



JBoss Portal

Julien Viet / Lead Developer
Roy Russo / Developer
JBoss Inc.

March 8, 2005

© JBoss Inc. 2005

Agenda

- Portal fundamentals
- Portlet fundamentals
- JBoss Portal
- More JBoss Portal
- Q and A



2

Portal fundamentals

- Goals
 - ✓ Content aggregation
 - ✓ Personalization
 - ✓ Integration
 - ✓ Portlet specification



3

Content aggregation

- Portal components generates markup fragments
- Fragments are decorated by the portal
- Fragments are aggregated by the container



4

Personalization & integration

- A portal can present personalized views or behaviors based on
 - ✓ Identity
 - ✓ Context
- A portal can be used to integrate different applications
 - ✓ Provides single sign on
 - ✓ Shared user base
 - ✓ Managed services



5

The Portlet spec a.k.a. JSR-168

- Defines only the contract between the portal and the components running in the portal
- Based on the Servlet specification with similarities and differences



6

Portlet contract

- Portals must implement an API:
 - Request/Response
 - Session
 - Personalization
 - Security
- Portlets must implement the `javax.portlet.Portlet` interface
- Portlets are bundled in war files



7

Portlets vs. Servlets

- Similarities
 - ✓ Served by servlets
 - ✓ Packaged in a war file
 - ✓ Similar life cycle
- Differences
 - ✓ Limited impact on the request/response
 - ✓ Not bound to a URL
 - ✓ Assumes a human end user (human-readable output)



8

Portlet fundamentals

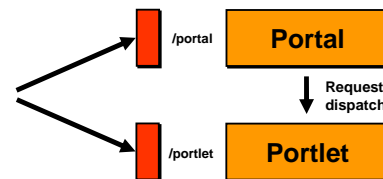
- Request dispatching
- Distinct action and render phases
- Portlet object concepts
- Personalization
- Misc



9

Request dispatching

- Portlets are always accessed through request dispatching



10

Constraints on the Session

- The HTTP session must follow the following constraints:
 - ✓ Portal sessions and portlet sessions are different
 - ✓ Web applications accessing the portlet directly must share the same session



11

Support in Tomcat

	Tomcat 4.x	Tomcat 5.0.x	Tomcat 5.5.x
Portal/Portlet Session Separation			
Session sharing across web apps			



12

Action/Render phases

- Portlet programming model differs from the servlet model
- Two independent phases:
 - ✓ Action phase to update the state
 - ✓ Render phase to create the markup



13

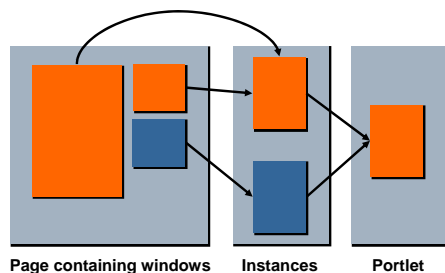
Portlet concepts

- Built around the concepts of portlet, instance and window:
 - ✓ The portlet corresponds to the class concept in Java
 - ✓ The instance represents the state of the portlet
 - ✓ The window manages the visibility of the instance information on a portal page
- Provides good flexibility at the cost of increased complexity



14

Portlet concepts



Page containing windows Instances Portlet



15

Personalization

- Happens at different levels
 - ✓ Portlet defines preferences
 - ✓ Instance can override portlet preferences
 - ✓ User can override instance preferences
- Example with forums portlet
 - ✓ Portlet defines 20 topics per forum page
 - ✓ Instance overrides with 10
 - ✓ Users wants to see more and set 50



16

Misc

- Possibility to cache a portlet rendered result for a limited amount of time
- View rendering can be delegated to JSP using a specific tag library
- Security based on servlet container
- Scales to different devices : web browser, cell phone, etc...



17

JBoss Portal

- Aspect-based implementation
- Caching
- Clustering



18

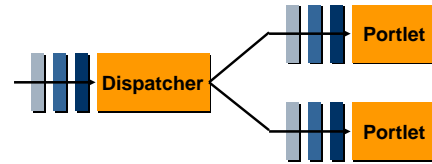
Aspect-based implementation

- Different functionalities are separated in aspects
 - ✓ Different scopes
 - Per portal aspects
 - Per portlet aspects
 - ✓ Better modularization
 - Each concern is isolated in one aspect
 - Easier to provide alternative implementations
 - Easier to add new behaviors
 - ✓ Aspect stack is described in XML configuration and can be overridden per portlet



19

Aspect-based implementation



20

Examples of available aspects

- Portal level
 - ✓ Select response content type
 - ✓ Select portlet that will participate in the view
- Portlet level
 - ✓ Cache rendered view
 - ✓ Load preferences
 - ✓ Update navigational state
 - ✓ Security



21

Caching

- An easy way to improve performances
- Portlets can greatly benefit from rendered view caching
- The portlet specification defines per user time based caching
- Cached content can be softened, i.e. utilize soft references to help curb memory usage



22

Clustering

- Replicating the complete portal state on a cluster is complex
- Decomposing the global state in sub-states helps but not a complete solution:
 - ✓ Some sub-states can be
 - Shared among all users or per user
 - Persistent or transient
 - ✓ But some sub-states cannot be replicated because:
 - of dependencies on other states
 - it's not implemented yet



23

Clustering

- States we can distinguish:
 - ✓ Identity
 - ✓ Personalization
 - ✓ Navigational state
 - ✓ Portal
 - ✓ Application
 - ✓ Services



24

More JBoss Portal

- Multi portals
- Deployment
- Authentication
- Managed authorization
- Services
- Default portlets



25

Multi portal

- JBoss Portal has the concept of multi portals
 - ✓ All portals share the same user base
 - ✓ Each portal has a set of distinct pages
 - ✓ Each portlet can adapt its generated content based on which portal it runs in (via the portal name stored in the request)



26

Deployment

- Portlet components
 - ✓ are packaged in a war file
 - ✓ along with servlets
 - ✓ Use a separate deployment descriptor : portlet.xml and other deployment descriptors specific to JBoss
- JBoss Portal leverages JBoss AS deployment capabilities



27

Authentication

- JBoss Portal uses the servlet container authentication
 - ✓ Custom authentication via JAAS login modules
 - ✓ Leverages Tomcat and JBoss single sign on (SSO) solutions
 - Tomcat SSO
 - JBoss clustered SSO
 - Any other JBoss-compatible solution



28

Authorization

- Authentication is addressed by the portlet spec
- But authorization is out of scope
 - ✓ Apart from using role-based authorization provided by the container
- Consequences:
 - ✓ Each application ends up having its own authorization model
 - ✓ No centralized management for authorization



29

Managed authorization

- JBoss Portal provides *managed authorization*
- Managed authorization defines a contract between portal and portlets:
 - ✓ Portlets expose a content model and can perform queries against this model
 - ✓ The portal manages the security scheme for the portlet
- Unified management via a nice console



30

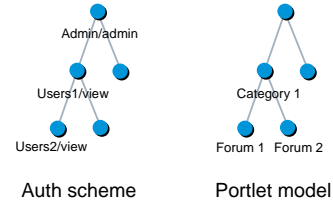
Managed authorization

- The model provider
 - ✓ Statically describes the permission it offers
 - Name and description
 - Implication relationship between the permission
 - ✓ Dynamically describes the model it owns
 - Tree based items



31

Managed authorization



32

Services

- Portal components services
 - ✓ User and Role
 - Default implementation based on Hibernate
 - Can provide own implementation
 - ✓ Mail
 - ✓ Indexing



33

Default Portlets

- User / Role
- Forums
- CMS / AdminCMS



34

Roadmap

- JBoss World: 2.0 beta 1
- May 2005: 2.0 beta 2
- June 2005: 2.0 final
- September 2005: 2.0.1 maintenance release



35

Future

- Web Services for Remote Portlet
- Java Server Faces integration
- Personal dashboard
- Advanced personalization
- Participation in portlet expert committee
- Support different output devices
- Pluggable Themes



36

Questions and answers

- Any question ?



37