



IoT-Architektur – sind traditionelle Architekturen gut genug oder braucht es neue Ansätze?

Guido Schmutz, Trivadis AG

„Internet of Things“ (IoT) ist eines der zentralen Themen der Digitalisierung. Es handelt sich dabei um die Vision, dass sämtliche Dinge auf der Welt das Potenzial besitzen, sich über das Internet zu verbinden und ihre Daten zur Verfügung zu stellen. Damit lassen sich neue, innovative Produkte und Services bauen, mit denen man sich von der Konkurrenz differenzieren kann.

Das Internet der Dinge wird unser Leben grundlegend verändern. Es wird Auswirkungen haben auf die Art und Weise, wie wir wohnen, arbeiten, einkaufen, zusammenleben und reisen. Unternehmen, die heute erste IoT-Anwendungsfälle umsetzen, können noch als „Early Adopter“ bezeichnet werden. Dies wird sich aber rasch ändern und man wird früher oder später mit der Aufgabe konfrontiert werden, eine Integration von IoT in eine bereits existierende Anwendungslandschaft zu schaffen.

Dieser Artikel zeigt, ob und wie sich eine IoT-Anwendung in eine bestehende Architektur integrieren lässt und wo die Grenzen einer solchen Lösung lie-

gen. Es wird darauf eingegangen, wie sich eine bestehende Systemumgebung durch neue, spezialisierte Komponenten so erweitern lässt, dass das äußerst hohe Nachrichtenvolumen, das durch die IoT-Devices verursacht wird, ohne Probleme verarbeitet werden kann. Bevor wir mit der Diskussion starten, hier die drei zentralen Fragen, auf die es bei einer IoT-Anwendung eine Antwort zu finden gilt:

- *Verbinden*

Wie werden die Daten von IoT Devices abgegriffen und wie kann die Komplexität der Verbindung zu den unterschiedlichen Geräten möglichst gering gehalten werden?

- *Analyse*

Wie können die IoT-Daten rasch und effizient analysiert werden, um die interessantesten von den nicht interessanten Ereignissen zu trennen?

- *Integrieren*

Wie werden die IoT-Daten beziehungsweise die wichtigen Ereignisse in die (bestehende) Unternehmens-Infrastruktur integriert, sodass sowohl bestehende als auch neue Prozesse genutzt werden können?

Wir nehmen traditionelle Service- beziehungsweise Nachrichten-orientierte Architektur als Ausgangspunkt und starten

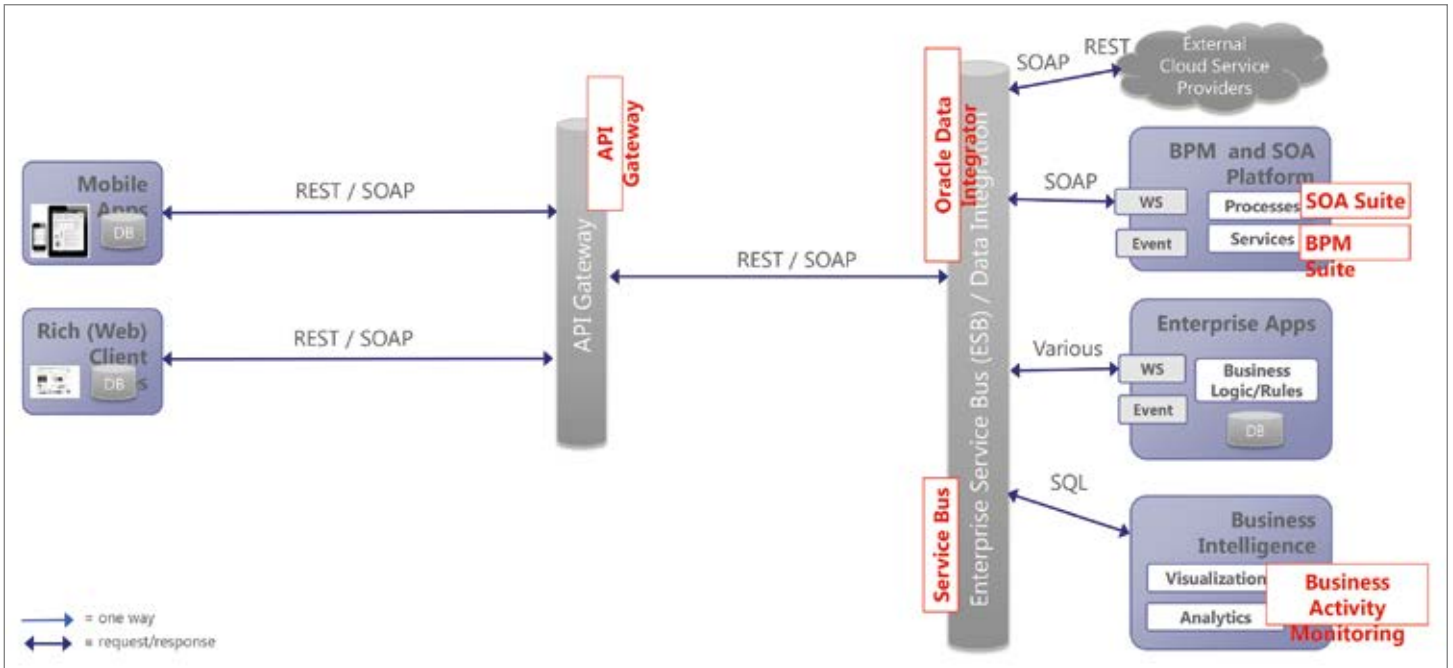


Abbildung 1: Traditionelle, SOA-basierte Architektur mit Oracle Fusion Middleware (in Rot)

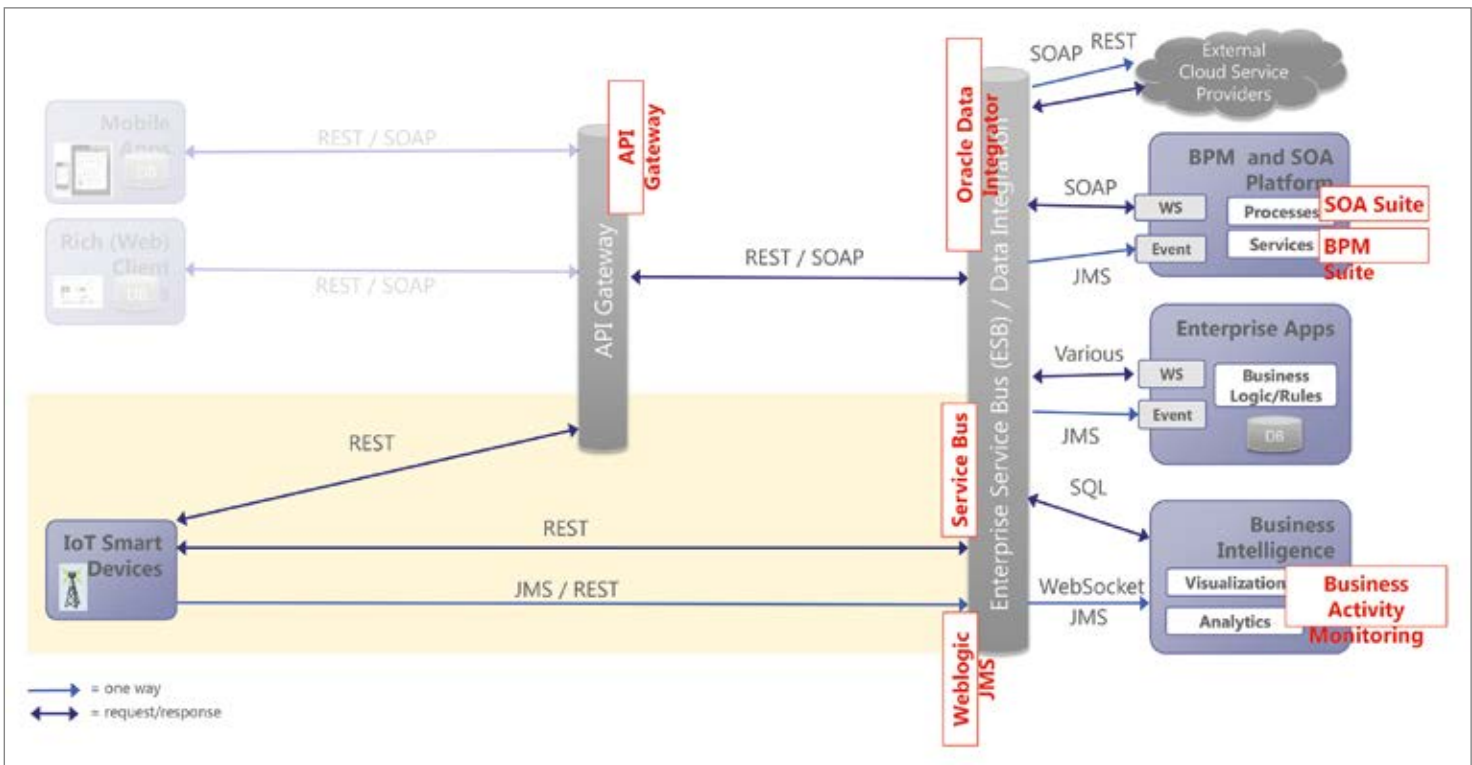


Abbildung 2: Erweiterung der traditionellen Architektur mit IoT-Smart-Devices

mit einer heute gängigen Architektur, wie sie in *Abbildung 1* vereinfacht dargestellt ist. Die Applikationen auf der linken Seite nutzen Dienste, die von den Systemen auf der rechten Seite über Services angeboten werden. In der Mitte dient ein Enterprise Service Bus (ESB) zur Entkoppelung der Services von den Nutzern (Ser-

vice-Virtualisierung) und das API-Gateway zur Sicherheit der Services. Es werden sowohl SOAP- wie auch REST-basierte Services unterstützt. Um es ein bisschen konkreter zu machen, sind mögliche Produkte aus dem Oracle Stack in Rot dargestellt. Es handelt sich dabei um die Produkte aus der Oracle Fusion Middleware.

Dieses Bild dient als Ausgangslage, um die Frage zu diskutieren, ob für eine IoT-Anwendung traditionelle Architekturen gut genug oder ob neue Architektur-Ansätze notwendig sind.

Mit der Umsetzung einer ersten IoT-Anwendung wird man sich zwangsläufig die Frage stellen müssen, ob eine beste-

hende Infrastruktur beziehungsweise Architektur dafür genutzt werden kann. Eine IoT-Anwendung unterscheidet sich von herkömmlichen Client-Applikationen primär dadurch, dass Daten ausschließlich automatisch, ohne menschliche Interaktionen generiert werden (vom Sensor), aber weniger durch die Business-Logik, die es situationsbedingt auszuführen gilt. Viele der bestehenden Backend-Applikationen wollen also auch bei IoT-Anwendungen weitergenutzt werden. Können wir die Geräte nicht einfach als neue Quelle sehen und analog zu den Web- oder Mobile-Applikationen in unsere traditionelle Architektur integrieren, wie dies in *Abbildung 2* dargestellt ist?

In diesem Fall kommunizieren die IoT-Devices über die Services, die vom API-Gateway oder vom ESB exponiert werden und in den Backend-Systemen implementiert sind. Für die „One-way“-Kommunikation, also vom Device zum Backend, könnte auch der Einsatz einer Nachrichten-orientierten Middleware ins Auge gefasst werden, mit dem Vorteil der besseren Entkopplung und eines zuverlässigeren Nachrichten-Transfers. Im Falle von Oracle wäre dies ein WebLogic JMS Server, der das Java Message API (JMS) unterstützt. Funktional gesehen ist dies

kaum ein Problem, man wird allerdings mit folgenden Problemen beziehungsweise Herausforderungen konfrontiert:

- Besitzen die IoT-Devices genügend Ressourcen, um direkt über ein Internet-Protokoll (HTTP) beziehungsweise über JMS zu kommunizieren? Dagegen kann sowohl der erhöhte Stromverbrauch als auch der Formfaktor eines IoT-Device sprechen. Man spricht hier auch von Smart Devices, falls diese genügende Intelligenz haben, um direkt mit dem Backend zu sprechen. Andere Devices sollen möglichst lange mit einer Batterieladung auskommen, weil sie einen Schlafmodus besitzen und diesen nur kurzzeitig für das Senden eines neuen Messpunkts verlassen.
- Soll das Device nur Daten aufnehmen und weiterleiten (Sensor) oder ist auch der Weg zurück notwendig, um den Kontext des Device zu verändern (Aktuator)? Wie kann in diesem Fall das Device erreicht werden?
- Lässt sich JMS von den externen IoT-Devices direkt nutzen? Sind die Firewalls für das JMS-Protokoll offen?

- Wie viele IoT-Devices sind insgesamt angedacht und in welcher Periodizität senden sie Nachrichten an das Backend? Kann das daraus resultierende Volumen durch den Service Bus oder JMS Server verkraftet werden? Hier gilt es zu beachten, dass 300 Nachrichten pro Sekunde für eine JMS Queue bereits die obere Grenze darstellen können. Falls also JMS genutzt wird, dann wird der JMS Server den ersten möglichen Engpass darstellen.
- Welche Operationen sind auf den IoT-Nachrichten (Ereignisse) notwendig, um entsprechende Aktionen abzuleiten? Sind es ausschließlich Filter-, Transformation- und Routing-Operationen, dann kann dies funktional auch von einem ESB problemlos bewältigt werden.

Es ist also sehr wohl möglich, erste IoT-Anwendungen über eine bestehende Service- beziehungsweise Nachrichten-orientierte Infrastruktur anzugehen; man sollte aber jederzeit die Grenzen der Plattform im Hinterkopf haben. Was geschieht, wenn eine IoT-Anwendung, die klein startet, zum Erfolg wird und

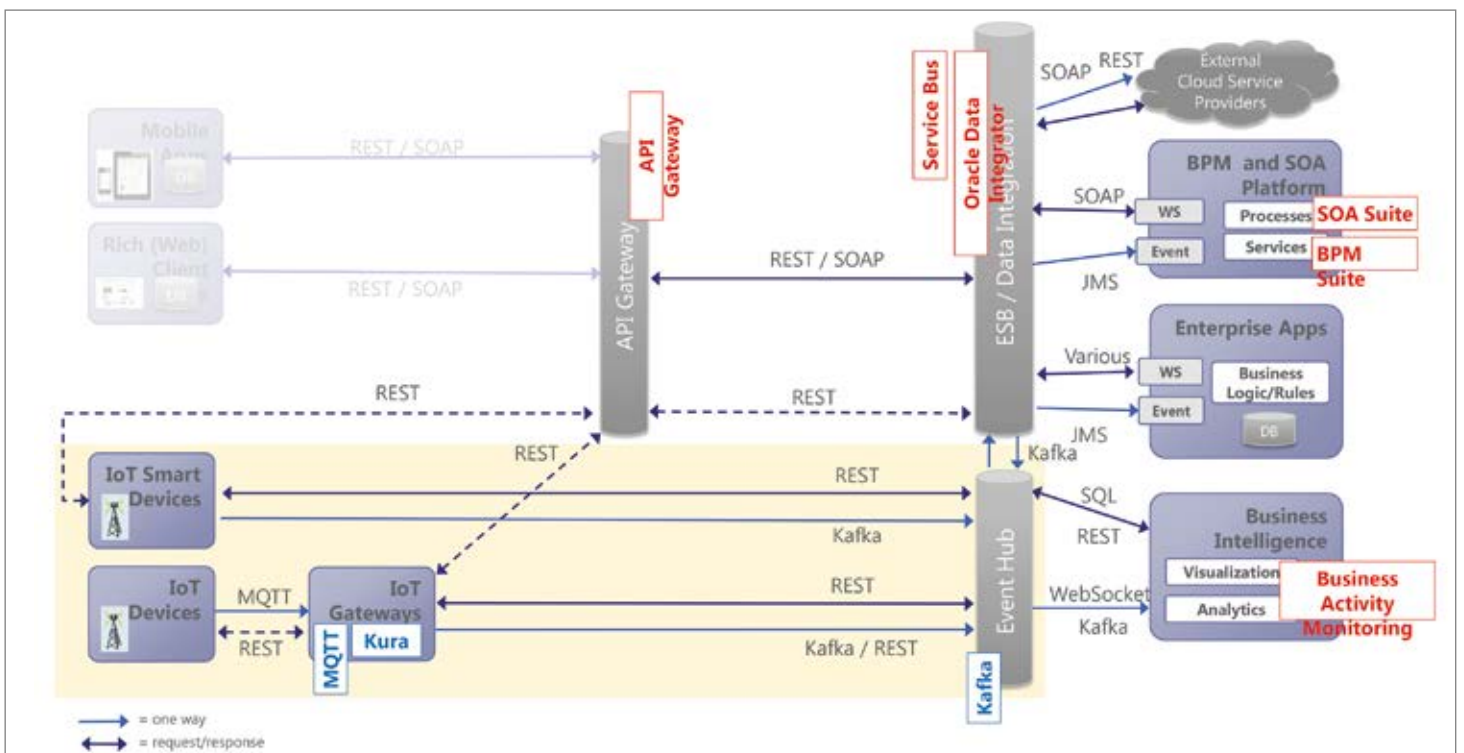


Abbildung 3: Erweiterung der traditionellen Architektur mit IoT-Gateways und modernen Message-Broker

sich die Nachrichtenmenge exponentiell vergrößert? Wie wirkt es sich aus, wenn die einzelnen IoT-Devices mit geringerer Frequenz messen und anstatt im 15-Sekunden-Takt neu im 1-Sekunden-Takt neue Daten übermitteln?

Eine bestehende Service- oder Nachrichten-orientierte Infrastruktur ist meis-

tens von zentraler Wichtigkeit für ein Unternehmen und die IoT-Anwendung darf die bisher garantierten Service-Level auf keinen Fall gefährden. Das Skalieren der bestehenden Infrastruktur kann rasch teuer werden, wenn es darum gehen soll, um Faktoren größere Nachrichtenmengen verkraften zu können.

Event Hubs und optionale IoT-Gateways einführen

Ein Event Hub ist darauf ausgelegt, riesige Datenströme effizient und zuverlässig entgegenzunehmen und weiterzuleiten. Er stellt dabei eine noch relativ neue Alternative zu einem klassischen JMS Server dar,

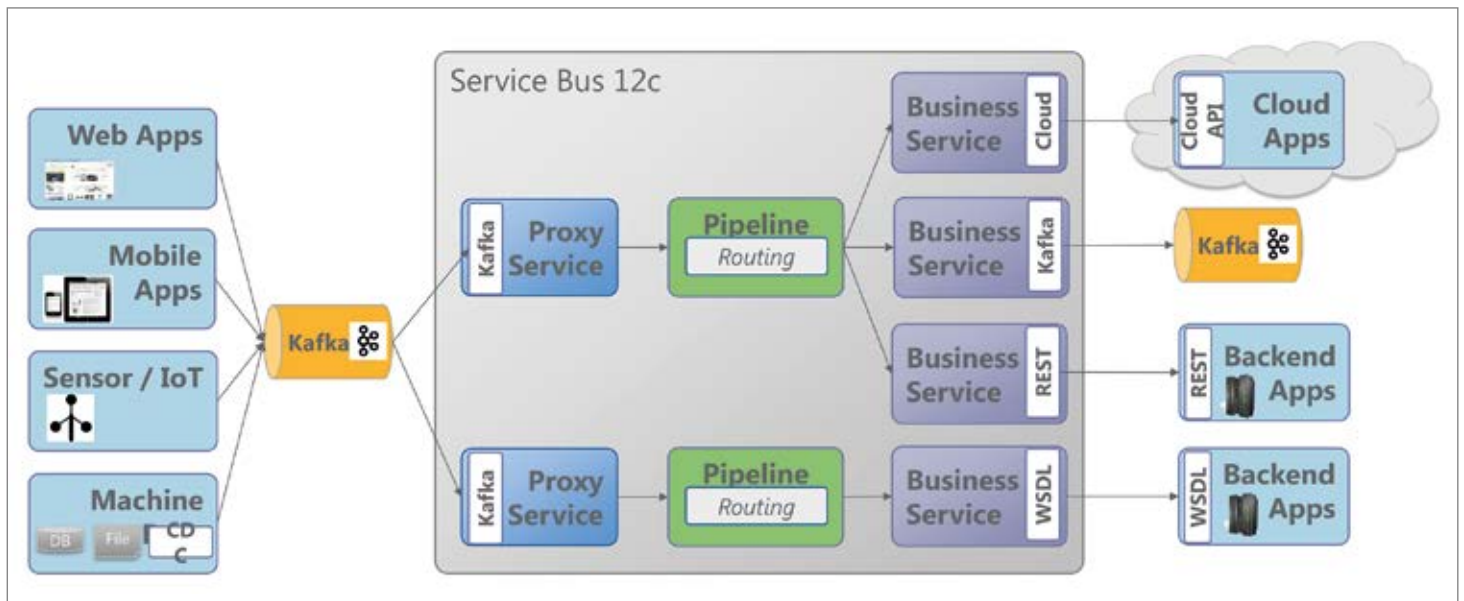


Abbildung 4: Integration von Apache Kafka mit dem Service Bus von Oracle

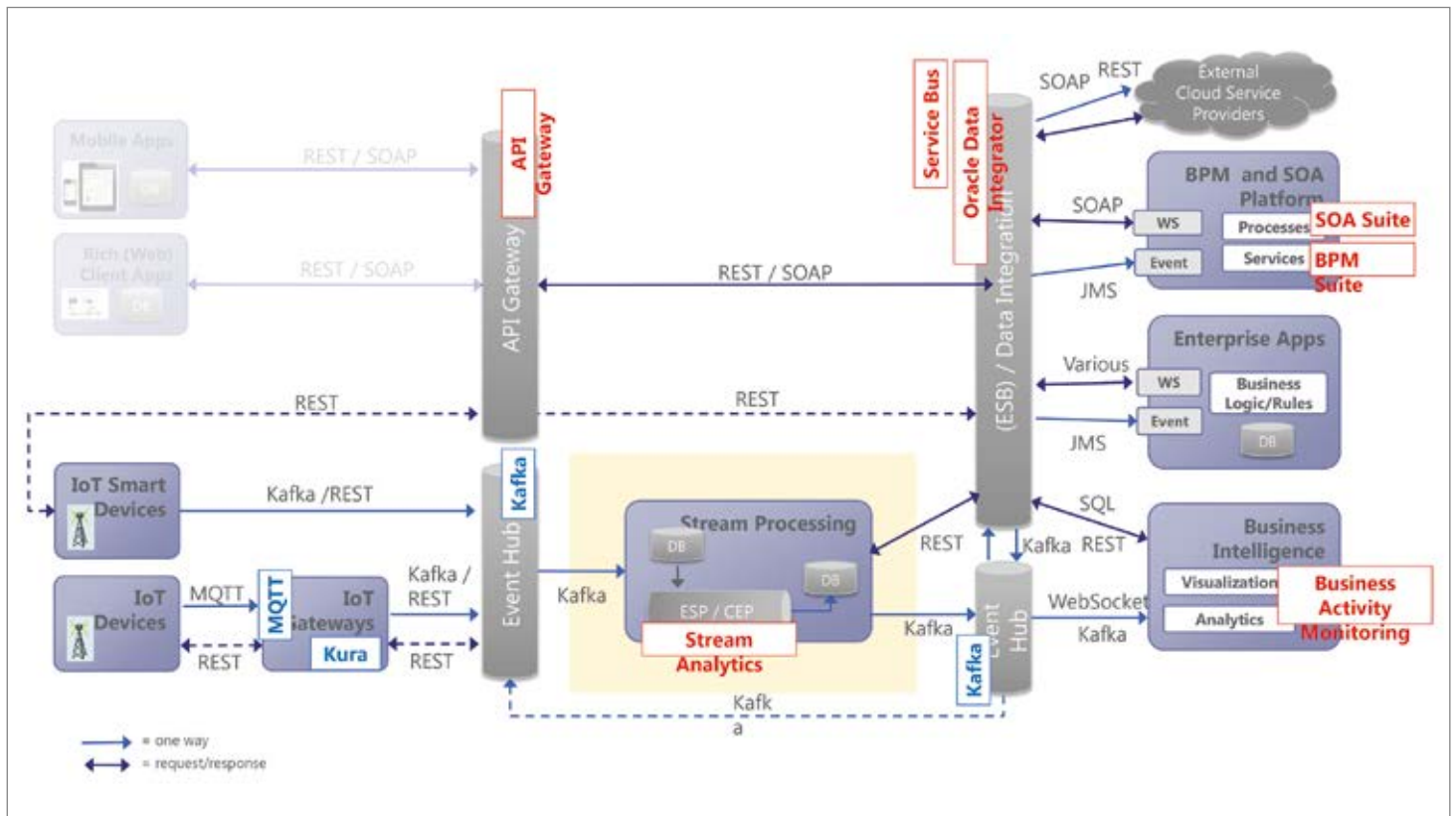


Abbildung 5: Erweiterung um Stream-Processing-Funktionalitäten

da er mit gleichen Ressourcen einen um Faktoren größeren Nachrichten-Durchsatz gewährleistet. Um auch nicht intelligente Devices unterstützen zu können, also Devices, die mit geringen Ressourcen auskommen müssen, können IoT-Gateways in die Kommunikation eingebunden werden. Ein Gateway stellt dabei die Konnektivität mit dem Backend sicher, wird möglichst nahe zu einer Gruppe von Devices platziert, die sich dann über spezialisierte, leichtgewichtige Protokolle mit dem Gateway verbinden. Dadurch sinken die Kosten und die Komplexität der Devices wird drastisch reduziert. *Abbildung 3* zeigt die Erweiterung der Architektur mit IoT-Gateways und dem Event Hub.

Bei dieser Variante kommuniziert das IoT-Device mit dem Gateway über MQTT, einem Vertreter der neuartigen IoT-Protokolle. Alternativen dazu wären ZigBee, BLE oder CoAP. Für die Gateway-Logik kann Eclipse Kura zum Einsatz kommen, ein Java-basiertes Framework für die Implementierung von IoT-Gateways. Als Event Hub agiert Apache Kafka, das ursprünglich von LinkedIn entwickelt und später der Apache Foundation zur Verfügung gestellt wurde. Es erfreut sich großer Popularität und wird mittlerweile von verschiedenen Oracle-Produkten unterstützt.

Apache Kafka organisiert Nachrichten über verschiedene Topics. Ein Produzent sendet Nachrichten in ein Topic und jede

einzelne Nachricht lässt sich immer von mehreren Konsumenten konsumieren (Publish-Subscribe-Verfahren). Konsumenten können auch erst Stunden oder Tage später Nachrichten am Stück konsumieren, ohne dass dies einen Einfluss auf die Performance hat.

Abbildung 4 zeigt, wie sich Kafka mit dem Service Bus von Oracle integrieren lässt. Wie man sieht, kann der Service Bus sowohl von einem Kafka Topic lesen als auch in ein Kafka Topic schreiben.

Wir können also die Nachrichten von Kafka direkt vom Service Bus konsumieren und dann wie bisher an die Backend-Systeme weiterleiten. Darüber hinaus können Dashboards oder BI-Applikationen Informationen auch direkt aus Kafka konsumieren, ohne über den ESB gehen zu müssen. Mit dem Ersetzen von JMS durch Kafka entfällt ein Engpass. Die Frage bleibt nun aber, ob und wann der Service Bus zum nächsten Flaschenhals wird.

Wie bereits erwähnt, lässt ein ESB nur einfache Operationen auf dem Ereignisstrom zu, da er zustandslos agiert, was auch für den Service Bus gilt. Für die Analyse des Ereignisstroms wären die Funktionalitäten des sogenannten „Complex Event Processing“ (CEP) sehr hilfreich, wie

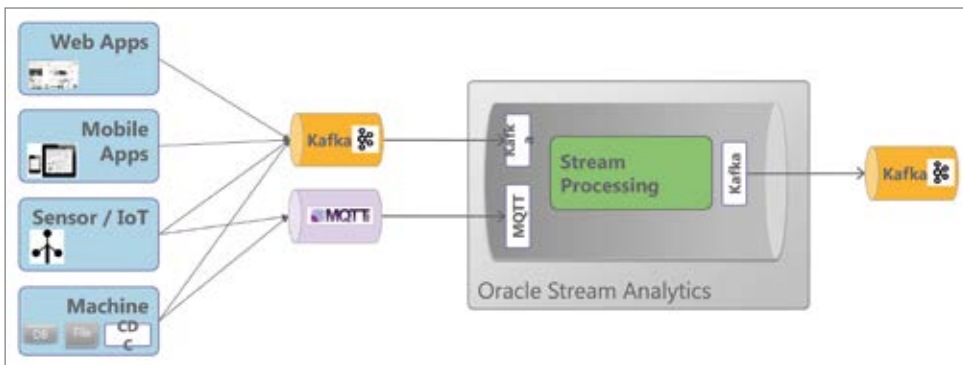


Abbildung 6: Integration von Apache Kafka mit Oracle Stream Analytics

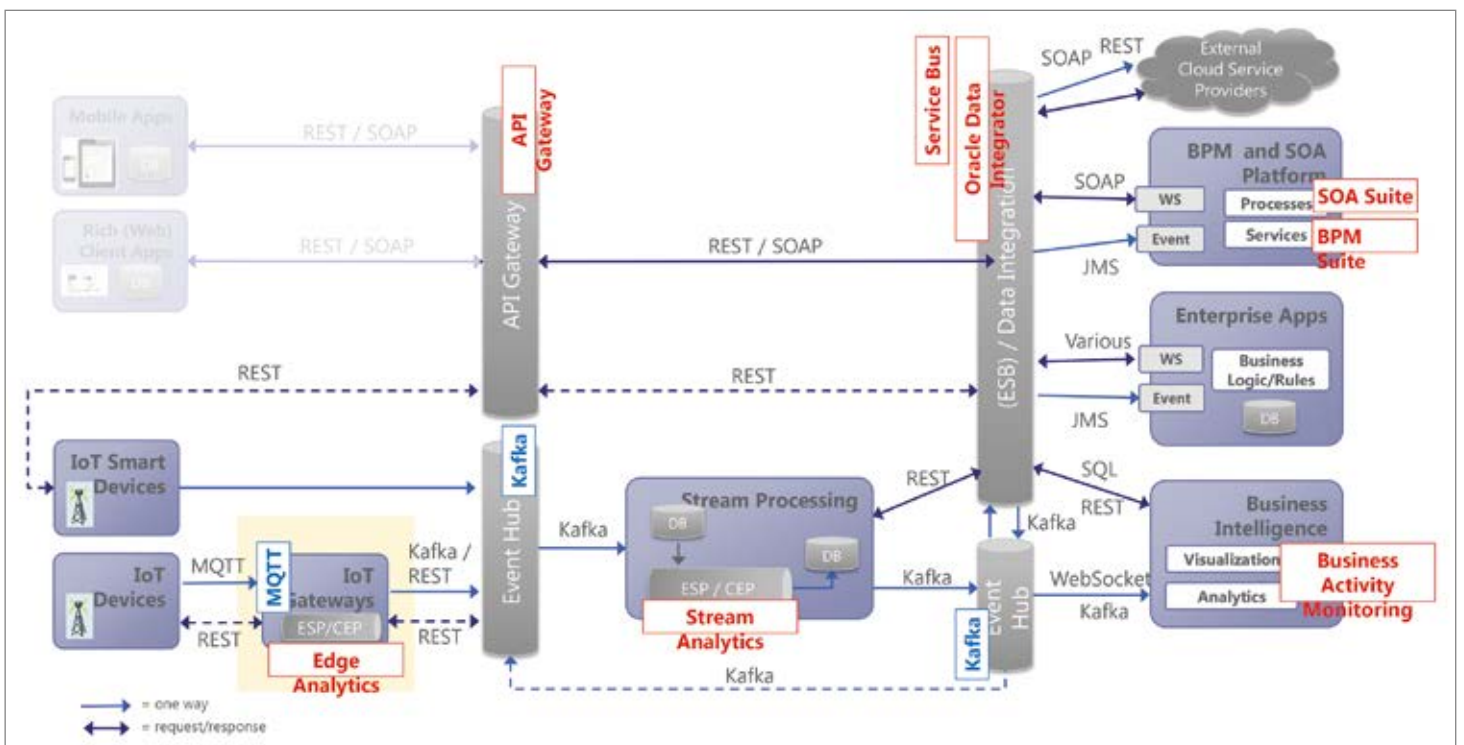


Abbildung 7: Erweiterung mit Stream Processing auf der Ebene des IoT-Gateway

sie von Stream-Processing-Plattformen angeboten werden. Dazu zählen unter anderem Aggregationen von mehreren Ereignissen über Zeitfenster, Erkennen von Patterns über mehrere unterschiedliche Ereignisse und raumbezogenen Operationen. Der nächste Schritt ist daher die Erweiterung der Architektur um eine Stream-Processing-Infrastruktur.

Stream-Processing-Infrastruktur für Analysen des Ereignisstroms

Um den riesigen Ereignisströmen Herr zu werden, die von den IoT-Smart-Devices beziehungsweise IoT-Gateways im Event Hub landen, lässt sich eine Stream-Processing-Infrastruktur einsetzen. Diese erlaubt es, die Ereignisse minimaler Latenzzeit zu analysieren und zu entscheiden, ob und – wenn ja – welche Aktion ausgeführt werden soll. *Abbildung 5* zeigt die Erweiterung der Architektur um eine Stream-Processing-Komponente.

Oracle bietet dabei mit Oracle Stream Analytics (OSA) eine sehr mächtige Plattform, die alle oben erwähnten CEP-Funktionalitäten anbietet. Sie unterstützt Apa-

che Kafka sowohl als Konsument wie auch als Produzent (*siehe Abbildung 6*).

In der Architektur agiert das Stream Processing quasi als Vorfilter. Die Nachrichten werden vom Event Hub konsumiert und durch einfache oder komplexe Analytik-Verfahren auf die Menge reduziert, für die es zu reagieren beziehungsweise zu agieren gilt. Dies können beispielsweise alle Ereignisse über einem bestimmten Schwellenwert sein oder aber das Erkennen eines ausbleibenden/ fehlenden Ereignisses aufgrund eines komplexen Pattern Matching über mehrere Ereignisse und die Zeit hinweg. Diese Erkenntnisse führen dann zu Aktionen in den Backend-Systemen, die vom Stream Processing über einen Service-Aufruf auf dem ESB ausgelöst werden.

Mit dem Stream Processing wurde die Architektur um eine mächtige Komponente erweitert, die hochgradig skalierbar ist und auch fehlertolerant aufgesetzt werden kann. Die Erweiterung der Backend-Architektur um Event Hub und Stream Processing macht es möglich, die massiven Ereignisströme effizient und in Echtzeit abzuarbeiten.

Die Frage, die sich als Nächstes stellt, ist, ob sich das Nachrichtenvolumen, das

von den IoT-Smart-Devices und IoT-Gateways erzeugt wird, überhaupt über das bestehende Netzwerk transferieren lässt. Falls nicht, lässt sich das Netzwerk in der Kapazität mit vertretbarem Aufwand erweitern? Falls dies auch nicht möglich oder finanziell gangbar ist, dann gilt es, die Nachrichtenmenge bereits auf den IoT-Gateways oder Smart-Devices sinnvoll zu reduzieren.

Stream Processing auf den IoT-Gateways

Mit dem IoT-Gateway steht ein Gerät mit Intelligenz zur Verfügung; im Prinzip ein kleiner Computer, der zu mehr genutzt werden kann als zum reinen Empfangen und Weiterleiten der Daten von den angehängten IoT-Devices. Wie in *Abbildung 7* gezeigt, kann ein Event Processing auch auf dem Gateway erfolgen.

Es handelt sich hier allerdings um eine wesentlich leichtgewichtige Variante als im Backend. Oracle bietet mit Oracle Edge Analytics genau so eine Komponente an, quasi den kleinen Bruder von Oracle Stream Analytics, die sich auch auf Embedded Rechnern wie einem Raspberry

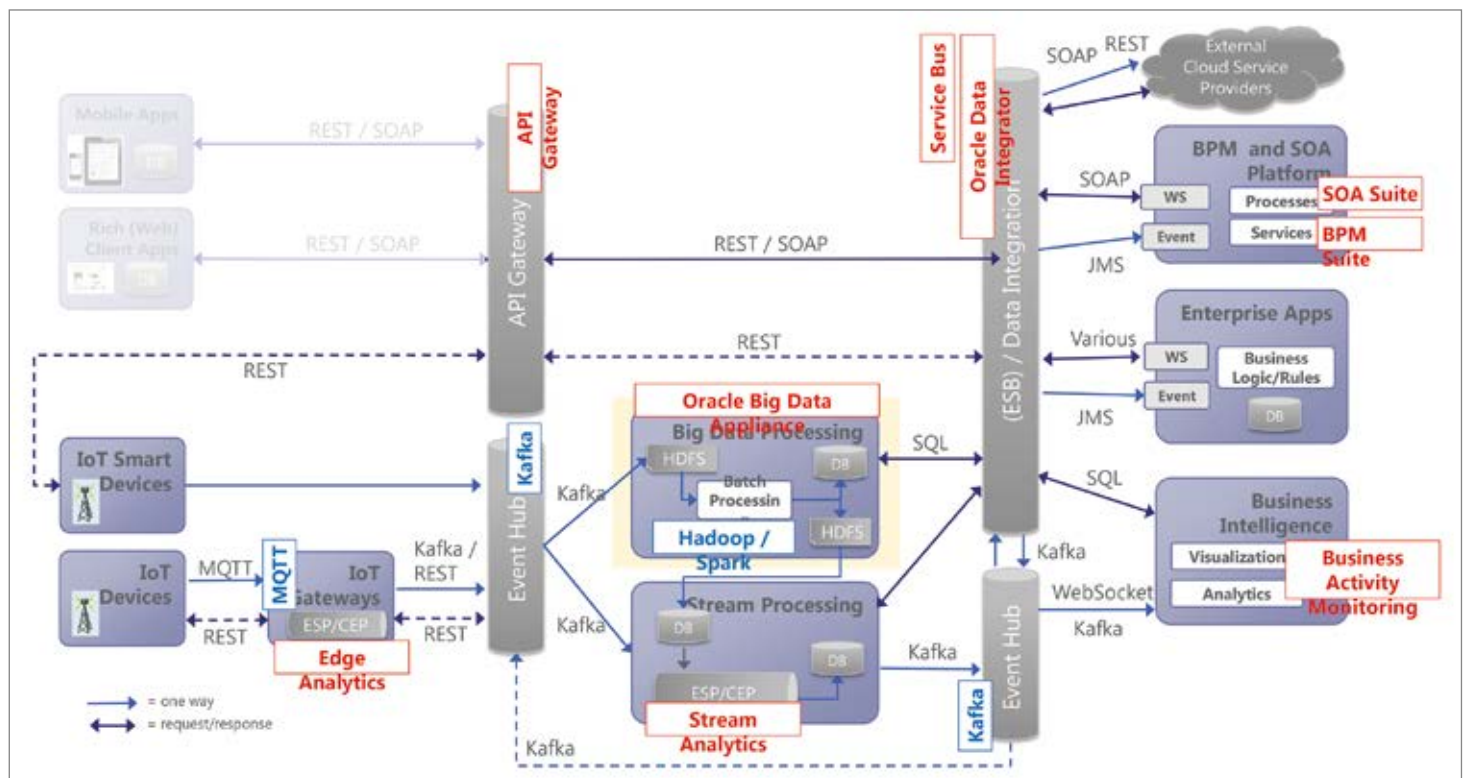


Abbildung 8: Erweiterung um eine Big-Data-Infrastruktur für die Speicherung der Rohdaten und mögliche Big-Data-Batch-Verarbeitung

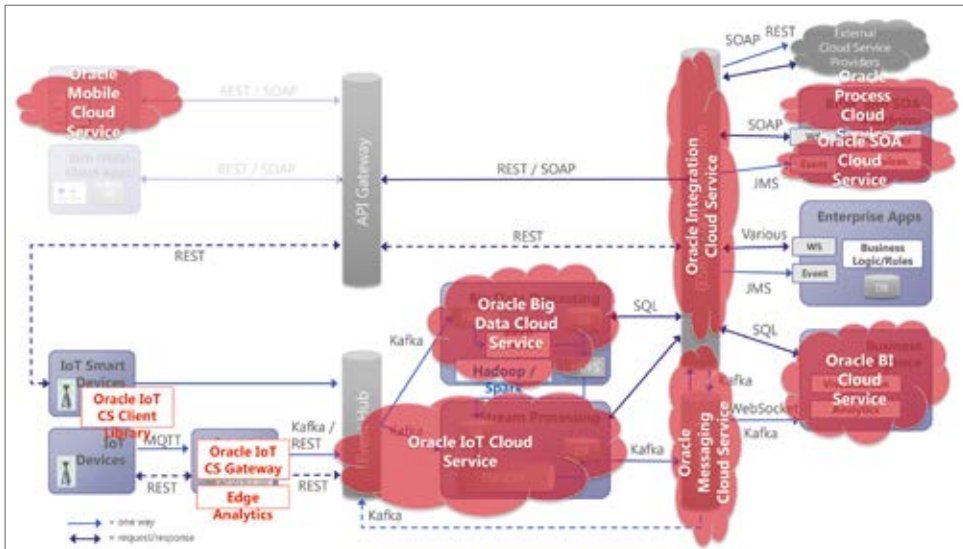


Abbildung 9: Oracle Cloud Services und Internet of Things

PI betreiben lässt. Damit lassen sich erste Filter, Aggregationen oder aber ein Event-Pattern-Matching direkt auf dem Gateway erledigen, sodass sich die Datenmenge, die es ins Backend zu übertragen gilt, entsprechend reduziert. Dies wird auch als „Edge Computing“ beziehungsweise „Fog Computing“ bezeichnet.

Nachteil einer solchen Erweiterung ist jedoch, dass man im Backend nicht mehr die Roh-Informationen des Sensors kennt, sondern nur noch die abgeleiteten Informationen vom Gateway. Rohdaten sind jedoch oft Grundlage für das Machine-Learning im Big-Data-Umfeld; das wird im nächsten Schritt thematisiert.

Erweiterung um eine Big-Data-Infrastruktur für Batch-Analysen

Mit der bisherigen Ergänzung der Architektur um Stream Processing (sowohl zentral wie auch auf dem Gateway) und Event Hub ist sichergestellt, dass die Ereignisse von den IoT-Devices rasch und zuverlässig verarbeitet werden können. Die Analyse beschränkt sich allerdings auf die aktuellen, gerade eintreffenden Daten. Um auch eine Analyse über historische Daten zu ermöglichen, kann die Architektur, wie in *Abbildung 8* gezeigt, um ein Big-Data-Processing ergänzt werden. Eine Infrastruktur dafür sollte zurzeit die Komponenten aus dem Hadoop- und Spark-Ökosystem unterstützen.

Oracle bietet mit der Big Data Appliance eine Möglichkeit, eine Hadoop-beziehungsweise Spark-Plattform als Appliance zu nutzen, also inklusive der Hardware. Wie in *Abbildung 8* dargestellt, konsumiert die Big-Data-Plattform ebenfalls die Ereignisse aus dem Event Hub, jedoch lediglich mit dem Ziel, diese als Rohdaten im Hadoop Distributed Filesystem (HDFS) für die spätere Nutzung abzuspeichern. Diese Daten dienen dann als Input für komplexe Analysen und Verarbeitungen. Dies sind immer Batch-Verarbeitungen, die vom Hadoop-Cluster parallel ausgeführt werden. Die entsprechenden Resultate, etwa ein Modell aus dem Machine-Learning, können dann zum Beispiel auch aus dem Stream Processing verwendet werden, um in Echtzeit die Ereignisse gegen das Modell zu testen (etwa für Predictive Analytics Use Cases).

Oracle Cloud Services und Internet of Things

Die Architektur lässt sich wie gezeigt im eigenen Data Center (On-Premise) aufbauen, es gibt aber auch entsprechende Cloud-Dienste von Oracle, mit denen sich das Ganze ebenfalls umsetzen lässt. *Abbildung 9* zeigt das Oracle-Cloud-Angebot auf die Architektur abgebildet.

Mit Ausnahme der Devices und Gateways können alle Funktionalitäten auch als Cloud-Dienst genutzt werden. Dabei ist natürlich auch eine Kombination von Cloud-

Dienst und eigenem Data Center denkbar. So kann der Oracle IoT Cloud Service für den Empfang und die Filterung der Nachrichten genutzt werden. Die Ergebnisse könnte man dann über den Oracle Integration Cloud Service in das eigene Data Center leiten, um sie den Backend Services zur Verarbeitung zuzuführen.

Fazit

Es lassen sich mit einer bestehenden Infrastruktur sehr wohl erste IoT-Anwendungen umsetzen. Dabei ist allerdings wichtig, sich jederzeit über das zu erwartende zusätzliche Nachrichtenvolumen im Klaren zu sein und die Grenzen der eingesetzten Komponenten zu kennen. Es gibt heute keine einzelne Architektur, die allen Anforderungen von IoT gerecht werden kann – zu vielfältig sind die Anwendungsfälle.

Gefordert ist eine modulare und skalierbare Architektur, die es ermöglicht, je nach Bedarf zusätzliche Komponenten hinzuzufügen. In den letzten Jahren sind einige neue, spezialisierte Produkte im Big-Data- und Fast-Data-Umfeld entstanden, um das Nachrichtenvolumen einer IoT-Lösung zu bewältigen. Diese lassen sich vor eine bestehende Anwendungslandschaft schalten; damit werden die Nachrichten auf eine Menge reduziert, mit der die bestehenden Backend-Systeme umgehen können. Entsprechende Angebote gibt es dabei sowohl als Cloud-Dienst als auch für die eigene Backend-Infrastruktur.



Guido Schmutz
guido.schmutz@trivadis.com