



Guten Morgen Geronimo

Kristian Köhler

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Agenda

- Einführung
- Architektur
- GBean Framework
- Demo



2

Geronimo? Was ist das...



- J2EE 1.4 konformer Application Server (Ziel)
 - „durchlaufen“ des J2EE 1.4 TCK (Technology Compatibility Kit)
- OpenSource Implementierung von Apache
 - Gestartet im August 2003
 - Top-level Apache Projekt
 - Steht unter Apache Software License nicht LGPL
- „Zusammenstellung“ mehrerer Projekte
 - Vervollständigung des J2EE Stacks durch ASF/BSD lizenzierten Code



3

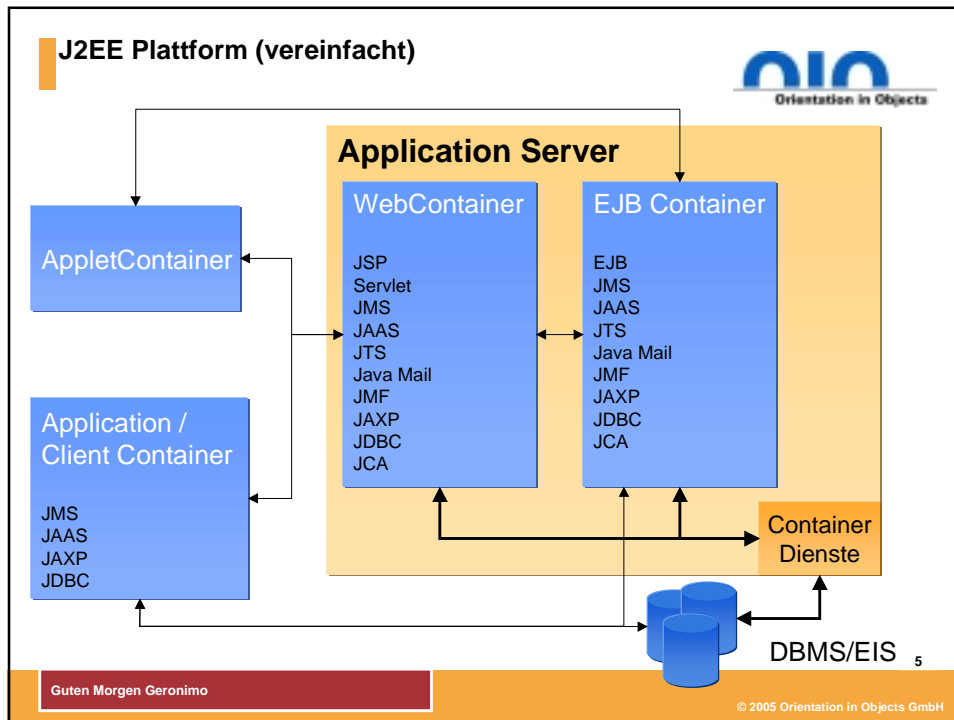
Was heißt J2EE Application Server?



- Realisiert die J2EE Architektur von Sun
 - J2EE Spezifikation
 - **EJB, Servlet, JSP, Connector, JTA, JavaMail, JAF, JAXP, JAAS, JNDI, JDBC, JMS**
- Ablaufumgebung für Komponenten über Container
 - EJB Container, WebContainer,...
- Teil einer Drei-Schicht Architektur
 - klassischerweise die Geschäftslogik Schicht



4



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

The Apache Software Foundation
<http://www.apache.org/>

Guten Morgen Geronimo

© 2005 Orientation in Objects GmbH

Welche Ziele verfolgt Geronimo?



- Integrationsframework für Systemdienste
 - Existierende Implementierungen nutzen
 - Wenn nötig Eigenimplementierung
 - Flexibilität für Austausch von Diensten
- Zielgerichtetes Deployment
 - gute Skalierbarkeit
 - Schwerpunkt auf Konfiguration und Verwaltung
- Hauptaugenmerk auf produktiven Einsatz
 - Server ist kein Selbstzweck
 - Zielgruppe sind Administratoren und Entwickler



7

High-level Überblick



Anwendungsentwicklungs Frameworks (J2EE, Spring, etc)

Security

Transaction

Deployment

Remoting

Naming

Work
Mgmt.

Configuration
Management

Dependency
Management

Repository

Lifecycle
Management

Apache Geronimo Kernel

8

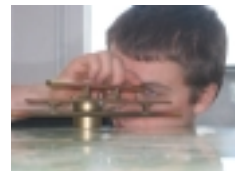
Wer macht was?

Service	Spezifikation	OpenSource Projekt
Transactions	JTA	Geronimo
Security	JAAS, JACC	
Threading/Work		
Naming	JNDI	
Mail	JavaMail, JAF	
WebService Container	JAX-RPC, EWS	
Timed Operation		
EJB Container	EJB	OpenEJB
Database	JDBC	Derby
Web Container	Servlet, JSP	Jetty, Apache Tomcat
Messaging	JMS	ActiveMQ

9

Geronimo Server - First Look

- Aktuelle Version 1.0-M3
 - erhältlich als Archiv oder Installer
 - <http://geronimo.apache.org/>
- Server muß nur entpackt werden
 - `java -jar bin/server.jar`
- Tools für Verwaltung enthalten
 - Kommandozeilenversion
 - Maven Plug-ins



10

Agenda

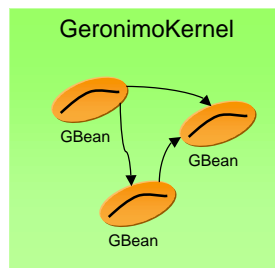


- Einführung
- **Architektur**
- GBean Framework
- Demo



11

Geronimo Kernel



12

Kernel Verantwortlichkeiten



- Komponenten Registry
 - Anmeldung und Lokalisierung von GBean Instanzen
- Repository
 - zentrale strukturierte Ablage von benötigten Bibliotheken
 - für Dienste und Konfigurationen
- Component Management
 - Ermöglicht die Verwaltung der GBeans über JMX
- Configuration Manager
 - Lifecyclemanagement für Konfigurationen (laden, starten, etc.)

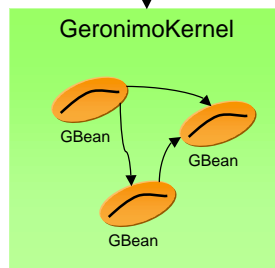
13

Geronimo ConfigurationBuilder



GBean Configuration Deployment Plan

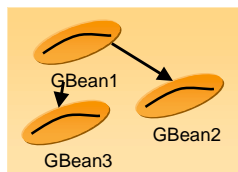
durch ConfigurationBuilder beim Übersetzen



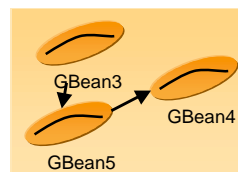
14

Konfigurationen (I)

- Zusammenstellung von konfigurierten Komponenten
 - elementare, zusammengehörende Bausteine
 - Komponenten sind GBeans
- „Alles“ ist ein GBean
 - Container, Connectoren, Anwendungen,...
- Jeder Dienst innerhalb Geronimo wird in Form eines oder mehrerer GBeans deployt



Configuration1



Configuration2

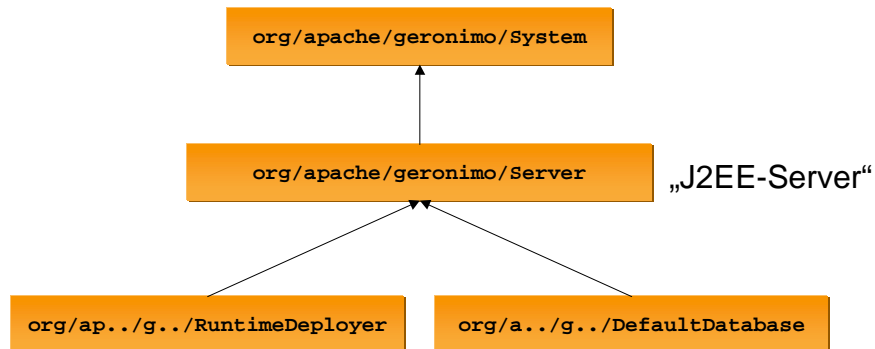
15

Konfigurationen (II)

- Können einzeln verwaltet werden
 - Starten, Stoppen und Anhalten zur Laufzeit möglich
- Können aufeinander aufbauen („Vererbungshierarchie“)
 - Beispiel: j2ee-server „erbt“ von system
 - Classloader-Hierarchie
- Eindeutige Namen innerhalb des Servers
 - Beispiel: org/apache/geronimo/DebugConsole
- Können in einzelnes Archiv gepackt und verteilt werden
 - Innerhalb des Servers Speicherung in ConfigurationStore

16

Beispielkonfigurationen innerhalb Geronimo



17

Guten Morgen Geronimo

© 2005 Orientation in Objects GmbH

Beispiel: Konfigurationen starten und stoppen



```
$java -jar bin/deployer.jar list-modules
```

```
$java -jar bin/deployer.jar start org/apache/geronimo/DebugConsole
```

```
$java -jar bin/deployer.jar stop org/apache/geronimo/DebugConsole
```



Guten Morgen Geronimo

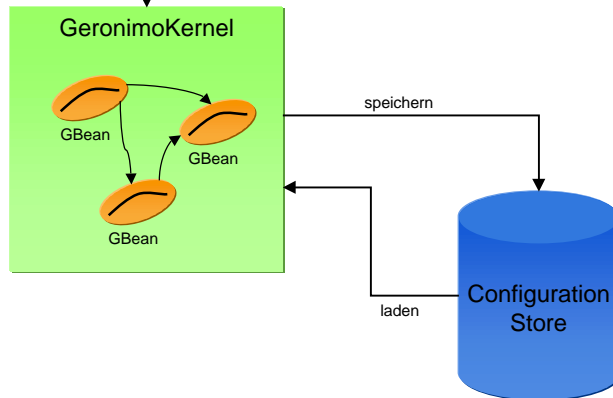
© 2005 Orientation in Objects GmbH

Geronimo ConfigurationStore



GBean Configuration
Deployment Plan

durch ConfigurationBuilder beim Übersetzen



19

Guten Morgen Geronimo

© 2005 Orientation in Objects GmbH

ConfigurationStore



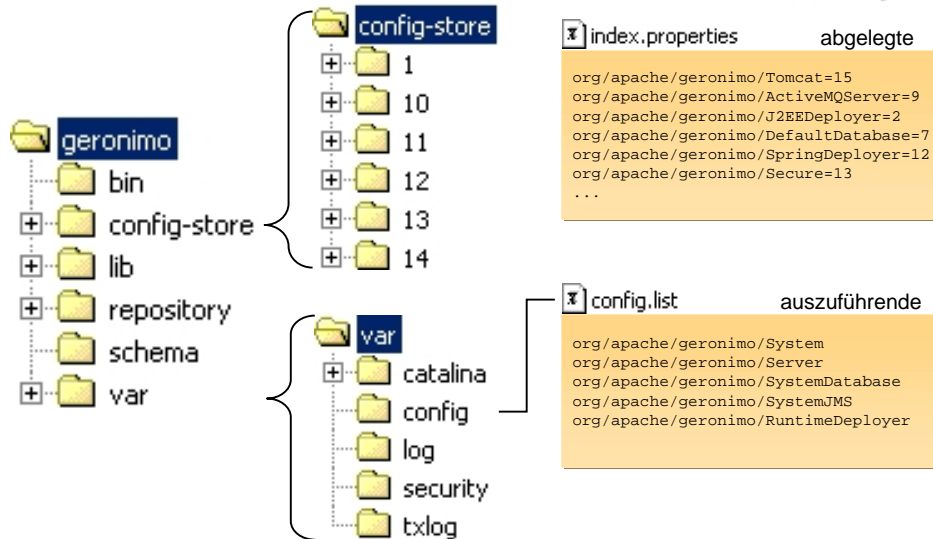
- Einheitliche Schnittstelle zur Ablage von Konfigurationen
 - Angedacht: Mögliche zentrale Ablage in Datenbank, LDAP
- Versionierte Ablage von Konfigurationen
 - Konfiguration mit eindeutigen Namen
- Gespeicherte Konfigurationen müssen nicht ausgeführt werden
 - Start und Stop Kommando vorhanden
- Standardimplementierung speichert in Dateisystem
 - Unterverzeichnis config-store

20

Guten Morgen Geronimo

© 2005 Orientation in Objects GmbH

Konfigurationen im Dateisystem



21

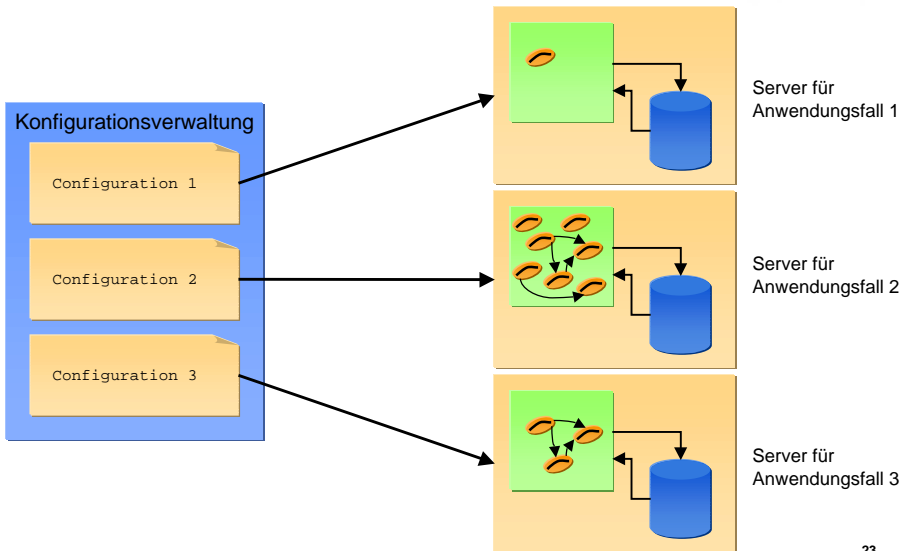
Zusammenstellungen von Komponenten



- Verschiedene Konfigurationen lösen unterschiedliche Probleme
 - nicht immer vollständiger J2EE Server nötig
 - Welche Dienste werden benötigt?
- Konfigurationen geben Funktionsumfang vor
 - welche Anwendungstypen werden unterstützt (ear, war, etc.)
- Zusammenstellung des Servers findet beim „Bauen“ statt
 - Extra Modul definiert „Plans“ (modules/assembly)
- Durch Lizenz auch eigene Verwendung möglich
 - Server auch in kommerziellen Projekten als Basis möglich

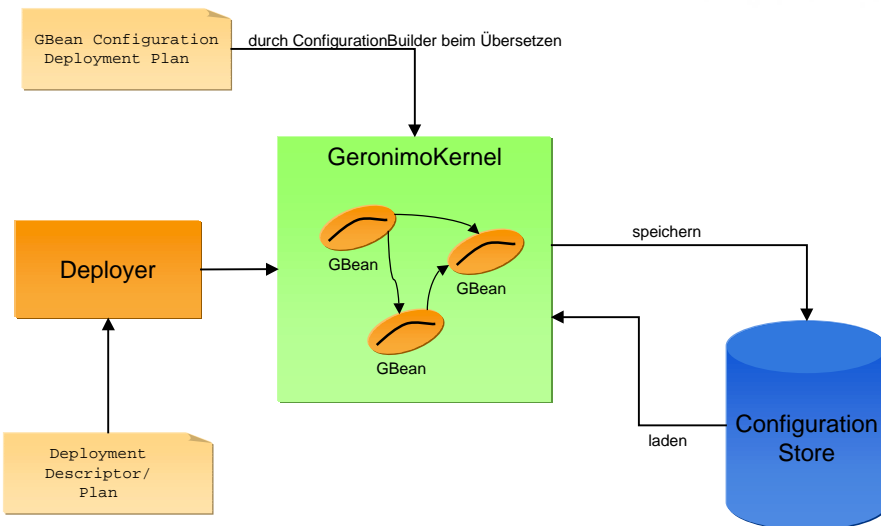
22

Konfigurationsauslieferung



23

Geronimo Deployer - „Hot Deployment“



24

Was kann deployt werden - Konfigurationsdateien



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



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



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



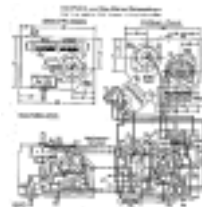
META-INF\ra.xml
geronimo-ra.xml

25

Konfigurationen erstellen - Deployment Plan



- XML Datei zur Konfigurationsbeschreibung
 - für Dienste, Ressourcen und Anwendungen
 - besitzt eindeutigen Namen
 - optionale Angabe der Elternkonfiguration
- Enthält zusätzliche Angaben für die deployte Einheit
 - z. B. PortNummer des WebServers
- Werden beim Deployment benötigt
 - im Archiv oder als Parameter für Deployer
 - META-INF/geronimo-application.xml



26

Beispiel einer „Stand-alone Konfiguration“



```
<?xml version="1.0" encoding="UTF-8"?>

<configuration
  xmlns="http://geronimo.apache.org/xml/ns/deployment"
  configId="de/oio/geronimo/ConfigurationExample"
  parentId="org/apache/geronimo/System"
  >

  <gbean name="BspGBean" class="de.oio.BspGBean">
    <attribute name="text">Hello World!</attribute>
  </gbean>

</configuration>
```

27

Guten Morgen Geronimo

© 2005 Orientation in Objects GmbH

Deployment Descriptoren / Deployment Plan



```
$java -jar bin/deployer.jar deploy configuration-example.xml
```

```
$java -jar bin/deployer.jar deploy YourApplication.ear
```

```
$java -jar bin/deployer.jar deploy MyApplication.ear
MyApplication-plan.xml
```



Guten Morgen Geronimo

© 2005 Orientation in Objects GmbH

Agenda



- Einführung
- Architektur
- **GBean Framework**
- Demo



29

GBean Framework (I)



- Grundlage des Geronimo Kernels
- J2EE unabhängiges IOC Server Framework
 - Inversion of Control (IOC)
 - Dependency Injection
- Caching von Attributen oder Methodenrückgabewerte möglich
 - Reduzierung der Aufrufe
- GBeans besitzen Lifecycle
 - unterstützen JSR 77 lifecycle



30

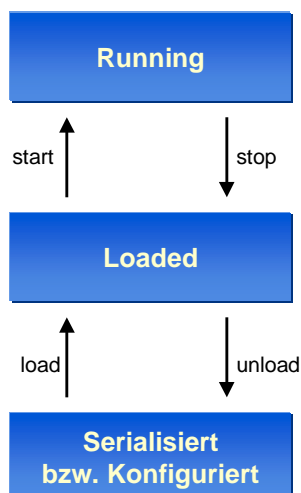
GBean Framework (II)



- Beanzustände können gespeichert und wieder hergestellt werden
 - Attributwerte können als persistent gekennzeichnet werden
 - Ablage innerhalb ConfigurationStore
- GBeans sind über JMX verwaltbar
 - sind ohne JMX verwendbar
 - *GBeanMBean* steht zur Verfügung
- Implementierungen sind POJOs
 - besitzen Metadaten zur Beschreibung
- „Kommunikation“ findet über CGlib statt
 - Klassengenerierung zur Laufzeit

31

GBean Lifecycle



32

GBean Beispiel I (Java)



```
public class GreaterBean {

    public static final GBeanInfo GBEAN_INFO;

    static {
        GBeanInfoBuilder infoBuilder =
            new GBeanInfoBuilder(GreaterBean.class);

        infoBuilder.addAttribute("wert", String.class, true);
        infoBuilder.addOperation("printWert", new Class[]{});
        infoBuilder.addReference("OtherBean", SimpleBean.class);

        GBEAN_INFO = infoBuilder.getBeanInfo();
    }

    public static GBeanInfo getGBeanInfo() {
        return GBEAN_INFO;
    }

    ...
}
```

33

GBean Beispiel II (XML - Deployment Plan)



```
<configuration xmlns="http://geronimo.apache.org/xml/ns/deployment"
    configId="de/oio/geronimo/Example"
    parent="org/apache/geronimo/system"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <dependency>
        <uri>oio/jars/example-gbean-1.0.jar</uri>
    </dependency>

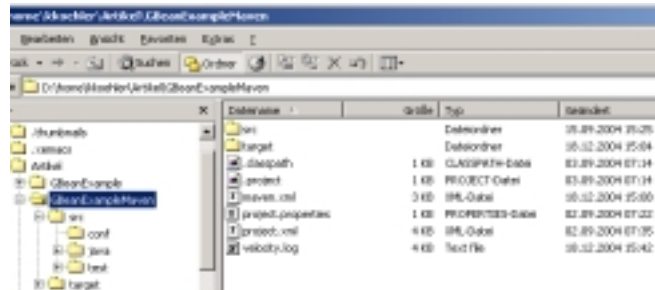
    <gbean name="SimpleGBean" class="de.oio.geronimo.SimpleBean">
        <attribute name="name" type="java.lang.String">Kristian</attribute>
    </gbean>

    <gbean name="GreaterGBean" class="de.oio.geronimo.GreaterBean">
        <reference name="OtherBean">
            <name>SimpleGBean</name>
        </reference>
    </gbean>

</configuration>
```

34

GBean Beispielprojekt



```
$java -jar bin/deployer.jar deploy YourApplication.ear
```



Agenda

- Einführung
- Architektur
- GBean Framework
- **Demo**





Fragen ?

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de



**Vielen Dank für Ihre
Aufmerksamkeit !**

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de