

Umfassende Applikationsplattform für IoT und Social auf dem Oracle Software Stack

**Frank Closheim
inxire GmbH
Frankfurt**

Schlüsselworte

Java, IoT, BigData, Social Enterprise, Open Source.

Einleitung

Das IoT ist derzeit ein großer Hype in der IT-Branche, von dem sich viele Unternehmen neue und attraktive Geschäftschancen erwarten. Nüchtern betrachtet, unterscheiden sich IoT-Applikationen jedoch nur geringfügig von klassischen Web- oder Client-Server-Anwendungen. Ob nun M2M für die Kommunikation von Mensch-Maschine oder Maschine-Maschine steht, es werden immer Daten produziert auf die entsprechend reagiert werden muss. Aufgrund des enorm hohen und weiter rasant wachsenden Datenvolumens müssen IoT-Applikationen dabei allerdings um einiges performanter, skalierbarer und sicherer sein.

Entwickler von IoT-Systemen sind also mit den gleichen Herausforderungen konfrontiert wie Software-Entwickler seit Jahrzehnten: Man muss Objekte und Dokumente verwalten, den Lebenszyklus der Objekte managen, aufgrund von Events, Aktionen bzw. Prozesse anstoßen und abwickeln und last but not most important: Man muss sicherstellen, dass Security- und Compliance-Regeln eingehalten werden. Schließlich sollte die Inbetriebnahme der neuen IoT-Plattform, nicht aufgrund von unzureichender Einhaltung von Datenschutz- oder anderen Regularien scheitern.

Offensichtlich ist, dass kaum ein Unternehmen auf der grünen Wiese operiert, d.h. es wurde sehr viel in unternehmenskritische Anwendungen investiert. Das dort vorhandene Wissen – Daten und Prozesse – muss in die Entwicklung der IoT-Applikationen einfließen, Lösungen von Anfang an „integriert gedacht“ werden. Auch hochspezialisierte, vertikale IoT-Systeme müssen sich in die bestehende Applikationslandschaft einbetten lassen, um nachhaltig zum wirtschaftlichen Erfolg des Unternehmens beizutragen.

Eine der wesentlichen Herausforderungen für IoT-Entwickler ist zwar auch nicht neu, gewinnt im Zusammenhang mit IoT jedoch besondere Bedeutung: das enorm große Datenvolumen, das mit zunehmender Anzahl an Teilnehmern weiter massiv anwachsen wird. Petabytes an Datenvolumen werden zur Normalität, wenn die online anfallenden Sensordaten von hunderttausend und mehr Geräten in Echtzeit erfasst werden. Diese müssen sinnvoll gespeichert, konsolidiert bzw. aggregiert, aber auch archiviert werden, um sie anderen Applikationen zur weiteren Verarbeitung in den verschiedenen Geschäftsprozessen zu übergeben. Skalierbarkeit kann bei IoT Applikationen sehr leicht den bisher bekannten Rahmen sprengen.

Um in wettbewerbsintensiven Marktsegmenten „die Nase vorn“ zu haben, wird der IoT-Bedarf in Unternehmen stetig wachsen. Dies zieht aber das Risiko nach sich, einen Kardinalfehler der IT zu wiederholen. Der Aufbau isolierter Insellösungen ist bei IoT bereits jetzt absehbar. Informationssilos, fehlende Standards, mangelnde Transparenz und Offenheit sind an der Tagesordnung. Jeder der IoT-Applikationen auf den Markt bringt, entwickelt eigene Konzepte, die Integration von Fremdkomponenten wird selten leicht gemacht.

Diesem Spannungsfeld wird durch Einsatz von Open Source Frameworks, die jedem zur Verfügung stehen, entgegen gewirkt. Dank der offenen Schnittstellen kann die Integration mit der bestehenden IT-Infrastruktur von Anfang sichergestellt werden.

Oracle Software Stack für IoT Infrastruktur nutzen

Auch der Oracle Software Stack lässt sich perfekt in dieses Bild einfügen. Auf Basis von Betriebssystem, Datenbank, Middleware und Java Entwicklungs- und Laufzeitumgebung lässt sich nicht nur eine IoT Plattform realisieren, sondern zeitgleich auch einige Silos im Unternehmen konsolidieren und integrieren, die sich sonst nur um spezielle Aufgaben, wie Social Enterprise, Dokumentenmanagement oder Archivierung gekümmert haben.

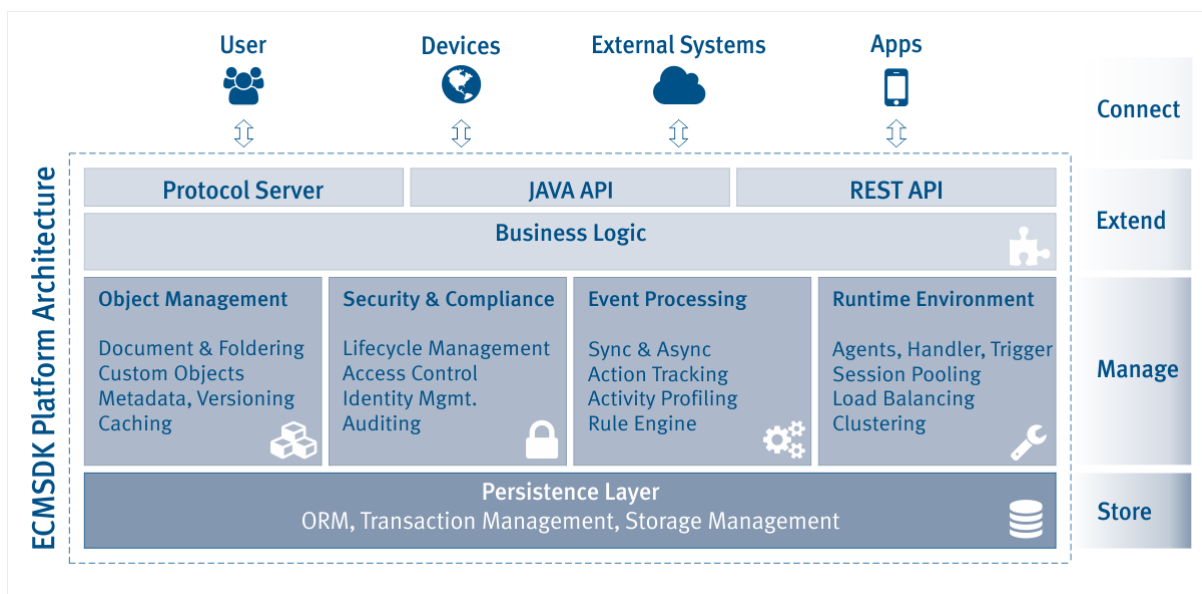


Abb. 1: ECMSDK Platform Architecture

Das ECMSDK ist eine lizenzkostenfreie Open Source API für unternehmenskritische, inhaltsgetriebene Datenbank-Anwendungen. Basierend auf der Oracle Datenbank, wurde die Entwicklung bereits 2000 von Oracle zunächst für den Bereich Enterprise Content Management vorangetrieben. Die vollständig in Java entwickelte, skalierbare Plattform bietet eine Laufzeit- und Entwicklungsumgebung, die bei zahlreichen Großunternehmen für Anwendungen in den verschiedensten Bereichen erfolgreich eingesetzt wird. 2014 wurde die Plattform an die Open Source Community übergeben und wird seitdem durch die inxire GmbH konsequent für die Anforderungen geschäftskritischer IoT- und Social-Lösungen ausgebaut. An Hand von Kundenbeispielen werden die wesentlichen IoT und Social Funktionen und Bestandteile dieser hochskalierbaren und flexiblen Plattform beleuchtet und erläutert.

Telematic, Smart Home und andere Daten

Die Hauptaufgabe der ECMSDK Infrastruktur ist das Verwalten von Objekten. Objekte können dabei die verschiedensten Formen haben. Seien es Gateways, Sensoren und Aktoren, Wikis, Blogs, Dokumente und Ordner, oder Zugriffberechtigungen, sogenannten Access Control Lists (ACL), Benutzerinformationen, Gruppen oder Klassen von Metadaten (Kategorien) – es können beliebige Objekte der realen Welt verwaltet werden. Als Verwaltung wird hierbei das Hinzufügen, Verändern, Löschen, Versionieren und Organisieren der Objekte bezeichnet.

Die Java API des ECMSDK kommuniziert mit dem Datenbank Repository über JDBC. Die einzelnen Objekte werden dabei als transiente Instanzen von Java Klassen innerhalb der API repräsentiert und anschließend persistent in der Datenbank abgespeichert. Der komplexe Vorgang des transaktionalen Lesens, Ändern und Speicherns der Objekte in der Datenbank wird dabei für den Entwickler komplett transparent von der API durchgeführt. Dazu gehören neben patentierten Caching Mechanismen auch Session- und Transaktions-Handling.

Es gibt Möglichkeiten die Objekte hierarchisch, flach, automatisiert oder undefiniert abzulegen. Als Beispiel kann im IoT Umfeld das Gateway als Ablageort für alle Geräte dienen, die mit diesem Gateway kommunizieren und ein Netzwerk bilden.

Die Benutzerdaten liegen eventuell schon in einem Oracle Internet Directory oder anderem LDAP vor und können problemlos von dort in das Repository übernommen werden. Ein Hintergrundprozess kann dabei die Aufgabe übernehmen, einen regelmäßigen Abgleich mit den Benutzern und Gruppen aus dem LDAP durchzuführen. Eine Authentifizierung der Benutzer findet üblicherweise ebenfalls direkt am vorhandenen LDAP statt (z.B. via Single-Sign-On). Eine 100%ige Mandantenfähigkeit durch das Repository ist gegeben, so dass mehrere voneinander abgegrenzte Benutzerkreise auf einem Repository realisiert werden können. Dies spart nicht nur Ressourcen, sondern ist für einen eventuellen Betrieb in der Cloud von großem Vorteil.

Neben den Standardrechten die ein Rechtekonzept mitbringt, gibt es bei ECMSDK auch die Möglichkeit eigene Rechte mittels der API zu definieren und somit das Rechtekonzept nach Belieben zu erweitern und den eigenen Anforderungen anzupassen.

Weiterhin gibt es die Möglichkeit, den Objekten zusätzliche Metadaten zu vergeben, welche nicht schon Teil der Objektklasse selbst sind. Diese Metadatenobjekte, auch Kategorie genannt, können dann verschiedenen anderen Objekten zugewiesen werden, ohne jedes Mal die Objektklasse selbst und damit die komplette Repräsentation im Repository anpassen zu müssen.

Aktionen und Aktivitäten

Jede Statusänderung einzelner IoT-Geräte sendet einen Event an das Gateway und die IoT-Plattform. Diese Statusänderungen, auch Aktionen genannt, werden über die Node ID des sendenden Gerätes eindeutig einer Objektinstanz im Repository zugeordnet und als Aktivität an der Instanz selbst abgelegt. Einzelne Events dienen in der Regel nur dem Anstoßen (Triggern) von weiteren Aktionen, werden aber selten dauerhaft gespeichert und für eine erweiterte Analyse aufbereitet. Der Sinn einer ganzheitlichen IoT-Plattform ist es hierbei alle Facetten, von der Speicherung bis hin zur Auswertung und eventuell Auslagerung oder Löschung der Informationen, abzudecken.

Weiterhin sollte nachvollziehbar sein, wer der Akteur war, der eine Aktion ausgeführt hat. Damit ist hierbei nicht die Person gemeint, die evtl. einen Lichtschalter betätigt hat, was mit gewisser Technik durchaus auch zu realisieren wäre. Vielmehr geht es darum, ob ein Licht durch einen Schalter

gesteuert wurde, oder durch das Öffnen einer Tür. Hat man eine Regel definiert, bei der jedes Mal wenn sich eine bestimmte Tür öffnet ein Licht einschaltet, wäre dies eine wichtige Information.

Da es sich nicht immer nur um einfache binäre Statusänderungen, wie im Falle eines Lichtschalters, handelt, sondern beispielsweise auch um Temperaturänderungen von Thermostaten oder Öffnungsgrad und Winkel eines Gaspedals handeln könnte, ist es möglich eigene Aktionstypen im System zu definieren. Die Definition geschieht über eine ActionSpecification (hier FollowAction).

```
m += createActionSpecification \  
Name = IFS.FollowAction \  
Description = "ActionSpecification for the basic Follow Action used in  
Social activity." \  
ActionFactoryClassName = oracle.ifs.core.activity.social.FollowAction \  
Tag = followSpec
```

Im ECMSDK werden die einzelnen Aktionen automatisch zu Aktivitäten aggregiert, über die man dann später einfach den aktuellen Status eines Gerätes feststellen kann und auch wie oft es seit der letzten Abfrage oder insgesamt den Status geändert hat. Die Aktivitätsdaten erben automatisch die Rechte des Objektes, dem sie entstammen. Weiter aufbereitet können Sie anschließend auch als eigenständige Objekte oder Dokumente im System mit anderen Rechten abgelegt werden.

Event Processing und Rule Engine

Das ECMSDK bietet verschiedene Möglichkeiten sich in das Event Processing einzuhängen. Es gibt zum einen die synchrone Abarbeitung von Aktionen und daraus resultierender Aufgaben die von einem Aktivitätsprozessor bearbeitet werden.

Im Aktivitätsprozessor gibt es auch die Möglichkeit, sich einer Rule Engine zu bedienen, die regelbasiert andere Geräte netzwerkübergreifend steuert. Kommuniziert ein Gassensor über ein anderes Protokoll mit einem anderen Gateway als ein Steuergerät, bietet sich auf dieser Ebene eine Integration der verschiedenen Systeme förmlich an. Die Daten aller Geräte liegen zentral und sicher im Repository und können für Analysen und Auswertungen direkt verwendet werden, ohne sich verschiedener Datentöpfe bedienen zu müssen.

Der Aktivitätsprozessor ist auch dafür da, eine Liste von Aktionen festzulegen, die an einem Objekt ausgeführt werden dürfen. Diese Funktionalität erweitert im Prinzip das Rechtekonzept auf Aktionsebene – welche Aktion wird an welchem Objekttyp erlaubt.

```
m += createActivitySpecification \  
Name = IFS.GeneralSocialActivity \  
Description = "ActivitySpecification which includes most generic Social  
Activity for any type of target object." \  
ActivityProcessorClassName =  
oracle.ifs.core.activity.social.SocialActivityProcessor \  
ActionSpecifications = "{IFS.LikeAction, IFS.FollowAction,  
IFS.CommentAction}" \  
Tag = generalSpec
```

Eine andere Möglichkeit sich in das Event Processing einzuhängen, ist die asynchrone Abarbeitung. Jede Aktion der Geräte erzeugt einen technischen Event innerhalb des ECMSDK, dem man eine Liste von Interessenten mitgeben kann. Ein spezieller Hintergrundprozess arbeitet die Queue der Events dann ab und kann so die mitgegebenen Interessenten über diese Aktion benachrichtigen, erneut steuern oder andere Aufgaben erledigen. Darüber wird standardmäßig auch die Folgen-Funktion im Social Kontext realisiert, bei der Folgende über Aktionen des Gefolgten informiert werden.

Der Unterschied zwischen der synchronen und asynchronen Abarbeitung besteht im Transaktionshandling. Während man sich beim synchronen Event Processing immer noch in der Transaktion der eigentlichen Aktion befindet und diese noch beeinflussen kann, ist die Aktion im asynchronen Event Processing bereits abgeschlossen und persistent gespeichert.

Informationen schnell und einfach finden

Das Sammeln und Speichern von Daten ist nur ein Aspekt einer IoT-Plattform. Bei der immensen Menge von Daten ist ein weiterer Schwerpunkt die Analyse und Auswertung dieser Daten für weitere Zwecke. Deshalb ist es wichtig, eine performante Suche über alle Arten von Inhalten zur Verfügung zu haben.

Werden alle Daten in einer Oracle Datenbank abgelegt, kann das ECMSDK seine eigene auf Oracle Text basierte Suche verwenden. Es ist jedoch durchaus möglich die Daten auch mit Crawlern anderer Search Frameworks (wie Solr, Elasticsearch, Compass, HSearch, etc.) zu indizieren. Werden die Daten z.B. in Hadoop abgelegt, wird sicherlich eine andere Methode gewählt werden.

Entwickler haben sehr flexible Möglichkeiten Suchanfragen an das Backend zu formulieren und die Ergebnisse zu verarbeiten. Dies geht hin bis zur Manipulation des abgeschickten SQL Statements. Wichtig hierbei zu erwähnen ist, dass die meisten Open Source Search Frameworks bei Ihrer Suche nicht die Möglichkeit haben, die Zugriffsberechtigungen und Ablagestrukturen bei der Suchanfrage mit zu berücksichtigen. Oft wird erst eine Suche abgeschickt und das Ergebnis dann anschließend auf Grundlage des Rechtekonzepts gefiltert. Dies ist oft ein Manko bei den auf Apache Lucene basierten Search APIs. Bereits bei einem wenig komplexen Rechtekonzept mit ACLs und ACEs, bei dem Benutzer in hierarchischen Gruppen organisiert sind, wird der Abgleich des Suchergebnisses mit dem Rechtekonzept aus einem LDAP sehr mühselig, langsam und wenig skalierbar. Es ist also von Vorteil, wenn das Rechtekonzept und die Daten im selben Repository liegen und man beide Welten bereits in der Suchabfrage selbst miteinander verbinden kann. Somit braucht sich ein Entwickler keine Gedanken zu machen, falls ein Benutzer in eine andere Gruppe wechselt und dadurch eventuell das Zugriffsrecht an einem Objekt verliert. Solche häufigen Vorgänge behandelt die Suchmaschine komplett transparent für den Entwickler, ohne nachgelagerte notwendige Filterung eines vorläufigen Ergebnisses.

Erweitert man ein Objekt im Repository um ein Attribut, so kann man direkt danach auch bereits nach diesem Suchen, ohne weitere Anpassungen an der Suche und deren Crawler vornehmen zu müssen. So hat man immer schnell und aktuell Zugriff auf die Daten, die der Kunde im jeweiligen Kontext benötigt.

Zusammenfassung

Auf Basis des Oracle Software Stacks lassen sich mit Java sehr schnell und effektiv, hoch skalierbare und performante IoT- oder Social-Plattformen realisieren. Dabei spielt die von Oracle initiierte und ausgereiften Open Source API eine zentrale Rolle. Hier werden alle Aspekte von Persistence und ORM (Object/Relational Mapping), über Caching, Transaction Management, Auditing, Compliance, Identity und Access Management, bis hin zum Enterprise Content Management zur Verwaltung aller Unternehmensinhalte unter einer Infrastruktur vereint. Man erspart sich dadurch die aufwendige Integration vieler anderer, einzelner Frameworks, schafft Lösungssilos im Unternehmen ab, und hat flexible Möglichkeiten, die Plattform on-premise oder in der Cloud zu betreiben.

Kontaktadresse:

Frank Closheim
inxire GmbH
Hanauer Landstraße 293B
D-60314 Frankfurt

Telefon: +49 (0) 69-87 0000 19
Fax: +49 (0) 69-87 0000 11
E-Mail frank.closheim@inxire.de
Internet: www.inxire.com, www.ecmsdk.com