

---

## C\_12628\_Anlage

---

# Inhaltsverzeichnis

<b>1</b>	<b>Änderungsbeschreibung.....</b>	<b>2</b>
<b>2</b>	<b>Änderung in gemSpec_Perf.....</b>	<b>3</b>
2.1.1	Kapitel 2.5.4 - Telemetriedaten.....	3
2.1.1.1	Traces.....	4
2.1.1.2	Logs.....	8
2.1.1.3	Metriken.....	10
2.1.1.4	Anforderungen an Telemetriedatenlieferungen.....	12
<b>2.2</b>	<b>Kapitel 3.27 - VSDM 2 Fachdienst.....</b>	<b>17</b>
2.2.1	Kapitel 3.27.2 - Telemetriedaten - Spezifika VSDM 2.....	18
<b>2.3</b>	<b>Kapitel 3.28 - Digitale Patientenrechnung Fachdienst.....</b>	<b>19</b>
2.3.1	Leistungsanforderungen Digitale Patientenrechnung.....	21
2.3.2	Telemetriedatenlieferung - Spezifika Digitale Patientenrechnung.....	21
2.3.3	Bestandsdatenlieferung - Spezifika Digitale Patientenrechnung.....	23
<b>3</b>	<b>Änderung in gemKPT_Betr.....</b>	<b>26</b>

19

---

## 1 Änderungsbeschreibung

---

20  
21  
22  
23  
24

Im Rahmen der VSDM2-Kommentierung sind einige Anpassungsbedarfe aufgetreten, welche für VSDM2 bereits umgesetzt wurden und nun noch einmal auch für die anderen Produkte (PoPP und DiPag) nachvollzogen werden. Zusätzlich dazu haben sich noch weitere Anpassungsbedarfe ergeben, welche bei der Gelegenheit gleich vorgenommen werden sollen.

## 2 Änderung in gemSpec\_Perf

Die Nummerierung der Kapitel kann von der Nummerierung in der Spezifikation abweichen. Die übergreifenden Anteile, welche im Rahmen der VSDM2-Komentierung für C\_12614 vorgenommen wurden, sind rosa hervorgehoben, da diese für die anderen betroffenen Produkte PoPP und DiPag ggf. noch nicht bekannt sind. Neue Anteile der hier vorliegenden Änderung sind grün, Löschungen rot markiert.

### 2.1.1 Kapitel 2.5.4 - Telemetriedaten

Dienste der TI2.0 Architektur werden Betriebsdaten, Selbstauskunft und Bestandsdaten in einem neuen Datenliefermodell übermitteln. Unter Verwendung des OpenTelemetry-Frameworks werden Telemetriedaten über verteilte Systeme eingesammelt und für die Weitergabe an die gematik aufbereitet. Aus den Telemetriedaten können Informationen zur Performance, Last und Fehlersituationen, sowie Daten über den Status und Versionsständen von Nutzern und Clients abgeleitet werden.

Die gesammelten Telemetriedaten werden auf Basis des OpenTelemetry Protokolls (OTLP) über gRPC gesendet und von der gematik empfangen. Die gematik prüft den Sender auf ein gültiges Client-Zertifikat und erlaubt nur validierte Zugriffe über mTLS.

Generell werden folgende drei Typen von Telemetriedaten unterschieden:

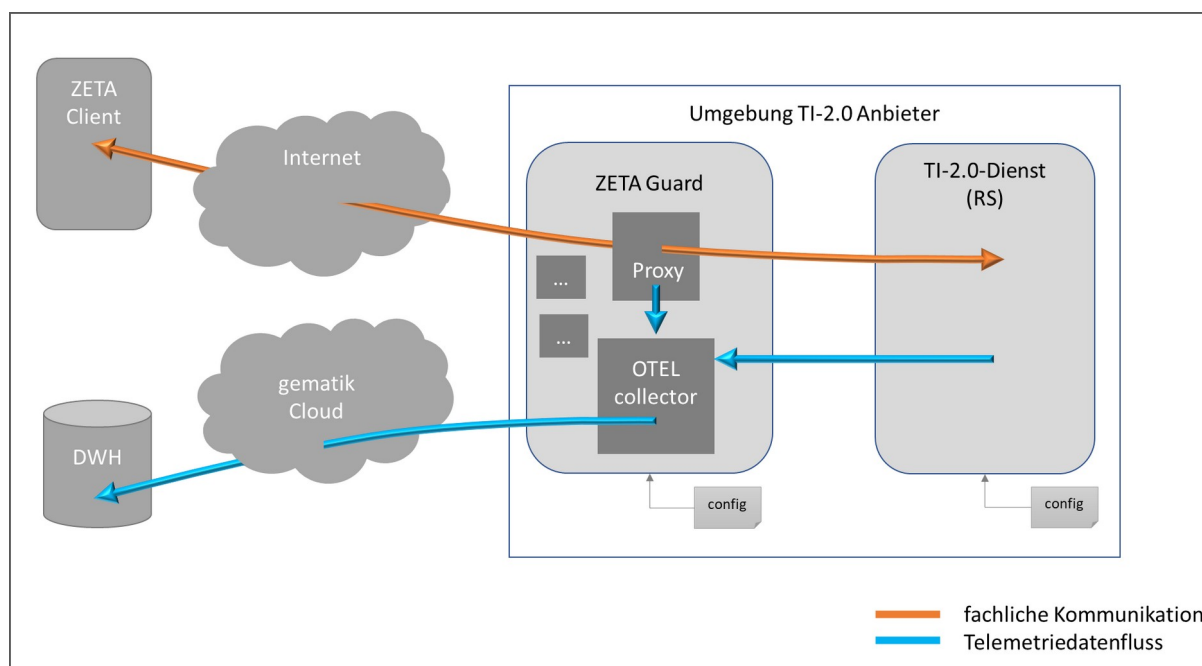
- Traces
- Logs
- Metriken

Die Tabelle 5 - Zuordnung der Telemetriedatentypen zu Datenlieferungen beschreibt die Zuordnung der Telemetriedatentypen zu den gematik bekannten Datenlieferungen.

**Tabelle 1: Zuordnung der Telemetriedatentypen zu Datenlieferungen**

Telemetriedatentyp	Zugeordnete Datenlieferung	Beschreibung
Trace	Betriebsdaten	Informationen über den Ablauf und die Dauer von verteilten Anfragen und Transaktionen über die einzelnen beteiligten Komponenten hinweg.
Logs	Selbstauskunft	Versions- und Statusinformationen werden zeitlich zugeordnet als Logs dokumentiert und können darüber ausgewertet werden
Metriken	Bestandsdaten	Numerische Werte, die den Zustand oder die Leistung über die Zeit messen, z.B. Anzahl erfolgreicher Anfragen pro Minute

Hinzufügen Abbildung



**Abbildung 1 Schema Telemetriedatenlieferung**

Eine zentrale Rolle in der TI2.0 Architektur nehmen die Komponenten von ZETA ein. Der ZETA Guard wird eine zwingende Komponente von TI2.0 Diensten und stellt einen Telemetriedaten-Service zur Verfügung. Dieser Service übernimmt die Kommunikation mit der gematik und führt die Informationen des dahinterliegenden Resource Servers mit den eigenen Aufrufen zusammen. Die dazugehörigen Spezifikationen sind in dem Dokument [gemSpec\_ZETA] näher skizziert und Anforderungen zum Aufbau des Telemetriedaten-Service innerhalb des ZETA Guard sind ebenfalls in dem Dokument verortet.

Der ZETA PIP / PAP Service ist ebenfalls Bestandteil der TI2.0 Architektur und wird Telemetriedaten entsprechend dem spezifizierten Aufbau liefern.

Die Konfiguration des ZETA Guards (insb. im Hinblick auf die Parametrisierung der Datenlieferung) und auch für die Datenlieferung zwischen RS und ZETA Guard müssen über ausgelagerte Konfigurationselemente geschehen, damit für die Anpassung der Datenlieferungen nicht die umfangreichen Prozesse der Softwareentwicklung genutzt werden müssen.

Für die Übermittlung der Telemetriedaten kommen sowohl bei der Übermittlung vom RS zu ZETA Guard, als auch vom ZETA Guard an die BDE der gematik, Standardmechanismen des OpenTelemetry Frameworks zu Einsatz.

### 2.1.1.1 Traces

Ein gesendeter Trace enthält 1-n Spans, die grundlegende Einheit von einzelnen Operationen in einem verteilten System. Die Gesamtheit der Spans mit gleicher Trace ID ermöglicht eine Nachvollziehbarkeit der gesamten Kette eines Aufrufes.

Hier ein Beispiel eines Traces, der aus zwei einzelnen Operationen zusammengesetzt wird.

{

```
79      "trace_id": "abcd1234efgh5678ijkl9012mnop3456",
80      "spans": [
81          {
82              "span_id": "1111222233334444",
83              "name": "HTTP GET /api/user",
84              "start_time_unix_nano": 1678886400000000000,
85              "end_time_unix_nano": 1678886410000000000,
86              "attributes": [
87                  {"http_fqdn": "vsdm2.aok.net"},
88                  {"http_route": "/ressource/path/1"},
89                  {"http_method": "GET"},
90                  {"http_status": 200} ,
91                  {"product_id": "CLIENTID1234567890AB"},
92                  {"product_version": "2.1.3-10"},
93                  {"profession_oid": "1.2.276.0.76.4.30"}
94              ],
95              "parent_id": null // Root-Span hat keinen Parent
96          },
97          {
98              "span_id": "5555666677778888",
99              "name": "rs-vsdm_2",
100             "start_time_unix_nano": 1678886430000000000,
101             "end_time_unix_nano": 1678886440000000000,
102             "attributes": [
103                 {"http_fqdn": "vsdm2.aok.net"},
104                 {"http_route": "/ressource/path/1"},
105                 {"http_method": "GET"},
106                 {"http_status": 200},
107                 {"rs_statuscode": 200},
108                 {"key": "pdt_size", "value": {"intValue": 456}},
109                 {"key": "pdt_iknumber", "value": {"intValue": 128}},
110                 {"key": "pdt_eTag", "value": {"boolValue": false}},
111                 {"key": "pdt_eTagValue", "value": {"boolValue": false}}
112             ],
113             "parent_id": "1111222233334444" // Parent-Span-ID
114             des Root-Spans
115         }
116     ]
117 }
118
```

```

119         }
120     ]
121 }

```

Fehlt in der Kette der Operationen ein Span eines Ressource Servers kann daraus abgeleitet werden, dass der Ressource Server ausgefallen ist. Ein weiterer Überwachungsmechanismus für den Ausfall eines Ressource Servers ist die Kontrolle einzelner Kubernetes Pods. Die Protokollierung von automatischen Neustarts kann hierzu ausgewertet werden.

Die folgende Tabelle gibt Aufschluss über die möglichen Trace-Parameter. Beschrieben werden die allgemein gültigen Parameter sowie die für alle Ressource Server gültigen, beginnend mit dem Suffix "rs". Im Beispiel sind zusätzlich Key / Value Paare mit dem Suffix "pdt", diese sind produktypspezifisch und werden in Unterkapiteln zu [ML-186188](#) ~~Missing cross-reference~~ beschrieben.

**Tabelle 2: Tab\_gemSpec\_Perf\_Trace\_Parameter\_Telemetriedatenlieferung**

Attributname	zugehörig e Operation	Beschreibung
trace_id	alle	Eindeutige ID eines Traces und Kennzeichnung zueinander gehörender Operationen. Das Feld darf nicht leer sein.
span_id	alle	Eindeutige ID einer einzelnen Operation innerhalb eines Traces. Das Feld darf nicht leer sein.
parent_id	alle	Eindeutige ID zur Identifizierung von Hierarchien innerhalb der Operationen eines Traces.
name	alle	Name der Operation bzw. Systemname. Das Feld darf nicht leer sein.
start_time_unix_nano	alle	Startzeitpunkt der Operation, Nutzung zur Berechnung der Duration aus der Differenz zum Endzeitpunkt. Das Feld darf nicht leer sein.
end_time_unix_nano	alle	Endzeitpunkt der Operation, Nutzung zur Berechnung der Duration aus der Differenz zum Startzeitpunkt. Das Feld darf nicht leer sein.
pod_name	alle	Name des Pods/der Instanz
attributes:http_fqdn	alle	Vollständiger FQDN der ausgeführten Operation. Das Feld darf nicht leer sein.

attributes:http_route	alle	HTTP Route der ausgeführten Operation. Das Feld darf nicht leer sein.
attributes:http_method	alle	HTTP Methode der ausgeführten Operation. Das Feld darf nicht leer sein.
attributes:http_status	alle	3-stelliger HTTP Statuscode gemäß [RFC9110]. Das Feld darf nicht leer sein.
attributes:product_id	alle	Product ID aus dem Token Self-Assessment des aufrufenden Clientsystem
attributes:product_version	alle	Product Version aus dem Token Self-Assessment des aufrufenden Clientsystems
attributes:profession_oid	alle	profession OID aus dem SM-B Zertifikat des aufrufenden Clientsystems
attributes:rs_ci	Ressource-Server des TI 2.0 Dienst	Dies ist die logische Produktinstanz und entspricht der CI-ID aus dem ITSM. Das Feld darf beim Span des Ressource Servers nicht leer sein. Format: String mit 10 Zeichen
attributes:rs_duration	Ressource Server des TI 2.0 Dienst	Dies ist eine vom Ressource Server berechnete Zeitdauer (in Millisekunden), wie lange die Verarbeitung der Operation intern benötigte. Das Feld ist optional.
attributes:rs_instance	Ressource Server des TI 2.0 Dienst	Dies ist eine zusätzliche fachliche Instanz des Resource Servers. Wenn notwendig vorhanden vom Ressource Server befüllt. Das Feld ist optional.
attributes:rs_statuscode	Ressource Server des TI 2.0 Dienst	Ein max. 5-stelliger Statuscode des Ressource Servers gemäß [A_27722]. Produktypspezifische fachliche Fehlercodes werden hier ebenfalls gemappt. Das Feld darf nicht leer sein.
attributes:rs_message_key1	Ressource-Server des TI 2.0 Dienst	JSON-formatierter String (UTF-8, ohne Steuerzeichen oder Binärdaten); evtl. Produktspezifische Zusatzinformationen
attributes:rs_message_key2	Ressource-Server des TI 2.0 Dienst	JSON-formatierter String (UTF-8, ohne Steuerzeichen oder Binärdaten); evtl. Produktspezifische Zusatzinformationen

135

136

### 2.1.1.2 Logs

Zusätzlich zur Nachvollziehbarkeit von verteilten Transaktionen in einer Anwendung mittels Traces kann OpenTelemetry Logs erfassen. Die Telemetriedatenlieferung stellt einen Custom Collector für die Selbstauskunft zur Verfügung. Der ZETA Guard greift diesen Collector auf und lässt ihn durch den Ressource Server befüllen, ebenso liefert der ZETA PIP PAP Server die Informationen auf direktem Weg.

Hier ein Beispiel eines OTEL Log records.

```
{
  "resourceLogs": [
    {
      "resource": {
        "attributes": [
          {"key": "product_name", "value": {"stringValue": "VSDM2_HerstellerX"}},
          {"key": "producttyp_name", "value": {"stringValue": "VSDM2"}},
          {"key": "pod_name", "value": {"stringValue": "VSDM2.pod"}}
        ]
      },
      "scopeLogs": [
        {
          "scope": {
            "name": "com.example.MyLogger",
            "version": "1.0.0"
          },
          "logRecords": [
            {
              "timeUnixNano": 1678886400000000000,
              "body": { "stringValue": "Selbstauskunft" },
              "severityText": "product_info",
              "severityNumber": 9,
              "attributes": [
                { "key": "product_version", "value": { "stringValue": "v1.2.0.0" } },
                { "key": "producttype_version",
```



```

180                                     "value":
181                                     { "stringValue":
182                                     "v.2.1.2.2" }},
183                                     { "key":
184                                     "configuration_version
185                                     ", "value":
186                                     {"stringValue":
187                                     "v1.3.4.2" }}
188                                     ]
189                                     }
190                                 ]
191                             }
192                         ]
193                     }
194                 ]
195             }

```

Die zum OpenTelemetry Log "product\_info" gehörenden Attribute werden in der folgenden Tabelle beschrieben.

**Tabelle 3: Tab\_gemSpec\_Perf\_Telemetriedaten\_product\_info**

Parameter	Beschreibung
product_name	Name des Produkts (durch Hersteller)
producttyp_name	Name des Produkttyp (PDT-Nummer der gematik)
body	{ "stringValue": "Selbstauskunft" }
severityText	"product_info"
severityNumber	9
product_version	Aktuelle Produktversion (durch Hersteller)
producttype_version	Aktuelle Produkttypversion (PTV des Steckbriefes)
configuration_version	Aktuelle Konfigurationsversion (durch Hersteller)
pod_name	Name des Pods/der Instanz (durch Hersteller)
timestamptimeUnixNano	Zeitpunkt der letzten Änderung

### 2.1.1.3 Metriken

Eine OpenTelemetry Metrik beschreibt quantitative Daten, die zur Überwachung und Auswertung eines TI2.0 Fachdienstes spezifiziert werden. Sie wird unter Nutzung des OTEL strukturierten Standardformats bereitgestellt und enthält alle relevanten Informationen, um Messwerte korrekt zu interpretieren und zu analysieren.

Die Metriken verfügen über einen eindeutigen Namen, die die Zuordnung zum TI2.0 Fachdienst ermöglichen und eine zusätzliche kurze Beschreibung. Es können folgende drei Typen von Metriken definiert werden:

- **Counter:** Kumulativer Wert, der nur steigt (z.B. Anzahl der Anfragen).
- **Gauge:** Dynamischer Wert, der sich ändern kann (z.B. CPU-Auslastung).
- **Histogram:** Verteilung von Werten (z.B. Antwortzeiten).

Die Auswertung erfolgt über den im Attribut "value" übermittelten Werte. Wenn dies für bestimmte TI2.0 Fachdienste nicht ausreichend ist, können zusätzlich Schlüssel-Wert-Paare spezifiziert werden.

Hier ein Beispiel eines OTEL Metric records vom Typ Counter.

```
{
  "resource":
  {
    "service.name": "my-service",
    "host.name": "example-host"
  },
  "instrumentation_library":
  {
    "name": "otel-library",
    "version": "1.0.0"
  },
  "metrics":
  [{
    "name": "http_request_count",
    "description": "The number of HTTP requests received",
    "unit": "1",
    "type": "COUNTER",
    "data_points":
    [{
      "labels":
      {
        "method": "GET",
        "status_code": 200
      },
```

```

243         "timeUnixNano": 1672531200000000000,
244         "value": 42
245     },
246     {
247         "labels":
248         {
249             "method": "POST",
250             "status_code": 201
251         },
252         "timeUnixNano": 1672531200000000000,
253         "value": 7
254     }
255 ]
256 }

```

Die zur OpenTelemetry Metrik gehörenden Attribute werden in der folgenden Tabelle beschrieben. Zusätzlich dazu werden die Inhalte der Attribute sowie die Details zu den Datenpunkten (data\_points) im produktspezifischen Kapitel definiert, sofern eine Bestandsdatenlieferung für das jeweilige Produkt vorgesehen ist.

**Tabelle 4: Tab\_gemSpec\_Perf\_Telemetriedaten\_metric**

Parameter	Beschreibung
name	Name der Metrik gemäß Festlegung im produktspezifischen Kapitel
description	Beschreibung der Metrik gemäß Festlegung im produktspezifischen Kapitel
type	Metriktyp gemäß Festlegung im produktspezifischen Kapitel
labels	Labels fassen die Kriterien zusammen, auf die sich der Metrikwert bezieht. Die Benennung von keys und values der labels erfolgt im produktspezifischen Kapitel.
value	Metrikwert des jeweiligen Datenpunktes
timeUnixNano	Zeitpunkt des Sendens

#### 2.1.1.4 Anforderungen an Telemetriedatenlieferungen

Folgende Anforderungen gelten für alle Telemetriedatenlieferungen.

#### **A\_27718 -Telemetriedatenlieferung - Konnektivität zur gematik gewährleisten**

Der Anbieter MUSS die Konnektivität zu der Telemetriedaten-Schnittstelle der gematik durchgehend gewährleisten.

[<=,Anb\_ZT\_PIP\_PAP, Anb\_PoPP\_Service, Anb\_TI-D\_ZT, Anb\_DiPag\_FD, Anb\_VSDM\_2\_FD,organ./betriebl. Eignung: Anbietererklärung]

#### **A\_27719 -Telemetriedatenlieferung - Nutzung eines gültigen Client-Zertifikats**

Der Anbieter MUSS für den Verbindungsaufbau zum Telemetriedaten-Service der gematik ein gültiges TLS Client-Zertifikat nutzen.

Hinweis: Ein entsprechendes TLS Client-Zertifikat muss vor Inbetriebnahme bei der gematik beantragt werden.[<=,Anb\_ZT\_PIP\_PAP, Anb\_PoPP\_Service, Anb\_TI-D\_ZT, Anb\_DiPag\_FD, Anb\_VSDM\_2\_FD,organ./betriebl. Eignung: Anbietererklärung]

#### **A\_27724-02 -Performance - Telemetriedatenlieferung - Korrektheit der Datenlieferung**

Der Produkttyp MUSS die Fachdienstinformationen syntaktisch und semantisch korrekt an den Telemetriedaten-Service senden.

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik.[<=,DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, ZT\_PIP\_PAP,funkt. Eignung: Test Produkt/FA]*

#### **A\_27723-02 -Performance - Telemetriedatenlieferung - Lieferung**

Der Produkttyp MUSS für jede ausgeführte Operation die Betriebsdaten gemäß der Tabelle "Tab\_gemSpec\_Perf\_Trace\_Parameter\_Telemetriedatenlieferung" Trace-Parameter einer Telemetriedatenlieferung an den Telemetriedaten-Service senden. Das Senden der Telemetriedaten SOLL asynchron erfolgen, um die Performance der Operation nicht negativ zu beeinflussen.

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik.[<=,DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, ZT\_PIP\_PAP,funkt. Eignung: Test Produkt/FA]*

neue Afo:

#### **A\_28786 -Performance - Telemetriedatenlieferung - Ausnahme der Vollständigkeit**

Sofern aufgrund einer nicht vollständig durchgeführten Operation die Werte für die Parameter aus den produktspezifischen Vorgaben nicht vorliegen, KANN der Produkttyp bei der Telemetriedatenlieferung auf die Übermittlung des Key-Value-Paares verzichten. [<=,,]

(DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, funkt. Eignung Test Produkt/FA)

Korrektur und Aufteilung in zwei Afos von

#### **A\_28729 - Performance - Telemetriedatenlieferung - Lieferintervalle**

*Die Telemetriedatenlieferung MUSS standardmäßig auf ein Lieferintervall von einer Minute eingestellt werden. Das Lieferintervall SOLL anpassbar in einer ausgelagerten Konfiguration sein und ermöglicht Intervalle von sofort bis 5 Minuten. Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik.[<=]*

neue abgewandelte Afo: wird entfernt

#### **A\_28729 -Performance - Telemetriedatenlieferung - Default-Wert Lieferintervall**

Die Telemetriedatenlieferung MUSS standardmäßig auf ein Lieferintervall von einer Minute eingestellt werden.

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik. [ $\leq$ , DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, funkt. Eignung: Test Produkt/FA]*

und ergänzende Afo: **wird entfernt**

#### **A\_28737 -Performance - Telemetriedatenlieferung - Konfiguration Lieferintervall**

Der Anbieter MUSS das Lieferintervall der Telemetriedatenlieferung in einer ausgelagerten Konfiguration anpassbar vorhalten. Das Intervall MUSS mindestens auf bis zu 600 Sekunden konfigurierbar sein.

*Hinweis: Ein Lieferintervall von 0 Sekunden bedeutet eine Just-In-Time Lieferung der beim Anbieter anfallenden Telemetriedaten an den ZETA-Guard.*

[ $\leq$ , Anb\_VSDM\_2\_FD, organ./betriebl. Eignung: Anbietererklärung]

#### **neue Anforderungen**

#### **A\_28780 -Performance - Telemetriedatenlieferung - Batchverarbeitung RS**

Das Produkt MUSS für die Telemetriedatenlieferung von Traces des Ressource Servers an den Telemetriedaten-Service des ZETA Guards als **ExportProcessorType** den Typ **Batch** verwenden (dies ist auch der Standardwert bei OpenTelemetry)

[ $\leq$ , „]

(DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, funkt. Eignung Herstellererklärung)

#### **A\_28779 -Performance - Telemetriedatenlieferung - Konfiguration Datenlieferung RS**

Das Produkt MUSS die Konfigurierbarkeit der Telemetriedatenlieferung des Ressource Servers an den Telemetriedaten-Service des ZETA Guards in einer ausgelagerten Konfiguration für folgende Parameter ermöglichen:

##### **Batch - BatchExportOptions**

- **maxQueueSize**(Default = 2048)
- **scheduledDelayMilliseconds**(Default = 5000)
- **exporterTimeoutMilliseconds**(Default = 30000)
- **maxExportBatchSize**(Default = 512)

##### **ExporterHelper - retry\_on\_failure**

- **enabled** (Default = true)
- **initial\_interval** (Default = 5s)
- **max\_interval** (Default = 1215s)
- **max\_elapsed\_time** (Default = 1820s)
- **multiplier** (Default = 3)

[ $\leq$ , „]

(DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, funkt. Eignung Herstellererklärung)

## A\_28782 -Performance - Telemetriedatenlieferung - Konfiguration

### Datenlieferung ZETA Guard

Das Produkt MUSS die Konfigurierbarkeit der Telemetriedatenlieferung des ZETA Guards an die gematik Telemetriedatencloud in einer ausgelagerten Konfiguration für folgende Parameter unterstützen:

#### Batch - BatchExportOptions

- **maxQueueSize** (Default = 2048)
- **scheduledDelayMilliseconds** (Default = 5000)
- **exporterTimeoutMilliseconds** (Default = 30000)
- **maxExportBatchSize** (Default = 512)

#### ExporterHelper - retry\_on\_failure

- **enabled** (Default = true)
- **initial\_interval** (Default = 5s)
- **max\_interval** (Default = 1215s)
- **max\_elapsed\_time** (Default = 1820s)
- **multiplier** (Default = 3)

#### Sampler

- **otel\_traces\_sampler** (Default always\_off)
- **otel\_traces\_sampler\_arg** (Default 0)

Hinweis: Sampler Typ wird auf traceidratio gesetzt, um Sampling einzuschalten und die Ratio kann variabel einen Wert zwischen 0..1 einnehmen. Der Typ garantiert, dass der Sampling Trace vollständig ist und alle Spans für eine Auswahl enthalten sind.

[<=,,]

(DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, funkt. Eignung Herstellererklärung)

## A\_28766 -Performance - Telemetriedatenlieferung - Konfiguration

Der Anbieter MUSS die Konfiguration der Parameter zur Telemetriedatenlieferung nach Aufforderung der gematik innerhalb von maximal 24h umsetzen. Dies gilt sowohl für die Telemetriedatenlieferung zwischen Ressource Server und dem Telemetriedaten-Service des ZETA Guards, als auch für die Telemetriedatenlieferung des ZETA Guards an die gematik Telemetriedatencloud.[<=,,]

(Anb\_DiPag\_FD, Anb\_VSDM\_2\_FD, Anb\_PoPP\_Service - organ./betriebl. Eignung: Anbietererklärung)

Hinweis: Der ZETA Guard stellt ebenfalls eine Konfigurationsmöglichkeit über eine ausgelagerte Konfiguration bereit.

Afo wird entfernt da der Inhalt durch ZETA-Guard abgedeckt wird:

## A\_28730 - Performance - Telemetriedatenlieferung - Retry Mechanismus der Lieferung

*OpenTelemetry Standardmechanismen für eine Batchverarbeitung SOLLEN eingesetzt werden. Zum Kompensieren von technischen Störungen MUSS eine Wiederherstellungsstrategie implementiert sein, die sicherstellt, dass die Telemetriedaten über einen konfigurierbaren Zeitraum von 60 Minuten nachgeliefert werden können.Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die*

*Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik.[<=]*

Durch die VSDM-Kommentierung angepasste VSDM2-Afo-Version **wird entfernt:**

#### **A\_28730 -Performance - Telemetriedatenlieferung - Sendeverhalten im Störfungsfall**

Der Produkttyp SOLL im Störfungsfall OpenTelemetry Standardmechanismen für eine Batchverarbeitung einsetzen.

Ein Störfungsfall in diesem Sinne ist unter Anderem ein Zustand, bei dem die Telemetriedatenlieferung an den ZETA Guard fehlschlägt.

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik.[<=,DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT,funkt. Eignung: Test Produkt/FA]*

Durch die VSDM-Kommentierung ergänzte VSDM2-Afo **wird entfernt:**

#### **A\_28738 -Performance - Telemetriedatenlieferung - Pufferung im Störfungsfall**

Der Anbieter SOLL zur Kompensation von Störfungsfällen gemäß [A\_28730] eine Wiederherstellungsstrategie implementieren, die sicherstellt, dass angefallene Telemetriedaten des Resource Servers über einen konfigurierbaren Zeitraum von (mindestens/bis zu) 60 Minuten nachgeliefert werden können.

[<=,Anb\_VSDM\_2\_FD,organ./betriebl. Eignung: Anbietererklärung]

#### **neue Afo**

#### **A\_28754 -Wiederherstellung Datenlieferung im Störfungsfall**

Der Anbieter MUSS zur Kompensation von Störfungsfällen eine Wiederherstellungsstrategie mit OpenTelemetry Standardmechanismen implementieren, die sicherstellt, dass angefallene Telemetriedaten nachgeliefert werden können (z.B. ausreichend Speicherplatz, ausreichend Bandbreite - nicht abschließend).

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik.*

[<=,,]

(Anb\_DiPag\_FD, Anb\_VSDM\_2\_FD, Anb\_PoPP\_Service - organ./betriebl. Eignung: Anbietererklärung)

#### **A\_27722-03 -Performance - Telemetriedatenlieferung - Status Code des Ressource Servers**

Der Produkttyp MUSS im Parameter *rs\_statuscode* entweder einen HTTP-Statuscode des Ressource Servers gemäß Tab\_gemSpec\_Perf\_Telemetriedaten\_RS\_Statuscodes oder gemäß produkttypspezifischer Definition an den Telemetriedaten-Service übermitteln.

**Tabelle 5: Tab\_gemSpec\_Perf\_Telemetriedaten\_RS\_Statuscodes**

HTTP-Statuscodes	Name der Statuscodegruppe	Beschreibung
1xx	INFORMATIONAL	Der Server hat die Anfrage erhalten und befindet sich in der Bearbeitung.
2xx	SUCCESSFUL	Die Operation wurde erfolgreich



		durchgeführt.
3xx	REDIRECTION	Der Client muss zusätzliche Maßnahmen ergreifen, um die Anfrage abzuschließen.
4xx	CLIENT_ERROR	Ein Client-seitiger Fehler verhindert die erfolgreiche Durchführung der Operation.
5xx	SERVER_ERROR	Ein Server-seitiger Fehler verhindert die erfolgreiche Durchführung der Operation.

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik. [≤, DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, ZT\_PIP\_PAP, funkt. Eignung: Test Produkt/FA]*

Hinweis: Es sind vom Anbieter, anstatt der Status Code Klassen (first digit of status code), die konkreten 3-stelligen HTTP-Statuscodes gemäß [RFC9110] zu verwenden.

#### **A\_27729-01 -Performance - Telemetriedatenlieferung - Selbstauskunft**

Der Produkttyp MUSS die in der Tabelle "Tab\_gemSpec\_Perf\_Telemetriedaten\_product\_info" gelisteten Parameter des Open Telemetry Logs "product\_info" befüllen und die Produktinformation bei jeder Änderungen den Telemetriedaten-Service überliefern.

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik.*

*[≤, DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, ZT\_PIP\_PAP, funkt. Eignung: Test Produkt/FA]*

neue Afo

#### **A\_28785 -Performance - Telemetriedatenlieferung - Bestandsdaten**

Der Produkttyp MUSS die in der Tabelle "Tab\_gemSpec\_Perf\_Telemetriedaten\_metric" gelisteten Parameter unter Berücksichtigung der produktspezifischen Vorgaben als eine Open Telemetry Metrik befüllen und im festgelegten, konfigurierbaren Zeitintervall an den Telemetriedaten-Service überliefern.

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

*Hinweis: Ist der ZETA Guard Teil des Produkttyps (z.B. VSDM2) gilt die Nachweispflicht für die Sendung an den Telemetriedaten-Service des ZETA Guards, in anderen Fällen (z.B. sektoraler IDP) an den Telemetriedaten-Service der gematik. [≤, „]*

*(DiPag\_FD, VSDM\_2\_FD, PoPP\_Service, Herst\_TI-D\_ZT, funkt. Eignung Herstellererklärung)*

#####

## **2.2 Kapitel 3.27 - VSDM 2 Fachdienst**

Tabelle 6: Tab\_gemSpec\_Perf\_VSDM: Performancerelevante UseCases

UseCase	Fachdienstoperation	Beschreibung
---------	---------------------	--------------



VSDM2.1	GET /VSDMBundle	Abruf der Versichertenstammdaten
VSDM2.ZT 1	GET /.well-known	ZETA: Abruf gültiger Autorisierungsserver
VSDM2.ZT 2	GET /nonce	ZETA: Nonce abrufen
VSDM2.ZT 3	POST /token <JWT Client Assert>	ZETA: Autorisierung ohne Refresh Token
VSDM2.ZT 4	POST /token <Refresh Token>	ZETA: Autorisierung mit Refresh Token

### Neue Afo-Version

#### A\_27479-04 -Performance - VSDM 2 - Datenlieferung an ZETA-Guard

Der Anbieter VSDM 2 Fachdienst MUSS Daten an die OpenTelemetry-Schnittstelle des Telemetrie-Daten-Service vom ZETA-Guard gemäß [Telemetrie-Daten Service#gemSpec\_ZETA], [Kapitel Telemetriedaten#gemSpec\_Perf] sowie die nachfolgenden produktspezifischen Telemetriedatenanforderungen senden.

Der Anbieter VSDM 2 Fachdienst MUSS folgende Daten an die OpenTelemetry-Schnittstelle des Telemetrie-Daten-Service vom ZETA-Guard gemäß [Kapitel 5.7Telemetrie-Daten Service#gemSpec\_ZETA] und [Kapitel 2.5.4.1 Traces] senden.

#### Daten zu jeder ausgeführten Operation:

- Den Wert "iknummer", Datentyp Integer: Wert von insurerId aus Request-Header ZETA-PoPP-Token-Content
- Den Wert "eTag", Datentyp Boolean: Das Ergebnis der Mismatch-Prüfung gemäß [A\_26776\*]
  - Übereinstimmung der eTag-Werte: true (Keine Übermittlung der VSD)
  - Nicht-Übereinstimmung der eTag-Werte: false (Übermittlung der VSD)
- Den Wert "eTagValue", Datentyp Boolean: Das Ergebnis der Prüfung des eTag-Wertes nach folgendem Schema:
  - eTag-Wert gleich 0: true (Initialabruf, gemäß [A\_26712\*]).
  - eTag-Wert ungleich 0: false (echte Änderung).

Daten zum Zustand des eingesetzten Produkts bei Zustandswechsel (zustandsbasiertes Schema gemäß [A\_27494\*#gemSpec\_ZETA]):

- Die Version des zugelassenen Produkttyps (Produkttypversion).
- Die Version des Resource-Servers (Produktversion).

[<=,Anb\_VSDM\_2\_FD,funkt. Eignung: Anbietererklärung]

### 2.2.1 Kapitel 3.27.2 - Telemetriedaten - Spezifika VSDM 2

In Ergänzung zu den allgemeinen Anforderungen an die Telemetriedatenlieferung befinden sich nachfolgend die produktspezifischen Festlegungen.

## neue Afo-Version

**A\_26823-03 -Performance - Telemetriedaten - Spezifika VSDM 2 - Operation**

Der VSDM 2 Fachdienst MUSS bei Telemetriedatenlieferungen den Use Case eindeutig über die Trace Parameter attributes:http\_method und attributes:http\_route kenntlich machen. Zu berücksichtigen sind die Operationen, die in der Spalte "UseCase" in der Tabelle "Tab\_gemSpec\_Perf\_VSDM: Performancerelevante UseCases" angegeben sind. [ $\leq$ ,VSDM\_2\_FD,funkt. Eignung: Test Produkt/FA]

**A\_26992-02 -Performance - Telemetriedaten - Spezifika VSDM 2 - Status**

Der VSDM 2 Fachdienst MUSS bei Telemetriedatenlieferung im Trace Span zugehörig zum Resource Server bzgl. des Feldes "rs\_statuscode" vorrangig den Fehlercode aus der Spalte "Fehler-Code" gemäß [A\_27012\*] verwenden. In allen anderen Fällen ist der gültige, an den Client zurückgemeldete, HTTP-Response Code in das Feld einzutragen. [ $\leq$ ,VSDM\_2\_FD,funkt. Eignung: Test Produkt/FA]

## neue Afo-Version

**A\_26825-03 -Performance - Telemetriedaten - Spezifika VSDM 2 - Message**

Der VSDM 2 Fachdienst MUSS bei Telemetriedatenlieferungen produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{ "key": "pdt_size", "value": { "intValue": $$$ } }
{ "key": "pdt_iknummer", "value": { "intValue": $$$ } }
{ "key": "pdt_eTag", "value": { "boolValue": true } }
{ "key": "pdt_eTagValue", "value": { "boolValue": false } }
```

Folgende spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind zu berücksichtigen.

- key "pdt\_size":
  - Größe des Requests in Kilobyte
  - Datentyp Integer
- key "pdt\_iknummer"
  - Wert von insurerId aus Request-Header ZTA-PoPP-Token-Content
  - Datentyp Integer
- key "pdt\_eTag"
  - Ergebnis der Mismatch-Prüfung gemäß [A\_26776]
  - Datentyp Boolean
  - Erläuterung:
    - Übereinstimmung der eTag-Werte: true (Keine Übermittlung der VSD)
    - Nicht-Übereinstimmung der eTag-Werte: false (Übermittlung der VSD)
- key "pdt\_eTagValue"
  - Ergebnis der Prüfung des eTag-Wertes nach folgendem Schema
  - Datentyp Boolean
  - Erläuterung
    - eTag-Wert gleich 0: true (Initialabruf, gemäß [A\_26712\*])

- eTag-Wert ungleich 0: false (echte Änderung)

【<=,VS DM\_2\_FD,funkt. Eignung: Test Produkt/FA】

#####

## 2.3 Kapitel 3.28 - Digitale Patientenrechnung Fachdienst

Im Folgenden werden die spezifischen Leistungsanforderungen und Anforderungen des Digitalen Patientenrechnung Fachdienstes aufgeführt.

**Tabelle 7: Tab\_gemSpec\_Perf\_DiPag: Performancerelevante UseCases**

UseCase	Fachdienstoperation	Beschreibung
DIPAG.2	GET /Patient	Abfrage des Rechnungsempfängers
DIPAG.3	GET /Patient/\$dipag-submit	Versand einer Patientenrechnung
DIPAG.4	POST /DocumentReference/\$retrieve	Abruf einer Patientenrechnung
DIPAG.5	GET /DocumentReference	Suche nach einer Patientenrechnung
DIPAG.6	POST /DocumentReference/<id>/ \$change-status	Ändern des Status einer Patientenrechnung
DIPAG.7	POST /DocumentReference/<id>/ \$process-flag	Markieren einer Patientenrechnung
DIPAG.8	POST /DocumentReference/<id>/\$erase	Patientenrechnung löschen
DIPAG.9	GET /AuditEvent	Auditprotokoll einsehen
DIPAG.10	GET /permission	Abruf einer Liste von Berechtigungen
DIPAG.11	POST /permission	Hinzufügen einer Berechtigung
DIPAG.12	DELETE /permission/<id>	Löschen einer Berechtigung
DIPAG.13	POST /permission/<id>/status	Status einer Berechtigung ändern
DIPAG.14	POST /InsuranceAccount	Registrieren eines Nutzerkontos für Kostenträger
DIPAG.15	DELETE /InsuranceAccount/<id>	Löschen eines Nutzerkontos für

		Kostenträger
DIPAG.16	POST /PractitionerAccount	Registrieren eines Nutzerkontos für Rechnungsersteller
DIPAG.17	DELETE /PractitionerAccount/<id>	Löschen eines Nutzerkontos für Rechnungsersteller
DIPAG.18	POST /InsurantAccount	Registrierung eines Nutzerkontos für Versicherte
DIPAG.19	DELETE /InsurantAccount/<id>	Löschen eines Nutzerkontos für Versicherte
DIPAG.20	PUT /InsurantAccount	Aktualisieren eines Nutzerkontos für Versicherte
DIPAG.ZT1	GET /.well-known	ZETA: Abruf gültiger Autorisierungsserver
DIPAG.ZT2	GET /nonce	ZETA: Nonce abrufen
DIPAG.ZT3	POST /token <JWT Client Assert>	ZETA: Autorisierung ohne Refresh Token
DIPAG.ZT4	POST /token <Refresh Token>	ZETA: Autorisierung mit Refresh Token

### neue Afo-Version

#### A\_27542-01 -Performance - Digitale Patientenrechnung - Datenlieferung an ZETA-Guard

Der Anbieter Digitale Patientenrechnung Fachdienst MUSS Daten an die OpenTelemetry-Schnittstelle des Telemetrie-Daten-Service vom ZETA-Guard gemäß [Telemetrie-Daten Service#gemSpec\_ZETA], [Kapitel Telemetriedaten#gemSpec\_Perf] sowie die nachfolgenden produktspezifischen Telemetriedatenanforderungen senden.

Der Anbieter Digitale Patientenrechnung Fachdienst MUSS folgende Daten an die OpenTelemetry-Schnittstelle des Telemetrie-Daten-Service vom ZETA-Guard gemäß [Kapitel 5.7Telemetrie-Daten Service#gemSpec\_ZETA] senden. Alternativ ist es möglich, die Datenlieferung von einer Adapterkomponente des Anbieters, außerhalb des Fachdienstes zur Digitalen Patientenrechnung, an die Schnittstelle des ZETA-Guard zu gewährleisten.

#### Daten zu jedem Schnittstellenaufruf (transaktionales Schema):

- Den Wert "resourceServerDuration", Datentyp Integer: Zeit in ms für die Bearbeitungszeit im Resource-Server.

#### Daten zum Zustand des eingesetzten Produkts (zustandsbasiertes Schema gemäß [A\_27494\*#gemSpec\_ZETA]):

- Die Version des zugelassenen Produkttyps (Produkttypversion).
- Die Version des Resource-Servers (Produktversion).

- Die Version der Konfiguration gemäß [A\_20219\*] (Konfigurationsversion).  
[<=,Anb\_DiPag\_FD,funkt. Eignung: Anbietererklärung]

## 2.3.1 Leistungsanforderungen Digitale Patientenrechnung

...

## 2.3.2 Telemetriedatenlieferung - Spezifika Digitale Patientenrechnung

### Neue Afo-Version

#### A\_27365-02 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Status

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferung im Trace Span zugehörig zum Resource Server bzgl. des Feldes "rs\_statuscode" vorrangig den Fehlercode aus der Spalte "Fehler-Code" gemäß [A\_27547\*#gemSpec\_DiPag\_FD] verwenden. In allen anderen Fällen ist der gültige, an den Client zurückgemeldete, HTTP-Response Code in das Feld einzutragen.

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferungen im Trace Span des Resource Servers bzgl. des Feldes "status" vorrangig den Fehlercode aus der Spalte "Errorcode" gemäß [A\_27547\*#gemSpec\_DiPag\_FD] verwenden. In allen anderen Fällen ist der gültige, an den Client zurückgemeldete, HTTP-Response Code in das Feld einzutragen. [<=,DiPag\_FD,funkt. Eignung: Test Produkt/FA]

### Neue Afo-Version

#### A\_27366-02 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Operation

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferungen den Use Case eindeutig über die Trace Parameter attributes:http\_method und attributes:http\_route kenntlich machen. Zu berücksichtigen sind die Operationen der Tabelle "Tab\_gemSpec\_Perf\_DiPag: Performancerelevante UseCases".

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferungen bzgl. des Feldes "operation" die Angabe der Spalte "UseCase" aus Tabelle [Tab\_gemSpec\_Perf\_DiPag: Performancerelevante UseCases] nutzen. [<=,DiPag\_FD,funkt. Eignung: Test Produkt/FA]

Afo irrelevant und wird entfernt:

#### A\_27565-01 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Konfigurierbarkeit der Lieferung

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferungen das an- und abschalten von gelieferten Operationen gemäß [A\_27366-01] konfigurativ ermöglichen. Standardmäßig ist die Lieferung aller definierten Operationen angeschaltet.

*Hinweis: Die Änderung dieser Konfiguration darf nur auf betriebliche Anordnung der gematik vorgenommen werden. [<=,DiPag\_FD,funkt. Eignung: Test Produkt/FA]*

### Neue Afo-Version

#### A\_27368-02 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Message

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferungen produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{ "key": "pdt_size", "value": { "intValue": $$$ } }
```

Folgende spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind zu berücksichtigen.

- key "pdt\_size":
  - Größe des Requests in Kilobyte
  - Datentyp Integer

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferungen bzgl. des Feldes "message\_keys" folgende spezifischen Festlegungen hinsichtlich des Formates und der Inhalte berücksichtigen:

```
{ "cid": "$clientid", "cv": "$clientversion", "size": $size, "rsdur":  
$resourceServerDuration }
```

- \$clientid: <product\_id> gemäß Token Self-Assessment des Clientsystems, Datentyp String
- \$clientversion: <product\_version> gemäß Token Self-Assessment des Clientsystems, Datentyp String
- \$size: Größe des Requests in kilobyte, Datentyp Integer
- \$resourceServerDuration: Zeit in ms für die Bearbeitungszeit im Resource-Server, Datentyp Integer

Bei der Erstellung des Feldes "message\_keys" ist darauf zu achten, dass weder Whitespaces noch Newlines zwischen JSON-Elementen enthalten sind (kein Indenting) und Vorgaben nach [RFC7493] eingehalten werden. [≤, DiPag\_FD, funkt. Eignung: Test Produkt/FA]

#### Neue AFO zur Duration

### **A\_28784 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Duration**

Der Digitale Patientenrechnung Fachdienst MUSS bei Telemetriedatenlieferung im Trace Span zugehörig zum Resource Server bzgl. des Feldes "rs\_duration" die ermittelte Verarbeitungszeit im Ressource Server übertragen. [≤, „]

DiPag\_FD, funkt. Eignung: Test Produkt/FA

### **2.3.3 Bestandsdatenlieferung - Spezifika Digitale Patientenrechnung**

Anforderung entfällt - Defaultwert wird in der Konfiguration definiert

### **A\_27369 -Performance - Bestandsdaten - Spezifika Digitale Patientenrechnung**

Der Anbieter Digitale Patientenrechnung Fachdienst MUSS in einem definierten, konfigurierbaren Zeitintervall folgende Informationen berichten:

- Anzahl der neu eingestellten Digitalen Patientenrechnungen
- Anzahl aller abgelegten Digitalen Patientenrechnungen
  - davon im Status "offen"
  - davon im Status "erledigt"

- davon im Status "Papierkorb"
- Anzahl aller registrierten Nutzerkonten
  - davon Versicherte
  - davon Kostenträger
  - davon Rechnungsersteller

(Das Default Zeitintervall ist täglich, beginnend mit 00:00:00)  
 [≤,Anb\_DiPag\_FD,organ./betriebl. Eignung: Anbietererklärung]

#### neue Afo-Version

### A\_27370-01 -Performance - Bestandsdaten - Spezifika Digitale Patientenrechnung - Anbieter

Der Anbieter Digitale Patientenrechnung Fachdienst MUSS jeweils zum Wechsel in den nächsten Berichtsintervall die definierten Metriken im Rahmen der Telemetriedatenlieferung übermitteln.

Der Anbieter Digitale Patientenrechnung Fachdienst MUSS jeweils zum Wechsel in den nächsten Berichtsintervall, folgende Informationen im JSON Format als HTTP Body an die Betriebsdatenerfassung gemäß [A\_23110-\*] liefern.

```
{
  "timestamp": "<Zeitstempel der Abfrage gemäß ISO 8601 unter expliziter Angabe der
  Zeitzone UTC im konkreten Format: YYYY-MM-DDTHH:mm:ss[.fff]Z, als String>",
  "ci": "<CI-ID der abgefragten Produktinstanz gemäß [A_17764], als String>",
  "rech": {
    "total": <Anzahl aller gespeicherten Digitalen Patientenrechnungen, als Integer>,
    "offen": <Anzahl im Status "offen", als Integer>,
    "erledigt": <Anzahl im Status "erledigt", als Integer>,
    "papierkorb": <Anzahl im Status "Papierkorb", als Integer>
  },
  "accounts": {
    "total": <Anzahl aller registrierten Nutzerkonten, als Integer>
    "versicherte": <Anzahl von Versicherten, als Integer>,
    "ktr": <Anzahl von Kostenträgern, als Integer>,
    "ersteller": <Anzahl von Rechnungserstellern, als Integer>
  }
}
```

[≤,Anb\_DiPag\_FD,organ./betriebl. Eignung: Anbietererklärung]

#### neue Anforderung:

### A\_28787 -Performance - Bestandsdaten - Spezifika Digitale Patientenrechnung - Metrik Requestanzahl

Der Digitale Patientenrechnung Fachdienst MUSS bei der Lieferung von Bestandsdaten (gem. Tab\_gemSpec\_Perf\_Telemetriedaten\_metric) unter "metrics" eine Metrik mit folgenden Informationen liefern:

"name": "dipag\_request\_count",

```

719         "description": "the number of dipag requests for various
720         statuses",
721         "unit": "1",
722         "type": "COUNTER",
723         "data_points": [...]

```

724  
725 und unter "data\_points" Datenpunkte für folgende Labels:

```

727         "status": "total" <-- Gesamtanzahl digitale Patientenrechnungen
728         "status": "new" <-- Anzahl neue digitale Patientenrechnungen
729         seit letzter Lieferung
730         "status": "open" <-- Anzahl digitale Patientenrechnungen im
731         Status "offen"
732         "status": "done" <-- Anzahl digitale Patientenrechnungen im
733         Status "erledigt"
734         "status": "deleted" <-- Anzahl digitale Patientenrechnungen im
735         Status "Papierkorb"

```

736 **[<=,,]**

737 **DiPag\_FD, funkt. Eignung: Test Produkt/FA**

738 **neue Anforderung:**

#### 739 **A\_28788 -Performance - Bestandsdaten - Spezifika Digitale Patientenrechnung -** 740 **Metrik Benutzergruppen**

741 Der Digitale Patientenrechnung Fachdienst MUSS bei der Lieferung von Bestandsdaten  
742 (gem. Tab\_gemSpec\_Perf\_Telemetriedaten\_metric) unter "metrics" eine Metrik mit  
743 folgenden Informationen liefern:

```

745         "name": "dipag_user_count",
746         "description": "the number and allocation of dipag accounts",
747         "unit": "1",
748         "type": "COUNTER",
749         "data_points": [...]

```

750  
751 und unter "data\_points" Datenpunkte für folgende Labels:

```

753         "user": "total" <-- Gesamtanzahl registrierter Nutzerkonten
754         "user": "insurer" <-- Anzahl Versicherte
755         "user": "insurence" <-- Anzahl Kostenträger
756         "user": "invoiceissuer" <-- Anzahl Rechnungsersteller

```

757 **[<=,,]**

758 **DiPag\_FD, funkt. Eignung: Test Produkt/FA**

759

760



---

### 3 Änderung in gemKPT\_Betr

---

Im Betriebskonzept werden für die definierten Usecases KPIs definiert bzw. bestehende KPIs angepasst. Die Anpassungen dienen der Dokumentation und haben keinen normativen Charakter für die Hersteller und Anbieter. Im Rahmen des Service-Level-Reviews verkörpern die PKIs die technischen SLAs, welche wiederum auf den Performancevorgaben basieren. Wo es keine Performancevorgaben gibt, haben die KPIs nur informativen Charakter.