gematik

C_12827_Anlage

Inhaltsverzeichnis

1 Beschreibung der Änderung....3

2 Änderungen in gemSpec_Perf....4

2.1 Kapitel 2.5 Datenliefermodelle....4

2.1.1 Kapitel 2.5.4 Telemetriedaten....4

2.1.1.1 Allgemeine Anforderungen zur Telemetriedatenlieferung....6

2.1.1.2 Kapitel 2.5.4.2 Traces....11

2.1.1.2.1 Allgemeine Anforderungen zu Traces....18

2.1.1.3 Kapitel 2.5.4.3 Logs....19

2.1.1.3.1 Kapitel 2.5.4.3.1 Eventlog....20

2.1.1.3.2 Kapitel 2.5.4.3.2 Selbstauskunft....22

2.1.1.3.3 Allgemeine Anforderungen zu Logs....25

2.1.1.4 Kapitel 2.5.4.4 Metriken....26

2.1.1.4.1 Kapitel 2.5.4.4.1 Counter-Metriken....28

2.1.1.4.2 Kapitel 2.5.4.4.2 Gauge-Metriken....32

2.1.1.4.3 Kapitel 2.5.4.4.3 Histogram-Metriken....35

2.1.1.4.4 Allgemeine Anforderungen zu Metriken....41

2.1.1.5 Kapitel 2.5.4.4 Anforderungen an Telemetriedatenlieferungen....42

3 Änderungen in gemSpec_Perf 3.26....44

3.1.1 Änderungen in gemSpec_Perf 3.26.2 - Telemetriedaten - Spezifika VSDM 2....44

3.1.1.1 Kapitel 3.26.2.1 - Telemetriedaten - Spezifika VSDM 2 - Traces....44

3.1.1.2 Kapitel 3.26.2.2 - Telemetriedaten - Spezifika VSDM 2 - Logs....45

3.1.1.3 Kapitel 3.26.2.3 - Telemetriedaten - Spezifika VSDM 2 - Metriken....45

3.1.2 3.26.3 Performance-Kenngrößen VSDM 2....45

4 Änderungen in gemSpec_Perf 3.27....46

4.1.1 Änderungen in gemSpec_Perf 3.27.2 - Telemetriedaten - Spezifika Digitale Patientenrechnung....47

4.1.1.1 Kapitel 3.27.2.1 - Telemetriedaten - Spezifika Digitale Patientenrechnung - Traces....47

4.1.1.2 Kapitel 3.27.2.2 - Telemetriedaten - Spezifika Digitale Patientenrechnung - Logs....48

4.1.1.3 Kapitel 3.27.2.3 - Telemetriedaten - Spezifika Digitale Patientenrechnung - Metriken....48

4.1.2 3.27.3 Performance-Kenngrößen Digitale Patientenrechnung....51

5 Änderungen in gemSpec_Perf 3.28....52

5.1.1 3.28.1 Leistungsanforderungen PoPP-Service....53

5.1.2 Änderungen in gemSpec_Perf 3.28.2 - Telemetriedaten - Spezifika PoPP-Service 54

5.1.2.1 Kapitel 3.28.2.1 - Telemetriedaten - Spezifika PoPP-Service - Traces.....55
5.1.2.2 Kapitel 3.28.2.2 - Telemetriedaten - Spezifika PoPP-Service - Logs.....55
5.1.2.3 Kapitel 3.28.2.3 - Telemetriedaten - Spezifika PoPP-Service - Metriken.....56
5.1.3 3.28.3 Performance-Kenngrößen PoPP-Service.....58

6 Änderungen in gemSpec_Perf 3.29.....59

6.1.1 Änderungen in gemSpec_Perf 3.29.1 Leistungsanforderungen TI-Flow-Fachdienst.....62
6.1.1.1 3.29.1.2 Bearbeitungszeiten TI-Flow-Fachdienst.....62
6.1.1.2 3.29.1.3 Performancevorgaben TI-Flow-Fachdienst.....63
6.1.2 Änderungen in gemSpec_Perf 3.29.2 - Telemetriedaten - Spezifika TI-Flow-Fachdienst.....63
6.1.2.1 Kapitel 3.29.2.1 - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Traces 63
6.1.2.1.1 Kapitel 3.29.2.1.1 Telemetriedaten - Spezifika TI-Flow-Fachdienst - Traces - Message.....64
6.1.2.2 Kapitel 3.29.2.2 - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Logs...77
6.1.2.3 Kapitel 3.29.2.3 - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Metriken 77
6.1.3 3.29.3 Performance-Kenngrößen TI-Flow-Fachdienst.....83

1 Beschreibung der Änderung

Die aktuelle Spezifikation der Telemetriedatenerfassung stellt teilweise Zusammenhänge zwischen BDE v2 und Telemetriedatenlieferung in der TI 2.0 unzureichend da und ist insgesamt hinsichtlich der Abgrenzung zwischen Traces, Logs und Metriken zu streng, sodass das Design der Telemetriedatenlieferung nicht den tatsächlichen Bedarf repräsentiert. Zudem fehlt es in der Spezifikation teilweise an klarer Struktur zur Einordnung der Anforderungen.

Ziel ist es die Anforderungen zu den Telemetriedaten und zur Telemetriedatenlieferung neu zu strukturieren und die Abhängigkeiten von Traces, Logs und Metriken klarer zu erläutern, sodass das Konzept und das Design der Telemetriedatenlieferung verständlicher wird und sich folglich durch die Produktteams und Hersteller besser anwenden lässt.

Darüber hinaus finden Rückmeldungen aus der Industrie und internen Fachbereichen Eingang, sodass weitere Schwächen in der Spezifikation der Telemetriedaten behoben werden.

Die produkttypspezifischen Anteile von DiPag, PoPP und VSDM2 werden in dem Zuge ebenfalls so angepasst, dass sie konsistent zu den veränderten übergreifenden Anforderungen sind.

Die Nummerierung der Kapitel entspricht nicht der Nummerierung aus den referenzierenden Dokumenten, da diese durch die Formatierung automatisch erzeugt wird. Dies wird bei der Einarbeitung der Änderungen entsprechend beachtet.

Hinweise zur Lesart:
<<Text, der zur Erklärung der Änderung dient - wird nicht mit eingearbeitet/übernommen.>>
Text, der neu ist.
~~Text, der entfernt wird.~~
Text, der durch Verschieben eingefügt wird.
~~Text, der durch Verschieben entfernt wird.~~

Mögliche Zuordnungen von Afos:
--> Anbieter<xxx> , organ./betriebl. Eignung: Anbietererklärung (obA)
--> Anbieter<xxx> , organ./betriebl. Eignung: Prozessprüfung (PP)
--> Anbieter<xxx> , organ./betriebl. Eignung: Betriebshandbuch (BHB)
--> Hersteller<xxx> , organ./betriebl. Eignung: Herstellererklärung (obH)
--><Produkttyp>: funkt. Eignung: Herstellererklärung (fH)
--><Produkttyp>: funkt. Eignung: Test Produkt/FA (fT)

2 Änderungen in gemSpec_Perf

[In Kapitel 2.5.1 Betriebsdatenlieferung gibt es noch zwei Anforderungen der Betriebsdatenerfassung, welche für die Dienste, welche nun Telemetriedaten liefern nicht mehr relevant sind. Inhaltlich sind die Aspekte über die Telematriedaten-Anforderungen ausreichend abgedeckt.]

Entfernung der Zuweisung zu den Produkttypen: TI-Flow_FD, PoPP_Service, DiPag_FD, VSDM_2_FD für die Afos: A_22057 und A_22482*:

A_22057 -Performance - Betriebsdatenlieferung - Verpflichtung des Anbieters
Der Anbieter MUSS die Erfassung, Aufbereitung und Übermittlung der Betriebsdaten gemäß der allgemeinen und spezifischen Anforderungen gewährleisten.**[**<=, Anb_TI-Flow_FD, Anb_FD_KOM-LE, Anb_SMC-B, Anbieterprofil_BL-1, Anb_eRp_FD, Anb_HBA, Anb_IDP-D, Anb_X.509_TSP_eGK, Anb_SigD, Anb_ZD, Anb_VPN_ZugD, Anb_IDP-Sek_KTR, Anb_Aktensystem_ePA, Anb_IDP_FedMaster, Anb_NCPeH_FD, Anb_FD_VSDM, Anb_TI_Gateway, organ./betriebl. Eignung: Prozessprüfung**]**

A_22482-02 -Performance - Betriebsdatenlieferung - Erfassung von Betriebsdaten
Der Produkttyp MUSS Betriebsdaten gemäß der Vorgaben an der Außenschnittstelle erfassen.

Hinweis: Der Begriff Außenschnittstelle ist im Kapitel 2.1 Bearbeitungszeit definiert.
[<=, TSP X.509 nonQES - HBA, Intermediär VSDM, TSP X.509 nonQES - eGK, Zentrales-Netz, KOM-LE FD, DiPag_FD, Konfigdienst, TSP X.509 nonQES - SMC-B, TI_GW_Zugangsmodul, TSP X.509 nonQES - gSMC, NCPeH_FD, IDP_FedMaster, gematik Root-CA, TSL-Dienst, SigD, VSDD, VSDM_2_FD, Aktensystem_ePA, Verzeichnisdienst, UFS, TIM_FD, SG_BestNetze, TSP X.509 QES, IDP-D, CMS, Namensdienst, eRp_FD, IDP-Sek, VZD_FHIR, TI-Flow_FD, Zugangsdienst, PoPP_Service, funkt. Eignung: Herstellererklärung, funkt. Eignung: Test Produkt/FA**]**

2.1 Kapitel 2.5 Datenliefermodelle

2.1.1 Kapitel 2.5.4 Telemetriedaten

Dienste der TI 2.0-Architektur werden die bisher bekannten 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 desOpenTelemetry Protokolls (OTLP) über gRPC gesendet und von der gematik empfangen.

Generell werden folgende drei Typen von Telemetriedaten unterschieden:

- Traces
- Logs
- Metriken

Die Tabelle 6 - Zuordnung der Telemetriedatentypen zu Datenlieferungen beschreibt die Zuordnung der Telemetriedatentypen zu den gematik bekannten Datenlieferungen. Eventlogs sind dabei eine neue Art der Datenlieferung (als spezifischer Log-Typ), welche durch die Telemetriedatenlieferung eingeführt wird.

Tabelle 1: Zuordnung der Telemetriedatentypen zu Datenlieferungen

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

<Image: 45-screenshot-20260310-164149-45.png>

Abbildung 1: Schema Telemetriedatenlieferung

Eine zentrale Rolle in der TI 2.0-Architektur nehmen die Komponenten von ZETA ein. Der ZETA Guard wird eine zwingende Komponente von TI 2.0-Diensten und stellt einen Telemetriedaten-Service zur Verfügung. Dieser Service übernimmt die Kommunikation mit der gematik und führt die Informationen des dahinterliegenden Ressource 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.

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.

[neues Unterkapitel]

2.1.1.1 Allgemeine Anforderungen zur Telemetriedatenlieferung

[hier hin verschoben]

Folgende Anforderungen gelten für alle Telemetriedatenlieferungen.

Anbindung:

[redaktionelle Anpassung im Titel: Ergänzung von "**Performance - ...**"]

A_27718 -Telemetriedatenlieferung - Konnektivität zur gematik gewährleisten
Der Anbieter MUSS die Konnektivität zu der Telemetriedaten-Schnittstelle der gematik durchgehend gewährleisten.
[<=, Anb_TI-Flow_FD, Anbieterprofil_BL-1, Anb_PoPP_Service, Anb_TI-D_ZT, Anb_DiPag_FD, Anb_VSDM_2_FD, organ./betriebl. Eignung: Anbietererklärung**]**

[redaktionelle Anpassung im Titel: Ergänzung von "**Performance - Telemetriedatenlieferung - ...**"]

A_28890 -Verbindungsaufbau zum Telemetriedaten-Service
Der Anbieter MUSS für den Verbindungsaufbau zum Telemetriedaten-Service der gematik einen OAuth 2.0 Token Exchange (gem. RFC 8693) unterstützen.

Hinweis:
Die Basis für die Authentifizierung muss im Rahmen der Inbetriebnahme bei der gematik durchgeführt werden.**[**<=, Anb_TI-Flow_FD, Anbieterprofil_BL-1, Anb_PoPP_Service, Anb_TI-D_ZT, Anb_DiPag_FD, Anb_VSDM_2_FD, organ./betriebl. Eignung: Anbietererklärung**]**

[Anpassung im Titel: Ergänzung von "**Performance - Telemetriedatenlieferung - ...**" + "des Produkts" wird verschoben hinter "Verbindungsaufbau"]

alt:

A_28891 -Verbindungsaufbau zum Telemetriedaten-Service des Produkts
Das Produkt MUSS für den Verbindungsaufbau zum Telemetriedaten-Service der gematik einen OAuth 2.0 Token Exchange (gem. RFC 8693) unterstützen.

Hinweis:
Die Basis für die Authentifizierung muss im Rahmen der Inbetriebnahme bei der gematik durchgeführt werden.**[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Herstellererklärung**]**

neu:

A_28891-01 -Performance - Telemetriedatenlieferung - Verbindungsaufbau des Produktes zum Telemetriedaten-Service
Das Produkt MUSS für den Verbindungsaufbau zum Telemetriedaten-Service der gematik einen OAuth 2.0 Token Exchange (gem. RFC 8693) unterstützen.

Hinweis:
Die Basis für die Authentifizierung muss im Rahmen der Inbetriebnahme bei der gematik durchgeführt werden.**[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Herstellererklärung**]**

Quantität:

A_28754 -Performance - Telemetriedatenlieferung - Wiederherstellung Datenlieferung im Störungsfall

Der Anbieter MUSS zur Kompensation von Störungsfällen 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_TI-Flow_FD, Anbieterprofil_BL-1, Anb_PoPP_Service, Anb_DiPag_FD, Anb_VSDM_2_FD, organ./betriebl. Eignung: Anbietererklärung**]**

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_TI-Flow_FD, Anbieterprofil_BL-1, Anb_PoPP_Service, Anb_DiPag_FD, Anb_VSDM_2_FD, organ./betriebl. Eignung: Anbietererklärung**]**

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

A_28782-02 -Performance - Telemetriedatenlieferung - Konfiguration Datenlieferung ZETA Guard
Das Produkt MUSS die Konfigurierbarkeit der Telemetriedatenlieferung des ZETA Guards an die gematik Telemetriedatencloud in einer ausgelagerten Konfiguration unterstützen.

Beispielhaft sind hier wichtige Parameter der YAML Konfiguration inkl. Default-Werten gelistet:

```
processors:
 batch:
   send_batch_size: 512
   timeout: 30s
exporters:
 otlp:
   timeout: 30s
   sending_queue:
     enabled: true
     queue_size: 2048
   retry_on_failure:
     enabled: true
     initial_interval: 5s
     max_interval: 1500s
     max_elapsed_time: 4400s
```

[<=, Herst_FdV_ZT, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, ZT_Cluster, funkt. Eignung: Herstellererklärung**]**

[neue Afo]

A_30372 -Performance - Telemetriedatenlieferung - Sampling
Der Anbieter DARF eine Aktivierung/Deaktivierung des Samplings ausschließlich in Abstimmung mit der gematik vornehmen. Die Umsetzungszeit richtet sich dabei nach A_28766*.
[<=, Anb_TI-Flow_FD, Anb_PoPP_Service, Anb_DiPag_FD, Anb_VSDM_2_FD, organ./betriebl. Eignung: Anbietererklärung**]**

Zuweisung an Anb_TI-Flow_FD, Anb_PoPP_Service, Anb_DiPag_FD, Anb_VSDM_2_FD, organ./betriebl. Eignung: Anbietererklärung

A_28988 -Performance - Telemetriedatenlieferung - Konfiguration Datenlieferung ZETA Guard - Sampling
Wenn für die Telemetriedatenlieferung des Produktes an die gematik Telemetriedatencloud die Option Sampling aktiviert wurde, MÜSSEN nachfolgende, konfigurierbare Parameter berücksichtigt werden:

tail_sampling:
decision_wait: 10s
num_traces: 50000
expected_new_traces_per_sec: 200
policies:
- name: errors-always
type: status_code
status_code:
status_codes: [ERROR]
- name: success-sampled
type: probabilistic
probabilistic:
sampling_percentage: 5

Hinweis: Sampling SOLL im Standard deaktiviert sein und kann über die oben genannten Parameter aktiviert werden. Durch die Policy "error-always" wird sichergestellt, dass Fehlermeldungen immer gesendet werden.**[**<=, Herst_FdV_ZT, DiPag_FD, VSDM_2_FD, PoPP_Service, ZT_Cluster, funkt. Eignung: Herstellererklärung**]**

A_28780-01 -Performance - Telemetriedatenlieferung - Batchverarbeitung RS
Das Produkt MUSS für die Telemetriedatenlieferung von Traces des Ressource Servers an den Telemetriedaten-Service des ZETA Guards den OTLP-Exporter mit eingeschaltetem Batchingverwenden.
[<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Herstellererklärung**]**

A_28779-02 -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.

Beispielhaft sind hier wichtige Parameter der YAML Konfiguration inkl. Default-Werten gelistet:

processors:
 batch:
   send_batch_size: 512
   timeout: 30s
exporters:
 otlp:
   timeout: 30s
   sending_queue:
     enabled: true
     queue_size: 2048
   retry_on_failure:
     enabled: true
     initial_interval: 5s
     max_interval: 1500s

max_elapsed_time: 4400s
Hinweis: Die dargestellte Parametrisierung orientiert sich an den OTEL-Standards und kann für die YAML Konfiguration eines OTEL-Collectors am RS genutzt werden. Sofern für die Datensammlung am RS kein OTEL-Collector zum Einsatz kommt, ist die Konfiguration der Datenübermittlung vom RS an den Telemetriedaten-Service des ZETA-Guards im Sinne dieser Konfigurationsparameter zu ermöglichen.
[<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Herstellererklärung**]**

[Korrektur der Afo: *Punkt statt Unterstrich*]

[alt]

A_27722-04 -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 2: Tab_gemSpec_Perf_Telemetriedaten_RS_Statuscodes

HTTP-Statuscodes	Name der Statuscodegruppe	Beschreibung	Bewertung
1xx	INFORMATIONAL	Der Server hat die Anfrage erhalten und befindet sich in der Bearbeitung.	OTHER
2xx	SUCCESSFUL	Die Operation wurde erfolgreich durchgeführt.	SUCCESS
3xx	REDIRECTION	Der Client muss zusätzliche Maßnahmen ergreifen, um die Anfrage abzuschließen.	OTHER
4xx	CLIENT_ERROR	Ein Client-seitiger Fehler verhindert die erfolgreiche Durchführung der Operation.	FAILED_OTHER
5xx	SERVER_ERROR	Ein Server-seitiger Fehler verhindert die erfolgreiche Durchführung der Operation.	FAILED_SERVICE

*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.***[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Test Produkt/FA**]**

[neu]

A_27722-05 -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 3: Tab_gemSpec_Perf_Telemetriedaten_RS_Statuscodes

HTTP-Statuscodes	Name der Statuscodegruppe	Beschreibung	Bewertung
1xx	INFORMATIONAL	Der Server hat die Anfrage erhalten und befindet sich in der Bearbeitung.	OTHER
2xx	SUCCESSFUL	Die Operation wurde erfolgreich durchgeführt.	SUCCESS
3xx	REDIRECTION	Der Client muss zusätzliche Maßnahmen ergreifen, um die Anfrage abzuschließen.	OTHER
4xx	CLIENT_ERROR	Ein Client-seitiger Fehler verhindert die erfolgreiche Durchführung der Operation.	FAILED_OTHER
5xx	SERVER_ERROR	Ein Server-seitiger Fehler verhindert die erfolgreiche Durchführung der Operation.	FAILED_SERVICE

*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.***[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, 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.

Qualität:

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.***[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Test Produkt/FA**]**

[neue Afo]

A_30223 -Performance - Telemetriedatenlieferung - Data Model & OTLP
Der Produkttyp MUSS sicherstellen, dass OpenTelemetry-Signale (Logs, Metrics und Traces) den jeweiligen OpenTelemetry Data Models entsprechen und im Format des OpenTelemetry Protocol (OTLP) erzeugt und übertragen werden.**[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, funkt. Eignung: Herstellererklärung**]**

Zuweisung der Anforderung an: TI-Flow_FD, DiPag_FD, PoPP_Service, VSDM_2_FD, funkt. Eignung: Herstellererklärung

[Anpassung der Afo - Hervorhebung, dass dies nur für Attribute mit dem Präfix "pdt." gilt + Hinweis unter der Afo]

[alt]

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. **[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Herstellererklärung, funkt. Eignung: Test Produkt/FA**]**

[neu]

A_28786-01 -Performance - Telemetriedatenlieferung - Ausnahme der Vollständigkeit
Sofern aufgrund einer nicht vollständig durchgeführten Operation die Werte für die Parameter aus den produktspezifischen Vorgaben (pdt.*) nicht vorliegen, KANN der Produkttyp bei der Telemetriedatenlieferung auf die Übermittlung des Key-Value-Paares verzichten.**[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Herstellererklärung, funkt. Eignung: Test Produkt/FA**]**

Hinweis: die Ausnahme beschränkt sich ausschließlich auf die produkttypspezifischen Attribute, welche aufgrund der jeweiligen Fehlersituation nicht ermittelbar sind. Insbesondere die Attribute "rs.duration" und "rs.statuscode" sind für das Monitoring der gematik elementar und deshalb zu ermitteln und zu übertragen.

[neue Afo]

A_30388 -Performance - Telemetriedatenlieferung - Dokumentation der Telemetriedatenherkunft
Der Anbieter MUSS für alle relevanten Usecases (gemäß der produkttypspezifischen Festlegung) dokumentieren, unter welchen Bedingungen und von welcher Schnittstelle/Komponente die Telemetriedaten bezogen werden.**[**<=, Anb_TI-Flow_FD, Anb_PoPP_Service, Anb_DiPag_FD, Anb_VSDM_2_FD, organ./betriebl. Eignung: Anbietererklärung**]**

Zuweisung der Anforderung an: Anb_TI-Flow_FD, Anb_DiPag_FD, Anb_PoPP_Service, Anb_VSDM_2_FD, org./betriebl. Eignung: Anbietererklärung

2.1.1.2 Kapitel 2.5.4.2 Traces

Traces MÜSSEN dem OpenTelemetry Trace Data Model entsprechen und im Format des OpenTelemetry Trace Protobuf erzeugt werden. ~~Die in dieser Spezifikation aufgeführten OpenTelemetry-Attribute stellen eine gezielte Teilmenge dar, die für sicherheitsrelevante Analysen und Auswertungen vorgesehen ist.~~ Die in dieser Spezifikation beschriebenen OpenTelemetry-Attribute stellen eine gezielte Teilmenge der durch OpenTelemetry definierten Attribute dar. Diese Teilmenge ist für die in dieser Spezifikation beschriebenen Analyse-, Monitoring- und Sicherheitsanwendungsfälle verbindlich zu berücksichtigen. Die Spezifikation definiert bewusst nicht den vollständigen Umfang möglicher OpenTelemetry-Attribute. Implementierungen DÜRFEN zusätzliche Attribute gemäß OpenTelemetry erfassen und übertragen; diese sind jedoch nicht Gegenstand der normativen Anforderungen dieser Spezifikation.

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.

Da es mit ZETA Guard und Resource Server unterschiedliche Komponenten gibt, die Traces erzeugen, werden die Codebeispiele und Parameter-Tabellen separat dargestellt. Bei der Übermittlung der Traces an die Telemetriedatenerfassung der gematik durch den ZETA Guard, spielt diese Trennung keine Rolle.

Hier in der Darstellung wird die OTLP/JSON Notation verwendet, die notwendigen und zugehörigen Attribute der OTLP/gRPC Notation (Proto) - welche für die technische Implementation der Anwendung relevant sind - sind der jeweils aktuell gültigen und veröffentlichten Beschreibung des OpenTelemetry Projektes zu entnehmen.

~~Hier ein~~ Beispiel eines Trace Spans des HTTP Proxys:

(Die Daten werden automatisch vom ZETA Guard PEP HTTP Proxy generiert, d.h. für die Erfassung dieser Daten und Generierung des Spans ist keine Implementation am Ressource Server notwendig.)

```
{
      "resourceSpans": [{
            "resource": {
                  "attributes": [{
                        "key": "service.name",
                        "value": { "stringValue": "ZETA Guard PEP HTTP
                        proxy" }
                  },{
                        "key": "service.version",
                        "value": { "stringValue": "1.2.3.0" }
                  },{
                        "key": "server.address",
                        "value": { "stringValue": "vsdm2.aok.net" }
                  }]
            },
            "scopeSpans": [{
                  "scope": {
                        "name": "proxy-tracing",
                        "version": "1.0.0"
                  },
                  "spans": [{
                        "traceId": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",
                        "spanId": "1111111111111111",
                        "name": "HTTP GET /vsdservice/v1/vsdmbundle",
                        "kind": 3,
                        "startTimeUnixNano": "1714311000000000000",
                        "endTimeUnixNano": "1714311000200000000",
                        "attributes": [{
                              "key": "http.method",
                              "value": { "stringValue": "GET" }
                        },{
                              "key": "http.route",
```

```
                                    "value": { "stringValue":
                                    "/vsdservice/v1/vsdmbundle" }
                              },{
                                    "key":
                                    "http.response.status_codehttp.request.header.
                                    x-forwarded-for",
                                    "value": { "stringValue": "203.0.113.10" }
                              },{
                                    "key": "server.address",
                                    "value": { "stringValue": "vsdm2.aok.net" }
                              },{
                                    "key": "product_id",
                                    "value": { "stringValue": "CLIENTID123456" }
                              },{
                                    "key": "product_version",
                                    "value": { "stringValue": "2.1.3-10" }
                              },{
                                    "key": "profession_oid",
                                    "value": { "stringValue":
                                    "1.2.276.0.76.4.30" }
                              }],
                              "status":{
                                    "code": 1
                              }
                        }]
                  }]
            }]
      }
```

Die folgende Tabelle gibt Aufschluss über die möglichen Trace-Parameter des ZETA HTTP Proxy Spans. Die tabellarische Auflistung berücksichtigt nicht explizitdie Hierarchie-Ebene, diese kann aus dem Schema Beispiel entnommen werden.

Tabelle 4 Tab_gemSpec_Perf_Trace_HTTPSpanParameter_Telemetriedatenlieferung

Daten	Open Telemetry Attribute (OTLP/JSON)	Pflichtfeld	Beschreibung
resourceSpans > resouce > attributes			
Service Name	**service.name**	**ja**	Identifikationsmerkmal, fester Wert "ZETA Guard PEP HTTP proxy"
Service	**service.version**	**ja**	Selbstauskunft zur Version

Version			des ZETA Guard
HTTP FQDN	**server.address**	**ja**	Identifikation des Zielservers bei Multi-Host-Umgebungen und Erkennung von Anfragen an nicht autorisierte oder verdächtige Domänen.
resourceSpans > scopeSpans > spans			
Startzeit	**startTimeUnixNano**	**ja**	Startzeitpunkt der Operation
Endzeit	**endTimeUnixNano**	**ja**	Endzeitpunkt der Operation
resourceSpans > scopeSpans > spans > attributes			
HTTP Methode	**http.method**	**ja**	Erkennung von unautorisierten Aktionen oder abnormalen Anfragen. Ein unerwarteter HTTP-Methoden-Typ auf einer bestimmten Route kann auf einen Angriffsversuch hindeuten.
HTTP Route	**http.route**	**ja**	Überwachung von Zugriffen auf spezifische Ressourcen und Erkennung von unautorisierten Zugriffen oder Versuchen, nicht existierende Routen zu erreichen (z.B. für Brute-Force-Angriffe oder Enumeration).
~~HTTP FQDN~~	~~**server.address**~~	~~**ja**~~	~~Identifikation des Zielservers bei Multi-Host-Umgebungen und Erkennung von Anfragen an nicht autorisierte oder verdächtige Domänen.~~
HTTP Status	**http.response.status_code**	**ja**	Erkennung von Fehlern, Ausfällen oder potenziellen Angriffsversuchen (z.B. viele 401/403-Fehler bei Brute-Force-Angriffen, viele 5xx-Fehler bei Denial-of-Service-Angriffen).
Product Id	**product_id**	**ja**	Product ID aus dem Token Self-Assessment des aufrufenden Clientsystems

Product Version	**product_version**	**ja**	Product Version aus dem Token Self-Assessment des aufrufenden Clientsystems
Profession OID	**profession_oid**	ja - sofern möglich	Profession OID aus dem SM-B Zertifikat des aufrufenden Clientsystems. Das Attribut ist nur für die Usecases zu übermitteln, in denen ein SM-B-Zertifikat Anwendung findet.

Ergänzend zum ZETA Guard Proxy Span, hier das Beispiel eines Trace Span eines Resource Servers:

```
{
      "resourceSpans": [{
            "resource": {
                  "attributes": [{
                        "key": "service.name",
                        "value": { "stringValue": "rs.vsdm2.service" }
                  },{
                        "key": "lpi.ci_id",
                        "value": { "stringValue": "CI-12312345" }
                  },{
                        "key": "lpi.instance",
                        "value": { "stringValue": "a" }
                  }]
            },
            "scopeSpans": [{
                  "scope": {
                        "name": "rs-tracing",
                        "version": "1.0.0"
                  },
                  "spans": [{
                        "traceId": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",
                        "spanId": "2222222222222222",
                        "parentSpanId": "1111111111111111",
                        "name": "HTTP GET backend-service",
                        "kind": 2,
                        "startTimeUnixNano": "1714311000205000000",
```

```
                        "endTimeUnixNano": "1714311000900000000",
                        "attributes": [{
                              "key": "rs.statuscode",
                              "value": { "stringValue": "200" }
                        },{
                              "key": "rs.duration",
                              "value": { "intValue": 1234535345 }
                        },{
                              "key": "pdt.ikn",
                              "value": { "intValue": 100000000 }
                        },{
                              "key": "pdt.eTag",
                              "value": { "boolValue": false }
                        },{
                              "key": "pdt.eTagValue",
                              "value": { "boolValue": false }
                        },{
                              "key": "pdt.size",
                              "value": { "intValue": 12 }
                        }],
                        "status": {
                              "code": 1
                        }
                  }]
            }]
      }]
}
```

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 Präfix "pdt_", diese sind produkttypspezifisch und werden in entsprechenden Unterkapiteln (siehe Kapitel 3) beschrieben.~~Die tabellarische Auflistung berücksichtigt nicht explizit~~die Hierarchie Ebene, diese kann aus dem Schema Beispiel entnommen ~~werden.~~

Tabelle 5 Tab_gemSpec_Perf_Trace_RSSpanParameter_Telemetriedatenlieferung

Daten	Open Telemetry Attribute (OTLP/JSON)	Pflichtfeld	Beschreibung
resourceSpans > resource > attributes			

Service Name	**service.name**	ja	Identifikationsmerkmal, **fester Wert (mit dem Präfix "rs.")** zur Erkennung des Resource Servers
CI-Identifikator	lpi.ci_id	ja	Identifikation des Konfiguration-Elementes "logische Produktinstanz", beginnend mit "CI" Entspricht "CI-ID" der Selbstauskunft v1
CI-Instanz	lpi.instance	Nein	Identifikation der logischen Produktinstanz, sofern redundante Instanzen vorhanden sind
resourceSpans > scopeSpans > spans			
Trace-Id	**traceId**	ja	Eindeutige ID eines Traces und Kennzeichnung zueinander gehörender Operationen (wird vom OTEL Framework gesetzt)
Parent Span-Id	**parentSpanId**	ja	Eindeutige ID zur Identifizierung von Hierarchien innerhalb der Operationen eines Traces (wird vom OTEL Framework gesetzt)
Startzeit	**startTimeUnixNano**	ja	Startzeitpunkt der Operation Entspricht "timestamp" der Betriebsdatenlieferung v2
Endzeit	**endTimeUnixNano**	ja	Endzeitpunkt der Operation
resourceSpans > scopeSpans > spans > attributes			
Resource Server Statuscode	**rs.statuscode**	ja	Ein max. 5-stelliger Statuscode des Ressource Servers gemäß [A_27722-*]. Produkttypspezifische fachliche Fehlercodes sind hier **vorrangig** vor "Standard-HTTP-Statuscodes" zu verwenden. Entspricht "status" der Betriebsdatenlieferung v2
Resource Server Duration	**rs.duration**	ja	Dies ist eine vom Ressource Server berechnete Zeitdauer (in Millisekunden), wie lange die

			Verarbeitung der Operation intern benötigte. Entspricht "duration_in_ms" der Betriebsdatenlieferung v2
Resource Server Instanz	**rs.instance**	nein	Dies ist eine zusätzliche fachliche Instanz des Resource Servers. Wenn vorhanden vom Ressource Server befüllt. Das Feld ist optional.
Produkt spezifische Attribute	**pdt.***	Produkttyp spezifische Anforderungen sind zu beachten	Im Beispiel sind zusätzlich Key / Value Paare mit dem Präfix "pdt.", diese sind produkttypspezifisch und werden in entsprechenden Unterkapiteln (siehe Kapitel 3) beschrieben. Enthält die in der Betriebsdatenlieferung v2 beschriebenen Inhalte von "message".

Fehlt in der Kette der Operationen ein Span eines Ressource Servers kann daraus abgeleitet werden, dass der Ressource Server ausgefallen ist.

[neues Unterkapitel]

2.1.1.2.1 Allgemeine Anforderungen zu Traces

[hier hin verschoben]

[redaktionelle Änderung des Afo-Titels --> Performance - Telemetriedatenlieferung-**Traces** - Lieferung RS]

A_27723-03 -Performance - Telemetriedatenlieferung - Lieferung RS
Der Produkttyp MUSS für jede ausgeführte Operation die Betriebsdaten (Traces) mit Traceparametern gemäß der Tabelle "Tab_gemSpec_Perf_Trace_RSSpanParameter_Telemetriedatenlieferung" in 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.***[**<=, TI-Flow_FD, DiPag_FD, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, funkt. Eignung: Test Produkt/FA**]**

[redaktionelle Änderung des Afo-Titels --> Performance - Telemetriedatenlieferung-**Traces** - Lieferung ZETA Guard HTTP Proxy]

A_29020 -Performance - Telemetriedatenlieferung - Lieferung ZETA Guard HTTP Proxy
Der Produkttyp MUSS für jede ausgeführte Operation die Betriebsdaten (Traces) mit Traceparametern gemäß der Tabelle "Tab_gemSpec_Perf_Trace_HTTPSpanParameter_Telemetriedatenlieferung" in einer Telemetriedatenlieferung an den Telemetriedaten-Service senden. Das Senden der

Telemetriedaten SOLL asynchron erfolgen, um die Performance der Operation nicht negativ zu beeinflussen.
[<=, ZT_Cluster, funkt. Eignung: Test Produkt/FA**]**

2.1.1.3 Kapitel 2.5.4.3 Logs

Zusätzlich zur Nachvollziehbarkeit von verteilten Transaktionen in einer Anwendung mittels Traces kann OpenTelemetry Logs erfassen. Die Selbstauskunft wird an den Telemetriedaten-Service über einen OTLP-Exporter als OTEL Log Signal gesendet. Der ZETA Guard stellt einen OTEL Collector für Empfang und Weiterversandt der Selbstauskunft undEventlogs bereit. Eventbasierte Logs werden durch definierte Vorgänge ausgelöst und stellen neben der Selbstauskunft einen allgemeinen Logdatensatz dar, welcher hinsichtlich seiner Attribute produktspezifisch beschrieben werden muss.

Logdaten MÜSSEN dem OpenTelemetry Log Data Model entsprechen und im Format des OpenTelemetry Log Protobuf erzeugt und übertragen werden.~~Die in dieser Spezifikation beschriebenen OpenTelemetry-Attribute stellen eine gezielte Teilmenge dar, die für sicherheitsrelevante Analysen und Auswertungen im Kontext der Selbstauskunft von Log Signalen vorgesehen ist.~~ Die in dieser Spezifikation beschriebenen OpenTelemetry-Attribute stellen eine gezielte Teilmenge der durch OpenTelemetry definierten Attribute dar. Diese Teilmenge ist für die in dieser Spezifikation beschriebenen Analyse-, Monitoring- und Sicherheitsanwendungsfälle verbindlich zu berücksichtigen.

Diese Spezifikation erhebt keinen Anspruch auf Vollständigkeit hinsichtlich aller durch OpenTelemetry definierten Log-Attribute. Implementierungen DÜRFEN zusätzliche Attribute gemäß OpenTelemetry erfassen und übertragen; diese sind jedoch nicht Gegenstand der normativen Anforderungen dieser Spezifikation, sofern sie die Auswertung der hier definierten Attribute nicht beeinträchtigen.

Hier in der Darstellung wird die OTLP/JSON Notation verwendet, die notwendigen und zugehörigen Attribute der OTLP/gRPC Notation (Proto) - welche für die technische Implementation der Anwendung relevant sind - sind der jeweils aktuell gültigen und veröffentlichten Beschreibung des OpenTelemetry Projektes zu entnehmen.

[neues Unterkapitel]

2.1.1.3.1 Kapitel 2.5.4.3.1 Eventlog

Beispiel eines allgemeinen OTEL Log records des Informationslevels.

```
{
      "resourceLogs": [{
            "resource": {
                  "attributes": [{
                        "key": "service.name",
                        "value": { "stringValue": "rs.popp.herst" }
                  },{
                        "key": "server.address",
                        "value": { "stringValue": "popp.abc.net" }
                  },{
```

```
                        "key": "lpi.ci_id",
                        "value": { "stringValue": "CI-12312345" }
                    },{
                        "key": "lpi.instance",
                        "value": { "stringValue": "a" }
                    }]
                },
                "scopeLogs": [{
                    "scope": {
                        "name": "com.example.MyLogger",
                        "version": "1.0.0"
                    },
                    "logRecords": [{
                        "timeUnixNano": 1678886400000000000,
                        "body": { "stringValue":
                        "event.popp_blocked_items"  },
                        "severityText": "product_info",
                        "severityNumber": 9,
                        "attributes": [{
                            "key": "number_of_elements",
                            "value": { "intValue": 5 }
                        },{
                            "key": "list_of_elements",
                            "value": { "stringValue": "fwlkdsamweafemh..."
                            }
                        }]
                    }]
                }]
            }]
        }
```

Die zum OpenTelemetry Log gehörenden Attribute für ein generisches Eventlog werden in der folgenden Tabelle beschrieben.

Tabelle 6: Tab_gemSpec_Perf_Telemetriedaten_Eventlog

Daten	OpenTelemetry Attribute (OTLP/JSON)	Pflichtfeld	Beschreibung
resourceLogs > resource > attributes			
Servicename	service.name	ja	Servicename, vom Hersteller vergeben **mit dem Präfix**

			"rs." Entspricht dem Service Name aus den Traces und hilft der gemeinsamen Identifizierung
HTTP FQDN	server.address	ja	Identifikation des Zielservers bei Multi-Host-Umgebungen und Erkennung von Anfragen an nicht autorisierte oder verdächtige Domänen.
CI-Identifikator	lpi.ci_id	ja	Identifikation des Konfiguration-Elementes "logische Produktinstanz", beginnend mit "CI" Entspricht "CI-ID" der Selbstauskunft v1
CI-Instanz	lpi.instance	Nein	Identifikation der logischen Produktinstanz, sofern redundante Instanzen vorhanden sind
resourceLogs > scopeLogs > logRecords			
Zeitstempel	timeUnixNano	Ja	Zeitstempel
Fester Wert zur Definition des Eventlogs	body	ja	produktspezifische Festlegung des Eventlog-Namens **mit dem Präfix "event."**
Fester Wert zur Definition des Eventlogs	severityText	ja	"product_info"
Fester Wert zur Definition des Eventlogs	severityNumber	ja	9(entspricht enum "SEVERITY_NUMBER_INFO")
resourceLogs > scopeLogs > logRecords > attributes			
Loginformation	Key-Value-Paar	Ja	produktspezifische Loginformationen - Key entspricht der zu überwachenden Messgröße des Fachdienstes - Value repräsentiert den Messwert z.B. "key" : "popp_blocked_Item", "value" : {"stringValue": "apdplmpmeemomw"} Die Festlegung hinsichtlich der Belegung von key und value ist der produkttypspezifischen Definition im

			entsprechenden Unterkapitel zu entnehmen.

[neues Unterkapitel]

2.1.1.3.2 Kapitel 2.5.4.3.2 Selbstauskunft

~~Hier ein~~ Beispiel eines OTEL Log records.

```
{
      "resourceLogs": [{
            "resource": {
                  "attributes": [{
                        "key": "service.name",
                        "value": { "stringValue": "rs.vsdm2.herst" }
                  },{
                        "key": "server.address",
                        "value": { "stringValue": "vsdm2.aok.net" }
                  },{
                        "key": "product.name",
                        "value": { "stringValue": "VSDM2 ProduktX" }
                  },{
                        "key": "product.code_name",
                        "value": { "stringValue": "VSDM2X" }
                  },{
                        "key": "product.type_name",
                        "value": { "stringValue": "PDT79" }
                  },{
                        "key": "lpi.ci_id",
                        "value": {"stringValue": "CI-9876543" }
                  },{
                        "key": "lpi.instance",
                        "value": { "stringValue": "a" }
                  },{
                        "key": "pod_name",
                        "value": { "stringValue": "VSDM2.pod" }
                  },{
                        "key": "product.vendor_id",
                        "value": { "stringValue": "HERST" }
                  },{
```

```
                        "key": "product.vendor_name",
                        "value": { "stringValue": "Hersteller ABC" }
                  }]
            },
            "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": "1.2.0-0" }
                        },{
                              "key": "product.type_version",
                              "value": { "stringValue": "1.0.3" }
                        },{
                              "key": "configuration.version",
                              "value": {"stringValue": "1.3.4-2" }
                        }]
                  }]
            }]
      }]
}
```

Die zum OpenTelemetry Log "product_info" gehörenden Attribute werden in der folgenden Tabelle beschrieben.

Tabelle 7: Tab_gemSpec_Perf_Telemetriedaten_product_info

Daten	OpenTelemetry Attribute (OTLP/JSON)	Pflichtfeld	Beschreibung
resourceLogs > resource > attributes			
Servicename	service.name	ja	Servicename, vom Hersteller vergeben **mit dem Präfix "rs."** Entspricht dem Service Name

			aus den Traces und hilft der gemeinsamen Identifizierung
HTTP FQDN	server.address	ja	Identifikation des Zielservers bei Multi-Host-Umgebungen und Erkennung von Anfragen an nicht autorisierte oder verdächtige Domänen.
Produktname	product.name	Ja	Produktname, vom Hersteller vergeben Entspricht "ProductName" der Selbstauskunft v1
Produktkurzname	product.code_name	Ja	Produktkurzname, vom Hersteller vergeben Entspricht "ProductCode" der Selbstauskunft v1
Produkttyp	product.type_name	Ja	Produkttyp mit numerischen Code, nach Vorgaben der gematik Entspricht "ProductType" der Selbstauskunft v1
CI-Identifikator	lpi.ci_id	Ja	Identifikation des Konfiguration-Elementes "logische Produktinstanz", beginnend mit "CI" Entspricht "CI-ID" der Selbstauskunft v1
CI-Instanz	lpi.instance	Nein	Identifikation der logischen Produktinstanz, sofern redundante Instanzen vorhanden sind
Pod Name	pod.name	Nein	Identifikation des laufenden Pods
Hersteller ID	product.vendor_id	Ja	Hersteller ID Entspricht "ProductVendorID" der Selbstauskunft v1
Herstellername	product.vendor_name	Ja	ausführlicher Herstellername Entspricht "ProductVendorName" der Selbstauskunft v1
resourceLogs > scopeLogs > logRecords			
Fester Wert zur Definition der	body	ja	"Selbstauskunft"

Selbstauskunft			
Fester Wert zur Definition der Selbstauskunft	severityText	ja	"product_info"
Fester Wert zur Definition der Selbstauskunft	severityNumber	ja	9(entspricht enum "SEVERITY_NUMBER_INFO ")
resourceLogs > scopeLogs > logRecords > attributes			
Produktversion	product.version	Ja	Aktuelle Produktversion, vom Hersteller vergeben Entspricht "ProductVersion" der Selbstauskunft v1
Produkttyp Version	product.type_version	Ja	Aktuelle Produkttypversion, nach Vorgaben der gematik Entspricht "ProductTypeVersion" der Selbstauskunft v1
Konfigurationsversion	configuration.version	Ja	Aktuelle Konfigurationsversion, vom Anbieter vergeben

[neues Unterkapitel]

2.1.1.3.3 Allgemeine Anforderungen zu Logs

[hier hin verschoben]

[redaktionelle Änderung des Afo-Titels --> Performance - Telemetriedatenlieferung**- Logs** - Selbstauskunft]

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 Änderungan 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.
[<=, Anb_TI-Flow_FD, TI-Flow_FD, Anbieterprofil_BL-1, DiPag_FD, Anb_PoPP_Service, VSDM_2_FD, PoPP_Service, Herst_TI-D_ZT, organ./betriebl. Eignung: Anbietererklärung, funkt. Eignung: Test Produkt/FA**]**

[neue Afo]

A_30264 -Performance - Telemetriedatenlieferung - Logs - Eventlog
Der Produkttyp MUSS die in der Tabelle "Tab_gemSpec_Perf_Telemetriedaten_Eventlog" gelisteten Parameter des Open Telemetry Logs "Eventlog" befüllen und die Information gemäß der produkttypspezifischen Festlegungen unmittelbar an den Telemetriedaten-Service übermitteln.**[**<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

Zuweisung der Anforderung an: PoPP_Service, funkt. Eignung: Test Produkt/FA

2.1.1.4 Kapitel 2.5.4.4 Metriken

~~Eine OpenTelemetry Metrik beschreibt quantitative Daten, die zur Überwachung und Auswertung eines TI 2.0-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 TI 2.0-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 TI 2.0-Fachdienste nicht ausreichend ist, können zusätzlich Schlüssel-Wert-Paare spezifiziert werden.~~

OpenTelemetry-Metriken dienen der standardisierten Erfassung, Aggregation und Übertragung quantitativer Telemetriedaten. Sie beschreiben den Zustand, das Verhalten und die Leistungsmerkmale eines Systems anhand zeitbezogener Messwerte und ermöglichen die Beobachtung von Anwendungen, Diensten und Infrastrukturkomponenten über Systemgrenzen hinweg. Das OpenTelemetry Metrics Data Model definiert hierfür ein einheitliches Datenmodell, bestehend aus Metriken, Datenpunkten, Attributen, Zeitstempeln und Aggregationen, das eine interoperable Verarbeitung durch Observability-Plattformen und Analysewerkzeuge ermöglicht.

Metriken MÜSSEN dem OpenTelemetry Metrics Data Model entsprechen und im Format des OpenTelemetry Protocol (OTLP) erzeugt und übertragen werden. Die Erzeugung der Metriken erfolgt über OpenTelemetry-Metrik-Instrumente, beispielsweise Counter, Gauge oder Histogram. Die durch diese Instrumente erfassten Messwerte werden durch das OpenTelemetry SDK im Resource Server des TI2.0 Dienstes aggregiert und anschließend als standardisierte OTLP-Metriken an den Telemetriedaten-Service des ZETA Guard zur Weiterleitung an die gematik bereitgestellt.

Diese Spezifikation definiert die Anforderungen an produktspezifische Metriken einschließlich der zugehörigen Attribute, Datentypen und semantischen Bedeutungen. Ziel ist die einheitliche Erfassung und Auswertung sicherheitsrelevanter sowie betrieblicher Metrikdaten über unterschiedliche TI2.0 Dienste hinweg. Die in diesem Kapitel getroffenen, allgemeinen Festlegungen, werden durch produkttypspezifische Ergänzungen in den jeweiligen Unterkapiteln der einzelnen Produkttypen vervollständigt.

Die in dieser Spezifikation beschriebenen OpenTelemetry-Attribute stellen eine gezielte Teilmenge der durch OpenTelemetry definierten Attribute dar. Diese Teilmenge ist für die in dieser Spezifikation beschriebenen Analyse-, Monitoring- und Sicherheitsanwendungsfälle verbindlich zu berücksichtigen.

Diese Spezifikation erhebt keinen Anspruch auf Vollständigkeit hinsichtlich aller durch OpenTelemetry definierten Metriken, Attribute, Instrumente oder semantischen Konventionen. Implementierungen DÜRFEN zusätzliche OpenTelemetry-konforme Attribute und Metriken erfassen und übertragen. Diese zusätzlichen Informationen sind jedoch nicht Gegenstand der normativen Anforderungen dieser Spezifikation, sofern sie die semantische Bedeutung, Auswertung oder Korrelation der hier definierten Metriken und Attribute nicht beeinträchtigen.

Die Darstellung von Metriken erfolgt in dieser Spezifikation grundsätzlich in der OTLP/JSON-Notation zur Verbesserung der Lesbarkeit. Für die technische Implementierung, Serialisierung und Übertragung sind die entsprechenden OTLP/gRPC-Protobuf-Definitionen maßgeblich. Die hierfür relevanten Nachrichtenstrukturen, Felder und Datentypen ergeben sich aus den jeweils gültigen und veröffentlichten Spezifikationen des OpenTelemetry-Projekts.

Soweit in den nachfolgenden Kapiteln nicht anders festgelegt, müssen alle Metriken mindestens einen eindeutigen Metriknamen, einen definierten Instrumententyp, einen Zeitbezug sowie die jeweils vorgeschriebenen Attribute enthalten. Die konkrete Ausgestaltung der einzelnen Metriktypen wird in den nachfolgenden Abschnitten zu den OpenTelemetry-Metrik-Instrumenten festgelegt.

Es ist außerdem zu berücksichtigen, dass "aufsummierte" Metrikdaten (hier Counter und Histogramme) einem festen Aggregationsfenster von **fünf Minuten** zugeordnet werden müssen. Die Zuordnung erfolgt über ein Epoch Window (Attribut epoch_window), das durchGanzzahldivision des Erfassungszeitpunkts (Genauigkeit in Sekunden) durch einen Umrechnungsfaktor gebildet wird. Da das Zwischenergebnis im Anschluss wieder mit dem gleichen Umrechnungsfaktor multipliziert wird, ergibt sich somit der Unix Timestamp für den Startzeitpunkt, des Aggregationszeitfensters. Die Verwendung des Epoch Windows ist notwendig, da durch die Aufsummierung innerhalb der Messgröße kein Zeitbezug enthalten ist. Bei Lieferung der Metrik an die Telemetriedatencloud wäre der Lieferzeitpunkt die einzige verbleibende zeitliche Bezugsgröße, wodurch eine Verzerrung der Messdaten erfolgen kann. Durch die Zuordnung zu klar definierten 5-Minuten-Zeitfenstern, ist der Übermittlungszeitpunkt für die Bewertung der Messgrößen nicht mehr relevant.

Berechnung des Epoch Unix Timestamps für den Startzeitpunkt des epoch_window:

Zeitstempel (UTC) 14.09.26 **13:14:29** = Epoch Unix Timestamp1789391669	>	**(1789391669 div 300) * 300** = 1789391400	>	Epoch Unix Timestamp 1789391400 = Zeitstempel (UTC) 14.09.26 **13:10:00**

Tabelle 8 : Tab_gemSpec_Perf_Telemetriedaten_metric_epoch_window_calc

Für die Übertragung im OpenTelemetry-Format wird das Attribut mit dem Key epoch_window verwendet. Das Epoch Window dient der fachlichen Auswertung und Korrelation von Metrikdaten. Der einzelne Messwert beschreibt somit die aufsummierte Anzahl aller beobachteten Ereignisse und deren Zuordnung zu einem spezifischen 5 Minuten Intervall. Für eine vollständige Auswertung müssen eventuell mehrere Messwerte zum gleichen Intervall zusammengezogen werden.

[neues Unterkapitel]

2.1.1.4.1 Kapitel 2.5.4.4.1 Counter-Metriken

Ein Counter dient zur Erfassung von Ereignissen oder Mengen, deren Wert ausschließlich zunimmt. Er repräsentiert einen monoton steigenden Zähler und MUSS für Messgrößen verwendet werden, bei denen ausschließlich positive Inkremente auftreten. Typische Anwendungsfälle sind die Anzahl verarbeiteter Anfragen, erzeugter Nachrichten, erfolgreicher Authentifizierungen, erkannter Sicherheitsereignisse oder aufgetretener Fehler. Ein Counter DARF nicht zur Abbildung von Zuständen oder Beständen verwendet

werden, die sich sowohl erhöhen als auch verringern können; hierfür sind andere Instrumenttypen, beispielsweise Gauge, zu verwenden.

Ein Counter wird im OpenTelemetry Metrics Data Model als Metrik des Typs Sum mit der Eigenschaft is_monotonic=true dargestellt. Die erfassten Werte repräsentieren die Summe aller seit einem definierten Startzeitpunkt beobachteten Ereignisse. Die Konfiguration des OTEL SDK im Resource Server des TI2.0 Dienstes SOLL so vorgenommen werden, dass die Übertragung mit kumulativer Temporalität erfolgt.

Counter-Metriken sollten so benannt werden, dass ihr aufsummierender Charakter unmittelbar erkennbar ist. Der Metrikname muss fachlich eindeutig sein und sollte die gezählte Größe beschreiben.

Das nachfolgende Beispiel beschreibt einen Counter zur Erfassung der erstellten PoPP-Token im definierten Zeitraum für vier verschiedene fachliche Szenarien.

Beispiel eines OTEL Metric records vom Typ Counter:

```
{
      "resourceMetrics": [{
            "resource": {
                  "attributes": [{
                        "key": "service.name",
                        "value": { "stringValue": "rs.poppservice" }
                  },{
                        "key": "server.address",
                        "value": { "stringValue": "popp.abc.net" }
                  },{
                        "key": "lpi.ci_id",
                        "value": {"stringValue": "CI-9876543" }
                  },{
                        "key": "lpi.instance",
                        "value": { "stringValue": "a" }
                  }]
            },
            "scopeMetrics": [{
                  "scope": {
                        "name": "otel-library",
                        "version": "1.0.0"
                  },
                  "metrics": [{
                        "name": "popp_stock_number",
                        "description": "numbers of various popp counter",
                        "unit": "1",
                        "sum": {
```

```
                                "aggregationTemporality":
AGGREGATION_TEMPORALITY_CUMULATIVE",
                                "isMonotonic": true,
                                "dataPoints": [{
                                      "attributes": [{
                                            "key": "status",
                                            "value": { "stringValue":
                                            "success" }
                                      },{
                                            "key": "type",
                                            "value": { "stringValue":
                                            "contact" }
                                      },{
                                            "key": "epoche_window",
                                            "value": { "intValue":1787303400 }
                                      }],
                                      "timeUnixNano": "1787303565000000000",
                                      "asInt": "42"
                                },{
                                      "attributes": [{
                                            "key": "status",
                                            "value": { "stringValue": "failed"
                                            }
                                      },{
                                            "key": "type",
                                            "value": { "stringValue":
                                            "contact" }
                                      },{
                                            "key": "epoche_window",
                                            "value": { "intValue":1787303400 }
                                      }],
                                      "timeUnixNano": "1787303565000000000",
                                      "asInt": "7"
                                },{
                                      ...
                                }]
                          }
                    }]
              }]
        }]
```

}

Im dargestellten Beispiel wurden zum Zeitpunkt timeUnixNano 42 PoPP Token erfolgreich kontaktbehaftet erzeugt und es gab 7 abgebrochene kontaktbehaftete Vorgänge (weitere Status sind unter ... verborgen).

Die zur OpenTelemetry Counter 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 Lieferung von Metriken für das jeweilige Produkt vorgesehen ist.

Tabelle 9: Tab_gemSpec_Perf_Telemetriedaten_metric_counter

Daten	OpenTelemetry Attribute (OTLP/JSON)	Beschreibung
resourceMetrics > resource > attributes		
Servicename	service.name	Servicename, vom Hersteller vergeben mit Präfix "rs." Entspricht dem Service Name aus den Traces und hilft der gemeinsamen Identifizierung
HTTP FQDN	server.address	Identifikation des Zielservers bei Multi-Host-Umgebungen und Erkennung von Anfragen an nicht autorisierte oder verdächtige Domänen.
CI-Identifikator	lpi.ci_id	Identifikation des Konfiguration-Elementes "logische Produktinstanz", beginnend mit "CI" Entspricht "CI-ID" der Selbstauskunft v1
CI-Instanz	lpi.instance	Identifikation der logischen Produktinstanz, sofern redundante Instanzen vorhanden sind
resourceMetrics > scopeMetrics > metric		
Name	name	Name der Metrik gemäß Festlegung im produktspezifischen Kapitel
Beschreibung	description	Beschreibung der Metrik gemäß Festlegung im produktspezifischen Kapitel
Einheit	unit	1 (für dimensionslose Zustandswerte)
resourceMetrics > scopeMetrics > metrics > sum		
Aggregation Typ	aggregationTemporarlity	AGGREGATION_TEMPORALITY_CUMULATIVE

Standardwert für "Monotonic"	isMonotonic	true
resourceMetrics > scopeMetrics > metrics > sum > dataPoints		
Summe	asInt	Metrikwert des jeweiligen summerischen Datenpunktes
Zeitpunkt	timeUnixNano	Zeitpunkt des Erfassens
resourceMetrics > scopeMetrics > metrics > sum > dataPoints > attributes		
Datenpunkt Definition	key / value Paare	Attribute fassen die Kriterien zusammen, auf die sich der Metrikwert bezieht. Die Benennung von keys und values der Labels erfolgt im produktspezifischen Kapitel.
Abfrage Zeitfenster	"key": "epoche_window"	key "Epoch_window" beschreibt als Epoch Unix Timestamp mit der Genauigkeit "Sekunde" (10-stellige Ganzzahl) den Beginn des 5 Minuten Intervals des Auswertungszeitraumes (absolutes und eindeutiges Intervall, mit einer Länge von 5 Minuten, beginnend mit 00:00:00 des jeweiligen Tages); value ist der entsprechende Wert

[neues Unterkapitel]

2.1.1.4.2 Kapitel 2.5.4.4.2 Gauge-Metriken

Ein Gauge dient zur Erfassung eines aktuellen Zustands oder Messwertes zu einem bestimmten Zeitpunkt. Im Gegensatz zu einem Counter, dessen Wert ausschließlich zunimmt, kann sich ein Gauge-Wert sowohl erhöhen als auch verringern. Ein Gauge repräsentiert somit eine Momentaufnahme einer Messgröße und eignet sich insbesondere zur Abbildung von Zuständen, Auslastungen, Beständen oder Kapazitäten. Typische Anwendungsfälle sind die Anzahl aktuell aktiver Sitzungen, die Auslastung von Ressourcen, die Anzahl offener Sicherheitsvorfälle oder der aktuelle Füllstand eines Zwischenspeichers.

Ein Gauge wird im OpenTelemetry Metrics Data Model durch den Metriktyp Gauge repräsentiert. Der übertragene Wert beschreibt den zum Messzeitpunkt gültigen Zustand eines Systems oder einer Komponente. Im Gegensatz zu Counter- und Histogram-Metriken erfolgt keine Aggregation über einen Zeitraum hinweg und es wird keine Temporalität definiert. Jeder Datenpunkt stellt den zuletzt ermittelten Wert der beobachteten Messgröße dar.

Die Verwendung eines Gauge sollte erfolgen, wenn die aktuelle Ausprägung einer Messgröße von Interesse ist und Änderungen sowohl in positiver als auch in negativer Richtung auftreten können.

Beispiel eines Gauge des Fachdienstes DiPag zur Bestandsaufnahme der Anzahl Rechnungen mit unterschiedlichen Statuswerten:

```
{
      "resourceMetrics": [{
            "resource": {
                  "attributes": [{
                        "key": "service.name",
                        "value": { "stringValue": "rs.dipagservice" }
                  },{
                        "key": "server.address",
                        "value": { "stringValue": "dipag.xyz.net" }
                  },{
                        "key": "lpi.ci_id",
                        "value": {"stringValue": "CI-9876543" }
                  },{
                        "key": "lpi.instance",
                        "value": { "stringValue": "a" }
                  }]
            },
            "scopeMetrics": [{
                  "scope": {
                        "name": "otel-library",
                        "version": "1.0.0"
                  },
                  "metrics": [{
                        "name": "dipag.invoice_status_count",
                        "description": "the number of stored invoices for
                        various statuses",
                        "unit": "1",
                        "gauge": {
                              "dataPoints": [{
                                    "timeUnixNano": "1718180300000000000",
                                    "asInt": "87",
                                    "attributes": [{
                                          "key": "status",
                                          "value": {"stringValue": "open"}
                                    }]
                              },{
                                    "timeUnixNano": "1718180300000000000",
```

```
                                    "asInt": "4",
                                    "attributes": [{
                                        "key": "status",
                                        "value": {"stringValue": "done"}
                                    }]
                                },{
                                    ...
                                }]
                            }
                        }]
                    }]
                }]
            }]
        }
```

Im dargestellten Beispiel liegen zum Zeitpunkt des Datenpunkts 87 Rechnungen im Status "offen", 4 im Status "erledigt" und (unter ... ausgeblendet) in anderen Status vor. Zu einem nachfolgenden Zeitpunkt können höhere oder niedrigere Werte übermittelt werden, ohne dass zwischen den Messwerten eine mathematische Beziehung bestehen muss.

Wie bei allen Metriken SOLLTEN Attribute mit hoher Kardinalität vermieden werden. Insbesondere Benutzerkennungen, Session-IDs, Request-IDs oder andere eindeutig identifizierende Werte SOLLTEN nicht als Attribute übertragen werden, da hierdurch eine unverhältnismäßig hohe Anzahl von Zeitreihen entstehen kann.

Die zur OpenTelemetry Gauge 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 Lieferung von Metriken für das jeweilige Produkt vorgesehen ist.

Tabelle 10: Tab_gemSpec_Perf_Telemetriedaten_metric_gauge

Daten	OpenTelemetry Attribute (OTLP/JSON)	Beschreibung
resourceMetrics > resource > attributes		
Servicename	service.name	Servicename, vom Hersteller vergeben mit Präfix "rs." Entspricht dem Service Name aus den Traces und hilft der gemeinsamen Identifizierung
HTTP FQDN	server.address	Identifikation des Zielservers bei Multi-Host-Umgebungen und Erkennung von Anfragen an nicht autorisierte oder verdächtige Domänen.
CI-Identifikator	lpi.ci_id	Identifikation des Konfiguration-Elementes "logische Produktinstanz",

		beginnend mit "CI" Entspricht "CI-ID" der Selbstauskunft v1
CI-Instanz	lpi.instance	Identifikation der logischen Produktinstanz, sofern redundante Instanzen vorhanden sind
resourceMetrics > scopeMetrics > metric		
Name	name	Name der Metrik gemäß Festlegung im produktspezifischen Kapitel
Beschreibung	description	Beschreibung der Metrik gemäß Festlegung im produktspezifischen Kapitel
Einheit	unit	Einheit ist die physikalische oder fachliche Größe der Metrik deren Verteilung gemessen wird und die Festlegung erfolgt im produktspezifischen Kapitel. Zum Beispiel für Zeitdauern die Einheiten ms oder s, für Datenmengen By und für dimensionslose Zustandswerte 1. Hinweis: OTEL verwendet UCUM-konforme Einheiten, für Byte also By. **Der Standard ist "1"**
resourceMetrics > scopeMetrics > metrics > gauge > dataPoints		
Anzahl der Operationen	asInt	Anzahl der zum Kriterien passenden Operationen
Zeitpunkt	timeUnixNano	Zeitpunkt des Erfassens
resourceMetrics > scopeMetrics > metrics > gauge > dataPoints > attributes		
Datenpunkt Definition	key / value Paare	Attribute fassen die Kriterien zusammen, auf die sich der Metrikwert bezieht. Die Benennung von keys und values der labels erfolgt im produktspezifischen Kapitel.

[neues Unterkapitel]

2.1.1.4.3 Kapitel 2.5.4.4.3 Histogram-Metriken

Ein Histogram dient zur Erfassung und Auswertung der Verteilung numerischer Messwerte innerhalb eines definierten Beobachtungszeitraums. Im Gegensatz zu einem Counter, der ausschließlich die Anzahl von Ereignissen zählt, ermöglicht ein Histogram Aussagen über

Häufigkeiten, Durchschnittswerte, Streuungen und die Verteilung von Messwerten. Histogramme sollten verwendet werden, wenn neben der Anzahl von Ereignissen auch deren Größenordnung oder Laufzeit von Bedeutung ist. Typische Anwendungsfälle sind Antwortzeiten von Diensten, Bearbeitungszeiten von Prozessen, Größen von Nachrichten oder Dateien sowie die Dauer sicherheitsrelevanter Vorgänge.

Ein Histogram aggregiert einzelne Messwerte in vordefinierte Wertebereiche (Buckets). Für jeden Bucket wird die Anzahl der enthaltenen Messwerte erfasst. Zusätzlich werden die Gesamtanzahl aller Beobachtungen (count) sowie die Summe aller Messwerte (sum) gespeichert. Optional können außerdem Minimal- und Maximalwerte übertragen werden. Dadurch können nachgelagerte Systeme Durchschnittswerte, Perzentile und weitere statistische Kennzahlen berechnen, ohne die ursprünglichen Einzelwerte verarbeiten zu müssen.

Die Definition der Bucket-Grenzen muss sich dabei an der fachlichen Bedeutung der Metrik orientieren. Die Anzahl der Buckets sollte so gewählt werden, dass eine ausreichende Aussagekraft erreicht wird, ohne eine unnötig hohe Anzahl von Zeitreihen oder ein übermäßiges Datenvolumen zu erzeugen. Für Metriken derselben fachlichen Bedeutung sollten innerhalb einer Produktfamilie einheitliche Bucket-Grenzen verwendet werden, um Vergleiche und aggregierte Auswertungen zu ermöglichen.

Histogramme werden im OpenTelemetry Metrics Data Model durch den Metriktyp Histogram repräsentiert. Ein Histogram-Datenpunkt enthält mindestens die Anzahl der Beobachtungen (count), die Summe aller Messwerte (sum), die definierten Bucket-Grenzen (explicitBounds) sowie die zugehörigen Häufigkeiten (bucketCounts).

Beispiel eines Histograms zur Erfassung von Authentifizierungszeiten in Millisekunden innerhalb eines fünfminütigen Beobachtungsfensters:

```
{
      "resourceMetrics": [{
            "resource": {
                  "attributes": [{
                        "key": "service.name",
                        "value": { "stringValue": "rs.vsdm2service" }
                  },{
                        "key": "server.address",
                        "value": { "stringValue": "vsdm2.aok.net" }
                  },{
                        "key": "lpi.ci_id",
                        "value": {"stringValue": "CI-9876543" }
                  },{
                        "key": "lpi.instance",
                        "value": { "stringValue": "a" }
                  }]
            },
            "scopeMetrics": [{
                  "scope": {
```

```
                    "name": "otel-library",
                    "version": "1.0.0"
                },
                "metrics": [{
                    "name": "vsdm2.request.duration",
                    "description": "Dauer erfolgreicher
Stammdatenabfragen",
                    "unit": "ms",
                    "histogram": {
                        "aggregationTemporality": "DELTA",
                        "dataPoints": [{
                            "startTimeUnixNano":
"1787562890000000000",
                            "timeUnixNano":
"1787563000000000000",
                            "count": "100",
                            "sum": 45230,
                            "min": 8,
                            "max": 920,
                            "explicitBounds": [ 10, 25, 50, 100,
250, 500, 1000 ],
                            "bucketCounts": [ 5, 24, 41, 11, 15, 3,
1, 0 ],
                            "attributes": [{
                                "key": "product.name",
                                "value": { "stringValue": "VSDM2-
Service" }
                            },{
                                "key": "deployment.environment",
                                "value": { "stringValue": "pu" }
                            },{
                                "key": "epoche_window",
                                "value": { "intValue":1787562900 }
                            }]
                        }]
                    }
                }]
            }]
        }]
}
```

Im dargestellten Beispiel wurden innerhalb des Aggregationszeitraums insgesamt 100 Stammdatenabfragen beobachtet. Die Summe aller gemessenen Laufzeiten beträgt 45.230 Millisekunden. Die Mehrheit der Messwerte liegt im Bereich zwischen 25 und 50 Millisekunden, während nur wenige Anfragen Laufzeiten oberhalb von 500 Millisekunden aufweisen. Auf Basis dieser Verteilung können Monitoring- und Analysewerkzeuge beispielsweise Median-, P95- oder P99-Perzentile berechnen.

Für Histogramme sollten ausschließlich numerische Messgrößen verwendet werden, deren Verteilung für die fachliche oder technische Bewertung relevant ist. Histogramme sollten nicht zur Erfassung reiner Ereigniszähler eingesetzt werden; hierfür sind Counter zu verwenden. Ebenso sollten Attribute mit hoher Kardinalität, insbesondere Benutzerkennungen, Session-IDs oder Request-IDs, vermieden werden, da hierdurch die Anzahl der erzeugten Zeitreihen erheblich ansteigen kann.

Die zur OpenTelemetry Histogram 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 Lieferung von Metriken für das jeweilige Produkt vorgesehen ist.

Tabelle 11: Tab_gemSpec_Perf_Telemetriedaten_metric_histogram

Daten	OpenTelemetry Attribute (OTLP/JSON)	Beschreibung
resourceMetrics > resource > attributes		
Servicename	service.name	Servicename, vom Hersteller vergeben mit Präfix "rs." Entspricht dem Service Name aus den Traces und hilft der gemeinsamen Identifizierung
HTTP FQDN	server.address	Identifikation des Zielservers bei Multi-Host-Umgebungen und Erkennung von Anfragen an nicht autorisierte oder verdächtige Domänen.
CI-Identifikator	lpi.ci_id	Identifikation des Konfiguration-Elementes "logische Produktinstanz", beginnend mit "CI" Entspricht "CI-ID" der Selbstauskunft v1
CI-Instanz	lpi.instance	Identifikation der logischen Produktinstanz, sofern redundante Instanzen vorhanden sind
resourceMetrics > scopeMetrics > metric		
Name	name	Name der Metrik gemäß Festlegung im produktspezifischen Kapitel
Beschreibung	description	Beschreibung der Metrik gemäß Festlegung im produktspezifischen

		Kapitel
Einheit	unit	Einheit ist die physikalische oder fachliche Größe der Metrik deren Verteilung gemessen wird und die Festlegung erfolgt im produktspezifischen Kapitel. Zum Beispiel "ms" oder "s" Hinweis: OTEL verwendet UCUM-konforme Einheiten, für Byte also By.
resourceMetrics > scopeMetrics > metrics > histogram		
Aggregation Typ	aggregationTemporarlity	DELTA
resourceMetrics > scopeMetrics > metrics > histogram > dataPoints		
Anzahl der Operationen	count	Anzahl der zum Kriterien passenden Operationen
Summe	sum	Summe aller Messwerte innerhalb des beobachteten Zeitfensters
Minimalwert	min	Minimalwert innerhalb des beobachteten Zeitfensters
Maximalwert	max	Maximalwert innerhalb des beobachteten Zeitfensters
Bucket-Grenzen	explicitBounds	definierten Bucket-Grenzen
Bucket-Häufigkeiten	bucketCounts	Häufigkeiten zugehörig zu den Bucket-Grenzen
Zeitpunkt	timeUnixNano	Zeitpunkt des Erfassens
resourceMetrics > scopeMetrics > metrics > histogram > dataPoints > attributes		
Abfrage Zeitfenster	"key": "epoche_window"	key "epoch_window" beschreibt als Epoch Unix Timestamp mit der Genauigkeit "Sekunde" (10-stellige Ganzzahl) den Beginn des 5 Minuten Intervals des Auswertungszeitraumes (absolutes und eindeutiges Intervall, mit einer Länge von 5 Minuten, beginnend mit 00:00:00 des jeweiligen Tages); value ist der entsprechende Wert
Datenpunkt Definition	key / value Paare	Attribute fassen die Kriterien zusammen, auf die sich der Metrikwert bezieht. Die Benennung von keys und

		values der labels erfolgt im produktspezifischen Kapitel.

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
            },
            "timeUnixNano": 1672531200000000000,
            "value": 42
        },
        {
            "labels":
            {
                "method": "POST",
                "status_code": 201
            },
            "timeUnixNano": 1672531200000000000,
```

~~"value": 7~~

~~}]~~

~~}]~~

~~}~~

~~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 12: 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~~
~~unit~~	~~1 (ist der Default und Standardwert für dimensionslose Metriken, die hier verwendet werden)~~
~~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~~

[neues Unterkapitel]

2.1.1.4.4 Allgemeine Anforderungen zu Metriken

[hier hin verschoben][Aktualisierung der Afo]

[alt]

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 Zeitintervallan 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.***[**<=, TI-Flow_FD, DiPag_FD, PoPP_Service, funkt. Eignung: Herstellererklärung**]**

[neu]

A_28785-01 -Performance - Telemetriedatenlieferung - Metriken
Der Produkttyp MUSS die in den Tabellen "Tab_gemSpec_Perf_Telemetriedaten_metric_counter", "Tab_gemSpec_Perf_Telemetriedaten_metric_gauge" oder "Tab_gemSpec_Perf_Telemetriedaten_metric_histogram" gelisteten Parameter, unter Berücksichtigung der produktspezifischen Vorgaben, als Open Telemetry Metriken verarbeiten und im festgelegten, konfigurierbaren Zeitintervallan den Telemetriedaten-Service überliefern, sofern entsprechende Metriken produktspezifisch definiert sind.

*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.***[**<=, TI-Flow_FD, DiPag_FD, PoPP_Service, funkt. Eignung: Herstellererklärung**]**

[neue Afo]

A_30265 -Performance - Telemetriedatenlieferung - Metriken - Erfassungszeitpunkt von Gauge-Metriken
Das Produkt MUSS sicherstellen, dass

- der Erfassungszeitpunkt für alle Gauge-Metriken identisch ist,
- ausgeschlossen ist, dass relevante Elemente durch Statuswechsel innerhalb einer Gauge-Metrik mehrfach oder nicht erfasst werden,
- die Erfassung für Gauge-Metriken täglich um 00:00 erfolgt, sofern keine andere Zeit vorgegeben ist.

[<=, TI-Flow_FD, DiPag_FD, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

Zuweisung der Anforderung an: TI-Flow_FD, DiPag_FD, PoPP_Service, funkt. Eignung: Test Produkt/FA

[Unterkapitel entfernen]

2.1.1.5 Kapitel 2.5.4.4 Anforderungen an Telemetriedatenlieferungen

[verschoben in Kapitel 2.5.4.1]

~~Folgende Anforderungen gelten für alle Telemetriedatenlieferungen.~~

A_27718-*

A_27724-*

A_28786-*

A_28766-*

A_28754-*

A_27722-*

A_28890-*

A_28891-*

A_28780-*

A_28782-*

A_28988-*

A_28779-*

[verschoben in Kapitel 2.5.4.2.1]

A_27723-*

A_29020-*

[verschoben in Kapitel 2.5.4.3.1]

A_27729-*

[verschoben in Kapitel 2.5.4.4.1]

A_28785-*

3 Änderungen in gemSpec_Perf 3.26

Kapitelname 3.26: VSDM 2 Fachdienst (PDT79)

Tabelle 13: 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~~

3.1.1 Änderungen in gemSpec_Perf 3.26.2 - Telemetriedaten - Spezifika VSDM 2

[neues Unterkapitel]

3.1.1.1 Kapitel 3.26.2.1 - Telemetriedaten - Spezifika VSDM 2 - Traces

[Afos werden unter das neue Unterkapitel einsortiert; redaktionelle Änderung der Afo-Titel:

A_26992* - Performance - Telemetriedaten - Spezifika VSDM 2 - Traces - Status,

A_26823* - Performance - Telemetriedaten - Spezifika VSDM 2 - Traces - Operation,

A_26825* - Performance - Telemetriedaten - Spezifika VSDM 2 - Traces - Message]

[zusätzlich Anpassung der Afo A_26823-03]

alt:

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 der Tabelle "Tab_gemSpec_Perf_VSDM: Performancerelevante UseCases".**[**<=, VSDM_2_FD, funkt. Eignung: Test Produkt/FA**]**

neu:

A_26823-04 -Performance - Telemetriedaten - Spezifika VSDM 2 - Traces - Operation
Der VSDM 2 Fachdienst MUSS Telemetriedaten für die Usecases der Tabelle "Tab_gemSpec_Perf_VSDM: Performancerelevante UseCases" liefern. Dazu ist es

zwingend notwendig, dass der Fachdienst die Funktionsaufrufe als Endpunkte bereitstellt und diese eindeutig über HTTP Methode (http.method, z.B. GET) und Route (http.route) aufgerufen werden.**[**<=, VSDM_2_FD, funkt. Eignung: Test Produkt/FA**]**

[neues Unterkapitel]

3.1.1.2 Kapitel 3.26.2.2 - Telemetriedaten - Spezifika VSDM 2 - Logs

Keine spezifischen Festlegungen.

[neues Unterkapitel]

3.1.1.3 Kapitel 3.26.2.3 - Telemetriedaten - Spezifika VSDM 2 - Metriken

Keine spezifischen Festlegungen.

[entfernen Unterkapitel]

~~**3.26.3 Bestandsdaten VSDM 2**~~

~~nicht definiert~~

[Unterkapitelnummerierung ändert sich von 2.26.4 auf 3.26.3]

3.1.2 3.26.3 Performance-Kenngrößen VSDM 2

Aus den Tabellen Tab_gemSpec_Perf_VSDM 2::O/A und Tab_gemSpec_Perf_VSDM 2_Performance-Kenngroessen werden die Schnittstellen S03-S06 (ZeroTrust-Schnittstellen des ZetaGuards) und deren dazugehörigen Kenngrößen entfernt, da sie an dieser Stelle keine Anwendung finden. Die Spalte "Berichtsformat-Alias" wird entfernt, da diese bei den Telemetriedaten keine Relevanz mehr hat.

4 Änderungen in gemSpec_Perf 3.27

Kapitelname 3.27: Digitale Patientenrechnung Fachdienst (PDT74)

Tabelle 14: 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~~

[neuer Titel des Unterkapitels "Telemetriedaten~~lieferung~~ - Spezifika Digitale Patientenrechnung"]

4.1.1 Änderungen in gemSpec_Perf 3.27.2 - Telemetriedaten - Spezifika Digitale Patientenrechnung

[neues Unterkapitel]

4.1.1.1 Kapitel 3.27.2.1 - Telemetriedaten - Spezifika Digitale Patientenrechnung - Traces

[Afos werden unter das neue Unterkapitel einsortiert; redaktionelle Änderung der Afo-Titel:

A_27365* - Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Traces - Status,

A_27366* - Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Traces - Operation,

A_27368* - Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Traces - Message,

A_28784* - Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Traces - Duration]

[zusätzlich Anpassung der Afo A_27366-02]

alt:

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".**[**<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

neu:

A_27366-03 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Traces - Operation
Der Digitale Patientenrechnung Fachdienst MUSS Telemetriedaten für die Usecases der Tabelle "Tab_gemSpec_Perf_VSDM: Performancerelevante UseCases" liefern. Dazu ist es zwingend notwendig, dass der Fachdienst die Funktionsaufrufe als Endpunkte bereitstellt und diese eindeutig über HTTP Methode (http.method, z.B. GET) und Route (http.route) aufgerufen werden.**[**<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

[neues Unterkapitel]

4.1.1.2 Kapitel 3.27.2.2 - Telemetriedaten - Spezifika Digitale Patientenrechnung - Logs

Keine spezifischen Festlegungen.

[neues Unterkapitel]

4.1.1.3 Kapitel 3.27.2.3 - Telemetriedaten - Spezifika Digitale Patientenrechnung - Metriken

[Afo A_27370 wird nicht mehr benötigt und kann entfallen]

~~A_27370* - Performance - Bestandsdaten - Spezifika Digitale Patientenrechnung - Anbieter~~

[Anpassung der Anforderung A_28787; Aufteilung in mehrere Metriken]

[alt]

A_28787 -Performance - Bestandsdaten - Spezifika Digitale Patientenrechnung - Metrik Rechnungsanzahl
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_invoice_count",
"description": "the number of stored invoices for various
statuses",
"unit": "1",
"type": "COUNTER",
"data_points": [...]
```

und unter "data_points" Datenpunkte für folgende Labels:

```
"status": "total" <-- Gesamtanzahl digitale Patientenrechnungen
"status": "new" <-- Anzahl neue digitale Patientenrechnungen
seit letzter Lieferung
"status": "open" <-- Anzahl digitale Patientenrechnungen im
Status "offen"
"status": "done" <-- Anzahl digitale Patientenrechnungen im
Status "erledigt"
```

```
"status": "trashed" <-- Anzahl digitale Patientenrechnungen im
Status "Papierkorb"
```

[<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

[neu]

A_28787-01 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Metrik Gauge - Rechnungsstatus
Der Digitale Patientenrechnung Fachdienst MUSS eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
"name": "dipag_invoice_status_count",
"description": "the number of stored invoices for various
statuses",
"unit": "1",
"data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
"key": "status", "value" : { "stringValue": "open" }
--> "asInt" = Anzahl digitale Patientenrechnungen im Status
"offen"

"key": "status", "value" : { "stringValue": "done" }
--> "asInt" = Anzahl digitale Patientenrechnungen im Status
"erledigt"

"key": "status", "value" : { "stringValue": "trashed" }
--> "asInt" = Anzahl digitale Patientenrechnungen im Status
"Papierkorb"
```

[<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

[neue Afos]

A_30228 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Metrik Gauge - Rechnungsanzahl global
Der Digitale Patientenrechnung Fachdienst MUSS eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
"name": "dipag_total_invoice_count",
"description": "the total number of stored invoices since
start",
"unit": "1",
"data_points": [...]
```

und unter "data_points" Datenpunkte mit folgendem Datum (die Definition des key-value-Paares ist nicht notwendig):

```
"asInt" = Anzahl digitale Patientenrechnungen seit Start der
Erfassung
```

[<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

Zuweisung an: DiPag_FD, funkt. Eignung: Test Produkt/FA

A_30226 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Metrik Counter - neue Rechnungen
Der Digitale Patientenrechnung Fachdienst MUSS eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_counter mit folgenden Informationen liefern:

```
"name": "dipag_new_invoice_count",
"description": "the number of new invoices",
"unit": "1"
"sum" > "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgendem Datum (die Definition des key-value-Paares ist nicht notwendig):

```
"asInt" = Anzahl neue digitale Patientenrechnungen
```

[<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

Zuweisung an: DiPag_FD, funkt. Eignung: Test Produkt/FA

[Anpassung der Anforderung A_28788]

[alt]

A_28788 -Performance - Bestandsdaten - Spezifika Digitale Patientenrechnung - Metrik Benutzergruppen
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_user_count",
"description": "the number and allocation of dipag accounts",
"unit": "1",
"type": "COUNTER",
"data_points": [...]
```

und unter "data_points" Datenpunkte für folgende Labels:

```
"user": "total" <-- Gesamtanzahl registrierter Nutzerkonten
"user": "insured_person" <-- Anzahl Versicherte
"user": "insurance" <-- Anzahl Kostenträger
"user": "invoiceissuer" <-- Anzahl Rechnungsersteller
```

[<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

[neu]

A_28788-01 -Performance - Telemetriedaten - Spezifika Digitale Patientenrechnung - Metrik Gauge - Benutzergruppen
Der Digitale Patientenrechnung Fachdienst MUSS eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
"name": "dipag_user_count",
"description": "the number and allocation of dipag accounts",
"unit": "1",
"data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
"key": "user", "value" : { "stringValue": "total" }
--> "asInt" = aktuelle Anzahl registrierter Nutzerkonten

"key": "user", "value" : { "stringValue": "insured_person" }
--> "asInt" = Anzahl Versicherte

"key": "user", "value" : { "stringValue": "insurance" }
--> "asInt" = Anzahl Kostenträger

"key": "user", "value" : { "stringValue": "invoiceissuer" }
--> "asInt" = Anzahl Rechnungsersteller
```

[<=, DiPag_FD, funkt. Eignung: Test Produkt/FA**]**

[entfernen Unterkapitel]

~~**3.27.3 Bestandsdatenlieferung - Spezifika Digitale Patientenrechnung**~~

[Unterkapitelnummerierung ändert sich von 2.27.4 auf 3.27.3]

4.1.2 3.27.3 Performance-Kenngrößen Digitale Patientenrechnung

Aus den Tabellen Tab_gemSpec_Perf_DiPag::O/A und Tab_gemSpec_Perf_DiPag_Performance-Kenngroessen werden die Schnittstellen A21-A24 (ZeroTrust-Schnittstellen des ZetaGuards) und deren dazugehörigen Kenngrößen entfernt, da sie an dieser Stelle keine Anwendung finden. Die Spalte "Berichtsformat-Alias" wird entfernt, da diese bei den Telemetriedaten keine Relevanz mehr hat.

5 Änderungen in gemSpec_Perf 3.28

[Umwandeln in "Tabelle"]

Table 1 Tab_gemSpec_Perf_PoPP_Service: Performancerelevante UseCases

UseCase	Fachdienstoperation	Beschreibung
~~PoPP.ZT1~~	~~GET /.well-known~~	~~ZETA: Abruf gültiger Autorisierungsserver~~
~~PoPP.ZT2~~	~~GET /nonce~~	~~ZETA: Nonce abrufen~~
~~PoPP.ZT3~~	~~POST /token <JWT Client Assert>~~	~~ZETA: Autorisierung ohne Refresh Token~~
~~PoPP.ZT4~~	~~POST /token <Refresh Token>~~	~~ZETA: Autorisierung mit Refresh Token~~
PoPP.1	~~POST /httpLoadPractitionerInformationRequest~~ GET /popp/practitioner/api/v1/token-generation-ehc	~~Anweisung zur Suche der Daten zu einer LEI~~ Anweisung zur Erstellung eines PoPP-Token nach Authentifizierung des Versicherten in der Umgebung des Telematik-ID-Inhabers
PoPP.2	~~POST /httpCheckInGIDRequest~~ GET/popp/patient/api/v1/mobile/fhirvzdsearch	~~Anweisung zur Erstellung eines PoPP-Token nach Authentifizierung des Versicherten über seine GesundheitsID~~ Anweisung zur Suche der Daten zu einer LEI im Verzeichnisdienst durch den Versicherten initiiert
PoPP.3	POST ~~/httpReadEgkRequest~~ /popp/practitioner/api/v1/ws-token-deliveries	~~Anweisung zur Prüfung einer eGK~~ Anweisung zur Erstellung eines PoPP-Token durch die LEI nach Authentifizierung des Versicherten mit Mobilgerät (erfolgt nach PoPP.7) (WebSocket)
PoPP.4	(reserviert) POST ~~/httpReadStatusPoPPRequest~~	(reserviert) ~~Anweisung für den Abruf des aktuellen Status zur~~

	/popp/patient/api/v1/mobile/checkin-egk	~~PoPP-Token Erstellung~~ Anweisung zur Prüfung einer eGK durch den Versicherten mit Mobilgerät
PoPP.5	POST /popp/patient/api/v1/mobile/checkin/poppStatus	Anweisung für den Abruf des aktuellen Status zur PoPP-Token Erstellung durch den Versicherten
PoPP.6	POST /popp/practitioner/api/v1/token-deliveries	Anweisung zur Erstellung eines PoPP-Token durch die LEI nach Authentifizierung des Versicherten mit Mobilgerät (erfolgt nach PoPP.7) (REST)
PoPP.7	POST /popp/patient/api/v1/mobile/checkin	Anweisung zum Check-in mitAuthentifizierung des Versicherten mit Mobilgerät bei einer LEI (Erstellung Datensatz beim PoPP-Service)

[neue Afo]

A_30519 -Performance - PoPP-Service - Datenlieferung an ZETA-Guard
Der Anbieter PoPP-Service 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.**[**<=, Anb_PoPP_Service, funkt. Eignung: Anbietererklärung**]**

Zuweisung an: Anb_PoPP_Service, funkt. Eignung: Anbietererklärung

[neuer Titel des Unterkapitels "Leistungsanforderungen ~~– Spezifika~~ PoPP-Service]

5.1.1 3.28.1 Leistungsanforderungen PoPP-Service

[Änderung der A_27030-02]

alt:

A_27030-02 -Performance - PoPP-Service - Bearbeitungszeit unter Last
Der PoPP-Service MUSS die Bearbeitungszeitvorgaben unter Last aus Tabelle "Tab_gemSpec_Perf_PoPP_Service: Last- und Bearbeitungszeitvorgaben" erfüllen.

Tabelle 15: Tab_gemSpec_Perf_PoPP_Service: Last- und Bearbeitungszeitvorgaben

Operation	Spitzenlast [1/sec]	Mittlere Bearbeitungszeit [msec]	Maximale Bearbeitungszeit [msec]	Erfüllungsquote [%]

PoPP.1	1400	600	1500	99,99
PoPP.2	1400	1400	1800	99,99
PoPP.3	1400	800	1200	99,99
PoPP.4	2000	200	350	99,99

[<=, PoPP_Service, funkt. Eignung: Herstellererklärung**]**

neu:

A_27030-04 -Performance - PoPP-Service - Bearbeitungszeit unter Last
Der PoPP-Service MUSS die Bearbeitungszeitvorgaben unter Last aus Tabelle "Tab_gemSpec_Perf_PoPP_Service: Last- und Bearbeitungszeitvorgaben" erfüllen.

Tabelle 16: Tab_gemSpec_Perf_PoPP_Service: Last- und Bearbeitungszeitvorgaben

Operation	Spitzenlast [1/sec]	Mittlere Bearbeitungszeit [msec]	Maximale Bearbeitungszeit [msec]	Erfüllungsquote [%]
PoPP.1	1400	600	1800	99,99
PoPP.3		600	1800	99,99
PoPP.6		600	1800	99,99
PoPP.2	1200	400	1500	99,99
PoPP.4 (reserviert)	1400	800	1200	99,99
PoPP.5	2000	200	350	99,99
PoPP.7	1200	600	1500	99,99

[<=, PoPP_Service, funkt. Eignung: Herstellererklärung**]**

5.1.2 Änderungen in gemSpec_Perf 3.28.2 - Telemetriedaten - Spezifika PoPP-Service

[Umbenennung des Unterkapitels: "PoPP-Service" statt "PoPP-Dienst"]

5.1.2.1 Kapitel 3.28.2.1 - Telemetriedaten - Spezifika PoPP-Service - Traces

[Afos werden unter das neue Unterkapitel einsortiert; redaktionelle Änderung der Afo-Titel:

A_28889* - Performance - Telemetriedaten - Spezifika PoPP-Service - Traces - RS Duration,

A_27316* - Performance - Telemetriedatenlieferung - Spezifika PoPP-Service - Traces - Status,

A_26333* - Performance - Telemetriedatenlieferung - Spezifika PoPP-Service - Traces - Operation,

A_26532* - Performance - Telemetriedatenlieferung - Spezifika PoPP-Service - Traces - Message]

[zusätzlich Anpassung der Afo A_26333-01]

alt:

A_26333-01 -Performance - Telemetriedatenlieferung - Spezifika PoPP-Service - Operation
Der PoPP-Service 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_PoPP_Service: Performancerelevante UseCases].**[**<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

neu:

A_26333-02 -Performance - Telemetriedaten - Spezifika PoPP-Service - Traces - Operation
Der PoPP-Service MUSS Telemetriedaten für die Usecases der Tabelle "Tab_gemSpec_Perf_VSDM: Performancerelevante UseCases" liefern. Dazu ist es zwingend notwendig, dass der Fachdienst die Funktionsaufrufe als Endpunkte bereitstellt und diese eindeutig über HTTP Methode (http.method, z.B. GET) und Route (http.route) aufgerufen werden.**[**<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

[Umbenennung des Unterkapitels: "PoPP-Service" statt "PoPP-Dienst"]

5.1.2.2 Kapitel 3.28.2.2 - Telemetriedaten - Spezifika PoPP-Service - Logs

~~Zu den Telemetriedaten Logs (ehemals Selbstauskunft) gibt es ***keine produktspezifischen*** (PoPP-Dienst) Ausprägungen und daher gelten nur die übergreifenden generischen Anteile abschließend.~~

[neue Anforderung]

A_30493 -Performance - Telemetriedaten - Spezifika PoPP-Service - Eventlog blocked Items
Der PoPP-Service MUSS bei den Telemetriedaten ein Log gem. Tab_gemSpec_Perf_Telemetriedaten_Eventlog,umgehend nach einer Blockierung, mit folgenden Informationen liefern:

- unter logRecords
  - `"body": { "stringValue": "event.popp_blocked_items" }`
- unter logRecords > attributes
  - `"key": "list_of_elements",`
  - `"value": { "stringValue": "???" }` mit ??? = Base64 codierte JSON Struktur des gemeldeten, blockierten Eintrages gemäß A_28899*

[<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

[neues Unterkapitel]

5.1.2.3 Kapitel 3.28.2.3 - Telemetriedaten - Spezifika PoPP-Service - Metriken

[Verschieben und Anpassung der Afo - Aufteilung in zwei Metriken, egkBlocked entfällt, weil bereits im Eventlog berücksichtigt]

[alt]

A_28835 -Telemetriedaten - Spezifika PoPP-Service - Metrik PoPP-vital
Der PoPP-Service MUSS bei den Telemetriedaten
(gem. Tab_gemSpec_Perf_Telemetriedaten_metric) unter "metrics" eine Metrik mit folgenden Informationen liefern:

```
"name": "popp_stock_number",
"description": "numbers of various popp counter",
"unit": "1",
"type": "GAUGE",
"data_points": [...]
```

und unter "data_points" Datenpunkte für folgende Labels:

```
"status": "egkHashImport" <-Anzahl der am Berichtstag durch
TSPs importierten Hashwert-Paare in die eGK-Hash-Datenbank
"status": "dailyContactTokenSucc" <-- Anzahl der am Berichtstag
erfolgreich ausgestellten PoPP-Token bei kontaktbehafteter
Nutzung einer eGK
"status": "dailyContactTokenFail" <-- Anzahl der am Berichtstag
abgebrochenen Anfragen bei kontaktbehafteter Nutzung einer eGK
"status": "dailyContactlessTokenSucc" <-- Anzahl der am
Berichtstag erfolgreich ausgestellten PoPP-Token bei
kontaktloser Nutzung einer eGK
"status": "dailyContactlessTokenFail" <-- Anzahl der am
Berichtstag abgebrochenen Anfragen bei kontaktloser Nutzung
einer eGK
"status": "egkBlocked" <-- Anzahl der gesperrten eGK
```

[<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

[neu]

A_28835-01 -Performance - Telemetriedaten - Spezifika PoPP-Service - Metrik Counter - PoPP-vital
Der PoPP-Service MUSS bei den Telemetriedaten eine Metrik
gem. Tab_gemSpec_Perf_Telemetriedaten_metric_counter mit folgenden Informationen liefern:

```
"name": "popp_stock_number",
"description": "numbers of various popp counter",
"unit": "1",
"sum" > "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

"key": "type", "value": { "stringValue" : "contact" }
"key": "status", "value": { "stringValue" : "success" }
--> "asInt" = Anzahl der erfolgreich ausgestellten PoPP-Token bei kontaktbehafteter Nutzung einer eGK

"key": "type", "value": { "stringValue" : "contact" }
"key": "status", "value": { "stringValue" : "failed" }
--> "asInt" = Anzahl der abgebrochenen Anfragen bei kontaktbehafteter Nutzung einer eGK

"key": "type", "value": { "stringValue" : "contactless" }
"key": "status", "value": { "stringValue" : "success" }
--> "asInt" = Anzahl der erfolgreich ausgestellten PoPP-Token bei kontaktloser Nutzung einer eGK

"key": "type", "value": { "stringValue" : "contactless" }
"key": "status", "value": { "stringValue" : "failed" }
--> "asInt" = Anzahl der abgebrochenen Anfragen bei kontaktloser Nutzung einer eGK

[<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

[neue Afo]

A_30325 -Performance - Telemetriedaten - Spezifika PoPP-Service - Metrik Counter - PoPP-hashImport
Der PoPP-Service MUSS bei den Telemetriedaten eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_counter mit folgenden Informationen liefern:

```
"name": "popp_hash_import_count",
"description": "numbers of hash imports",
"unit": "1",
"sum" > "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgendem Datum (die Definition des key-value-Paares ist nicht notwendig):

"asInt" = Anzahl der durch TSPs importierten Hashwert-Paare in die eGK-Hash-Datenbank

[<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

Zuweisung an: PoPP_Service, funkt. Eignung: Test Produkt/FA

[neue Afo]

A_30505 -Performance - Telemetriedaten - Spezifika PoPP-Service - Metrik Counter - PoPP-blocked_eGK
Der PoPP-Service MUSS bei den Telemetriedaten eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_counter mit folgenden Informationen liefern:

```
"name": "popp_blocked_egk_count",
"description": "numbers of blocked eGKs",
"unit": "1",
"sum" > "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgendem Datum (die Definition des key-value-Paares ist nicht notwendig):

```
"asInt" = Anzahl der blockierten eGKs
```

[<=, PoPP_Service, funkt. Eignung: Test Produkt/FA**]**

Zuweisung an: PoPP_Service, funkt. Eignung: Test Produkt/FA

[Afo A_28860 wird abgelöst]

~~A_28860 - Telemetriedaten - Spezifika PoPP-Service - Metrik PoPP-blockedItems~~

[entfernen Unterkapitel]

~~**3.28.3 Bestandsdaten PoPP**~~

[Unterkapitelnummerierung ändert sich von 2.28.4 auf 3.28.3; Änderung in "PoPP-Service"]

5.1.3 3.28.3 Performance-Kenngrößen PoPP-Service

Aus den Tabellen Tab_gemSpec_Perf_PoPP-Service::O/A und Tab_gemSpec_Perf_PoPP-Service_Performance-Kenngroessen werden die Schnittstellen S02-S05 (ZeroTrust-Schnittstellen des ZetaGuards) und deren dazugehörigen Kenngrößen entfernt, da sie an dieser Stelle keine Anwendung finden. Die Spalte "Berichtsformat-Alias" wird entfernt, da diese bei den Telemetriedaten keine Relevanz mehr hat.

6 Änderungen in gemSpec_Perf 3.29

Im Folgenden werden die spezifischen Leistungsanforderungen und Anforderungen des TI-Flow-Fachdienstes aufgeführt.

Tabelle 17 Tab_gemSpec_Perf_TI-Flow: Performancerelevante UseCases

Usecase	Fachdienstoperation	Beschreibung
TIF.UC_1_1	GET /Device	Fachdienstinformationen abrufen
TIF.UC_1_2	GET /metadata	FHIR CapabilityStatement abrufen
TIF.UC_2_1	POST /Task/$create	Vorgang durch Verordnenden erzeugen
TIF.UC_2_3	POST /Task/<id>/$activate	Vorgang durch Verordnenden einstellen
TIF.UC_2_5	POST /Task/<id>/$abort	Vorgang durch Verordnenden löschen
TIF.UC_3_1	GET /Task	Nachrichten durch Abgebenden übermitteln/empfangen
TIF.UC_3_2	POST /Task/<id>/$abort	Vorgang durch Versicherten löschen
TIF.UC_3_3	POST /Communication	Nachrichten durch Versicherten übermitteln/empfangen
TIF.UC_3_4	GET /Communication	Nachricht durch Versicherten empfangen
TIF.UC_3_5	GET /AuditEvent	Zugriffsprotokoll durch Versicherten abrufen
TIF.UC_3_6	GET /Task/<id>	Vorgang durch Vertreter abrufen
TIF.UC_3_7	GET /ChargeItem/<id>	Abrechnungsinformationen durch den Versicherten abrufen
TIF.UC_3_8	DELETE /Communication/<id>	Nachricht durch Versicherten löschen
TIF.UC_3_9	GET /MedicationDispense?<id>	Dispensierinformationen durch Versicherten abrufen
TIF.UC_3_10	GET /ChargeItem	Abrechnungsinformationen durch

		Versicherten abrufen
TIF.UC_3_11	DELETE /ChargeItem/<id>	Abrechnungsinformation durch Versicherten löschen
TIF.UC_3_12	PATCH /ChargeItem/<id>	Abrechnungsinformation durch Versicherten markieren
TIF.UC_3_13	GET /Consent	Einwilligung durch Versicherten abrufen
TIF.UC_3_14	POST /Consent	Einwilligung durch Versicherten erteilen
TIF.UC_3_15	DELETE /Consent	Einwilligung durch Versicherten widerrufen
TIF.UC_3_16	POST /$grant-eu-access-permission	Zugriffsberechtigung durch Versicherten erstellen
TIF.UC_3_17	DELETE /$revoke-eu-access-permission	Zugriffsberechtigung durch Versicherten löschen
TIF.UC_3_18	GET /$read-eu-access-permission	Zugriffsberechtigung durch Versicherten einsehen
TIF.UC_3_19	PATCH /Task/<id>	Vorgang markieren
TIF.UC_3_20	POST /pushers/set	App-Registrierung durch Versicherten erstellen
TIF.UC_3_21	GET /pushers	App-Registrierung durch Versicherten abrufen
TIF.UC_3_22	GET /channels	Channels durch Versicherten abrufen
TIF.UC_3_23	GET /channels/{pushkey)	Channels des Geräts durch Versicherten abrufen
TIF.UC_3_24	POST /channels/{pushkey}	Channels durch Versicherten konfigurieren
TIF.UC_4_1	POST /Task/<id>/$accept	Vorgang durch Abgebenden abrufen
TIF.UC_4_2	POST /Task/<id>/$reject	Vorgang durch Abgebenden zurückgeben
TIF.UC_4_3	POST /Task/<id>/$abort	Vorgang durch Abgebenden löschen

TIF.UC_4_4	POST /Task/<id>/$close	Vorgang durch Versicherten abrufen
TIF.UC_4_6	GET /Communication	Nachrichten durch Abgebenden empfangen
TIF.UC_4_7	POST /Communication	Abgabe durch Abgebenden vollziehen
TIF.UC_4_8	GET /Task/<id>?secret	Quittung durch Abgebenden erneut abrufen
TIF.UC_4_9	DELETE /Communication/<id>	Nachricht durch Abgebenden löschen
TIF.UC_4_10	GET /ChargeItem/<id>	Abrechnungsinformationen durch Abgebenden abrufen
TIF.UC_4_11	POST /ChargeItem	Abrechnungsinformationen durch Abgebenden bereitstellen
TIF.UC_4_12	GET /Task(PNW)	Vorgänge vom Versicherten durch Abgebenden abrufen
TIF.UC_4_13	PUT /ChargeItem/<id>	Abgabedatensatz durch Abgebenden aktualisieren
TIF.UC_4_14	POST /Subscription	Subscription durch Abgebenden registrieren
TIF.UC_4_15	GET /Task(PoPP)	Vorgänge vom Versicherten durch Abgebenden abrufen (PoPP)
TIF.UC_4_16	POST /Task/<id>/$dispense	Dispensierinformationen durch Abgebenden bereitstellen
TIF.UC_4_17	GET /Task/<id>?accesscode	Vorgang erneut abrufen
TIF.UC_4_19	POST /$get-eu-prescriptions mit Requesttype demographics	Demographische Daten eines Versicherten abrufen
TIF.UC_4_20	POST /$get-eu-prescriptions mit Requesttype e-prescriptions-list	Liste der einlösbaren Vorgänge eines Versicherten abrufen
TIF.UC_4_21	POST /$get-eu-prescriptions mit Requesttype e-prescriptions-retrieval	Liste ausgewählter Vorgänge eines Versicherten abrufen
TIF.UC_4_22	POST /Task/<id>/$eu-close	Abgabe eines Vorganges im europäischen Ausland

TIF.UC_5_1	Verordnungsdaten in Aktenkonto einstellen	Verordnungsdaten in Aktenkonto einstellen
TIF.UC_5_2	Löschinformation Verordnungsdaten an Aktenkonto übermitteln	Löschinformation Verordnungsdaten an Aktenkonto übermitteln
TIF.UC_5_3	Dispensierinformationen in Aktenkonto einstellen	Dispensierinformationen in Aktenkonto einstellen
TIF.UC_5_4	Löschinformation Dispensierinformationen an Aktenkonto übermitteln	Löschinformation Dispensierinformationen an Aktenkonto übermitteln
TIF.UC_5_5	ePA-Aktensystem ermitteln und Widerspruch prüfen	ePA-Aktensystem ermitteln und Widerspruch prüfen
TIF.UC_5_6	Login ePA-Aktensystem	Login ePA-Aktensystem
TIF.UC_5_7	POST /ords/rezepte/oauth/token	Login BfArM Service
TIF.UC_5_8	POST /ords/rezepte/t-rezept/v1	Digitaler Durchschlag E-T-Rezept beim BfArM einstellen
TIF.UC_5_9	GET [baseUrl]/search	FHIR VZID Suche
TIF.UC_5_10	POST /notifications	Notifications an Push Gateway versenden

[neue Afo]

A_30518 -Performance - TI-Flow-Fachdienst - Datenlieferung an ZETA-Guard
Der Anbieter TI-Flow-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.**[**<=, Anb_TI-Flow_FD, funkt. Eignung: Anbietererklärung**]**

Zuweisung an: Anb_TI-Flow-FD, funkt. Eignung Anbietererklärung

6.1.1 Änderungen in gemSpec_Perf 3.29.1 Leistungsanforderungen TI-Flow-Fachdienst

[Umbenennung Kapitel "Bearbeitungszeiten~~E-Rezept~~TI-Flow-Fachdienst"]

6.1.1.1 3.29.1.2 Bearbeitungszeiten TI-Flow-Fachdienst

[Umbenennung Kapitel "Performancevorgaben~~E-Rezept~~TI-Flow-Fachdienst"]

[in der Tabelle werden die ERP-Usecases in TIF-Usecases umbenannt]

6.1.1.2 3.29.1.3 Performancevorgaben TI-Flow-Fachdienst

[Afotitel wird redaktionell überarbeitet]

A_29785 Performance – TI-Flow-Fachdienst - ~~E-Rezept~~TI-Flow-Fachdienst - Bearbeitungszeit unter Last]

[in den Tabellen werden die ERP-Usecases in TIF-Usecases umbenannt]

[Umbenennung Kapitel "Telemetriedatenlieferung - Spezifika TI-Flow-Fachdienst"]

6.1.2 Änderungen in gemSpec_Perf 3.29.2 - Telemetriedaten - Spezifika TI-Flow-Fachdienst

[Afotitel wird redaktionell überarbeitet]

A_29796* - Performance - Telemetriedatenlieferung - Spezifika TI-Flow - Pseudonymisierte Telematik-ID

[neues Unterkapitel]

6.1.2.1 Kapitel 3.29.2.1 - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Traces

[Afos werden unter das neue Unterkapitel einsortiert; ggf. redaktionelle Änderung der Afo-Titel:

A_29796* - Performance - Telemetriedatenlieferung- Spezifika TI-Flow - Pseudonymisierte Telematik-ID

A_29798* - Performance - Telemetriedatenlieferung - Spezifika TI-Flow-Fachdienst - Traces - Status

A_29801* - Performance - Telemetriedatenlieferung - Spezifika TI-Flow-Fachdienst - Traces - Status - ePA Medication Service

A_29799* - Performance - Telemetriedatenlieferung - Spezifika TI-Flow-Fachdienst - Traces - Duration,

A_29797* - Performance - Telemetriedatenlieferung - Spezifika TI-Flow-Fachdienst - Traces - Operation,

[zusätzlich Anpassung der Afo A_29797]

alt:

A_29797 -Performance - Telemetriedatenlieferung - Spezifika TI-Flow - Operation
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen die Operationen der Tabelle "Tab_gemSpec_Perf_Berichtsformat_TI-Flow-Fachdienst_E-Rezept" berücksichtigen und den Use Case eindeutig über die Trace Parameter attributes:http_method und attributes:http_route kenntlich machen.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

neu:

A_29797-01 -Performance - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Traces - Operation
Der TI-Flow-Fachdienst MUSS Telemetriedaten für die Usecases der Tabelle "Tab_gemSpec_Perf_TI-Flow: Performancerelevante UseCases" liefern. Dazu ist es zwingend notwendig, dass der Fachdienst die Funktionsaufrufe als Endpunkte bereitstellt und diese eindeutig über HTTP Methode (http.method, z.B. GET) und Route (http.route) aufgerufen werden.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

[zusätzlich Anpassung der Afo A_29801]

alt:

A_29801 -Performance - Telemetriedatenlieferung - Spezifika TI-Flow - Status - ePA Medication Service
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferung im Trace Span zugehörig zum Medication Exporter bzgl. des Feldes "rs_statuscode" den vom ePA-Aktensystem im äußeren Http Request zurückgegeben HTTP Status Code übermitteln.
Der TI-Flow-Fachdienst MUSS, wenn der HTTP Status Code des äußeren HTTP Requests gleich 200 ist, den vom ePA-Aktensystem im inneren Http Request zurückgegeben HTTP Status Code übermitteln.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

neu:

A_29801-01 -Performance - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Traces - Status - ePA Medication Service
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für die Usecases TIF.UC_5_1, TIF.UC_5_2, TIF.UC_5_3, TIF.UC_5_4, TIF.UC_5_5 bzgl. des Feldes "rs.statuscode" die folgende Besonderheit im Zusammenhang mit der Kommunikation mit dem ePA-Aktensystem beachten:
1. Grundsätzlich ist der HTTP-Status Code des äußeren HTTP Requests zu verwenden
2. Sofern der HTTP Status Code des äußeren HTTP Requests gleich 200 ist, ist der HTTP Status Code des inneren Http Request zu verwenden.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

[neues Unterkapitel]

6.1.2.1.1 Kapitel 3.29.2.1.1 Telemetriedaten - Spezifika TI-Flow-Fachdienst - Traces - Message

Folgende spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte gelten für die Telemetrie-Keys in allen nachfolgenden Anforderungen.

Tabelle 18 Tab_TI-Flow_Telemetrie_Keys

key	Beschreibung	Datentyp
pdt.cid	Wert von zeta-client-data.client_id	String
pdt.pn.telematikID	Pseudonymisierte Telematik-ID der aufrufenden Institution	String
pdt.size	Größe des Requests in Kilobyte	Integer
pdt.bkdur	Zeit in ms für synchron ausgeführte Abfragen an andere Systeme (bspw. OCSP-Responder)	Integer
pdt.mvonr	Falls Flag für Mehrfachverordnung gesetzt: Nummerator der Teilverordnung	Integer
pdt.vnr	Vorgangsnummer Für Aktivitäten mit Bezug auf einen einzelnen Task: Task-ID Für Aktivitäten ohne Bezug auf einen Task oder mit Bezug auf mehrere Tasks: null	String
pdt.anr	Falls fehlerhafte Prüfung von identifier:ANR.value gemäß IG-TIFLOW-CORE-190: identifier:ANR.value	String

pdt.zanr	Falls fehlerhafte Prüfung von identifier:ZANR.value gemäß IG-TIFLOW-CORE-190: identifier:ZANR.value	String
pdt.wf	Für Aktivitäten mit Bezug auf einen einzelnen Task: flowtype	Integer
pdt.cov	Für Aktivitäten mit Bezug auf einen einzelnen Task: coverage typ	String
pdt.fsc	Falls POST /Task/<id>/$close für DiGA-Verordnung: Angabe, ob ein Freischaltcode in der Abgabeinformation enthalten ist	Boolean
pdt.tldur	Throttling duration Drosselungszeit eines Clientsystems in ms	Integer
pdt.gwp	FHIR Profil Version des gematik eRezept Workflow Package	String
pdt.gpr	FHIR Profil Version des gematik eRezept Patientenrechnung Package	String
pdt.kbv	FHIR Profil Version des KBV Verordnungsdatensatz Package	String
pdt.dav	FHIR Profil Version des DAV Abgabedatensatz Package	String
pdt.ext_sys	FQDN des externen Systems, das über den DNS-Resolver ermittelt wurde Dient zur Nachvollziehbarkeit im Fehlerfall, an welches System der Request gegangen ist	String
pdt.ec	der Inhalt des OperationOutcome.issue.details.code ohne das Prefix "MEDICATIONSVC_" für die UseCases 5_1 - 5_4, bzw. den Inhalt des healthcareProcess - erp-submission für den Usecase 5_5	String

[alle Message-Anforderungen werden dem TI_Flow_FD funkt Eignung: Test Produkt/FA zugeordnet]

A_30520 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_1_1 (GET /Device)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_1_1 (GET /Device) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{"key": "pdt.cid", "value": { "stringValue": "Text" }}
{"key": "pdt.pn.telematikID", "value": { "stringValue": "Text" }}
{"key": "pdt.size", "value": { "intValue": $$$ }}
{"key": "pdt.bkdur", "value": { "intValue": $$$ }}
{"key": "pdt.tldur", "value": { "intValue": $$$ }}
```

Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß

Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30521 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_1_2 (GET /metadata)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_1_2 (GET /metadata) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{"key": "pdt.cid", "value": { "stringValue": "Text" }}
{"key": "pdt.pn.telematikID", "value": { "stringValue": "Text" }}
{"key": "pdt.size", "value": { "intValue": $$$ }}
{"key": "pdt.bkdur", "value": { "intValue": $$$ }}
{"key": "pdt.tldur", "value": { "intValue": $$$ }}
```

Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.
[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30522 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_2_1 (POST /Task/$create)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_2_1 (POST /Task/$create) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30523 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_2_3 (POST /Task/<id>/$activate)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_2_3 (POST /Task/<id>/$activate) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30524 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_2_5 (POST /Task/<id>/$abort)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_2_5 (POST /Task/<id>/$abort) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30525 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_1 (GET /Task)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_1 (GET /Task) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30526 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_2 (POST /Task/<id>/$abort)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_2 (POST /Task/<id>/$abort) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30527 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_3 (POST /Communication)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_3 (POST /Communication) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30528 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_5 (GET /AuditEvent)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_5 (GET /AuditEvent) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30529 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_6 (GET /Task/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_6 (GET /Task/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30530 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_7 (GET /ChargeItem/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_7 (GET /ChargeItem/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30531 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_8 (DELETE /Communication/<id>)

Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_8 (DELETE /Communication/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30532 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_9 (GET /MedicationDispense/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_9 (GET /MedicationDispense/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30533 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_10 (GET /ChargeItem)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_10 (GET /ChargeItem) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30534 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_11 (DELETE /ChargeItem/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_11 (DELETE /ChargeItem/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30535 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_12 (PATCH /ChargeItem/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_12 (PATCH /ChargeItem/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30536 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_13 (GET /Consent)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_13 (GET /Consent) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30537 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_14 (POST /Consent)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_14 (POST /Consent) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30538 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_15 (DELETE /Consent)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_15 (DELETE /Consent) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30539 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_16 (POST /$grant-eu-access-permission)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_16 (POST /$grant-eu-access-permission) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30540 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_17 (DELETE /$revoke-eu-access-permission)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_17 (DELETE /$revoke-eu-access-permission) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30541 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_18 (GET /$read-eu-access-permission)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_18 (GET /$read-eu-access-permission) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30542 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_3_19 (PATCH /Task/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_3_19 (PATCH /Task/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30543 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_1 (POST /Task/<id>/$accept)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_1 (POST /Task/<id>/$accept) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30544 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_2 (POST /Task/<id>/$reject)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_2 (POST /Task/<id>/$reject) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30545 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_3 (POST /Task/<id>/$abort)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_3 (POST /Task/<id>/$abort) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30546 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_4 (POST /Task/<id>/$close)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_4 (POST /Task/<id>/$close) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30547 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_6 (GET /Communication)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_6 (GET /Communication) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30548 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_7 (POST /Communication)

Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_7 (POST /Communication) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30549 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_8 (GET /Task/<id>?secret)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_8 (GET /Task/<id>?secret) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30550 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_9 (DELETE /Communication/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_9 (DELETE /Communication/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30551 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_10 (GET /ChargeItem/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_10 (GET /ChargeItem/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30552 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_11 (POST /ChargeItem)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_11 (POST /ChargeItem) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30553 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_13 (PUT /ChargeItem/<id>)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_13 (PUT /ChargeItem/<id>) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30554 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_14 (POST /Subscription)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_14 (POST /Subscription) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30555 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_15 (GET /Task(PoPP))
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_15 (GET /Task(PoPP)) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30556 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_16 (POST /Task/<id>/$dispense)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_16 (POST /Task/<id>/$dispense) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30557 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_17 (GET /Task/<id>?accesscode)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_17 (GET /Task/<id>?accesscode) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30558 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_19 (POST /$get-eu-prescriptions mit Requesttype demographics)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_19 (POST /$get-eu-prescriptions mit Requesttype demographics) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30559 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_20 (POST /$get-eu-prescriptions mit Requesttype e-prescriptions-list)

Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_20 (POST /$get-eu-prescriptions mit Requesttype e-prescriptions-list) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30560 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_21 (POST /$get-eu-prescriptions mit Requesttype e-prescriptions-retrieval)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_21 (POST /$get-eu-prescriptions mit Requesttype e-prescriptions-retrieval) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30561 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_4_22 (POST /Task/<id>/$eu-close)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_4_22 (POST /Task/<id>/$eu-close) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30562 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_1 (Verordnungsdaten in Aktenkonto einstellen)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_1 (Verordnungsdaten in Aktenkonto einstellen) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{"key": "pdt.size", "value": { "intValue": $$$ }}
{"key": "pdt.bkdur", "value": { "intValue": $$$ }}
{"key": "pdt.vnr", "value": { "stringValue": "Text" }}
{"key": "pdt.wf", "value": { "intValue": $$$ }}
{"key": "pdt.cov", "value": { "stringValue": "Text" }}
{"key": "pdt.ext_sys", "value": { "stringValue": "Text" }}
{"key": "pdt.ec", "value": { "stringValue": "Text" }}
```

Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30563 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_2 (Löschinformation Verordnungsdaten an Aktenkonto übermitteln)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_2 (Löschinformation Verordnungsdaten an Aktenkonto übermitteln) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{"key": "pdt.size", "value": { "intValue": $$$ }}
{"key": "pdt.bkdur", "value": { "intValue": $$$ }}
{"key": "pdt.vnr", "value": { "stringValue": "Text" }}
{"key": "pdt.wf", "value": { "intValue": $$$ }}
```

```
{"key": "pdt.cov", "value": { "stringValue": "Text" }}
{"key": "pdt.ext_sys", "value": { "stringValue": "Text" }}
{"key": "pdt.ec", "value": { "stringValue": "Text" }}
```

Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30564 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_3 (Dispensierinformationen in Aktenkonto einstellen)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_3 (Dispensierinformationen in Aktenkonto einstellen) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{"key": "pdt.size", "value": { "intValue": $$$ }}
{"key": "pdt.bkdur", "value": { "intValue": $$$ }}
{"key": "pdt.vnr", "value": { "stringValue": "Text" }}
{"key": "pdt.wf", "value": { "intValue": $$$ }}
{"key": "pdt.cov", "value": { "stringValue": "Text" }}
{"key": "pdt.ext_sys", "value": { "stringValue": "Text" }}
{"key": "pdt.ec", "value": { "stringValue": "Text" }}
```

Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30565 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_4 (Löschinformation Dispensierinformationen an Aktenkonto übermitteln)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_4 (Löschinformation Dispensierinformationen an Aktenkonto übermitteln) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{"key": "pdt.size", "value": { "intValue": $$$ }}
{"key": "pdt.bkdur", "value": { "intValue": $$$ }}
{"key": "pdt.vnr", "value": { "stringValue": "Text" }}
{"key": "pdt.wf", "value": { "intValue": $$$ }}
{"key": "pdt.cov", "value": { "stringValue": "Text" }}
{"key": "pdt.ext_sys", "value": { "stringValue": "Text" }}
{"key": "pdt.ec", "value": { "stringValue": "Text" }}
```

Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.**[**<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30566 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_5 (ePA-Aktensystem ermitteln und Widerspruch prüfen)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_5 (ePA-Aktensystem ermitteln und Widerspruch prüfen) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

```
{"key": "pdt.size", "value": { "intValue": $$$ }}
{"key": "pdt.bkdur", "value": { "intValue": $$$ }}
{"key": "pdt.vnr", "value": { "stringValue": "Text" }}
{"key": "pdt.wf", "value": { "intValue": $$$ }}
{"key": "pdt.cov", "value": { "stringValue": "Text" }}
```

```
      {"key": "pdt.ext_sys", "value": { "stringValue": "Text" }}
      {"key": "pdt.ec", "value": { "stringValue": "Text" }}
```

Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.
[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30567 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_6 (Login ePA-Aktensystem)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_6 (Login ePA-Aktensystem) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30568 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_7 (POST /ords/rezepte/oauth/token)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_7 (POST /ords/rezepte/oauth/token) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30569 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_8 (POST /ords/rezepte/t-rezept/v1)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_8 (POST /ords/rezepte/t-rezept/v1) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30570 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_9 (GET [baseUrl]/search)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_9 (GET [baseUrl]/search) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

A_30571 -Performance - Telemetriedaten - Spezifika TI-Flow - TIF.UC_5_10 (POST /notifications)
Der TI-Flow-Fachdienst MUSS bei Telemetriedatenlieferungen für den Use Case TIF.UC_5_10 (POST /notifications) produkttypspezifische Informationen als key / value Paar im Trace mitliefern.

<pre>Die spezifischen Festlegungen hinsichtlich der Attribute und der Inhalte sind gemäß Tab_TI-Flow_Telemetrie_Keys zu berücksichtigen.[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA]

[Afos ablösen da durch Neuaufteilung anderweitig berücksichtigt]

~~A_29800 - Performance - Telemetriedaten - Spezifika TI-Flow - Message~~

~~A_29802 - Performance - Telemetriedatenlieferung - Spezifika TI-Flow - Duration - ePA Medication Service~~

~~A_29804 - Performance - Telemetriedaten - Spezifika TI-Flow - Message - ePA Medication Service~~

~~A_29805 - Performance - Telemetriedatenlieferung - Spezifika TI-Flow - Duration - BfArM Webservice~~

~~A_29806 - Performance - Telemetriedaten - Spezifika TI-Flow - Message - BfArM Webservice~~

~~A_29807 - Performance - Telemetriedatenlieferung - Spezifika TI-Flow - Duration - Push Gateway~~

~~A_29808 - Performance - Telemetriedaten - Spezifika TI-Flow - Message - Push Gateway~~

~~A_29803 - Performance - Telemetriedatenlieferung - Spezifika TI-Flow - Message - Error-Component - ePA Medication Service~~

[neues Unterkapitel]

6.1.2.2 Kapitel 3.29.2.2 - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Logs

Keine spezifischen Festlegungen.

[neues Unterkapitel]

6.1.2.3 Kapitel 3.29.2.3 - Telemetriedaten - Spezifika TI-Flow-Fachdienst - Metriken

[Afo A_29811 wird nicht mehr benötigt und kann entfallen]

~~A_29811 - Performance - Bestandsdaten - Spezifika TI-Flow - Anbieter~~

[alle Metriken werden dem Produkt TI-Flow_FD, funkt. Eignung: Test Produkt/FA zugeordnet]

A_30573 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Workflow_160

Der TI-Flow-Fachdienst MUSS eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
"name": "tiflow_workflow_160_count",
"description": "the number of stored workflows with flowtype 160 for various statuses",
"unit": "1",
"data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
"key": "status", "value": { "stringValue" : "ready" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit FlowType=160 im Status "ready"
```

```
"key": "status", "value": { "stringValue" : "inprogress" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=160 im Status "in-progress"

"key": "status", "value": { "stringValue" : "completed" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=160 im Status "completed"

"key": "status", "value": { "stringValue" : "cancelled" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=160 im Status "cancelled"

"key": "status", "value": { "stringValue" : "deleteready" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=160 zur Löschung am Folgetag im Status "ready"

"key": "status", "value": { "stringValue" : "deleteinprogress" }
--> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=160 zur Löschung am Folgetag im Status "in-progress"
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30574 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Workflow_162

Der TI-Flow-Fachdienst MUSS eine Metrik gem. Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
"name": "tiflow_workflow_162_count",
"description": "the number of stored workflows with flowtype 162 for
various statuses",
"unit": "1",
"data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
"key": "status", "value": { "stringValue" : "ready" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=162 im Status "ready"

"key": "status", "value": { "stringValue" : "inprogress" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=162 im Status "in-progress"

"key": "status", "value": { "stringValue" : "completed" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=162 im Status "completed"

"key": "status", "value": { "stringValue" : "cancelled" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=162 im Status "cancelled"

"key": "status", "value": { "stringValue" : "deleteready" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=162 zur Löschung am Folgetag im Status "ready"

"key": "status", "value": { "stringValue" : "deleteinprogress" }
```

```
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=162 zur Löschung am Folgetag im Status "in-progress"
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30575 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Workflow_166
Der TI-Flow-Fachdienst MUSS eine Metrik gem.
Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
      "name": "tiflow_workflow_166_count",
      "description": "the number of stored workflows with flowtype 166 for
      various statuses",
      "unit": "1",
      "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
      "key": "status", "value": { "stringValue" : "ready" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=166 im Status "ready"

      "key": "status", "value": { "stringValue" : "inprogress" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=166 im Status "in-progress"

      "key": "status", "value": { "stringValue" : "completed" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=166 im Status "completed"

      "key": "status", "value": { "stringValue" : "cancelled" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=166 im Status "cancelled"

      "key": "status", "value": { "stringValue" : "deleteready" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=166 zur Löschung am Folgetag im Status "ready"

      "key": "status", "value": { "stringValue" : "deleteinprogress" }
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=166 zur Löschung am Folgetag im Status "in-progress"
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30576 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Workflow_169
Der TI-Flow-Fachdienst MUSS eine Metrik gem.
Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
      "name": "tiflow_workflow_169_count",
      "description": "the number of stored workflows with flowtype 169 for
      various statuses",
      "unit": "1",
      "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
"key": "status", "value": { "stringValue" : "ready" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=169 im Status "ready"

"key": "status", "value": { "stringValue" : "inprogress" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=169 im Status "in-progress"

"key": "status", "value": { "stringValue" : "completed" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=169 im Status "completed"

"key": "status", "value": { "stringValue" : "cancelled" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=169 im Status "cancelled"

"key": "status", "value": { "stringValue" : "deleteready" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=169 zur Löschung am Folgetag im Status "ready"

"key": "status", "value": { "stringValue" : "deleteinprogress" }
--> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=169 zur Löschung am Folgetag im Status "in-progress"
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30577 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Workflow_200
Der TI-Flow-Fachdienst MUSS eine Metrik gem.
Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
"name": "tiflow_workflow_200_count",
"description": "the number of stored workflows with flowtype 200 for
various statuses",
"unit": "1",
"data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
"key": "status", "value": { "stringValue" : "ready" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=200 im Status "ready"

"key": "status", "value": { "stringValue" : "inprogress" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=200 im Status "in-progress"

"key": "status", "value": { "stringValue" : "completed" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
FlowType=200 im Status "completed"

"key": "status", "value": { "stringValue" : "cancelled" }
--> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
```

```
      FlowType=200 im Status "cancelled"

      "key": "status", "value": { "stringValue" : "deleteready" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=200 zur Löschung am Folgetag im Status "ready"

      "key": "status", "value": { "stringValue" : "deleteinprogress" }
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=200 zur Löschung am Folgetag im Status "in-progress"
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30578 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Workflow_209
Der TI-Flow-Fachdienst MUSS eine Metrik gem.
Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
      "name": "tiflow_workflow_209_count",
      "description": "the number of stored workflows with flowtype 209 for
      various statuses",
      "unit": "1",
      "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
      "key": "status", "value": { "stringValue" : "ready" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=209 im Status "ready"

      "key": "status", "value": { "stringValue" : "inprogress" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=209 im Status "in-progress"

      "key": "status", "value": { "stringValue" : "completed" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=209 im Status "completed"

      "key": "status", "value": { "stringValue" : "cancelled" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=209 im Status "cancelled"

      "key": "status", "value": { "stringValue" : "deleteready" }
      --> "asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=209 zur Löschung am Folgetag im Status "ready"

      "key": "status", "value": { "stringValue" : "deleteinprogress" }
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen Tasks mit
      FlowType=209 zur Löschung am Folgetag im Status "in-progress"
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30579 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Exporter_eML
Der TI-Flow-Fachdienst MUSS eine Metrik gem.
Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
      "name": "tiflow_eml_count",
      "description": "the number of stored items in eml exporter queue",
      "unit": "1",
      "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
      "key": "status", "value": { "stringValue" : "kvnr" }
      --> "'asInt" = Anzahl der KVNRs zum Abfragezeitpunkt mit Einträgen in
      der Queue des eML Exporter

      "key": "status", "value": { "stringValue" : "pending" }
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen
      Übermittlungsaufträge im Status "pending" in der Queue des eML
      Exporter

      "key": "status", "value": { "stringValue" : "dLQ" }
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen
      Übermittlungsaufträge im Status "deadLetterQueue" in der Queue des
      eML Exporter
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

A_30580 -Performance - Telemetriedaten - Spezifika TI-Flow - Metrik Gauge - Exporter_BfArM
Der TI-Flow-Fachdienst MUSS eine Metrik gem.
Tab_gemSpec_Perf_Telemetriedaten_metric_gauge mit folgenden Informationen liefern:

```
      "name": "tiflow_bfarm_count",
      "description": "the number of stored items in bfarm exporter queue",
      "unit": "1",
      "data_points": [...]
```

und unter "data_points" Datenpunkte mit folgenden key-value-Paaren:

```
      "key": "status", "value": { "stringValue" : "kvnr" }
      --> "'asInt" = Anzahl der KVNRs zum Abfragezeitpunkt mit Einträgen in
      der Queue des BfArM Exporter

      "key": "status", "value": { "stringValue" : "pending" }
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen
      Übermittlungsaufträge im Status "pending" in der Queue des BfArM
      Exporter

      "key": "status", "value": { "stringValue" : "dLQ" }
      --> "'asInt" = Anzahl der zum Abfragezeitpunkt vorhandenen
      Übermittlungsaufträge im Status "deadLetterQueue" in der Queue des
      BfArM Exporter
```

[<=, TI-Flow_FD, funkt. Eignung: Test Produkt/FA**]**

[Metriken wurden in Gauge-Metriken umgewandelt - Afos der Counter-Metriken werden nicht mehr benötigt und können entfallen:]

~~A_29812 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Workflow 160~~

~~A_29813 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Workflow 162~~

~~A_29814 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Workflow 166~~

~~A_29815 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Workflow 169~~

~~A_29816 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Workflow 200~~

~~A_29817 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Workflow 209~~

~~A_29818 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Exporter eML~~

~~A_29819 - Performance - Bestandsdaten - Spezifika TI-Flow - Metrik Anzahl Exporter BfArM~~

[Kapitel 3.29.3.1 wird entfernt]

~~3.29.3.1 Service Level Werte~~

[neues Unterkapitel]

6.1.3 3.29.3 Performance-Kenngrößen TI-Flow-Fachdienst

Schnittstellen::Operation bzw. Anwendungsfall

Tabelle 19 Tab_gemSpec_Perf_TI-Flow::O/A

Produkttyp / Anwendungstyp	ID	Anwendungsfall	Beschreibung
PDT87	A01	TIF*	Generische Schnittstelle
PDT87	A02	TIF.UC_2_1	Vorgang durch Verordnenden erzeugen
PDT87	A03	TIF.UC_2_3	Vorgang durch Verordnenden einstellen
PDT87	A04	TIF.UC_3_1	Nachrichten durch Abgebenden übermitteln/empfangen
PDT87	A05	TIF.UC_3_3	Nachrichten durch Versicherten übermitteln/empfangen
PDT87	A06	TIF.UC_3_6	Vorgang durch Vertreter abrufen
PDT87	A07	TIF.UC_4_1	Vorgang durch Abgebenden abrufen
PDT87	A08	TIF.UC_4_4	Vorgang durch Versicherten abrufen
PDT87	A09	TIF.UC_4_7	Abgabe durch Abgebenden vollziehen

PDT87	A11	TIF.UC_3_7	Abrechnungsinformationen durch den Versicherten abrufen
PDT87	A12	TIF.UC_4_11	Abrechnungsinformationen durch Abgebenden bereitstellen
PDT87	A16	TIF.UC_4_10	Abrechnungsinformationen durch Abgebenden abrufen
PDT87	A17	TIF.UC_4_12	Vorgänge vom Versicherten durch Abgebenden abrufen
PDT87	A18	TIF.UC_1_1	Fachdienstinformationen abrufen
PDT87	A19	TIF.UC_1_2	FHIR CapabilityStatement abrufen
PDT87	A20	TIF.UC_2_5	Vorgang durch Verordnenden löschen
PDT87	A21	TIF.UC_3_2	Vorgang durch Versicherten löschen
PDT87	A22	TIF.UC_3_4	Nachricht durch Versicherten empfangen
PDT87	A23	TIF.UC_3_5	Zugriffsprotokoll durch Versicherten abrufen
PDT87	A24	TIF.UC_3_8	Nachricht durch Versicherten löschen
PDT87	A25	TIF.UC_3_9	Dispensierinformationen durch Versicherten abrufen
PDT87	A26	TIF.UC_3_10	Abrechnungsinformationen durch Versicherten abrufen
PDT87	A27	TIF.UC_3_11	Abrechnungsinformation durch Versicherten löschen
PDT87	A28	TIF.UC_3_12	Abrechnungsinformation durch Versicherten markieren
PDT87	A29	TIF.UC_3_13	Einwilligung durch Versicherten abrufen
PDT87	A30	TIF.UC_3_14	Einwilligung durch Versicherten erteilen

PDT87	A31	TIF.UC_3_15	Einwilligung durch Versicherten widerrufen
PDT87	A32	TIF.UC_4_2	Vorgang durch Abgebenden zurückgeben
PDT87	A33	TIF.UC_4_3	Vorgang durch Abgebenden löschen
PDT87	A34	TIF.UC_4_6	Nachrichten durch Abgebenden empfangen
PDT87	A35	TIF.UC_4_8	Quittung durch Abgebenden erneut abrufen
PDT87	A36	TIF.UC_4_9	Nachricht durch Abgebenden löschen
PDT87	A37	TIF.UC_4_13	Abgabedatensatz durch Abgebenden aktualisieren
PDT87	A38	TIF.UC_4_14	Subscription durch Abgebenden registrieren
PDT87	A44	TIF.UC_4_15	Vorgänge vom Versicherten durch Abgebenden abrufen (PoPP)
PDT87	A47	TIF.UC_4_16	Dispensierinformationen durch Abgebenden bereitstellen
PDT87	A48	TIF.UC_4_17	Vorgang erneut abrufen
PDT87	A52	TIF.UC_5_1	Verordnungsdaten in ePA Medication Service einstellen
PDT87	A53	TIF.UC_5_2	Löschinformation Verordnungsdaten an ePA Medication Service übermitteln
PDT87	A54	TIF.UC_5_3	Dispensierinformationen in ePA Medication Service einstellen
PDT87	A55	TIF.UC_5_4	Löschinformation Dispensierinformationen an ePA Medication Service übermitteln
PDT87	A56	TIF.UC_5_5	ePA-Aktensystem ermitteln und Widerspruch prüfen
PDT87	A57	TIF.UC_5_6	Login ePA-Aktensystem

PDT87	A59	TIF.UC_3_16	Zugriffsberechtigung durch Versicherten erstellen
PDT87	A60	TIF.UC_3_17	Zugriffsberechtigung durch Versicherten löschen
PDT87	A61	TIF.UC_3_18	Zugriffsberechtigung durch Versicherten einsehen
PDT87	A62	TIF.UC_4_19	Demographische Daten eines Versicherten abrufen
PDT87	A63	TIF.UC_4_20	Liste der einlösbaren Vorgänge eines Versicherten abrufen
PDT87	A64	TIF.UC_4_21	Liste ausgewählter Vorgänge eines Versicherten abrufen
PDT87	A65	TIF.UC_4_22	Abgabe eines Vorganges im europäischen Ausland
PDT87	A67	TIF.UC_5_7	Login BfArM Service
PDT87	A68	TIF.UC_5_8	Durchschlag E-T-Rezept beim BfArM einstellen
PDT87	A69	TIF.UC_5_9	FHIR VZID Suche
PDT87	A70	TIF.UC_3_19	Vorgang markieren
PDT87	A71	TIF.UC_3_20	App-Registrierung durch Versicherten
PDT87	A72	TIF.UC_3_21	App-Registrierung durch Versicherten abrufen
PDT87	A73	TIF.UC_3_22	Channels durch Versicherten abrufen
PDT87	A74	TIF.UC_3_23	Channels des Geräts durch Versicherten abrufen
PDT87	A75	TIF.UC_3_24	Channels durch Versicherten konfigurieren
PDT87	A76	TIF.UC_5_10	Notifications an Push Gateway versenden

[Inhalte aus ehemaligem Kapitel 3.29.3.1 werden eingefügt, "ERP" in "TIF" umgewandelt und fehlende Kenngrößen ergänzt]

Performance-Kenngrößen / SL-Werte

Der Bearbeitungszeitraum T für die aufgeführten Soll-Werte beträgt ein Kalendermonat.

Die Bildung der Performance-Kenngrößen basiert auf folgenden PG-Schemata: PG-Schema-I

Tabelle 20 : Tab_gemSpec_Perf_TI-Flow_Performance-Kenngroessen

Performance-Kenngröße (PKG-ID)	Beschreibung	berechnet aus (Betriebsdaten, Probing)	SL-Wert (Soll-Wert)	min / max	Normative Referenz
TI-Flow-Fachdienst - PDT87 - (~~ERP~~TIF*)					
PDT87-A01-D3-G14	Relative Verfügbarkeit im Betrachtungszeitraum zur Hauptzeit inkl. Wartung. [%*1000]	Probing	99990 (PU)	min	A_29788
PDT87-A01-D3-G14	Relative Verfügbarkeit im Betrachtungszeitraum zur Hauptzeit inkl. Wartung. [%*1000]	Probing	99500 (RU, TU)	min	kein SL
PDT87-A01-D3-G16	Relative Verfügbarkeit im Betrachtungszeitraum zur Nebenzeit inkl. Wartung. [%*1000]	Probing	99970 (PU)	min	A_29788
PDT87-A01-D3-G16	Relative Verfügbarkeit im Betrachtungszeitraum zur Nebenzeit inkl. Wartung. [%*1000]	Probing	85000 (RU, TU)	min	kein SL
...					