

The Oracle of Things

Arne Brüning
ORACLE Deutschland B.V. & Co. KG
Hamburg

Schlüsselworte

Internet-of-things, IoT, Java, JME, Java Micro Edition, Raspberry Pi, Event Processing, Complex Event Processing, NetBeans, Java Embedded Suite

Einleitung

Das Internet of Things (IoT) ist aktuell in aller Munde, und praktisch alle Analysten prognostizieren für diese Thematik ein starkes Wachstum über die nächsten Jahre. Das sollte Grund genug sein, sich einmal näher mit der Thematik zu beschäftigen. Allerdings wird unter dem Begriff IoT aktuell praktisch alles mögliche angepriesen. Passenderweise bietet Oracle bereits auch ein umfassendes Angebot zum IoT. Aber was genau ist damit eigentlich gemeint?

Für eine lauffähige Lösung ist meistens eine Kombination von mehreren Oracle-Produkten nötig. Hier stellt sich die Frage, welches Tool für welchen Zweck das richtige ist. Durch teilweise nebenläufige Funktionalitäten ist die Antwort manchmal nicht ganz eindeutig: it-depends-on. Auch geht es um die Frage, wie die verschiedenen Tools miteinander integriert werden sollen, hierfür gibt es verschiedene Möglichkeiten. Zusätzlich müssen auch Open Source Tools und Frameworks integriert werden.

Auch wenn das auf Slideware einfach aussieht, sind in der Praxis doch einige Herausforderungen zu meistern. Dieser Vortrag soll einen ersten Einstieg in die Thematik bieten und helfen, die ersten Klippen zu umschiffen. Dies wird anhand eines Prototypen demonstriert. Es wird gezeigt wie die Tools auf einem exemplarischen Raspberry Pi Sensordaten eingesammelt, vorverarbeitet und dann an einen Event-Server (=Laptop) über eine Netzwerkverbindung zur Weiterverarbeitung übergeben werden. Von hier werden die generierten komplex Events an die SOA Suite übergeben um hier entsprechende Geschäftslogik zum reagieren auf diese Events zu implementieren.

Nach diesem Vortrag sollten die Teilnehmer in der Lage sein, für eigene Projekte die passenden Oracle-Werkzeuge auszuwählen und zu lauffähig zu integrieren.

Oracle und das Internet of Things

Wie eingangs beschrieben, ist das IoT ein weites Betätigungsfeld. Oracle baut (noch) keine eigenen Kühlschränke, die mit Lieferanten sprechen. Aber Oracle bietet einiges an Komponenten, die man für den Aufbau eines solchen Systems gebrauchen könnte. Das folgende Schaubild der Industrial Internet Architecture veranschaulicht, für welche Bereiche des IoT Oracle heute bereits passende Produkte anbietet.

Industrial Internet Architecture

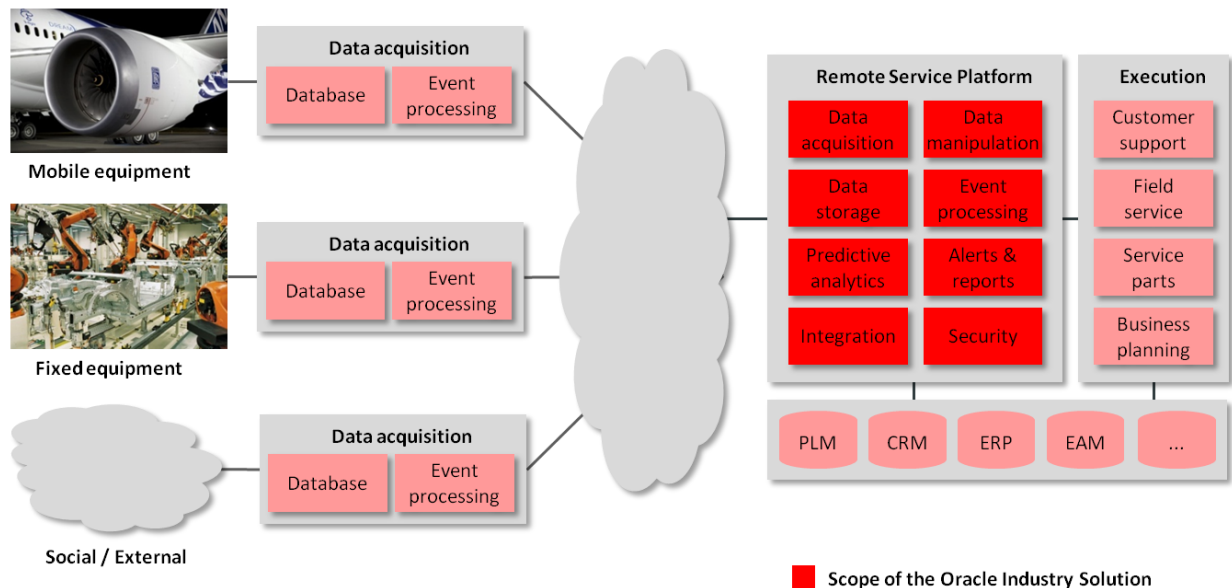


Abb. 1: Die Oracle Industrial Internet Referenz-Architektur

Dieser Vortrag stellt im weiteren vor, was genau Oracle in welchen Bereichen zu bieten hat.

Die Zutaten – was bietet Oracle

Java ME Embedded

Auf Geräte-Ebene bietet Oracle die Java ME Embedded für eine Vielzahl von Geräten an. Die JME eignet sich hervorragend, um auf dem Device selbst die Hardware anzusprechen und die Sensordaten auszulesen. Eine für einfache Entwicklungsumgebung kann z.B. ein Raspberry Pi genutzt werden. Eine IDE mit entsprechender Unterstützung liefert Oracle mit Netbeans gleich dazu.

Java Embedded Suite

Die Java Embedded Suite implementiert einen leichtgewichtigen Application Server, bestehend aus dem GlassFish for Embedded, der JavaDB und dem Jersey Web Services Framework um REST-Services anzubieten. Die Embedded Suite ist kann z.B. eingesetzt werden auf Gateway-Systemen.

Oracle Event Processing for Oracle Java Embedded

Das OEP Embedded erweitert die Embedded Suite um eine leichtgewichtige Implementierung des Oracle Event Processing. Genau wie die Embedded Suite, lässt sich das OEP Embedded z.B. auf einem Gateway-Device betreiben.

Oracle Event Processing

Die grosse Version des Oracle Event Processing läuft typischerweise auf einem Server in der Backend-Infrastruktur

Oracle SOA Suite

Die Oracle SOA-Suite integriert verschiedene Komponenten einer SOA-Infrastruktur in ein einheitliches Framework auf Basis der Service Component Architektur. Die SOA-Suite unterstützt beispielsweise den Standard Business Process Execution Language für die Orchestrierung im Backend.

Das Rezept: Wie passt das alles zusammen – ein Lösungsansatz

Für die Beispiellösung sollen Sensordaten erfasst werden. Die Sensoren werden an einen Raspberry Pi angeschlossen. Zur einfacheren mechanischen Anbindung wird ein Breadboard benutzt. Auf dem Raspberry Pi läuft ein Raspbian-Linux. Der Raspberry Pi Port bietet verschiedene Anschlussmöglichkeiten. Neben frei programmierbaren Pins, bietet der Pi auch einen I²C-Bus an. Dieser wird genutzt, um einen BMP180-Sensor Temperatur anzuschließen und damit Luftdruck-Daten zu erhalten.

Unter dem Raspbian-Linux wird ein Java ME Embedded installiert. Eine darauf deployte Anwendung nutzt die Device-Integration von JME, um auf einfache Weise an die Sensor-Daten zu gelangen.

Wie einfach sich an den Raspberry Pi angeschlossene Hardware aus Java ansprechen lässt, verdeutlicht das folgende Code-Beispiel:

```
GPIOPin led=DeviceManager.open(23);  
led.setValue(true);  
led.close();
```

Hiermit wird eine LED, die an Pin 23 angeschlossen ist, zum leuchten gebracht.

Als Entwicklungsumgebung dient Netbeans mit dem JME SDK. Dieses bringt einen Device-Emulator mit, so dass einfache Anwendungen ohne spezielle Hardware-Anbindung direkt auf dem Entwickler-PC getestet werden können. Ausserdem wird ein Device-Management mitgeliefert, so dass auf ein registriertes Device direkt aus der Netbeans-IDE die Anwendung deployed, gestartet und sogar debugged werden kann. Aus dieser Anwendung werden die Daten via REST-Service an die nächste Instanz, das OEP Embedded weitergegeben.

Auf einem Gateway-Device würden typischerweise die Daten von vielen Devices eingesammelt und über das Complex Event Processing zur Laufzeit auf bestimmte Muster untersucht werden. Hierzu wird ein sogenanntes Event-Processing-Network (EPN) definiert, die Suche nach Mustern kann über die SQL-ähnliche Sprache CQL (Continuous Query Language) umgesetzt werden. Grundsätzlich können die Devices über jedes beliebige Protokoll miteinander kommunizieren. Letztendlich handelt es sich um Java-Klassen, somit liegt die Wahl beim Entwickler. Für dieses Beispiel implementiert der eingehende Adapter des EDN einen REST-Service via Jetty.

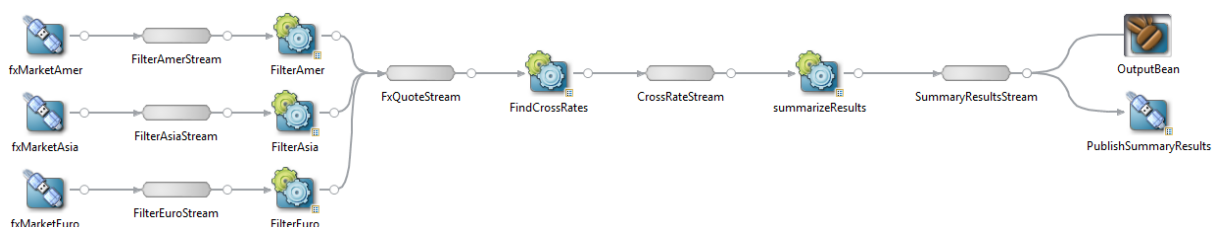


Abb. 2: Ein typisches Event-Processing-Network (EPN)

In einer sehr grossen Umgebung würden typischerweise mehrere Complex Event Processing Systeme hintereinandergeschaltet werden. So würden ggf. mehrere Gateway-Systeme die Datenströme an entsprechende Backend-Systeme liefern, auf denen ebenfalls ein Oracle Event Processing läuft. Hier würde die ‚große‘ Variante des OEP laufen, nicht die Embedded-Version.

Die Event-Processing Stufen lassen sich je nach Anforderung beliebig kaskadieren, aber irgendwann kommt man an den Punkt wo alle Datenströme durchsucht wurden und man bekommt als Ergebnis die entsprechenden komplex-Events. Auf diese sollte nun in irgend einer passenden Art reagiert werden, sonst hätte man sie ja gar nicht erst erstellen müssen. An dieser Stelle bietet sich die Oracle SOA Suite als Mittel der Wahl an. Auch die SOA Suite kennt Events, auch wenn die Implementierung nicht auf die extremen Durchsatzraten des CEP ausgelegt sind. Und seit Version 12c ist die Server-Variante des OEP in der Lage, entsprechende Events für die SOA-Suite direkt zu generieren. Auch wenn natürlich auch an dieser Stelle grundsätzlich eine Vielzahl von möglichen Verbindungsprotokollen zwischen OEP und SOA-Suite möglich wären, bietet sich an dieser Stelle der bereits fertige Mechanismus an.

Die SOA-Suite bietet sich wiederum dafür an, die complex Events weiterzubearbeiten und ggf. an weitere Systeme, die damit arbeiten sollen, entsprechend weiterzuleiten. Eines der herausragenden Merkmale der SOA Suite ist ja die Eigenschaft, mit einer Vielzahl unterschiedlicher Systeme interagieren zu können. Somit bietet Sie sich für so ein Szenario an. Mit der Option BPM-Suite liessen sich von hier auch komplexe Geschäftsprozesse, ggf. mit Benutzerinteraktion, steuern.

In diesem Vortrag werden die oben aufgeführten Schritte genauer beleuchtet, Schritt-für-Schritt durchgesprochen und teilweise auch live demonstriert. Die Teilnehmer lernen hier das Rüstzeug kennen, um hinterher eigene IoT-Anwendungen auf Basis von Oracle-Technologie umzusetzen.

Kontaktadresse:

Arne Brüning
Oracle Deutschland B.V. & Co. KG
Kühnehöfe 5
D-22761 Hamburg

Telefon: +49 (0) 40-89091 220
Fax: +49 (0) 177-5949 220
E-Mail: arne.bruening@oracle.com
Internet: www.oracle.com
fmw-deutsch.blogspot.com