

# Deutsche Post - Renten Service –

Praxisbeispiel: OSGi im Massendruck

# Praxisbeispiel: OSGi im Massendruck

OSGi Thementag

**03.11.2009**

**Autor:**

Alexander Wolff (Deutsche Post)

## Agenda

- **Deutsche Post - NL Rentenservice**
- **OMS – Output Management System**
- **OMS Architekturprinzipien**
- **OMS und OSGi**
- **Projektstand und –bewertung**
- **Zusammenfassung**

## Agenda

- **Deutsche Post - NL Rentenservice**
- OMS – Output Management System
- OMS Architekturprinzipien
- OMS und OSGi
- Projektstand und –bewertung
- Zusammenfassung

## Deutsche Post NL Rentenservice

- 26 Millionen Konten von ca. 18 Millionen Rentnern
- 305 Millionen Rentenzahlungen pro Jahr
- 24 Millionen Rentenberechnungen pro Jahr
- 17 Millionen Rentenzahlungen ins Ausland pro Jahr
- Auszahlung von 195 Milliarden € pro Jahr

## Massendruck beim NL Rentenservice

### ■ Rentenanpassungsverfahren der gesetzlichen Rente

- Jährliche Anpassungsbescheide
- > 20 Millionen in 2 Wochen

### ■ Bestandsverwaltung in der gesetzlichen Rente

- Begrüßungsschreiben, Bestätigung von Adressänderungen etc.

### ■ Konteninformationen in der Zulagenverwaltung

- Gesetzliche Bescheide
- Kundenspezifische Schreiben
- > 1 Million pro Woche

## Agenda

- Deutsche Post - NL Rentenservice
- **OMS – Output Management System**
- OMS Architekturprinzipien
- OMS und OSGi
- Projektstand und –bewertung
- Zusammenfassung

## Zielsetzung

- Aufsetzen auf offenen Standards
- Gewährleistung der notwendigen Performance
  - Hohe Spitzenlast
- Geschäftslogik soll unter unterschiedlichen Technologien laufen
  - Batch (Massendruck), Web (Online-Druck)
- Dynamische Anpassung an unterschiedliche Druckanforderungen
- Ableitung an Druckdienstleister und Archivsystem
- Ableitung an Archivsystem



## OMS – Output Management System

### ■ Prinzipielles Verfahren

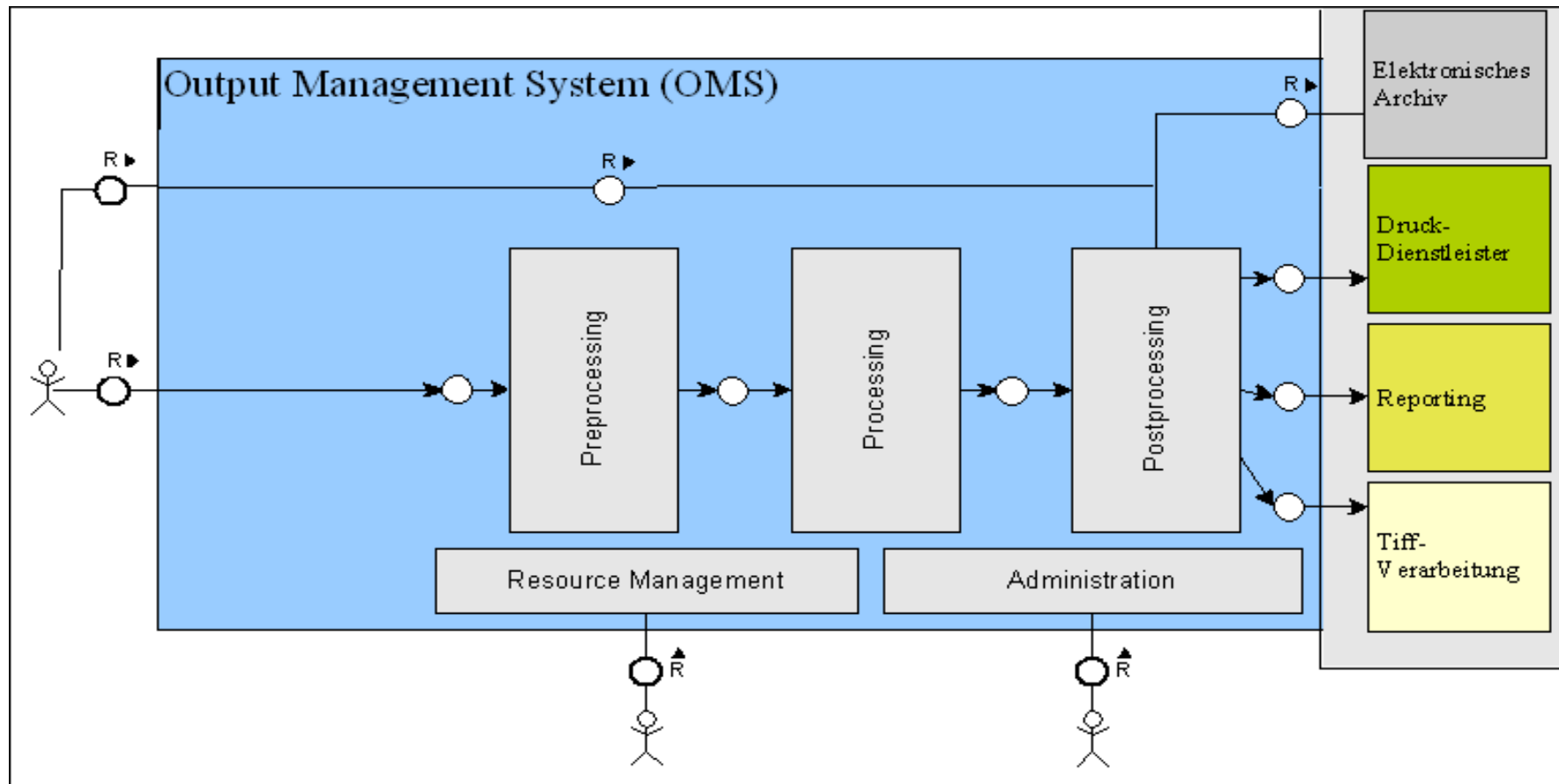
- Rohdaten werden als XML aufgeliefert
- Inhaltliche Anreicherung
  - Ersetzen von Textbausteinen
- Rendering in die gewünschten Formate
  - Gerendertes Zwischenformat
- Zusammenstellung der Sendung
- Zielformate erstellen
  - PDF, TIFF, AFP
- Ableitung an Druckdienstleister

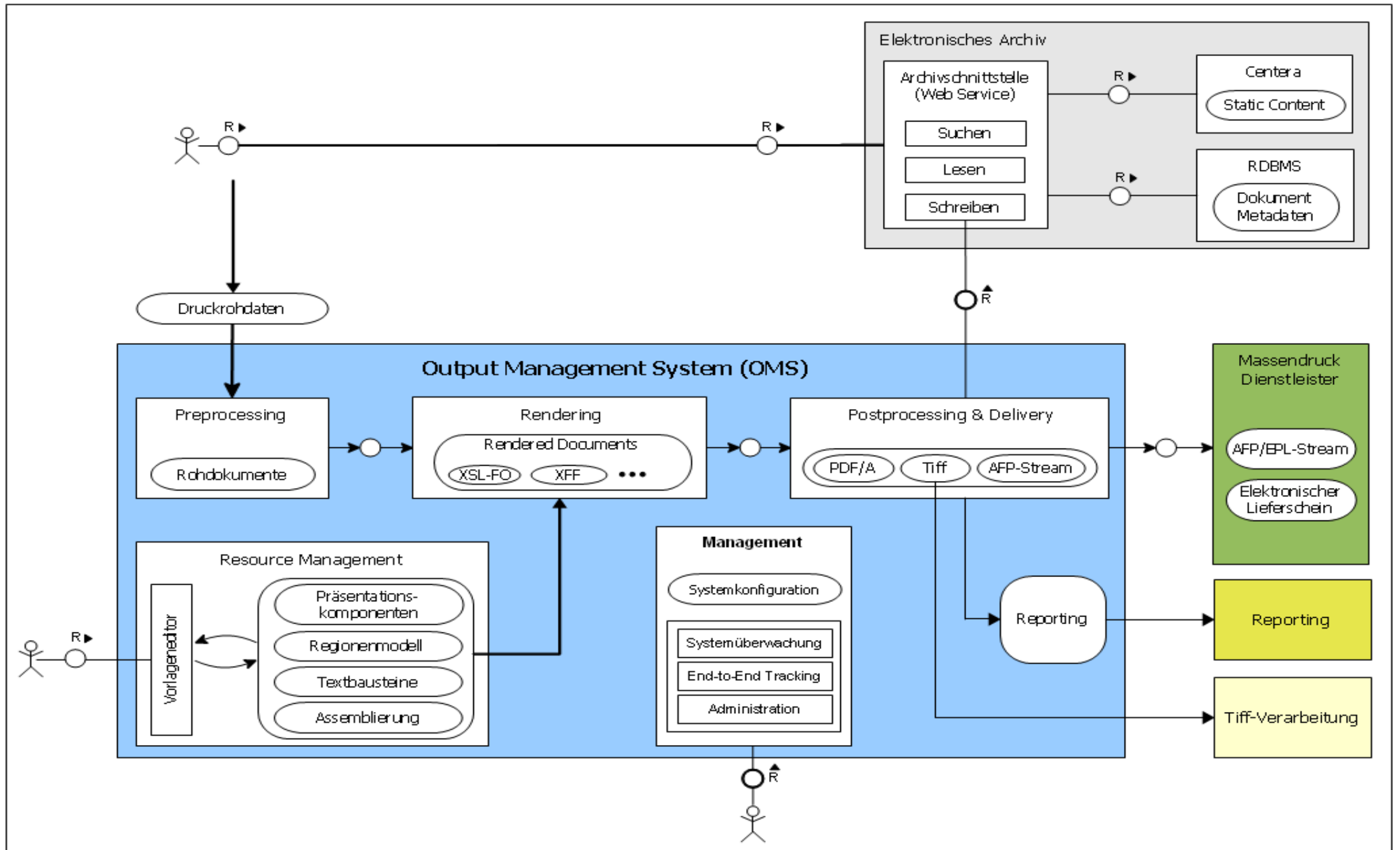
Preprocessing

Processing

Postprocessing

# OMS Systemkontext





## OMS – Parametrisierung

### ■ Dokumententyp / Druckvorfall

- Jedes Dokument wird anders behandelt
  - Anderes Layout, andere Textbausteinen etc.
- Hängt ab von Parametern wie Typ, Jahrgang, Version, ....
- Ein spezifisches Aufbereitungsverfahren heißt *Druckvorfall*

### ■ Woher bekommt OMS den Druckvorfall zu einem Dokument?

## OMS – Parametrisierung

### ■ Druckvorfall

- Wird einem Dokument aufgrund dessen Metadaten zugewiesen

### ■ Druckvorfall stellt folgende Informationen bereit:

- Textbausteine und ähnliche Parameter
- Sendungsregelwerk
- Rendering
- Ressourcen wie Schriften, Vorlagen, ...
- Ableitungsregeln

### ■ Es kann beliebig viele und versionierte Druckvorfälle geben

### ■ Druckvorfälle sind im OMS installiert

## OMS Konzept

- OMS ist Infrastruktur, auf der sich Druckvorfälle einschreiben können
- Alle Druckvorfälle werden nach den selben Prinzipien verarbeitet
- Das Verfahren wird durch Druckvorfälle parametrisiert
- Druckvorfälle schreiben sich dynamisch ein

## OMS – Infrastruktur

### ■ OMS Infrastruktur ist verantwortlich für

- Rendering gemäß *OpenSource* Standards
- Ressourcenverwaltung
- Administration
- Anbindung an externe Systeme
  - Archiv
  - Druckdienstleister
- End-to-End-Tracking
- Performance / Lastverteilung
- Parallelisierung der Verarbeitung

## Agenda

- Deutsche Post - NL Rentenservice
- OMS – Output Management System
- **OMS Architekturprinzipien**
- OMS und OSGi
- Projektstand und –bewertung
- Zusammenfassung



## OMS Architekturprinzipien

### ■ Plugin-Architektur (Druckvorfälle==Plugin)

- Druckvorfälle werden dynamisch de-/installiert
  - Durch Administration / Betrieb
- Druckvorfälle werden versioniert

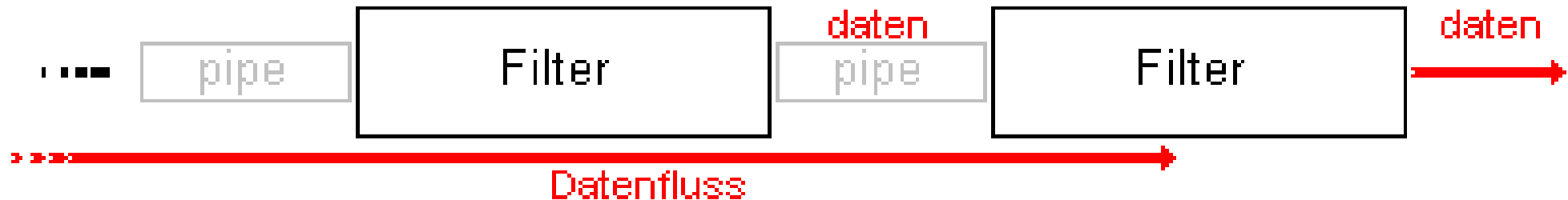
### ■ ⇒ Druckvorfälle ,on demand‘

### ■ ⇒ Universeller Workflow parametrisiert durch Druckvorfälle

### ■ ⇒ Infrastruktur und Geschäftslogik strikt getrennt

## OMS Architekturprinzipien

### Pipes&Filters – Architektur



- Filter verarbeiten die Daten
- Pipes sind für den Datenfluss zwischen den Filtern verantwortlich

#### ■ Massendruck

- Datenfluss und Kommunikation mittels *Message Queues*

#### ■ ⇒ Infrastruktur und Geschäftslogik sind unabhängig

- Einsatz in unterschiedlicher techn. Umgebung

## OMS Architekturprinzipien

### ■ Serviceorientierter Ansatz

- Druckvorfälle installieren ihre Logik in Service-Registratur
- Diese werden über Metadaten der Eingangsdaten aufgelöst

### ■ ⇒ **Dynamische Kopplung zwischen Eingangsdaten und Geschäftslogik**

## Agenda

- Deutsche Post - NL Rentenservice
- OMS – Output Management System
- OMS Architekturprinzipien
- **OMS und OSGi**
- Projektstand und –bewertung
- Zusammenfassung

## OMS und OSGi

- **Plugin-Architektur wird unterstützt durch:**
  - OSGi Modularisierung
  - Dynamik der Module
  
- **Pipe & Filter Architektur wird unterstützt durch:**
  - OSGi Modularisierung
  
- **Serviceorientierter Ansatz wird unterstützt durch:**
  - OSGi Services
  
- **Zusätzlich liefert OSGi:**
  - OSGi Compendium Services
  - Integration von ‚Web in OSGi‘

## Technologie

### ■ OSGi

- Equinox
- Spring DM

### ■ Message Queuing

- OAQ (Oracle Advanced Queueing)

### ■ Business Integration

- Apache Camel

### ■ Rendering

- XSL-FO
- Doc BridgeMill / Dope/FO (Compart)

## Technologie

### ■ IDE

- Eclipse 3.5.2

### ■ Buildprozess

- Maven 2.1.0 mit felix-maven-plugin (basiert auf BND)

### ■ Testausführungsumgebung

- PAX-EXAM von OPS4J zu Integrationstest
- Hudson zur *Continuous Integration*

### ■ Produktionsumgebung

- Equinox
- PAX-Runner von OPS4J (als *Provisioning*)

## Agenda

- Deutsche Post - NL Rentenservice
- OMS – Output Management System
- OMS Architekturprinzipien
- OMS und OSGi
- **Projektstand und –bewertung**
- Zusammenfassung



## Aktueller Stand ( Okt. 2009 )

### ■ Vorprojekt abgeschlossen

- Test- und Entwicklungsumgebung sind aufgebaut
- Weite Teile der OMS Infrastruktur sind aufgebaut

### ■ Konzept zur Inbetriebnahme fehlt noch

- *Provisioning / Roll-out* ist noch nicht gelöst

## Bewertung

- **Konzept von OSGi ist überzeugend ... , aber**
- **Werkzeugunterstützung lässt zu wünschen übrig**
  - IDE hatte Fehler (sind in nächster Version wohl behoben)
  - Werkzeuge zum Integrationstest haben noch Kinderkrankheiten
- **Produktionsplattformen müssen sich noch verbessern**
- **Grundsätzlich muss sich OSGi als Standard durchsetzen**

## Bewertung

- **Anforderungen an Teamkompetenz ist hoch**
  - Neue Programmierparadigmen/-modelle
  - Neuer Technologiestack
  
- **Betrieb und Testmanagement**
  - Es wird eine neue Plattform eingeführt
  - Stellen Sie sich vor, was dies in Ihrem Unternehmen bedeutet ?
  
- **Einführung von OSGi zieht sich durch die gesamte IT**
  - Benötigt Management Intention
  - Sollte als strategische Ausrichtung definiert sein

## Lessons learned

- **Management intention muss expliziter eingefordert werden**
  - Management sind die Auswirkung dieser Technologie nicht klar
  
- **Mischung aus Pilot- und Kernentwicklung**
  - Vorprojekt
  - Trennung hätte klarer sein müssen

## Agenda

- Deutsche Post - NL Rentenservice
- OMS – Output Management System
- OMS Architekturprinzipien
- OMS und OSGi
- Projektstand und –bewertung
- **Zusammenfassung**

## Zusammenfassung

- **OSGi ermöglicht es, Anforderungen umzusetzen,**
  - die mit herkömmlicher Mitteln nicht möglich sind
  - oder zumindest nur sehr schwierig
  
- **Der Preis ist derzeit noch hoch**
  - OSGi ist für viele Entwickler Neuland
  - Werkzeuge sind derzeit noch nicht ausgereift
  - Konzepte für Betrieb sind noch nicht ausgereift