

# **OSGi: Anwendungsszenarien, Auswahlkriterien und Ausblick**

**Thementag OSGi**

**03.11.2009**

**Autor:**

**Christoph Schmidt-Casdorff**

# Agenda

- **Wo wird OSGi derzeit eingesetzt?**
- **Grundsätzliche Anwendungsszenarien**
- **OSGi Status Quo und Ausblicke**
- **Zukunft von OSGi**
- **Modularisierung im JCP**

# Agenda

- **Wo wird OSGi derzeit eingesetzt?**
- Grundsätzliche Anwendungsszenarien
- OSGi Status Quo und Ausblicke
- Zukunft von OSGi
- Modularisierung im Java

# OSGi Rekapitulation

## ● Modularisierung

- Statische Sicht auf die Auslieferungseinheiten
- Reduziert und definiert die Abhängigkeiten der Module

## ● Life Cycle Layer

- Module können zur Laufzeit de-/installiert werden
- Dynamische Funktionalität
- Selbstheilung des Systems

## ● Services

- Korrespondenz der Funktionalität via Service-Registry
- SOA light

## Wo wird OSGi derzeit eingesetzt?

### ● Application Server

- WebSphere, JonAS, Glassfish
  - setzen auf OSGi
- JBoss, BEA
  - werden den Weg gehen
- Siehe [http://www.infoq.com/news/2008/02/osgi\\_je](http://www.infoq.com/news/2008/02/osgi_je)

### ● Enterprise Service Bus

- Apache ServiceMix 4.x
  - Siehe <http://architects.dzone.com/articles/fuse-esb-4-osgi-based>

### ● Frameworks

- Spring, Guice

## Wo wird OSGi derzeit eingesetzt?

### ● Eclipse IDE

- Eclipse Plugins sind OSGi-Bundles (seit 3.x)
- Eclipse Equinox Project ist ein Open Source OSGi Framework
- Integration in Laufzeitumgebung via Export/Import
- Dynamische De-/Installation

### ● LinkedIn (<http://www.linkedin.com/>)

- Soziales Netzwerk für Führungs- und Fachkräfte
- [http://raibledesigns.com/rd/entry/building\\_linkedin\\_s\\_next\\_generation](http://raibledesigns.com/rd/entry/building_linkedin_s_next_generation)

# Agenda

- Wo wird OSGi derzeit eingesetzt?
- **Grundsätzliche Anwendungsszenarien**
- OSGi Status Quo und Ausblicke
- Zukunft von OSGi
- Modularisierung im JCP

## Einsatzszenarien

- **Large Scale Application**
  - Anwendung besteht aus sehr vielen funktionalen Einheiten
    - OSGi ermöglicht Management dieser Menge
- **Funktionalität kann kommen und gehen (dynamische Funktionalität)**
  - Dynamisches Monitoring / Profiling
  - Businessstatistiken / -auswertungen
- **Funktionalität koexistiert in unterschiedlichen Versionen**
  - Verarbeitungsvarianten je nach Eingabe
    - z.B. unterschiedlichen rechtlichen Grundlagen je nach Eingang
- **Teile der Businesslogik können zeitweise ausfallen**
  - Nicht-primäre Funktionalität fällt zeitweise aus



## Einsatzszenarien II

### ● 7\*24 Verfügbarkeit

- Neue Versionen/Bugfixes können ohne *down time* eingespielt werden

### ● Anwendung mit Beiträgen aus unterschiedlichen Quellen

- Standortübergreifende Entwicklung
- Integration von *third party* Software
- Gesamtplattform mit unterschiedlichen Systemen
  - Jedes System hat einen eigenen Lebenszyklus

### ● Anwendung wird in unterschiedlichen Profilen/Varianten ausgeliefert

- Jede Variante wird durch spezielle Funktionalität beschrieben
- Siehe Eclipse
- Software von der Stange

# Agenda

- Wo wird OSGi derzeit eingesetzt?
- Grundsätzliche Anwendungsszenarien
- **OSGi Status Quo und Ausblicke**
- Zukunft von OSGi
- Modularisierung im JCP

## OSGi –Status Quo

- **Modularisierung unter Java**
- **Dynamik von Modulen**
- **SOA light**
- **Betrieb von großen und / oder dynamischen Anwendungen**

## OSGi – Status Quo

- **Noch nicht ausgereifte Unterstützung der Entwicklungswerkzeuge**
  - IDE, Build, Testing
  - Entwickeln sich aber mit Hochdruck
  - Die Community weiß um das Problem
    - Open Source wird es richten

## OSGi –Status Quo

### ● Programmiermodell

- Programmiermodell ist komplex
- *Service Component Models* wie Spring DM werden Bestandteil der Spezifikation
- Ansätze, Komponentenmodelle auf eine höhere Sprachebene bringen
  - Domänenspezifische Sprache für OSGi
  - Siehe <http://romanroe.blogspot.com/2008/03/dsl-for-osgi.html>

## OSGi –Status Quo

- **Integrationstest**
  - Verfahren für Integrationstest ist komplex
  - Werkzeugunterstützung ist noch nicht ausgereift

## OSGi –Status Quo

- **OSGi-fizierung bestehender Frameworks**
  - Viele bestehende Frameworks sind nicht in OSGi einzusetzen
    - Keine Manifest-Datei
    - Nicht-konformer Einsatz des Classloaders
    - Log4J, Hibernate
  - Alle wichtigen Frameworks arbeiten daran
  - Spring Source OSGi-fiziert wichtige Frameworks

# OSGi –Status Quo

## ● Betrieb unter OSGi

- *Provisioning*
  - Wie bekomme ich meine Bundles in das Zielsystem
    - Werkzeugunterstützung ist noch nicht ausgereift
    - Plattformen wie Spring DM Server werden dieses Problem in naher Zukunft entschärfen
  
- Wie gewährleiste ich einen konsistenten Abnahmeprozess
  
- Betrieb
  - Neue Technologie erfordert neue Kenntnisse



## OSGi – Status Quo

- **OSGi ist Hype in der Javawelt, aber ...**
- **Es fehlt an ausreichender Werkzeugunterstützung**
  - Kommt kurz- bis mittelfristig
- **Programmiermodell ist komplex**
  - Wird kurz- bis mittelfristig entschärft
- **Reine OSGi Frameworks sind komplex zu betreiben**
  - Kommerzielle Plattformen adressieren dieses Problem
    - Spring DM Server
- **JCP hat sich derzeit nicht auf OSGi vollständig eingelassen**
  - Jigsaw als konkurrierendes Modulsystem

## OSGi Ausblicke

- **OSGi Spec. 4.2 liefert**

- Integration weiterer JEE Standards
  - JNDI, JTA, ....
- Remote OSGi
  - Verteiltes OSGi über mehrere Knoten

- **Werkzeugumgebung bewegt sich**

- Verbesserungen in Eclipse 3.5.x
- Werkzeug-Familie PAX entwickelt sich aktiv

# Agenda

- Wo wird OSGi derzeit eingesetzt?
- Grundsätzliche Anwendungsszenarien
- OSGi Status Quo und Ausblicke
- **Zukunft von OSGi**
- Modularisierung im JCP

## Zukunft von OSGi

- **Wird OSGi ein zweites CORBA ?**
  - Eine schwergewichtige Technologie, die sich überlebt hat
- **OSGi ist reifer**
  - Hat daher bessere Überlebenschancen
- **OSGi hat einflussreiche Unterstützer**
  - WebServer Anbieter
  - Apache, Eclipse, ...

## Zukunft von OSGi

- **... aber OSGi-Tooling muss seine Mängel möglichst schnell beheben**
  - Insbesondere in der Infrastruktur
- **... dann hat es aus unserer Sicht langfristige Perspektiven**
  - Was immer langfristig in der heutigen IT bedeutet
- **... aber die Welt braucht OSGi**
  - Modularisierung und Dynamik

# Agenda

- Wo wird OSGi derzeit eingesetzt?
- Grundsätzliche Anwendungsszenarien
- OSGi Status Quo und Ausblicke
- Zukunft von OSGi
- **Modularisierung im JCP**

## Modularisierung in JCP

- **Java Community Process definiert, wie Standards in Java entstehen**

  - Diese werden als Java Specification Request eingereicht
  - Sun hat die Hoheit über diesen Prozess

- **Modularisierung im JCP**
  - JSR 291 definiert ein Komponentenmodell für Java SE
  - JSR 232 definiert ein Komponentenmodell für Java ME
  - Beide spiegeln im wesentlichen OSGi wieder

## JSR 294

### ● ... aber

- JSR 294 definiert Spracherweiterungen für eine Modularisierung
- Modulsysteme setzen auf JSR 294 auf
- Siehe <http://jcp.org/en/jsr/detail?id=294>

### ● Spracherweiterungen umfassen

- *Superpackages* fassen *packages*, *superpackages* zusammen
- Definieren Sichtbarkeiten
- Statisches Sprachmittel
- JDK 7 ist Referenzimplementierung



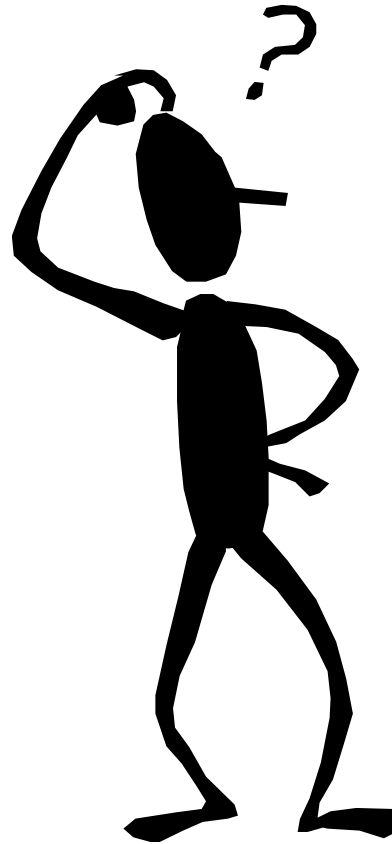
## JSR 294 und Jigsaw

- **OpenJDK implementiert ein Modulsystem über JSR 294**
  - Project Jigsaw
  - Konkurrenz zu OSGi
  - OSGi ist ebenfalls bestrebt, auf JSR 294 aufzusetzen
  - OSGi ist bewährt, Jigsaw noch nicht fertig
  - Warum ein weiteres Modulsystem? -> offensichtlich politisches Minenfeld
  
  - Siehe
    - <http://java.dzone.com/news/java-modularity-osgi-and>
    - <http://www.osgi.org/blog/2008/12/project-jigsaw-ii.html>

## JSR 294 - Beispiel

```
superpackage A {  
    member package P;  
    member superpackage B;  
}  
superpackage B member A {  
    member superpackage C;  
    export superpackage C;  
}  
superpackage C member B {  
    member package Q;  
    export Q.X;  
}
```

# Fragen?



[www.iks-gmbh.com](http://www.iks-gmbh.com)