

# RIA Frameworks im Vergleich - JavaFX vs. Adobe Flex. Are you ready JavaFX?

Frank Szilinski  
esentri software GmbH  
Ettlingen

## Schlüsselworte:

RIA, Adobe Flex, JSF, Java Server Faces, Rich Internet Application

## Einleitung

Vor allem durch die grafischen Möglichkeiten werden Rich Internet Applications (RIA) immer beliebter. Im Oracle Technologie-Stack ist JavaFX für die zukünftigen Frontends vorgesehen und es stellt sich die Frage, wie weit die Entwicklung inzwischen ist, insbesondere vor dem Hintergrund, dass dieser Tage die Beta von JavaFX 2.0 veröffentlicht wurde.

Außerhalb des Java-Stacks gibt es mit Adobe Flex eine weiteres sehr ausgereiftes RIA-Framework als Referenz. Dieser Vortrag beschäftigt sich mit dem Vergleich von Adobe Flex und JavaFX. Dabei wird auf alle relevanten Themen, u.a. von der zu Grunde liegenden Softwarearchitektur bis hin zu den Möglichkeiten bei der Komponentenentwicklung eingegangen sowie Vor- und Nachteile der beiden Technologien in den neusten Versionen beleuchtet. Abgerundet wird der Vortrag durch einen Ausblick, was sich in der Welt der User Interfaces zukünftig noch alles entwickeln könnte.

Hinweis: zum Abgabeschluss dieses Abstracts hat die JavaOne Konferenz mit den aktuellsten Entwicklungen noch nicht stattgefunden, so dass im folgenden vielmehr auf allgemeine Unterschiede der Technologien eingegangen wird. Der Inhalt der Präsentation wird dann auf aktuellste Entwicklungen und Ankündigungen eingehen und diese ausführlich darstellen.

## Rückblick – der Weg von JavaFX war nicht geradlinig

Seit geraumer Zeit wurde im Java Universum wenig an der Modernisierung der Benutzeroberflächen gearbeitet. Neben Spring/AWT und den Webtechnologien rund um Java Server Pages und Java Server Faces gab es seit dem großen Erfolg der Applets keine wirklichen Innovationen im Bereich Frontends. Zwar war die Java Technologie lange wirklich solide Lösungen für die Entwicklung von Rich Client Applications im Angebot, jedoch mussten Entwickler zu viele Technologien dafür erlernen. Es dauerte bis zum Jahr 2007 als Sun Microsystems auf der JavaOne eine neue und aufregende Technologie mit dem Namen JavaFX als Pendant und Alternative zu den Rich Internet Application Frameworks Flex von Adobe und Silverlight von Microsoft angekündigte. Bis heute gibt es aber kaum reale Projekte auf Basis von JavaFX und die Marktdurchdringung ist sehr gering. Viele hatten die neue Technologie schon abgeschrieben.

Mit der Übernahme von Sun durch Oracle im April 2009 waren viele der Anhänger und Evangelisten des neuen Frameworks zum einen verunsichert, wohin die Reise führen soll, hatten andererseits aber auch die Hoffnung, dass Oracle der Technologie zum Durchbruch verhelfen würde. Tatsächlich gab Oracle schon kurz nach der Eingliederung von Sun ein klares Statement, dass die Entwicklung von JavaFX fortgeführt wird und darüber hinaus, ein Restrukturierung und Verbesserung des Kerns

erfahren sollte. Dabei stand die Leichtgewichtigkeit und eine optimale Hardwareunterstützung sowie die Integration von Seit Mai 2011 wird Oracle JavaFX 2.0 in einer Beta Version bei Oracle zum Download angeboten.

### Adobe Flex wurde ebenfalls zugekauft

Neben Sun und Oracle hat auch Adobe die Notwendigkeit eines echten Rich Internet Application Frameworks erkannt, um im Web anspruchsvolle Anwendungen anbieten zu können. So kaufte Adobe im Jahr 2005 seinen größten Konkurrenten Macromedia um seine strategische Position in der Welt der multimedialen und interaktiven Webanwendungen zu stärken. Nach der Übernahme und im Besitz der Flash Technologie erweiterte und rekonstruierte Adobe das Rich Internet Application Framework und hat es bis heute zum Big Player ausgebaut, an dem kein Entwicklungsteam bei der Auswahl zukunftsweisender Frontend Technologien für das Web vorbeikommt.

### Allgemeine Vorteile von Rich Internet Applications

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.

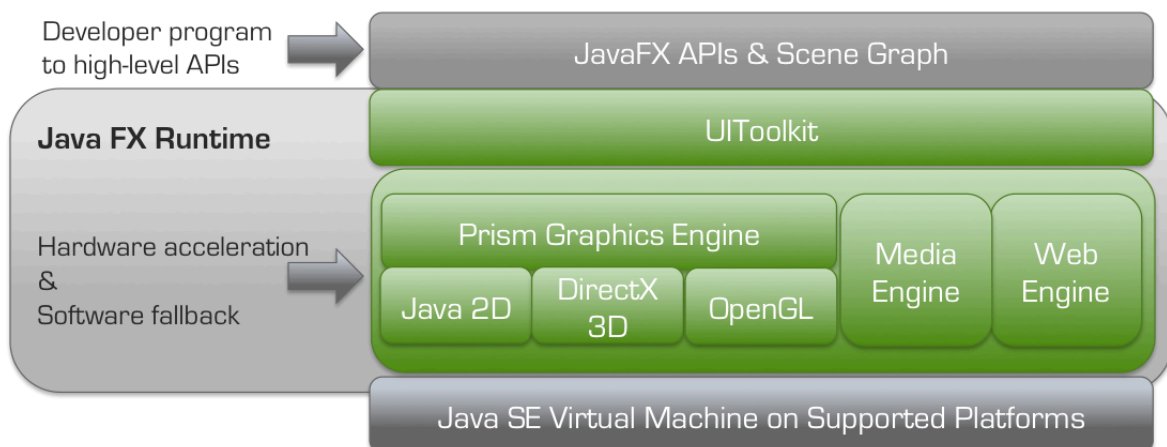


Abbildung 1: Architektur von JavaFX

## UI-Komponenten und Skinning

Der erste Blick bei der Auswahl einer neuen Frontend-Technologie fällt typischerweise auf die vorhandenen UI-Komponenten. Adobe Flex bietet den Entwicklern hier deutlich umfangreichere und praxiserprobtere Standard-Komponentenbibliotheken als JavaFX 2.0, 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 ActionScript Komponenten hervorragend unterstützt. Insbesondere im Bereich UI-Komponenten und Skinning hat sich mit JavaFX 2.0 einiges getan. So sind inzwischen immerhin mehr als 50 Komponenten und Charts verfügbar, die unter anderem mit Cascading Style Sheets Techniken angepasst werden können. Ein großes Ziel dabei war es auch die Hardwareunterstützung zu verbessern und so Performancevorteile und optimale Multimediaunterstützung zu erzielen. Dies gilt sowohl für 2D als auch für aufwendige 3D Visualisierung, bspw. bei der Berechnung von Schatten. Als zusätzliches Feature ist es nun auch möglich Webinhalte in die JavaFX 2.0 Oberflächen zu integrieren. Hierzu wird eine neue WebView Komponente angeboten, die als Embedded Browser klassische Webseiten darstellen kann. Verantwortlich für die Darstellung ist eine WebEngine, die die Webseiten vorab rendert und u.a. auch für die Interaktion, z.B. das Absenden von Formularen, zuständig ist (siehe Abbildung 2). Durch eine gute Integration in verschiedenste Entwicklungsumgebungen, allen Voran aber das hauseigene Netbeans, hat sich hier inzwischen auch in Richtung Produktivität einiges getan.

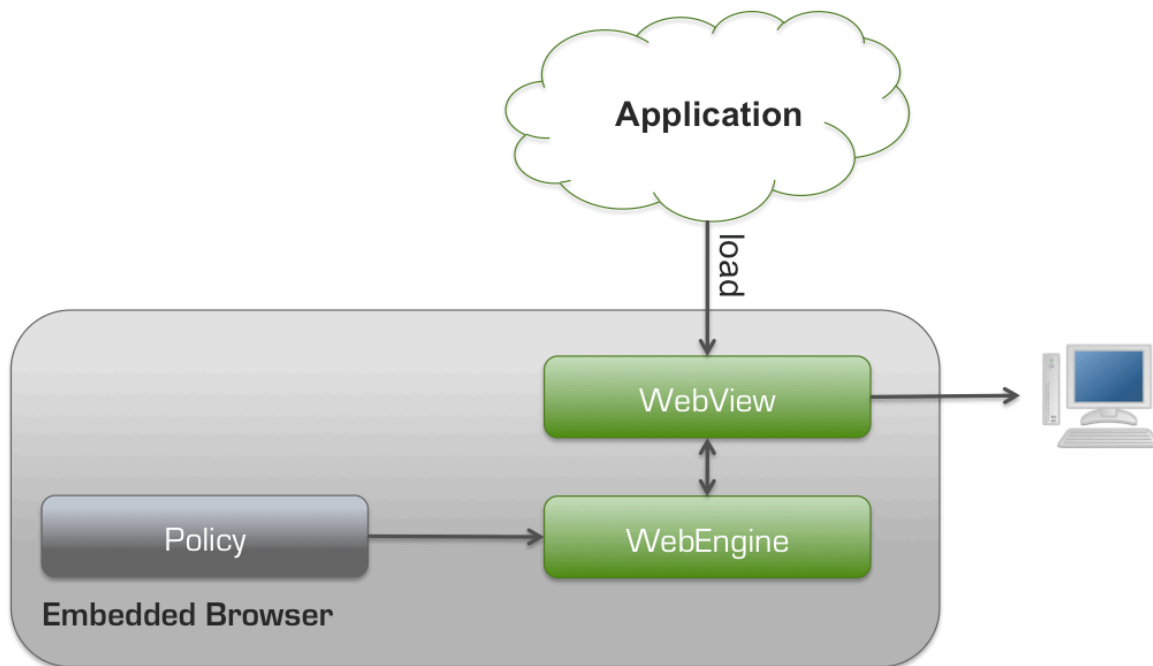


Abbildung 2: Integration klassischer Webinhalte in JavaFX 2.0

## **Offenheit und Integration: JavaFX 2.0 für alle Java Sprachen**

Adobe Flex ist seit je her recht offen für eine Integration in andere Sprachen und versteht sich als flexibles Frontend, das mit unterschiedlichsten Technologien integriert werden kann. Mit JavaFX 2.0 öffnet sich auch Oracles Frontendtechnologie. So werden zukünftig nicht nur die etablierten JVM Sprachen wie JRuby, Scala und Groovy unterstützt, sondern auch Nischensprachen wie Gosu und Visage eingebunden.

Neben dem allgemeinen Event-Konzept wurde zusätzlich auch an der Zusammenarbeit mit Swing Applikationen gearbeitet. So ist es zwar weiterhin nicht möglich bewährte Swing Komponenten in JavaFX 2.0 einzubauen, jedoch besteht nun die Möglichkeit JavaFX Komponenten auch unter gewissen Voraussetzungen in Swing Anwendungen zu verwenden und somit die Vorteile wie z.B. im Bereich von Animationen direkt nutzen zu können.

### **Zusammenfassung**

Mit Adobe Flex steht Entwicklern eine ausgereifte Technologie für anspruchsvolle Webanwendungen zur Verfügung. Flex hat sich inzwischen über Jahre bewährt und mit der aktuellsten Version bietet Adobe weitere spannende Möglichkeiten für die Realisierung von Rich Internet Applications und mobilen Anwendungen an.

JavaFX 2.0 steht nun endlich in den Startlöchern, und kann als eine echte Alternative zu den gängigen Technologien in Betracht gezogen zu werden. Nachdem die Betaphase in Kürze abgeschlossen sein soll, wird es spannend sein, inwiefern sich die guten Konzepte auch in praxistaugliche Lösungen umwandeln lassen. Bisher sind es vor allem kleine experimentelle Applikationen und technische Blueprints, die das Potential von JavaFX 2.0 aufzeigen. So bleibt es bis zur Präsentation der Neuerungen auf der JavaOne spannend – der Vortrag wird ausführlich auf die letzten Neuerungen eingehen und einen detaillierten Featurevergleich aufzeigen.

Kontaktadresse:

Frank Szilinski, B.Sc.  
esentri software GmbH  
Pforzheimer Straße 132  
D-76275 Ettlingen

Telefon: +49 (0) 7243-354 90 0  
Fax: +49 (0) 7243-354 90 99  
E-Mail [frank.szilinski@esentri.com](mailto:frank.szilinski@esentri.com)  
Internet: [www.esentri.com](http://www.esentri.com)