

# Softwareproduktinformation

SPI Fabasoft app.telemetry 2025

**Fabasoft**<sup>®</sup>

Copyright ©

Fabasoft R&D GmbH, A-4020 Linz, 2024.

Alle Rechte vorbehalten. Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Marken der jeweiligen Hersteller.

Diese Unterlagen sind streng vertraulich. Durch die Übermittlung und Präsentation dieser Unterlagen alleine werden keine Rechte an unserer Software, an unseren Dienstleistungen und Dienstleistungsergebnissen oder sonstigen geschützten Rechten begründet.

Die Weitergabe, Veröffentlichung oder Vervielfältigung ist nicht gestattet.

# Inhalt

<b>1 Einleitung</b>	<b>6</b>
<b>2 Systemarchitektur</b>	<b>7</b>
2.1 Fabasoft app.telemetry Server	7
2.2 Fabasoft app.telemetry Agent	8
2.3 Fabasoft app.telemetry WebAPI	8
2.4 Fabasoft app.telemetry Client	8
<b>3 Application-Service-Level-Management</b>	<b>9</b>
3.1 Services	9
3.2 Servicegruppen	9
3.3 Tests von Leistungsindikatoren	9
3.3.1 Linux Plattformen	9
3.3.2 Microsoft Windows Plattformen	10
3.4 Verfügbarkeitstests	10
3.4.1 Linux Plattformen	10
3.4.2 Microsoft Windows Plattformen	11
3.4.3 Service Level Berechnung	12
3.5 Tests des Antwortzeitverhaltens	12
3.5.1 Durchschnittliches Antwortzeitverhalten	12
3.5.2 Prozentsatz der Anfragen im Grenzbereich	12
3.6 Benachrichtigungen	12
3.6.1 Benachrichtigungskanäle	12
3.6.2 Benachrichtigungskontakt	12
<b>4 Fabasoft app.telemetry Integration (SDK)</b>	<b>13</b>
4.1 Registrierung	13
4.1.1 Applikationsregistrierung	13
4.1.2 Filterregistrierung	13
4.1.3 Modulregistrierung	13
4.1.4 Eventregistrierung	13
4.2 Events	13
4.3 Kontext	14
4.4 Automatische Software-Telemetry Session	14
<b>5 Software-Telemetry</b>	<b>15</b>
5.1 Online Software-Telemetry Logs	15
5.2 Software-Telemetry Sessions	15
5.2.1 Explizite Software-Telemetry Session	15
5.2.2 Analyse von Software-Telemetry Sessions	15

5.3 Persistierung von Software-Telemetry Logs.....	15
5.4 Auswertung von Software-Telemetry Logs.....	16
5.5 Service Desk Integration.....	16
5.6 Syslog Integration Modul .....	16
5.7 Telemetry Rohdaten Analyse .....	16
<b>6 Unterstützte Plattformen und Systemanforderungen .....</b>	<b>17</b>
6.1 Fabasoft app.telemetry Agent .....	17
6.1.1 Unterstützte Plattformen .....	17
6.1.2 Systemanforderungen.....	17
6.1.3 Voraussetzungen .....	17
6.2 Fabasoft app.telemetry WebAPI.....	19
6.2.1 Unterstützte Plattformen .....	19
6.2.2 Systemanforderungen.....	19
6.2.3 Voraussetzungen .....	19
6.3 Fabasoft app.telemetry Server .....	21
6.3.1 Unterstützte Plattformen .....	21
6.3.2 Systemanforderungen.....	21
6.3.3 Voraussetzungen .....	21
6.3.4 Unterstützte Datenbanksysteme.....	22
6.4 Fabasoft app.telemetry Client.....	23
6.4.1 Unterstützte Plattformen .....	23
6.4.2 Systemanforderungen.....	23
6.5 Fabasoft app.telemetry Integration.....	23
6.5.1 Fabasoft app.telemetry Integration für C++, C#, Java.....	23
6.5.2 Fabasoft app.telemetry Integration für JavaScript .....	24
6.5.3 Fabasoft app.telemetry Integration für Syslog .....	24
6.5.4 Fabasoft app.telemetry Support Matrix.....	24
6.6 Fabasoft app.telemetry SNMP Agent .....	26
6.6.1 Unterstützte Plattformen .....	26
6.6.2 Systemanforderungen.....	26
6.6.3 Voraussetzungen .....	26
6.7 Fabasoft app.telemetry Proxy Server .....	27
6.7.1 Unterstützte Plattformen .....	27
6.7.2 Systemanforderungen.....	27
6.7.3 Voraussetzungen .....	27
6.8 Allgemeines .....	28
<b>7 Einsatzbedingungen .....</b>	<b>29</b>



# 1 Einleitung

Ungestörtes und schnelles Funktionieren geschäftskritischer Anwendungen ist zunehmend ausschlaggebend für die Schlagkräftigkeit der Organisationen und die Motivation ihrer Mitarbeiterinnen und Mitarbeiter.

Die Produkte von Fabasoft app.telemetry messen und dokumentieren die objektive Gebrauchstauglichkeit (Service Level) komplexer Softwaresysteme (Applikationen) in Form von Kennzahlen wie Antwortzeitverhalten (Performance) und Verfügbarkeit (Availability).

Fabasoft app.telemetry stellt dabei durchgängig und übersichtlich den Bezug her zwischen den Anwendungsfällen (Use Cases) auf der Nutzerseite und den damit verbundenen systemtechnischen Vorgängen und Parametern – über alle Hardware- und Software-Ebenen der beteiligten Systeme hinweg.

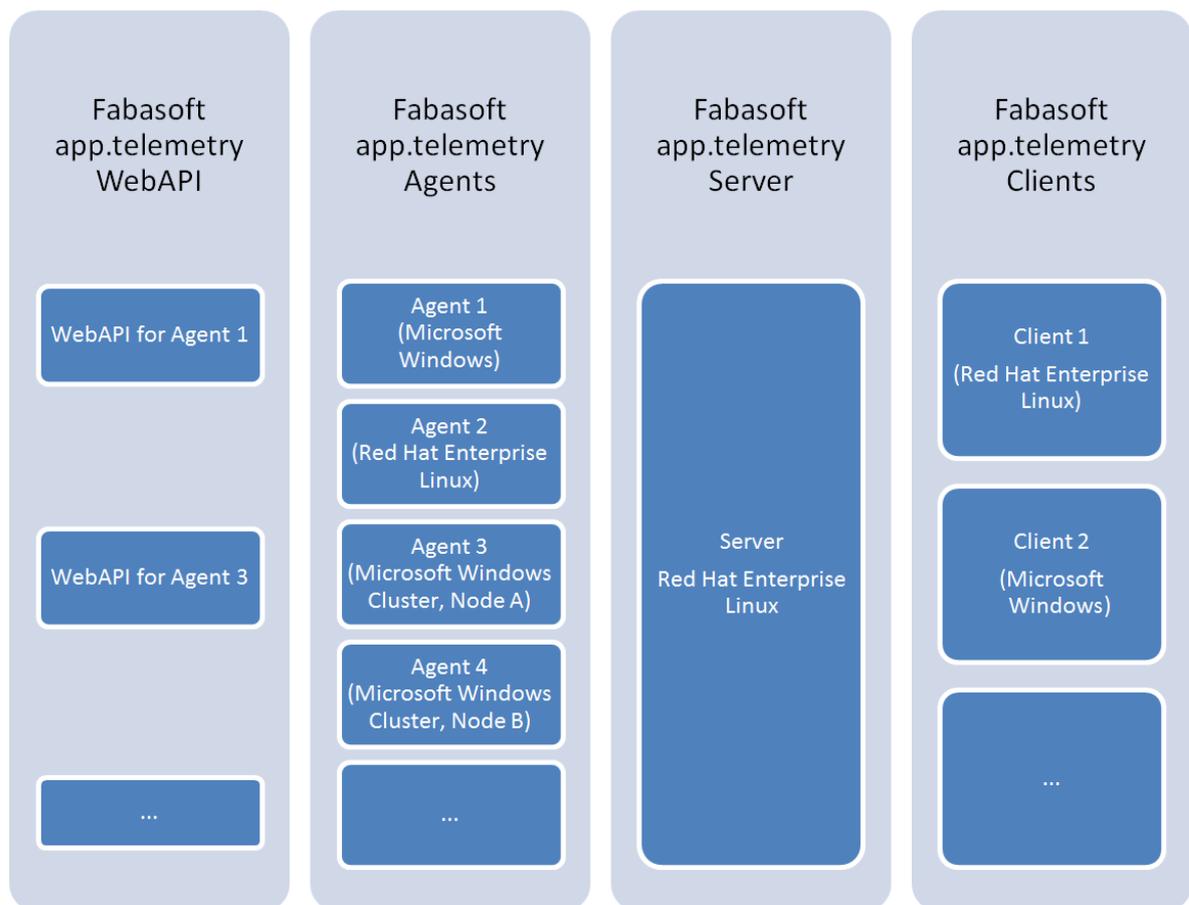
Wirksame Optimierung, vorausschauende Planung und Betriebsführung, Störungseingrenzung und -behebung werden durch diese einzigartige Technologie mit bislang unerreichter Effizienz ermöglicht. Anwenderorientierte Benutzerbetreuung erhält durch die gemeinsame inhaltliche Gesprächsbasis (Use Cases) zwischen Anwender und Systemspezialist eine völlig neue Qualität.

## 2 Systemarchitektur

Eine einfache Fabasoft app.telemetry Installation umfasst die folgenden Komponenten:

- Einen Fabasoft app.telemetry Server,
- Fabasoft app.telemetry Agents,
- Fabasoft app.telemetry WebAPI und
- Fabasoft app.telemetry Clients.

Die folgende Darstellung illustriert ein Beispiel einer Fabasoft app.telemetry Installation in einer heterogenen Systemumgebung:



### 2.1 Fabasoft app.telemetry Server

Ein Fabasoft app.telemetry Server ist ein Microsoft Windows Service oder ein Linux Daemon, der auf einem dedizierten Managementserver betrieben wird. Er bildet die zentrale Komponente von Fabasoft app.telemetry. Der Server speichert die vollständige Fabasoft app.telemetry Konfiguration, koordiniert die konfigurierten Fabasoft app.telemetry Agents, verwaltet Software-Telemetry Sessions und steuert die Evaluierung von Tests von Leistungsindikatoren und Verfügbarkeitstests. Der Fabasoft app.telemetry Server kann von den Clients über einen Webserver (der auf dem Server, auf dem auch der Fabasoft app.telemetry Server läuft, betrieben wird) wie beispielsweise Apache oder Microsoft Internet Information Services angesprochen werden. Die unterstützten Authentisierungsmöglichkeiten sind Microsoft Windows Integrated Authentication (ausschließlich für Microsoft Windows Plattformen) oder Basic Authentication.

## 2.2 Fabasoft app.telemetry Agent

Ein Fabasoft app.telemetry Agent ist ein Microsoft Windows Service oder Linux Daemon, der für die Lieferung von Leistungsindikatoren, Verfügbarkeitsinformationen und Software-Telemetry Daten an den Fabasoft app.telemetry Server verantwortlich ist. Der Fabasoft app.telemetry Agent muss auf den Systemen installiert werden, die potentiell Daten für Fabasoft app.telemetry liefern können.

## 2.3 Fabasoft app.telemetry WebAPI

Das Fabasoft app.telemetry WebAPI ist ein Webservice für Apache oder Microsoft Internet Information Services, das für die Lieferung von Software-Telemetry Daten aus JavaScript Instrumentierungen an einen Fabasoft app.telemetry Agent verantwortlich ist. Das Fabasoft app.telemetry WebAPI muss auf Systemen installiert werden, auf denen auch ein Fabasoft app.telemetry Agent betrieben wird. Das Fabasoft app.telemetry WebAPI ist Voraussetzung für die Verwendung der Fabasoft app.telemetry Integration auf Basis JavaScript und muss daher für die entsprechenden Web-Browser verfügbar sein.

## 2.4 Fabasoft app.telemetry Client

Der Fabasoft app.telemetry Client ist eine webbasierte Client-Anwendung, die über einen der gängigen Webbrowser Clients bedient werden kann. Über den Fabasoft app.telemetry Client können die Anwender die Fabasoft app.telemetry Funktionalität nutzen.

## 3 Application-Service-Level-Management

### 3.1 Services

Fabasoftware app.telemetry stellt Funktionalitäten im Bereich Application-Service-Level-Management auf Basis von Services zur Verfügung. Ein Service ist ein Fabasoftware app.telemetry Element, das für die Konfiguration von zwei Arten von Tests (Checks) verwendet werden kann, Tests von Leistungsindikatoren (Counter-Checks) und Verfügbarkeitschecks (Availability-Checks). Abhängig von den konfigurierten Tests können sich Services im Zustand OK, WARNING oder CRITICAL befinden.

Für jedes Service ist ein Fabasoftware app.telemetry Agent definiert, der für die Ausführung der konfigurierten Tests verantwortlich ist.

### 3.2 Servicegruppen

Services können in Servicegruppen organisiert werden. Eine Servicegruppe ist ein Fabasoftware app.telemetry Element, das ein oder mehrere Services oder auch ein oder mehrere Servicegruppen enthalten kann. Abhängig vom Zustand der enthaltenen Services kann sich eine Servicegruppe im Zustand OK, WARNING oder CRITICAL befinden.

### 3.3 Tests von Leistungsindikatoren

Fabasoftware app.telemetry Agents können für das periodische Lesen von Leistungsindikatoren und die Überwachung der so gelesenen Werte konfiguriert werden. Abhängig von den gelesenen Werten und von konfigurierbaren Grenzwerten kann ein Test eines Leistungsindikators die Zustände OK, WARNING oder CRITICAL melden.

Für heterogene Umgebungen unterstützt Fabasoftware app.telemetry die Verwendung von Counter-Definitionen. Eine Counter-Definition beschreibt, wie ein Leistungsindikator in den jeweiligen unterstützten Betriebssystemen gelesen werden kann. Somit ist ein, auf einer Counter-Definition basierender Test eines Leistungsindikators auf allen Betriebssystemen verfügbar, die in der entsprechenden Counter-Definition berücksichtigt wurden.

#### 3.3.1 Linux Plattformen

Fabasoftware app.telemetry Agents, die auf Linux Plattformen (gemäß Kapitel 6.1 Fabasoftware app.telemetry Agent) betrieben werden, unterstützen Tests für die folgenden, lokal (auf dem Server, auf dem der Fabasoftware app.telemetry Agent betrieben wird) verfügbaren Leistungsindikatoren:

##### 3.3.1.1 SNMP-Counter Check

Ein Fabasoftware app.telemetry Agent kann alle lokal verfügbaren SNMP Werte sowie nicht lokale SNMP Werte, die von SNMP Services zur Verfügung gestellt werden, lesen. Der Zugriff auf nicht lokale SNMP Werte kann durch die Verwendung von Proxy-Agents konfiguriert werden.

##### 3.3.1.2 Linux /proc Counter Check

Ein Fabasoftware app.telemetry Agent unterstützt für laufende Prozesse und im Betriebssystem sichtbare Prozessoren die Leistungsindikatoren "*% idle time*", "*% privileged time*", "*% processor time*" und "*% user time*" sofern diese Leistungsindikatoren durch das `/proc` Filesystem zur Verfügung stehen.

### 3.3.1.3 Counter Check mit Formel

Leistungsindikatoren können durch die Verwendung von einfachen Formeln (Addition, Subtraktion, Multiplikation oder Division) auch dynamisch unter Einbeziehung von unterstützten SNMP Werten, Linux `/proc` Werten und/oder Counter-Definitionen berechnet werden.

## 3.3.2 Microsoft Windows Plattformen

Fabasoftware app.telemetry Agents, die auf Microsoft Windows Plattformen (gemäß Kapitel 6.1 Fabasoftware app.telemetry Agent) betrieben werden, unterstützen Tests für die folgenden, lokal (auf dem Server, auf dem der Fabasoftware app.telemetry Agent betrieben wird) verfügbaren Leistungsindikatoren:

### 3.3.2.1 SNMP-Counter Check

Ein Fabasoftware app.telemetry Agent kann alle lokal verfügbaren SNMP Werte sowie nicht lokale SNMP Werte, die von SNMP Services zur Verfügung gestellt werden, lesen. Der Zugriff auf nicht lokale SNMP Werte kann durch die Verwendung von Proxy-Agents konfiguriert werden.

### 3.3.2.2 Microsoft Windows Performance-Counter Check

Ein Fabasoftware app.telemetry Agent kann alle lokal verfügbaren Microsoft Windows Leistungsindikatoren (Microsoft Windows Performance-Counter) lesen.

### 3.3.2.3 Counter Check mit Formel

Leistungsindikatoren können durch die Verwendung von einfachen Formeln (Addition, Subtraktion, Multiplikation oder Division) auch dynamisch unter Einbeziehung von Microsoft Windows Performance-Counter und/oder Counter-Definitionen berechnet werden.

## 3.4 Verfügbarkeitstests

Fabasoftware app.telemetry Agents können für die periodische Ausführung von Verfügbarkeitstests und die Überwachung der entsprechenden Ergebnisse konfiguriert werden. Abhängig vom Ergebnis kann ein Verfügbarkeitstest die Zustände OK, WARNING oder CRITICAL melden.

### 3.4.1 Linux Plattformen

Fabasoftware app.telemetry Agents, die auf Linux Plattformen (gemäß Kapitel 6.1 Fabasoftware app.telemetry Agent) betrieben werden, unterstützen die folgenden, lokal (auf dem Server, auf dem der Fabasoftware app.telemetry Agent betrieben wird) verfügbaren Verfügbarkeitschecks:

#### 3.4.1.1 Linux Process Availability Check

Ein Fabasoftware app.telemetry Agent kann die Verfügbarkeit eines Linux Prozesses durch die Suche nach einer konfigurierbaren Befehlszeile (Command-Line) im `/proc` Filesystem prüfen.

#### 3.4.1.2 Red Hat Cluster Suite Service Availability Check

Ein Fabasoftware app.telemetry Agent kann die Verfügbarkeit eines Red Hat Cluster-Services durch das Auslesen des Zustands des entsprechenden Cluster-Services vom Red Hat Cluster-Manager prüfen.

### 3.4.1.3 Webservice Availability Check

Ein Fabasoft app.telemetry Agent kann die Verfügbarkeit eines Webservices durch das Senden einer HTTP-Anfrage und die Evaluierung des dabei erhaltenen HTTP-Status-Codes prüfen.

Webservice-Availability-Checks sind für die Authentisierungsmethoden „Anonymous“, „Basic“ und „Certificate“ (X.509 Zertifikate im PEM Format) unterstützt.

### 3.4.1.4 TCP-Ping Availability Check

Ein Fabasoft app.telemetry Agent kann die Verbindung zu einem TCP-Port auf einem entfernten System prüfen.

### 3.4.1.5 ICMP-Ping Availability Check

Ein Fabasoft app.telemetry Agent kann die Verfügbarkeit eines entfernten System anhand eines ICMP-Ping Checks prüfen.

## 3.4.2 Microsoft Windows Plattformen

Fabasoft app.telemetry Agents, die auf Microsoft Windows Plattformen (gemäß Kapitel 6.1 Fabasoft app.telemetry Agent) betrieben werden, unterstützen die folgenden, lokal (auf dem Server, auf dem der Fabasoft app.telemetry Agent betrieben wird) verfügbaren Verfügbarkeitschecks:

### 3.4.2.1 Microsoft Windows Service Availability Check

Ein Fabasoft app.telemetry Agent kann die Verfügbarkeit eines Microsoft Windows Services durch das Auslesen des Zustands des entsprechenden Services vom Microsoft Windows Service Control Manager prüfen.

### 3.4.2.2 Microsoft Windows Cluster-Resource Availability Check

Ein Fabasoft app.telemetry Agent kann die Verfügbarkeit einer Microsoft Windows Cluster Ressource durch das Auslesen des Zustands der entsprechenden Cluster Ressource vom Microsoft Windows Cluster Manager prüfen.

### 3.4.2.3 Webservice Availability Check

Ein Fabasoft app.telemetry Agent kann die Verfügbarkeit eines Webservices durch das Senden einer HTTP-Anfrage und die Evaluierung des dabei erhaltenen HTTP-Status-Codes prüfen.

Webservice-Availability-Checks sind für die Authentisierungsmethoden „Anonymous“, „Basic“, „Certificate“ (X.509 Zertifikate im PEM Format) und „Microsoft Windows Integrated“ unterstützt.

### 3.4.2.4 TCP-Ping Availability Check

Ein Fabasoft app.telemetry Agent kann die Verbindung zu einem TCP-Port auf einem entfernten System prüfen.

### 3.4.2.5 ICMP-Ping Availability Check

Ein Fabasoft app.telemetry Agent kann die Verfügbarkeit eines entfernten System anhand eines ICMP-Ping Checks prüfen.

### 3.4.3 Service Level Berechnung

Fabasoftware app.telemetry kann konfiguriert werden, um auf Basis des Ergebnisses eines Verfügbarkeitsstests den Prozentsatz der verfügbaren Zeit innerhalb einer konfigurierbaren Kernzeit und innerhalb einer konfigurierbaren Randzeit zu berechnen. Bei dieser Berechnung können konfigurierbare Zeiträume (beispielsweise für Wartungsfenster) ausgenommen werden.

## 3.5 Tests des Antwortzeitverhaltens

Fabasoftware app.telemetry kann auf Basis von Online Software-Telemetry Logs (gemäß Kapitel 5.1 Online Software-Telemetry Logs) für das periodische Berechnen des Antwortzeitverhaltens von Anfragen, die von integrierten Applikationen entgegengenommen werden und die Überwachung der so berechneten Werte konfiguriert werden. Abhängig von den berechneten Werten und von konfigurierbaren Grenzwerten kann ein Test des Antwortzeitverhaltens die Zustände OK, WARNING oder CRITICAL melden.

### 3.5.1 Durchschnittliches Antwortzeitverhalten

Fabasoftware app.telemetry unterstützt einen Test für die Berechnung des durchschnittlichen Antwortzeitverhaltens von Anfragen. Das durchschnittliche Antwortzeitverhalten wird für einen konfigurierbaren Zeitrahmen berechnet.

### 3.5.2 Prozentsatz der Anfragen im Grenzbereich

Fabasoftware app.telemetry unterstützt einen Test für die Berechnung des Prozentsatzes von Anfragen, die ein Antwortzeitverhalten kleiner eines konfigurierbaren Grenzwertes aufweisen. Der Prozentsatz der Anfragen wird für einen konfigurierbaren Zeitrahmen berechnet.

## 3.6 Benachrichtigungen

Fabasoftware app.telemetry erlaubt die Konfiguration von Benachrichtigungen. Benachrichtigungen können durch die Verwendung von Benachrichtigungskanälen und Benachrichtigungskontakten konfiguriert werden.

### 3.6.1 Benachrichtigungskanäle

Ein Benachrichtigungskanal, beschreibt die Methode, die für Benachrichtigungen verwendet werden soll. Der Fabasoftware app.telemetry Server unterstützt das Versenden von Benachrichtigungen über SMTP auf Microsoft Windows Plattformen oder über eine lokale verfügbare „sendmail“ Installation auf Linux Plattformen oder das Ausführen einer Kommandozeile am Fabasoftware app.telemetry Server als Benachrichtigungskanäle.

### 3.6.2 Benachrichtigungskontakt

Ein Benachrichtigungskontakt ist der Empfänger einer Benachrichtigung. Benachrichtigungskontakte können für Servicegruppen, Services, Tests von Leistungsindikatoren und Verfügbarkeitsstests definiert werden. Der Benachrichtigungskontakt wird informiert, wenn sich der Zustand der dafür definierten Servicegruppe oder Service, bzw. der Zustand aus der Evaluierung des Tests des Leistungsindikators oder des Verfügbarkeitsstests ändert.

Benachrichtigungskontakte können auch für Online Software-Telemetry Logs (gemäß Kapitel 5.1 Online Software-Telemetry Logs) definiert werden. Dieser Benachrichtigungskontakt wird informiert, wenn eine auf Basis der Fabasoftware app.telemetry Integration instrumentierte Applikation eine automatische Software-Telemetry Session (gemäß Kapitel 4.4 Automatische Software-Telemetry Session) erstellt.

## 4 Fabasoft app.telemetry Integration (SDK)

Fabasoft app.telemetry stellt ein SDK zur Verfügung, mit dem sich Applikationen in Fabasoft app.telemetry integrieren können. Applikationen können sich dabei selbst in Fabasoft app.telemetry registrieren und Informationen (Events), die im Rahmen von Software-Telemetry-Sessions verwendet werden, liefern. Das Fabasoft app.telemetry SDK unterstützt die folgenden Integrationsmöglichkeiten:

### 4.1 Registrierung

Applikationen können sich selbst in Fabasoft app.telemetry registrieren und somit zur Reduktion von manuellem Konfigurationsaufwand beitragen.

#### 4.1.1 Applikationsregistrierung

Eine Applikation kann sich durch die Bereitstellung von Applikationsname, einer eindeutigen Applikations-ID, Schichtname (Tier) und Schicht-ID (Tier-ID) in Fabasoft app.telemetry registrieren.

#### 4.1.2 Filterregistrierung

Fabasoft app.telemetry bietet die Funktionalität von Software-Telemetry-Sessions. Eine Software-Telemetry-Session kann durch die Auswahl einer registrierten Applikation, die Auswahl eines Detailgrads (Recording Level) und die Auswahl eines applikationsspezifischen Filterwertes gestartet werden. Das SDK erlaubt Applikationen, deren spezifische Filterwerte zu registrieren.

#### 4.1.3 Modulregistrierung

In einer Software-Telemetry-Session liefern registrierte Applikationen Informationen, die die Analyse des Pfades einer Anfrage durch das entsprechende System unterstützen. Software-Telemetry-Requests werden dabei grundsätzlich nach Applikationen und Server, sowie nach Prozessen oder Threads unterteilt dargestellt. Das SDK erlaubt Applikationen, diese Unterteilung durch die Verwendung von Modulen zu detaillieren. Ein Modul ist ein Teil einer Applikation, der eine präzisere Darstellung des Pfades einer Anfrage innerhalb dieser Applikation ermöglicht.

#### 4.1.4 Eventregistrierung

Innerhalb von Software-Telemetry-Requests liefern registrierte Applikationen Informationen durch die Verwendung von Events. Jeder Event besteht aus einer ID (eindeutig innerhalb eines Moduls) und einer Beschreibung. Um die Datenmenge der übertragenen Software-Telemetry-Daten zu reduzieren, ermöglicht es das SDK, dass Applikationen deren Events (ID und Beschreibung) registrieren, sodass anstatt der vollständigen Event-Beschreibungen nur noch die Event-IDs übertragen werden.

## 4.2 Events

Innerhalb von Software-Telemetry-Requests liefern registrierte Applikationen Informationen durch die Verwendung von Events. Durch die Übermittlung von Events können diese Applikationen die folgenden Informationen an Fabasoft app.telemetry übermitteln:

- Eine Anfrage betritt einen spezifischen Bereich der Applikation
- Eine Anfrage verlässt einen spezifischen Bereich der Applikation
- Zusätzliche Debug- oder Trace-Informationen

Applikationen übergeben für jeden Event einen Detailgrad (Standard, Detail oder Debug). Abhängig vom konfigurierten Detailgrad einer Software-Telemetry-Session werden nur die Events an den Fabasoft app.telemetry Server übermittelt, die für diesen Detailgrad zugelassen sind.

### 4.3 Kontext

In einer Software-Telemetry-Session kann der Beginn einer Anfrage, der Pfad einer Anfrage durch mehrere Threads, Prozesse oder Services von integrierten Applikationen sowie das Ende einer Anfrage dargestellt werden. Das SDK ermöglicht Applikationen, die dafür benötigten Informationen an Fabasoft app.telemetry zu übergeben. Applikationen können zu Beginn einer Anfrage einen „Kontext“ allokkieren und am Ende einer Anfrage diesen Kontext wieder freigeben. Bei der Übergabe einer Anfrage an einen anderen Thread oder Prozess kann auf der Seite des Aufrufers dieser Kontext gelesen werden und auf der Seite des Aufgerufenen kann dieser Kontext wieder referenziert werden. Für die Übergabe/Übertragung des Kontexts zwischen Threads und Prozessen ist die Applikation verantwortlich.

### 4.4 Automatische Software-Telemetry Session

Fabasoft app.telemetry bietet Applikationen die Möglichkeit, die von der Applikation abgesetzten Anfragen zu einer Software-Telemetry Session zusammenzufassen. Dafür werden die zuletzt von Fabasoft app.telemetry entgegengenommenen Anfragen berücksichtigt.

## 5 Software-Telemetry

Mit Hilfe der Fabasoft app.telemetry Integrationsmöglichkeiten wie in Kapitel 4 Fabasoft app.telemetry Integration (SDK) beschrieben, können Applikationen Log-, Trace- und/oder Debug-Informationen an Fabasoft app.telemetry übermitteln.

Fabasoft app.telemetry kann auf dieser Basis für die Analyse des Pfades von einzelnen Anfragen durch eine komplexe und heterogene Infrastruktur verwendet werden.

### 5.1 Online Software-Telemetry Logs

Fabasoft app.telemetry kann konfiguriert werden, sodass Anfragen, die von integrierten Applikationen entgegengenommen werden, dargestellt werden. Eine Anfrage wird als eine eigene Zeile gemeinsam mit allgemeinen Informationen wie Startzeit, Ausführungszeit oder Name des Fabasoft app.telemetry Agents, der die Anfrage von der Applikation entgegengenommen hat, dargestellt.

### 5.2 Software-Telemetry Sessions

#### 5.2.1 Explizite Software-Telemetry Session

Fabasoft app.telemetry liefert im „Start Software-Telemetry Session“ Dialog eine Liste von Applikationen, die für explizite Software-Telemetry-Sessions verfügbar sind. Eine explizite Software-Telemetry-Session kann durch die Auswahl der entsprechenden Applikation, die Auswahl eines von dieser Applikation bereitgestellten Filterwertes, die Auswahl eines unterstützten Detailgrades (Recording Level) und die Spezifikation einer Session-Beschreibung sowie der Dauer, die die Session laufen soll, gestartet werden.

Eine explizite Software-Telemetry-Session kann mit Hilfe des „Stop Software-Telemetry Session“ Dialogs beendet werden. Nachdem eine Software-Telemetry-Session beendet wurde, werden alle verfügbaren Daten automatisch von den Fabasoft app.telemetry Agents zum Fabasoft app.telemetry Server übertragen, verarbeitet und als Session persistiert. Die Software-Telemetry Session kann nun im Fabasoft app.telemetry Client geladen werden, wodurch die Session am Webserver aufbereitet und am Client graphisch dargestellt wird.

Eine aktive Software-Telemetry-Session hat Auswirkungen auf die analysierten Systeme. Diese Auswirkungen sind abhängig vom gewählten Detailgrad, der Selektivität der gewählten Filter und/oder den von den integrierten Applikationen übermittelten Daten.

#### 5.2.2 Analyse von Software-Telemetry Sessions

Die Analyse von Software-Telemetry Daten, die im Rahmen von Software-Telemetry Sessions empfangen werden, erfolgt durch die Auswahl der entsprechenden Anfrage aus den Online Software-Telemetry Logs.

### 5.3 Persistierung von Software-Telemetry Logs

Fabasoft app.telemetry kann konfiguriert werden, sodass Anfragen, die von integrierten Applikationen entgegengenommen werden, in einer relationalen Datenbank persistiert werden. Die relationale Datenbank ist nicht Bestandteil von Fabasoft app.telemetry. Eine Anfrage wird als eigene Zeile gemeinsam mit allgemeinen Informationen wie Startzeit, Ausführungszeit oder Name des Fabasoft app.telemetry Agents, der die Anfrage entgegengenommen hat, gespeichert. Die unterstützten Datenbanksysteme sind im Kapitel 6.3.4 Unterstützte Datenbanksysteme dokumentiert.

## 5.4 Auswertung von Software-Telemetry Logs

Fabasoftware app.telemetry kann konfiguriert werden, sodass Anfragen, die von integrierten Applikationen entgegengenommen werden, für Auswertungen aufbereitet werden. Im Rahmen dieser Konfiguration können applikationsspezifische Spalten der Software-Telemetry Logs definiert werden, über die Aggregationen wie Anzahl der Anfragen, Gesamtdauer der Anfragen und durchschnittliche Dauer der Anfragen berechnet werden können. Die berechneten Werte können gruppiert nach einer dieser definierten Spalten dargestellt werden. Fabasoftware app.telemetry ermöglicht auf dieser Basis die Auswahl eines Spaltenwerts als Filter und die erneute Gruppierung nach anderen entsprechenden Spalten (Drilldown).

Voraussetzung für diese Analysen sind die Persistierung von Software-Telemetry Logs und die Verfügbarkeit einer relationalen Datenbank für das Speichern der aggregierten Werte. Die relationale Datenbank ist nicht Bestandteil von Fabasoftware app.telemetry. Die unterstützten Datenbanksysteme sind im Kapitel 6.3.4 Unterstützte Datenbanksysteme dokumentiert.

## 5.5 Service Desk Integration

Fabasoftware app.telemetry kann konfiguriert werden, sodass ausgewählte Anfragen oder eine ausgewählte Software-Telemetry Session an den Fabasoftware Support Service Desk übermittelt werden. Für diese Konfiguration ist eine Freischaltung durch den Fabasoftware Support basierend auf einer Fabasoftware Support Vereinbarung Voraussetzung. Die Fabasoftware Support Vereinbarung ist nicht Bestandteil von Fabasoftware app.telemetry.

## 5.6 Syslog Integration Modul

Fabasoftware app.telemetry kann konfiguriert werden, sodass Linux Syslog Einträge als Software-Telemetry Requests in einem Software-Telemetry Logpool aufgezeichnet werden. Für diese Konfiguration muss sowohl ein Fabasoftware app.telemetry Agent als auch das Fabasoftware app.telemetry Syslog Forwarder Modul am gewünschten Zielsystem installiert sein. Die unterstützten Plattformen sind im Kapitel 6.5.3 Fabasoftware app.telemetry Integration für Syslog dokumentiert.

## 5.7 Telemetry Rohdaten Analyse

Seit der Version 2014 Fall Release können die Telemetrie-Rohdaten, die von instrumentierten Anwendungen gesammelt werden, nachträglich nach unterschiedlichen Gesichtspunkten analysiert werden. Nach der Auswahl eines Zeitbereichs für den Report sowie der gewünschten Auflösung kann die Analyse der Telemetrydaten gestartet werden, aber beachten Sie mögliche Auswirkungen auf die Systemlast bei Analysen über einen zu großen Zeitraum mit zu kleiner Auflösung (insb. wenn die Datenbasis der instrumentierten Anwendungen sehr groß ist).

## 6 Unterstützte Plattformen und Systemanforderungen

### 6.1 Fabasoft app.telemetry Agent

#### 6.1.1 Unterstützte Plattformen

Fabasoft app.telemetry Agents sind auf den folgenden Plattformen unterstützt:

Betriebssystem	Architektur
Microsoft Windows Server 2016	x64
Microsoft Windows Server 2019	x64
Microsoft Windows Server 2022	x64
Red Hat Enterprise Linux 8.1 – 8.4, Community ENTERprise Operating System 8.1 – 8.4	x64
Red Hat Enterprise Linux 9.4	x64

#### 6.1.2 Systemanforderungen

Fabasoft app.telemetry Agents benötigen die folgende Minimalkonfiguration zur exklusiven Verwendung:

- CPU: 1 GHz
- Hard Disk: 100 MB
- Memory: 250 MB

Software-Telemetry Caching oder intensive Check-Definitionen haben Auswirkungen auf diese Systemanforderungen und können ggf. entsprechende Anpassungen erfordern.

#### 6.1.3 Voraussetzungen

Für den Betrieb von Fabasoft app.telemetry Agents gelten folgende Voraussetzungen:

Betriebssystem	Voraussetzung
Red Hat Enterprise Linux 8.1 – 8.4, Community ENTERprise Operating System 8.1 – 8.4	RPM curl 7.61.1 RPM net-snmp 5.8 RPM net-snmp-libs 5.8 RPM libxml2 2.9.7 RPM openssl 1.1.1 RPM systemd 239 RPM apptelemetry-libs RPM checkpolicy 2.9 RPM policycoreutils-python 2.9 RPM selinux-policy-targeted 3.14.3
Red Hat Enterprise Linux 9.4	RPM curl 7.76.1 RPM net-snmp 5.9.1 RPM net-snmp-libs 5.9.1 RPM libxml2 2.9.12

	RPM openssl 3.0.1 RPM systemd 249 RPM apptelemetry-libs RPM checkpolicy 3.3 RPM policycoreutils-python 3.3 RPM selinux-policy-targeted 34.1.28
--	---

## 6.2 Fabasoft app.telemetry WebAPI

### 6.2.1 Unterstützte Plattformen

Das Fabasoft app.telemetry WebAPI ist auf den folgenden Plattformen unterstützt:

Betriebssystem	Architektur
Microsoft Windows Server 2016	x64
Microsoft Windows Server 2019	x64
Microsoft Windows Server 2022	x64
Red Hat Enterprise Linux 8.1 – 8.4, Community ENTerprise Operating System 8.1 – 8.4	x64
Red Hat Enterprise Linux 9.4	x64

### 6.2.2 Systemanforderungen

Das Fabasoft app.telemetry WebAPI benötigt die folgende Minimalkonfiguration zur exklusiven Verwendung:

- CPU: 1 GHz
- Hard Disk: 50 MB
- Memory: 100 MB

Software-Telemetry Caching hat Auswirkungen auf diese Systemanforderungen und kann ggf. entsprechende Anpassungen erfordern.

### 6.2.3 Voraussetzungen

Für den Betrieb des Fabasoft app.telemetry WebAPIs gelten folgende Voraussetzungen

Betriebssystem	Voraussetzung
Microsoft Windows Server 2016	Microsoft Internet Information Services 10.0
Microsoft Windows Server 2019	Microsoft Internet Information Services 10.0
Microsoft Windows Server 2022	Microsoft Internet Information Services 10.0
Red Hat Enterprise Linux 8.1 – 8.4, Community ENTerprise Operating System 8.1 – 8.4	RPM curl 7.61.1 RPM httpd 2.4.37 RPM apr 1.6.1 RPM apr-util 1.6.1 RPM openssl 1.1.1 RPM apptelemetry-libs RPM checkpolicy 2.9 RPM policycoreutils-python 2.9 RPM selinux-policy-targeted 3.14.3
Red Hat Enterprise Linux 9.4	RPM curl 7.76.1 RPM httpd 2.4.51 RPM apr 1.7.0 RPM apr-util 1.6.1

	RPM openssl 3.0.1 RPM apptelemetry-libs RPM checkpolicy 3.3 RPM policycoreutils-python 3.3 RPM selinux-policy-targeted 34.1.28
--	--

## 6.3 Fabasoft app.telemetry Server

### 6.3.1 Unterstützte Plattformen

Fabasoft app.telemetry Server sind auf den folgenden Plattformen unterstützt:

Betriebssystem	Architektur
Microsoft Windows Server 2022	x64
Red Hat Enterprise Linux 8.4, Community ENTERprise Operating System 8.4	x64
Red Hat Enterprise Linux 9.4	x64

### 6.3.2 Systemanforderungen

Der Fabasoft app.telemetry Server benötigt die folgende Minimalkonfiguration zur exklusiven Verwendung:

- CPU: 1 GHz
- Hard Disk: 2 GB
- Memory: 1 GB

Software-Telemetry Caching, implizite und/oder explizite Software-Telemetry-Sessions oder intensive Check-Definitionen haben Auswirkungen auf diese Systemanforderungen und können ggf. entsprechende Anpassungen erfordern.

### 6.3.3 Voraussetzungen

Für den Betrieb des Fabasoft app.telemetry Servers gelten folgende Voraussetzungen:

Betriebssystem	Voraussetzung
Microsoft Windows Server 2022	Microsoft Internet Information Services 10 inklusive folgender Security Role Services: Basic Authentication, Windows Authentication, IP and Domain Restriction, Request Filtering and URL Authorization
Red Hat Enterprise Linux 8.4, Community ENTERprise Operating System 8.4	RPM httpd 2.4.37 RPM apr 1.6.1 RPM apr-util 1.6.1 RPM net-snmp 5.8 RPM net-snmp-libs 5.8 RPM libpq 12.1 RPM libxml2 2.9.7 RPM minizip 2.8.9 RPM xmlsec1 1.2.25 RPM xmlsec1-openssl 1.2.25 RPM openssl 1.1.1 RPM systemd 239 RPM apptelemetryagent RPM apptelemetryweb RPM checkpolicy 2.9 RPM policycoreutils-python 2.9

	RPM selinux-policy-targeted 3.14.3
Red Hat Enterprise Linux 9.4	RPM httpd 2.4.51 RPM apr 1.7.0 RPM apr-util 1.6.1 RPM net-snmp 5.9.1 RPM net-snmp-libs 5.9.1 RPM libpq 13.2 RPM libxml2 2.9.12 RPM xmlsec1 1.2.29 RPM xmlsec1-openssl 1.2.29 RPM openssl 3.0.1 RPM systemd 249 RPM apptelemetryagent RPM apptelemetryweb RPM checkpolicy 3.3 RPM policycoreutils-python 3.3 RPM selinux-policy-targeted 34.1.28

### 6.3.4 Unterstützte Datenbanksysteme

Der Fabasoft app.telemetry Server unterstützt die folgenden relationalen Datenbanksysteme für die Speicherung von Software-Telemetry Request Informationen:

Fabasoft app.telemetry Server Betriebssystem	Datenbanksystem
Microsoft Windows Server 2022	Microsoft SQL Server 2022 PostgreSQL 16.5
Red Hat Enterprise Linux 8.4, Community ENTERprise Operating System 8.4	PostgreSQL 12.1
Red Hat Enterprise Linux 9.4	PostgreSQL 16.5

## 6.4 Fabasoft app.telemetry Client

### 6.4.1 Unterstützte Plattformen

Fabasoft app.telemetry Clients sind auf den folgenden Plattformen und Web Browsern unterstützt:

Betriebssystem	Architektur	Web Browser
Microsoft Windows 10	x64	Mozilla Firefox 132 Google Chrome 131 Microsoft Edge 131

### 6.4.2 Systemanforderungen

Fabasoft app.telemetry Clients benötigen die folgende Minimalkonfiguration zur exklusiven **Verwendung**:

- CPU: 1 GHz
- Hard Disk: 100 MB
- Memory: 512 MB

Software-Telemetry Auswertungen, intensive Check-Definitionen, Dashboards oder große Infrastruktur-Definitionen haben Auswirkungen auf diese Systemanforderungen und können ggf. entsprechende Anpassungen erfordern.

## 6.5 Fabasoft app.telemetry Integration

Die Fabasoft app.telemetry Integration (SDK) ist nur auf den Fabasoft app.telemetry Agent Plattformen unterstützt. Eine Liste der unterstützten Agent Plattformen ist in Kapitel 6.1.1 Unterstützte Plattformen zu finden.

### 6.5.1 Fabasoft app.telemetry Integration für C++, C#, Java

Die Fabasoft app.telemetry Integration für C++, C# und Java ist auf den folgenden Plattformen und Schnittstellen unterstützt:

Betriebssystem	Umgebung
Microsoft Windows Server 2016	Microsoft Visual Studio 2022 (C++) Microsoft Visual Studio 2022 (C#) Java Development Kit (JDK) 17
Microsoft Windows Server 2019	Microsoft Visual Studio 2022 (C++) Microsoft Visual Studio 2022 (C#) Java Development Kit (JDK) 17
Microsoft Windows Server 2022	Microsoft Visual Studio 2022 (C++) Microsoft Visual Studio 2022 (C#) Java Development Kit (JDK) 17
Red Hat Enterprise Linux 8.1 – 8.4, Community ENTERprise Operating System 8.1 - 8.4	gcc 8.3.1 (C++) OpenJDK 17

Red Hat Enterprise Linux 9.4	gcc 11.3.1 (C++) OpenJDK 17
------------------------------	--------------------------------

### 6.5.2 Fabasoft app.telemetry Integration für JavaScript

Die Fabasoft app.telemetry Integration für JavaScript unterstützt die JavaScript Schnittstelle folgender Web Browser mit den aufgelisteten Funktionen:

Unterstützter Web Browser	JavaScript SDK	Feedback Dialoge	HTML5 Screenshot	Navigation Timing
Mozilla Firefox 132	Ja	Ja	Ja	Ja
Google Chrome 131	Ja	Ja	Ja	Ja
Microsoft Edge 131	Ja	Ja	Ja	Ja

### 6.5.3 Fabasoft app.telemetry Integration für Syslog

Die Fabasoft app.telemetry Integration für Syslog ist nur auf Linux Systemen für "rsyslog" auf den folgenden Plattformen unterstützt:

Betriebssystem	Voraussetzung
Red Hat Enterprise Linux 8.1 – 8.4, Community ENTERprise Operating System 8.1 - 8.4	RPM rsyslog 8.1911.0 RPM apptelemetryagent RPM checkpolicy 2.9 RPM policycoreutils-python 2.9 RPM selinux-policy-targeted 3.14.3
Red Hat Enterprise Linux 9.4	RPM rsyslog 8.37.0 RPM apptelemetryagent RPM checkpolicy 3.3 RPM policycoreutils-python 3.3 RPM selinux-policy-targeted 34.1.28

### 6.5.4 Fabasoft app.telemetry Support Matrix

Die folgende Support Matrix zeigt die wichtigsten Funktionen der Fabasoft app.telemetry Integration (Software-Telemetry SDK) mit der ersten Produktversion, ab der die Funktion freigegeben und unterstützt wurde:

Fabasoft app.telemetry SDK Funktionen	Freigegeben/unterstützt mit Produktversion
Base SDK	Fabasoft app.telemetry 2008
Software-Telemetry Sessions	Fabasoft app.telemetry 2008
Online Software-Telemetry Logs, Reported Sessions	Fabasoft app.telemetry 2009
JavaScript SDK / End-2-End Software-Telemetry	Fabasoft app.telemetry 2009 Summer Release
Feedback Button	Fabasoft app.telemetry 2010 Spring Release

Feedback Button mit Screenshot	Fabasoft app.telemetry 2010 Summer Release
SDK für Android (Java) und Apple iOS	Fabasoft app.telemetry 2013 Winter Release
Feedback Dialog Screenshot Vorschau mit Editier-Funktionen	Fabasoft app.telemetry 2013 Summer Release
Entworfene on-premise Forms alternativ einsetzbar zu SDK Report Dialog – Ressourcen über WebAPI	Fabasoft app.telemetry 2013 Fall Release
Mehrsprachige Feedback Forms	Fabasoft app.telemetry 2014 Winter Release
Speedtest jQuery control JavaScript API lädt CSS-Bilder über WebAPI	Fabasoft app.telemetry 2014 Spring Release
Konfigurierbare Event Displaynames-Abbildung über module_registration.xml (in Konfig-Package)	Fabasoft app.telemetry 2014 Summer Release
Verbessertes SDK für Android (Java) und Apple iOS	Fabasoft app.telemetry 2015
Application Properties und Values	Fabasoft app.telemetry 2018 UR3
Software-Telemetry Counters	Fabasoft app.telemetry 2019

Diese Version (aktuelle) des Fabasoft app.telemetry SDK wird nur auf den definierten Fabasoft app.telemetry Agent Plattformen (siehe Kapitel 6.1.1 Unterstützte Plattformen) unterstützt.

## 6.6 Fabasoft app.telemetry SNMP Agent

### 6.6.1 Unterstützte Plattformen

Der Fabasoft app.telemetry SNMP Agent ist nur auf Linux Systemen auf den folgenden Plattformen unterstützt:

Betriebssystem	Architektur
Red Hat Enterprise Linux 8.4, Community ENTERprise Operating System 8.4	x64
Red Hat Enterprise Linux 9.4	x64

### 6.6.2 Systemanforderungen

Fabasoft app.telemetry SNMP Agents benötigen die folgende Minimalkonfiguration zur exklusiven Verwendung:

- CPU: 1 GHz
- Hard Disk: 100 MB
- Memory: 100 MB

Größere Fabasoft app.telemetry Installationen haben Auswirkungen auf diese Systemanforderungen und können ggf. entsprechende Anpassungen erfordern.

### 6.6.3 Voraussetzungen

Für den Betrieb von Fabasoft app.telemetry SNMP Agents gelten folgende Voraussetzungen:

Betriebssystem	Voraussetzung
Red Hat Enterprise Linux 8.4, Community ENTERprise Operating System 8.4	RPM net-snmp-libs 5.8 RPM net-snmp-agent-libs 5.8 RPM openssl 1.1.1 RPM systemd 239 RPM checkpolicy 2.9 RPM policycoreutils-python 2.9 RPM selinux-policy-targeted 3.14.3
Red Hat Enterprise Linux 9.4	RPM net-snmp-libs 5.9.1 RPM net-snmp-agent-libs 5.9.1 RPM openssl 3.0.1 RPM systemd 249 RPM checkpolicy 3.3 RPM policycoreutils-python 3.3 RPM selinux-policy-targeted 34.1.28

## 6.7 Fabasoft app.telemetry Proxy Server

### 6.7.1 Unterstützte Plattformen

Fabasoft app.telemetry Proxy Server sind auf den folgenden Plattformen unterstützt:

Betriebssystem	Architektur
Microsoft Windows Server 2022	x64
Red Hat Enterprise Linux 8.4, Community ENTERprise Operating System 8.4	x64
Red Hat Enterprise Linux 9.4	x64

### 6.7.2 Systemanforderungen

Der Fabasoft app.telemetry Proxy Server benötigt die folgende Minimalkonfiguration zur exklusiven Verwendung:

- CPU: 1 GHz
- Hard Disk: 100 MB
- Memory: 1 GB

Größere Fabasoft app.telemetry Installationen haben Auswirkungen auf diese Systemanforderungen und können ggf. entsprechende Anpassungen erfordern.

### 6.7.3 Voraussetzungen

Für den Betrieb des Fabasoft app.telemetry Proxy Servers gelten folgende Voraussetzungen:

Betriebssystem	Voraussetzung
Microsoft Windows Server 2022	Fabasoft app.telemetry Server $\geq$ 19.0
Red Hat Enterprise Linux 8.4, Community ENTERprise Operating System 8.4	RPM apptelemetryagent $\geq$ 20.1 RPM apptelemetryserver $\geq$ 20.1 RPM openssl 1.1.1 RPM systemd 239 RPM checkpolicy 2.9 RPM policycoreutils-python 2.9 RPM selinux-policy-targeted 3.14.3
Red Hat Enterprise Linux 9.4	RPM apptelemetryagent $\geq$ 24.0 RPM apptelemetryserver $\geq$ 24.0 RPM openssl 3.0.1 RPM systemd 249 RPM checkpolicy 3.3 RPM policycoreutils-python 3.3 RPM selinux-policy-targeted 34.1.28

## 6.8 Allgemeines

Die Installation von Fabasoft app.telemetry Services in einer heterogenen Betriebssystemumgebung ist unterstützt, solange jedes einzelne Fabasoft app.telemetry Service auf einer unterstützten Plattform installiert wird.

Die Installation von Fabasoft app.telemetry Services unterschiedlicher Versionen auf einem Computer ist nicht unterstützt.

Der Installation von Fabasoft app.telemetry Services unterschiedlicher Versionen in einer Fabasoft app.telemetry Installation ist nicht unterstützt.

Fabasoft app.telemetry Agents können nur von einem Fabasoft app.telemetry Server angesprochen werden.

Die angegebenen Werte der Systemanforderungen (Prozessor, verfügbaren Arbeitsspeicher und freien Festplattenspeicher) sind Minimalwerte, die einen Demobetrieb und eine kleine Installation erlauben.

Die unterstützten Plattformen, notwendigen Hard- und Softwareanforderungen für eine zukünftige Version dieses Produkts können von den Anforderungen der aktuellen Version abweichen.

Eine bestimmte Softwareproduktversion eines Drittherstellers, die in einem Fabasoft app.telemetry Softwareprodukt integriert ist, wird von Fabasoft R&D GmbH maximal solange produktseitig unterstützt wie diese Softwareproduktversion vom Dritthersteller selbst unterstützt wird.

Beispielsweise wird eine bestimmte Version eines integrierten Microsoft Softwareprodukts nur maximal bis zu dem Datum von Fabasoft R&D GmbH unterstützt, an dem die Microsoft „Extended Support Phase“ für diese Softwareproduktversion abläuft.

Softwareprodukte von Drittherstellern, die von Fabasoft app.telemetry Produkten vorausgesetzt oder unterstützt werden, deren ordnungsgemäße Lizenzierung und Installation, die notwendigen Tests vor der Produktionsfreigabe sowie der entsprechende Herstellersupport sind nicht im Liefer- und Leistungsumfang enthalten und unterliegen daher insbesondere hinsichtlich ihrer Funktionalität, Funktionsweise oder Eigenschaften nicht der Gewährleistung durch Fabasoft R&D GmbH. Fabasoft R&D GmbH haftet nicht für Fehler oder Funktionsstörungen, welche durch Softwareprodukte von Drittherstellern und/oder solchen, die nicht im Liefer- und Leistungsumfang enthalten sind, verursacht werden.

## 7 Einsatzbedingungen

Die in der Softwareproduktinformation enthaltenen Aussagen über den bestimmungsgemäßen Gebrauch und die Einsatzbedingungen des Programms verstehen sich nur als Grundsatzinformation. Um bei der täglichen Arbeit mit dem Programm umfassend den bestimmungsgemäßen Gebrauch und die Einsatzbedingungen sicherzustellen und zu gewährleisten, empfiehlt sich jedenfalls eine qualifizierte Einarbeitung durch geschultes Fachpersonal.

Performance und Antwortzeitverhalten der Fabasoft app.telemetry Softwareprodukte hängen wesentlich von der zum Einsatz kommenden Infrastruktur ab. Für Auslegung, Konfiguration und Dimensionierung der entsprechenden Infrastruktur verweisen wir auf die Angaben, Empfehlungen und Spezifikationen der jeweiligen Hersteller.

Das Programm ist für eine übliche kommerzielle Verwendung bestimmt. Einsatzbereiche, an die besondere Anforderungen gestellt werden (wie z.B. Steuerung von Verkehrsmitteln, Maschinen und Anlagen, Echtzeitanwendungen), sind hiervon nicht mit umfasst.

## 8 Abgrenzung zum Leistungsumfang

Insbesondere die folgenden Punkte dienen zur Abgrenzung des Leistungsumfangs von Fabasoft app.telemetry.

- Fabasoft app.telemetry enthält im Standardprodukt keine rechenzentrumsspezifischen Anpassungen.
- Fabasoft app.telemetry unterstützt bestimmte Benutzer-Log-in-Verfahren durch Integration von Drittprodukten und Betriebssystem-Funktionen gemäß Softwareproduktinformation von Fabasoft app.telemetry. Die technische Umsetzung eines der unterstützten Log-in-Verfahren durch die Auswahl des geeigneten Log-in-Verfahrens, die Lizenzierung und Installation von Drittprodukten und die Integration in die Netzwerkinfrastruktur ist nicht Bestandteil der Produktfunktionalität.
- Fabasoft app.telemetry unterstützt bestimmte E-Mail -Systeme durch die Integration von Drittprodukten und Betriebssystem-Funktionen gemäß Softwareproduktinformation von Fabasoft app.telemetry. Die Lizenzierung, Installation und Wartung dieser Systeme ist nicht Bestandteil der Produktfunktionalität.
- Das systemtechnische Realisierungskonzept für eine skalierbare Lösung, sowie die dafür notwendige Dimensionierung der Server und die damit verbundenen systemtechnischen und organisatorischen Begleitmaßnahmen sind nicht Bestandteil der Produktfunktionalität.
- Für den Aufbau einer Produktionsumgebung sind umfassende Infrastruktur-Tests erforderlich, insbesondere in den Bereichen Skalierbarkeit, Stabilität, Ausfallsicherheit, Verfügbarkeit, Performance, Backup and Recovery, Disaster Recovery und Sicherheit.
- Die Definition und Umsetzung von Service Level Agreements (SLAs) ist nicht Bestandteil der Produktfunktionalität.