Mobiles Einkaufen & Bezahlen mit Smartphones und Oracle CRM on Demand

Thorwald Schubert
ec4u expert consulting ag
Karlsruhe

Schlüsselworte:

Oracle Fusion Middleware, Titanium, HTML5, PayPal, Google Wallet, Google Checkout, Oracle CRM on Demand

Einleitung

Anfang 2011 haben sich bereits allein in Deutschland ca. 11 Millionen Smartphones im Umlauf befunden. Diese Smartpones werden natürlich nicht nur zum Telefonieren, als mobiles Office oder für die klassische Unterhaltung genutzt, sondern auch für den elektronischen Einkauf von physikalischen und digitalen Gütern. Bereits in 2009 betrug das weltweite Brutto-Handelsvolumen, das über Smartphones erzielt wurde, ca. 1,6 Milliarden US$. Für 2014 wird das weltweite Handelsvolumen auf 119 Milliarden US$ geschätzt.

Ein derart großes Handelsvolumen setzt natürlich eine entsprechend große Aktivität seitens der Einkäufer voraus. Die in diesem Rahmen anfallenden Daten über die ausgewählten Produkte, Kundendaten, Warenkorbinformationen und Transaktionsdaten sind für CRM-Systeme von großem Interesse, da basierend hierauf präzisere Analysen gefahren werden können, die Kundenbedürfnisse besser verstanden werden und die eigenen Produkte und Services hierdurch besser auf die lukrativen Kundengruppen abgestimmt werden können.

Die Integration dieser Mobilen Einkaufsdaten in ein CRM-System erfordert neben der Kenntnis mobiler Bezahltechnologien, wie beispielsweise Google Wallet, Google Checkout oder PayPal auch die Kenntnis möglichst plattformunabhängiger mobiler App-Technologien und wie diese über eine Middleware mit einem CRM-System integriert werden können.

Use Case und Prototyp

Der Vortrag beinhaltet auch die Vorführung eines Prototyps, der folgenden hypothetischen Anwendungsfall abbildet:

- Ein Hersteller für Unterhaltungstechnik bietet eine mobile App für Einkauf, Information und Support zu seinen Produkten an
- Außerhalb der Geschäftszeiten kann über diese App ein „on Demand Premium-Support“ direkt via PayPal eingekauft werden, die Abrechnung soll direkt über die Demo-Umgebung von PayPal einsehbar sein
- Zur genaueren Klassifikation des Support-Falls muss die Produktnummer und eine Fehlerbeschreibung bereitgestellt werden
- Produktnummer und Fehlerbeschreibung sollen zusammen mit der Bezahlinformation im CRM-System nachgeführt werden

Die Nachfolgende Abbildung beschreibt die Interaktionsschritte der nativen Titanium-Implementierung:

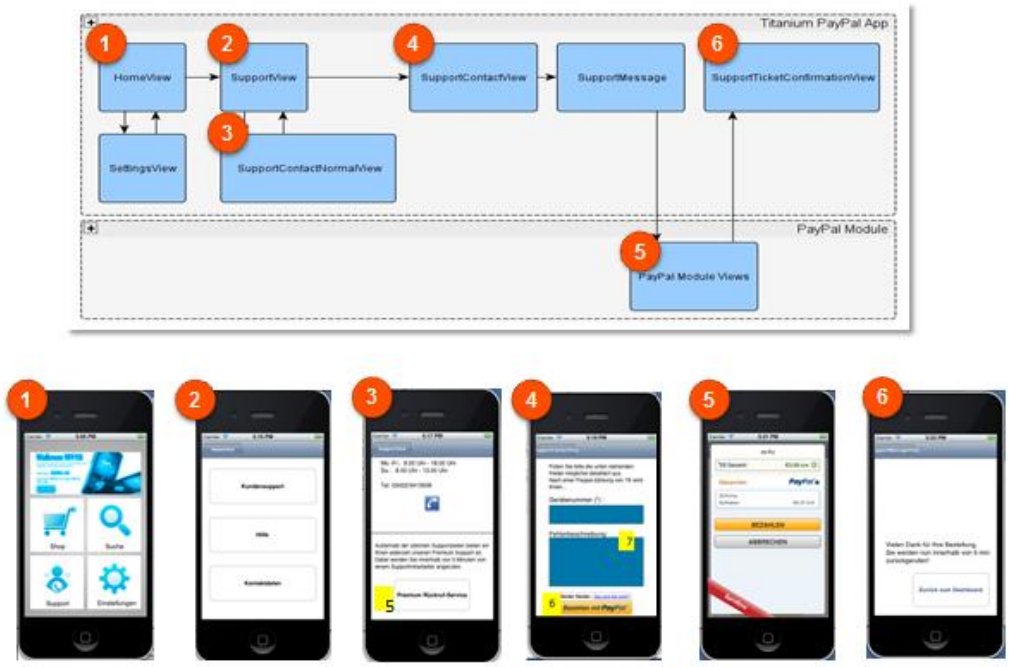

Abb. 1: Screenflow der „on Demand premium-Support“-App

Der Prototyp selbst basierend auf folgenden technischen Komponenten/Systemen:

- Native und plattformunabhängige implementierte App basierend auf Titanium
- Integration der PayPal-Bezahltechnologie zwischen mobiler App und Middleware
- Nutzung der Fusion Middleware als Integrationsschicht
- Berücksichtigung eines DMZ-Konzepts
- Integration der Cloud-CRM-Lösung Oracle CRM on Demand

System-Architektur

Während die mobile App entweder als native Implementierung client-seitig läuft oder als web-basierte App über einen Web Server zur Verfügung gestellt wird, befindet sich das CRM-System normalerweise im eigenen Intranet. Das gleiche Sicherheitsbedürfnis besteht für die Middleware, die ebenfalls im Intranet platziert ist. Die Bezahl-Technologien wie PayPal oder Google hingegen werden über das Internet zur Verfügung gestellt. Daraus ergibt sich der Bedarf für eine DMZ-Zone als Sicherheits-Gate.

Die nachfolgende Abbildung beschreibt die System-Architektur des Prototyps:

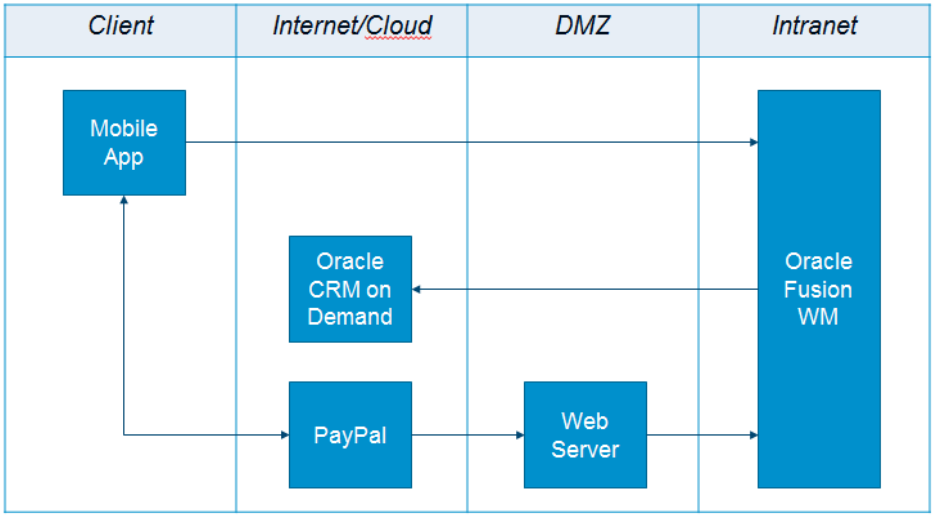

Abb. 2: System-Architektur des Prototyps

Die native auf Titanium basierende App-Implementierung läuft lokal auf der client-Seite (Smartphone). Der PayPal-Service, ebenso wie Oracle CRM on Demand werden über die Cloud zur Verfügung gestellt. Der PayPal-Service kommuniziert mit der Oracle Fusion Middleware ausschließlich über einen separaten Web Server in der DMZ, wohingegen die mobile App direkt mit der Fusion Middleware im Intranet kommuniziert.

Service-Architektur

Die Service-Architektur des Prototyps wird nachfolgend dargestellt und ist derart angelegt, dass zunächst auf client-Seite vor dem eigentlichen Auslösen des Bezahlvorgangs mit PayPal via SOAP-Request der „Create OCoD SR"-Service der SOA-Suite der Fusion Middleware aufgerufen wird, der wiederum einen weiteren SOAP-Request gegenüber Oracle CRM on Demand aufruft, um hier initial einen Service Request für den Nutzer der App zu erstellen.

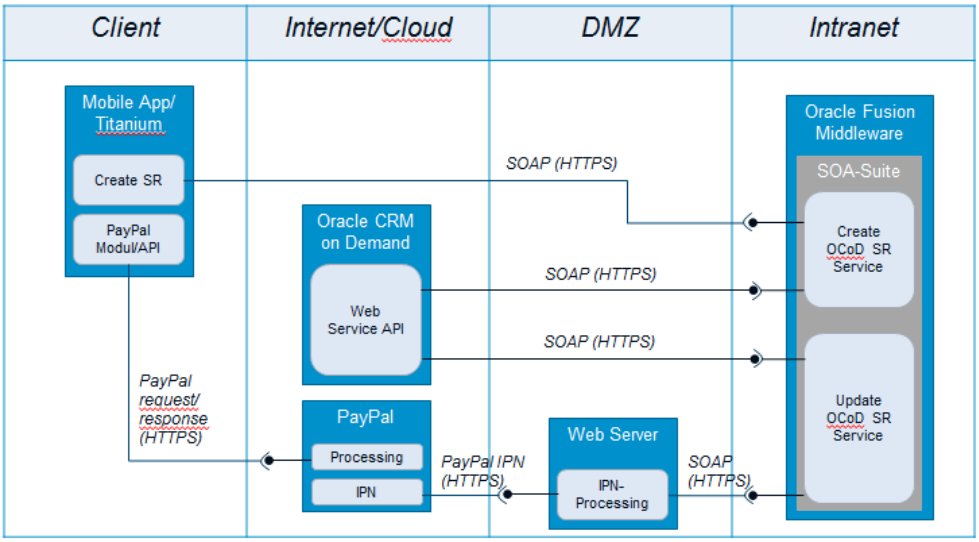

Abb. 3: Service-Architektur des Prototyps

Erst nach Auslösung des Requests zur SR-Erstellung wird über die PayPal API der eigentliche Bezahlvorgang mit PayPal angestoßen, die Kommunikation an dieser Stelle ist eine Black Box, in die nicht weiter Einsicht genommen werden kann. Wurde der Bezahlvorgang schließlich auf der Seite von PayPal abgeschlossen, erfolgt eine „Instant Payment Notification"-Message an den Web Server in der DMZ, der die Bezahlinformation an den „Update ODoD SR"-Service der SOA-Suite weitergibt. Dieser Service aktualisiert wiederum die Bezahlinformation auf dem zuvor in Oracle CRM on Demand angelegten Service Request.

Mobile App basierend auf Titanium

Der Prototyp der mobilen Applikation wurde auf einer plattformunabhängigen Technologie, Titanium, entwickelt. Im Fall von Titanium handelt es sich um ein Java-Script basiertes Framework, das den Zugriff auf native Funktionen der mobilen Endgeräte erlaubt.

Die implementierte Code-Architektur wird im folgenden Schaubild dargestellt:

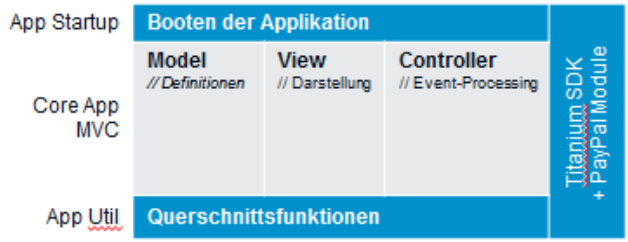

Abb. 4: Code-Architektur der Titanium-Implementierung des Prototyps

Wird die Applikation gestartet, werden zunächst Setup-Funktionalitäten ausgeführt („App Startup" in Abb. 4). Das Kernelement bildet das Model View Control-Pattern (MVC-Pattern), mit dessen Hilfe die wesentliche Applikationslogik realisiert wurde. Zusätzlich gibt es noch einige Querschnitts-funktionalitäten, die in allen Applikationsteilen genutzt werden („App Util" in Abb. 4). Zur Integration der Bezahltechnologie PayPal wird direkt auf die PayPal-API zugegriffen.

PayPal-Integration

Die Einbindung von PayPal wird in der folgenden Abbildung beschrieben:

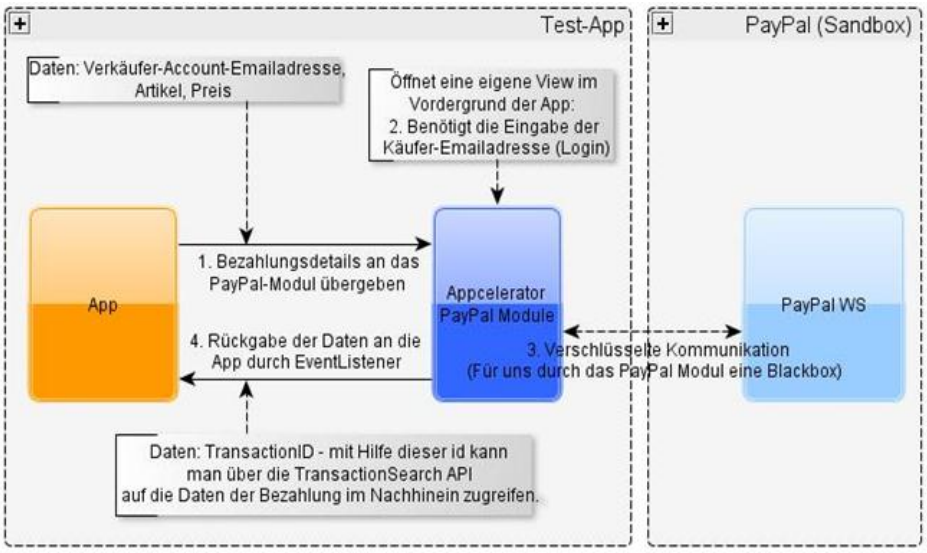

Abb. 5: PayPal-Integration

Die mobile App sendet über die Einbindung der PayPal-API einen Request an den PayPal-Server. Dieser wiederum liefert eine Referenz auf die Login-Seite von PayPal zurück, die direkt in die mobile App eingebunden wird. Nach Eingabe der Zugangsdaten auf dieser PayPal Login-Seite findet direkt eine verschlüsselte Kommunikation mit den PayPal-Servern statt, um den Bezahlvorgang abzuwickeln. Ist der Bezahlvorgang erfolgreich bei PayPal platziert, wird eine Benachrichtigung inklusive der zugehörigen PayPal-Transaktionsnummer an die mobile App zurückgegeben und die Ablaufsteuerung liegt wieder bei der mobilen App. Der Bezahlvorgang erfolgt von diesem Zeitpunkt an asynchron. Wird die Zahlung finalisiert, erfolgt durch PayPal eine Instant Payment Notification (IPN) als Quittung für den erfolgreichen Abschluss.

Oracle Fusion Middleware und DMZ Web Server

Die Kommunikation der mobilen App, wie auch des PayPal-Servers mit Oracle CRM on Demand erfolgt über die Oracle Fusion Middleware zusammen mit einem Web Server, der in der DMZ platziert ist.

Während die mobile App direkt auf einen SOAP Service der Fusion Middleware, bzw. der SOA-Suite zugreift, verläuft die IPN-Message von PayPal aus über den Web Server in der DMZ (siehe Abb. 2). Einerseits liegt dieser Architektur der Sicherheitsgedanke zugrunde, dass wir keinen direkten Zugriff eines SOAP-Service direkt auf ein System im eigenen Intranet zulassen wollen, wenn wir diesen nicht kontrollieren können (im Fall der mobilen App haben wir den SOAP-Service Aufruf selbst implementiert und bei OCoD handelt es sich um einen Aufruf, der von der Middleware aus getriggert wird). Zu diesem Zweck wird der Web Server in der DMZ als weiterer Intermediär eingebunden. Andererseits bietet der Web Server über Caching-Konzepte die Möglichkeit, die Last auf der Middleware zu reduzieren.

Die Services der Fusion Middleware zum Erstellen und Aktualisieren der Service Requests in Oracle CRM on Demand (OCoD) werden hierbei als SOA-Services abgebildet, deren Architektur im folgenden Schaubild beschrieben wird:

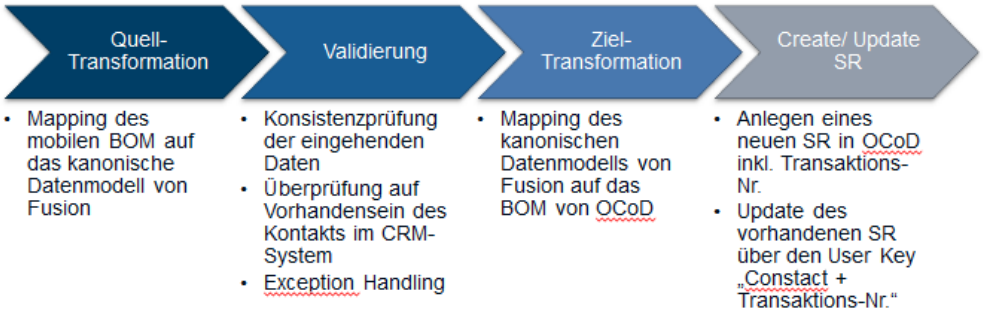

Abb. 6: Architektur der SOA-Services des Prototyps

Grundsätzlich wird zuerst das Datenmodell des Quell-Systems auf das kanonische Datenmodell der Middleware abgebildet. Anschließend werden alle bereitgestellten Daten validiert und ein Exception Handling durchgeführt, falls bestimmte Validierungen fehlschlagen. Sind alle Validierungen erfolgreich, werden die Daten aus dem kanonischen Datenmodell in das Business Object Model (BOM) des Ziel-Systems, in diesem Fall OCoD, überführt. Ist dies erst geschehen, findet der eigentliche SOAP Web Service Aufruf für das Anlegen oder Aktualisieren eines Service Requests in OCoD statt. Die OCoD-Services können direkt über eine eigene API angesprochen werden.

Kontaktadresse:

Thorwald Schubert
ec4u expert consulting ag
Zur Giesserei 19-27B
D-76227 Karlsruhe

Telefon: +49 (0) 174 1880 457
Fax: +49 (0) 721 46 476-299
E-Mail thorwald.schubert@ec4u.de
Internet: www.ec4u.de
www.fusion.de