

Aufbau einer Web 2.0 Community mit Grails

Nicole Hoock
esentri consulting GmbH
Karlsruhe

Schlüsselworte:

Web 2.0, Grails, Community, Social Network, Java, Open Source, Rapid Application Development

Einleitung

Im Zeitalter des Internets hat sich die Nutzung von Web 2.0 Communities (wie z.B. Facebook, XING oder LinkedIn) in weiten Bereichen des Alltags etabliert. Aber auch immer mehr Unternehmen entscheiden sich für den Einsatz von sozialen Netzwerken, um mit Ihren Kunden und Mitarbeitern besser in Kontakt treten zu können.

Für den Aufbau von sozialen Netzwerken gibt es zahlreiche Frameworks und Open Source Projekte, die unterschiedliche Anforderungen bedienen.

Individualentwicklung als Option

Gerade die unterschiedlichen Schwerpunkte der Frameworks sorgen jedoch auch dafür, dass es nicht einfach ist die „richtige“ Software zu finden. Es stellt sich die Frage, wie aufwendig es mit heutigen Technologien ist, eine Web 2.0 Community Plattform selbst zu entwickeln.

Folgende Rahmenbedingungen bzw. Herausforderungen waren gesetzt:

- Kurze Entwicklungszeit
- Erweiterbarkeit durch Plugins und Bibliotheken
- Open-ID Authentifizierung
- Bereitstellung von Remote Schnittstellen (API)
- Geringer Einarbeitungsaufwand und Eignung für unerfahrene Entwickler
- Laufzeitumgebung Java
- Trennung von Logik & Präsentation (MVC)
- Einfach anpassbare Oberflächen
- Oracle Datenbank

Grails als Allheilbringer?

Basis für diesen Vortrag ist das Framework Grails, das für eine exemplarische Implementierung aus den vielen Möglichkeiten ausgewählt wurde. Der Vortrag bietet, neben einer kurzen Einführung in Grails, einen Einblick in die Herausforderungen der Entwicklung einer Web 2.0 Community und ist zeitgleich ein Erfahrungsbericht zum praktischen Umgang mit der Sprache Groovy und Grails. Dabei wird aufgezeigt, welche Stärken und Schwächen das Framework hat und die Frage beantwortet, ob Grails bereits praxistauglich ist.

Fachlich: die Idee der Web 2.0-Community

Die Tatsache, dass Communities bei den Internetnutzern durchgesetzt haben und sich immer mehr zum Einstiegspunkt in das Internet etablieren, veranlasst immer mehr Unternehmen dazu eigene Communities anzubieten. Unternehmen wollen damit sowohl die interne als auch die externe Kommunikation und Vernetzung fördern.

Das primäre Ziel bei der Förderung der internen Kommunikation ist der Wissenstransfer innerhalb des Unternehmens, da Wissen zu einer wichtigen Ressource geworden ist. Aber auch der Informationsaustausch und eine bessere Zusammenarbeit (Kollaboration) sollen mit dem Einsatz bezweckt werden.

Auf der externen Seite erkennen Unternehmen immer mehr, dass nicht mehr der reine Konsument sondern der Prosumer (Producer & Consumer), ein Kunde der nicht nur konsumiert sondern auch Inhalte produziert, im Vordergrund steht. Dies führt wiederum dazu das nicht mehr das Unternehmen allein Inhalte zu seinen Produkten und Dienstleistungen im Netz erzeugt, vielmehr ist dies die Aufgabe des Kunden geworden. Folglich spielt die Community eine immer größerer Rolle im Bereich der Kundenbindung und Marketings, denn es gibt keine bessere Werbung und Imagepflege, als die Zufriedenheit eines Kunden.

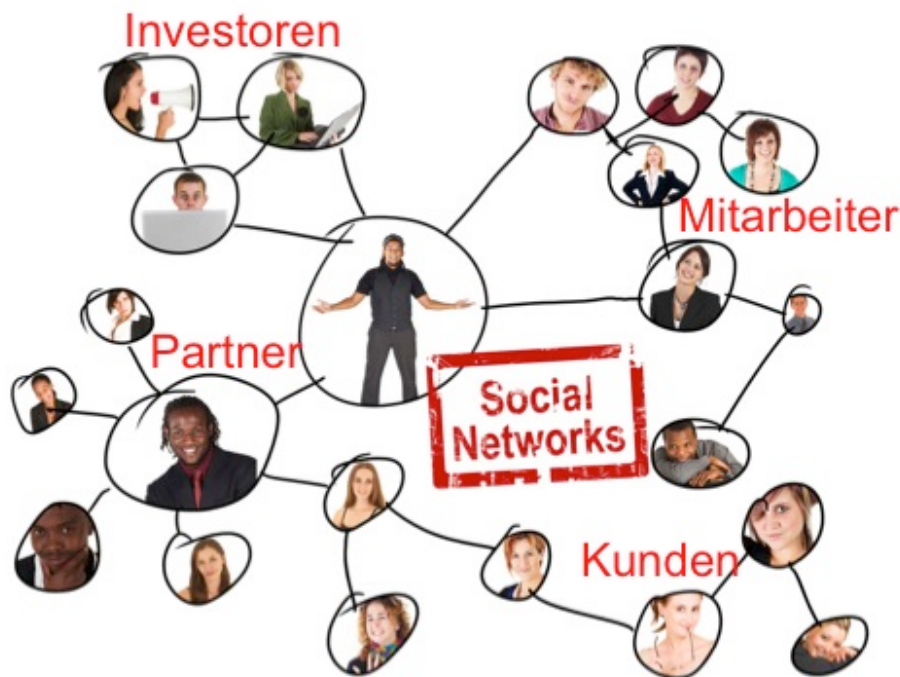


Abb. 1: Social Network

Dennoch haben Unternehmen mit Herausforderungen wie mangelnder Akzeptanz und der Unternehmenskultur zu kämpfen, wenn es um die Entwicklung und Einführung einer Community geht. Ursachen hier für sind meistens die fehlende Unterstützung durch das Management sowie eine fehlende Kultur der Offenheit und des Vertrauens.

Aus technischer Sicht stoßen Unternehmen häufig bei der Integration von neuen Anwendungen in ihre Unternehmensinfrastruktur auf Probleme. Es ist daher wichtig gängige Industriestandards zu berücksichtigen. In den meisten Fällen werden diese Infrastrukturen im Java-Umfeld betrieben und Integrationsprobleme können dann vermieden werden, wenn die Laufzeitumgebung die Java Virtual Maschine (JVM) ist.

Technisch: Was ist Grails?

Grails (Groovy on Rails) ist ein erweitertes und innovatives Web Applikation Framework, das auf den Sprachkonstrukten von Groovy und Ruby on Rails beruht. Darüber hinaus nutzt Grails die etablierten Java Frameworks Spring, Spring MVC, Hibernate und SiteMesh. Das Ergebnis ist immer Java Bytecode und so verhält sich eine Grails Anwendung auch, wie eine JEE-Anwendung. Sie ist dadurch auf jedem gängigen Application Server lauffähig.

Das Grails-Projekt wurde im Jahre 2005 gestartet. Das erste Release von Grails, Release 0.1, wurde 2006 veröffentlicht. Im Jahr 2008 folgte Release 1.0, außerdem, wurde das Unternehmen Groovy und Grails Anfang 2008 von Spring Source aufgekauft und Ende 2008 zusammen mit Spring Source von VMware übernommen. Im Jahr 2009 erschien das finale Release 1.2 und Anfang diesen Jahres das finale Release 1.3. Das aktuell verfügbare und für die Entwicklung der Community verwendete Release ist die Version 1.3.4, die im August diesen Jahres veröffentlicht wurde.

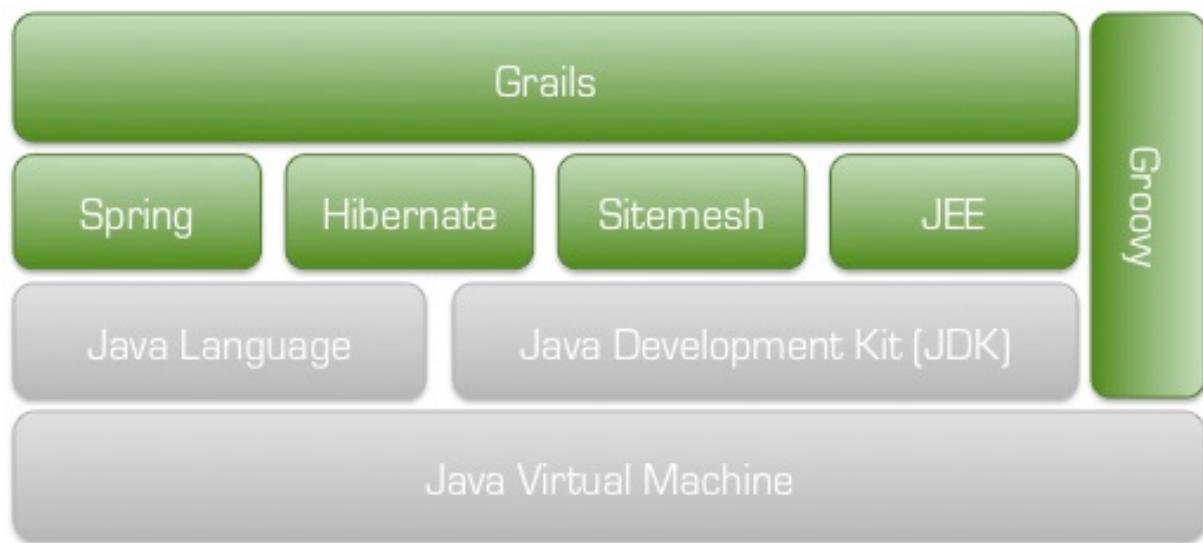


Abb. 2: Grails Aufbau

Konzept hinter Grails

Grails basiert auf verschiedenen Prinzipien und Konzepten, die es dem Framework ermöglichen Entwicklungszeit einzusparen und Code zu vereinfachen.

Der wesentliche Grundsatz von Grails ist es, Konventionen vor die Konfiguration (Convention over Configuration) zu stellen. Damit wird der Konfigurationsaufwand deutlich reduziert aber nicht ersetzt. Weitere Konzepte sind zum einen das Scaffolding, welches für die Generierung von CRUD-Schnittstellen (Create, Read, Update, Delete) für vorhandene Domainklassen verantwortlich ist. Dies kann dynamisch zur Laufzeit aber auch statisch durch manuelles Generieren der Controller und Views geschehen. Zum anderen GORM (Groovy objektrelationales Mapping), welches Grails seine Einfachheit und seine Power verleiht. Es basiert auf der Hibernate Bibliothek und verknüpft die Objekte der Domainklassen mit den Tabellen einer relationalen Datenbank.

Plugins sind die Features, welche Grails so flexibel und umfangreich machen. Denn 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.

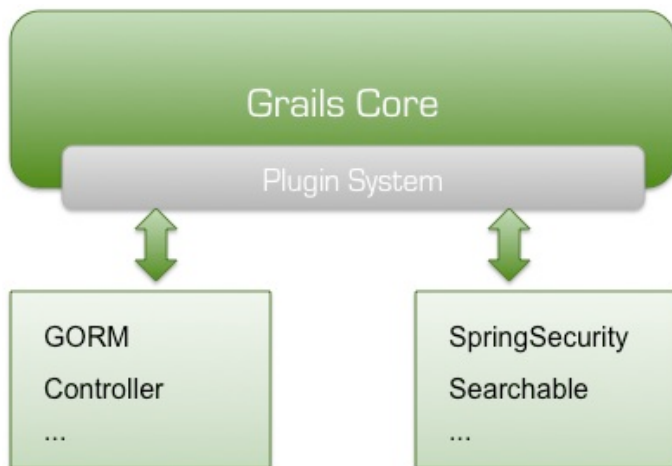


Abb. 3: Plugin Konzept

Im Zusammenhang mit dem Aufbau der Community war die Nutzung verschiedenster Plugins sehr hilfreich, da mit ihnen viele grundlegende Anforderungen, wie die Authentifizierung (auch Open ID), die Suche oder der Upload von Bildern, in kürzester Zeit umgesetzt werden konnte. Zudem ist die Auswahl an Plugins relativ groß. Die Liste auf der Grails Homepage umfasst in etwa 130 Plugins.

Der Kern von Grails

Der Grundaufbau von Grails basiert auf dem MVC-Pattern (Model-View-Controller), das in Grails mit Domainklassen, Controllerklassen und Groovy Server Pages (GSP) umgesetzt wird. Die Domainklassen (Model) sind der Kern der Anwendung und bilden die Kompetenz des Geschäftsmodells ab. Domainklassen sind persistent und entsprechen einer physischen Tabelle in der Datenbank. Die Zuordnung der Objekte wird durch das bereits erwähnte GORM durchgeführt.

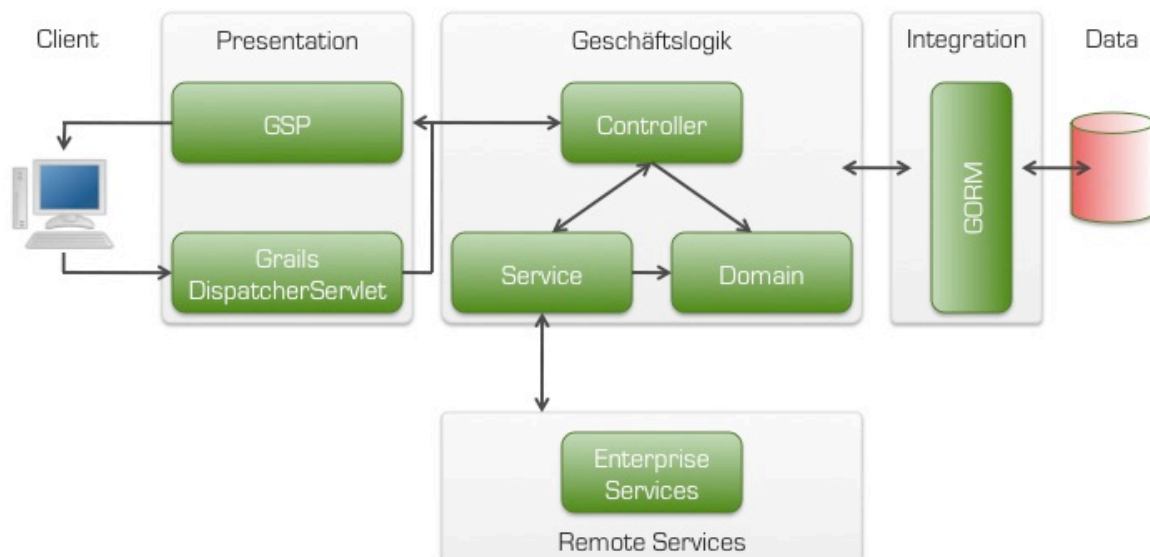


Abb. 4: Aufbau einer Grails Anwendung

Die Controller entsprechen der Controllersicht und sind der Einstiegspunkt und das Steuerelement der Anwendung, da ein Controller für die Beantwortung und Weiterleitung aller eingehenden Requests verantwortlich ist. Die View wird durch die Groovy Server Pages (GSP) abgebildet. Diese entsprechen den bekannten JSP-Seiten nutzen aber als Scriptletsprache Groovy, die sie allerdings ebenso flexibel macht.

Zusammenfassung

In Bezug auf die Erfahrung bei der Entwicklung der Community lässt sich sagen, dass Grails in der Tat eine neue Ära der Web-Anwendungsentwicklung eingeläutet hat. Schon nach kürzester Zeit sind auch für unerfahrene Entwickler erste Ergebnisse sichtbar. All dies ist möglich, weil in Grails Konvention vor Konfiguration gestellt werden und durch Scaffolding das einfache aber wichtige CRUD-Grundgerüst aufgebaut wird. Vergleicht man den Entwicklungsaufwand mit Grails mit dem Vorhaben eine Webanwendung in Java zu implementieren, erkennt man, dass jedem Entwickler enorm viel Zeit und Aufwand bei der Entwicklung von Webanwendungen abgenommen wird.

Kontaktadresse:

Nicole Hoock

esentri consulting GmbH
Stephanienstr. 36
D-76133 Karlsruhe

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