



SpringSource Enterprise & Application Platform: Wo geht die Reise hin?

Eberhard Wolff

Regional Director & Principal Consultant
SpringSource

- SpringSource Enterprise
 - Application Management Suite
 - SpringSource Tool Suite
 - Oracle Advanced Pack

- SpringSource Application Platform:
OSGi im Enterprise



SpringSource Enterprise

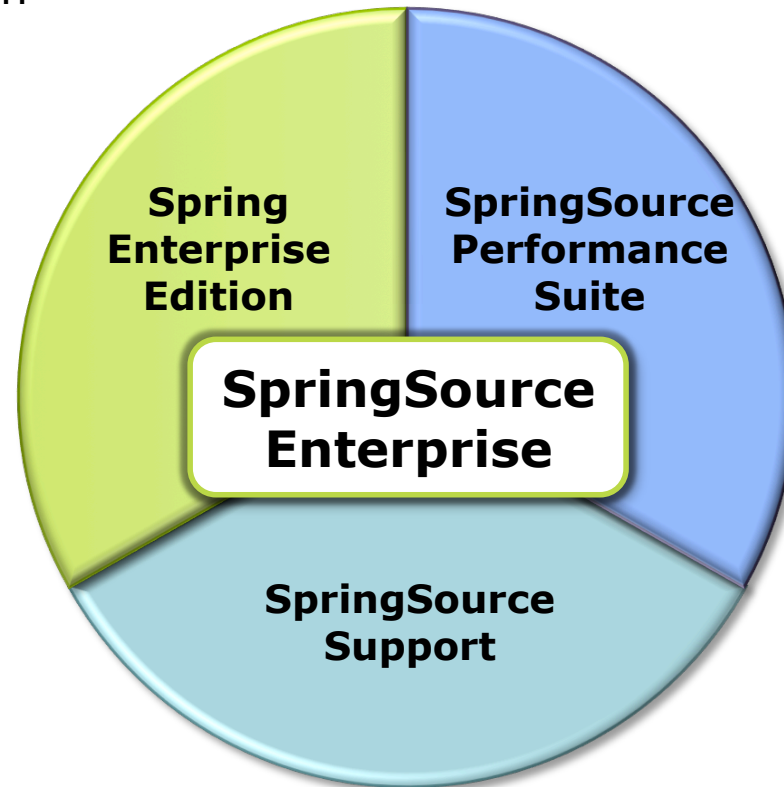
- SpringSource ist erfolgreich mit Trainings- und Consulting-Dienstleistungen
- Aber wir können nicht überall sein
- Können wir den Umgang mit Spring noch anders verbessern?
- Kommerzielles Produkt: SpringSource Enterprise

SpringSource Enterprise



Spring Enterprise Edition

- SpringSource zertifiziert
- Instrumentiert für Management



SpringSource

- Performance Suite
- SpringSource Application Management Suite
- SpringSource Tool Suite
- SpringSource Advanced Pack for Oracle Database

SpringSource Support

- Produktions- und Entwickler-Support

SpringSource Application Management Suite (AMS)



- Management von Spring-Anwendungen
- Relevante Daten automatisch verfügbar
 - Transaktions-Raten
 - Fehler/Exceptions
 - View-Rendering
 - Methoden-Ausführungszeiten
- Werden über JMX mit Integrationsmöglichkeiten in viele Tools exportiert
- ...oder über Hyperic in einer Web-Management-Oberfläche verfügbar

Zwei Worte über Hyperic



- Web-basierte Management Lösung
- Sehr einfaches Set-Up:
- Zu managender Computer benötigt nur einen Agent
- Rest ist Auto-Discovery
- Überwacht
 - Betriebssystem
 - Tomcat, WebSphere, WebLogic...
 - Spring-Anwendungen



SpringSource Application Management Suite (AMS): Demo

SpringSource Tool Suite (STS)



- Basiert auf Eclipse
- Integration der „üblichen“ Enterprise Java Plug Ins
 - Spring IDE
 - AspectJ Development Toolkit
 - Data Tools Platform
 - J2EE Standard Tools
 - Web Standard Tools
- Basiert auf Mylyn (task-basierte IDE)

SpringSource Tool Suite (STS)



- Eigene Features:
 - Task-basierte Tutorial für verschiedene Spring-Technologien
 - Spring Best Practises
 - Runtime Error Analysis



SpringSource Tool Suite (STS) Live Demo

SpringSource Advanced Pack for Oracle



- Oracle Real Application Cluster:
Ausfallsichere Datenbank
- Aber: Bei einem Ausfall geht die aktuelle
Transaktion verloren
- ...nicht mehr mit dem Advanced Pack
for Oracle und Spring JDBC
- Außerdem Unterstützung für Oracle
DataSources, Oracle Data Types,
Advanced Queueing (AQ) ...

- Evaluations-Lizenzen unter <http://springsource.com/products/enterprise/performancesuite>



SpringSource Application Platform: OSGi für's Enterprise

Was ist OSGi?

1. Versuch



- Es gibt ein Tutorial in STS...

Was ist OSGi?

2. Versuch



OSGi™

- The Dynamic Module System for Java

- Partitioniert ein System in Module („Bundles“), die aus Packages bestehen
- Bundles importieren und exportieren Packages
- Packages können versioniert sein
- OSGi löst die Abhängigkeiten auf
- Striktes Management der Sichtbarkeiten von Packages zwischen Modulen

- Module können *zur Laufzeit*:
 - gestartet
 - gestoppt
 - installiert
 - deinstalliert
 - und geupdatet werden

Und service-orientiert ist es auch...



- Bundles können Services anbieten
 - Dynamisch zur Laufzeit
- Durch die Service Registry kann man Services suchen
 - ... und sich an sie binden
- Services können zur Laufzeit kommen und gehen

Woher kommt es?



- Standardisiert von der OSGi Alliance
 - <http://www.osgi.org/>
- Schon immer leichtgewichtig und dynamisch
 - 1999 Fokus auf embedded
 - 2003 mobile
 - 2006 Beginn von OSGi in Enterprise -Anwendungen
- Und natürlich: Basis für Eclipse
- Und zunehmend in Application Servern

Was sagt der Markt?



- JavaMagazin QuickVote

OSGi: Die neue Basistechnologie für Java-Anwendungen?

Habe OSGi noch nie verwendet. (14%)



OSGi wird keinen großen Einfluss haben. (5%)



Werde mir OSGi in nächster Zeit anschauen. (51%)



Setze OSGi bereits großflächig ein. (30%)



Teilnehmer: 152

- Grundlegende Einheit für Deployment und Modularisierung
- Ein JAR-File
- Mit zusätzlichen Einträgen in META-INF/MANIFEST.MF, z.B.:
 - Bundle-SymbolicName
 - Bundle-Version
 - Bundle-Name
 - Bundle-ManifestVersion
 - Bundle-Vendor

Import-Package

- Importiert Packages aus anderen Bundles

```
Import-Package: com.xyz.foo;
```

```
Import-Package: com.xyz.foo;version="1.0.3"
```

```
Import-Package:
```

```
    com.xyz.foo;version="[1.0.3,1.1.0)"
```

```
Import-Package:
```

```
    com.xyz.foo;version="[1.0.3,1.1.0)",
```

```
    com.xyz.bar;version="[1.0.3,2.0.0)"
```

Export-Package

- Definiert von anderen Bundles importierbare Typen

```
Export-Package: com.xyz.foo
```

```
Export-Package: com.xyz.foo;version="1.0.5"
```

```
Export-Package: com.xyz.foo;version="1.0.5",  
                com.xyz.bar;version="2.0.0"
```

```
Export-Package: com.xyz.foo;  
                com.xyz.bar;version="1.4.1"
```

```
Export-Package:  
    com.xyz.foo;version="1.0.3";uses:="org.bar"
```

- Man stelle sich das für das Spring-Framework vor....
- Dafür gibt es aber Lösungen

Service Registry

- Importiert und Exportiert *Objekte* keine *Typen*
- Export

```
ServiceRegistration reg =  
    bundleContext.registerService(  
        Bar.class.getName(), null);  
  
...  
reg.unregister();
```

- Import

```
ServiceReference ref =  
    bundleContext.getServiceReference(  
        Bar.class.getName());
```

Service Registry



- Bind

```
Bar bar = (Bar)  
    bundleContext.getService(ref);
```

```
...
```

```
bundleContext.ungetService(ref);
```

```
// bar sollte nicht mehr verwendet werden
```

- Das sollte natürlich alles irgendwie einfacher gehen

Vorteil: Modularität



- Ein Bundle ist eine Black Box
- Nur explizit exportierte Klassen sind außerhalb sichtbar
- Verhindert nicht beabsichtigte Abhängigkeiten zwischen Bundles
- Ermöglicht unabhängige Entwicklung
- Schneller Compile-Run-Zyklen: Nur einzelne Bundles müssen erneuert werden

Vorteil: Bessere Administration



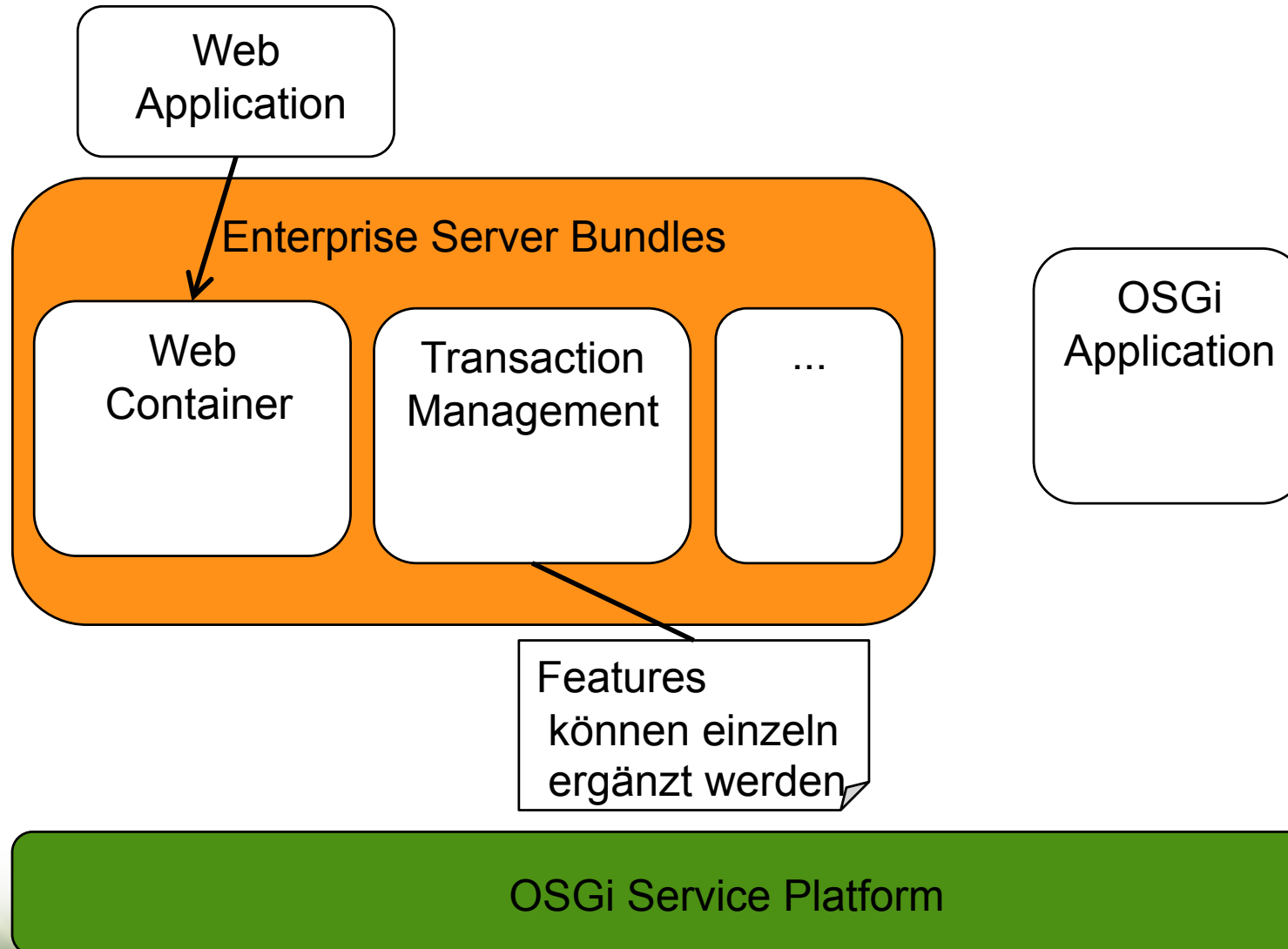
- Bundles sind auch für Administratoren sichtbar
- Ohne OSGi sehen sie nur eine große Anwendung

- Welche Bundles sind installiert?
- Bundles installieren und deinstallieren
- Bundles aktivieren und deaktivieren
- Bundles updaten
- Services administrieren

- *Weder Anwendung noch Server muss neu gestartet werden*

-
- OSGi auf dem Server
 - Application Design
 - Enterprise Libraries mit OSGi verwenden

OSGi als Server Plattform: Build your own AppServer



- Applikationen werden eine Sammlung von Bundles
 - Vertikal: Funktionalitäten
 - Horizontal: Klassische Schichten (Logik, Persistenz)
- Bundles bieten Services an und verwenden Service
- Services definieren also die Schnittstellen der Bundles

Warum machen wir das nicht einfach?



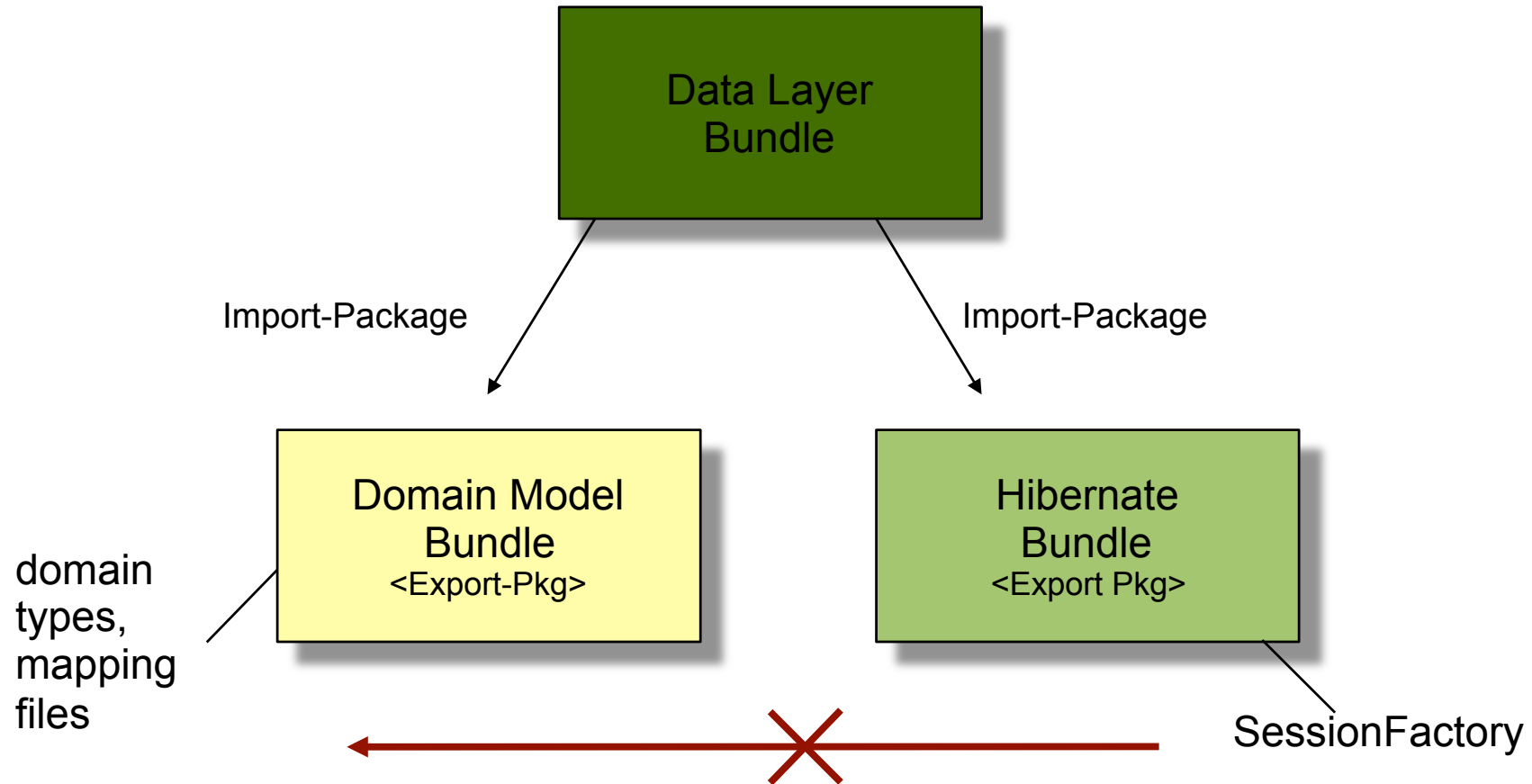
- Dynamik der Services ist komplex zu programmieren
- Basis mit OSGi und z.B. Web fehlt
- Was macht man mit WARs?
- Viele Enterprise Libraries kommen mit OSGi nicht zurecht

Enterprise Libraries unter OSGi



- Probleme beim Laden von Classes und Resources-Sichtbarkeit
 - Context Class Loader ist nicht definiert
 - Ressourcen in META-INF
 - ...
- SpringSource Enterprise Bundle Repository enthält angepasste Versionen
- Z.B. kann man JMS mit ActiveMQ nachrüsten
- Einige Bibliotheken kommen mit Sichtbarkeit konzeptbedingt nicht klar

Example: Class visibility



SpringSource Application Platform



*"The Healthy New Way
to Run Your Apps"*

SpringSource Application Platform



- Der einzige Application Server mit OSGi als Programmier-Modell
- Open Source Application Server (GPL)
- Basiert auf Equinox (Eclipse OSGi)
- Basiert auf Spring Dynamic Modules for the OSGi Platform
- Enthält Tomcat für Web
- Und andere Nettigkeiten (z.B. Libraries, um Imports einfacher zu machen)

SpringSource Application Platform



- Dynamik der Services ist komplex zu programmieren
 - Spring Beans werden deklarativ zu OSGi Services
- Basis mit OSGi und z.B. Web
 - Equinox + Tomcat
- Was macht man mit WARs?
 - Migrationspfad
- Viele Enterprise Libraries kommen mit OSGi nicht zurecht
 - PARs definieren Sichtbarkeiten

Zwei Worte über PARs



- Platform Archive
- OSGi kennt keine Anwendungen
- PARs limitieren...
 - ...die Sichtbarkeit von Services und Typen, so dass nicht unbeabsichtigt außerhalb der Anwendung genutzt werden können.
- Außerdem löst sich so Load Time Weaving / Hibernate



SpringSource Application Platform: Demo

- Rund um Spring entsteht vieles Neues!
- Mit AMS kann man nicht nur Spring-Anwendungen administrieren
- STS hilft bei der Entwicklung
- Oracle Advanced Pack mit Oracle
- ...und AP ist ein Application Server der nächsten Generation
- Die einzige OSGi-Enterprise-Umgebung

In eigener Sache: Trainings



- 16.-19.09.08 Core Spring Hamburg
- 21.-24.10.08 Core Spring Zürich
JUGS Rabatt!
- 18.-21.11.08 Core Spring Stuttgart
- 25.-28.11.08 Core Spring Wien
- 02.-05.12.08 Core Spring München

- 28.-30.10.08 Rich Web Applications München

- 04.-06.11.2008 Spring.Net München
- Siehe <http://springsource.com/de>