Apache Geronimo: 
A Peek Under the Hood

Bruce Snyder 
Jailey Solutions, LLC
What Is Apache Geronimo?

It is Not...
- Yet another lightweight container
- Yet another web framework
- Yet another AOP framework
- An MVC framework

It is...
- Designed for long running servers
- Designed to tolerate partial component failures
- System oriented services
From the Ground on Down

- Open Source Enterprise Infrastructure
- Highly modular architecture
- Initial manifestation as a J2EE Server
- Other possible configurations
  - Native Spring Integration
  - Bi-directional PicoContainer Integration
  - Native Portlet Integration
  - Native Web Services Integration
  - Deployment Chaining
  - and many more ...
Geronimo Kernel

- Fundamental Core
  - Small memory consumption ~150KB code
  - Component Configuration
  - Component Registration
  - Integrated Repository
  - Lifecycle Control
  - Dependency Manager

- Components are called GBeans
  - Simple object, plus some metadata
What Are GBeans?

- A J2EE managed object (MBeans)
- Used to bridge JSR-77 lifecycle requirements
- GBean wrappers allow just about anything to be plugged in to Geronimo
- Implement the GBeanLifecycle interface

```java
public interface GBeanLifecycle {
    void doStart() throws WaitingException, Exception;
    void doStop() throws WaitingException, Exception;
    void doFail();
}
```
What Are GBeans? (Continued)
GBean Archive

- A JAR file containing:
  - Persisted GBean instances
  - GBean metadata
  - Required Java classes reside:
    - Either in the jar, or
    - As dependencies from a central repository

- Must have a unique ID
- Can be signed for distribution
- Can be executable
GBean Descriptor Example

- Provides IoC configuration for a GBean
- `<gbean>` elements can be placed in any descriptor - usually `geronimo-application.xml`

```xml
<gbean name="geronimo.system:role=LogAppender,type=ConsoleAppender" class="org.apache.geronimo.system.logging.log4j.appender.ConsoleAppenderService">
  <attribute name="threshold" type="java.lang.String">INFO</attribute>
  <attribute name="layoutPattern" type="java.lang.String">
    %d{ABSOLUTE} %-5p [%c{1}] %m%n
  </attribute>
  <attribute name="target" type="java.lang.String">System.out</attribute>
</gbean>
```
Demo (Time Permitting)

- Demonstrate simple GBean wrapper
Repository

Structured collection of jars

- Designed to work in conjunction with Maven
  - Pluggable implementation
- Every jar has a unique group and artifact ID, e.g.
  - geronimo-spec
  - geronimo-j2ee-1.4-RC2

- Default repository is local file system
  - Others allow auto-download
Basic Configuration Builder

- **Deployers** are J2EE specific
  - JSR-88

- **Builders** are Geronimo specific
  - Create configuration objects containing GBeans
  - Serialized to a file upon shutdown

- From a pure GBean configuration
  - Very raw – used to bootstrap a server
  - You specify GBeans in an XML descriptor
  - Not intended for end users
Application Builders

- Two stages make up deployment (JSR-88)
  - Deployment
    - Read the ear file and the server config(s)
  - Distribution
    - Send the ear file and server config(s) to the server
- Geronimo handles both stages in single step for the user
- Application-specific configurations
Application Builders *(Continued)*

- Most complex Builder is J2EE deployer
  - Implements JSR-88 deployment specification
  - Take EJB-jars, wars, rars
  - Add in a deployment plan (XML)
  - Output a Geronimo Configuration ARchive (.car)
- Can produce very complex GBean definitions
  - *e.g.* entire EJB gets mapped to a GBean
- Allows for deploy-time optimizations
  - *e.g.* Precompile EJBQL, MessageSelectors, *etc.*
Configurations

- **Hot Swap**
  - Local Environment (DataSources, etc.)
  - J2EE Configuration (Jetty, OpenEJB, ActiveMQ)
  - System Services (Logging, URL Handler, XML Catalog)
  - Geronimo Kernel (Lifecycle, Config Manager, Repository)

- **Cold Swap**
  - EAR
  - EJB
  - WAR
  - RAR
  - Pico
  - Pico
  - Spring App
Running the Server

$ java -jar bin/server.jar <config>

$ java -jar bin/server.jar

- `<config>` is a specific configuration to boot
- With M2, default is to restart all configurations that were running on last clean shutdown
- Maven Geronimo deploy plugin simplifies this greatly
Geronimo Community

- MX4J
- Geronimo
- OpenEJB
- JOTM
- ActiveMQ
- TranQL
- cglib
- Jetty
- ActiveCluster
Jetty

- Well established web container
- Implements Servlet 2.4 specification
- Designed for embedding and high performance

- Why not Apache Tomcat yet?
  - No real reason, just hasn’t been done yet
OpenEJB

- Well established EJB container
- Upgraded from 1.1 to 2.1 specification
  - No, it doesn’t do EJB3 (yet)
- Fully integrated into Geronimo
- Optimized for highly concurrent workloads
ActiveMQ

- Yes, it’s another JMS implementation
  - Looked at OpenJMS and JORAM
  - Needed BSD, JMS1.1 and JCA1.5
- Designed for performance
  - NIO
  - Content based routing
  - Rules based routing (drools)
TranQL

- Framework for Persistence Frameworks
- Common data model across frameworks
- Multiple front ends:
  - EJB CMP, JDO, Castor, JDBC, Groovy DO
- Multiple back ends:
  - SQL92, SQL03, XML, LDAP, JCA
MX4J

- Well established JMX implementation
- Fully compliant JMX and JMX Remote API (JSR-160)
- Highly robust
- Widely used
ObjectWeb JOTM

- Java Open Transaction Manager
- Well established transaction manager
- Implements the JTA API
  - Full XA support
    - RMI/JRMP
    - RMI/IIOP
ActiveCluster

- Framework for creating cluster-based applications
- Small, simple implementation
- Pluggable service providers
  - Sockets
  - JMS
  - JGroups
  - Jabber
cglib

- High performance code generation library
- Well established class enhancer
- Provides runtime byte code manipulation
Security

- Built in at low level
  - Secure network protocols
  - Support security manager (sandbox)

- JAAS and JACC
  - Possible pluggable policy providers
    - HIPPA
    - SOX

- Kerberos integration (including .NET)

- Apache Directory integration (LDAP)
Transactions

- Lightweight Transaction Manager
  - XA Coordinator
  - XATerminator for JCA transaction import
  - Logging (HOWL) and basic recovery
- Future is JOTM integration
Project Status

- A top-level Apache project
- J2EE 1.4 CTS testing in progress
  - Three people working on certification
- Official unofficial release date: ?/??/2004
  - Open Source: It’s done, when it’s done
- M1 milestone release in May
- M2 milestone release in August
M2 Release Features

- Geronimo Kernel is stable
- Web integration complete
- OpenEJB stable
  - Mostly working, except some bits of CMP2.1
- ActiveMQ stable
- Transaction and Connector Complete
- Enabled Hot (re)-deployment
- To-do
  - Finish web services, application client, CORBA
Who Is Bruce Snyder?

- Independent software consultant
- Speaker at software conferences
  - JavaOne
  - ApacheCon
  - Colorado Software Summit
- The Castor Project (http://www.castor.org/)
- Apache Geronimo (http://geronimo.apache.org/)
- TranQL Project (http://tranql.codehaus.org/)
- Author
  - Castor Live (SourceBeat Publishing)
Project Contributors

- Aaron Mulder
- Bruce Snyder
- Davanum Srinivas
- David Jencks
- Gianny D’Amour
- Jacek Laskowski
- Jan Bartel
- Jeremy Boynes
- Richard Monson-Haefel
- Srinath Perera
- Alan Cabrera
- Dain Sundstrom
- David Blevins
- Geir Magnusson Jr
- Greg Wilkins
- James Strachan
- Jason Dillon
- Jules Gosnell
- Simone Bordot
Q&A

- Open up for Questions and Answers