

# Java Frameworks im Vergleich - ADF vs. Grails vs. Spring

Frank Szilinski  
esentri software GmbH  
Karlsruhe

## Schlüsselworte:

ADF, Java, JEE, JSF, Grails, Spring, Open Source, Rapid Application Development

## Einleitung

In diesem Vortrag werden drei beliebte Application Development Frameworks miteinander verglichen und ein Blick über den Tellerrand der Oracle JEE Entwicklung gewagt. Hierzu werden die spezifischen Eigenschaften der unterschiedlichen Frameworks herausgestellt - individuelle Stärken und Schwächen spielen dabei genauso eine Rolle wie praktische Tipps & Tricks zu den einzelnen Technologien und konkreten Erfahrungen in Projekten.

Die Zuhörer können am Ende selbst entscheiden, welches Framework sich für welchen Anwendungsfall am besten eignet.

## Die Qual der Wahl

Obwohl heute im JEE Standard schon sehr viele vereinfachende Konzepte enthalten sind, werden viele Java Enterprise Anwendungen auf Basis von Frameworks entwickelt. Die Entwickler haben dabei die Qual der Wahl und müssen sich zwischen verschiedenen Ansätzen und Alternativen entscheiden.

Zentral sind dabei insbesondere folgende Fragestellungen:

- wie schnell kommt man zum Ergebnis?
- eignet sich das Framework für den konkreten Anwendungsfall?
- wie hoch ist die Lernkurve?
- wie gut werden gängige Standards unterstützt?
- wie flexibel ist die Lösung?
- lässt sich das Framework auch mit anderen kombinieren?
- welche Oberflächentechnologien passen zum Framework?
- ist die Lösung ausgereift, gibt es Support und eine gute Dokumentation?

Der Vortrag beschäftigt sich mit diesen Fragestellungen und gibt Entscheidungshilfen zu den drei ausgewählten Frameworks für die Praxis. Zum besseren Verständnis werden im folgenden die Grundlagen und Schwerpunkte der drei ausgewählten Frameworks herausgestellt.

## ADF - produktiv mit Oracle im Standard

Oracle hat mit ADF und der perfekten Integration in die IDE JDeveloper Maßstäbe bei der JEE Entwicklung gesetzt. Seit geraumer Zeit können mit dem Oracle ADF Framework produktiv und konform zum JEE Standard komplexe Webanwendungen realisiert werden. Der Hauptvorteil von ADF liegt in der Toolunterstützung und der Integration in den Oracle Technologiestack. Dies gilt

insbesondere für die Möglichkeit die komplette Persistenzschicht für bestehende Datenbankstrukturen zu generieren und mittels Data Bindings entsprechende Oberflächen per Drag & Drop zu designen. Die generelle Herangehensweise lehnt sich an der Entwicklung von Forms Anwendungen an und bietet zunächst auch unerfahrenen Java Entwicklern die Möglichkeit aufwendige JEE Anwendungen zu implementieren.

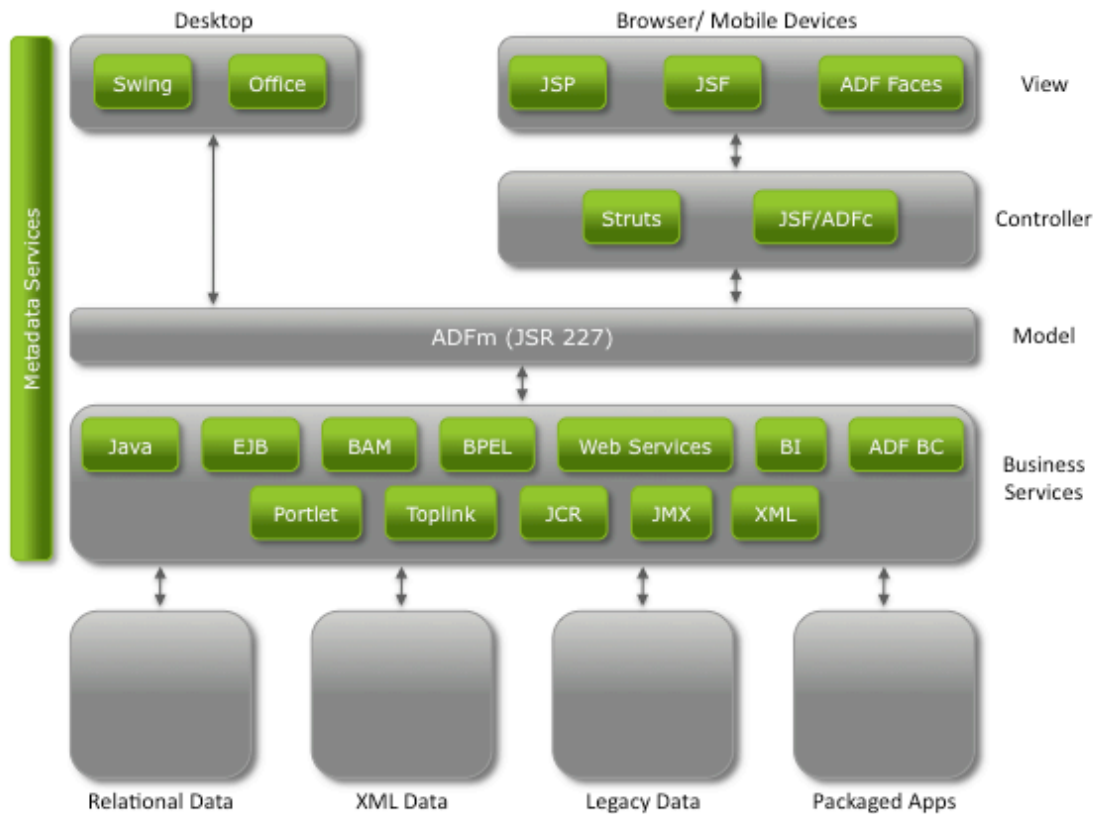


Abb. 1: Die Architektur von Oracle ADF

Gerade die Implementierung der Geschäftslogik (Business Services) ist bei JEE Anwendungen extrem wichtig und oft ein Auswahlkriterium für ein Framework. ADF bietet von Haus aus die Möglichkeit viele Standardtechnologien einzubinden. Neben den ADF Business Components können Enterprise Java Beans, Web Services oder einfache Java Klassen verwendet werden, was eine saubere Trennung zwischen der Geschäftslogik und dem Frontend ermöglicht. Oracle setzt dabei auf den JSR 227 Standard und das Konzept der Data Bindings, um von der konkreten Implementierung der Services zu abstrahieren. Dabei achtet Oracle darauf, dass ausschließlich gängige Standards verwendet werden. Die Entscheidung, welche Technologie im konkreten Projekt zum Einsatz kommt, trifft der Entwickler (Development by Choice).

Trotzdem wird ADF außerhalb der Oracle Community kaum verwendet. Die Gründe hierfür sind vielfältig, z.B.:

- ADF ist nicht kostenlos verfügbar
- die Abhängigkeit von den Oracle Bibliotheken wird z.T. als zu groß eingeschätzt
- erfahrenen Java Entwicklern wird tendenziell zu viel generiert

- Standardanforderungen sind zwar einfach umzusetzen, spezifische Anforderungen erfordern jedoch tiefgreifende Kenntnisse über den internen Aufbau von ADF
- Unzureichende Integration in Konfigurationsmanagement-Tools wie Maven
- Testbarkeit von ADF Anwendungen ist z.T. schwierig

Das Framework ist dennoch sehr ausgereift und Großprojekte wie z.B. die Oracle Fusion Application zeigen, wie effektiv sich neue JEE Anwendungen mit Oracle ADF entwickeln lassen.

### Spring - flexible Open Source Alternative

Mit dem Open Source Framework Spring steht ein einfach verständliches, konsistentes Programmier- und Konfigurationsmodell als Alternative zur Verfügung. Als Hauptmerkmal ist es besonders leichtgewichtig und kommt ohne große Abhängigkeiten aus. Viele Vorteile des Spring Frameworks sind inzwischen aber auch im JEE 6 Standard enthalten und es bleibt die Frage, wann sich der Einsatz des Spring Frameworks lohnt. Schließlich wird der Marktführer im Open Source Umfeld weltweit von Millionen Entwicklern eingesetzt und als weiteres Indiz für den großen Erfolg wurde Springsource 2009 für 420 Mio Dollar von der Firma VMware gekauft.

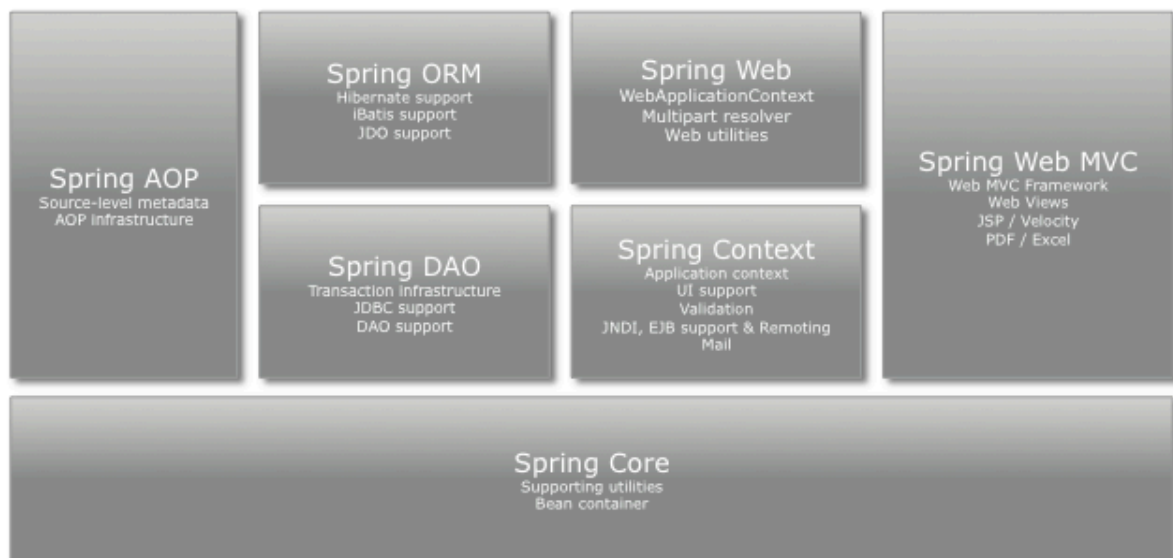


Abb. 2: Die Hauptkomponenten des Spring Frameworks

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. Der Grundgedanke dabei geht allerdings weiter wie bei ADF, das eher auf die End-to-End Programmierung ausgelegt ist. Spring ist vielmehr eine echte Integrationsplattform, das hohen Wert auf die Austauschbarkeit einzelner Frameworks legt. 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. Diese Flexibilität wird von

vielen Entwicklern weltweit geschätzt und garantiert Unternehmen eine hohe Investitionssicherheit. So ist es nicht verwunderlich, dass immer mehr Unternehmen entscheiden, große Teile ihrer Java Projekte mit Spring zu realisieren, zumal Spring Projekte grundsätzlich auf beliebigen Application Servern ausgeführt werden können.

### Grails - Rapid Application Development

Insbesondere für kleine JEE Projekte eignet sich auch Grails. Grails (Groovy on Rails) ist ein erweitertes und innovatives Rapid Applikation Framework, das auf den Sprachkonstrukten von Groovy und Ruby on Rails beruht. Grails basiert auf verschiedenen Prinzipien und Konzepten, die es dem Framework ermöglichen Entwicklungszeit einzusparen und Code zu vereinfachen. Es eignet sich daher hervorragend für kleinere Projekte, da es Entwicklern einen sehr schnellen Einstieg in die JEE Entwicklung ermöglicht. Als Basis nutzt Grails die etablierten Java Frameworks Spring, Spring MVC, Hibernate und SiteMesh. Das Ergebnis ist somit immer Java Bytecode und Anwendungen sind dadurch auf jedem gängigen Application Server lauffähig. Die Open Source Quellcodes werden inzwischen von SpringSource gepflegt und stehen indirekt auch in Konkurrenz zum Spring eigenen Rapid Application Framework Roo.

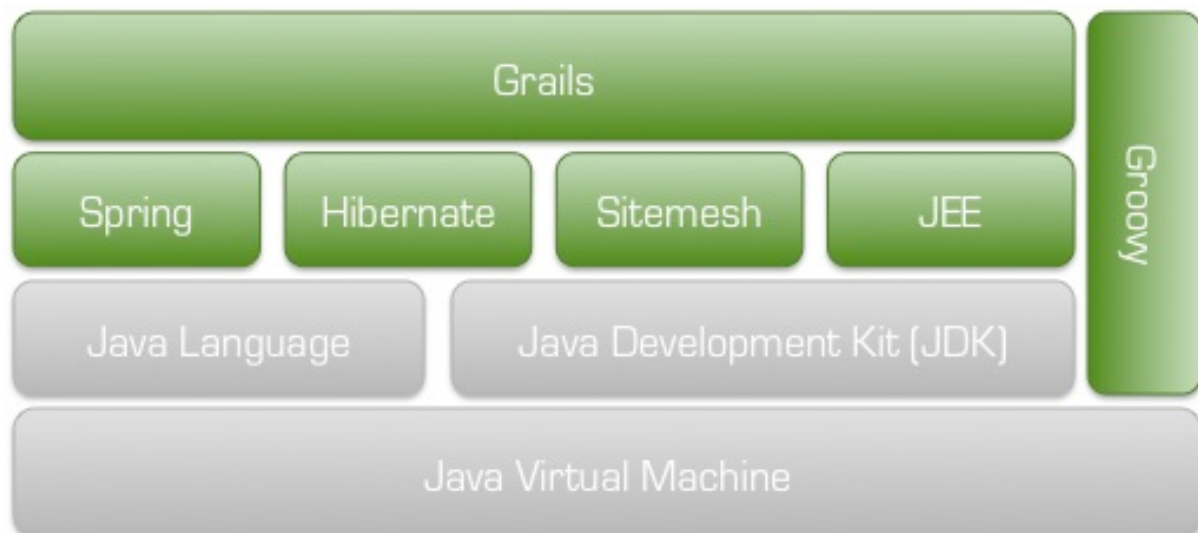


Abb. 3: Bestandteile von Grails

Der Grundaufbau von Grails basiert auf dem MVC-Pattern (Model-View-Controller), das in Grails mit Domäinklassen, Controllerklassen und Groovy Server Pages (GSP) umgesetzt wird. Prinzipiell ist es aber auch möglich andere Oberflächentechnologien wie Adobe Flex in der View einzusetzen.

Die Domäinklassen (Model) sind der Kern der Anwendung und bilden das Geschäftsmodell ab. Domäinklassen sind persistent und entsprechen einer physischen Tabelle in der Datenbank.

Der wesentliche Vorteil von Grails ist es, Konventionen vor die Konfiguration (Convention over Configuration) zu stellen. Damit wird der Konfigurations- und Entwicklungsaufwand deutlich reduziert. Durch ein ausgefeiltes Plugin-Konzept können Grails Anwendungen zudem schnell durch vordefinierte Funktionen erweitert werden. Grails ist eine Plugin-Laufzeitumgebung, die Standardplugins (z.B. GORM) umfasst und andere Plugins lädt und konfiguriert. Grails-Plugins ansich sind kleine Grailsanwendungen, die in andere eingebunden werden in dem alle relevanten Konfigurationen zusammengeführt werden.

## **Zusammenfassung**

ADF & Spring sind ausgereifte Frameworks, das sich hervorragend für JEE Projekte eignen. Während Oracle mit ADF vor allem auf standardisierte Entwicklung setzt, steht bei Spring die Flexibilität und Erweiterbarkeit im Vordergrund, die in vielen Projekten entscheidend sein kann. Grails ist ein junges Framework, mit dem sehr schnell einfache JEE Anwendungen erstellt werden können.

## **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