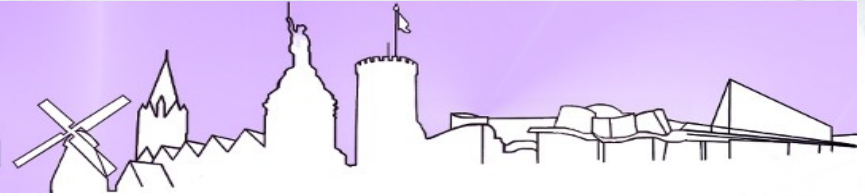


JBUG:OWL



JBoss Community

JBoss AS7.x – der etwas andere JBoss

Wolf-Dieter Fink

AS7 der etwas andere JBoss

- Schneller starten
- Weniger Speicher 'Verbrauch'
- Modularer Aufbau
- Nur noch eine Konfiguration
- Multi-node (Domain) Management

AS7

- Komplette Neuentwicklung mit Fokus
 - Performance
 - Modularität
 - Einfache Konfiguration
 - Einfaches Management

AS7 Konfiguration

- Zentrale Konfiguration in wenigen Dateien
- Alle Möglichkeiten sind persistent
ManagementConsole, CLI
- Keine Vermischung mit internen Properties
- Vereinheitlichung gemeinsamer
Eigenschaften
 - Socket bindings
 - Pfade
 - System Properties

Domain Management Terms

- **Domain Controller**
Management Instanz einer Domain (nur einmal vorhanden)
- **Host Controller**
Manager einer Slave Instanz, verbindet sich mit dem DC.
- **Process Controller**
Host lokaler Prozess zur Überwachung
- **Server Group**
Gruppe von Servern mit gleicher Konfiguration
- **Server**
Instanz eines Applikations Servers
- **Cluster**
Ein oder mehrere Server als Applikation Cluster

Domain oder Standalone Modus

Domain

- Domain oder Host-Controller starten mehrere JVM's
- Zentrale Konfiguration
- Zentrales Deployment
- 0...n Cluster möglich
- KEIN deploy per Filesystem

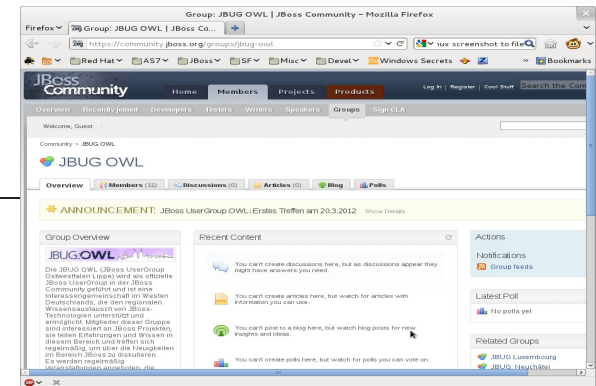
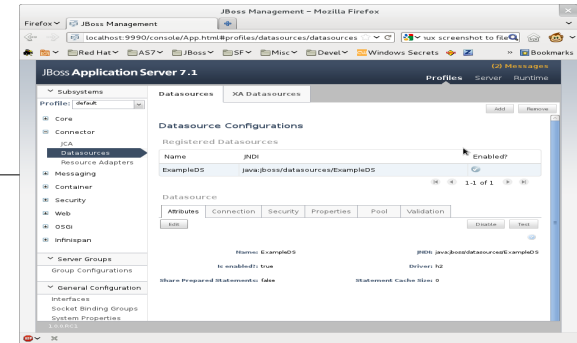
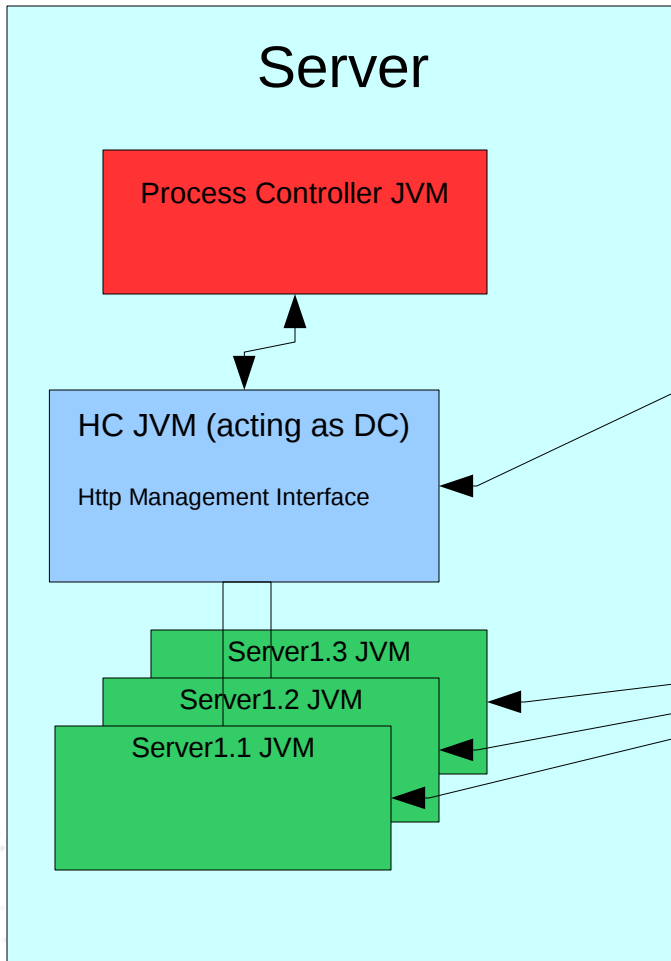
Standalone

- Verhalten wie AS4/5/6
Eine JVM für alle Aufgaben
- Kein gemeinsames Deployment (Farming)
- Cluster möglich
- Deploy per Filesystem

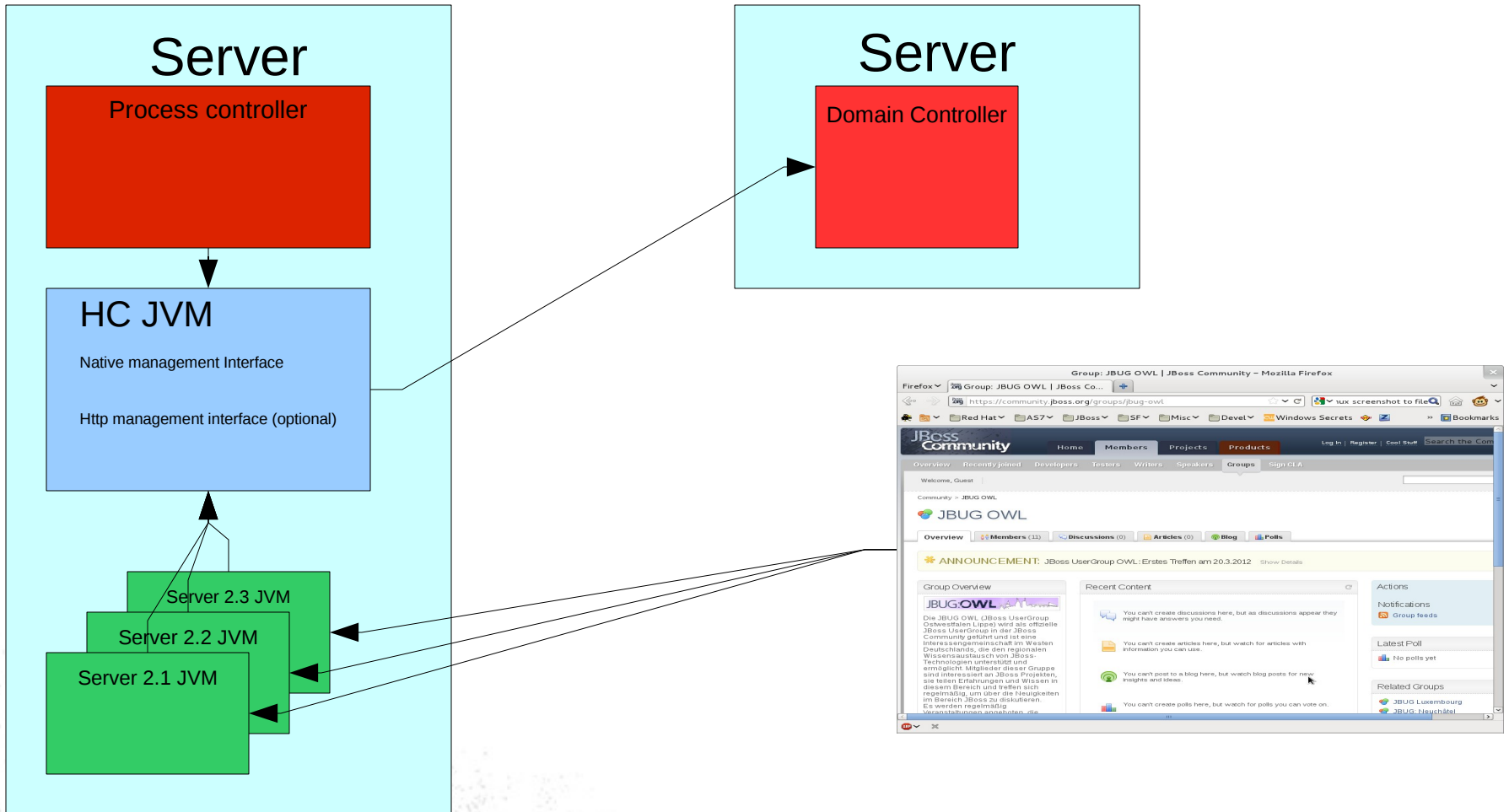
Konfiguration einer Domain

- domain.xml
 - Primäre Konfiguration
port-bindings, interfaces, server-groups
 - Repräsentiert den aktuellen Zustand
- host.xml
 - Host spezifisches environment
IP-Adresse, Anzahl Server

Domain Controller



Slave Instanz



Domain Server Konfiguration

host.xml

```
<host name="server1">  
  <management/>  
  <interfaces>  
    <interface name="" .../> binding, port  
  </interfaces>  
  <jvms>  
    <jvm name="" .../> heap permgen options  
  </jvms>  
  <servers>  
    <server name="">  
      Zuordnung zu Gruppe, JVM, Interface, Ports (offset)  
      Individuelle Parameter  
    </server>  
  </servers>
```

Migration to AS7 EAP6

- **Classpath und ClassLoader**
Durch die Einführung von Modules ist die Deklaration von Abhängigkeiten und das Laden von Klassen die erste Hürde.
- **Caching**
Anwendungen die JBoss-Cache verwendet haben müssen angepasst werden für Infinispan
- **JPA**
Wird JPA mit Second-Level caching verwendet muss die persistence.xml angepasst werden.
- **Security**
login-config ist obsolet, alle Konfigurationene sind in der standalone.xml oder domain.xml

Migration to AS7 EAP6

- **Resources**

Datasources, JMS destinations, RAR, SAR und andere Ressourcen werden nicht mehr 'deployed' sondern in standalone.xml oder domain.xml konfiguriert

- **Messaging**

JBoss-Messaging wird komplett ersetzt durch HornetQ

- **JNDI Lookups**

Alle lookups müssen angepasst werden entsprechend der neuen JNDI Namespace Konventionen.

Ein 'customizing' der Namen ist zur Zeit nicht vorgesehen.

Classloading

Erhebliche Unterschiede zu vorhergehenden Versionen

- Deployments haben keinen Zugriff auf Klassen der Module oder anderer Deployments wenn nicht explizit angegeben
- Implizite (automatische) Abhängigkeiten werden geladen Aufgrund von verschiedenen Triggern
z.Bsp. javax.api, javaee.api, javax.persistence.api
<https://docs.jboss.org/author/display/AS7/Implicit+module+dependencies+for+deployments>
- Deklaration in MANIFEST.MF ist der Portable in der Spec angegebene Weg
Dependencies: org.mysql, org.javassist
- Deklaration in der jboss-deployment-structure.xml
- Globale Module können in stantalone.xml oder domain.xml angegeben werden.

```
<subsystem xmlns="urn:jboss:domain:ee:1.0" >  
  <global-modules>  
    <module name="com.mysql"/>  
  </global-modules>
```

Classloading

Reihenfolge

- **Container**
Module die vom Container hinzugefügt werden incl. der automatischen wie JEE API
- **Deklariert**
Abhängigkeiten aus MANIFEST und jboss-deployment-structure.xml
- **Klassen innerhalb des deployments**
WEB-INF/classes
- **Inter deployment**
Klassen anderer module innerhalb eines EAR deployments oder EAR/lib

Classloading WAR

- WEB-INF/lib und WEB-INF/classes sind gleichgestellt
- Alle Klassen in einem WAR werden im gleichen Classloader geladen

Classloading

EAR

- Standardmäßig kein gemeinsamer ClassLoader
- WAR EJB und EAR/lib sind getrennte Module
- SubDeployments 'erben' den Zugriff auf den Parent und damit auf EAR/lib
- Keine Zugriffe auf andere SubDeployments ohne deklaration

Verhalten kann konfiguriert werden:

```
<subsystem xmlns="urn:jboss:domain:ee:1.0" >  
  <ear-subdeployments-isolated>>false</ear-subdeployments-isolated>  
</subsystem>
```


Helpful AS7 links

<https://docs.jboss.org/author/display/AS71/Documentation>

https://community.jboss.org/wiki/DataSourceConfigurationinAS7#Installing_a_JDBC_driver_as_a_module

<https://community.jboss.org/wiki/AS7ClusterHowto>