



ANWENDUNGSARCHITEKTUR

1 FRAMEWORK

Für das Projekt wurde eine Anwendungsinfrastruktur gefordert, die in Java umgesetzt sein sollte, da das Know-how des Projektteams zu überwiegenden teilen im Bereich der Java-Entwicklung liegt. Außerdem sollte das Framework maximale Unterstützung bieten eine solche Webanwendung in einen kurzen Zeitraum zu realisieren.

Spring Roo schien zur Umsetzung am meisten geeignet, da es eine umfassende Unterstützung durch Generatoren viele allgemeine Aufgaben den Entwicklern abnimmt (z.B. CRUD-Prozesse), so dass sich das Projektteam vermehrt mit Problemstellungen der Daten- und Prozessmodellierung beschäftigen kann.

2 KURZBESCHREIBUNG: SPRING ROO

Spring Roo ist eine Erweiterung des Spring Frameworks und konzentriert sich auf das unterstützen von „rapid-development“ Entwicklungsprozessen durch Umsetzung der Prinzipien „Dont-Repeat-Yourself“ (DRY) und „Convention over Configuration“. Spring Roo ähnelt daher anderen Web-Frameworks wie „Ruby on Rails“ oder „Django“ für Python. Spring Roo stellt keine gesonderte Laufzeitumgebung bereit, sondern unterstützt bei der Generierung von Sourcen, aber auch der Klassen zum Compile-Zeitpunkt. Über eine Eclipse Umgebung mit Roo-Addons oder eine laufende Roo-Shell kann Source-Code editiert werden. Roo beobachtet die Projekt-Dateien und modifiziert diese automatisch abhängig von den Aktionen des Entwicklers. Zur Identifikation der von Roo „überwachten“ Dateien werden @Roo-Annotations genutzt. Diese Annotationen werden vor dem Kompilieren aus dem Quelltext entfernt (Source Retention Policy). Entsprechend steht sie zur Laufzeit eines Programms nicht zur Verfügung.¹

¹ Vgl. Shekhar Gulati: Introducing Spring Roo. <http://www.ibm.com/developerworks/library/os-springroo1/>, Abgerufen 23.04.2012

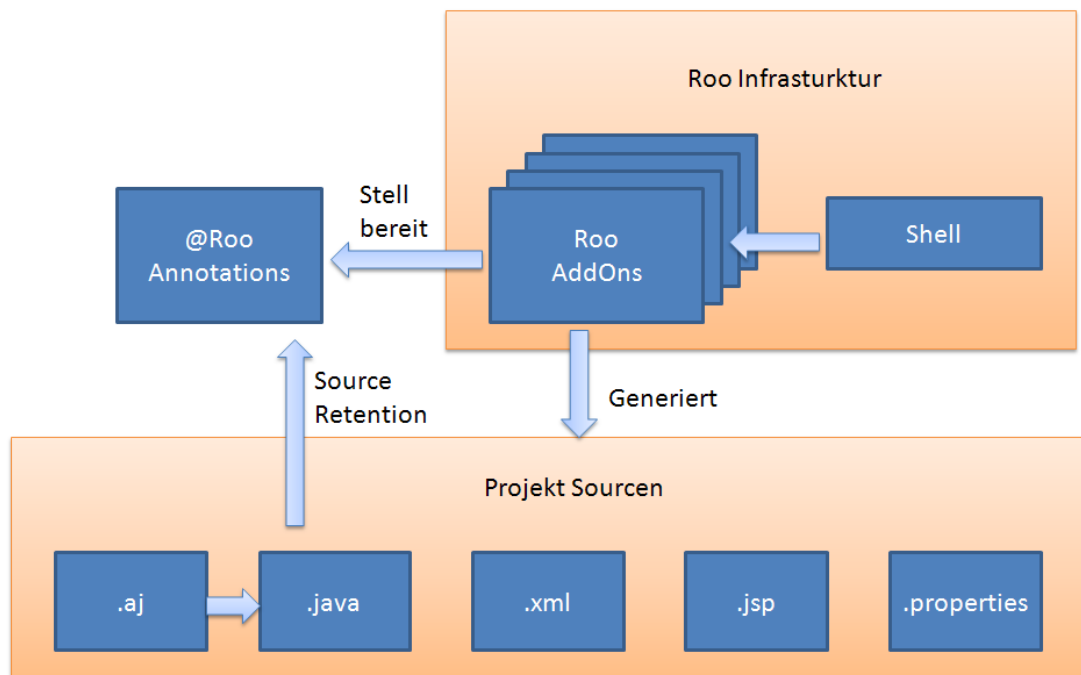


ABBILDUNG 1 SPRING ROO FRAMEWORK²

Die daraus resultierenden Applikationen nutzen im Allgemeinen:

- eine relationale Datenbank im Backend, angesprochen über die Java Persistence API (JPA),
- das Spring Framework und Transaktionsmanagement
- Aspektorientierung (aspectJ)
- Build-Konfiguration durch Maven
- ein MVC basiertes Frontend (Spring MVC), welches JSPs als Views verwendet.

Mehr Informationen zu Spring Roo unter: <http://www.springsource.org/spring-roo>.

² Eigene Darstellung basierend auf Schmidt, S. (2011): デブサミ 2011 レポート Java 開発者向け RAD 「Spring Roo」 ～Spring Framework の展望, S. 8, <http://codezine.jp/article/detail/5813>, Abgerufen 23.04.2012.

3 HARDWARE-ARCHITEKTUR

Das Zielsystem besteht aus einer Client-Serverarchitektur die vorwiegend über den Browser der End-nutzer aufgerufen wird. Die Hardware besteht auf der Server-Seite aus folgenden Komponenten:

- Ein J2EE-Anwendungsserver
- Eine relationales-Datenbanksystem

Ein separater Webserver (Apache Webserver) wie er häufig in Produktionsszenarien vor dem Tomcat verwendet wird, wird für dieses Projekt nicht betrachtet/verwendet.

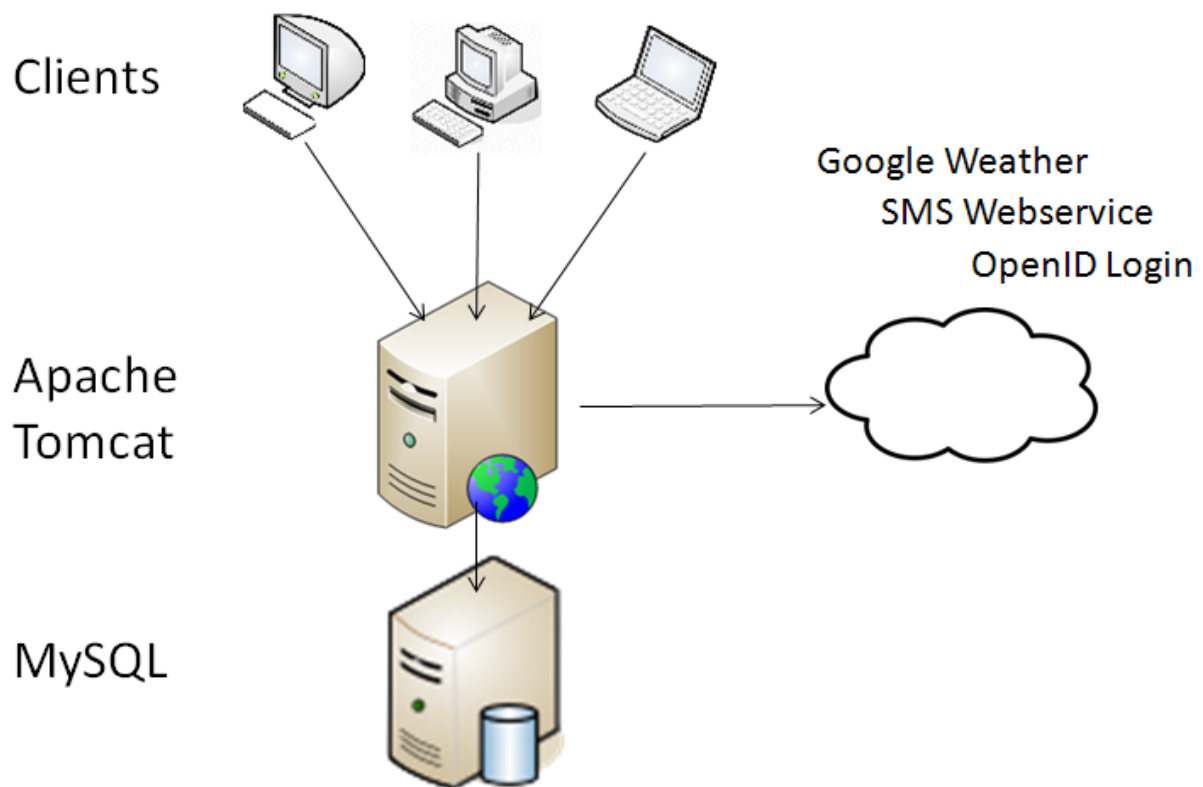


ABBILDUNG 2: HARDWARESICHT DES SYSTEMS³

³ Die „InternetWolke“ zwischen Clients und Webserver/Anwendungsserver wurde vorsätzlich nicht in die Darstellung aufgenommen.
Eigene Darstellung.

4 SOFTWARE-ARCHITEKTUR

Die Softwarearchitektur ist durch die Roo-Unterstützung stark vorgegeben. Grundsätzlich kann gesagt werden, dass es sich um eine klassische 4-Schichtarchitektur

Die Anwendungslogik orientiert sich am Model-View-Controller-Pattern. Entitäten liegen als „JPA Active Record“ im Entity Layer oder Persistence Layer. Über die JPA-Annotationen stellt das Framework die Kommunikation zur Datenbank bereit. Benutzerinteraktion wird durch Views bereitgestellt. Diese liegen in Form von JSPs vor.⁴

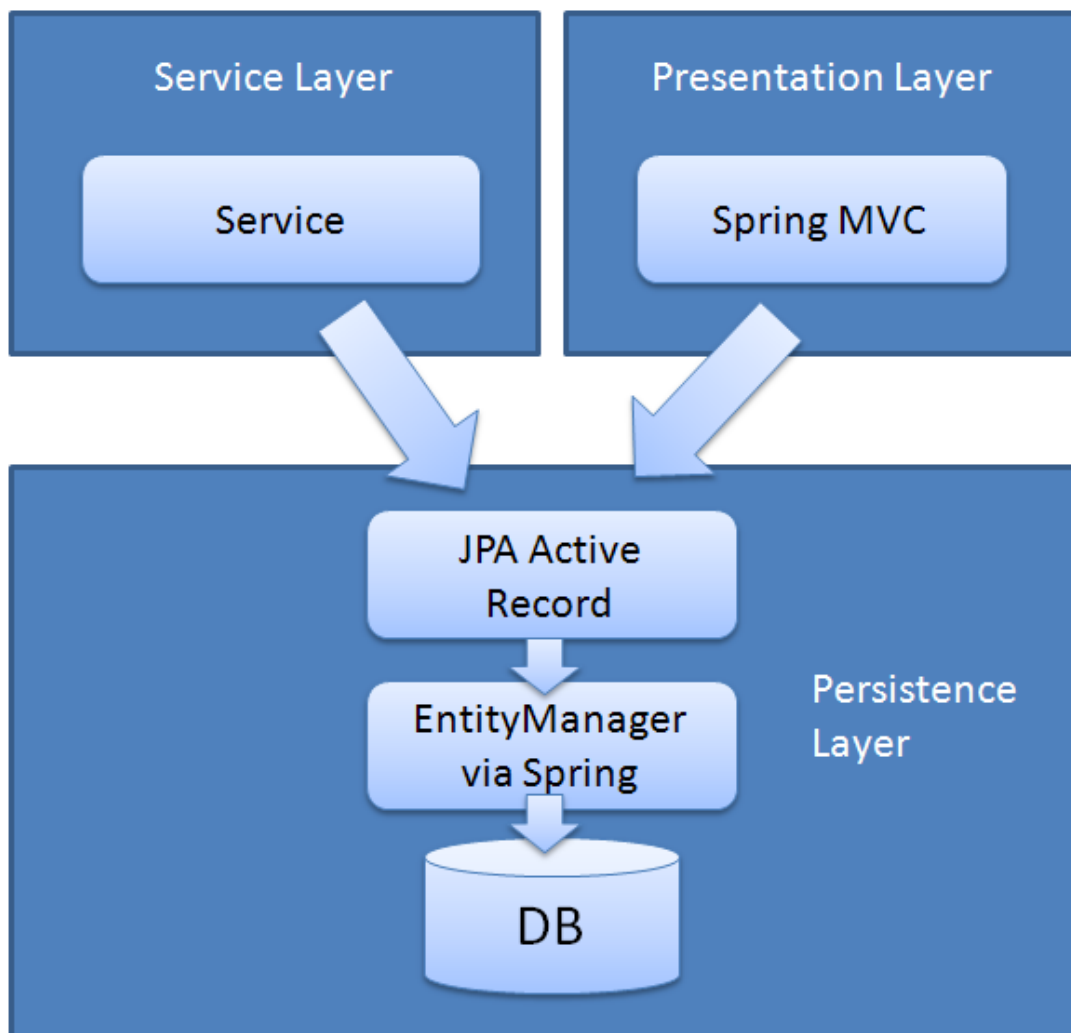


ABBILDUNG 3 SPRING ROO SOFTWAREARCHITEKTUR⁵

⁴ Vgl. o.V. (2012) Spring Framework – Chapter 10. Application Layering, <http://static.springsource.org/spring-roo/reference/html/base-layers.html>, abgerufen am 23.04.2012.

⁵ Eigene Darstellung in Anlehnung an o.V. (2012).