

RIA in der Praxis - Webanwendungen mit Oracle, Spring und Flex

Frank Szilinski
esentri software GmbH
Karlsruhe

Schlüsselworte:

RIA, Adobe Flex, Java, JEE, JSF, Spring, Open Source, Rich Internet Application

Einleitung

Durch das Look & Feel von Desktop Anwendungen werden Rich Internet Applications (RIA) immer beliebter. Sowohl für die Benutzeroberflächen als auch für das Backend stehen inzwischen fast unzählige Frameworks zur Verfügung. Im Oracle Technologie-Stack ist JavaFX für die zukünftigen Frontends vorgesehen und ADF bzw. JEE für die Geschäftslogik. Dieser Vortrag beschäftigt sich mit einer Alternative zu JavaFX & ADF und zeigt, wie sich mit der Oracle Datenbank, dem Open Source Framework Spring sowie Adobe Flex anspruchsvolle Webanwendungen entwickeln lassen.

Dabei wird bei der Vorstellung auch auf die Vor- und Nachteile gegenüber dem Einsatz reiner Oracle Technologien eingegangen, sowohl aus Sicht des Anwenders als auch aus Sicht des Entwicklers.

Der Bedarf - verbesserte Usability durch neue Webtechnologien

Seit dem Boom des Internets gibt es zwei gegensätzliche Architekturansätze für die Realisierung von Anwendungen. Etablierte aber alternde Fat-Client Applikationen konkurrieren mit leichtgewichtigen web-basierten Client-Server Applikationen. Diese können mit einigen Vorteilen punkten, wie z.B. einfacher zentraler Wartung, Zugriff von beliebigen Clients und Betriebssystemunabhängigkeit durch die Verwendung von Browsern. Leider mangelt es den Webanwendungen allerdings nach wie vor an einer komfortablen Bedienung. Bei den Endanwendern kam dadurch nie richtig das Gefühl auf, mit einer richtigen Anwendung zu arbeiten - im Gegenteil: fehlende Features, die man von den „alten“ Desktop Anwendungen gewohnt war, sind bis heute im Web nur schwer oder über Umwege (JavaScript und AJAX als vermeintliche Heilsbringer) zu realisieren.

Bis heute gibt es daher keinen eindeutigen Sieger und beide Ansätze werden nach wie vor je nach Anwendungsfall benötigt. Seit einiger Zeit gibt es eine vielversprechende Alternative, eine neue hybride Architektur mit dem Namen Rich Internet Application (RIA), die aus beiden Ansätzen hervorgeht und versucht die Vorteile von beiden Seiten zu vereinen. Große Firmen wie Oracle (JavaFX), Adobe (Flex) und Microsoft (Silverlight) investieren viel in diese neue Technologie, verspricht sie doch erstmals echte Vorteile für die Endanwender, da web-basierte Anwendungen den Bedienkomfort von Desktop Applikationen bieten können. Oracle hat bisher versucht mehr „Richness“ über AJAX Features mit dem hauseigenen Framework ADF zu integrieren auf dem u.a. auch die neuen Fusion Applications basieren. Mit JavaFX hat Oracle seit der Sun Übernahme jetzt aber auch ein eigenes echtes RIA Framework im Technologiestack, das jedoch bisher nur bedingt praxistauglich ist.

Seit vielen Jahren etabliert dagegen ist die RIA Technologie von Adobe. Mit Flex besteht bereits seit 2004 die Möglichkeit komplexe und anspruchsvolle Webapplikationen zu entwickeln. Als Laufzeitumgebung dient dabei der weit verbreitete Flash-Player.

Rich Internet Applications im Detail

Wie immer bei neuen Technologien verspricht man sich im Idealfall auch von RIA unterschiedlichste Dinge. Allen voran werden folgende Features und Vorteile mit RIA in Verbindung gebracht.

- Aufwendige und hübsche Benutzeroberflächen ohne Nachladen
- Rich-Client Funktionen auch im Web, z.B. Drag & Drop
- Browser- und Betriebssystemunabhängigkeit
- Einheitliches Look & Feel (Skinning)
- Einfache Möglichkeit eigene UI-Komponenten einzubinden
- Optionale Offline Funktionalität
- Logik auch im Client, Zugriff auf lokale Ressourcen (z.B. Dateisystem)
- Gleiche Codebasis für Web und Desktop
- Aufwendige Installationen entfallen
- Echtzeit-Features, wie z.B. die Synchronisation von Daten über mehrere Clients hinweg (Server Push, Publish & Subscribe)

Den Endanwendern wird bei diesen Anwendungen das Gefühl geben, sie würden mit einer normalen Fat-Client-Anwendung arbeiten. RIAs gehen dabei jedoch weit über die Möglichkeiten von AJAX und HTML5 hinaus und benötigen für die komplexen Oberflächen zusätzliche Browser-Plugins oder andere Laufzeitumgebungen.

Adobe Flex im Frontend

Der erste Blick bei der Auswahl einer neuen Frontend-Technologie fällt typischerweise auf die vorhandenen UI-Komponenten. Flex bietet den Entwicklern hier umfangreiche und praxiserprobte Standard-Komponentenbibliotheken, die durch kommerzielle Bibliotheken oder Eigenentwicklungen sogar noch ergänzt werden können. Dabei ist es auch besonders einfach eigene Themes bzw. Skins für die Oberflächengestaltung der Benutzeroberflächen zu definieren. Durch den Adobe Flex Builder, der auf der Eclipse Plattform basiert, wird die Oberflächenentwicklung der MXML und ActionScript3 Komponenten hervorragend unterstützt.

Die größten Vorteile von RIAs liegen aber nicht allein in den neuen Komponentenbibliotheken, sondern vor allem in der Echtzeit-Interaktion mit dem Endanwender, ohne dass Server-Roundtrips und nerviges Neuladen der kompletten Seite für Unterbrechungen sorgen. Durch Offline-Features, bei Flex durch die Adobe Air Technologie, wird es zusätzlich sogar möglich, dass Anwendungen beim Wegbrechen der Internet-Verbindung trotzdem funktionieren. Lokale Änderungen werden dann gecacht und nach Wiederaufbau der Verbindung mit dem Server synchronisiert. Im Hintergrund wird dafür ein Offline Integration Framework verwendet, das beispielsweise eine lokale SQLite Datenbank verwenden könnte.

Wie bereits erwähnt gewinnt der Client bei RIAs immer mehr an Bedeutung. Um all die erwähnten Features zu ermöglichen, ist es notwendig, den Zustand auf Client Seite in einem lokalen Datenmodell abzubilden und wesentliche Teile der Geschäftslogik dort unterzubringen. Für eine saubere Trennung dieser Bestandteile ist das MVC-Pattern unabdingbar, das in diesem Fall vollständig clientseitig zu implementieren ist. Bei Flex wird dieses Backend technisch durch ActionScript Objekte abgebildet (siehe Abbildung 1).

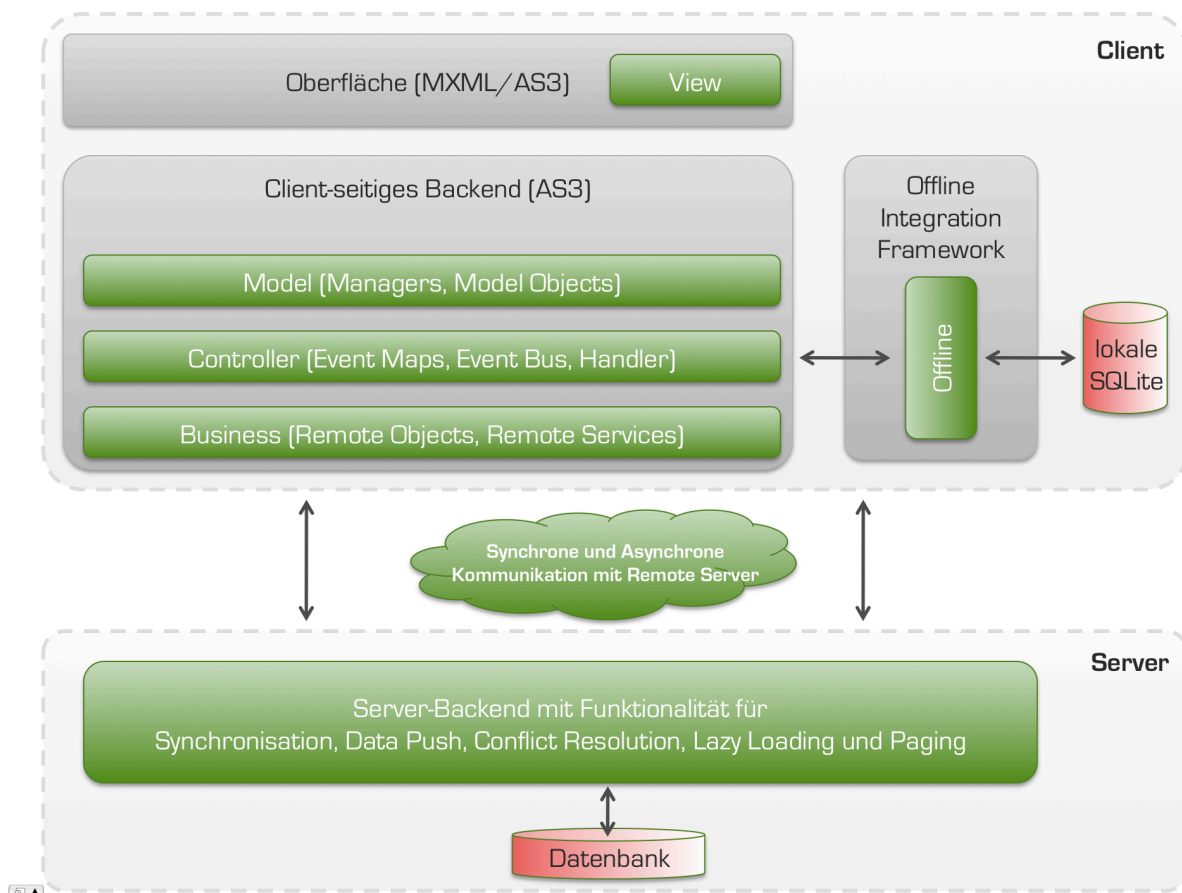


Abb. 1: Klassischer Aufbau einer RIA mit Flex

Spring - Backend

Von Anfang an verfolgten die Erfinder von Spring das Ziel dem Entwickler möglichst viel Flexibilität einzuräumen. Je nach Anwendungsfall können alle Module des Spring Frameworks verwendet werden, oder nur Teile davon (z.B. Spring Security). Spring ist eine echte Integrationsplattform und legt hohen Wert auf die Austauschbarkeit einzelner Frameworks. So kann man in Spring innerhalb kürzester Zeit die Persistenzschicht austauschen und beispielsweise iBatis anstatt Hibernate einsetzen. Gleiches gilt für die Anbindung des Frontends und der dort verwendeten Technologien. Egal ob man sich entscheidet mit JSP, JSF oder Adobe Flex Benutzeroberflächen zu realisieren – Spring hat die passenden Konzepte dafür bereits vorgesehen.

Flex schöpft seine Vorteile gegenüber klassischen Webanwendungen gerade dann aus, wenn wie bei Spring auch das Backend mitspielt und mehrere Clients zeitgleich effektiv mit einem (oder mehreren) Servern arbeiten. Adobe hat hier über Jahre hinweg Konzepte entwickelt, wie man die Serveranbindung optimiert und bietet den Entwicklern weitreichende Möglichkeiten Features wie Lazy Loading, Publish/ Subscribe Messaging sowie Conflict Resolution zu verwenden. Letzteres bietet die Möglichkeit Änderungen auf dem gleichen Datensatz konfliktfrei aufzulösen. Spring

unterstützt an dieser Stelle durch eine vorbildliche Integration der Remote Data Service Frameworks für Adobe Flex. Dies gilt sowohl für die freien Varianten BlazeDS und GraniteDS, als auch für die kostenpflichtige Version LCDS. Doch auch ohne den Einsatz weiterer Frameworks ist es mit Spring recht einfach möglich standardisierte Schnittstellen wie beispielsweise JSON/REST für die Kommunikation mit dem Frontend zur Verfügung zu stellen – in diesen Fällen, können dann jedoch die Vorteile der Remote Data Service Frameworks nicht genutzt werden.

Bei der Kommunikation mit dem Backend spielt zudem die Performance eine große Rolle. Kein Endanwender will heute noch 5 Sekunden nach Knopfdruck warten, bis seine Liste mit den tagesaktuellen Umsätzen auf dem Bildschirm erscheint, auch wenn er dank RIA Technologie nun in der Lage ist in der Zwischenzeit alle anderen Elemente der Oberfläche zu bedienen (asynchrone Kommunikation). Deshalb werden effektive binäre Protokolle gebraucht, welche die benötigten Daten komprimiert und ohne zusätzlichen XML-Overhead übertragen. Bei Adobe Flex wird hierfür das AMF Protokoll verwendet. Für die Anbindung der Datenbank und das objektrelationale Mapping unterstützt Spring den JPA Standard, der insbesondere mit der Oracle Toplink Implementierung für eine hervorragende Performance sorgt.

Zusammenfassung

Mit Adobe Flex steht Entwicklern eine ausgereifte Technologie für anspruchsvolle Webanwendungen zur Verfügung. Gerade bei komplexen Anforderungen wie der Offline- und Synchronisationsfähigkeit von RIA-Anwendungen oder aufwendigen Oberflächen kann das Framework seine Vorteile gegenüber klassischen Webtechnologien voll ausspielen.

Das Open Source Framework Spring ergänzt die Anforderungen an eine Rich Internet Application im Backend hervorragend. Mit durchdachten Security- und Schnittstellenkonzepten kann die Kommunikation zwischen Client und Server besonders effektiv implementiert werden. Gleiches gilt für die Anbindung der Oracle Datenbank mittels JPA oder alternativen Persistenzframeworks.

Die Kombination dieser Technologien stellt eine echte Alternative zur klassischen Webanwendung auf Basis von HTML, AJAX und Javascript dar und kann insbesondere bei komplexen Anforderungen einen hohen Mehrwert liefern.

Kontaktadresse:

B.Sc. Frank Szilinski
esentri software GmbH
Stephanienstr. 36
D-76133 Karlsruhe

Telefon: +49 (0) 721-7540 7530
Fax: +49 (0) 721-7540 7539
E-Mail: frank.szilinski@esentri.com
Internet: www.esentri.com