

Building your own portal with Apache Jetspeed-2 Enterprise Portal

Ate Douma

ApacheCon EU 2008, Amsterdam April 11



Agenda

- An overview of the main Jetspeed-2 features
- A new way to build your own Jetspeed-2 Portal
- Demo of a custom build Portal and Jetspeed-2 features
- Jetspeed-2 Architecture and Configuration
- Online customization
- A roadmap to version 2.2 and beyond
- Questions and Answers



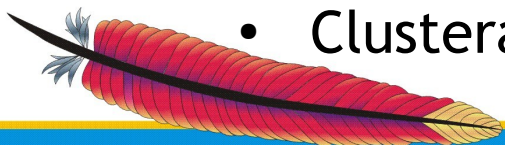
Main Jetspeed-2 features

- A *Portal* as well as a *Portal Integration Framework*
- Fully open and standards based, JSR-168 compliant
- Component based from top to bottom
- Extendable and programmable through the Jetspeed API
- Fully customizable through IOC
- Assembled and wired together using Spring
- Runs on Tomcat, Jetty, Geronimo, Jboss, Glassfish as well as on WebSphere (5..6) and Weblogic
- Authentication/authorization through LDAP or database



Main Jetspeed-2 features

- Pluggable and configurable request pipeline(s)
- Extensive configurable and pluggable security layer
- Profiler for dynamic mapping of requests to pages:
 - hostname/role/group/user/browser/language/mediatype/[custom criteria]/... based matching
- Subsites for hosting multiple communities
- Hierarchical site model for themes, decorations, menus
- Everything is rendered through portlets
- Parallel rendering
- Clusterable



A new way to build your own portal

Challenges

- everyone wants it differently
- configurability is great but also difficult to automate
- integration with different app servers, security systems, databases can be very complex
- maven (1 & 2) hasn't been, and still isn't, everybody's favorite and at times really `*@%~(` hard to use



A new way to build your own portal

History and current state

- maven-1 plugin (since 2004)
- maven-2 archetypes (since late 2005)
- Starting from scratch with maven-2 (since Aug 2007)
- Now trunk is 100% maven-2 buildable (since yesterday):
 - a new project structure (in line with maven-2 guidelines)
 - using brand new maven plugins for database initialization, configuring your build resources and deploying



A new way to build your own portal

Following the maven-2 rules:

- Everything defined through the pom.xml
- Use of the standard maven plugins as much as possible
- Write custom maven plugins for custom build features or standalone execution
- Keep everything “standard”
- A Portal is “just” an ordinary war project now



A new way to build your own portal

An example of a new jetspeed-2 maven plugin: jetspeed-db

```
<plugin>
  <groupId>org.apache.portals.jetspeed-2</groupId>
  <artifactId>jetspeed-db-maven-plugin</artifactId>
  <version>${org.apache.portals.jetspeed.version}</version>
  <goals>
    <goal>init</goal>
  </goals>
  <configuration>
    <connection>
      ...
    </connection>
    ...
    <sql>
      ...
    </sql>
    <seed>
      ...
    </seed>
  </configuration>
  <dependencies>
    ...
  </dependencies>
</plugin>
```



A new way to build your own portal

Another example: jetspeed-deploy

```

<plugin>
  <groupId>org.apache.portals.jetspeed-2</groupId>
  <artifactId>jetspeed-deploy-maven-plugin</artifactId>
  <version>${org.apache.portals.jetspeed.version}</version>
  <configuration>
    <targetBaseDir>${org.apache.jetspeed.server.home}</targetBaseDir>
    <portalName>${portal.context.name}</portalName>
    <deployments>
      <!-- jetspeed portal expanded -->
      <deployment>
        <artifact>${pom.groupId};${pom.artifactId};war</artifact>
        <expand>true</expand>
        <targetName>${portal.context.name}</targetName>
        <delete>./${portal.context.name}</delete>
      </deployment>
      <deployment>
        <artifact>org.apache.portals.applications:apache-pa:war</artifact>
        <infuse>true</infuse>
        <expand>true</expand>
        <delete>./apache-pa</delete>
      </deployment>
      ...
    </deployments>
  </configuration>
  ...

```

A new way to build your own portal

Already available maven-2 plugins:

- Jetspeed-unpack (unpack)
- Jetspeed-db (ddl, init)
- Jetspeed-deploy (deploy)
- Jetspeed-attached-deploy (deploy)

Other maven-2 plugins planned for:

- db sql data export, psml import/export
- installerBuilder



A new way to build your own portal

New feature: dynamic Spring configuration filtering

- Allows using multiple sets of configuration
- Mix and match your environment without touching the configuration files
- Only change the filter properties file or the used filter property itself
- Assembly override can still be used



A new way to build your own portal

Dynamic Spring configuration filtering:

```
<bean id="JetspeedDS" class="org.springframework.jndi.JndiObjectFactoryBean">
  <meta key="j2:cat" value="jndiDS" />
  <property name="resourceRef" value="false" />
  <property name="jndiName" value="java:comp/env/jdbc/jetspeed" />
</bean>

<bean class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
  <meta key="j2:cat" value="jdbcDS" />
  <meta key="j2:alias" value="JetspeedDS" />
  <property name="driverClassName" value="{org.apache.jetspeed.database.driver}" />
  <property name="url">
    <value>{org.apache.jetspeed.database.url}</value>
  </property>
  <property name="username">
    <value>{org.apache.jetspeed.database.user}</value>
  </property>
  <property name="password">
    <value>{org.apache.jetspeed.database.password}</value>
  </property>
</bean>
```



A new way to build your own portal

Dynamic Spring configuration filtering:

WEB-INF/conf/spring-filter.properties:

```
default=default
```

```
portal=${default},jndiDS
```

```
serializer=jdbcDS,serializer,capabilities,security,profiler,registry,  
search,transaction,cache,prefs,springProperties,noRequestContext,noPageManager
```



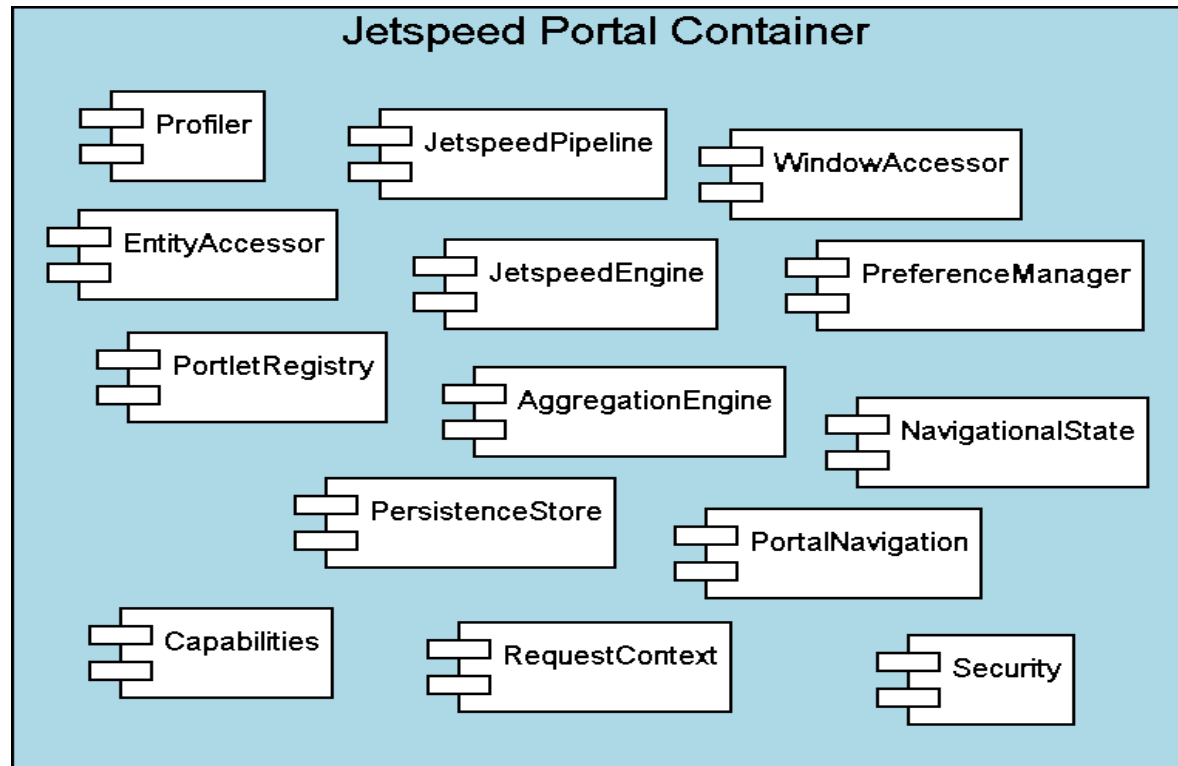
A custom Portal and Jetspeed-2 features

Demo



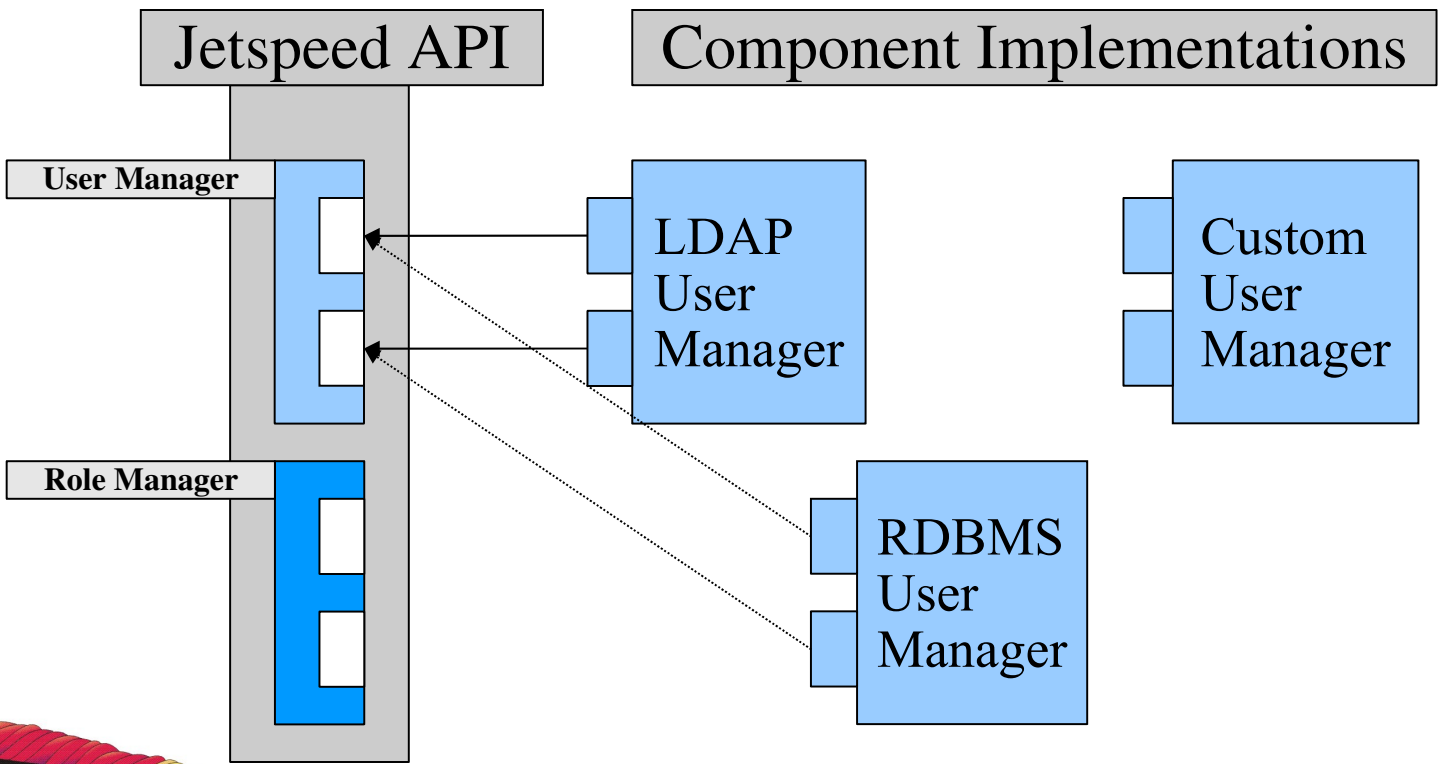
Architecture and Configuration

A Portal of pluggable components



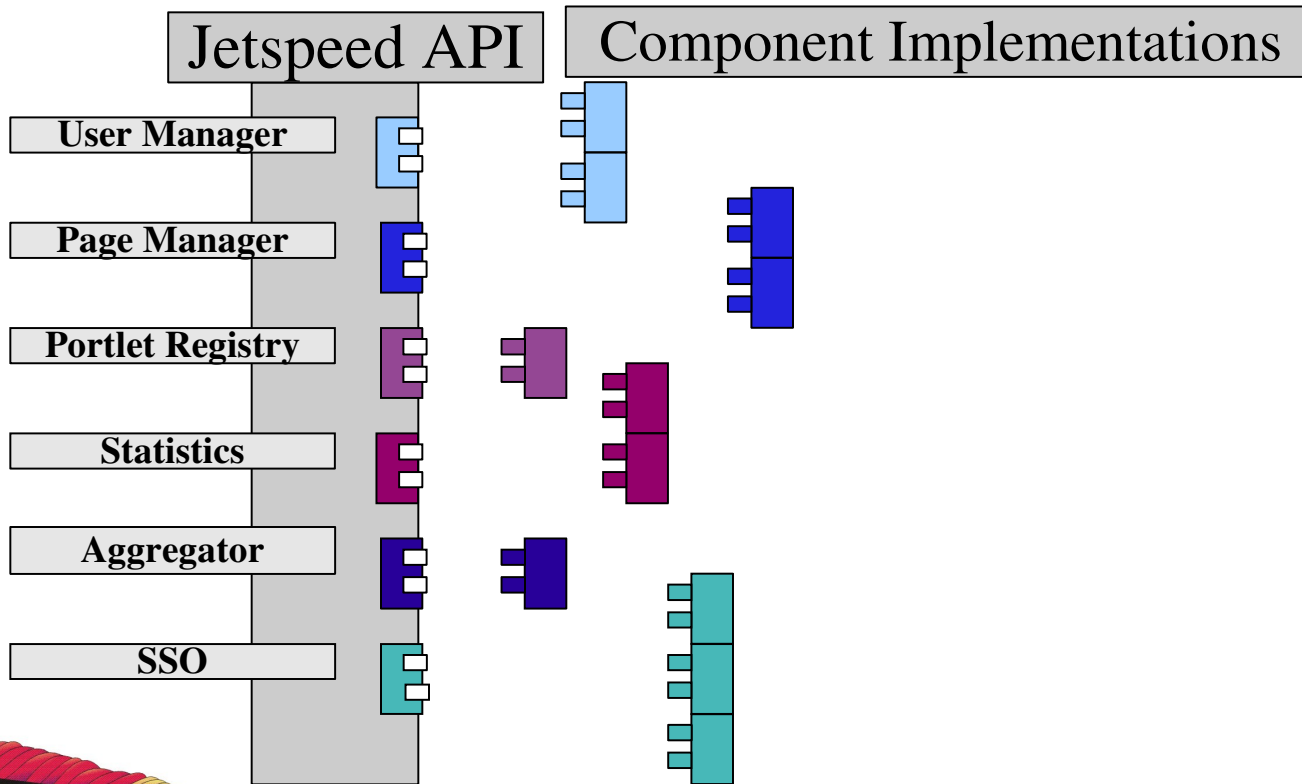
Architecture and Configuration

Interface Driven Development: the Jetspeed API



Architecture and Configuration

All Jetspeed components are setup to be pluggable:



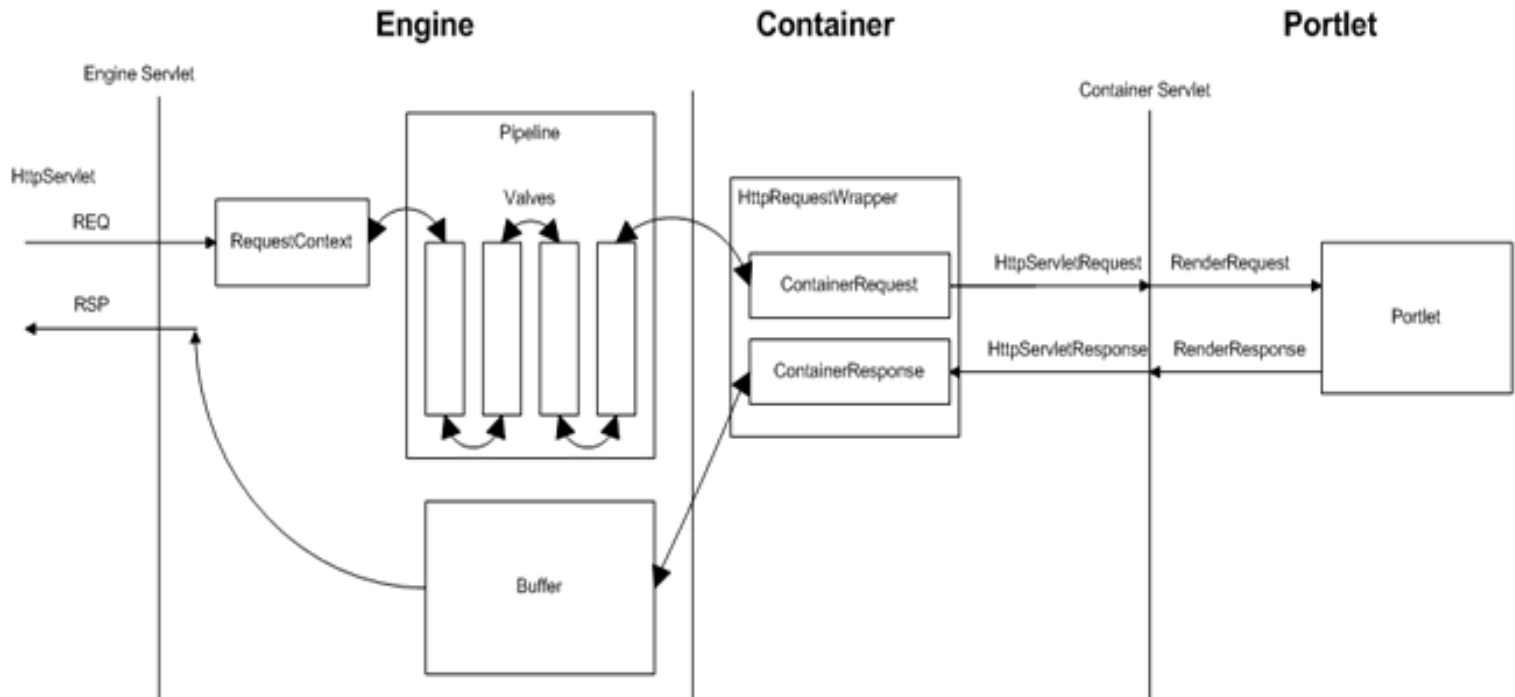
Architecture and Configuration

- Configuration changes best done without touching the defaults to support easier upgrade and maintenance:
 - Use the new dynamic Spring assembly filtering
 - Put custom Spring assembly under WEB-INF/assembly/override
 - Configure global properties in WEB-INF/conf/override.properties
- Common configuration changes:
 - request pipelines processing through custom valves
 - security (authentication & authorization)
 - page manager
 - profiler
 - administration (mailserver, root user etc.)



Architecture and Configuration

Pipeline Request Processing



Architecture and Configuration

XML Seed Data

- abstracted from database (native) sql format
- one format which also is more endurable
- both import and export for:
 - security (users, groups, roles, permissions)
 - capabilities
 - preferences
 - site (psml pages and configuration)



Architecture and Configuration

XML Seed Data:

```
j2-seed.xml
263 <Role>user</Role>
264 <Role>guest</Role>
265 <Role>wicket</Role>
266 </Roles>
267 <Groups>
268 <Group>accounting</Group>
269 <Group>engineering</Group>
270 </Groups>
271 <Users>
272 <User name="admin">
273 <credentials password="admin" enabled="TRUE" requiresUpdate="FALSE"/>
274 <roles>user,manager,admin</roles>
275 <userinfo>
276 <preference>
277 <name value="user.name.given"/>
278 <value value="Jetspeed"/>
279 </preference>
280 <preference>
281 <name value="user.name.family"/>
282 <value value="Administrator"/>
283 </preference>
284 </userinfo>
285 <Rules>
286 <Rule locator="page" rule="j2"/>
287 </Rules>
288 </User>
289 <User name="guest">
290 <roles>guest</roles>
291 <groups></groups>
292 <preferences/>
293 <userinfo/>
294 <Rules>
295 <Rule locator="page" rule="subsidiary-by-hostname"/>
296 </Rules>
297 </User>
298 <User name="wicket">
299 <credentials password="wicket" enabled="TRUE" requiresUpdate="FALSE"/>
300 <roles>user</roles>
301 <preferences/>
```

Architecture and Configuration

Decorators

- “Everything is a portlet”, including the “layout” portlets
- scriptable with Velocity or JSP
- (hot) deployable in Jetspeed 2
- layout decorator: css, header/footer, menus, structure
- layouts can be nested as well
- portlet decorator: css, window, actions



Online customization

Demo



Roadmap to 2.2 and beyond

- Release 2.2: 2nd quarter 2008
 - Complete new maven-2 (only) build environment
 - Upgrade to Pluto 2.0 (but no JSR-286 support yet)
 - Portlet Preferences reimplemented (no more Java Prefs)
 - Admin portlets designed and rewritten in Wicket
- Release 2.3: 4th quarter 2008
 - JSR-286 compliance and support
 - OpenJPA based O/R mapping, replacing OJB
 - Jetspeed-2 Portal “Light”



ApacheCon

Questions?



**Leading the Wave
of Open Source**

Building your own portal with Apache Jetspeed-2 Enterprise Portal