eclipsIST 2007
Türk eclipse zirvesi
Turkish eclipse summit

OSGi and Eclipse Equinox explained

Martin Lippert, akquinet agile GmbH
lippert@acm.org

A Few Words about Myself…

- Martin Lippert
  - Senior IT consultant at akquinet agile GmbH, Germany
  - lippert@acm.org
- Focus
  - Agile software development
  - Refactoring
  - Eclipse technology
- Equinox incubator committer

Overview

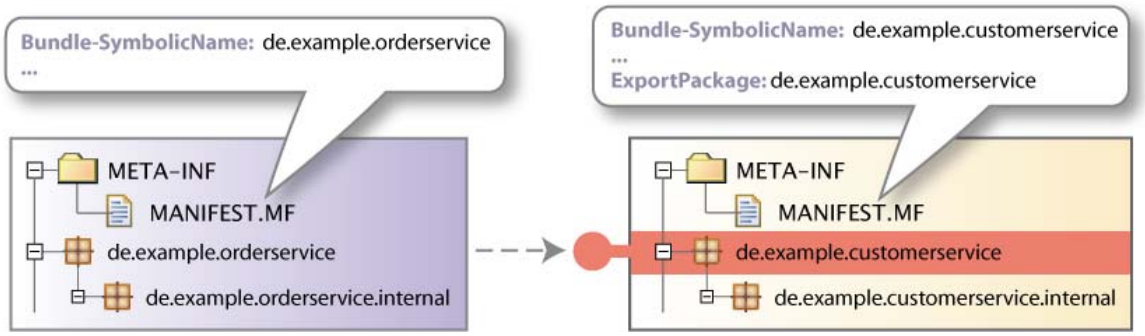
- An OSGi Overview
- Eclipse Equinox
- Use Cases and Examples
- More Cool Things using Equinox

OSG – What?

- OSGi™:
  - **„A dynamic module system for Java“**

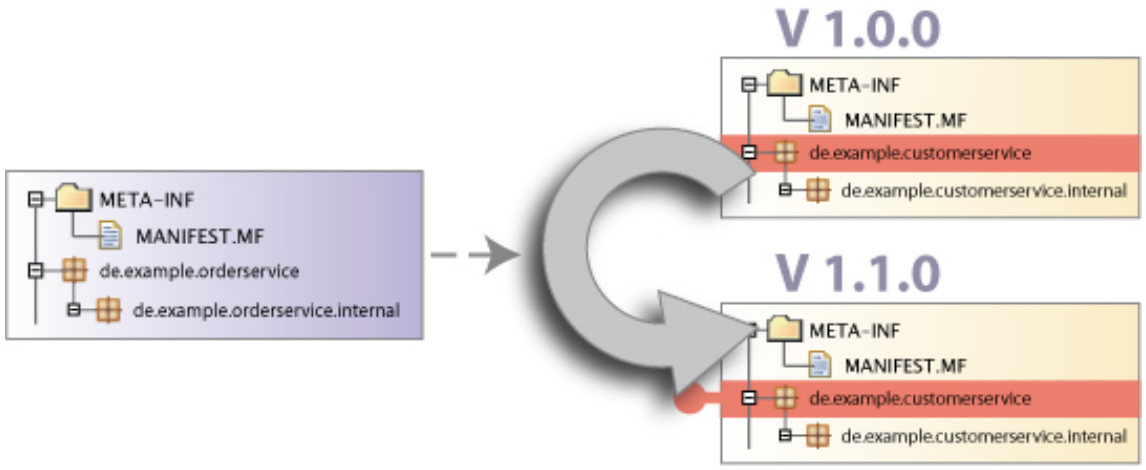
OSGi is …

- … a module system for Java that allows the definition of …
  - **Modules** (called „bundles“),
  - **Visibility** of the bundle contents (public-API vs. private-API)
  - **Dependencies** between modules
  - **Versions** of modules



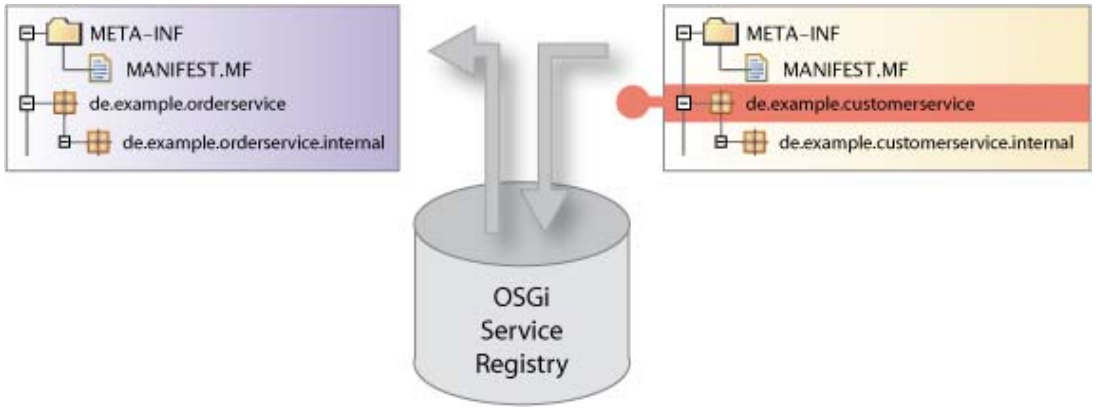
OSGi is …

- … dynamic
  - Bundles can be installed, started, stopped, uninstalled and updated at runtime



OSGi is …

- … service oriented
  - Bundles can publish services (dynamically)
  - Bundles can find and bind to services through a service registry
  - The runtime allows services to appear and disappear at runtime



What does OSGi look like? (Low Level)

Identification

```
Bundle-SymbolicName: org.eclipse.equinox.registry
Bundle-Version: 3.2.100.v20060918
Bundle-Name: Eclipse Extension Registry
Bundle-Vendor: Eclipse.org
```

Classpath

```
Bundle-ClassPath: ., someOtherJar.jar
```

Lifecycle

```
Bundle-Activator: org.eclipse.core.internal.registry.osgi.Activator
```

Dependencies

```
Import-Package: javax.xml.parsers,
 org.xml.sax,
 org.osgi.framework;version=1.3
Require-Bundle: org.eclipse.equinox.common;bundle-version="[3.2.0,4.0.0)"
Bundle-RequiredExecutionEnvironment: CDC-1.0/Foundation-1.0,J2SE-1.3
```

Exports

```
Export-Package: org.eclipse.equinox.registry
```

OSGi, Eclipse and Equinox

- OSGi Alliance
  - Produces open specifications for runtime environments
  - Traditional focus on embedded (home gateway, telematics, …)
  - Broadening scope – Mobile devices, desktops, enterprise and servers
  - Several open source implementations including Equinox
- Eclipse
  - Eclipse 3.0 saw the rise of Eclipse the Rich Client Platform (RCP)
  - Needed a standard, open, flexible, dynamic, modular runtime to replace the home-grown Eclipse runtime
  - Eclipse has been OSGi-based since 3.0 (3 years, 3 releases)
- Equinox
  - Eclipse OSGi implementation
  - OSGi R4.0 and R4.1 reference implementation
  - Consistent component story across computing environments/domains

Implementations

- Open source implementations
  - Eclipse Equinox (http://www.eclipse.org/equinox/)
  - Apache Felix (http://cwiki.apache.org/FELIX/index.html)
  - Knopflerfish (http://www.knopflerfish.org/)
  - ProSyst mBedded Server Equinox Edition (http://www.prosyst.com/products/osgi_se_equi_ed.html)
- Commercial implementations
  - ProSyst (http://www.prosyst.com/)
  - Knopflerfish Pro (http://www.gatespacetelematics.com/)

(not necessarily complete)

Equinox adds More…

- The Equinox projects is more than just an OSGi R4.1 implementation
  - Adds the Eclipse Extension Point mechanism
- Incubator work in different areas
  - AOP
  - Server-side OSGi (graduated)
  - Management
  - Provisioning
  - …

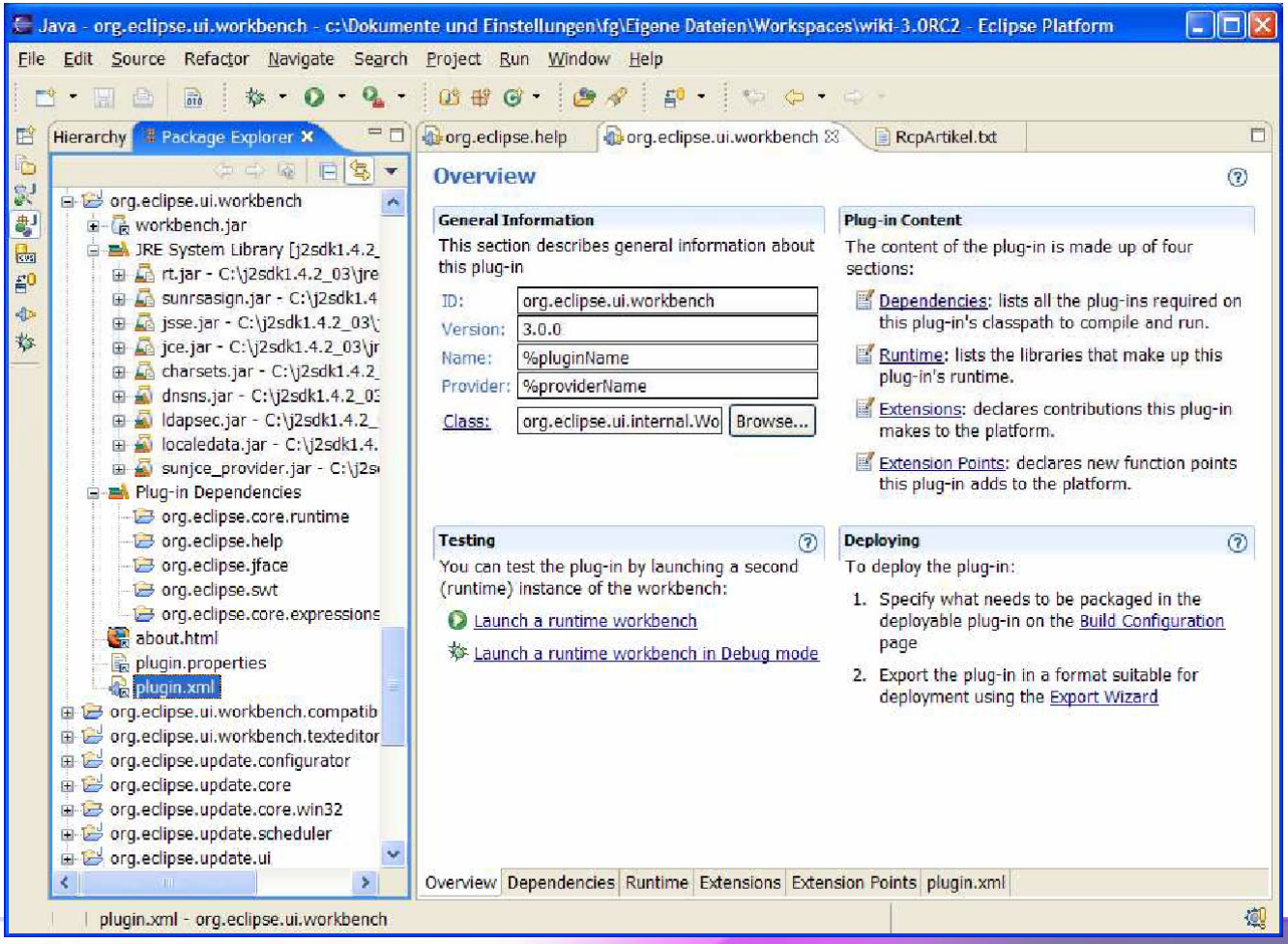
Extension Registry

- Build-in extensibility support
  - Allows the definition of extension-points and extensions
  - Some kind of minimal component model (**how** do bundles interact aside of API calls and class reuse)
- Widely known for the Eclipse SDK
  - Adding views, editors, etc. to the workbench
- But a very powerful mechanism for a wide range of extensions
  - Self-build extension points for all possible areas of an application
  - Very flexible system architectures

Where is OSGi & Equinox used?

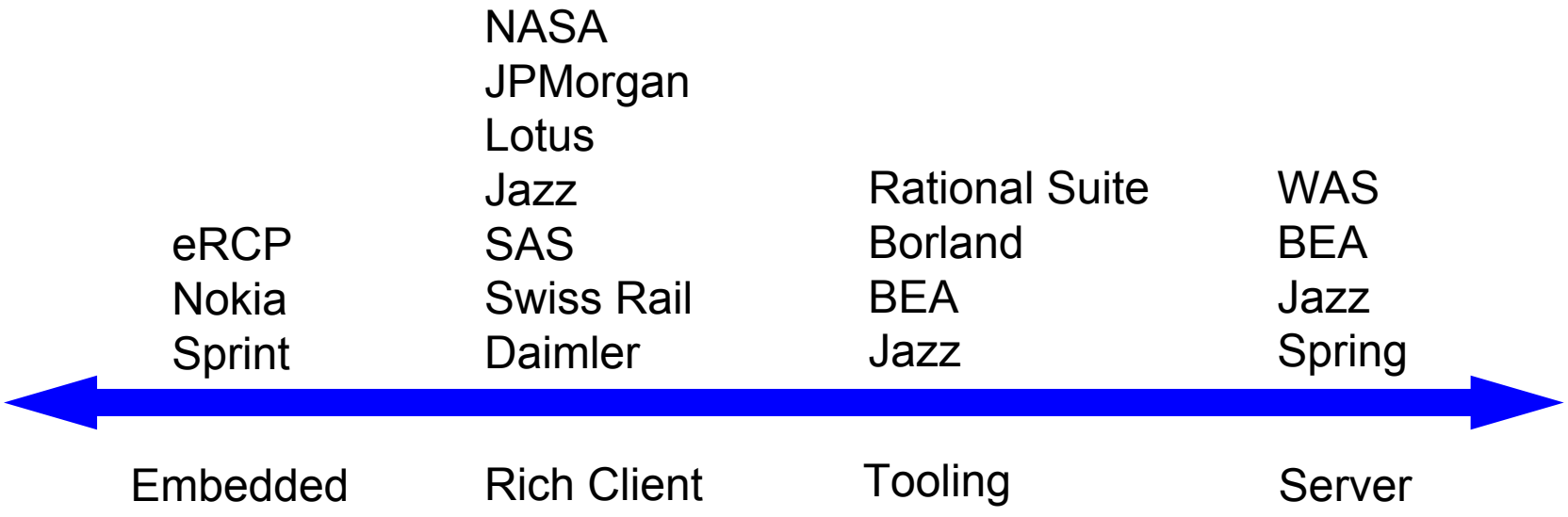
The Most Famous Use Case

- Plugging into the Eclipse SDK



Range of Use

- Eclipse (i.e., Equinox) as a modular runtime
- Consistent programming model: embedded to server
- Reuse components across the spectrum
- Some examples…



Eclipse and Open Source Projects

- Eclipse
  - Equinox
  - Rich Ajax Platform
  - Rich Server Platform – UI
  - Communications Framework
  - Corona
  - Enterprise Component Project
- Apache
  - Felix
  - Directory
  - Cocoon
  - James
  - Geronimo
- Spring community integrating with OSGi

OSGi as a Server Platform

- Modular
- Dynamic
- Small
- Fast
- Flexible

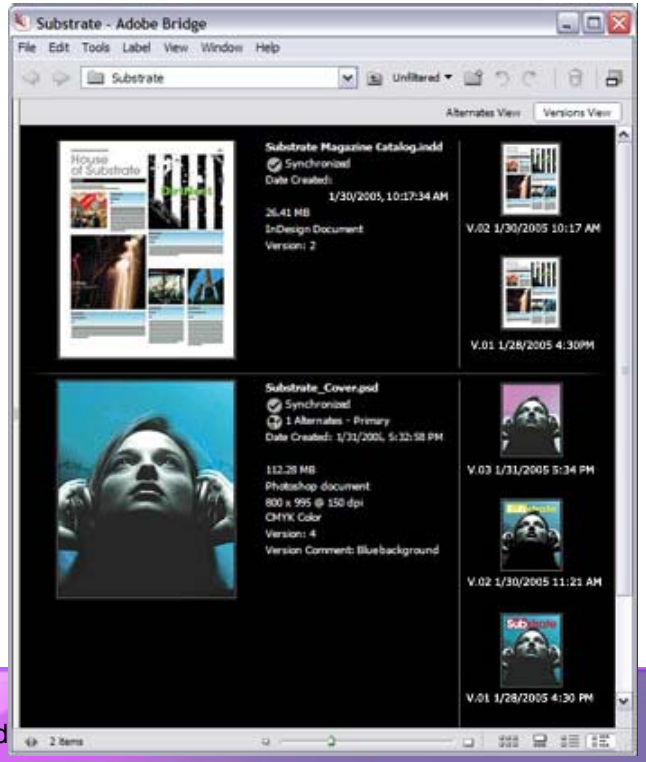
Ideal Server Platform

WebSphere and BEA

- WAS 6.1 is based on Equinox
- BEA's micro kernel architecture based on Equinox

Adobe Version Cue

- Embedded client/server document management system
- Project management functionality for small workgroups
  - version control, file collaboration, streamlined reviews
- Eclipse offers
  - Multi-platform support
  - Strong, dynamic, standard component model (Equinox/OSGi)
  - Configuration management
  - Reuse components on clients and servers



IBM Rational Jazz Platform

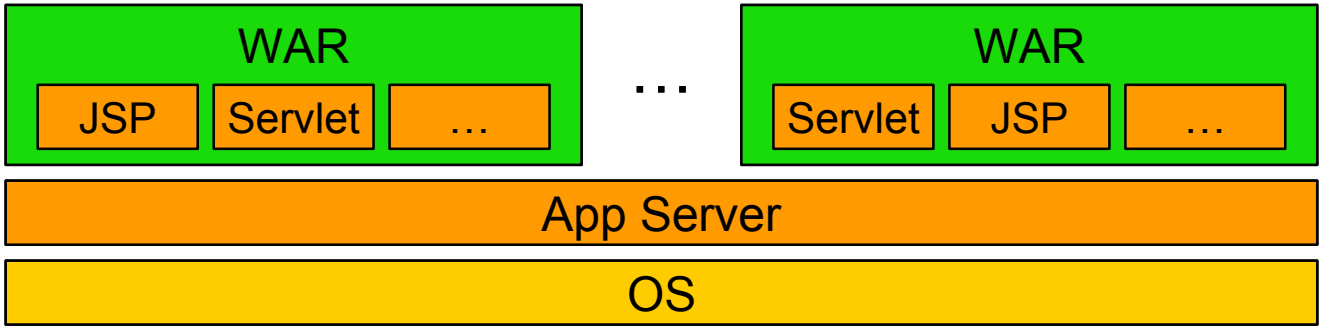
- Team Collaboration platform
- Eclipse/Equinox on Client and Server
- Server serves
  - Traditional content
  - Dynamic, modular Dojo/Ajax
  - Web Services with DB2 etc backends
- Same programming model on client and server
- Run same components on client and server

Server-side Variations

- Traditional App Server
- Equinox nested in an App Server
- Raw Equinox
- Equinox nested in another Equinox
- App Server on Equinox

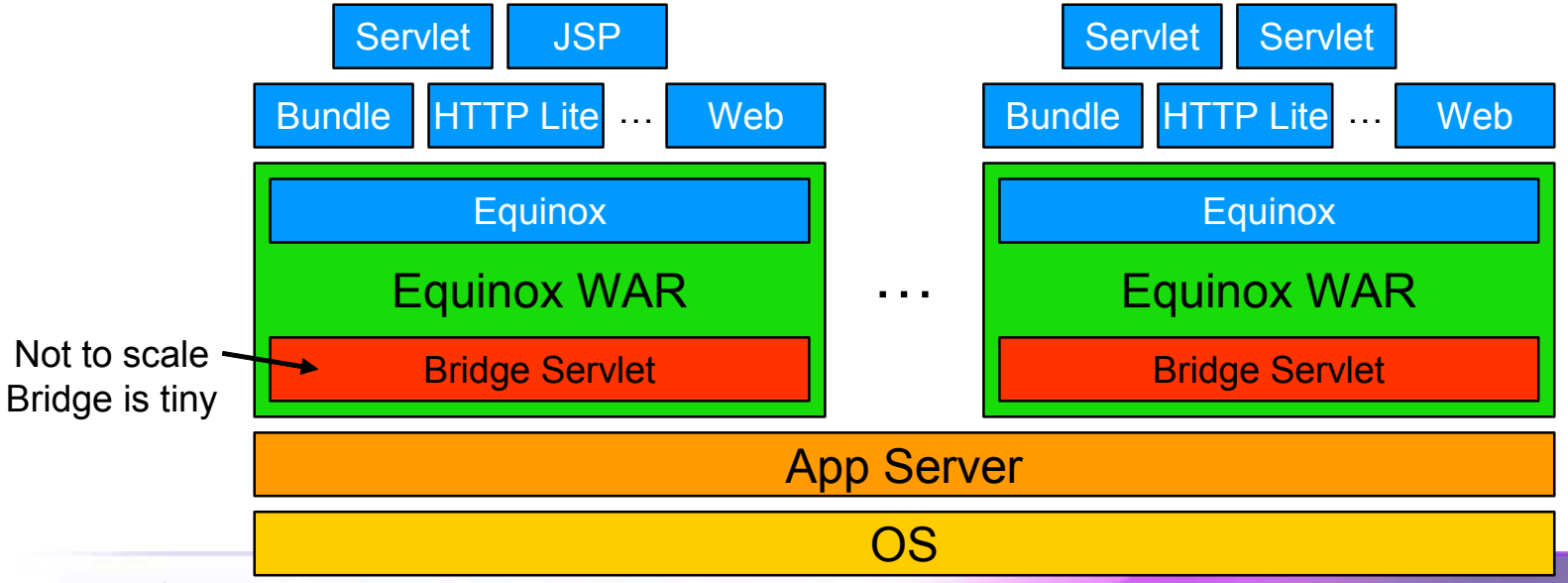
Traditional Server Example

- Server function (e.g., servlets) packaged in a WAR
- Application Install/Update/Manage whole WARs
- Application isolation
- No OSGi



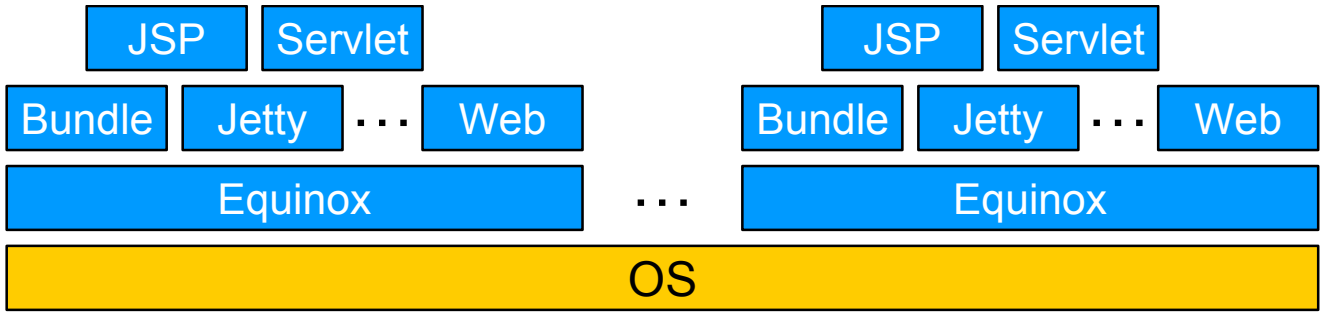
Equinox in an App Server

- Bridge servlet hosts Equinox in traditional App Server
- *Application* isolation
- Integration with existing infrastructure
- Forwarding (Lite) HTTP Service
  - Expose underlying App Server capabilities
- Add application function as bundles or servlets or JSPs, …
- Install/Update/Manage “WAR” by managing bundles



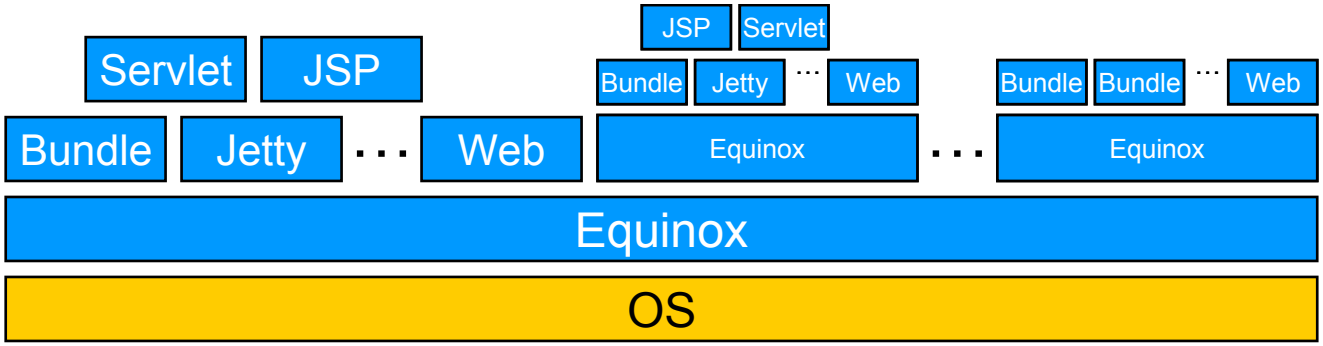
Raw Equinox

- Run Equinox directly
- *Process* isolation
- HTTP Service (e.g., embedded Jetty bundle)
- Add application function as bundles, servlets, JSPs, AJAX, …
- Install/Update/Manage server by managing bundles
- Web Services



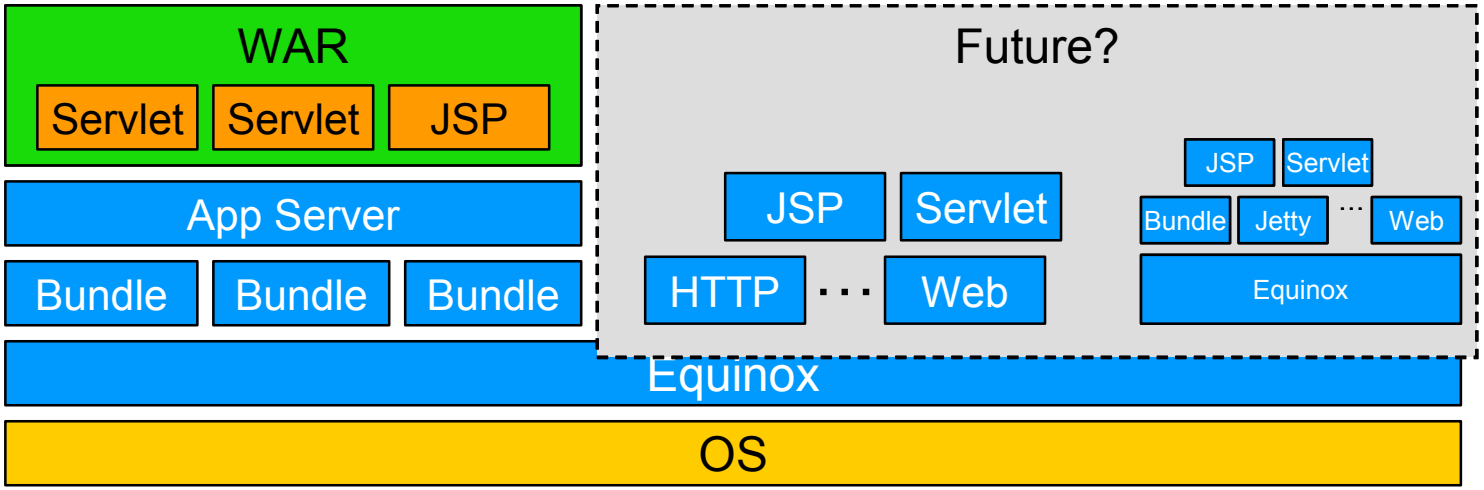
Equinox nested in Equinox

- Run Equinox directly, nest other Equinox instances
- *Nested framework* isolation
- HTTP Service (e.g., embedded Jetty bundle)
- Add server function as bundles, servlets, JSPs, AJAX, …
- Install/Update/Manage server by managing bundles
- Web Services, …



App Server on Equinox

- Add ***App Server*** function as bundles
  - For example, Tomcat, Jetty, IBM WebSphere …
- Tailor server configuration to match application needs
  - Dynamically
- Potential to combine all other approaches!



Advantages

- Incremental update of server function
- Run multiple versions simultaneously
- Individual configuration and management
- Accommodate disparate application prerequisites
- Class loading performance
- Share components across client and server
  - E.g., support disconnected mode

Technical Challenges

- Classloaders
  - Classloader parenting
  - Isolate nested entities from outside world
  - Context Classloader use
- System property isolation
- Statics and factories in the JRE
  - URLStreamHandlerFactory can only be set once

Cool stuff you can do

- The Equinox runtime is extensible
  - Hooks for enhancing the runtime behavior
  - But be aware of the OSGi concepts
    - it's easy to break things
- Modifying, for example
  - Classloading
  - Manifest contents
  - Bundle contents
  - Much more…

Example: J9 Class Sharing

- IBM J9 Virtual Machine offers class sharing across multiple running VMs
  - Caches loaded classes for faster startup
  - Shares loaded classes across VMs for smaller footprint
- Equinox extension allows to make use of this
  - A classloader hook that calls the J9 API in addition to the JDK classloading

Example: Transformers

- Transformers are additions to the runtime to modify the contents of bundles at load-time
- For example removing extensions from the plugin.xml with regards to the active user
  - Can be used to customize the app depending on the active user

AOP and OSGi

- Equinox Aspect incubator project
  - Adds AspectJ load-time weaving to the Equinox runtime
  - Allows you to weave aspects into bundles at class-loading time
- Modularity for aspects and bundles combined
  - Case 1:
    - Provide abstract aspects as bundles
  - Case 2:
    - Add aspects to the system and weave them into existing bundles

Spring and OSGi

- Spring is a de-facto standard for enterprise Java apps
  - Dependency injection
  - AOP
  - Many technology abstractions and implementations

- Spring-OSGi:
  - New subproject
  - Allows easy combination of Spring and OSGi
  - Very promising for server-side and client-side apps

Conclusions

- OSGi is small, simple, easy and fast
- It's an ideal platform for general app development
  - Even if you don't use any of the other Eclipse technologies or platforms
- Watch JSR 291:
  - "Dynamic Component Support for Java SE"

Thank you for your attention

- Questions always welcome!!!

Martin Lippert
lippert@acm.org
http://www.martinlippert.org/

Special thanks to Jeff McAffer for the material
(see the Copyright statements on the slides)