



Der Oracle Mobile Cloud Service

Dr. Jürgen Menge, ORACLE Deutschland B.V. & Co. KG

Die Entwicklung von Applikationen für mobile Endgeräte rückt zunehmend in den Mittelpunkt der Anwendungs-Entwicklung („Mobile First“). Viele der umgesetzten mobilen Anwendungsfälle benötigen Services (Daten und Funktionen) aus Systemen, die im Backend des Unternehmens laufen.

Mobile Endgeräte sollten aus folgenden Gründen nicht direkt mit diesen Backend-Services kommunizieren:

- Da die mobilen Endgeräte zumeist außerhalb des Firmennetzes laufen, sind Maßnahmen erforderlich, um die Backend-Systeme gegen Attacken (DoS,

XML Bombs etc.) beziehungsweise gegen eine Überlastung zu schützen.

- Der mobile Anwendungsfall erfordert oft eine andere (meistens einfachere) Datenstruktur, als sie die vorhandenen Services im Backend bereitstellen. Man könnte dafür zusätzliche Services entwickeln, was jedoch mit steigender

Zahl der Apps und der Geräteklassen (mobile Geräte, Wearables etc.) schnell zu einer unübersichtlichen und nicht mehr wartbaren Vielfalt von Schnittstellen (APIs) der Backend-Systeme führen würde.

- Mobile Apps kommunizieren idealerweise mit REST-Services, die die Nach-

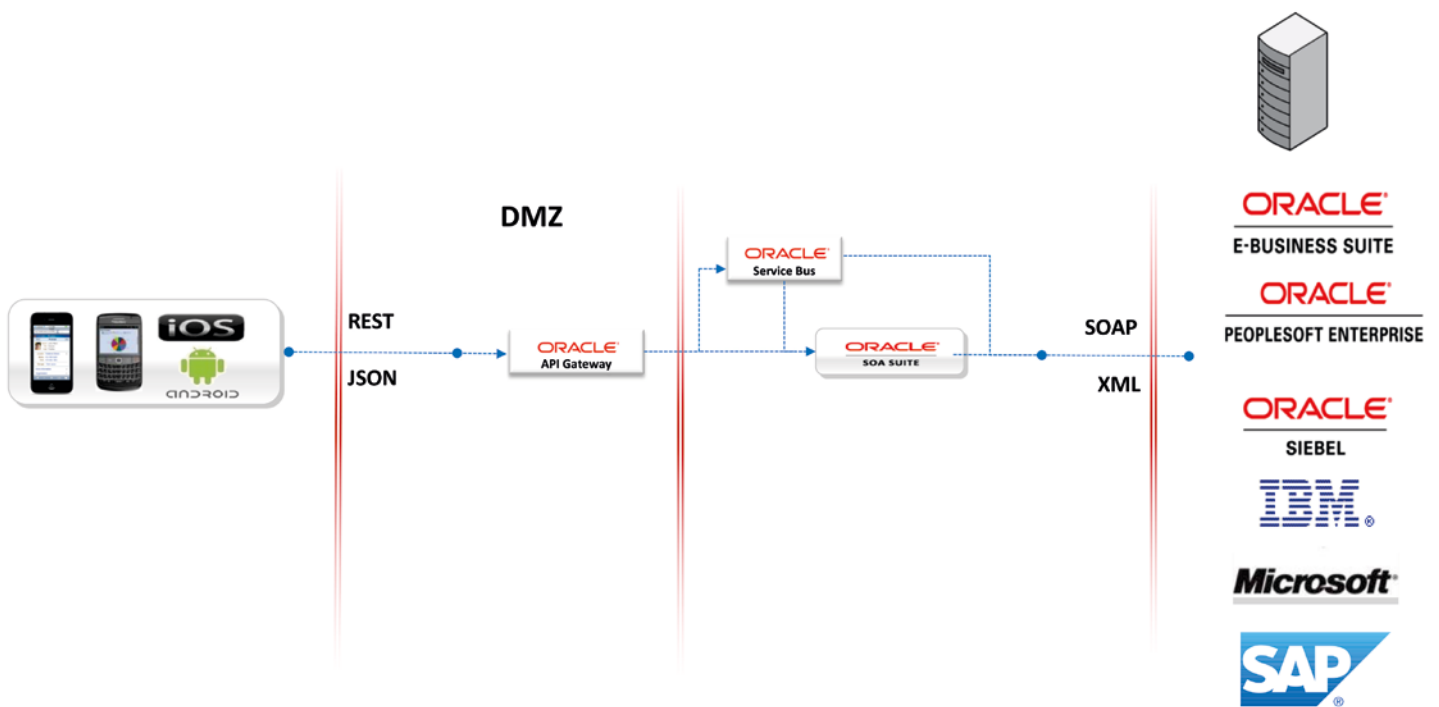


Abbildung 1: Oracle Mobile Platform – Integration des Backend

richten (Payload) im JSON-Format übertragen. Viele Backend-Services bieten aber entweder keine oder nur Schnittstellen für SOAP-Services an. Die Bereitstellung zusätzlicher REST-Services führt jedoch zum gleichen Problem wie im vorhergehenden Punkt.

Ein Lösungsansatz würde darin bestehen, einen Service Bus („Oracle Service Bus“) und beziehungsweise oder ein API-Gateway („Oracle API Gateway“) zwischen der mobilen Welt und dem Backend aufzustellen. Diese würden die Transformation der Nachrichten-Formate und -Protokolle, das Routing und im Falle des Gateways auch bestimmte Sicherheitsfunktionen übernehmen (siehe Abbildung 1).

Für die Unternehmen bedeutet das, dass sie diese Middleware-Komponenten lizenzieren, installieren, konfigurieren und administrieren sowie die dafür erforderliche Hardware bereitstellen müssen. Alternativ können die gewünschten Funktionen als Mobile Backend as a Service (MBaaS) aus der Public Cloud bezogen werden. Die Nutzung eines Cloud Service bietet zusätzlich den Vorteil der Elastizität, also mit einer schwer zu prognostizierenden Zahl installierter Apps Schritt halten zu können.

Oracle stellt seit diesem Jahr mit dem Oracle Mobile Cloud Service (OMCS) einen

solchen Service bereit, der mobile Endgeräte mit Services verbindet, die im Backend des Unternehmens laufen oder öffentlich verfügbar sind (etwa Google Maps).

Die Komponenten des Mobile Cloud Service

Der Oracle Mobile Cloud Service (OMCS) ist modular aufgebaut. Jegliche Kommunikation zwischen den Komponenten des OMCS und der Außenwelt mit Ausnahme

der Adapter zu den Backend-Systemen erfolgt über RESTful Web Services (siehe Tabelle 1).

Zusätzlich werden für die mobilen Plattformen iOS, Android sowie für die Entwicklungs-Frameworks Oracle Mobile Application Framework (MAF) und Xamarin SDKs angeboten, die es dem Entwickler der mobilen App ermöglichen, die Funktionen des OMCS in der mobilen App komfortabel zu benutzen. Anstelle einer langwierigen Beschreibung jeder Komponente soll im folgenden Abschnitt der Pro-

Mobile Backend (MBE)	
Platform APIs	User Management Service (UMS)
	Notifications API
	Data Offline (Sync) API
	Storage API
	Database API
	Analytics API
Custom APIs	
Connectors	SOAP Connector
	REST Connector

Tabelle 1: Komponenten des Oracle Mobile Cloud Service (OMCS)

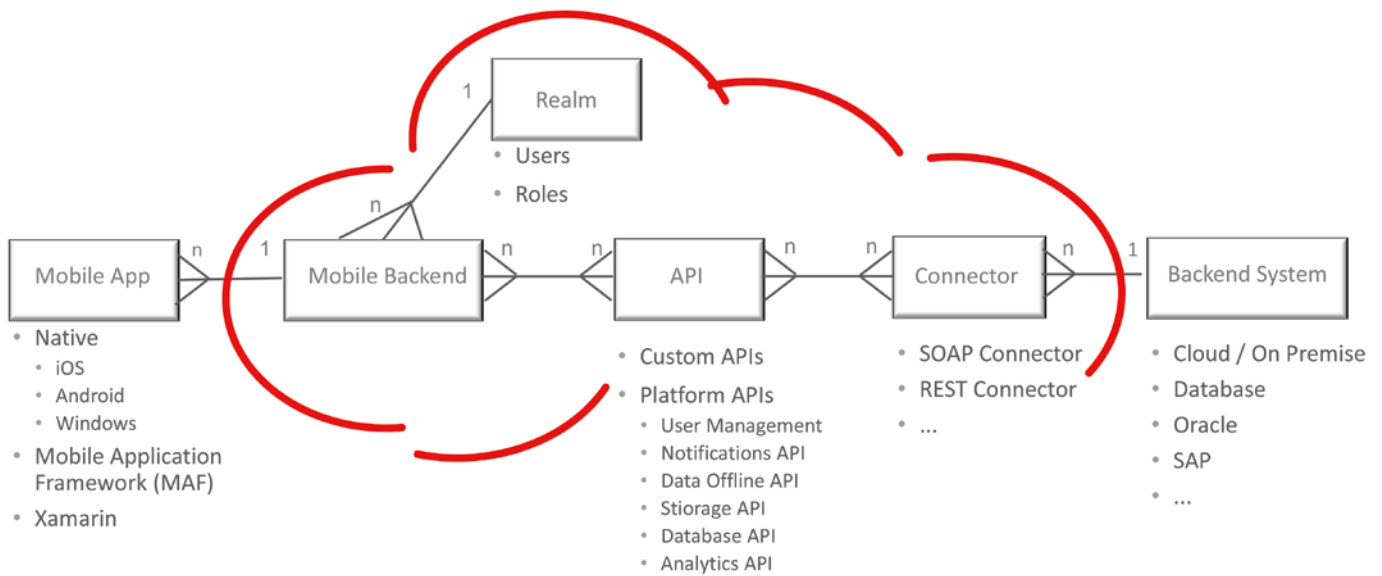


Abbildung 2: Oracle Mobile Cloud Service – Architektur

zess der Kommunikation zwischen einer mobilen App und einem Backend-System beschrieben werden. Gleichzeitig werden die verschiedenen Rollen vorgestellt, die Personen in diesem Prozess wahrnehmen (siehe Abbildung 2).

Die mobile App und der Entwickler

Der Entwickler ist für die Entwicklung und den Test der mobilen App verantwortlich. Die App kann nativ für die jeweilige Plattform (iOS, Android, Windows) oder Plattform-übergreifend mithilfe von Frameworks wie Oracle MAF oder Xamarin entwickelt werden. Der OMCS stellt dafür spezielle Software Development Kits (SDK) zum Download bereit. Der Entwickler nutzt mithilfe des SDK bestimmte Funktionen des OMCS in der App. So kann er beispielsweise Events bei der Nutzung der App registrieren, die im OMCS erfasst, ausgewertet und visualisiert werden (OMCS Analytics).

Die Kommunikation zwischen der mobilen App und dem OMCS verläuft immer über ein Mobile Backend (MBE), das vom Entwickler für jede mobile App im OMCS anzulegen ist. Im MBE werden wichtige Festlegungen in Bezug auf die Sicherheit (Authentifizierung, spezielle Keys) getroffen. Durch die Zuordnung von Realms mit Benutzern und Gruppen zum MBE wird der Personenkreis festgelegt, der die App nutzen darf.

Durch die Zuordnung von APIs zum MBE werden der mobilen App bestimmte Services als Schnittstellen zur Verfügung gestellt. Services, die häufig benötigte Standard-Funktionalität bereitstellen, sind im OMCS als Platform-APIs enthalten und können deklarativ mit dem MBE verknüpft sein. So bietet beispielsweise das Storage-API die Möglichkeit, Bilder, Dokumente oder Präferenzen der Benutzer im OMCS zu speichern.

Anforderungen, die von den Platform-APIs nicht abgedeckt werden, lassen sich durch Custom-APIs im OMCS umsetzen. Der Entwickler der App beschreibt zunächst, welche Daten aus dem Backend von der App benötigt werden. Diese Beschreibung legt die Schnittstelle des API, also die Endpunkte, Parameter und Methoden des REST Service, fest. Er kann dabei zwischen einer deklarativen Arbeitsweise im GUI oder einer textuellen Beschreibung auf Basis der RESTful API Modeling Language (RAML) umschalten. Sofern der Entwickler dieser Beschreibung Beispieldaten (Mockups) hinzufügt, kann er mit der Entwicklung und dem Test der App fortfahren, ohne auf die Implementierung des API durch den Service-Entwickler warten zu müssen.

Die Implementierung

Der Service-Entwickler ist für die Implementierung des Custom-API, also der vom

App-Entwickler festgelegten Schnittstellen des API, verantwortlich. Dazu lädt er sich zunächst ein Grundgerüst des Codes (Scaffold) aus dem OMCS auf seinen Desktop. Die Implementierung erfolgt mithilfe von node.js, einem populären Framework für Server-seitiges JavaScript. Der Entwickler kann dazu einen beliebigen Text-Editor (wie Atom) oder IDEs verwenden, die die Bearbeitung von JavaScript unterstützen (wie NetBeans).

Node.js bietet dem Service-Entwickler die Möglichkeit, im Code nicht-blockierende, asynchrone Calls zu verwenden, wie sie von mobilen Apps gebraucht werden, die Daten aus vielen verschiedenen Systemen bearbeiten und darstellen sollen. Ein weiterer Vorteil von node.js besteht in der freien Verfügbarkeit zahlreicher fertiger Module. Der Service-Entwickler kann im Code der Implementierung andere Custom- oder Platform-APIs (wie das Database API) sowie Konnektoren aufrufen.

Konnektoren stellen die Verbindung zu Systemen im Backend her. Gegenwärtig werden mit dem OMCS vordefinierte Konnektoren für SOAP- und REST-Services zur Verfügung gestellt. Weitere sollen in künftigen Releases folgen. Der Service-Entwickler konfiguriert diese Konnektoren, um auf Services im Backend zuzugreifen. Erfolgreich getestet, kann er die Konnektoren im Code der Implementierung per node.js aufrufen. Hat der Service-Entwickler die Implementierung in node.js fer-

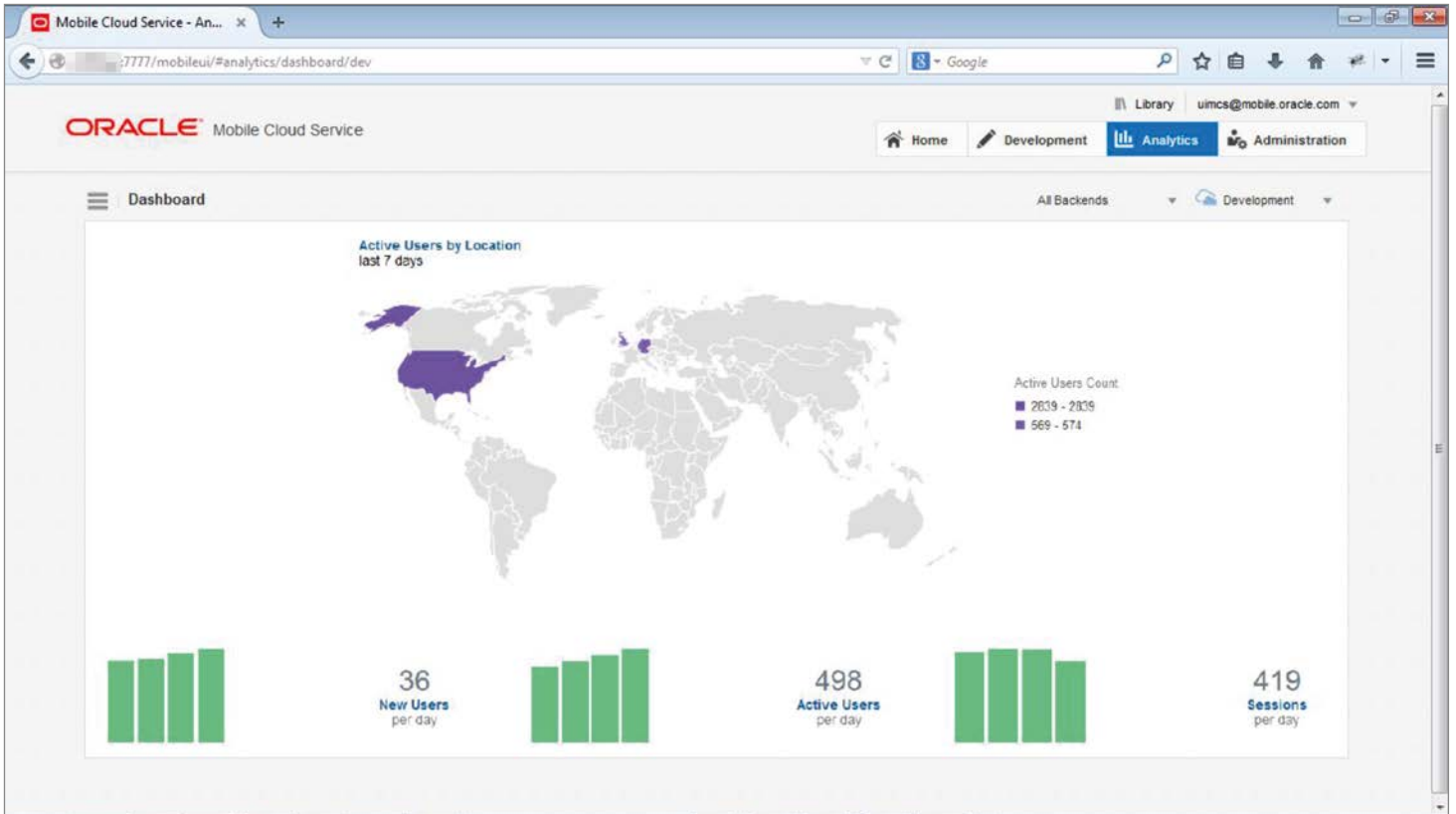


Abbildung 3: OMCS Analytics – Dashboard

tiggestellt, wird der Code zurück in den OMCS geladen.

Analytics und der Program Manager

Ist eine Applikation nach Entwicklung und Test in Betrieb gegangen, stellen sich Fragen nach dem Erfolg der App beziehungsweise auftretenden Problemen. Dies ist vor allem für den Program Manager wichtig, der mit der Einführung der App beauftragt wurde und an deren Erfolg gemessen wird. Er kann OMCS Analytics nutzen, um sich anhand von Statistiken über die Akzeptanz der Applikation und aufgetretene Probleme zu informieren. Ein Dashboard und verschiedene Berichte visualisieren Kennzahlen über folgende Funktionen (siehe Abbildung 3):

- API Calls
- Sessions
- User und deren geografische Verteilung
- Antwortzeiten
- Fehlermeldungen

Zusätzlich kann der Entwickler zu bestimmten Zeitpunkten Custom Events (etwa Bestellung eines Artikels) aus der App senden. Diese Events werden im OMCS gesammelt und stehen für eine Auswertung in Form von Berichten und Diagrammen zur Verfügung.

Der Architekt legt fest, aus welchen Backend-Systemen welche Daten für die mobile App bereitgestellt werden. Er kümmert sich außerdem um übergreifende Themen wie Sicherheit und Skalierbarkeit der Services. OMCS unterstützt den Architekten durch eine Versionierung und einen Lebenszyklus der Artefakte (Test, Freigabe, Produktion).

Fazit

Der Oracle Mobile Cloud Services (OMCS) ist ein Frontend Cloud Service, der über REST Services mit mobilen Endgeräten kommuniziert. Über die im OMCS definierten APIs wird die Verbindung zu Services im Backend hergestellt. Dabei kann es sich sowohl um Cloud Services (Database Service, Java Cloud Service, Integration Cloud Service) als auch um instal-

lierte On-Premise-Systeme im Backend handeln.

Der Oracle Mobile Cloud Service wird über die URL „<http://cloud.oracle.com/mobile>“ aufgerufen. Um diesen näher kennenzulernen, bietet Oracle einen Testzugang („Try it“) und Workshops an.



Dr. Jürgen Menge
juergen.menge@oracle.com