

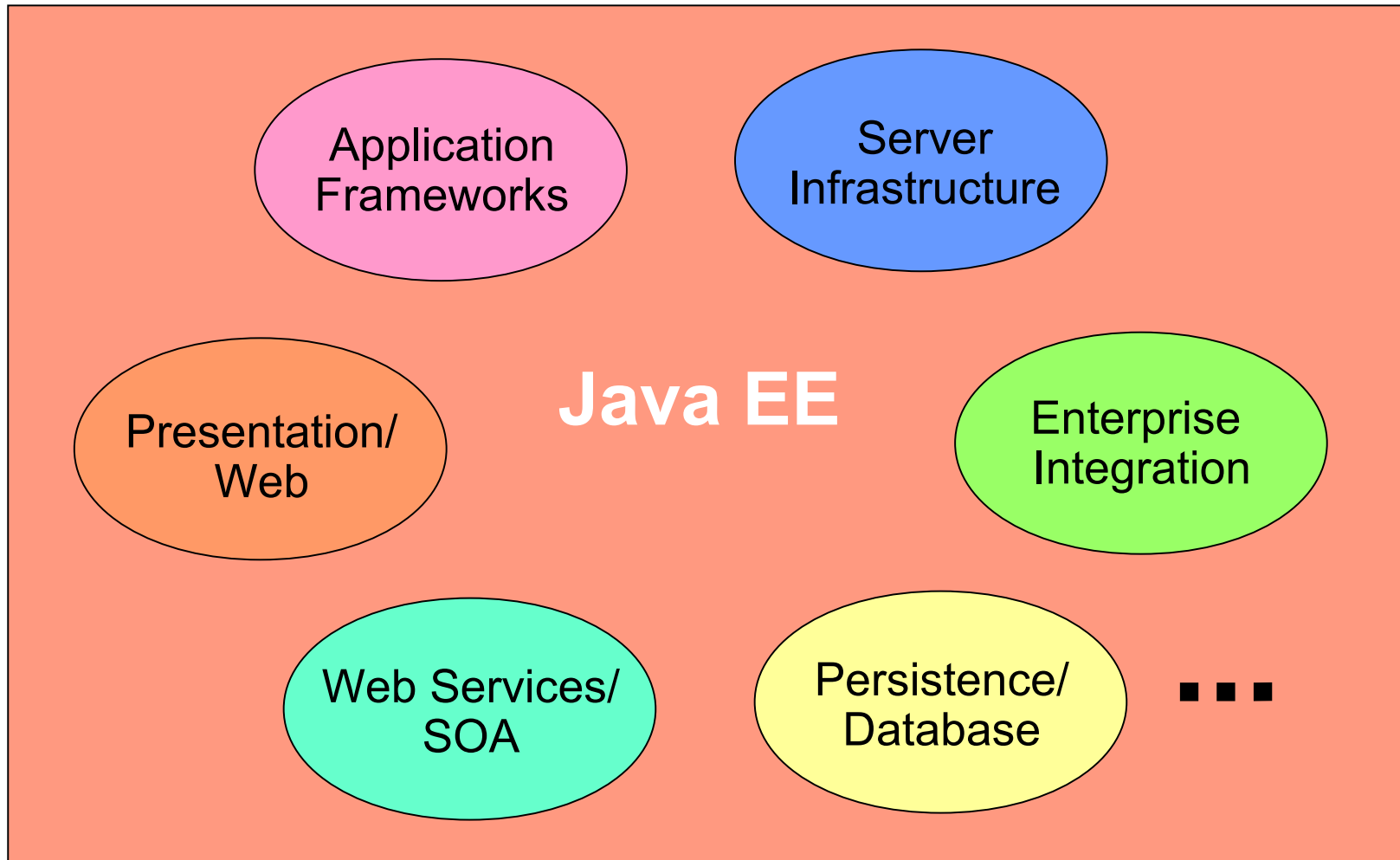


ORACLE[®]

Gemini – Shaping the Future of Enterprise Java

Mike Keith
michael.keith@oracle.com

What is Enterprise Java?



The Good and the Problems

- Horizontal choice
It's all (or at least most) or nothing
- 3rd party add-ons and frameworks
Non-standard ways of integrating
- Depth and maturity
Time-honored ways of coupling
- Standards and pluggability
Pluggability achieved using fragile mechanisms
- Reusable infrastructure
Sometimes an impediment

Solution

Modularity – the red pill of enlightenment?

- Leverage existing standards
- Decouple the technologies
- Integration mechanism uses module standards
- Infrastructure is available but does not impose
- It's about the granularity of Choice

Enterprise Modules Project

- Nicknamed “Gemini”
- Subproject of Eclipse Runtime Project
- An OSGi-based project, but its philosophy of “Modularity” is more general
- Collection of subprojects, each demonstrating modularity of a standard technology
- Dual-licensed – EPL and Apache v2

Questions about Gemini

- Is it an Application server?
- Why at Eclipse?
- Is it focused specifically at Equinox?
- How does it fit in with Java EE?
- Why are there two similar projects – Gemini and Aries?
- Who is using Gemini now?

Project Goals

Prime Directive:

Provide a home for modular adaptations of Java EE and enterprise technologies

- Visible, accessible and open
- Community input and community contribution
- Living implementations

Participation

- Initial committers are the companies that contributed the RI code to the OSGi Alliance
 - Oracle
 - SpringSource

- A few of the other interested parties:
 - EclipseSource
 - SAP
 - Swordfish
 - Red Hat
 - ECF
 - Infor

Specification Implementations

Gemini Subproject	OSGi Specification
Gemini Naming	JNDI Services
Gemini DBAccess	JDBC Service
Gemini Management	JMX Management Model
Gemini JPA	JPA Service
Gemini Blueprint	Blueprint Container
Gemini Web	Web Applications

Gemini DBAccess

- Implementation of OSGi JDBC Service specification
- Exposes JDBC driver as a service
- Depending on existing driver OSGi support, it either wraps a driver JAR or imports it
- Create implementations for a number of different databases
- Initial support for Derby embedded and client/server, Java DB

Gemini Naming

- Provides implementation of JNDI that follows the rules of OSGi JNDI Services Specification
- Useful for traditional Java EE applications that have JNDI lookups throughout the code
- Not expected to *replace* service lookup or service instance injection

Gemini Management

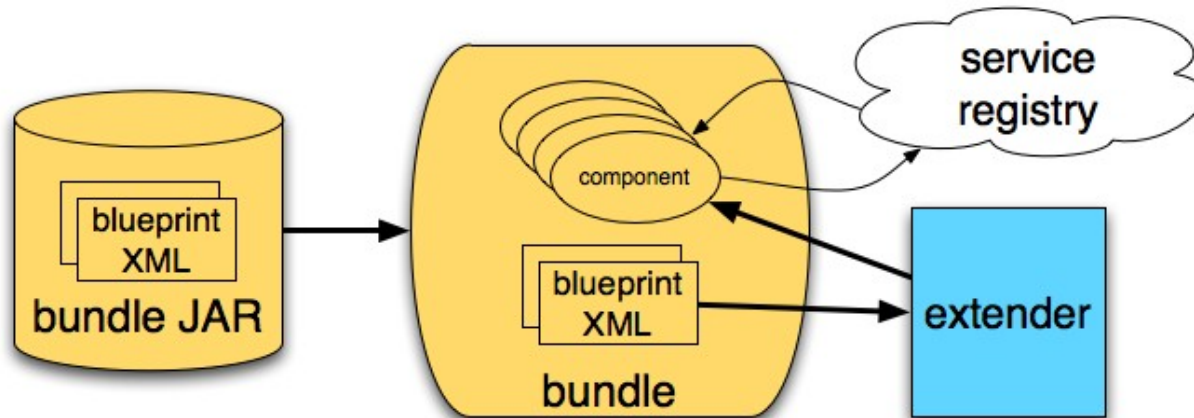
- Implementation of EEG JMX Management Model Specification
- Provides the MBeans for managing the framework via JMX
- Thin layer that hooks into the existing management interfaces on framework classes
- Minimal future development (only needs to change when new APIs are added to OSGi)

Gemini JPA

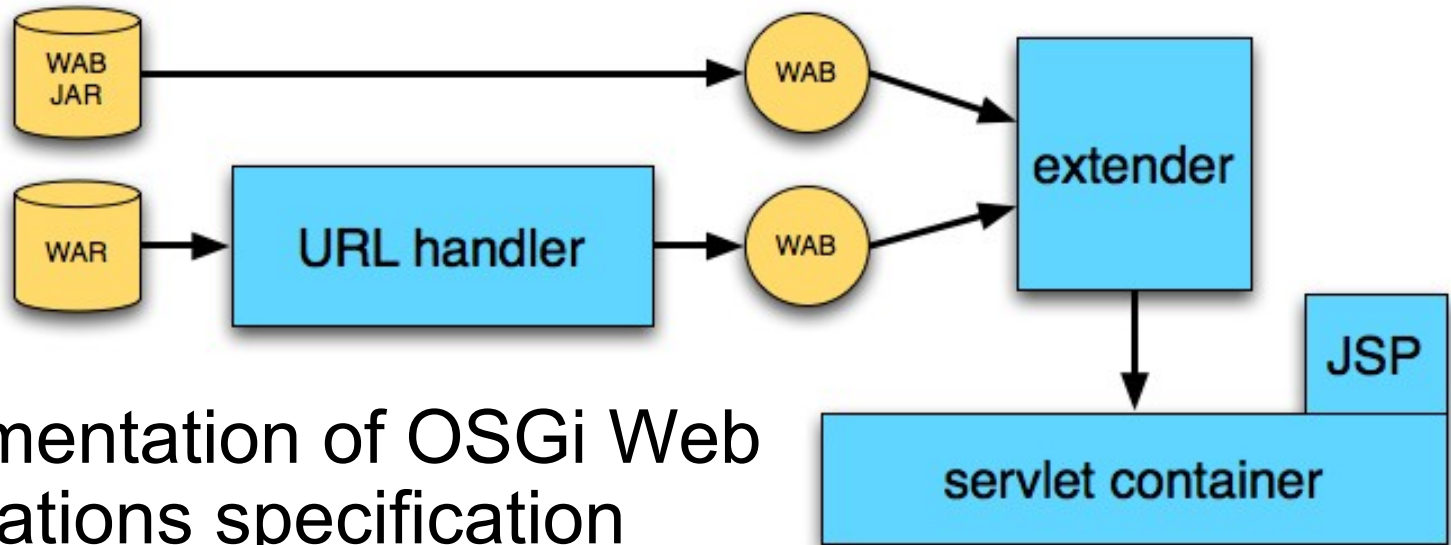
- Code to provide support for the OSGi JPA Service
- Library can be used by other providers to more quickly be able to implement the spec
- Hooks into EclipseLink provider
- EclipseLink already supported OSGi for 2 years, but now being retrofitted to use the current spec
- Innovation in areas that spec does not cover

Gemini Blueprint

- Implementation of OSGi Blueprint Container
 - Based on Spring DM project
 - Uses Spring framework
- Used by Eclipse Virgo project and others
- Developed by VMware (SpringSource) and Oracle/BFA





Gemini Web



- Implementation of OSGi Web Applications specification
 - Servlet 2.5, JSP 2.1
 - Uses Apache Tomcat, Jasper
- Used by Eclipse Virgo project and others
- Developed by VMware (SpringSource)

Status

- ✓ Project proposal and acceptance
- ✓ Resource and Repository Provisioning
- ✓ Code completed and tested
-  Code CQ'ed and committed to Gemini
-  Project makeup

<http://eclipse.org/proposals/gemini>

<http://www.eclipse.org/gemini>

Future

- Use the project as a place for people to access the specs separately or together
- Experiment and innovate in additional technologies
- Leverage the experience as input to next round of specifications
- Virgo will show how Gemini can be used in a container