

JBoss Community



Bridging today's most popular frameworks



RichFaces



Seam



GateIn

Prabhat Jha

Principal Engineer
JBoss, by Red Hat

Take Aways

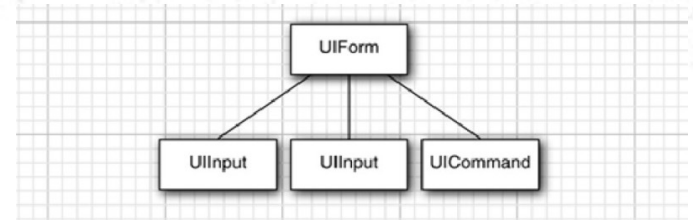
- Understand what a portlet is and does
- Learn how a JSF application maps to the portlet API
- Take an existing JSF app and port that to portlet
- Best practices & tips
- Update your resume. ;-)

Game Plan

- How did we get here?
- A Quick Recap of JSF & Richfaces
- A Quick Recap of Portal & Portlets
- Important points of spec
- Demos of JSF, Richfaces and Seam in a portal environment
- Future directions

JSF Intro in 2 slides..(1/2)

- Components
 - Basic UI building block in JSF
 - Assembled into component tree (view)
- Validators
 - Check correctness of user inputs
- Renderers
 - Output components to the client
- Converters
 - Responsible for data conversion between strings and another data types (in both ways)
- Backing beans
- Listeners
- Events



JSF Intro in 2 slides..(2/2)

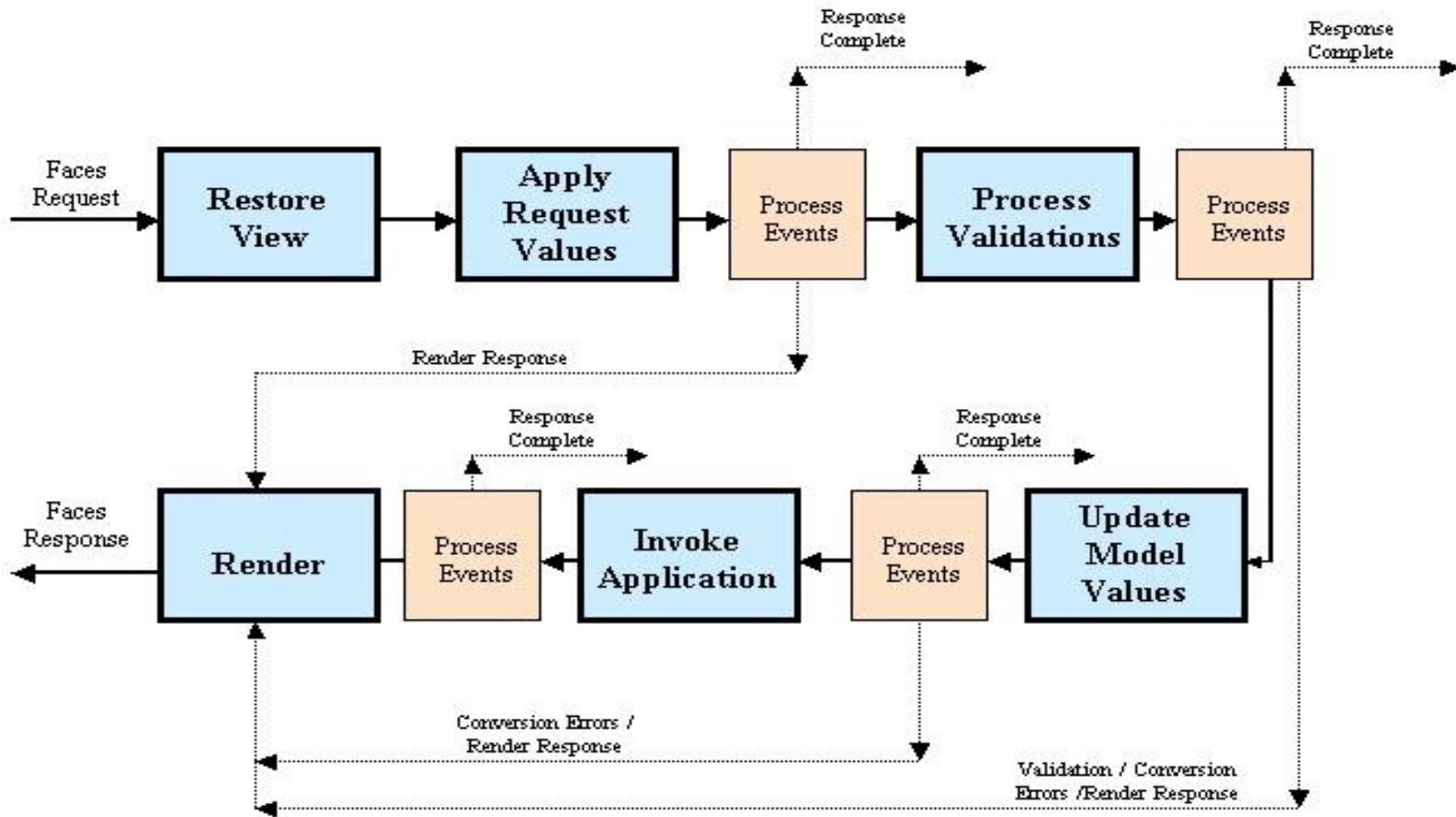


Figure 2: Faces Lifecycle

RF summary in one slide

- RichFaces is a JSF components library NOT an implementation of JSF like Mojarra & Myfaces.
- Adds Ajax support to JSF components as of JSF 1.2
- Provides OOB set of rich JSF components
- Skins & Themes
- Dynamic Resource Handling
- Component Development Kit (CDK)

What is a Portal ?

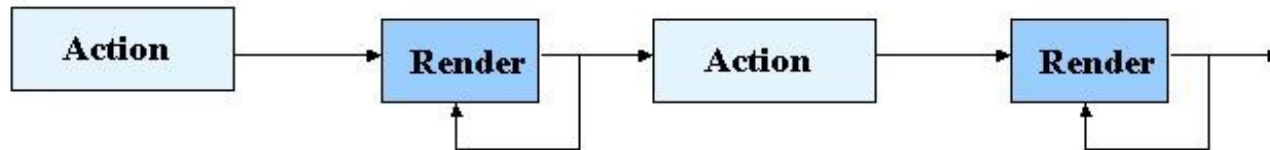
- Integration, Aggregation, Personalization
- Websites that require presentation of different pages depending on user's role and preference
- Websites that require the integration of multiple existing web applications may be **built on different web frameworks**
- Websites that require collaboration through applications, on content, or with documents.
- JBoss view: “ ..meets the demands of internal and external end users who require access to, and interaction between, back-end applications, SOA services, and Web 2.0 applications”

Portal sub parts?

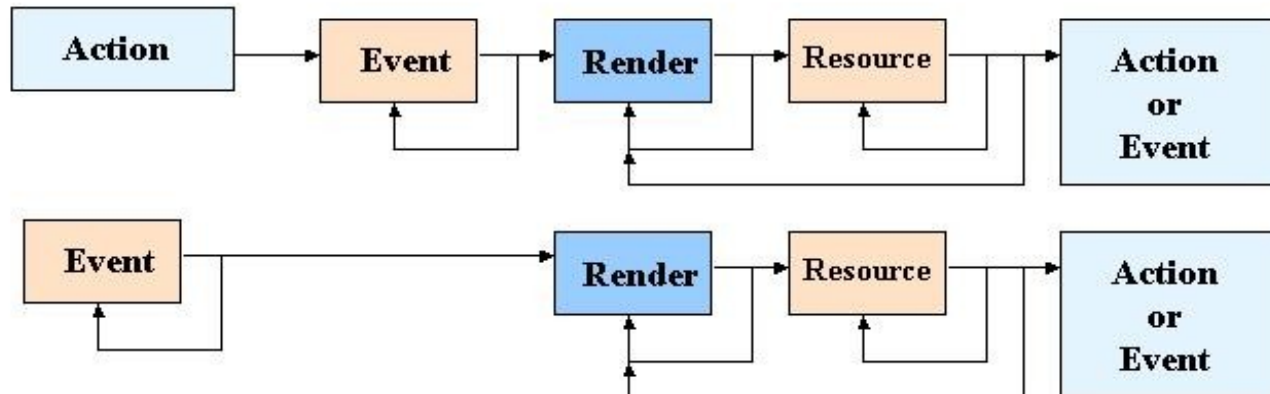
- Portal
 - Page
 - Portlet
- Portlet Container
- portlet.xml
- portletUrl

Portlet Lifecycle

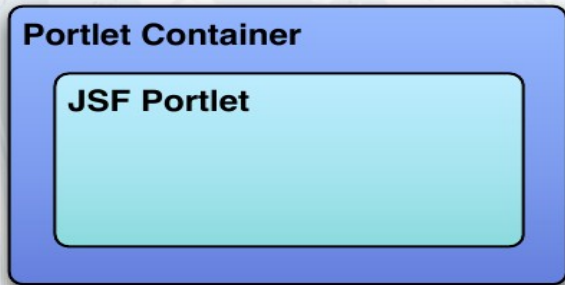
Portlet 1.0 request(s) lifecycle:



Portlet 2.0 request(s) lifecycle:

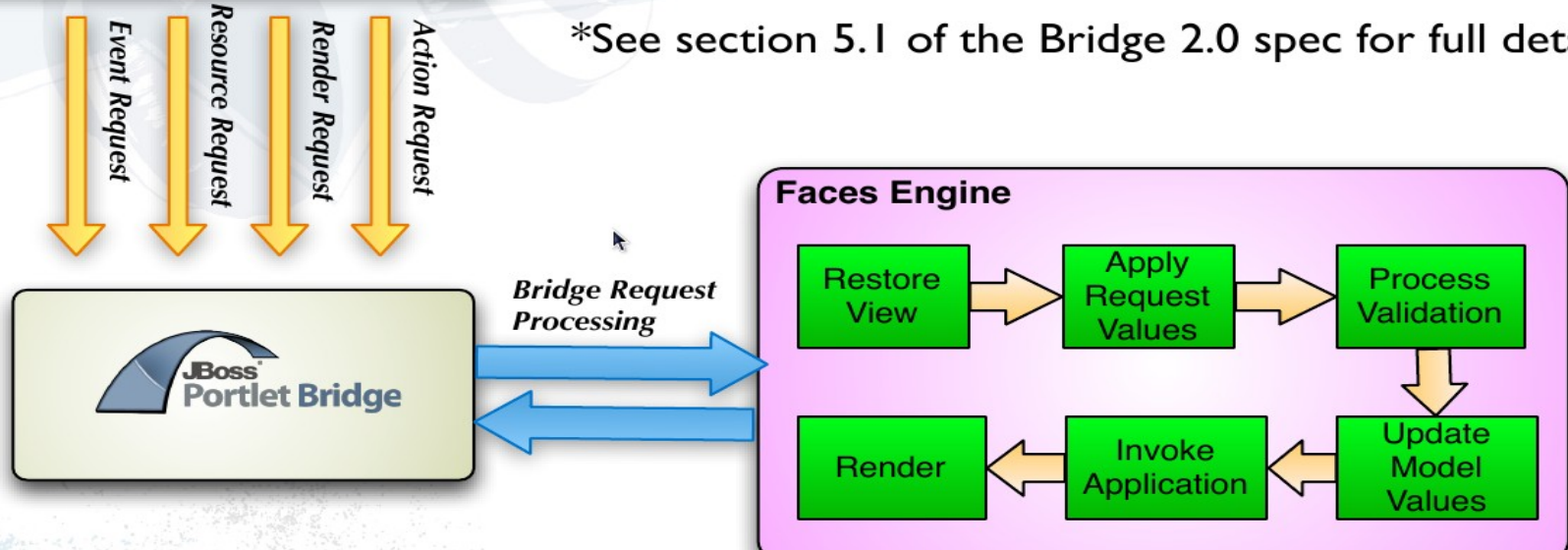


Portlet & JSF Lifecycle

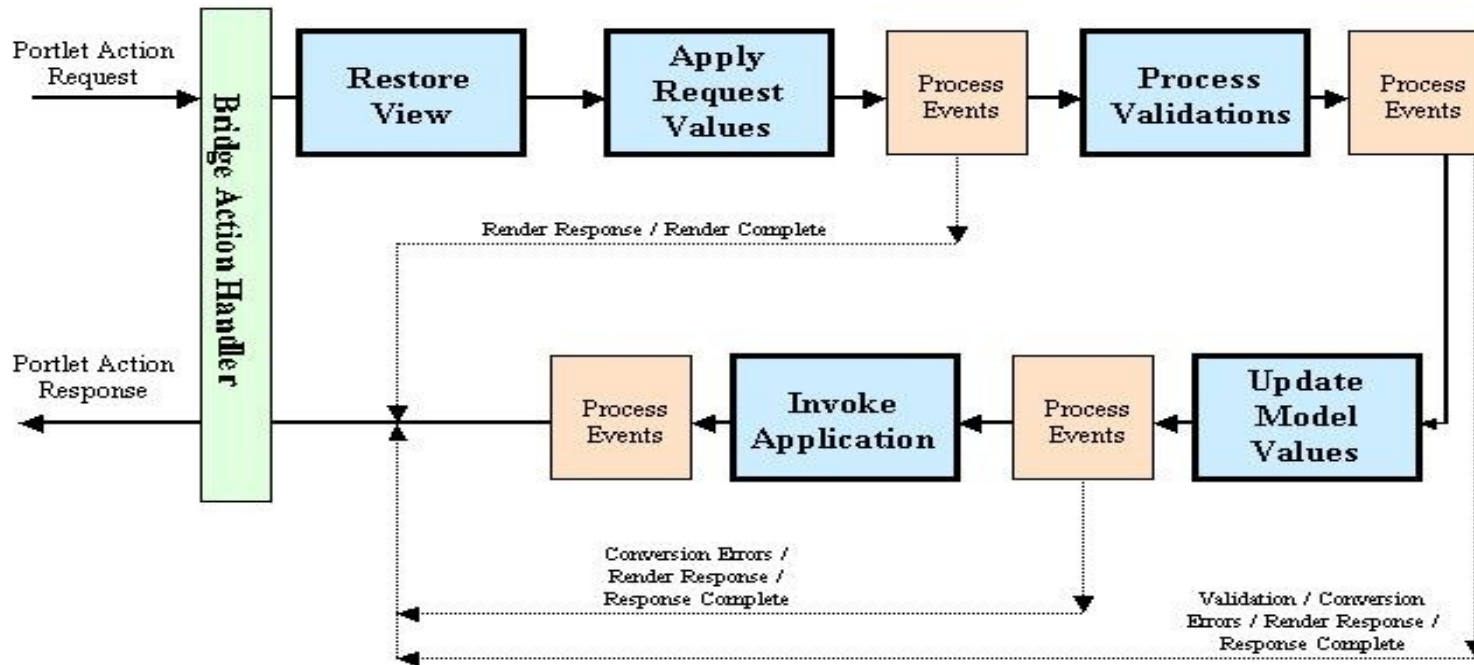


How the Portlet Bridge manages the Faces single request lifecycle

*See section 5.1 of the Bridge 2.0 spec for full details



Portlet & JSF Lifecycle



Portlet & JSF Lifecycle

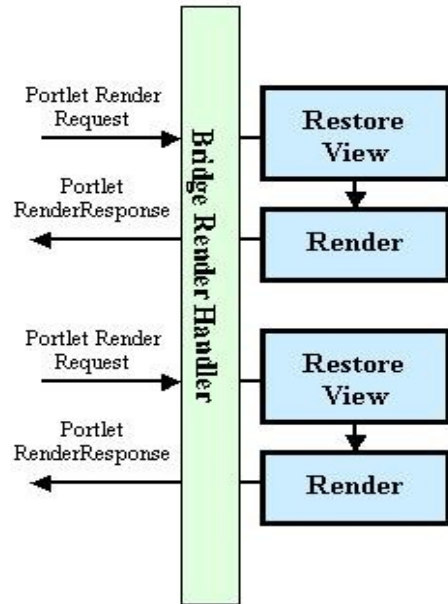


Figure 2: Faces Lifecycle in a Portlet Environment

Some inner details ..0

- Spec 1 : **JSR-301**
- Spec 2 : **JSR-329**

Some inner details ..1

- Servlet Container & Portlet Container
- Request Scopes
 - Faces request scope
 - Portlet request scope
 - Bridge request scope aka Managed Scope (The bridge request scope manages two types of Faces data: the Faces view state and the additional Faces request scope data Faces relies on when executing both the action and the render in a single request.)
- Mapping different stages of lifecycle
- Many instances of same portlet

Inner Details ..2

FacesContext.getExternalContext()

- getRequest and check for
 - actionRequest, renderRequest, ..
- getResponse
 - RenderResponse, ..
- getSession

Some inner details ..3

- **javax.portlet.faces.Bridge**
 - main abstraction of the Portlet Bridge API
 - doFacesRequest(ActionRequest request, ActionResponse response)
 - doFacesRequest(RenderRequest request, RenderResponse response)
 - doFacesRequest(ResourceRequest request, ResourceResponse response)
 - doFacesRequest(EventRequest request, EventResponse response)
- JBoss PortletBridge
 - **AjaxPortletBridge**

Some inner details ..4

- **Action request example**
 - Bridge acts as a Faces controller in a portlet environment
 - Sets the `javax.portlet.faces.phase` request attribute to `Bridge.PortletPhase.ACTION_PHASE`
 - Acquires a faces lifecycle
 - Acquires the `FacesContext`
 - Determines the target view
 - Executes the action phase of the Faces Lifecycle
 - Recognizes if current request processing is complete.
 - Releases the `FacesContext`
 - Removes the `javax.portlet.faces.phase` request attribute

Life made simpler

- Drop a portlet.xml and bridge jars into your JSF, RF or Seam app
- Or use the archetypes as a guide
- Or use JBoss Tools
- See it all in screen casts at vimeo.com/channels/gatein
- Or use a **tool** to convert (Work in progress)

But that's cheating. Show me some stuff

- **Jars :**

portletbridge-api.jar and portletbridge-impl.jar

- **portlet.xml**

```
<portlet>
```

```
  <portlet-name>NetflixPortlet</portlet-name>
```

```
  <portlet-class>javax.portlet.faces.GenericFacesPortlet</portlet-class>
```

```
  <init-param>
```

```
    <name>javax.portlet.faces.defaultViewId.view</name>
```

```
    <value>/pages/netflix.xhtml</value>
```

```
  </init-param>
```

```
</portlet>
```

- **faces-config.xml (only when using facelets which you should anyway)**

```
<application>
```

```
  <view-handler> org.jboss.portletbridge.application.PortletViewHandler</view-handler>
```

```
  <state-manager>org.jboss.portletbridge.application.PortletStateManager</state-manager>
```

```
</application>
```

End to End In action

- Use maven archetype to create a seam portlet project
- Import project into JBoss Tools (Eclipse + JBoss stuff)
- Deploy the artifact into GateIn portal
- See PRP and Events in action
- <http://vimeo.com/11484018>

Going beyond spec

- Richfaces : Automatic Script wrapping
- Seam SSO with Portal
- Seam sharing beans across portlets (Application Scope annotation)
- EL extension for bridge

All these allow JSF 1.2, JSF2 (wip), Richfaces, and Seam applications to run as portlets.

Richfaces config tips

- In web.xml

```
<context-param>
    <param-name>org.jboss.portletbridge.WRAP_SCRIPTS</param-name>
    <param-value>>true</param-value>
</context-param>
<context-param>
    <param-name>javax.portlet.faces.RENDER_POLICY</param-name>
    <param-value>ALWAYS_DELEGATE</param-value>
</context-param>
<context-param>
    <param-name>org.richfaces.LoadStyleStrategy</param-name>
    <param-value>DEFAULT</param-value>
</context-param>
<context-param>
    <param-name>org.richfaces.LoadScriptStrategy</param-name>
    <param-value>ALL</param-value>
</context-param>
```

Seam config tips

1.

```
<context-param>
```

```
    <param-name>org.jboss.portletbridge.ExceptionHandler</param-name>
```

```
    <param-value>org.jboss.portletbridge.SeamExceptionHandlerImpl</param-value>
```

```
</context-param>
```

2. IMP: Put your navigation in pages.xml. Since a portlet may get rendered many times and seam page actions are performed on the RenderRequest

3. SSO Between Seam and Portal (Bridgelet)

```
<security:portal-identity authenticate-method="#{authenticator.authenticate}"/>
```

Multiple JSF apps on one page

- Bridge takes care of JSF + Portlet namespacing
- Shared custom JS will require its own NS
- To get bridge NS:
`#{facesContext.externalContext.response.namespace}`

Follow & Contribute

- Project Page: <http://www.jboss.org/portletbridge>
- Source: <http://anonsvn.jboss.org/repos/portletbridge>
- User forum: <http://www.jboss.org/index.html?module=bb&op=viewforum&f=273>
- Videos: <http://vimeo.com/channels/gatein>
- Ref Card: <http://refcardz.dzone.com/refcardz/mastering-portals-portlet>

Personal Appeal



<http://www.eejot.org>