



Rich Internet Applications mit Java

Rich Internet Applications (RIA) bezeichnen  
Web-Applikationen, die mit einer wesentlich  
interaktiveren Benutzerschnittstelle ausgestattet  
sind, als man das bisher von den auf HTML  
basierten Web-Anwendungen gewohnt war.

# Hohe Benutzerproduktivität für Unternehmensanwendungen

Illustration: Shutterstock



Während die Effekte des Einsatzes von RIA-Technologie der Öffentlichkeit vor allem durch Internet-Dienste wie Flickr oder GoogleMaps bekannt sind, bringt diese neue Entwicklungsstufe der Web-Technologie auch für die Informatik in Unternehmen massive Verbesserungen und erweitert den Anwendungsbereich von Web-Applikationen entscheidend.

RIA ermöglicht die Bereitstellung von Schnittstellen für sowohl Gelegenheitsbenutzer als auch Experten-Benutzer mit höchsten Anforderungen auf Basis derselben Technologie. Unternehmen, die heute diese unterschiedlichen Benutzergruppen noch mit verschiedenen Benutzerschnittstellenimplementierungen (etwa HTML und Rich Clients) bedienen, können mit RIA nun ihr Applikationsportofolio konsolidieren. Globale Geschäftsprozesse lassen sich erheblich einfacher automatisieren, da RIA für die notwendige Interaktivität der am Prozess beteiligten Benutzer sorgt und niedrigere Anforderungen an die Netzwerkinfrastruktur zur Erzielung der notwendigen Performanz stellt. In Kombination mit einer service-orientierten Architektur lassen sich zentrale, häufig mainframebasierte Informationsdienste, in ein modernes Gewand kleiden. Und erst RIA-Technologie macht die Bereitstellung von Produktiv-Software wie beispielsweise ERP-Systeme als Service (SaaS) möglich.

## Auswahlkriterien

Für die Umsetzung von Rich Internet Applikationen stehen heute unterschiedliche Technologien zur Verfügung. Die wichtigsten Kriterien für die Auswahl der geeigneten Technologie sind:

- An welche Benutzer richtet sich die Anwendung – soll sie innerhalb eines Unternehmens oder als B2B-Lösung mit Geschäftspartnern eingesetzt werden oder richtet sich die Anwendung an beliebige Benutzer im Internet?
- Wie arbeiten die Benutzer mit der Anwendung? Wird sie gelegentlich aufgerufen oder wird sie von Experten täglich und sehr intensiv zur Erledigung von wichtigen Aufgaben verwendet?
- Handelt es sich bei der Anwendung um eine Geschäftsanwendung (z.B. ein CRM- oder ERP-System) oder um

eine Anwendung mit Unterhaltungsscharakter, in denen Animationen und Multimedia eine grosse Rolle spielen?

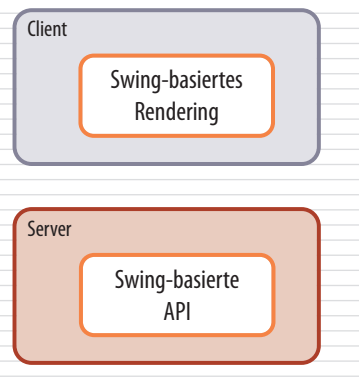
Insbesondere für Geschäftsanwendungen im Unternehmensumfeld, die von Experten für ihre tägliche Arbeit benutzt werden, empfiehlt sich der Einsatz von rein Java-basierter Technologie, die nicht von den Spezifika der Browser und ihren inhärenten Beschränkungen, die Anwendungen substantiell langsamer machen, abhängt. Ein kürzlich veröffentlichter Bericht von Forrester hat die Probleme von Anwendungen, die auf Ajax/JavaScript basieren, analysiert. Zudem verspricht die Verwendung bewährter Java-Technologie beim heute noch hoch fragmentierten Markt mit vielfältigen aber unreifen Lösungen einen höheren Investitionsschutz und langfristig niedrigere Wartungskosten. Und dass Java auch noch eine Zukunft hat und man hier nicht auf eine End-of-Life Technologie setzt, zeigen die vielfältigen Bestrebungen, Java endlich auch erfolgreich auf mobile Geräte zu bringen. Sun wagt einen neuen Vorstoss mit Java FX. Und ein vielleicht noch besseres Indiz dafür ist der Umstand, dass Google die Entwicklung mobiler Anwendungen mit Java durch die Android-Plattform unterstützen will.

### JavaScript oder eigene Laufzeitumgebung

Eine Rich Internet Applikation kann man heute auf viele Arten implementieren wobei sich hinsichtlich der Ausführung einer solchen Applikation zwei Hauptunterscheidungsmerkmale identifizieren lassen:

- In welcher Laufzeitumgebung wird die Anwendung ausgeführt? Direkt im Browser (Ajax/JavaScript), in einem Plugin des Browsers (JRE, Flash-Player, Silverlight) oder in einer vom Browser unabhängigen Laufzeitumgebung (Adobe AIR, JRE, Silverlight)?
- Welcher Teil der Applikation wird auf dem Client und welcher Teil wird auf dem Server ausgeführt?

Ajax/JavaScript-Anwendungen haben den großen Vorteil, dass sie ohne Installation einer zusätzlichen Komponente auskommen. Ideal für Anwendungen, die sich an beliebige Benutzer



**Bild 1:** UltraLightClient kommt ohne JavaScript aus, erfordert aber, dass das JRE-Plugin (in beliebiger Version) im Browser installiert ist.

am Internet richten, über deren Client-Umgebung man keine Annahmen machen kann. Dafür erkaufte man sich aber die Abhängigkeit von den Beschränkungen des Browsers, die sich durch zwei sehr unangenehme Eigenschaften auf die Anwendungen auswirken:

- Die Anwendung kann sich auf unterschiedlichen Plattformen (Betriebssystem, Browser) unterschiedlich verhalten – selbst wenn eine Ajax-Bibliothek eingesetzt wird, die versucht, diese Browser-Abhängigkeiten vor dem Entwickler zu verbergen. Dies führt zu erhöhtem Test- und Supportaufwand auf Seiten des Anbieters der Anwendung.
- Ajax/JavaScript-Anwendungen sind deutlich langsamer als vergleichbare Anwendungen, die in dedizierten Laufzeitumgebungen ausgeführt werden. Dies liegt daran, dass Browser ursprünglich als reine Anzeigewerkzeuge für Dokumente konzipiert waren und das dafür verwendete interne Modell – das Document Object Model (DOM) – nicht als Ausführungsumgebung für komplexe Anwendungen geeignet ist.

Will man diese Nachteile vermeiden, bleibt nur der Ausweg auf eine nicht

browser-basierte Laufzeitumgebung. Setzt man hierbei auf die Java-Laufzeitumgebung, so kann man Applikationen auf der Basis einer homogenen, objektorientierten Technologie erstellen, ausgereifte, stabile GUI-Bibliotheken (etwa Swing) mit sehr hoher Funktionalität sowie die einschlägigen Umgebungen für Entwicklung und Testen verwenden.

### Auslagerung der Präsentationslogik

Um eine RIA mit dem Flash-Player ausführen zu lassen, muss diese mit Flex implementiert werden. Die Programmiersprache ist hier ActionScript (eine JavaScript-artige Sprache) ergänzt um spezifische XML-Dialekte zu Beschreibung und Animierung der Benutzerschnittstelle, d.h. die Entwickler müssen mehrere Sprachen und Technologien und ihr Zusammenspiel beherrschen. Bei Silverlight werden alle Programmiersprachen, die unter .NET ausführbar sind, unterstützt. Adobe AIR stellt eine Laufzeitumgebung völlig unabhängig vom Browser zur Verfügung, auf der Web-Applikationen, die mit HTML, CSS, JavaScript, Flex oder auch Java implementiert wurden, als Desktop-Applikationen ausgeführt werden. Alle diese Ansätze lagern die Präsentationslogik auf den Client aus.

### Nachteilige Entwicklungskosten

Über die Vor- und Nachteile einer bestimmten Programmiersprache mag man unterschiedlicher Auffassung sein. Folgende Aspekte sind allerdings generell nachteilig für die Entwicklungs- und Wartungskosten sowie für die Skalierbarkeit einer Anwendung hinsichtlich Komplexität und Performance:

- Die Heterogenität des Programmiermodells. Je mehr unterschiedliche Typen von Programmkomponenten



## JavaScript-Anwendungen

sind durch Browser eingeschränkt, da diese nicht für komplexe Anwendungen konzipiert sind.

zu einer Anwendung zusammengefügt werden müssen (HTML, CSS, JavaScript, Java, XML) desto komplexer wird Testen, Fehlersuche, Optimierung und Erweiterung eines solchen Systems.

- Die physische Verteilung von Anwendungen auf Client und Server erhöht die Komplexität der Entwicklung und Wartung erheblich.

Das sind Gründe, die dafür sprechen, für die Entwicklung einer komplexeren Geschäftsanwendung auf eine Technologie zu setzen, die sowohl ein homogenes Programmiermodell unterstützt als auch die Verteilung von Anwendungslogik soweit wie möglich vor dem Entwickler verbirgt. Die im nachfolgenden kurz beschriebene rein Java-basierte Lösung erfüllt diese Kriterien.

### Ein homogener Java-basierter Ansatz

Die Vorzüge des homogenen Programmiermodells von Java haben verschiedene Ansätze zur Implementierung von Rich Internet Applications auf unterschiedliche Art umgesetzt:

- GWT (Google Window Toolkit – <http://code.google.com/webtoolkit/>) erlaubt dem Entwickler seine Anwendung in Java zu entwickeln. Der Java-Code wird jedoch anschließend in JavaScript übersetzt – und damit „erbt“ dieser Ansatz alle der oben erwähnten Nachteile einer JavaScript-basierten Lösung.
- Eclipse RCP ([http://wiki.eclipse.org/index.php/Rich\\_Client\\_Platform](http://wiki.eclipse.org/index.php/Rich_Client_Platform)), eigentlich eine Entwicklungsplattform für Desktop-Anwendungen, geht einen ähnlichen Weg, um Rich (Fat) Clients internetfähig zu machen: GUI-Komponenten werden in JavaScript übersetzt – mit den gleichen Nachteilen wie bei GWT.



Es gibt viele Gründe, für die Entwicklung einer komplexeren Geschäftsanwendung auf nur eine Technologie zu setzen.

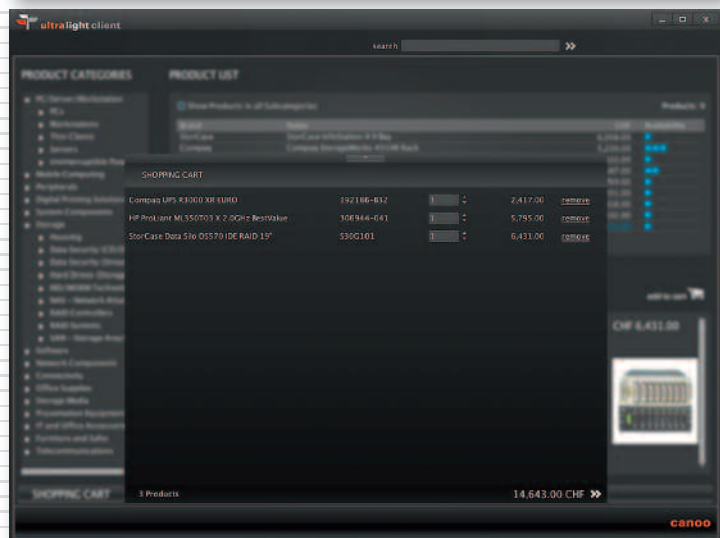
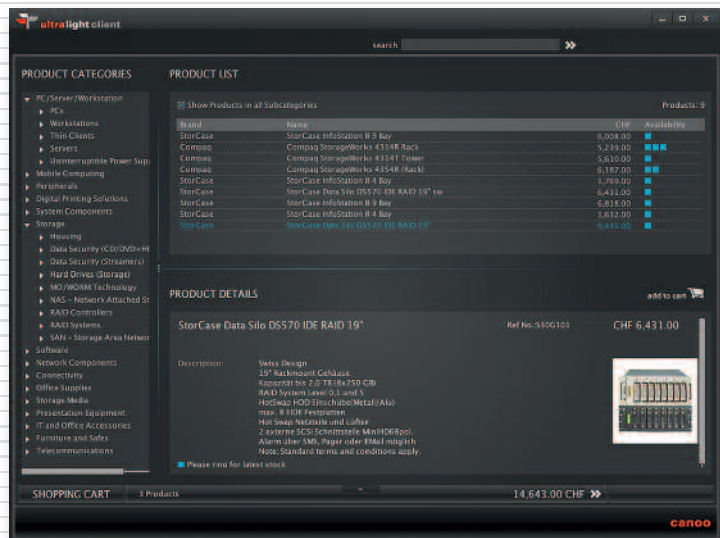


Bild 2 und Bild 3: Die Grafiken zeigen Screenshots eines mit UltraLightClient implementierten Online-Shops.

- UltraLightClient (<http://www.canoo.com/ulc>), eine GUI-Bibliothek, verwendet Java/Swing auf dem Client, um die Benutzerschnittstellenkomponenten darzustellen. Dieser Ansatz kommt daher ohne JavaScript aus, erfordert aber, dass das JRE-Plugin (in beliebiger Version) im Browser installiert ist. Dieser Ansatz soll nachfolgend kurz beschrieben werden.

### Ultra Light Client

Die Grundidee von UltraLightClient besteht darin, auf dem Client das GUI auf der Basis von Swing darzustellen (statt HTML) und die gesamte Applikation auf dem Server gegen ein ebenfalls auf Swing-basierendes API zu programmieren. Zur Laufzeit wird eine Beschreibung der Benutzerschnittstelle an einen Java-basierten Renderer übermittelt, der die Benutzerschnittstelle mithilfe von Swing zur Darstellung bringt und relevante Events wieder an den Server übermittelt. Der Renderer ist sehr kompakt sowie applikations- und JRE-versionsunabhängig und muss daher nur einmal auf dem Client installiert werden.

## Warum Swing?

Warum hat man sich ausgerechnet für Swing als Basis einer RIA-Lösung entschieden obwohl das Look & Feel von Swing-Anwendungen als langweilig und nicht mehr den heutigen Ansprüchen an UI-Ästhetik genügend gilt? Swing ist in der Java-Welt ein de-facto Standard, der eine vollständige komponentenbasierte Bibliothek für die Benutzerschnittstellen zur Verfügung stellt. Auch heute noch müssen sich alternative Komponentenbibliotheken, die zum Beispiel im Ajax-Umfeld realisiert werden, an diesem Standard messen lassen. Swing stellt damit eine umfassende Basis für die Entwicklung hoch-interaktiver, reichhaltiger Benutzerschnittstellen dar. Ferner ist die Desktop-Integration von Swing in die verbreiteten Betriebssysteme (Windows, Mac OS X, Linux) relativ gut und es bietet ein kohärentes und über viele Jahre bewährtes Programmiermodell an. Zudem benötigt ein Swing-Entwickler praktisch keine Zeit für den

Umstieg auf die Entwicklung von UltraLightClient-Applikationen.

Und Bild 2 und Bild 3 belegen, dass Swing-basierte Anwendungen keineswegs hinter den heutigen Standards attraktiver Benutzerschnittstellen zurückbleiben müssen. Die Grafiken zeigen Screenshots eines mit UltraLightClient implementierten Online-Shops.

## Kompatibilität

Um das Modell HTML-basierter herkömmlicher Web-Applikationen nicht zu brechen und vorhandene Infrastruktur für den Betrieb von Web-Applikationen auszunutzen, ist UltraLightClient kompatibel mit der Java EE Spezifikation, das heißt alle server-seitigen Komponenten können in einem Standard-Servlet- oder EJB-Container ausgeführt werden.

Die Kommunikation zwischen der clientseitigen Rendererkomponente und der serverseitigen Präsentationslogik ist hochoptimiert und ermöglicht auch bei niedrigen Bandbreiten mit

hoher Latenzzeit eine performante Benutzerschnittstelle.

Normalerweise läuft eine UltraLightClient-Anwendung auf dem Server. Es ist aber auch möglich sie ähnlich wie eine Adobe AIR-Anwendung als Desktop-Anwendung vollständig auf dem Client zu installieren. Ein wichtiger Grund für diese Installationsvariante kann die Notwendigkeit sein, die Anwendung offline, also ohne Verbindung zu einem Server, zu betreiben.

## Fazit

Java stellt eine solide Plattform für Rich Internet Applikationen dar, auf deren Grundlage Benutzerschnittstellen implementiert werden können, die auch die hohen Ansprüche von Fachkräften in einem Unternehmen, die täglich ihre Arbeit mit dieser Anwendung verrichten, erfüllen lassen. Und diese Applikationen können auf der Standard-Web-Plattform der herkömmlichen Web-Anwendungen betrieben werden.

*HANS-DIRK WALTER*