Internet of Things: Referenz-Architektur

Marcel Amende, ORACLE Deutschland B.V. & Co. KG

Vor zwei Jahren befand Kanzlerin Merkel noch, das Internet sei „... für uns alle Neuland“. In diesem Jahr findet sie Facebook bereits „... so schön, wie ... eine ordentliche Waschmaschine.“ Dabei ist die nächste tiefgehende Umwälzung in der IT bereits in vollem Gange, getrieben von einer explosionsartigen Verbreitung und Vernetzung kleiner, kostengünstiger und intelligenter Geräte.

In Deutschland wird häufig der Begriff „Industrie 4.0“ verwendet, der wie eine unnötige Selbstbeschränkung erscheint. Bei aufgeschlossener und kreativer Betrachtung der entstehenden technischen Möglichkeiten zeigt sich, dass es um weit mehr gehen kann als die Vernetzung von Fabriken und Maschinen. Tatsächlich werden alle Lebensbereiche durchdrungen, das uns bekannte Internet verändert sich zu einem „Internet der Dinge“ (IoT). Salopp gesagt, hat Merkels Waschmaschine heute Facebook. In Anbetracht der Bedeutung des Themas drängt es sich auf, belastbare Referenz-Architekturen für IoT-Umsetzungen zu entwickeln. Der Artikel zeigt dies aus einer On-Premise- und Cloud-Sicht.

Anforderungen einer IoT-Lösung

Eine vollständige IoT-Lösung *[1]* besteht aus verschiedenen Komponenten *(siehe Abbildung 1)*:

- *Geräte*
  Weit verteilte oder sogar mobile Geräte und Maschinen interagieren, mit einer Vielzahl von Sensoren und Aktoren ausgestattet, messend, steuernd und regelnd möglichst energieeffizient mit ihrer Umwelt. Sie sammeln dabei eine Vielzahl von Daten ein: Position, Bewegung, Umgebungstemperatur, Luftfeuchte, Vibrationen, aber auch Nutzungs- und Produktionskennzahlen. Dies kann vom schaltbaren Leuchtmittel über das regelbare Heizungs-Thermostat, die fernwartbare Waschmaschine bis hin zur Produktionsmaschine reichen. Apple Manager Jeff Williams bezeichnete jüngst das kommunizierende Auto als „ultimatives Mobilgerät“.

- *Gateways*
  Das Gateway sammelt und verarbeitet Daten von einem oder mehreren IoT-Geräten in seiner näheren Umgebung ein. Es nimmt eine intelligente Vorverarbeitung und Filterung der Daten vor und wandelt sie für eine sichere Übertragung an zentrale Systeme in ein möglichst standardisiertes Format. Zusätzlich werden dem Gateway Wartungs-, Verwaltungs- und Diagnose-Aufgaben übertragen: Es überwacht, (de-)aktiviert die Geräte der Umgebung und bietet Zugang etwa für die Verteilung von Software-Updates. Je nach Ökosystem des jeweiligen Anbieters kann die Gateway-Funktionalität in Netzwerkgeräte, Mediacenter oder Heizungssteuerungen eingebettet sein.

- *Netzwerk*
  Gateways kommunizieren üblicherweise in verschiedenen Netzwerken. Geräte können in persönlichen Netzen (PAN) per USB oder Low-Energy Bluetooth angebunden sein. Oft existieren bereits lokale drahtgebundene Ethernet- oder drahtlose WiFi-Netzwerke (LAN). Server sind in weitreichenden Netzen (WAN) über das öffentliche Internet, auch per Glasfaser- oder Satelliten-Verbindungen erreichbar. Dieses kann über sichere Tunnel zwischen Client und Server virtuell privat (VPN) erfolgen. Hier besteht die Herausforderung darin, die Netzwerk-Kommunikation zuverlässig, sicher, vertraulich, aber auch möglichst schnell und effizient zu gestalten.

- *Server*
  Die zentrale Infrastruktur übernimmt die weitere Filterung, Speicherung und

```
// I/O Port Nr. 23 für den Zugriff öffnen
GPIOPin myLED = DeviceManager.open(23);
// LED einschalten
myLED.setValue(true);
```

Listing 1

Abbildung 1: Die IoT-Lösungs-Komponenten

Analyse der übermittelten Daten. Diese können je nach Anforderung in unstrukturierter, strukturierter, anwendungsoptimierter und voraggregierter Form verwaltet werden. Oft werden hier mit Datenstrom-Analysen und Batchverfahren gleichzeitig parallel verlaufende Auswertepfade angewendet, um sowohl schnell Aktionen auslösen als auch zum Zeitpunkt der Erhebung unbekannte Fragestellungen auf historischen Daten und Rohdaten beantworten zu können.

- *Applikationen*
  Schließlich bleibt noch die Einbindung von Unternehmensapplikationen, um die aus der Vernetzung gewonnene Informationsfülle im Sinne der Geschäftsprozesse des Unternehmens zu nutzen. Neben Infrastruktur- und Lösungs-Design ist hier die unternehmerische Intelligenz und Kreativität gefragt, um aus innovativen IoT-Lösungsansätzen ein profitables Geschäft zu machen.

In der Folge werden die IoT-Komponenten-Bausteine mit Oracle-Technologien belegt, um zu einer tragfähigen Gesamtlösung zu kommen.

IoT-Geräte und Sensoren mit Java ME 8

Software für die Geräte im Internet der Dinge zu entwickeln, kann herausfordernd sein, da wegen weiter Verbreitung (auch räumlich), beschränkter Ressourcen, Variantenvielfalt und meist langer Lebensdauer manuelle Eingriffe in die Geräte-Software nicht möglich sind. Java-Entwickler sind hier in einer komfortablen Situation: Mit der Java Micro Edition (ME) bietet sich die Möglichkeit, auf eine standardisierte und wohlbekannte Entwicklungsplattform zu setzen, um die Abhängigkeiten von Hardware und Betriebssystem zu eliminieren. So lassen sich bereits erlernte Java-Fähigkeiten nutzen und proprietäre, teure Speziallösungen vermeiden. Dies beschleunigt und vereinfacht die Umsetzung neuer, eventuell noch experimenteller IoT-Anwendungen und Ideen mit unbekanntem Erfolgspotenzial.

Das Java ME 8 SDK ist ein für Embedded-Entwicklung optimierter Werkzeugkasten, der – unter Beachtung der limitierten Ressourcen und Bandbreiten von Embedded-Systemen – die Entwicklung von sicheren und zuverlässig betreibbaren Anwendungen ermöglicht. Das SDK zielt auf kleinere und mittlere Embedded-Systeme mit mindestens 128 KB RAM und 1 MB Flash oder ROM, wie etwa Sensoren, Netzwerk-, Mess- oder medizinische Geräte, ab.

Das Java-ME-8-API ist eine strikte Untermenge von Java SE 8, bietet aber viele Funktionalitäten, um spezifische Embedded-Anforderungen zu unterstützen: Eher ungewöhnlich für Java, ermöglicht das Device-I/O-API dem Entwickler einfachen Zugriff auf die Geräte-Peripherie und Bussysteme, um beispielsweise Sensoren und Micro Controller anzusteuern. Mit zwei Zeilen Java-Code lassen sich so Leuchtsignale über die I/O-Ports eines IoT-Geräts schalten *(siehe Listing 1)*.

Verschiedene Referenz-Implementierungen von Java ME 8 sind verfügbar, etwa die Raspberry-PI- und die Qualcom-IoE-Plattform. Beide sind mit ARM-Prozessoren ausgestattet. Zudem stehen Vorabversionen für Cortex-M4-basierte Systeme von Freescale und STM zum Download bereit. Die spezifischen Eigenschaften der Systeme werden unterstützt, indem zum Beispiel die I/O-Ports und Bussysteme der Geräte vorkonfiguriert sind. Beim Gerät von Freescale betrifft dies den Zugriff auf das integrierte Gyroskop und den Kompass, was die Entwicklung von Applikationen etwa für Datenbrillen erleichtert, die eine räumlichen Orientierung erfordern.

Java SE Embedded und Oracle Event Processing

Gateways sind leistungsfähigere Embedded-Systeme wie beispielsweise Industrie-PCs. Hier stehen mit der Java Standard Edition (SE) und Java SE Embedded ebenfalls die passenden und gegebenenfalls modularen Java-Versionen zur Verfügung. Sie übernehmen die dezentrale Kommunikation mit den lokalen Sensorsystemen und transportieren die relevanten Sensordaten auf sicherem Wege in das zentrale Rechenzentrum. Java SE ist in diesem Rahmen geeignet, komplexere Logik und Anwendungen zu betreiben. Es enthält eine Embedded-Java-Datenbank für die lokale Datenverwaltung und Auswertung mit SQL.

Auch das Oracle Event Processing (OEP) lässt sich mit Java SE Embedded betreiben. So kann komplexe Entscheidungslogik auf den eingehenden Datenströmen graphisch modelliert und abgebildet werden. Ein ty-

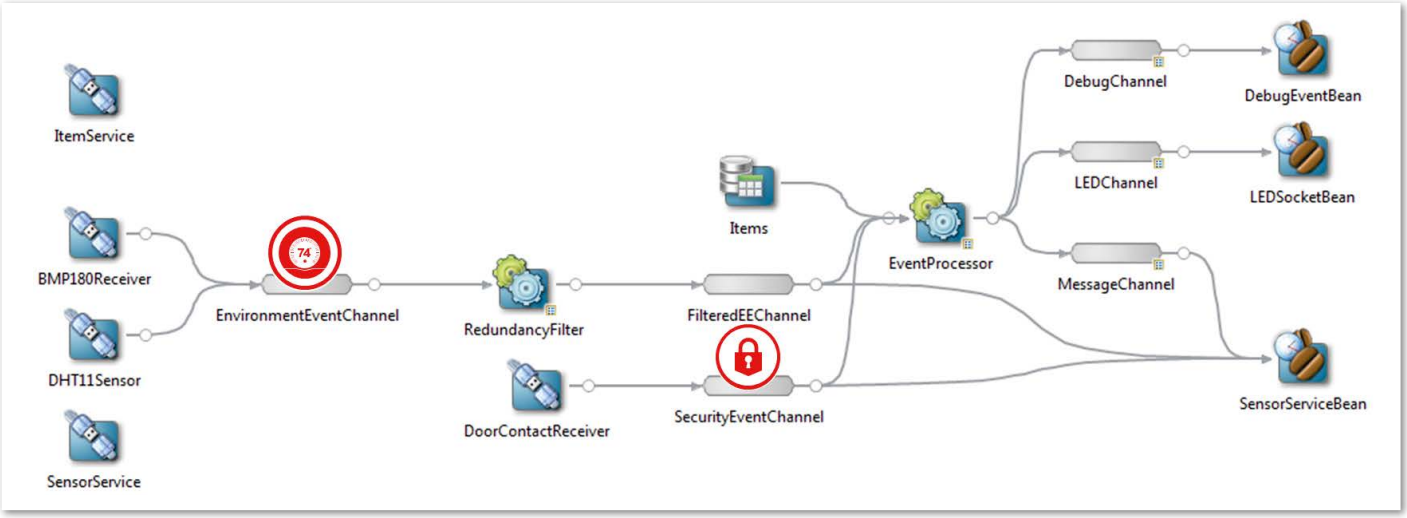

Abbildung 2: Ereignisverarbeitendes Netzwerk in Oracle Event Processing

pischer Einsatzbereich von Event Processing ist die Filterung und Vorverdichtung von Gerätedaten. Sie sorgt dafür, dass nur wirklich relevante Daten im Bedarfsfall über sichere Transportkanäle an zentrale Systeme übertragen werden. *Abbildung 2* zeigt ein in Oracle Event Processing modelliertes, ereignisverarbeitendes Netzwerk für ein Anwendungsbeispiel aus der Transport-Logistik.

Adapter empfangen die Sensordatenströme verschiedener Sensoren, die mit dem Gateway in einem Transport-Container verbaut sind. Temperatur, Luftfeuchtigkeit und Türschließkontakte werden kontinuierlich überwacht und mit einer – in einer lokalen Java-Datenbank abgelegten – Beladungsliste abgeglichen. Bei Zuständen, die auf Verderb der Ware durch Überschreiten der Temperatur-Schwellwerte oder auf einen möglichen Diebstahl bei unerlaubtem Öffnen der Türen hindeuten, werden in Echtzeit Alarme und Benachrichtigungen ausgelöst, die an ein Mobilgerät oder die zentrale Unternehmens-IT für die Weiterverarbeitung gesendet werden *(siehe Abbildung 3)*.

Die Demonstration eines Prototyps, der Java ME, Java Embedded und OEP in Kombination nutzt, ist in YouTube eingestellt *(siehe „https://www.youtube.com/watch?v=WBmZnOoRWhI“)*.

Abbildung 3: Demo einer sensorgetriebenen Transportüberwachung

Netzwerk und Sicherheit

Weitverbreitete Sensoren und Geräte bilden ein IoT-Netzwerk. Am zentralen Eingangspunkt für die Datenströme dieser Geräte, etwa in der DMZ eines Rechenzentrums, ist ein Security-Gateway wie das Oracle API Gateway erforderlich. Es dient der gesicherten Kommunikation über potenziell unsichere IP-Netze wie das Internet und ist vollständig transparent für die Anwendungen. Dafür kann ein sicherer Tunnel eingerichtet oder ein sicherer Endpunkt bereitgestellt werden. Zwei wichtige Aufgaben werden vom Gateway übernommen: die Schnittstellen-Adaption, also die Bereitstellung von Zugangspunkten für verschiedene Protokolle (SOAP, REST) und Datenformate (XML, JSON), sowie die Sicherheitskonfiguration. Die Grundfunktionalitäten in Sachen Sicherheit sind die Authentifizierung (Wer?), die Autorisierung (Was?), die Sicherstellung von Nachrichtenintegrität (Woher? Unverändert?) und die Verschlüsselung (Vertraulichkeit). Das ist vergleichbar mit der Bordkartenkontrolle an einem Flughafen, bei der sichergestellt wird, dass die richtige Person ins richtige Flugzeug steigt.

Das Oracle API Gateway bietet aber noch weitere Funktionalitäten. Entsprechend einer Sicherheitskontrolle am Flughafen, bei der Gepäck und Person eingehend durchsucht werden, kann es in den Nachrichten nach bedrohlichen Inhalten suchen. Angreifer versuchen typischerweise, Schadcode in die übermittelten Daten einzubauen (Viren), durch das Variieren der Anfrage-Parameter mehr Daten zu erhalten ((SQL-)Injection) als vorgesehen oder den Dienst oder Server durch gezieltes Überlasten außer Funktion zu setzen (Denial of Service (DOS), XML Bombs).

Man schützt sich vor Angreifern, indem man sicherstellt, dass ein bereitgestellter Dienst genauso wie vorgesehen genutzt wird. Noch vor der Weiterleitung an den eigentlichen Dienst sucht das Gateway nach Viren, injizierten SQL-Statements, überprüft Parameter-Signaturen auf ihre Gültigkeit, schränkt Größe und Komplexität von Nachrichten ein, gleicht Absender-Adressen mit Black- und White-Listen ab und schränkt die erlaubte Nutzungshäufigkeit in Abhängigkeit des Aufrufers (Throttling) ein.

Serverseitige Verarbeitung von IoT-Daten

Milliarden von Menschen interagieren heute im Internet. Die elektronische Ausstattung und Vernetzung von Dingen wird die Zahl der Sensoren und Aktoren im Internet der Dinge beträchtlich erhöhen. Eine wahre Flut von Daten ist zu beherrschen. Für das Daten-Management bieten sich heute vier Varianten an, die in beliebiger Kombination nutzbar sind:

- *Relationale Datenbanken*
  Die klassische relationale Oracle-Datenbank stellt die komfortabelste und flexibelste Möglichkeit zur Datenverwaltung dar. Sie eignet sich für die Speicherung von strukturierten und unstrukturierten Daten und für alle Last-Szenarien, von transaktionaler bis zu In-Memory-Verarbeitung.
- *Data Warehouse*
  Im Data Warehouse werden die Daten nach bestimmten Dimensionen voraggregiert und somit auswertungsorientiert vorgehalten. Das spart Speicherplatz und beschleunigt das Abfragen der Daten für bekannte Fragestellun-

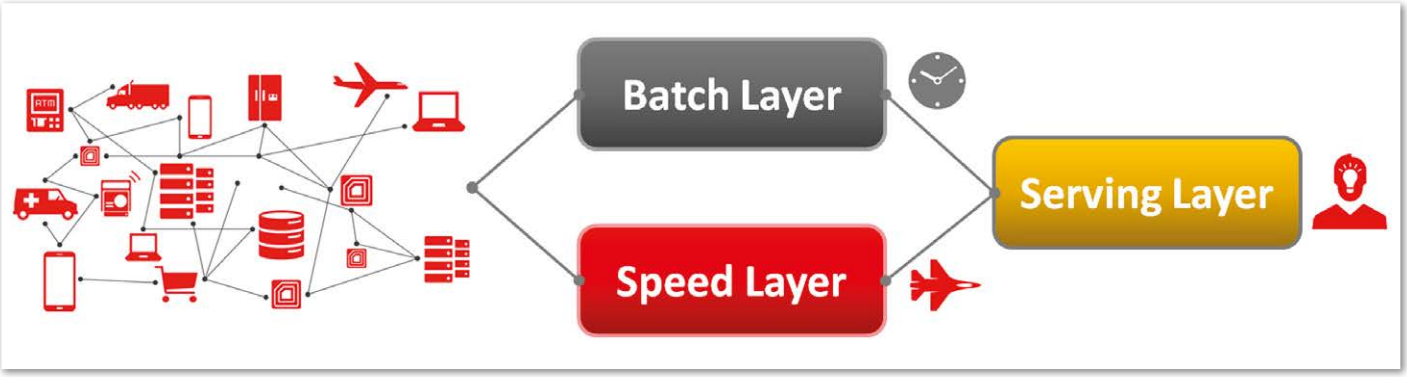

Abbildung 4: Lambda-Architektur

gen. Die Daten liegen allerdings nicht mehr in Rohform vor. Neue Fragestellungen bedürfen gegebenenfalls einer neuen Modellierung der Daten.

- *NoSQL-Datenbanken*
  Hier werden die Daten als Anwendungsobjekte in Schlüssel-Wert-Paaren vorgehalten. Auf die Objekte wird von Anwendungen in direkt konsumierbarer beziehungsweise instanziierbarer Form zugegriffen.
- *Verteilte Dateisysteme (Big Data/Hadoop)*
  Ist der Wert der zu verwaltenden Daten eher unbestimmt und sollen Daten in ihrer Rohform einschließlich aller Redundanzen gespeichert werden, greift man auf kostengünstige, Datei-basierte Verfahren zurück. Hier werden die Daten unstrukturiert und in großen Blöcken (Hunderte MB) gespeichert. Dadurch lassen sich zu späteren Zeitpunkten beliebige, auch bei der Erfassung der Daten völlig unbekannte Fragestellungen beantworten. Dafür müssen aber verteilte und hochparallele Verarbeitungsprozesse (Map/Reduce) über die Datenblöcke laufen, die in kaskadierenden Batchläufen die unstrukturierten Daten interpretieren, sortieren und verdichten, bis eine Ergebnismenge als Liste von Schlüssel-Wert-Paaren ermittelt ist.Alle genannten Verfahren zur Datenverwaltung haben ihre spezifischen Vor- und Nachteile, weshalb sie oft in Kombination angewendet werden. Sie teilen den Nachteil, dass eine Auswertung der Daten in Echtzeit nicht möglich ist. Im Internet der Dinge verlieren Informationen jedoch schnell an Wert: Überhitzt eine Maschine, muss sie sofort abgeschaltet werden, bevor ein Lagerschaden zu größeren Beschädigungen führt. Braucht die nachgelagerte Daten-Analyse in Form eines Batchlaufs zu lange, lässt sich der Schaden nicht verhindern. Für Reaktionen in Echtzeit sind in IoT-Architekturen daher parallele Verarbeitungsprozesse vorgesehen.

Lambda-Architektur für parallele Echtzeitverarbeitung

Bei der Lambda-Architektur führt man parallel zu einem Batch Layer, der alle verfügbaren Daten in Form von Batch-Prozessen verarbeitet, einen Speed Layer ein. Dieser füllt durch eine Echtzeitverarbeitung der Eingangs-Datenströme die zeitliche Lücke, die der Batch Layer hinterlässt *(siehe Abbildung 4)*.

Für die Verarbeitung hochvolumiger IoT-Eingangsdatenströme eignet sich der Oracle Stream Explorer. Er ermöglicht die Korrelation, Filterung und Verarbeitung von Datenströmen auch ohne IT-Kenntnisse, beispielsweise durch Fachabteilungen. Immer ausgehend von einer Vorschau des Live-Datenstroms, können so Muster in der Abfolge der Ereignisse eines Datenstroms erkannt werden *(siehe Abbildung 5)*.

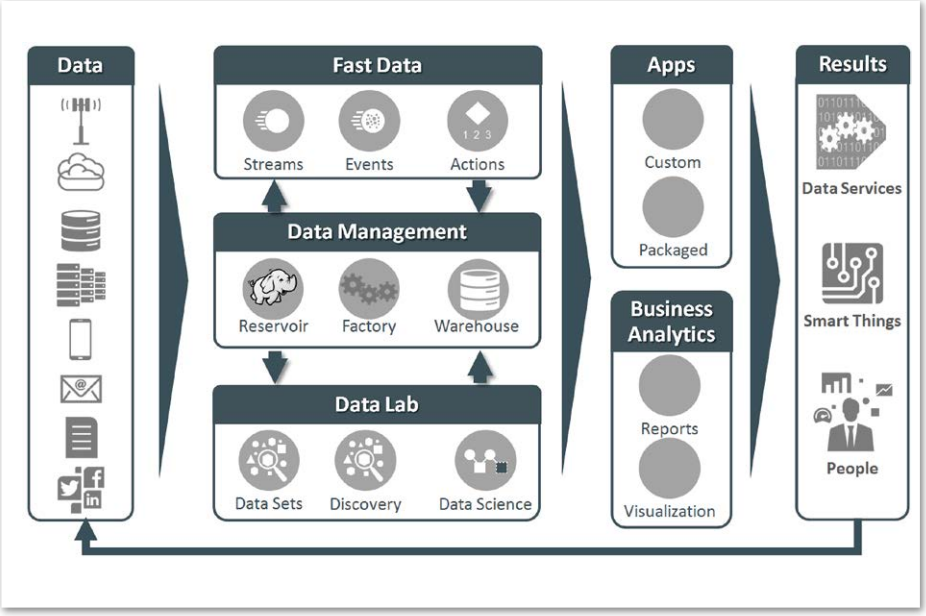

Abbildung 6: Oracle-Big-Data-Plattform

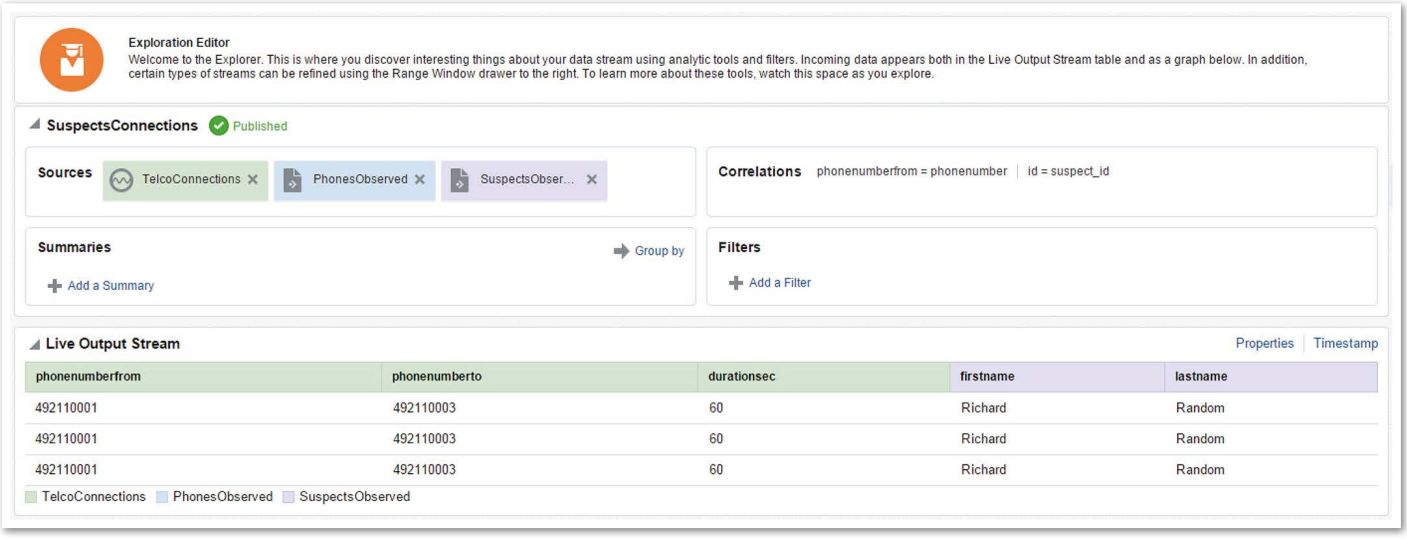

Abbildung 5: Stream Explorer, Analyse ausgehend vom Live-Datenstrom

Abbildung 7: Die Oracle-IoT-Cloud

Die Besonderheit ist, dass auch die zeitliche Beziehung zwischen Einzel-Ereignissen in die Analyse einfließen kann. Damit ist es möglich, weit komplexere Mustererkennungen anzuwenden als reine Schwellwert-Betrachtungen: Trends (kontinuierliche Auf- und Abwärtsbewegungen), Fluktuationen (Abweichungen vom Mittel), ausbleibende Ereignisse (Nicht-Ereignisse) oder „W"- und „M"-Formationen werden erkannt und Duplikate herausgefiltert.

Ein nachfolgender Serving Layer stellt sicher, dass es einen einheitlichen Zugang zu Daten und Analysen gibt. Neue Funktionalitäten der Oracle-Datenbank wie Big Data SQL ermöglichen, verteilte Hadoop-Filesysteme (HDFS) als externe Tabellen in der Oracle-Datenbank zu registrieren und per gewohnter SQL darauf zuzugreifen. Datenbank-Abfragen können sich über relational und blockweise gespeicherte Daten hinweg erstrecken. Spezialkenntnisse, wie die Programmierung verteilter Map/Reduce-Verarbeitungsprozesse in Java, sind nicht nötig.

Oracle Big Data Discovery geht den nächsten Schritt hin zu einfachen Daten-Analysen: Es bietet eine intuitive und einfache Benutzerschnittstelle für Hadoop. Daten aus Hadoop-Systemen können katalogisiert und grafisch analysiert werden. Durch die interaktiven Visualisierungen der Daten werden dabei immer neue Perspektiven eingenommen, um Muster in den Daten auch für Nicht-Mathematiker und -Statistiker erkennbar zu machen.

Das sich daraus ergebende Gesamtbild zeigt, dass die Lambda-Architektur in völligem Einklang mit der Oracle-Big-Data-Plattform ist *(siehe Abbildung 6)*. Der Batch Layer entspricht dem Block der Data-Management-Werkzeuge, von der Datenbank über Warehouse bis hin zu Hadoop. Der Speed Layer spiegelt die Fast-Data-Kategorie mit Stream Explorer wider und der Serving Layer wird durch das Data Lab mit Big Data Discovery repräsentiert.

Oracle-IoT-Cloud-Service

In einer aktuellen Umfrage *[2]* unter hundert Unternehmen zu den größten Frustrationspunkten in IoT-Projekten wurden neben einer generellen Verunsicherung um den entstehenden Hype vor allem Implementierungsschwierigkeiten, Datensicherheit, fehlende Standards und hohe Kosten als Hemmnis genannt. All diese Punkte adressiert der ab Sommer 2015 verfügbare Oracle-IoT-Cloud-Service *[3]*. Dieser wird von der Daten-Akquirierung über die Daten-Analyse bis zur Integration der Geschäftsprozesse und Unternehmens-Anwendungen alle benötigten Funktionen schnell, flexibel und bei deutlich reduziertem Risiko als Plattformdienst in der Oracle-Cloud bereitstellen. Die Schichten übernehmen dabei folgende Aufgaben *(siehe Abbildung 7)*:

- *Daten-Akquirierung*
  Für die Anbindung, Kontrolle und Verwaltung von IoT-Geräten stehen die IoT-Cloud-Service-SDKs und -APIs für verschiedene Programmier-Umgebungen und das IoT-Cloud-Service-Gateway zur Verfügung. Sie sorgen für eine sichere und vertrauenswürdige bidirektionale Kommunikation der Geräte mit der Cloud.
- *Daten-Analyse*
  Die eingehenden Datenströme können, in Echtzeit aggregiert, gefiltert, miteinander korreliert und analysiert werden. Wie vom Oracle Stream Explorer bekannt, können auch prädiktive Analysen und Trend-Erkennungen ausgeführt werden, die regelbasiert Alarme und Benachrichtigungen auslösen. Zudem stehen mit Big Data für IoT und dem Oracle Business Intelligence Cloud Service Lösungen für die Abfrage, Analyse und Visualisierung von IoT-Massendaten zur Verfügung.
- *Integration*
  Natürlich können vorhandene Unternehmens-Applikationen dynamisch mit Daten aus der IoT-Cloud versorgt werden, um Prozesslücken zu schließen und schnell auf kritische Ereignisse reagieren zu können.

Fazit

Mit Oracle-Standard-Produkten lässt sich eine komplette, durch die Nutzung von Java offene und zukunftssichere sowie in allen Schichten belastbare IoT-Architektur abbilden. Wenn man keine eigene IoT-Infrastruktur aufbauen möchte oder langfristige Investitionen in ein neuartiges Geschäftsumfeld scheut, bietet die Oracle IoT Cloud ab diesem Sommer die Möglichkeit, schnell und risikofrei eigene IoT-Anwendungen zu starten.

Weitere Informationen

[1] JD Edwards EnterpriseOne Internet of Things Platform, Oracle White Paper, 2015
[2] https://www.linkedin.com/grp/post/108418-6016658510509592578
[3] https://cloud.oracle.com/iot

Marcel Amende
marcel.amende@oracle.com