliche "kid" des Schlüssels referenziert statt des "alias".

## Änderungen in Kapitel 7

### Änderung in Kapitel 7.7 Aufbau des Discovery Document

Es wird der "signed\_jwks\_uri" Eintrag unter „jwks\_uri“ ergänzt

```
01 | {
02 |   "alg": "BP256R1",
```

```

03|  "kid": "puk_disc_sig",
04|  "x5c": [
05|    "[Enthält das verwendete Signer-Zertifikat als Base64 ASN.1 DER-
06|    Encoding. Hier kommt ausnahmsweise NICHT URL-safes Base64-Encoding zum Einsatz!]"
07|  ]
08| }
09| {
10|   "authorization_endpoint": "[URL des Authorization Endpunkts.]",
11|   "federation_authorization_endpoint": "[URL des Authorization Endpunkt für
12|   Anfragen an IDP der Föderation - dieser ist nur im Internet verfügbar]",
13|   "auth_pair_endpoint": "[URL des Pairing-Authorization-Endpunkts - dieser ist
14|   nur im Internet verfügbar]",
15|   "third_party_authorization_endpoint": "[URL des third_party-Authorization-
16|   Endpunkts - dieser ist nur im Internet verfügbar]",
17|   "sso_endpoint": "[URL des SSO-Authorization Endpunkts.]",
18|   "uri_pair": "[URL des Pairing-Endpunkts.]",
19|   "token_endpoint": "[URL des Authorization-Endpunkts.]",
20|   "uri_disc": "[URL des Discovery Document.]",
21|   "issuer": "http://url.des.idp",
22|   "jwks_uri": "[URL einer JWKS-Struktur mit allen vom Server verwendeten
23|   Schlüsseln]",
24|   "signed_jwks_uri": "[URL einer JWKS-Struktur mit allen vom Server
25|   verwendeten Schlüsseln]",
26|   "exp": "[Gültigkeit des Token. Beispiel: 1618330390]",
27|   "iat": "[Zeitpunkt der Ausstellung des Token. Beispiel: 1618243990]",
28|   "uri_puk_idp_enc": "http://url.des.idp/idpEnc/jwk.json",
29|   "uri_puk_idp_sig": "http://url.des.idp/idpSig/jwk.json",
30|   "subject_types_supported": [ "pairwise" ],
31|   "id_token_signing_alg_values_supported": [ "BP256R1" ],
32|   "response_types_supported": [ "code" ],
33|   "scopes_supported": [ "openid", "e-rezept", "pairing" ],
34|   "response_modes_supported": [ "query" ],
35|   "grant_types_supported": [ "authorization_code" ],
36|   "acr_values_supported": [ "gematik-ehealth-loa-high" ],
37|   "token_endpoint_auth_methods_supported": [ "none" ],
38|   "code_challenge_methods_supported": [ "S256" ]
39| }
40| .<SIGNATURE>

```

Hinweis: Der "kid" des "puk\_disc\_sig" (Zeile 3) Schlüssels im Header des Discovery Document wird nach einem Schlüsselwechsel im UUID7-Format sein

## Neues in Kapitel 7.8 Aufbau der JSON Web Key Sets

### Kapitel 7.8.1 Aufbau vor/nach einem Schlüsselwechsel

Im folgenden werden exemplarisch die Strukturen und Inhalte der verschiedenen Bereitstellungsorte (signed\_jwks\_uri, jwks\_uri, uri\_puk\_idp\_enc, uri\_puk\_idp\_sig) vor einem Schlüsselwechsel dargestellt. Die aufgeführten Schlüssel sind allen Clientsystemen bereits bekannt und werden in diesen auch verarbeitet, also enc-Schlüssel zum Verschlüsseln benutzt und sig-Schlüssel zur Verifikation herangezogen.

Beispiel dekodierte (signed\_jwks\_uri) signed\_jwks.jws

```

01| <HEADER>.{
02|   "keys": [
03|     {
04|       "kid": "01981609-b701-789b-a5f4-4a24bcd0d347",
05|       "alias": "puk_idp_sig",

```

```

06|         "use": "sig",
07|         "kty": "EC",
08|         "crv": "BP-256",
09|         "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
10|         "y": "qBNddqxoOK_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
11|         "x5c": [
12|             "<Zertifikat im Base64 ASN.1 DER-Encoding>"
13|         ]
14|     },
15|     {
16|         "kid": "01981832-c2d6-7c00-ad32-688ed41831e8",
17|         "alias": "puk_idp_enc",
18|         "use": "enc",
19|         "kty": "EC",
20|         "crv": "BP-256",
21|         "x": "pkU8LLTZsoGTL007yjIkV626aGtwpelJ2Wrx7fZt0To",
22|         "y": "VliGWQLNtyGuQFs9nXbWdE909PFtxb42miy4yaCkCi8"
23|     },
24|     {
25|         "kid": "01981832-f279-738d-b0ba-f803d7638cfb",
26|         "alias": "puk_idp_sig_sek",
27|         "use": "sig",
28|         "kty": "EC",
29|         "alg": "ES256",
30|         "crv": "P-256",
31|         "x": "sk8Cig9IjJqATxrJkWRdw2gJ7Qut7ygToC8o3z2C_IU",
32|         "y": "LGXTzotnGJuMThRp0QWa2HldCfNoxbMh-PownRgAKko"
33|     },
34|     {
35|         "kid": "019975cb-c3e7-7e9e-9988-76bb0efd8684",
36|         "alias": "puk_disc_sig",
37|         "alg": "BP256R1",
38|         "x5c": [
39|             "<Zertifikat im Base64 ASN.1 DER-Encoding (nicht URL-safe)>"
40|         ]
41|     }
42| ].<SIGNATURE>

```

Die im vorherigen signed\_jwks.jws Beispiel gezeigten Schlüssel lassen sich nach den Vorgaben von A\_27937-\* als jwks.json wie folgt darstellen. Dieser Bereitstellungsort enthält nicht den "puk\_disc\_sig", da dieser im JWT-Header des Discovery Document enthalten ist.

Beispiel (jwks\_uri) jwks.json

```

01| {
02|     "keys": [
03|         {
04|             "kid": "puk_idp_sig",
05|             "use": "sig",
06|             "kty": "EC",
07|             "crv": "BP-256",
08|             "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
09|             "y": "qBNddqxoOK_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
10|             "x5c": [
11|                 "<Zertifikat im Base64 ASN.1 DER-Encoding>"
12|             ]
13|         },
14|     ]
15| }

```

```

15|         "kid": "puk_idp_enc",
16|         "use": "enc",
17|         "kty": "EC",
18|         "crv": "BP-256",
19|         "x": "pkU8LLTZsoGTlo07yjIkV626aGtwpelJ2Wrx7fZt0To",
20|         "y": "VliGWQLNtyGuQFs9nXbWdE909PFtxb42miy4yaCkCi8"
21|     },
22|     {
23|         "kid": "puk_idp_sig_sek",
24|         "use": "sig",
25|         "kty": "EC",
26|         "alg": "ES256",
27|         "crv": "P-256",
28|         "x": "sk8Cig9IjJqATxrJkWRdw2gJ7Qut7ygToC8o3z2C_IU",
29|         "y": "LGXTzotnGJuMThRp0QWa2HldCfNoxbMh-PownRgAKko"
30|     }
31| ]
32| }

```

Nach A\_27962-\* ist der encryption key aus der signed\_jwks.jws nach den Vorgaben von A\_27937-\* wie folgt in die enc\_jwk.json zu übernehmen:

Beispiel (uri\_puk\_idp\_enc) enc\_jwk.json

```

1| {
2|   "kid": "puk_idp_enc",
3|   "use": "enc",
4|   "kty": "EC",
5|   "crv": "BP-256",
6|   "x": "pkU8LLTZsoGTlo07yjIkV626aGtwpelJ2Wrx7fZt0To",
7|   "y": "VliGWQLNtyGuQFs9nXbWdE909PFtxb42miy4yaCkCi8"
8| }

```

Nach A\_27962-\* ist der signature key aus der signed\_jwks.jws nach den Vorgaben von A\_27937-\* wie folgt in die sig\_jwk.json zu übernehmen:

Beispiel (uri\_puk\_idp\_sig) sig\_jwk.json

```

01| {
02|   "kid": "puk_idp_sig",
03|   "use": "sig",
04|   "kty": "EC",
05|   "crv": "BP-256",
06|   "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
07|   "y": "qBNddqxo0K_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
08|   "x5c": [
09|     "<Zertifikat im Base64 ASN.1 DER-Encoding>"
10|   ]
11| }

```

## Kapitel 7.8.2 Aufbau während der puk\_idp\_sig Schlüsselwechsel-Phase

Im Kapitel 7.8.1 wurde gezeigt, wie die Schlüssel über die verschiedenen Bereitstellungsorte abzubilden sind.

Im Kapitel 7.8.2 wird nun exemplarisch gezeigt, wie die Schlüssel in den Bereitstellungsorten abzubilden sind, wenn ein neuer Signaturschlüssel hinzu kommt. Der neue Schlüssel wird jedoch noch nicht verwendet. Es ist nur ein gültiger Signatur- und Verschlüsselungsschlüssel vorhanden.

Der neue Signaturschlüssel (im Beispiel: "01981831-b7a6-7e59-a148-e1797eefd02e") wird in die signed\_jwks.jws aufgenommen, so dass Client-Systeme diesen in ihren Truststore importieren können, bevor die erstmalige Nutzung dieses Signaturschlüssels durch den IDP-Dienst erfolgt.

Wenn der IDP-Dienst anschließend auf diesen neuen Signaturschlüssel wechselt, diesen also produktiv nutzt, entsprechen die Ausgaben strukturell wieder den Beispielen in Kapitel 7.8.1, mit dem Unterschied, dass der Schlüssel (im Beispiel: "01981831-b7a6-7e59-a148-e1797eefd02e") an die Stelle des alten Schlüssels (im Beispiel: "01981609-b701-789b-a5f4-4a24bcd0d347") tritt.

Beispiel dekodierte (signed\_jwks\_uri) signed\_jwks.jws

```

01 | <HEADER>.{
02 |   "keys": [
03 |     {
04 |       "kid": "01981831-b7a6-7e59-a148-e1797eefd02e",
05 |       "alias": "puk_idp_sig",
06 |       "use": "sig",
07 |       "kty": "EC",
08 |       "crv": "BP-256",
09 |       "x": "rugLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXvv4",
10 |       "y": "aDnDdqxo0K_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dCeB",
11 |       "x5c": [
12 |         "<Zertifikat im Base64 ASN.1 DER-Encoding>"
13 |       ]
14 |     },
15 |     {
16 |       "kid": "01981609-b701-789b-a5f4-4a24bcd0d347",
17 |       "alias": "puk_idp_sig",
18 |       "use": "sig",
19 |       "kty": "EC",
20 |       "crv": "BP-256",
21 |       "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
22 |       "y": "qBNddqxo0K_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
23 |       "x5c": [
24 |         "<Zertifikat im Base64 ASN.1 DER-Encoding>"
25 |       ]
26 |     },
27 |   ],
28 |   {
29 |     "kid": "01981832-c2d6-7c00-ad32-688ed41831e8",
30 |     "alias": "puk_idp_enc",
31 |     "use": "enc",
32 |     "kty": "EC",
33 |     "crv": "BP-256",
34 |     "x": "pkU8LLTZsoGTlo07yjIkV626aGtwpeLJ2Wrx7fZt0To",
35 |     "y": "VliGWQLNtyGuQFs9nXbWdE909PFtxb42miy4yaCkCi8"
36 |   },
37 |   {
38 |     "kid": "01981832-f279-738d-b0ba-f803d7638cfb",
39 |     "alias": "puk_idp_sig_sek",
40 |     "use": "sig",
41 |     "kty": "EC",
42 |     "alg": "ES256",
43 |     "crv": "P-256",

```

```

43|         "x": "sk8Cig9IjJqATxrJkWRdw2gJ7Qut7ygToC8o3z2C_IU",
44|         "y": "LGXTzotnGJuMThRp0QWa2HldCfNoxbMh-PownRgAKko"
45|     },
46|     {
47|         "kid": "019975cb-c3e7-7e9e-9988-76bb0efd8684",
48|         "alias": "puk_disc_sig",
49|         "alg": "BP256R1",
50|         "x5c": [
51|             "<Zertifikat im Base64 ASN.1 DER-Encoding (nicht URL-safe)>"
52|         ]
53|     }
54| ]
55| }.<SIGNATURE>

```

An den Inhalten der jwks.json ändert der neu aufgenommene Signaturschlüssel (im Beispiel: "01981831-b7a6-7e59-a148-e1797eefd02e") nichts, da die Abbildung nach A\_27937-\* als jwks.json vorsieht, nur den aktuell verwendeten Signaturschlüssel an dieser Stelle auszugeben.

Beispiel (jwks\_uri) jwks.json - unverändert im Vergleich zur Darstellung in Kap. 7.8.1

```

01| {
02|     "keys": [
03|         {
04|             "kid": "puk_idp_sig",
05|             "use": "sig",
06|             "kty": "EC",
07|             "crv": "BP-256",
08|             "x": "pogLhok59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
09|             "y": "qBNddqxoOK_2Vd5ocnuQtP1q_PuRsLxfAQjv4E4dReA",
10|             "x5c": [
11|                 "<Zertifikat im Base64 ASN.1 DER-Encoding>"
12|             ]
13|         },
14|         {
15|             "kid": "puk_idp_enc",
16|             "use": "enc",
17|             "kty": "EC",
18|             "crv": "BP-256",
19|             "x": "pkU8LLTZsoGTlo07yJikV626aGtwpeLJ2Wrx7fZt0To",
20|             "y": "VliGWQLNtyGuQFs9nXbWdE909PFtxb42miy4yaCkCi8"
21|         },
22|         {
23|             "kid": "puk_idp_sig_sek",
24|             "use": "sig",
25|             "kty": "EC",
26|             "alg": "ES256",
27|             "crv": "P-256",
28|             "x": "sk8Cig9IjJqATxrJkWRdw2gJ7Qut7ygToC8o3z2C_IU",
29|             "y": "LGXTzotnGJuMThRp0QWa2HldCfNoxbMh-PownRgAKko"
30|         }
31|     ]
32| }

```

An den Inhalten der enc\_jwk.json ändert der neu aufgenommene Signaturschlüssel (im Beispiel: "01981831-b7a6-7e59-a148-e1797eefd02e") nichts, dahier nur der Verschlüsselungsschlüssel ausgegeben wird.

Beispiel (uri\_puk\_idp\_enc) enc\_jwk.json -*unverändert im Vergleich zur Darstellung in Kap. 7.8.1*

```

1| {
2|   "kid": "puk_idp_enc",
3|   "use": "enc",
4|   "kty": "EC",
5|   "crv": "BP-256",
6|   "x": "pkU8LLTZsoGTlo07yjIkV626aGtwpelJ2Wrx7fZt0To",
7|   "y": "VliGWQLNtyGuQFs9nXbWdE909PFtxb42miy4yaCkCi8"
8| }

```

An den Inhalten der sig\_jwk.json ändert der neu aufgenommene Signaturschlüssel (im Beispiel: "01981831-b7a6-7e59-a148-e1797eefd02e") nichts, da die Abbildung nach A\_27937-\* als jwks.json vorsieht, nur den aktuell verwendeten Signaturschlüssel an dieser Stelle auszugeben.

Beispiel (uri\_puk\_idp\_sig) sig\_jwk.json -*unverändert im Vergleich zur Darstellung in Kap. 7.8.1*

```

01| {
02|   "kid": "puk_idp_sig",
03|   "use": "sig",
04|   "kty": "EC",
05|   "crv": "BP-256",
06|   "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
07|   "y": "qBNddqxo0K_2Vd5ocnuQtP1q_PuRsLxfAQjv4E4dReA",
08|   "x5c": [
09|     "<Zertifikat im Base64 ASN.1 DER-Encoding>"
10|   ]
11| }

```

## Kapitel 7.8.3 Aufbau während der puk\_idp\_enc Schlüsselwechsel-Phase

Im Kapitel 7.8.2 wurde gezeigt, wie die Schlüssel über die verschiedenen Bereitstellungsorte abzubilden sind, während ein neuer Signaturschlüssel dazu kommt. Dieser neue Signaturschlüssel zu diesem Zeitpunkt jedoch noch nicht zum Signieren benutzt wird.

Es wird in diesem Kapitel exemplarisch gezeigt, wie die Schlüssel in den Bereitstellungsorten abzubilden sind, wenn ein neuer Verschlüsselungsschlüssel hinzu kommt.

Der neue Verschlüsselungsschlüssel (im Beispiel: "01982cab-c519-7890-9d21-edecac5b591a") ist sofort gültig und unverzüglich durch Drittsysteme zu verwenden, wenn Inhalte verschlüsselt an den IDP-Dienst übertragen werden sollen. Der bisherige Schlüssel (im Beispiel: "01981832-c2d6-7c00-ad32-688ed41831e8") wird mit dem Attribut "deprecated" und mit dem Zeitstempel vom Zeitpunkt, als der neue Schlüssel in das Key Set aufgenommen wurde, versehen.

Beispiel dekodierte (signed\_jwks\_uri) signed\_jwks.jws

```

01 | <HEADER>: {
02 |   "keys": [
03 |     {
04 |       "kid": "01981609-b701-789b-a5f4-4a24bcd0d347",
05 |       "alias": "puk_idp_sig",
06 |       "use": "sig",
07 |       "kty": "EC",
08 |       "crv": "BP-256",
09 |       "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
10 |       "y": "qBNddqxo0K_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
11 |       "x5c": [
12 |         "<Zertifikat im Base64 ASN.1 DER-Encoding>"
13 |       ]
14 |     },
15 |     {
16 |       "kid": "01982cab-c519-7890-9d21-edecac5b591a",
17 |       "alias": "puk_idp_enc",
18 |       "use": "enc",
19 |       "kty": "EC",
20 |       "crv": "BP-256",
21 |       "x": "FP8L87ijwzRoELmSouPSrzhY03BfmXKNDvJlrI1b8Bc=",
22 |       "y": "pUOnVot6NYYs6_xKluD0B30-BuWwvoxVd2-Ww1Gy5Ao="
23 |     },
24 |     {
25 |       "kid": "01981832-c2d6-7c00-ad32-688ed41831e8",
26 |       "alias": "puk_idp_enc",
27 |       "use": "enc",
28 |       "kty": "EC",
29 |       "crv": "BP-256",
30 |       "x": "pkU8LLTZsoGTlo07yJikV626aGtwpelJ2WrX7fZt0To",
31 |       "y": "VliGWQLNtyGuQFs9nXbWdE909PFtxb42mIy4yaCkCi8",
32 |       "deprecated": 1752752527,
33 |     },
34 |     {
35 |       "kid": "01981832-f279-738d-b0ba-f803d7638cfb",
36 |       "alias": "puk_idp_sig_sek",
37 |       "use": "sig",
38 |       "kty": "EC",
39 |       "alg": "ES256",
40 |       "crv": "P-256",
41 |       "x": "sk8Cig9IjJqATxrJkWRdw2gJ7Qut7ygToC8o3z2C_IU",
42 |       "y": "LGXTzotnGJuMThRp0QWa2HldCfNoxbMh-PownRgAKko"
43 |     },
44 |     {
45 |       "kid": "019975cb-c3e7-7e9e-9988-76bb0efd8684",
46 |       "alias": "puk_disc_sig",
47 |       "alg": "BP256R1",
48 |       "x5c": [
49 |         "<Zertifikat im Base64 ASN.1 DER-Encoding>"
50 |       ]
51 |     }
52 |   ]
53 | }.<SIGNATURE>

```

Mit Aufnahme des neuen Verschlüsselungsschlüssel (im Beispiel: "01982cab-c519-7890-9d21-edecac5b591a") in die signed\_jwks.jws ändert sich die Abbildung in der jwks.json gemäß A\_27937-\* dahingehend, dass der neue Schlüssel den bisherigen Schlüssel (im Beispiel: "01981832-c2d6-7c00-ad32-688ed41831e8") dort ersetzt.

Beispiel (jwks\_uri) jwks.json



```

01| {
02|   "keys": [
03|     {
04|       "kid": "puk_idp_sig",
05|       "use": "sig",
06|       "kty": "EC",
07|       "crv": "BP-256",
08|       "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
09|       "y": "qBNddqxoOK_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
10|       "x5c": [
11|         "<Zertifikat im Base64 ASN.1 DER-Encoding>"
12|       ]
13|     },
14|     {
15|       "kid": "puk_idp_enc",
16|       "use": "enc",
17|       "kty": "EC",
18|       "crv": "BP-256",
19|       "x": "FP8L87ijwzRoELmSouPSrzhY03BfmXKNDvJlrI1b8Bc=",
20|       "y": "pUOnVot6NYYs6_xKluD0B30-BuWwvoxVd2-Ww1Gy5Ao="
21|     },
22|     {
23|       "kid": "puk_idp_sig_sek",
24|       "use": "sig",
25|       "kty": "EC",
26|       "alg": "ES256",
27|       "crv": "P-256",
28|       "x": "sk8Cig9IjJqATxrJkWRdw2gJ7Qut7ygToC8o3z2C_IU",
29|       "y": "LGXTzotnGJuMThRp0QWa2HldCfNoxbMh-PownRgAKko"
30|     }
31|   ]
32| }

```

Mit Aufnahme des neuen Verschlüsselungsschlüssels (im Beispiel: "01982cab-c519-7890-9d21-edecac5b591a") in die signed\_jwks.jws ändert sich auch die Abbildung in der enc\_jwk.json gemäß A\_27937-\* dahingehend, dass der neue Schlüssel den bisherigen Schlüssel (im Beispiel: "01981832-c2d6-7c00-ad32-688ed41831e8") dort ersetzt.

Beispiel (uri\_puk\_idp\_enc) enc\_jwk.json

```

1| {
2|   "kid": "puk_idp_enc",
3|   "use": "enc",
4|   "kty": "EC",
5|   "crv": "BP-256",
6|   "x": "FP8L87ijwzRoELmSouPSrzhY03BfmXKNDvJlrI1b8Bc=",
7|   "y": "pUOnVot6NYYs6_xKluD0B30-BuWwvoxVd2-Ww1Gy5Ao="
8| }

```

An den Inhalten der sig\_jwk.json ändert sich hier nichts, da der Verschlüsselungsschlüssel hier nicht enthalten ist.

Beispiel (uri\_puk\_idp\_sig) sig\_jwk.json -*unverändert im Vergleich zur Darstellung in Kap. 7.8.1*

```
01 | {  
02 |   "kid": "puk_idp_sig",  
03 |   "use": "sig",  
04 |   "kty": "EC",  
05 |   "crv": "BP-256",  
06 |   "x": "pogLhoK59j_BX70KqZWQ0GkEckCbr2IJ5HZLRLkXyn8",  
07 |   "y": "qBNddqxo0K_2Vd5ocnuQtP1q_PuRsLxfAQjv4E4dReA",  
08 |   "x5c": [  
09 |     "<Zertifikat im Base64 ASN.1 DER-Encoding>"  
10 |   ]  
11 | }
```

---

## 3 Änderungen in gemSpec\_Perf

---

### <neues Kapitel 3.1.4>

Der Schlüsselwechselprozess des IDP-Dienstes ist derzeit so gestaltet, dass der Anbieter des IDP-Dienstes eine "harte" Umstellung (hard cutover) mit Vorankündigung (per TI-ITSM-Change) durchführt.

Zukünftig wird in einer Übergangsphase das neu zu verwendende Schlüsselmaterial bereits bekannt gemacht, während das aktuelle Schlüsselmaterial noch genutzt werden kann.

Die Bekanntmachung erfolgt über eine parallel zur existierenden "jwks\_uri" im Discovery Document des IDP-Dienstes hinterlegten neuen "signed\_jwks\_uri". Die Client-Systeme können sich entscheiden, welche Variante für sie besser geeignet ist. Im neuen E-Mail-Report berichtet der Anbieter des IDP-Dienstes täglich der gematik über die Nutzung der beiden Schnittstellen.

### <kurze erklärende Sätze vorweg>

#### 3.1.1 E-Mail-Report IDP-Dienst

neu

##### **A\_27774 -E-Mail-Report - Spezifika IDP-Dienst - Lieferung**

Der Anbieter IDP-Dienst MUSS täglich für den jeweiligen Vortag einen Report an[[reporting@gematik.de](mailto:reporting@gematik.de)] schicken, in dem die aufgetretenen Client-ID-/UserAgent-Kombinationen mit

der Anzahl der Aufrufe derjwks\_uri [gem. A\_27963-\*) im FeldidpdJwksUri [gem. A\_27710-\*) und der Anzahl der Aufrufe dersigned\_jwks\_uri [gem. A\_20458-\*) im FeldidpdSignedJwksUri [gem. A\_27710-\*) berichtet werden.

Der Versand des Reports MUSS mit einem Vorlauf von mindestens 14 Kalendertagen nach Aufforderung der gematik an- oder wieder abgeschaltet werden können.

Das Default-Zeitintervall ist täglich, beginnend mit 00:00:00.

Der Anbieter IDP-Dienst MUSS beim Versand des Reports folgende Vorgaben einhalten:

- Zieladresse: [[reporting@gematik.de](mailto:reporting@gematik.de)]
- Betreff: IDP-Report-ClientId\_YYYYDDMM
- Reportdatei als Anhang der E-Mail mit dem Dateinamen: IDP-Report-ClientId\_YYYYDDMM.json

Die angehängte JSON-Datei MUSS dem vorgegebenen Format [gem. A\_27710-\*) entsprechen.

*Hinweis: Groß- und Kleinschreibung von Betreff und Dateinamen sind einzuhalten! YYYYDDMM sind durch Jahr, Monat und Tag des Berichtsdatums zu ersetzen. [≤, Anb\_IDP-D, organ./betriebl. Eignung: Anbietererklärung]*

##### **A\_27710 -E-Mail-Report - Spezifika IDP-Dienst - Format**

Der Anbieter IDP-Dienst MUSS die angehängte JSON-Datei der E-Mail [gem. A\_27774-\*] in folgendem Format verwenden:

```
{
  "date": "<Datum für das die Werte erfasst wurden im konkreten Format: : YYYY-MM-DD>",
  "ci": "<CI-ID der abgefragten Produktinstanz gemäß [A_17764-*], als String>",
  "idpdJwksUri": [
    {
      "total": <Anzahl aller anfragenden Clientsysteme mit gleichem useragent,
      als Integer>,
      "clientId" : "<Client-ID vergeben durch die gematik, sofern vorhanden,
      Datentyp String>",
      "useragent": "<User-Agent gemäß Anforderungslage für Clientsysteme am
      Fachdienst [A_24060-*], Datentyp String>"
    }
  ],
  "idpdSignedJwksUri": [
    {
      "total": <Anzahl aller anfragenden Clientsysteme mit gleichem useragent, als
      Integer>,
      "clientId" : "<Client-ID vergeben durch die gematik, sofern vorhanden, Datentyp
      String>",
      "useragent": "<User-Agent gemäß Anforderungslage für Clientsysteme am
      Fachdienst [A_24060-*], Datentyp String>"
    }
  ]
}
```

*Hinweis: Die beiden Felder "idpdJwksUri" und "idpdSignedJwksUri" sind jeweils ein Array und für jeden User-Agent zu befüllen.*

*Sollten die Informationen für die Client-ID nicht vorliegen, gilt für diesen Parameter A\_22513-\*.【<=, Anb\_IDP-D, organ./betriebl. Eignung: Anbietererklärung】*

---

## 4 Änderungen in Steckbriefen

---

### 4.1 Änderungen in gemProdT\_IDP-Dienst\_PTV

Anmerkung: Die Anforderungen der folgenden Tabelle stellen einen Auszug dar und verteilen sich innerhalb der Tabelle des Originaldokuments gemProdT\_IDP-Dienst\_PTV\_2.8.0-0\_V1.0.0. Alle Anforderungen der Tabelle des Originaldokuments, die in der folgenden Tabelle nicht ausgewiesen sind, bleiben unverändert bestehenden.

**Tabelle 1: Anforderungen zur funktionalen Eignung "Produkttest/Produktübergreifender Test"**

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
A_20458-02	Inhalte des Discovery Document	gemSpec_IDP_Dienst
A_27936	Signaturschlüssel des JWK-Schlüsselsets in signiertem Format	gemSpec_IDP_Dienst
A_27937	Konsistente Repräsentation des JWK-Schlüsselsets in zwei Formaten (befristet)	gemSpec_IDP_Dienst

### 4.2 Änderungen in gemAnbT\_IDP-Dienst\_ATV

Anmerkung: Die Anforderungen der folgenden Tabelle stellen einen Auszug dar und verteilen sich innerhalb der Tabelle des Originaldokuments gemAnbT\_IDP-Dienst\_ATV\_1.1.0\_V1.0.0. Alle Anforderungen der Tabelle des Originaldokuments, die in der folgenden Tabelle nicht ausgewiesen sind, bleiben unverändert bestehenden.

**Tabelle 2: Anforderungen zur funktionalen Eignung "Produkttest/Produktübergreifender Test"**

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
A_20458-03	Inhalte des Discovery Document	gemSpec_IDP_Dienst
A_27963	jwks_uri im Discovery Document (befristet)	gemSpec_IDP_Dienst
A_27962	PUK_IDP_SEK und PUK_IDP_ENC uri im Discovery Document (befristet)	gemSpec_IDP_Dienst
A_28376	Inhalte des signierten JWK-Schlüsselsets	gemSpec_IDP_Dienst
A_27940	Einheitliche Schlüssel-Identifikatoren	gemSpec_IDP_Dienst
A_27938	Ergänzung des Parameters „alias“ in signierten	gemSpec_IDP_Dienst

	JWK-Schlüsselsets	
A_27941	Zweifaches Vorkommen von JWK mit gleichem „alias“ im Key-Set während des Übergangszeitraums beim Schlüsselwechsel	gemSpec_IDP_Dienst

**Tabelle 3: Anforderungen zur betrieblichen Eignung "Anbietererklärung"**

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
A_28376	Inhalte des signierten JWK-Schlüsselsets	gemSpec_IDP_Dienst
A_28381	signed_JWKS IDP_SIG Schlüssel Verwendung und Nachhaltung	gemSpec_IDP_Dienst
A_28382	signed_JWKS DISC_SIG Schlüssel Verwendung	gemSpec_IDP_Dienst