



# Apache Geronimo

**Sprecher:**  
Christian Dedek  
Kristian Köhler

Orientation in Objects GmbH

Weinheimer Str. 68  
68309 Mannheim

[www.oio.de](http://www.oio.de)  
[info@oio.de](mailto:info@oio.de)



## Agenda

- Einführung
- Deployment
- Architektur
- Build your own server



## Geronimo? Was ist das...



- J2EE 1.4 konformer Application Server
  - J2EE 1.4 TCK (Technology Compatibility Kit) im Okt `05 bestanden
- OpenSource Implementierung von Apache
  - Steht unter Apache Software License nicht LGPL
- „Zusammenstellung“ mehrerer Projekte
  - Vervollständigung des J2EE Stacks durch ASF/BSD lizenzierten Code
- Erstellung eines „eigenen Servers“ möglich



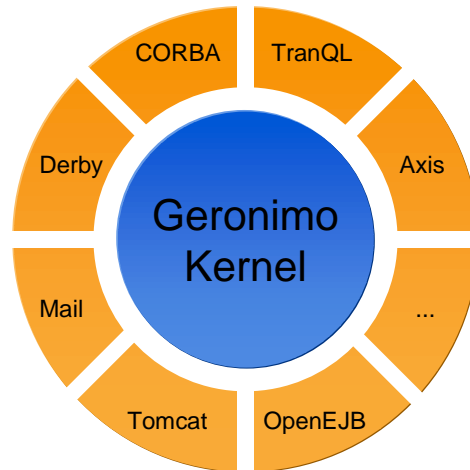
## Warum Application Server von Apache?



- Bisher kein J2EE Server mit BSD abgeleiteter Lizenz
  - JBoss und JOnAS haben GPL abgeleitete Lizenz
- Apache Software Foundation hat einige J2EE Projekte
  - einzelne Bestandteile aber keine Integration
- Best of breed app server reusing Open Source from the community



## Geronimos Integrationsgedanke



5

## Benutzte „FremdKomponenten“

- OpenEJB
  - EJB Container (<http://www.openejb.org>)
- Jetty
  - WebContainer (<http://jetty.mortbay.org/>)
- Tomcat
  - WebContainer (<http://jakarta.apache.org/tomcat>)
- ActiveMQ
  - JMS Provider (<http://activemq.codehaus.org>)
- TranQL
  - EJB CMP Engine/ DB Connection Pools (<http://tranql.codehaus.org>)
- HOWL
  - Logging (Transactions) (<http://howl.objectweb.org>)

6

## Simple things that make your life easier



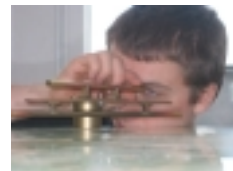
- Zentrale Dienste Konfiguration über config.xml
- Administration, Konfiguration und Monitoring von Diensten und Anwendungen über Webconsole
- JMX-Schnittstelle
- JSR-88 Schnittstelle
- Apache Derby und ActiveMQ Dienste integriert
- optional Apache LDAP-Server mit Userverwaltung

7

## Geronimo Server - First Look



- Aktuelle Version 1.0.0 (1.1 in Vorbereitung)
  - Vorkonfiguriert mit Jetty oder Tomcat als WebContainer
  - erhältlich als Archiv oder Installer
    - <http://geronimo.apache.org/>
- Server muß nur entpackt und gestartet werden
  - `bin/startup.bat`
- Tools für Verwaltung enthalten
  - Kommandozeilenversion
  - WebConsole
  - Maven Plug-ins



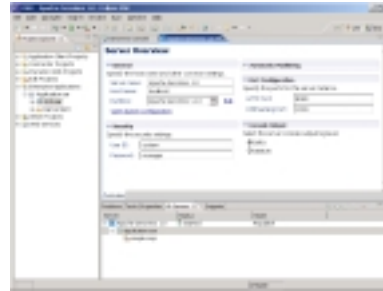
8



## Anwendungsdeployment



- WebConsole unterstützt Remote Deployment
  - Upload der Anwendung durch remote deploy Application
- Kommandozeilendeployment
  - mitgelieferte Skripte
- JSR-88 Treiber
  - Build Once, run anywhere
- Eclipse WTP
  - Plugin für Geronimo 1.0 vorhanden
- Hot-Deployment zur Laufzeit
  - Kopieren des Anwendungsarchiv in deploy Verzeichnis

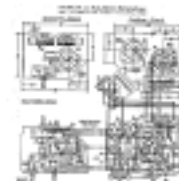


11

## Konfigurationen erstellen - Deployment Plan



- XML Datei zur Modulbeschreibung
  - für Dienste, Ressourcen und Anwendungen
  - besitzt eindeutigen Namen
- deklariert Administrations- und Konfigurationseinstellungen für die Deploymenteinheit
  - z. B. PortNummer des WebServers, Datenbankangabe für EJBs
- Deploymentprozess benötigt in jedem Fall einen Deployment Plan
  - im Archiv oder als Parameter für Deployer
    - **META-INF/geronimo-application.xml**



## Mögliche Deploymenteinheiten



META-INF\ejb-jar.xml  
openejb-jar.xml



META-INF\web.xml  
geronimo-web.xml



META-INF\application.xml  
geronimo-application.xml



META-INF\ra.xml  
geronimo-ra.xml

13

## Deployment Plan Aufbau



- XML-Elemente
  - Geronimo-spezifischen Schemata zugeordnet
  - den J2EE-Deploymentdeskriptoren entlehnt
- XML - Namensräume
  - erlauben eine Strukturierung der XML-Elemente nach Semantik der Deklaration
- Vorteil:
  - gleichartige Deklarationen in unterschiedlichen Deploymentformen --> identische XML-Elemente
  - gemeinsame Deklaration aller Bestandteile und Features eines EAR innerhalb eines Dokuments

14

## Deployment Plan Beispiel

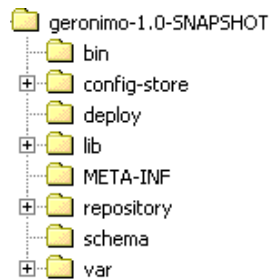
```
<module>
<ejb>ejb-cmp-sample-1.0-SNAPSHOT.jar</ejb>
<sys:environment>
<sys:configId>
<sys:groupId>de.oio.jax06</sys:groupId>
<sys:artifactId>cmp</sys:artifactId>
<sys:version>1.0</sys:version>
</sys:configId>
</sys:environment>
< naming:cmp-connection-factory>
< enterprise-beans>
< relationships>
</relationships>
</openejb-jar>
</module>
<module>
<ejb>ejb-sb-sample-1.0-SNAPSHOT.jar</ejb>
<openejb-jar xmlns="http://www.openejb.org/xml/ns/openejb-jar-2.1">
<sys:environment>
<sys:configId>
<sys:groupId>de.oio.jax06</sys:groupId>
<sys:artifactId>sb</sys:artifactId>
<sys:version>1.0</sys:version>
</sys:configId>
</sys:environment>
< enterprise-beans>
</openejb-jar>
</module>
```

15

## Deployment

```
$. /deploy.bat deploy YourApplication.ear
```

```
$. /deploy.bat undeploy YourApplication.ear
```





## Agenda



- Einführung
- Deployment
- **Architektur**
- Build your own server



17

## Geronimo Bean (GBean) Architektur - (I)



- Zur Abbildung des Integrationscharakters des Servers
  - relativ lose Zusammenstellung der Serverbestandteile
- Dependency Injection basiertes Konfigurations- und Management System
  - über XML Dateien werden Komponenten konfiguriert
- Keine Bindung an JMX Spezifikation
  - Kein zentraler JMX Kernel über den kommuniziert wird
  - Komponenten lassen sich allerdings über JMX verwalten

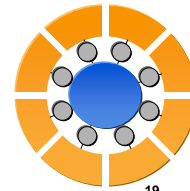


18

## Geronimo Bean (GBean) Architektur - (II)



- GBean kleinste verwaltbare Einheit innerhalb des Servers
  - Eigener Lebenszyklus, Metadaten
    - **statische getBeanInfo Methode**
  - Separates Starten und Stoppen möglich
- Sämtliche Komponenten sind über GBeans angemeldet
  - Serverbestandteile und Anwendungen
  - Verwaltung für alle Komponenten gleich

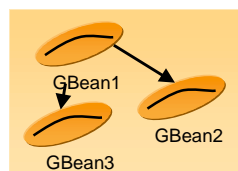


19

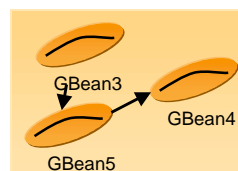
## Geronimo Module/ Konfigurationen - (I)



- Zusammenstellung von konfigurierten GBeans
  - elementare, zusammengehörende Bausteine
    - **WebContainer Modul besteht z. B. aus mehreren Komponenten**
- Modul ist selbst wieder GBean und kann verwaltet werden
  - besitzt Lebenszyklus, separates Starten und Stoppen



Module1



Module2

20

## Geronimo Module/ Konfigurationen - (II)



- Eindeutige Namen innerhalb des Servers (ModuleId)
  - Beispiel: `geronimo/welcome-tomcat/1.1-SNAPSHOT/car`  

}	}	}	}
groupId	artifactId	version	type
- Können in einzelnes Archiv gepackt werden
  - Innerhalb des Servers Speicherung in ConfigurationStore/ Repository
  - Ablage in Maven Repository als CAR Datei
  - CAR Datei enthält Konfigurationsinformationen

21

## Geronimo Server und Module



- Server arbeitet modulbasiert
  - Verwaltungseinheit des Servers
- Serverbestandteile sowie Applikationen sind Module
  - können auch zusammen gepackt werden
    - z. B. EJB Anwendung mit konfigurierter Datenbankanbindung
- Module müssen nicht ausgeführt werden
  - können sich „offline“ im Server befinden
    - **Stop bedeutet kein Undeployment**

22

## Beispiel: Module starten und stoppen



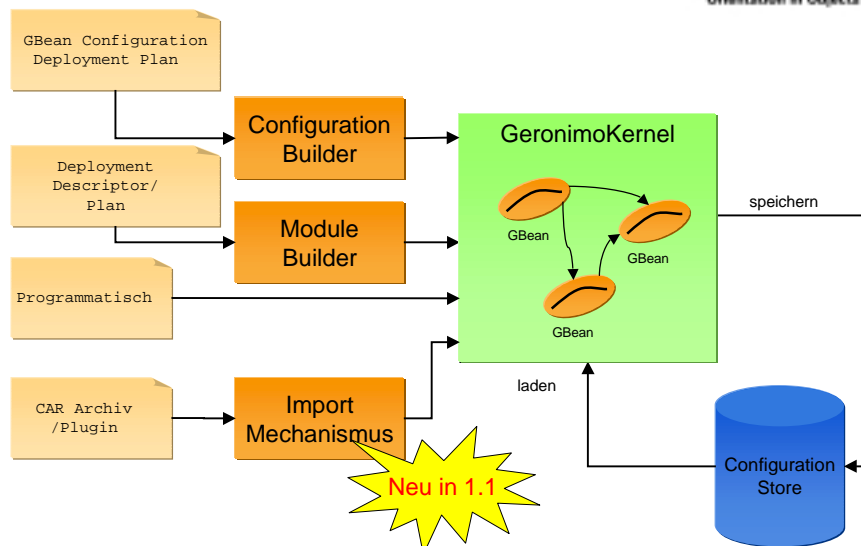
```
$/deploy.bat list-modules
```

```
$/deploy.bat stop geronimo/welcome-tomcat/1.1-SNAPSHOT/car
```

```
$/deploy.bat start geronimo/welcome-tomcat/1.1-SNAPSHOT/car
```



## Geronimo Deploymechanismus für Module



## Geronimo Plugin



- Beliebiges Geronimo Module
  - Serverbestandteil oder Anwendung
  - „kompilierte Version“ eines Moduls
- Besitzen zusätzlich Metainformationen
  - Abhängigkeiten zur ausführenden Umgebung
    - **z. B. Geronimo Version, JVM Version, WebContainer**
  - Angabe zu externen Abhängigkeiten
    - **Welche Bibliotheken werden benötigt**
- Können einfach zwischen Servern übertragen werden
  - Abhängigkeiten können automatisch nachgeladen werden
    - **aus Remote Maven Repository**

25

## Geronimo Plugin Import/Export (I)



- Module/ Plugins können exportiert und importiert werden
  - Weitergabe als Archiv (momentan CAR)
  - über WebConsole
- Abhängigkeiten können aus Repository nachgeladen werden
  - Wildcardangaben sind möglich
    - **geronimo/j2ee-server//car - z. B. keine Versionsangabe!**
- Zentrale Ablage in Repository
  - Layout an Maven Repository angelehnt
  - Plugin und Metadaten müssen bereitgestellt werden

26

## Geronimo Plugin Import/Export (II)



- Über Plugins schnelles Aufsetzen neuer Instanzen möglich
  - Datenbankeinrichtung, Anwendungen mit Abhängigkeiten
- Softwareverteilung für Serverbestandteile oder Anwendungen
  - momentan eine Seite vorhanden
    - <http://www.geronimoplugins.com>
- Geronimo Server kann selbst als Repository auftreten
  - `http://<host>/console-standard/maven-repo/geronimo-plugins.xml`
  - „Clone“ einer Geronimo Instanz

27

E15 - Apache Geronimo

© 2006 Orientation in Objects GmbH

## Beispiel: Module Importieren/ Exportieren



E15 - Apache Geronimo

© 2006 Orientation in Objects GmbH

## Agenda



- Einführung
- Deployment
- Architektur
- **Build your own server**



29

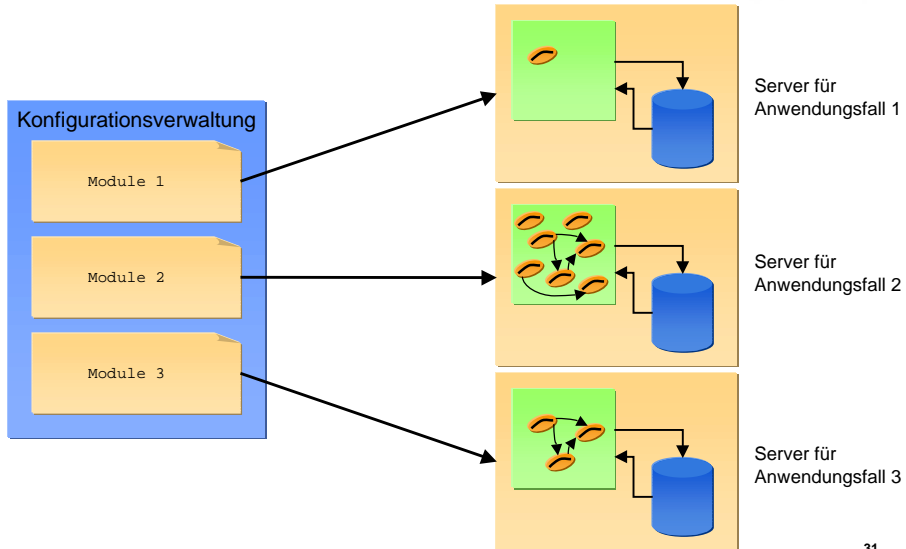
## Zusammenstellungen von Komponenten



- Verschiedene Konfigurationen lösen unterschiedliche Probleme
  - nicht immer vollständiger J2EE Server nötig
  - Welche Dienste werden benötigt?
- Eventuell eigene kommerzielle Erweiterung gewünscht
  - z. B. Verschlüsselter ConfigStore, eigenes Anwendungsformat
- Konfigurationen geben Funktionsumfang vor
  - welche Anwendungstypen werden unterstützt (ear, war, etc.)
- Durch Lizenz auch eigene Verwendung möglich
  - Server auch in kommerziellen Projekten als Basis möglich

30

## Konfigurationsauslieferung



31

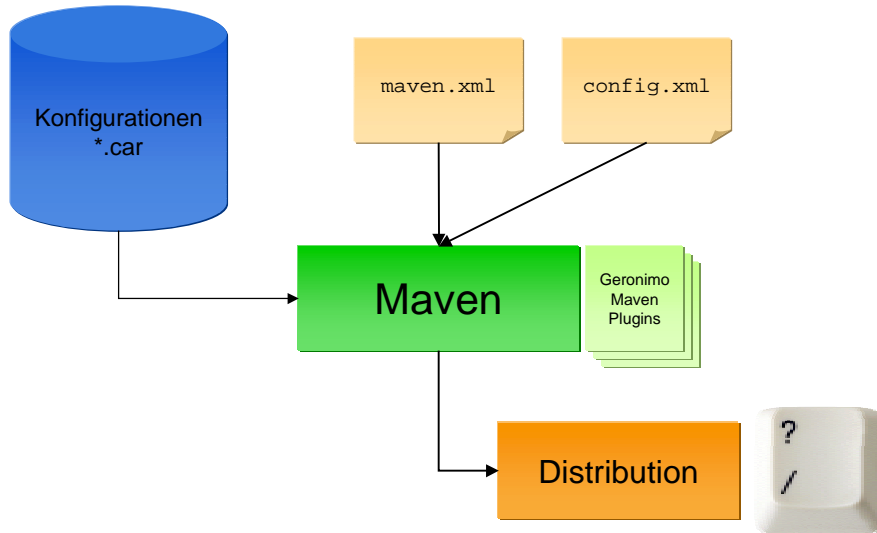
## Geronimo Assemblies

- Server Module werden einzeln gebaut
  - Beim Übersetzen Ablage in Maven Repository (\*.car Dateien)
- Zusammenstellen des Servers über Assembly Konfiguration
  - maven.xml und config.xml
- Eigentliches Assembly übernehmen Geronimo Maven Plugins
  - geronimo-assembly-plugin
- Geronimo kommt mit 5 Standardassemblies (1.1-SNAPSHOT)
  - J2EE-Jetty, J2EE-Tomcat, Installer, Minimal-Jetty, Minimal-Tomcat

32



## Weg zum eigenen Server



E15 - Apache Geronimo

© 2006 Orientation in Objects GmbH



# Vielen Dank für Ihre Aufmerksamkeit !



Orientation in Objects GmbH

Weinheimer Str. 68  
68309 Mannheim

[www.oio.de](http://www.oio.de)  
[info@oio.de](mailto:info@oio.de)



Orientation in Objects GmbH  
Weinheimer Str. 68  
68309 Mannheim  
[www.oio.de](http://www.oio.de)  
[info@oio.de](mailto:info@oio.de)