



Adobe

Building Modular Enterprise Applications

Carsten Ziegeler | Adobe Research Switzerland



- RnD Team at Adobe Research Switzerland
- Member of the Apache Software Foundation
 - Apache Felix and Apache Sling (PMC and committer)
 - And other Apache projects
- OSGi Core Platform and Enterprise Expert Groups
- Member of the OSGi Board
- Book / article author, technical reviewer, conference speaker

Motivation

- Enjoy the power of modularity
- OSGi is everywhere
 - From embedded systems to enterprise applications and THE CLOUD
- Adapting a technology is always a challenge
 - Common pitfalls
 - Potential solutions

But wait a second...

- Adobe & OSGi ?

But wait a second...

- Adobe & OSGi ?



Adobe & OSGi & Open Source & Open Development

- Using OSGi in various products
- Strategic member of the OSGi Alliance
 - Board of directors
- Participating in CPEG and EEG
- Highly contributing to open source
 - Especially Apache Software Foundation
- Participating in open standards
- Embracing open development



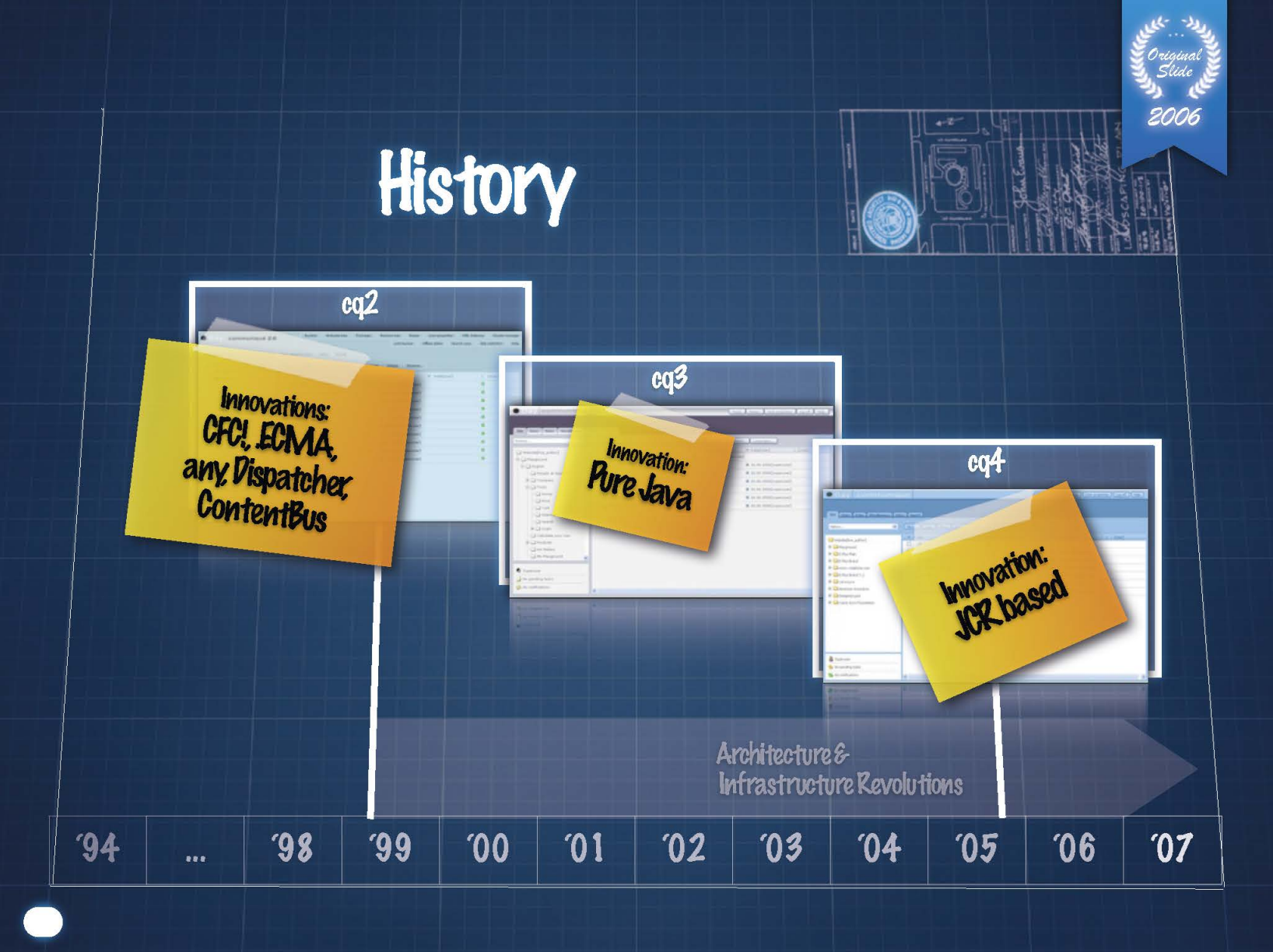
DELIVER RELEVANT EXPERIENCES EVERYWHERE YOUR CUSTOMERS ARE.

Customers connect with your brand on multiple devices. Adobe can help you create and manage an interactive digital experience that keeps them engaged on every screen.

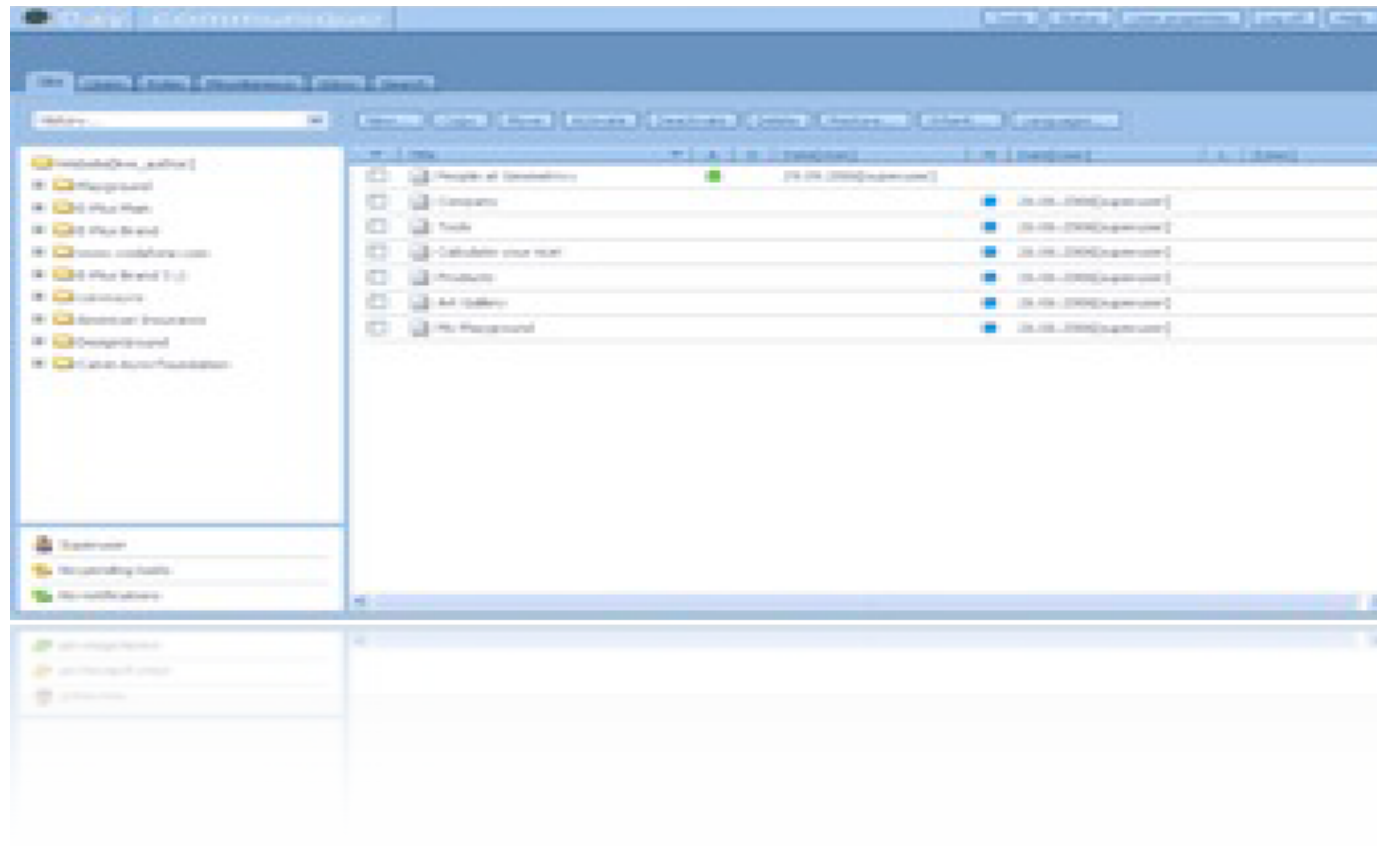


[See how Experience Manager works >](#)

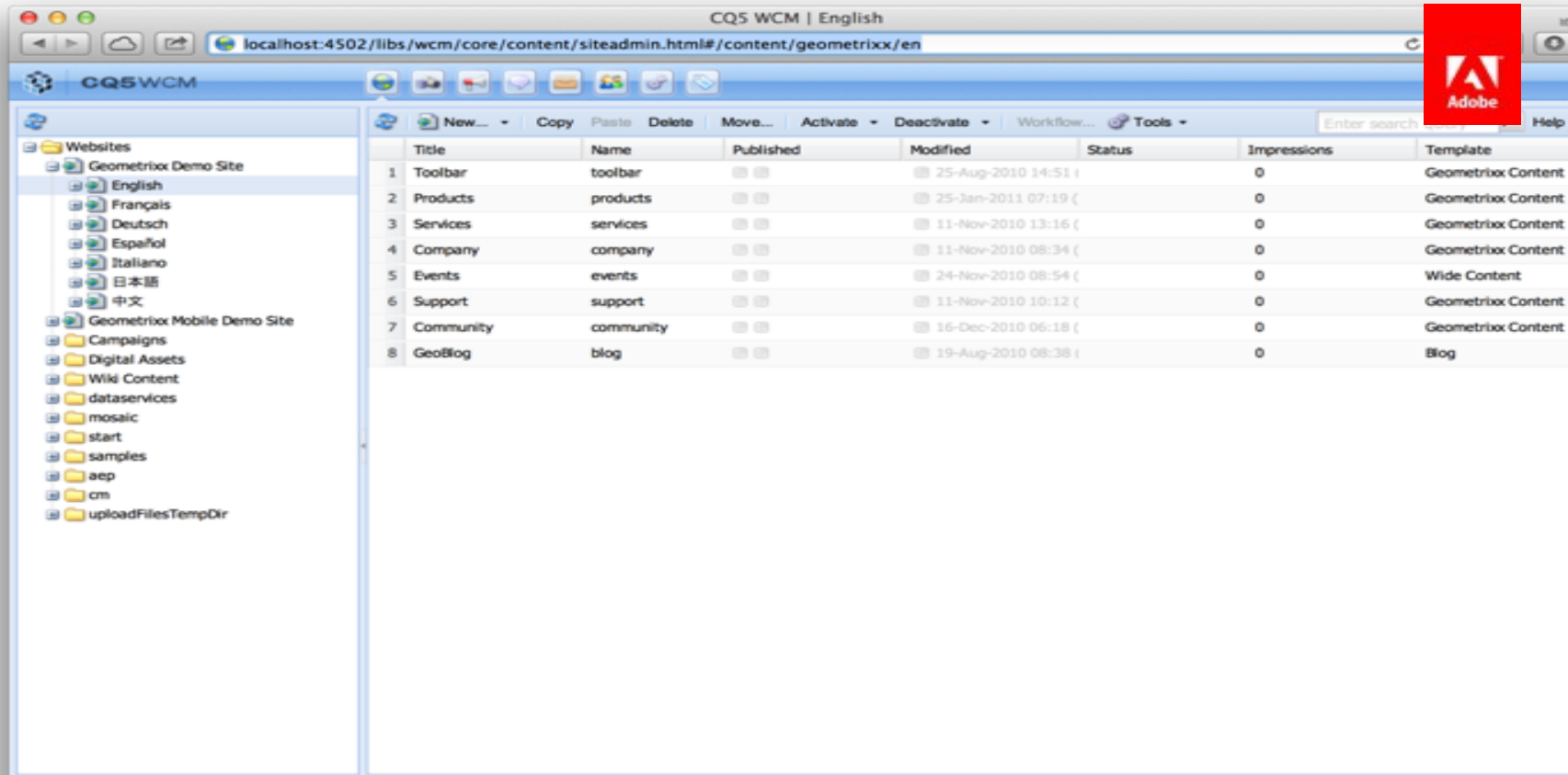
Back to the Future: Web Content Management Code History



Back in the Days...CQ 4



OSGi based Enterprise Web Content Management – Adobe CQ5

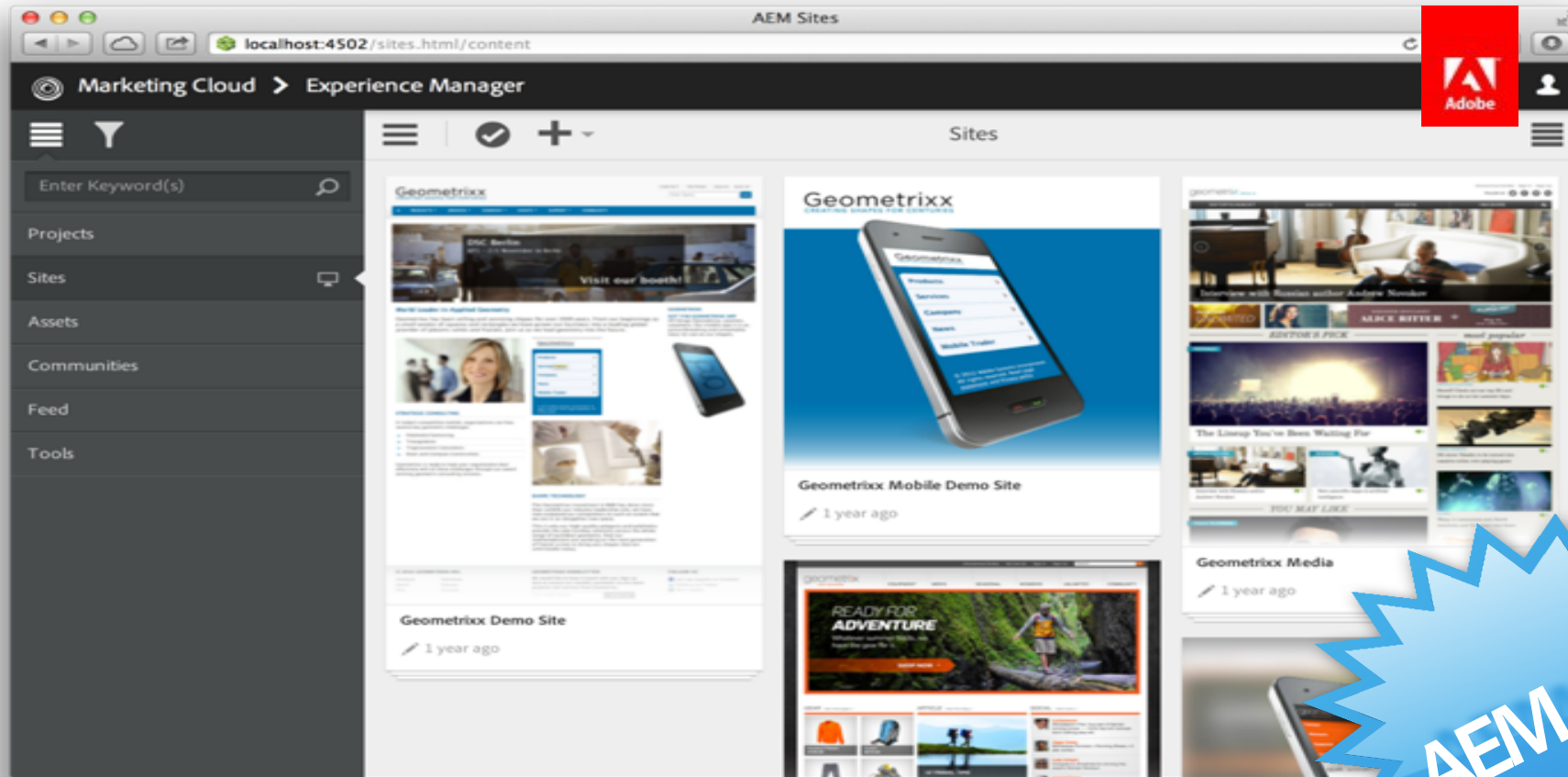


The screenshot displays the Adobe CQ5 WCM interface. The browser address bar shows the URL: localhost:4502/libs/wcm/core/content/siteadmin.html#/content/geometrix/en. The interface includes a navigation pane on the left with a tree view of the site structure, a main content area with a table of content items, and a top navigation bar with various menu options.

Content List Table:

| | Title | Name | Published | Modified | Status | Impressions | Template |
|---|-----------|-----------|--------------------------|-------------------|--------|-------------|----------------------|
| 1 | Toolbar | toolbar | <input type="checkbox"/> | 25-Aug-2010 14:51 | | 0 | Geometrix Content .. |
| 2 | Products | products | <input type="checkbox"/> | 25-Jan-2011 07:19 | | 0 | Geometrix Content .. |
| 3 | Services | services | <input type="checkbox"/> | 11-Nov-2010 13:16 | | 0 | Geometrix Content .. |
| 4 | Company | company | <input type="checkbox"/> | 11-Nov-2010 08:34 | | 0 | Geometrix Content .. |
| 5 | Events | events | <input type="checkbox"/> | 24-Nov-2010 08:54 | | 0 | Wide Content |
| 6 | Support | support | <input type="checkbox"/> | 11-Nov-2010 10:12 | | 0 | Geometrix Content .. |
| 7 | Community | community | <input type="checkbox"/> | 16-Dec-2010 06:18 | | 0 | Geometrix Content .. |
| 8 | GeoBlog | blog | <input type="checkbox"/> | 19-Aug-2010 08:38 | | 0 | Blog |

Adobe Experience Manager



Motivations for Going Modular (2006)

- Aged code base
 - Monolithic
 - Highly entangled
 - Growing complexity and functionality
- No clear separation between API and implementation
- Increasing customer base
- Team growth

Modularity is Key

- Manage growing complexity
- Support dynamic extensibility
- Define contracts between modules

Potential Solutions (2006)

- Spring?
- OSGi?
- Other open source component frameworks?
- DIY?

Benefits of OSGi

- OSGi modules = bundles
 - jars with strict enforcements
 - Exported packages with version
 - Imported packages with version range
 - Private packages
- Service Registry
- Separation of Concerns
- Loose coupling

Modularity - How do we get there? (2006)

- Start with a single big bundle?
- Embedding OSGi framework?
- Complete rewrite?

The Fun Way : Complete Rewrite

- New architecture
 - New backend (Apache Jackrabbit)
 - New web application framework (Apache Sling)
 - New UI
- Build modules from the start
- Control evolution
- (Provide tools for migration from old solution)

OSGi and Open Source (2006)

- Apache Felix
- Equinox

Framework runs and now?

- We need bundles!
- Build tools
 - Ant? (BND)
 - Maven? (Apache Felix Maven Bundle Plugin)

Maven Bundle Plugin (2007)

- Specifying all OSGi metadata in the POM
 - Exported packages
 - Private packages
- Imported packages and versions get calculated
- Parent POM to define rules for bundle symbolic name

Maven Bundle Plugin – First problems

- Specifying all OSGi metadata in the POM
 - Exported packages
 - Private packages
- Imported packages and versions get calculated
- Parent POM to define rules for bundle symbolic name
- Exported package version = POM version
- New private package needs updating the pom
- Slight difference in version handling between OSGi and Maven (The SNAPSHOT problem)
 - $2.0.1 < (\text{OSGi}) 2.0.1\text{-SNAPSHOT} < (\text{Maven}) 2.0.1$
- Semantic versioning?

Excursion on Versioning

- Marketing Versions vs. Engineering Versions
- Versioning Package Exports
 - Engineering Versions
 - Apply versions !
 - Semantic Versioning crucial
- Bundle Versions
 - Grey area
 - Semantic Versioning helpful
- Application Versions
 - Marketing Versions

Versioning Best Practice

- Package Exports
 - Always apply a version to package exports
 - Start with 1.0
 - Only increment when changing the API
 - Think hard before increasing the major version !
- Package Imports
 - Ensure proper version range import

Semantic Versioning

- Nice whitepaper on the OSGi homepage
- Versioning policy for exported packages
- OSGi versions: <major>.<minor>.<micro>.<qualifier>
- Updating package versions:
 - Fix/patch (no API change) : update micro
 - Extend API (affects implementers, not clients) : update minor
 - API breakage: update major

Versioning Annotations

- BND or new with OSGi R6
- Annotations for documenting semantic versioning information
 - @Version
 - @ProviderType
 - @ConsumerType

Versioning Best Practice (cont.)

- Bundle Versions
 - Always use and manage bundle version
 - Align bundle version with Maven version
 - Use odd micro versions for SNAPSHOTs
 - e.g. 0.0.1-SNAPSHOT, 1.3.5-SNAPSHOT
 - Use even micro versions for release
 - e.g. 1.0.0, 1.3.6
- Dependencies
 - Always use least required version for dependencies

Maven Bundle Plugin – Today

- Specifying all OSGi metadata in the POM
 - Exported packages
 - Private packages
- Imported packages and versions get calculated
- Parent POM to define rules for bundle symbolic name
- Parent POM to define rules for bundle symbolic name and private packages
- No package versioning information in POM (annotations)
- Follow semantic versioning
- Odd/even versioning policy
- Usually no OSGi metadata in the POM

What about Components and Services (2007)?

- Framework API is usually too low level
- Make development as easy as possible
 - Try to hide dynamics
- And there are OSGi compendium specs
 - Declarative Services
 - Configuration Admin
 - Metatype

- Declarative Services (OSGi Compendium Spec)
 - Defines Service Component Runtime (SCR)
 - Declarative component model
 - XML descriptor for components including
 - Provided services
 - References to other services
 - Configuration properties

- OSGi Configuration Admin
 - “The” solution to handle configurations
 - Configuration Manager
 - Persistence storage
 - **Service API** to retrieve/update/remove configuration
- OSGi Metatype Service
 - Description of bundle metadata
 - Description of service configurations
 - Property type, name, and description

What about Components and Services (2007)?

- Framework API is usually too low level
- And there are OSGi compendium specs
 - Declarative Services
 - Configuration Admin
 - Metatype
- No (Apache) open source implementations available
- Nearly no tooling
- Declarative Services uses XML descriptors
- Metatype uses XML descriptors and property files

DIY – Donations to Apache Felix

- Apache Felix SCR (Declarative Service Implementation)
- Apache Felix Config Admin
- Apache Felix Metatype
- Apache Felix Preferences
- Apache Felix SCR Tooling
 - Java annotations for generating DS and Metatype descriptors
 - Maven, ANT, BND

- Declarative Services (OSGi Compendium Spec)
 - Defines Service Component Runtime (SCR)
 - Apache Felix SCR Annotations (DS annotations)
- Some advantages (in combination with the tooling)
 - POJO style
 - Declarative
 - Single source: just the Java code, no XML etc.
 - "Integration" with Configuration Admin and Metatype Service

- Service interface
 - Public (if exported for other bundles)
 - Versioned through package version (Semantic versioning)
 - Private for internal services (sometimes useful)
- Component / service implementation
 - Always private

My First Component

```
package com.adobe.osgitraining.impl;  
  
import org.apache.felix.scr.annotations.Component;  
  
@Component  
public class MyComponent {  
  
}
```

Component Lifecycle

```
package com.adobe.osgitraining.impl;

import org.apache.felix.scr.annotations.Activate;
import org.apache.felix.scr.annotations.Component;
import org.apache.felix.scr.annotations.Deactivate;

@Component
public class MyComponent {

    @Activate
    protected void activate() {
        // do something
    }

    @Deactivate
    protected void deactivate() {
        // do something
    }
}
```


Providing a Service

```
package com.adobe.osgitraining.impl;  
  
import org.apache.felix.scr.annotations.Component;  
import org.apache.felix.scr.annotations.Service;  
import org.osgi.service.event.EventHandler;  
  
@Component  
@Service(value=EventHandler.class)  
public class MyComponent implements EventHandler {  
  
    ...  
}
```

Using a Service

```
package com.adobe.osgitraining.impl;

import org.apache.felix.scr.annotations.Component;
import org.apache.felix.scr.annotations.Service;
import org.osgi.service.event.EventHandler;

@Component
@Service(value=EventHandler.class)
public class MyComponent implements EventHandler {

    @Reference
    private ThreadPool threadPool;

    ...
}
```

Using an Optional Service

```
package com.adobe.osgitraining.impl;

import org.apache.felix.scr.annotations.Component;
import org.apache.felix.scr.annotations.Service;
import org.osgi.service.event.EventHandler;

@Component
@Service(value=EventHandler.class)
public class MyComponent implements EventHandler {

    @Reference(cardinality=ReferenceCardinality.OPTIONAL_UNARY,
               policy=ReferencePolicy.DYNAMIC)
    private volatile ThreadPool threadPool;

    @Reference(cardinality=ReferenceCardinality.MANDATORY_UNARY)
    private Distributor distributor;
```

Component Properties -> Service Properties

```
import org.apache.sling.commons.osgi.PropertiesUtil;

@Component
@Service(value=EventHandler.class)
@Properties({
    @Property(name="service.vendor", value="Who?"),
    @Property(name="service.ranking", intValue=500)
})
public class DistributingEventHandler
    implements EventHandler {
```

- OSGi Configuration Admin
 - “The” solution to handle configurations
 - Configuration Manager
 - Persistence storage
 - **Service API** to retrieve/update/remove configuration
- Integration with Declarative Services
 - Configuration changes are propagated to the components
 - Configurations are stored using the **PID**

Configuration – Supports Configuration Admin

- Provided map contains
 - Configuration properties from Configuration Admin
 - Defined component properties

```
@Activate  
protected void activate(final Map<String, Object> props) {  
    ...  
}
```

Configuration – Supports Configuration Admin

```
import org.apache.sling.commons.osgi.PropertiesUtil;

@Component
@Service(value=EventHandler.class)
@Properties({
    @Property(name="event.topics", value="*", propertyPrivate=true),
    @Property(name="event.filter", value="(event.distribute=*)",
        propertyPrivate=true)
})
public class DistributingEventHandler
    implements EventHandler {

    private static final int DEFAULT_CLEANUP_PERIOD = 15;

    @Property(intValue=DEFAULT_CLEANUP_PERIOD)
    private static final String PROP_CLEANUP_PERIOD = "cleanup.period";

    private int cleanupPeriod;

    @Activate
    protected void activate(final Map<String, Object> props) {
        this.cleanupPeriod =
            PropertiesUtil.toInteger(props.get(PROP_CLEANUP_PERIOD));
    }
}
```

Configuration Update

```
import org.apache.sling.commons.osgi.OsgiUtil;

public class DistributingEventHandler
    implements EventHandler {

    ...

    @Modified
    protected void update(final Map<String, Object> props) {
        this.cleanupPeriod =
            PropertiesUtil.toInteger(props.get(PROP_CLEANUP_PERIOD));
    }
}
```

Without @Modified:
Component is restarted on config change!

Metatype

- OSGi Metatype Service
 - Description of bundle metadata
 - Description of service configurations
 - Property type, name, and description

Configuration – Supports Metatype

```
import org.apache.sling.commons.osgi.PropertiesUtil;

@Component(metatype=true, label="Distributing Event Handler",
           description="This handler is awesome.")

@Properties({
    @Property(name="event.topics", value="*", propertyPrivate=true)
})
public class DistributingEventHandler
    implements EventHandler {

    private static final int DEFAULT_CLEANUP_PERIOD = 15;

    @Property(intValue=DEFAULT_CLEANUP_PERIOD,
              label="Cleanup Period",
              description="This is the cleanup period in seconds.")
    private static final String PROP_CLEANUP_PERIOD = "cleanup.period";
```

Component Development with Declarative Services

- Straight forward component development
- Single source of truth
- (Will even be easier with new Declarative Service spec from R6)
- Typical problems
 - OSGi dynamics is hidden
 - Handling references of cardinality multiple
 - Copying of bad practices

And what about THE WEB? (2007)

- OSGi Http Service Specification
 - Based on servlet API 2.1
 - Programmatic API to register servlets and resources
- Apache Felix HTTP implementation
- Later whiteboard support in Apache Felix implementation for
 - Servlets
 - Filters
 - Some listeners

And what about THE WEB?

- OSGi Http Service Specification
 - Based on servlet API 2.1
 - Programmatic API to register servlets and resources
- Apache Felix HTTP implementation
- Invented Apache Sling Web Framework
 - Resource based approach
 - Scripting
 - Written as OSGi bundles
- Later whiteboard support in Apache Felix implementation for
 - Servlets
 - Filters
 - Some listeners
- New OSGi Http Whiteboard Service Spec in R6

But what about Provisioning and Deliverables? (2007)

- Wishes
 - Single definition of deliverable (bundles, configurations, proprietary packages)
 - Different flavors: web application, standalone
 - Easy and flexible updates

But what about Provisioning and Deliverables? (2007)

- Wishes
 - Single definition of deliverable (bundles, configurations, proprietary packages)
 - Different flavors: web application, standalone
 - Easy and flexible updates
- No open source solution available

Contributing Apache Sling Launchpad Ecosystem

- Creating a distribution is easy
 - Standalone jar file and/or web application
- Maven Launchpad Plugin
 - Define bundle list
 - Add other artifacts
 - Configurations
- Leverages Apache Sling's OSGi installer

Not tied to Sling – just bundles & tooling

Bundle List Example

```
<?xml version="1.0"?>
<bundles>
  <startLevel level="boot">
    <bundle>
      <groupId>org.apache.sling</groupId>
      <artifactId>org.apache.sling.commons.log</artifactId>
      <version>2.1.2</version>
    </bundle>

    ...

  <startLevel level="9">
    <bundle>
      <groupId>org.apache.felix</groupId>
      <artifactId>org.apache.felix.eventadmin</artifactId>
      <version>1.3.2</version>
    </bundle>

    ...

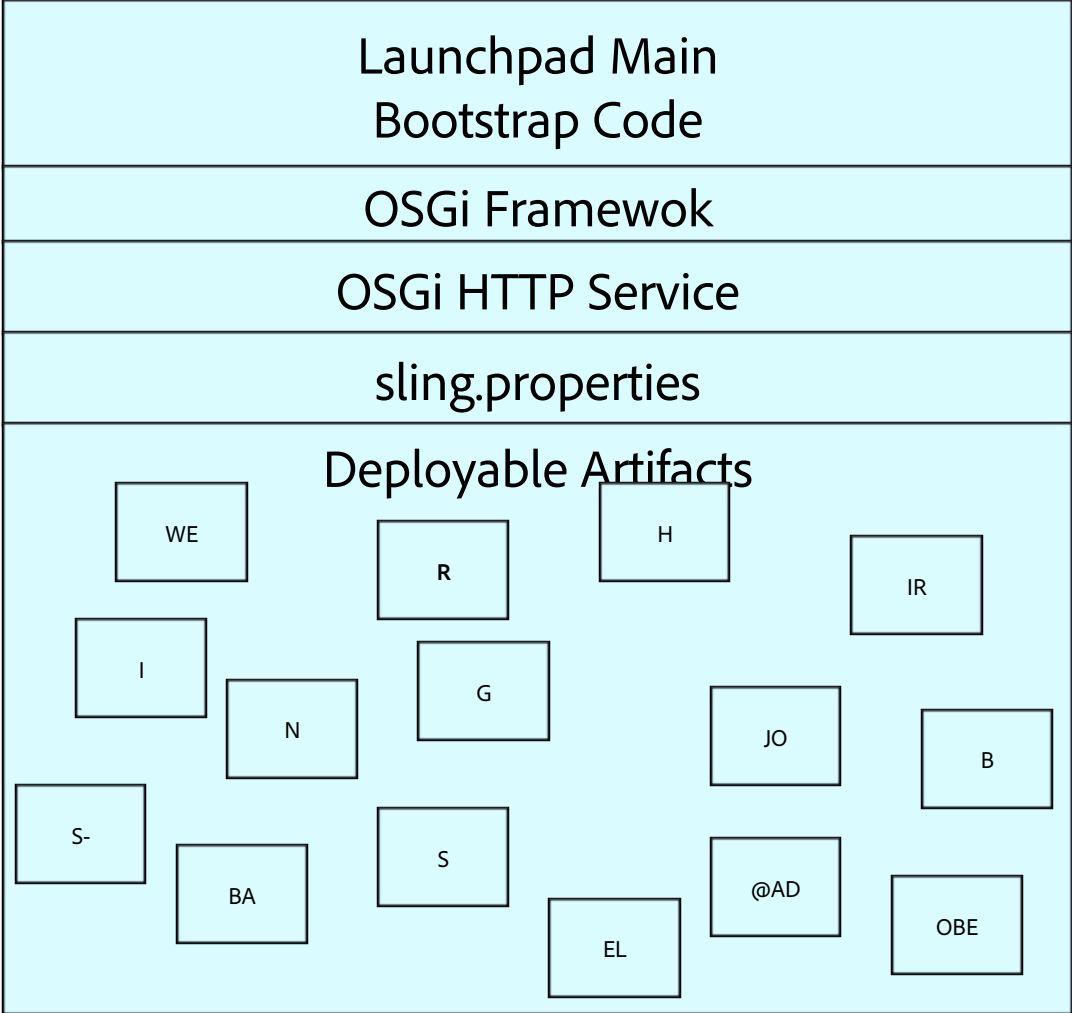
  <startLevel level="0">
    <bundle>
      <groupId>org.some.company</groupId>
      <artifactId>a.strange.artifact</artifactId>
      <version>1.2.0</version>
      <type>zip</zip>
    </bundle>
  </startLevel>
</bundles>
```

Apache Sling Launchpad

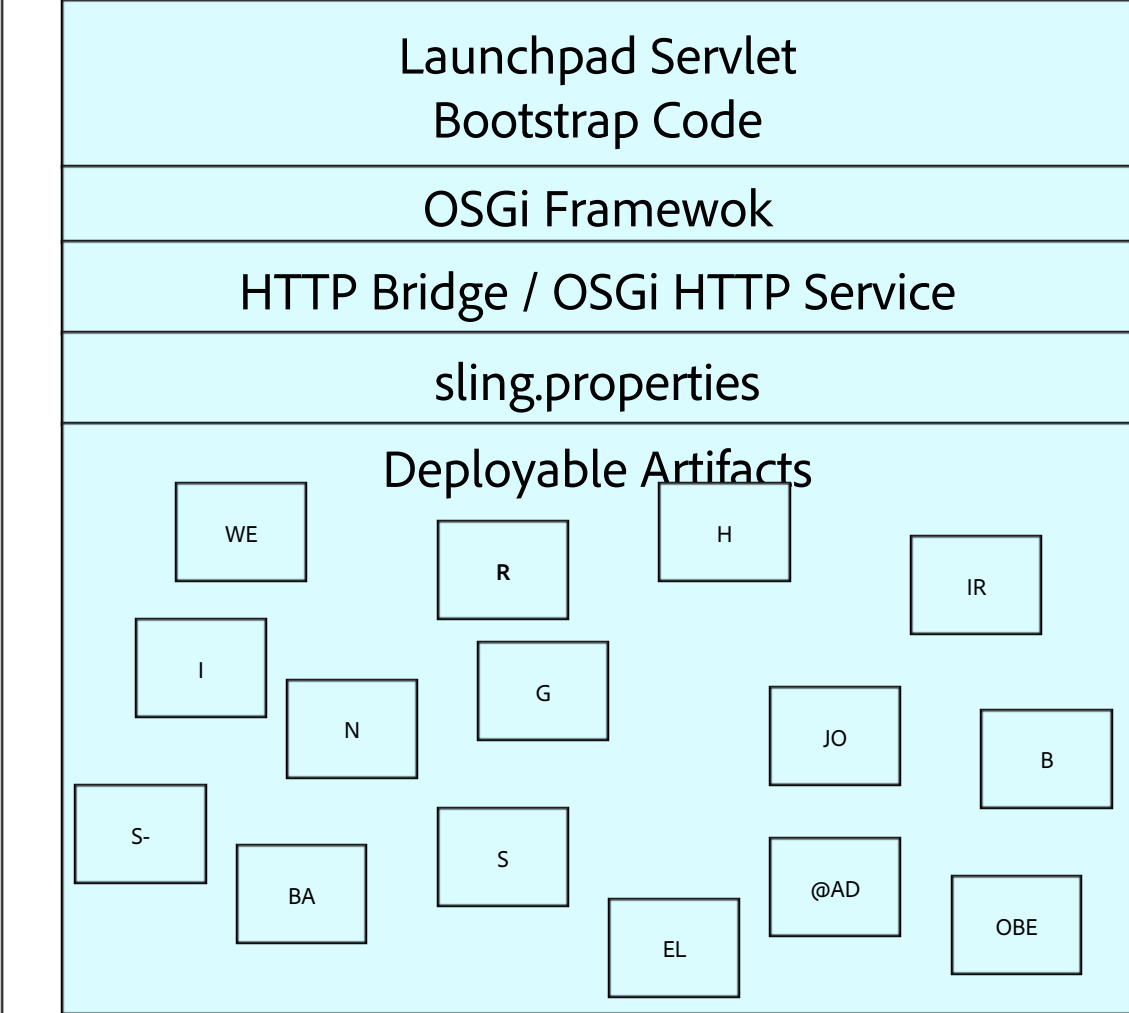
- Bootstrap code
- Includes Apache Felix
 - Handles framework updates
- Launch an OSGi framework
 - Standalone jar with http service
 - As a web application with http bridge
- Deploy packaged artifacts
- Configurable
 - sling.properties
 - (system properties / servlet parameters)

OSGi Standalone App or OSGi Webapp with Launchpad

JAR File



WAR File

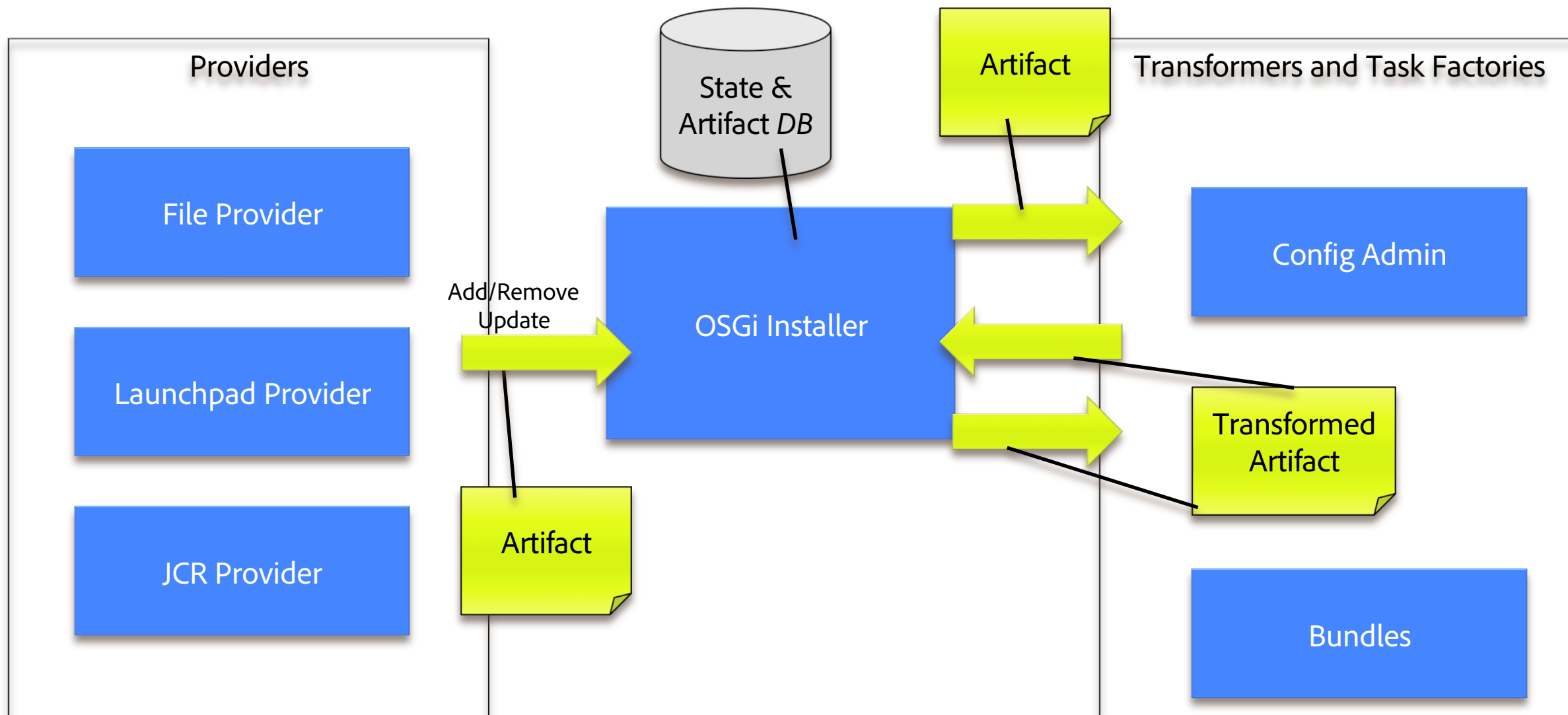


Apache Sling OSGi Installer

- General service covering
 - Install
 - Update
 - Uninstall
- Of
 - Bundles
 - Configurations
 - Custom artifacts

Not tied to Sling – just a bundle!

Apache Sling OSGi Installer



Advantages of the OSGi Installer Family

- Pluggable and highly customizable
 - New artifact types: transformer + task factory
 - New artifact sources: provider
- Batch handling
- Retry mechanism in the installer core
 - Bundle installation order doesn't matter
- State management in the installer core
 - Defined workflow
 - Copes with 3rd party changes

Running Deliverables and what about administration? (2007)

- Nothing really available in the open source world
- Especially not web based

Donating Apache Felix Web Console + Plugins

- Rich set of core functionality
 - Bundles
 - Services
 - Configuration Admin
 - System Information
- (jQuery based UI)
- Extensible
- Pluggable authentication
- Still light-weight
- REST interface

Adobe Experience Manager Web Console Bundles













Main OSGi Sling Status Web Console

Bundle information: 393 bundles in total - all 393 bundles active

Apply Filter Filter All Reload Install/Update... Refresh Packages

| Id | Name | Version | Category | Status | Actions |
|----|--------------------------------------------------------------------------------------------|----------------|----------|----------|----------|
| 0 | System Bundle (<i>org.apache.felix.framework</i>) | 4.3.0.R1558704 | | Active | |
| 1 | Adobe Granite Startup Module (<i>com.adobe.granite.startup</i>) | 0.6.2 | granite | Active | ■ ↻ ↺ 🗑️ |
| 2 | jcl-over-slf4j (<i>jcl.over.slf4j</i>) | 1.7.6 | | Active | ■ ↻ ↺ 🗑️ |
| 3 | log4j-over-slf4j (<i>log4j.over.slf4j</i>) | 1.7.6 | | Active | ■ ↻ ↺ 🗑️ |
| 4 | Apache Felix Configuration Admin Service (<i>org.apache.felix.configadmin</i>) | 1.8.0 | osgi | Active | ■ ↻ ↺ 🗑️ |
| 5 | Apache Sling SLF4J Implementation (Logback) (<i>org.apache.sling.commons.log</i>) | 4.0.0 | sling | Active | ■ ↻ ↺ 🗑️ |
| 6 | Apache Sling OSGi LogService Implementation (<i>org.apache.sling.commons.logservice</i>) | 1.0.2 | sling | Active | ■ ↻ ↺ 🗑️ |
| 7 | Apache Sling System Bundle Extension: WS APIs (<i>org.apache.sling.fragment.ws</i>) | 1.0.2 | sling | Fragment | ↻ ↺ 🗑️ |
| 8 | Apache Sling System Bundle Extension: XML APIs (<i>org.apache.sling.fragment.xml</i>) | 1.0.2 | sling | Fragment | ↻ ↺ 🗑️ |
| 9 | Apache Sling Installer (<i>org.apache.sling.installer.core</i>) | 3.5.0 | sling | Active | ■ ↻ ↺ 🗑️ |
| 10 | Apache Sling File Installer (<i>org.apache.sling.installer.provider.file</i>) | 1.0.2 | sling | Active | ■ ↻ ↺ 🗑️ |
| 11 | Apache Sling javax.activation bundle (<i>org.apache.sling.javax.activation</i>) | 0.1.0 | sling | Active | ■ ↻ ↺ 🗑️ |
| 12 | Apache Sling Launchpad Installer (<i>org.apache.sling.launchpad.installer</i>) | 1.2.0 | sling | Active | ■ ↻ ↺ 🗑️ |
| 13 | Apache Sling Settings (<i>org.apache.sling.settings</i>) | 1.3.0 | sling | Active | ■ ↻ ↺ 🗑️ |
| 14 | slf4j-api (<i>slf4j.api</i>) | 1.7.6 | | Active | ■ ↻ ↺ 🗑️ |
| 15 | Apache Aries JMX API (<i>org.apache.aries.jmx.api</i>) | 1.0.0 | | Active | ■ ↻ ↺ 🗑️ |
| 16 | Apache Aries JMX Core (<i>org.apache.aries.jmx.core</i>) | 1.0.1 | | Active | ■ ↻ ↺ 🗑️ |
| 17 | Apache Aries JMX Whiteboard (<i>org.apache.aries.jmx.whiteboard</i>) | 1.0.0 | | Active | ■ ↻ ↺ 🗑️ |
| 18 | Apache Aries Transaction Manager (<i>org.apache.aries.transaction.manager</i>) | 1.0.1 | | Active | ■ ↻ ↺ 🗑️ |
| 19 | Apache Aries Util (<i>org.apache.aries.util</i>) | 1.0.0 | | Active | ■ ↻ ↺ 🗑️ |
| 20 | Apache Sling Discovery API (<i>org.apache.sling.discovery.api</i>) | 1.0.0 | sling | Active | ■ ↻ ↺ 🗑️ |
| 21 | Adobe Granite JMX Support (<i>com.adobe.granite.jmx</i>) | 0.3.0 | granite | Active | ■ ↻ ↺ 🗑️ |
| 22 | Granite Webconsole Plugins (<i>com.adobe.granite.webconsole.plugins</i>) | 1.0.2 | granite | Active | ■ ↻ ↺ 🗑️ |
| 23 | Apache Commons FileUpload (<i>org.apache.commons.fileupload</i>) | 1.3.1 | | Active | ■ ↻ ↺ 🗑️ |
| 24 | Commons IO (<i>org.apache.commons.io</i>) | 2.4.0 | | Active | ■ ↻ ↺ 🗑️ |
| 25 | Commons Lang (<i>org.apache.commons.lang</i>) | 3.0.1 | | Active | ■ ↻ ↺ 🗑️ |

Web Console Demo II

| | | |
|--------------------------------------------------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AEM Social Communities Related Search Servlet | - |    |
| AEM Social Communities Related search utility | - |    |
| AEM Social Communities Reverse Replicator | - |    |
| AEM Social Communities Reverse Replicator (DEPRECATED) | - |    |
| AEM Social Communities ReviewOperationProvider | - |    |

Apache Felix Event Admin Implementation ✕

Configuration for the Apache Felix Event Admin Implementation. This configuration overwrites configuration defined in framework properties of the same names.

| | | | | | | | | | | | | | |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|----------------------------------|----------------------------------|------------------------------------------------|----------------------------------|----------------------------------|---------------------------------------|----------------------------------|----------------------------------|-----------------------------------------|----------------------------------|----------------------------------|
| Thread | <input type="text" value="20"/> | | | | | | | | | | | | |
| Pool Size | The size of the thread pool. The default value is 10. Increase in case of a large amount of synchronous events where the event handler services in turn send new synchronous events in the event dispatching thread or a lot of timeouts are to be expected. A value of less then 2 triggers the default value. A value of 2 effectively disables thread pooling. (org.apache.felix.eventadmin.ThreadPoolSize) | | | | | | | | | | | | |
| Timeout | <input type="text" value="5000"/> The black-listing timeout in milliseconds. The default value is 5000. Increase or decrease at own discretion. A value of less then 100 turns timeouts off. Any other value is the time in milliseconds granted to each event handler before it gets blacklisted (org.apache.felix.eventadmin.Timeout) | | | | | | | | | | | | |
| Require Topic | <input checked="" type="checkbox"/> Are event handlers required to be registered with a topic? This is enabled by default. The specification says that event handlers must register with a list of topics they are interested in. Disabling this setting will enable that handlers without a topic are receiving all events (i.e., they are treated the same as with a topic=*). (org.apache.felix.eventadmin.RequireTopic) | | | | | | | | | | | | |
| Ignore Timeouts | <table border="1"><tr><td><input type="text" value="org.apache.felix*"/></td><td><input type="button" value="+"/></td><td><input type="button" value="-"/></td></tr><tr><td><input type="text" value="org.apache.sling*"/></td><td><input type="button" value="+"/></td><td><input type="button" value="-"/></td></tr><tr><td><input type="text" value="com.day*"/></td><td><input type="button" value="+"/></td><td><input type="button" value="-"/></td></tr><tr><td><input type="text" value="com.adobe*"/></td><td><input type="button" value="+"/></td><td><input type="button" value="-"/></td></tr></table> <p>Configure event handlers to be called without a timeout. If a timeout is configured by default all event handlers are called using the timeout. For performance optimization it is possible to configure event handlers where the timeout handling is not used - this reduces the thread usage from the thread pools as the timeout handling requires an additional thread to call the event handler. However, the application should work without this configuration property. It is a pure optimization! The value is a list of strings. If a string ends with a dot, all handlers in exactly this package are ignored. If the string ends with a star, all handlers in this package and all subpackages are ignored. If the string neither ends with a dot nor with a star, this is assumed to define an exact class name. (org.apache.felix.eventadmin.IgnoreTimeout)</p> | <input type="text" value="org.apache.felix*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> | <input type="text" value="org.apache.sling*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> | <input type="text" value="com.day*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> | <input type="text" value="com.adobe*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> |
| <input type="text" value="org.apache.felix*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> | | | | | | | | | | | |
| <input type="text" value="org.apache.sling*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> | | | | | | | | | | | |
| <input type="text" value="com.day*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> | | | | | | | | | | | |
| <input type="text" value="com.adobe*"/> | <input type="button" value="+"/> | <input type="button" value="-"/> | | | | | | | | | | | |

Configuration Information

| | |
|---------------------------|----------------------------------------------------------------------|
| Persistent Identity (PID) | org.apache.felix.eventadmin.impl.EventAdmin |
| Configuration Binding | Apache Felix EventAdmin (org.apache.felix.eventadmin), Version 1.3.2 |

| | | |
|----------------------------------------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Apache Felix Web Console Event Plugin | - |    |
| Apache Felix Web Console Memory Usage Plugin | - |    |
| Apache Jackrabbit Oak Default Sync Handler | - | <input type="button" value="+"/> |
| Apache Jackrabbit Oak External Login Module | - | <input type="button" value="+"/> |
| Apache Jackrabbit Oak LDAP Identity Provider | - | <input type="button" value="+"/> |
| Apache Sling Authentication Logout Servlet | - |    |

Web Console Demo III

Apache Felix Web Console - System Properties

localhost:4502/system/console/status-System Proper

Apache Felix Web Console System Properties

Main OSGi Sling **Status** Web Console

Bundlelist

Date: October 25, 2011

Download As Text Download As Zip Download Full Text Download Full Zip

apple.awt.graphics apple.awt.graphics
apple.awt.graphics awt.nativeDoubleBuf
awt.toolkit = apple
com.rsa.crypto.def
com.rsa.crypto.jce
crx.index.maxSize.4
crx.index.maxSize.4
crx.index.maxSize.4
derby.stream.error
file.encoding = Mac
file.encoding.pkg =
file.separator = /
ftp.nonProxyHosts =
gopherProxySet = f
http.nonProxyHosts
java.awt.graphicsse
java.awt.headless =
java.awt.printerjob
java.class.path =
java.class.version
java.endorsed.dirs
java.ext.dirs = /L
java.home = /System
java.io.tmpdir = /
java.library.path =
java.runtime.name =
java.runtime.version
java.specification
java.specification
java.specification.version = 1.6
java.vendor = Apple Inc.
java.vendor.url = http://www.apple.com/
java.vendor.url.bug = http://bugreport.apple.com/
java.version = 1.6.0_65
java.vm.info = mixed mode
java.vm.name = Java HotSpot(TM) 64-Bit Server VM
java.vm.specification.name = Java Virtual Machine Specification
java.vm.specification.vendor = Sun Microsystems Inc.
java.vm.specification.version = 1.0

PRNG
mand
am = 1000
= 1000
core.CRXRepositoryImpl.DEV_NULL

54/16|*.169.254/16

54/16|*.169.254/16
csEnvironment

Job

ra/JavaVirtualMachines/1.6.0.jdk/Contents/Home/lib/endorsed
:/System/Library/Java/Extensions:/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Content
tualMachines/1.6.0.jdk/Contents/Home
xx6tb7tqmc08vnmFh0000gn/T/
veloper/Tools/YourKit_Java_Profiler_10.0.6.app/bin/mac:::/Library/Java/Extensions:/System/I
Environment
-11M4609
API Specification
stems Inc.

- Status zip contains
 - Output from all inventory printers
 - Output from special web console plugins
 - Attachments
 - JSON files
- Automated analysis
 - Bundle list and status
 - Configurations
 - Log files

Problems solved...

- Modularity
 - OSGi bundles
 - Semantic versioning
 - Services and components
- Development
 - Frameworks and tooling
- Deliverables and provisioning
- Administration

Hitting the Real World – 3rd Party Libraries

- Use them as bundles
 - Project delivers already a bundle
 - Apache Commons, Apache Sling etc.
 - Use special bundle repositories
 - Apache Felix Commons, Spring etc.
 - But check included metadata!
 - Create your own wrapper
 - Easy with the Felix maven bundle plugin
- Include classes in your bundle
 - Again: easy with the Felix maven bundle plugin
- **Convince the projects to provide useful OSGi metadata!**

Hitting the Real World – "Enterprise" Requirements

- Testing under high load / high concurrency
 - Reimplementing some (open source) modules
 - Not every developer is aware that code can be called concurrently
 - And how to solve this in an acceptable way
- Dynamic nature of OSGi
 - Relatively painless with DS and annotations
 - Gets trickier with plain framework API and/or cardinality multiple
 - Again: concurrency

Hitting the Real World – All Time Favorites

- When is my system ready?
 - Different strategies
 - All bundles started and specific start level reached
 - Special startup handler service for notification (Apache Sling)
- Fighting against FUD
 - OSGi is slow, consumes too much memory, complicated,
 - has no dependency injection, is not used in the enterprise...

Additional Challenges

- Clustering
- Discovery
- Distributed Eventing

On the Roadmap

- OSGi subsystems
 - Grouping of bundles
 - Export / import boundaries on subsystem level
- OSGi Remote Services
- OSGi Asynchronous Services
- OSGi Distributed Eventing

Think Modular...

- This is not a new concept or latest hype!
- Think about modularity!
 - Create a clean package space
 - What is your API?
 - public vs private
 - Make things only public if necessary/used
 - Starting with private stuff going public is easy
- Use proper versioning contracts
 - Semantic Versioning

...Be Modular!

- Provide bundles
 - Add manifest information
- Think about dependencies
 - Additional bundle vs include
 - Optional
 - Version ranges
- Manage releases and versions properly
- Benefits even without OSGi

Component Development

- Contract defined by service interface
- Declarative Services is a powerful component model
 - Minimal OSGi knowledge required with available solutions
 - Apache Felix SCR Tooling
- Alternatives available 😊
 - Inter-operability through OSGi service registry

Conclusion

- Today a lot of the hard work is already done
- Open Source & Open Development are your friends
- OSGi made the Adobe Experience Manager success story possible
 - Team growth from 40+ to several hundreds
 - Immense growth in functionality
 - Still (dynamically) extensible and manageable
- OSGi Core and Compendium provide nearly everything you need
 - Framework, Declarative Services, Configuration Admin, Metatype, Event Admin
- Open Source Implementations and Extensions
 - Apache Felix and friends

Check It Out!

- Read the OSGi specs
 - Framework
 - Declarative Services, Config Admin, Metatype, Event Admin
- Checkout the various Apache projects!
- Explore available tooling
- Embrace modularity!
- Join the OSGi Alliance and help shaping the future!

And...New OSGi Specifications!

- Last week (week of June 2nd) the **OSGi Core Release 6 Specification** received its final approval and will be available this week to the public for downloading.
 - <http://www.osgi.org/Specifications/HomePage>
- Also happening last week, the OSGi Board of Directors approved the publication of an **Early Draft Specification of OSGi Enterprise Release 6** for downloading.
 - <http://www.osgi.org/Specifications/Drafts>

References

- OSGi Semantic Versioning Whitepaper: <http://www.osgi.org/wiki/uploads/Links/SemanticVersioning.pdf>
- OSGi Specifications: <http://www.osgi.org>
- Apache Felix: <http://felix.apache.org>
- Apache Sling: <http://sling.apache.org>
- Apache Aries: <http://aries.apache.org>
- BND: <http://www.aqute.biz/Bnd/Bnd>
- Bndtools: <http://njbartlett.name/bndtools.html>
- Planet OSGi: <http://www.osgi.org/Planet/Feed>



Adobe