



# *Real World Experiences Using New Portal Capabilities to Enhance Application Development*

Graham Harper  
Application Architect  
IBM Software Services for Lotus

**Lotus** software



# ***Contents***

- **Precise layout control with static pages**
- **Enhancing portlet look and feel with Dojo**
- **Creating Ajax portlets**
- **Options for inter-portlet communication and coordination**





# Precise layout control with static pages

**Lotus** software



## ***What is a “static page”?***

- **A new type of portal page**
  - Sits in the portal's normal label / page hierarchy
  
- **Created by defining HTML directly**
  - Instead of letting portal lay out portlets for you in nested row and column containers
  
- **Can be accompanied by related assets, such as:**
  - Graphics
  - Stylesheets
  - JavaScript



## ***Why would you use a static page?***

- **Much more precise control over page layout**
- **Easier to arrange layout via CSS for accessibility**
  - Hard to remove all tables from standard themes and skins
- **Can facilitate easier reuse of non-portal web layouts and prototypes**



WUG List Page

WUG Viewer Page

WUG List Portlet Info

WUG CSA

WUG Non Dojo

WUG Dojo

WUG No Theme Page

You are here: [WUG Demo](#) / [WUG List Page](#)

Search →

Clipboard



[Edit Settings](#)

### Content

★ [First Content Item](#) | 26 Feb 2009 at 11:29AM



★ [Second Content Item](#) | 13 Jan 2009 at 10:14AM



★ [Fourth Content Item](#) | 12 Dec 2008 at 9:13AM



[Edit Settings](#)

### ★ [First Content Item](#)



Published 26 Feb 2009

A really interesting article about the future of ..... [more](#)

### ★ [Third Content Item](#)



Published 12 Dec 2008

The senior vice president was quoted as saying..... [more](#)

### ★ [Fourth Content Item](#)



Published 12 Dec 2008

An entirely expected conclusion to the least controversial news this year..... [more](#)

[Edit Settings](#)

### Content

★ [Second Content Item](#) | 13 Jan 2009 at 10:14AM



★ [Third Content Item](#) | 12 Dec 2008 at 9:13AM



★ [Fourth Content Item](#) | 12 Dec 2008 at 9:13AM



## ***Creating a static page***

- **Create main HTML page and other artefacts (CSS, images etc.)**
  - Put in a directory hierarchy
  - Use relative URLs between artefacts
  
- **ZIP artefacts into a single file**
  - Limited to 1 MB in size
  
- **Create a new page in portal**
  - Select “Static Content” under “Type of Page”
  - Select “Set page layout properties”
    - Upload ZIP file
    - Specify name of main HTML page within file



# Adding dynamic elements to a static page

## ■ Portlets

### – Custom

```
<div class="portlet-window" name="clipboard"  
      style="portlet-definition:'com.ibm.swg.isl.wugdemo.portlets.Clipboard'">  
</div>
```

### – Standard

- *Navigation*
- *Breadcrumb Trail*
- *Page List*

## ■ Containers

- Creates target zone for drag and drop operations





# *Demo 1*



You are here: [WUG Demo](#) / [WUG List Portlet Info](#)

Search →

- WUG Content List Portlet
- Actions
  - Back
  - Edit Shared Settings
  - Restore
  - Minimize
  - Maximize

*Edit Settings*

**Content**

- ★ [First Content Item](#) | 26 Feb 2009 at 11:29AM
- ★ [Second Content Item](#) | 13 Jan 2009 at 10:14AM
- ★ [Fourth Content Item](#) | 12 Dec 2008 at 9:13AM

- WUG Content List Portlet
- Actions
  - Back
  - Edit Shared Settings
  - Restore
  - Minimize
  - Maximize

*Edit Settings*

**Content**

- ★ [First Content Item](#) | 26 Feb 2009 at 11:29AM

WUGClipboardPortlet

Actions

Back

Restore

Minimize

Maximize

Clipboard ▾



## The dynamic markup produced

```
<div class='portlet-window' id='...' name='...' >
  <ul class='xoxo portlet-info' >
    <li class='portlet-title' lang='en'>
      Some Portlet
    </li>
    <li class='portlet-actions' >
      Actions
      <ul class='xoxo portlet-action' >
        <li class='portletoperation-view selected' >
          <a href='...' rel='view' >Back</a>
        </li>
        <li class='portletoperation-normal selected' >
          <a href='...' rel='normal' >Restore</a>
        </li>
        <li class='portletoperation-maximized' >
          <a href='...' rel='maximized' >Maximize</a>
        </li>
        <li class='portletoperation-delete_portlet' >
          <form method='POST' action='...' rel='delete_portlet' >Delete</form>
        </li>
      </ul>
    </li>
  </ul>
  <div class='portlet-window-body' >
    ...
  </div>
</div>
```



## ***Styling the dynamic markup produced***

- Can customise the “virtual skin” markup using CSS
- So if you just wanted to hide the portlet title and actions:

```
<style type="text/css">
  ul.portlet-info {
    display: none;
  }
</style>
```

# *Demo 2*



You are here: [WUG Demo](#) / [WUG List Page](#)

Clipboard

[Edit Settings](#)

**Content**

- ★ [First Content Item](#) | 26 Feb 2009 at 11:29AM
- ★ [Second Content Item](#) | 13 Jan 2009 at 10:14AM
- ★ [Fourth Content Item](#) | 12 Dec 2008 at 9:13AM

[Edit Settings](#)

- ★ **First Content Item**   
Published 26 Feb 2009  
A really interesting article about the future of ..... [more](#)
- ★ **Third Content Item**   
Published 12 Dec 2008  
The senior vice president was quoted as saying..... [more](#)
- ★ **Fourth Content Item**   
Published 12 Dec 2008  
An entirely expected conclusion to the least controversial news this year..... [more](#)

[Edit Settings](#)

**Content**

- ★ [Second Content Item](#) | 13 Jan 2009 at 10:14AM
- ★ [Third Content Item](#) | 12 Dec 2008 at 9:13AM
- ★ [Fourth Content Item](#) | 12 Dec 2008 at 9:13AM

## ***What is actually created for a static page?***

- A page definition in portal
- Can go into “Edit Page Layout” and see it
- More importantly, can create wires
- *Cannot* choose skins for portlets of course



## *How is a static page rendered?*

- **If no “<HTML>” tag in main page markup**
  - Page markup replaces just the “portlet area”
  - Theme navigation elements etc. are still rendered
- **If main page markup has an “<HTML>” tag**
  - Static page forms entire markup delivered by portal
  - Theme is not used





## Source for “full screen” static page

```
<html>
  <head>
    <title>Full Screen Static Page</title>
    <style type="text/css">
      (some styling for the navigation portlet)
    </style>
  </head>
  <body>
    <div>
      <div class="portlet-window" name="main-navigation" style="portlet-definition:wps.p.SpaNavigation">
        <dl>
          <dt>root</dt>
          <dd>.</dd>
          <dt>levels</dt>
          <dd>1</dd>
        </dl>
      </div>
      <div>
        <div class="portlet-window" name="imagePortlet1"
          style="portlet-definition:'com.ibm.swg.issl.wugdemo.portlets.ImagePortlet'"></div>
      </div>
      <div>
        <div class="portlet-window" name="imagePortlet2"
          style="portlet-definition:'com.ibm.swg.issl.wugdemo.portlets.ImagePortlet.2'"></div>
      </div>
    </body>
</html>
```



# *Demo 3*



*WUG List Page*

*WUG Viewer Page*

*WUG List Portlet Info*

*WUG CSA*

*WUG Non Dojo*

*WUG Dojo*

*WUG No Theme Page*





# Enhancing portlet look and feel with Dojo

**Lotus** software



CREATED WITH LOTUS<sup>®</sup> SYMPHONY<sup>™</sup>



## ***Benefits of Dojo inclusion with WebSphere Portal 6.1***

- **Provides an Ajax framework**
  - *More details later...*
  
- **Includes many JavaScript helper features**
  - Especially for DOM manipulation and styling
  - Not very exciting to demonstrate!
  
- **Plus Dojo user interface widgets (“Dijits”)**
  - Layout components
  - Form elements
  - Rich text editor
  - Menus and dialogs
  - Many more...



# *Demo 4*



# Basic portlet from RAD wizard

**WUGBasicNonDojoPortlet**

**Order Details**  
Fill in details of your order:

Order id:

Order date:

Order amount:

**WUGBasicDojoPortlet**

**Order Details**  
Fill in details of your order:

Order id:

Order date:

Order amount:

**WUGBasicDojoPortlet**

**Order Details**  
Fill in details of your order:

Order id:

Order date:

Order amount:

March						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11
2008			2009	2010		



## Steps for using Dojo in portlets

- “Require” the necessary modules:

```
dojo.require("dijit.form.DateTextBox");
```

- Create dijit content in markup or code:

```
<input name="formDate" type="text" dojoType="dijit.form.DateTextBox" />
```

- If created in markup, parse it (Portal turns off “parse on load”):

```
dojo.parser.parse(dojo.byId("PortletContent_<portlet:namespace/>"));
```





***More dijits than just form elements...***

# ***Demo 5***



## ***Dojo context menu***



# Dojo dialog for Edit Defaults mode

Edit Settings

## Content

★ First Content Item | 26 Feb 2009 at 11:29AM

★ Second Content Item | 13 Jan 2009 at 10:14AM

★ Fourth Content Item | 12 Dec 2008 at 9:13AM



Clipboard



Edit Settings

## Content

★ Second Content Item | 13 Jan 2009 at 10:14AM

★ Third Content Item | 12 Dec 2008 at 9:13AM

★ Fourth Content Item | 12 Dec 2008 at 9:13AM

Content selection:

List Template:

Detail Template:

Save and Close

Cancel



# Dojo dialog and stack container = wizard

**Add to Subscriptions**

Any content by author

Any content in publication



**Add to Subscriptions**

person



**Add to Subscriptions**

The subscription has been saved.

Go to the Subscription Manager.



# Creating Ajax portlets

**Lotus** software



CREATED WITH LOTUS® SYMPHONY™



## ***Client-side aggregation***

- **“Free” (very basic) Ajax enablement for any portlet**
  - Use the supplied “PortalWeb2” theme in 6.1
- **Framework of page rendered on initial page load**
- **Portlet content rendered as response to subsequent asynchronous calls**
  - A little like parallel portlet rendering although the user gets to see and interact more whilst waiting



# *Demo 6*



# Page with PortalWeb2 theme

**WebSphere Portal** | **WUG Demo** | Home | Administration | Applications | More... | wpsadmin | Edit My Profile | Create Link to this Page | Help | Log Out

**WUG Demo** | **WUG List Page** | **WUG Viewer Page** | **WUG List Portlet Info** | **WUG CSA** | **WUG Non Dojo** | **WUG Dojo** | **WUG No Theme Page**

All Sources

### WUGImageSelector

Grow graphic

Industry graphic

### WUG Image Portlet

**Your world  
Your way**

### WUG Image Portlet 2

**Portals are all about  
people**



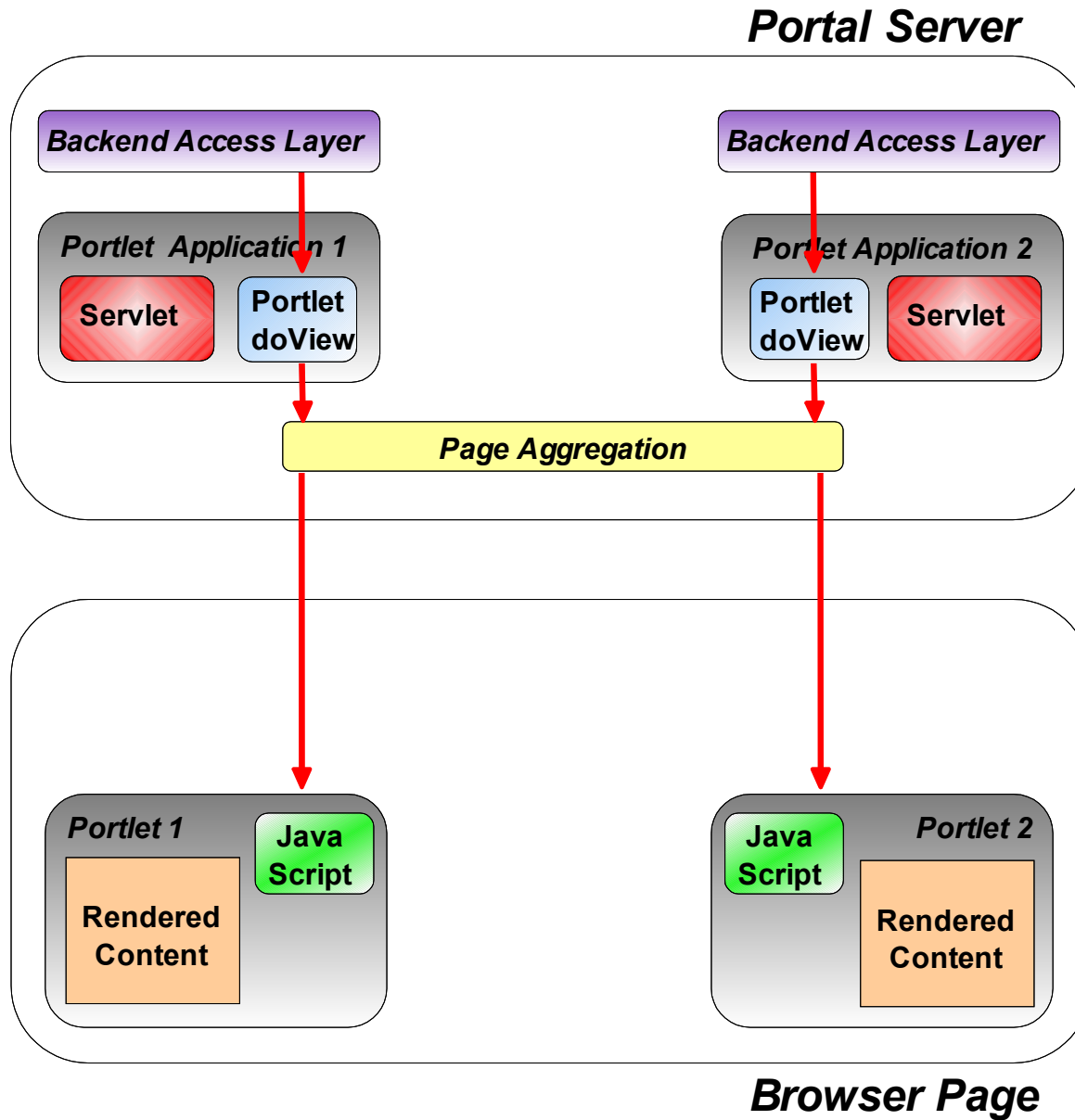
## ***Ajax for custom portlets before WebSphere Portal 6.1***

- **Common pattern:**

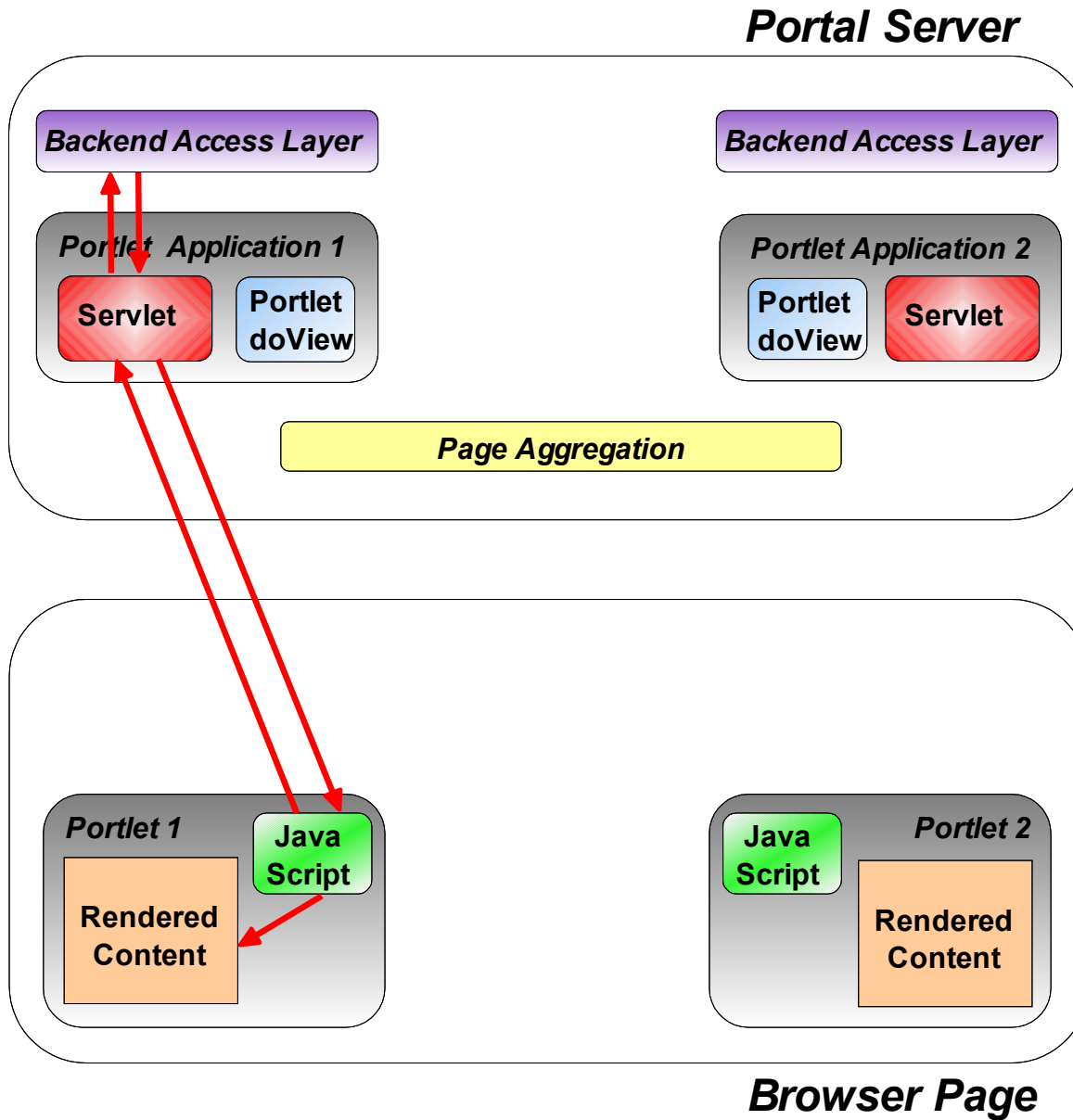
- Create servlet in same web application as portlet
- Keep common state information in application session
- Import JavaScript framework of choice into portlet
- Client-side JavaScript code provided by portlet makes asynchronous requests to servlet
  - Servlet updates state in application session so changes are not lost when portlet reloaded (e.g. by actions on other portlets on page)



*Initial rendering of Ajax-ready page (with content)*



**Update of content via Ajax**

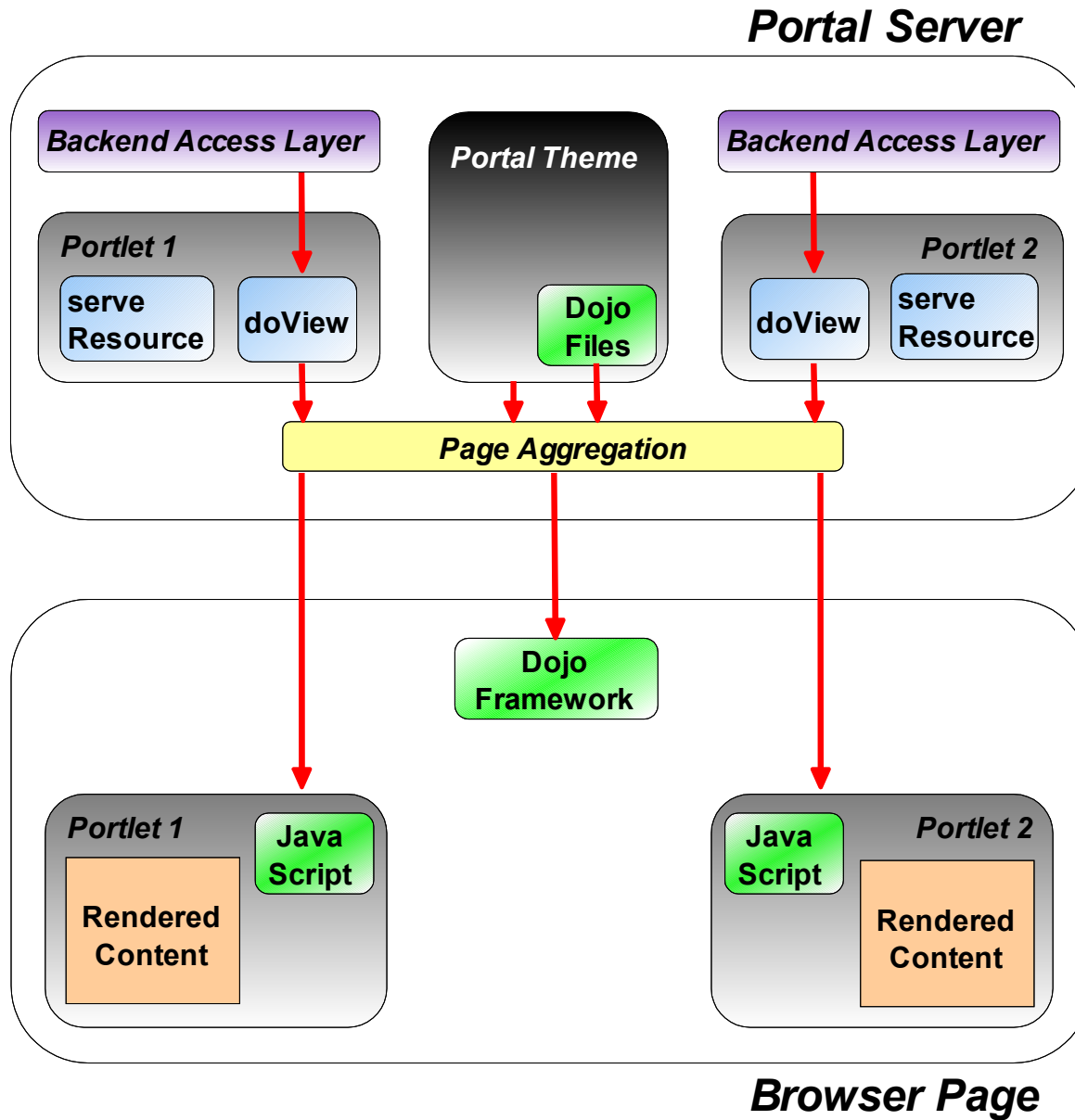


## ***Ajax for custom portlets in WebSphere Portal 6.1***

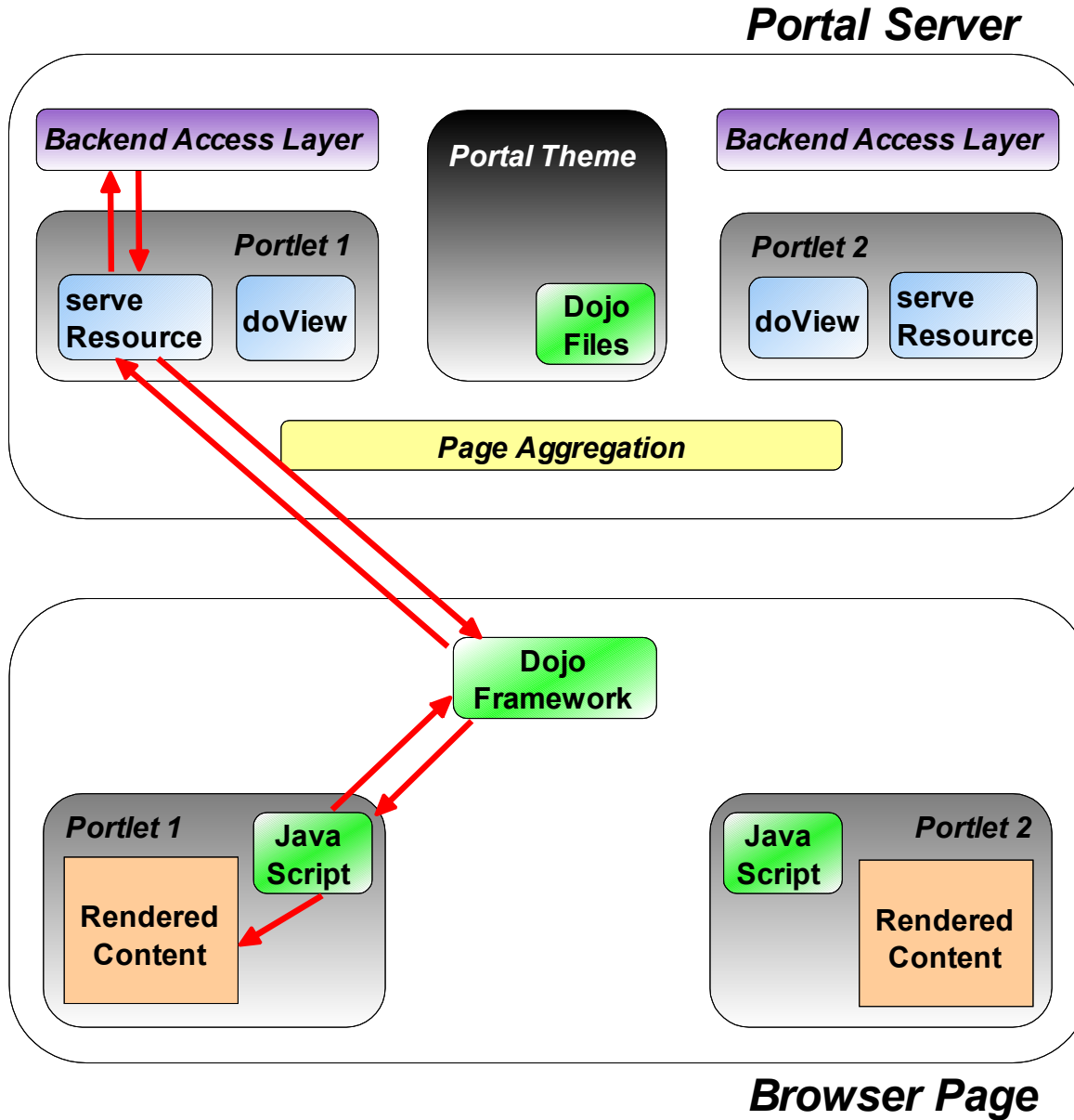
- **JSR 286 adds “serveResource” method in portlet API**
  - In addition to “doView”, “processAction” etc.
  - Other portlets and theme artefacts not rendered in the response from this method
  - Removes the need for a separate servlet in the portlet web application
  
- **Standard portal theme bundles Dojo JavaScript framework**
  - Gives support for the client-side parts of Ajax operations
  - e.g. “xhrGet” method to encapsulate XML HTTP requests cross-browser



*Initial rendering of Ajax-ready page (with content)*



**Update of content via Ajax**



# *Demo 7*



## ***Ajax implementation details...***

- Can use JSON rather than XML as a transport format
  - Easier to consume in JavaScript
- To prevent the need for multiple calls (and if data set is small), can also return a new data set as response to update operations
- Can put server code in a portlet base class, if used across multiple portlets
- Much client code is in form of “callback” functions for successful and unsuccessful server requests







## ***Options for inter-portlet communication and coordination***

**Lotus** software



CREATED WITH LOTUS® SYMPHONY™



## ***Many new options for communication and coordination***

- **Updated server-side inter-portlet communication**
  - JSR 286 events
- **Client-side inter-portlet communication**
  - Click-to-action
  - Dojo publish / subscribe
- **Public render parameters**



## ***Use cases for these techniques***

- **Coordinated response to user action**
  - Server-side inter-portlet communication
  - Client-side inter-portlet communication
  
- **Bookmarking (and return to a particular coordinated state across portlets)**
  - Public render parameters



## ***Server-side inter-portlet communication with JSR 286***

- Adds `processEvent` method to the Portlet class
- Can send an event from `processAction` using `setEvent` in the response object:

```
response.setEvent(EVENT_SELECT_CONTENT, contentId);
```

- Events declared in `portlet.xml` file
- Wired together as in previous versions of portal
  - For cross-page wiring, the recipient must be flagged as a “global target” via the wiring tool



## Extract from portlet.xml

```
<portlet-app xmlns="..." version="2.0" xmlns:xsi="..." xsi:schemaLocation="..."
  id="..." xmlns:x="http://WUGDemo/">
  ...
  <portlet>
    <portlet-name>ContentListPortlet</portlet-name>
    ...
    <supported-processing-event>
      <name>SelectContent</name>
    </supported-processing-event>
    <supported-publishing-event>
      <name>SelectContent</name>
    </supported-publishing-event>
    ...
  </portlet>
  ...
  <event-definition>
    <description>Content item is selected</description>
    <name>SelectContent</name>
    <alias xmlns:customns="http://WUGDemo/">customns:ContentId</alias>
    <value-type>java.lang.String</value-type>
  </event-definition>
  ...
</portlet-app>
```

# *Demo 8*



## ***Client-side inter-portlet communication***

- **WebSphere Portal 6.1 provides “live text” markup for click-to-action**
  - “Hover” menu appears on sources and shows available targets
- **Targets are coded as HTML forms**
  - On receiving event can post form to server URL
  - Or can execute JavaScript function as “onSubmit” handler
  - Therefore action can be processed client-side or server-side
- **Only strings can be sent**
  - Namespaced for disambiguating different logical types
  - Message also stored in JavaScript variable for convenience:  
`window.ibm.portal.c2a.event.value`



## Source markup (in JSP)

```

<div style="margin: 6px">
<div class="c2a:source">
  <span class="c2a:typename" style="display:none">http://www.ibm.com/xmlns/prod/datatype#filename </span>
  <span class="c2a:value" style="display:none"><%=renderResponse.encodeURL(renderRequest.getContextPath()
+
"/images/portals-industry.jpg") %>
  </span>
  <span>Industry graphic</span>
  "/>
  <p class="c2a:display" style="display:none;">
    <b>Send image change</b>
  </p>
</div>
</div>

```

## Target markup (in JSP)

```

<form class="c2a:target" onSubmit="swapImage_<portlet:namespace/>(window.ibm.portal.c2a.event.value);return false"
      action="javascript:void(0)" style="display:none">
  <span class="c2a:typename">http://www.ibm.com/xmlns/prod/datatype#filename</span>
  <p class="c2a:action-label">Change image</p>
  <input type="text" class="c2a:action-param" name="imagePath">
</form>

```





# *Demo 9*



## *Client-side inter-portlet communication (revisited)*

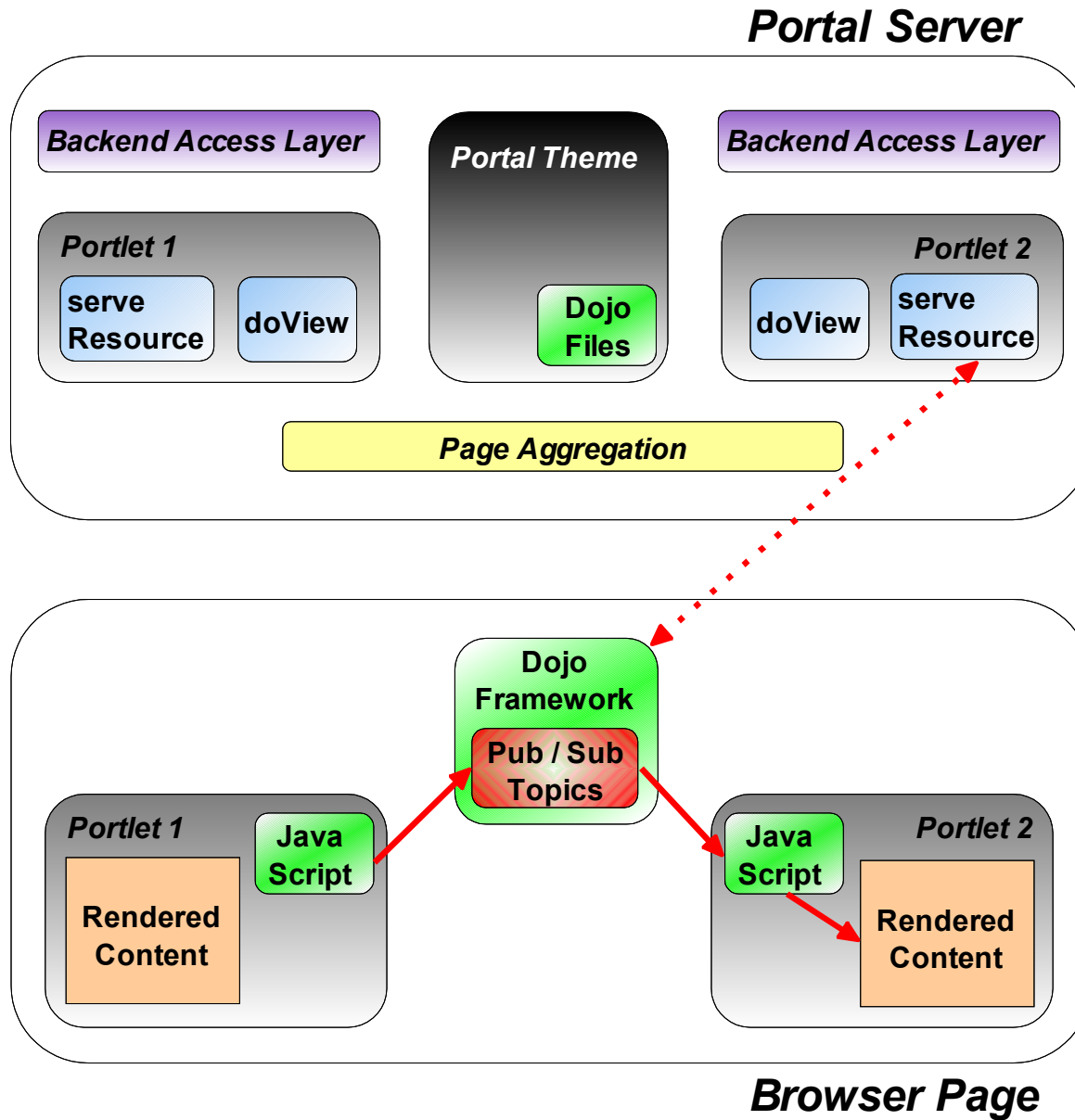
- **But what if you need programmatic communication client-side?**
  - Can use Dojo publish / subscribe support:

```
dojo.subscribe("<%=Constants.TOPIC_CLIPBOARD%>",  
              refreshClipboard_<portlet:namespace/>);
```

```
dojo.publish("<%=Constants.TOPIC_CLIPBOARD%>", ["<portlet:namespace/>"]);
```



**Dojo-based  
inter-portlet  
communication  
and content  
update**



# *Demo 10*



## ***Dojo-based client-side IPC details...***

- **When subscribing, you provide a function that is called when someone publishes to the topic**
  - Gets any parameters publisher sends
  - For instance, can send the portlet namespace so you can ignore your own events
  
- **Publish changed data or just notification?**
  - Receiver could request data from own portlet's `serveResource` method
    - Ensures good encapsulation
  - Sender could include change in common data if that is what is being notified (e.g. clipboard contents)
    - Better performance (fewer server requests)



## ***Public render parameters***

- Encoded in URL (*like all render parameters*)
- Read as request parameters (*like all render parameters*)
- Set on response in action or event phases (*like all render parameters*)
- Declared in `portlet.xml`
  - Namespace is *required*
- Allow sharing of data during render phase between participating portlets



## Public render parameters in portlet.xml

```
<portlet-app xmlns="..." version="2.0" xmlns:xsi="..." xsi:schemaLocation="..."
  id="..." xmlns:x="http://WUGDemo/">
  ...
  <portlet>
    <portlet-name>ContentListPortlet</portlet-name>
    ...
    <supported-public-render-parameter>contentId
      </supported-public-render-parameter>
    ...
  </portlet>
  ...
  <public-render-parameter>
    <identifier>contentId</identifier>
    <qname>x:contentId</qname>
  </public-render-parameter> ...
</portlet-app>
```

## ***Use cases for public render parameters***

- **Provide shared context between portlets**
  - With no wiring needed
  
- **Better than using the application session in some cases**
  - Persistent across sessions if bookmarked
  - No action or event phase required to recreate
    - Useful as, for instance, opening a page only calls the render phase
  
- ***Example:* the Content List portlet sets parameter in the event phase when the cross-page wire is activated:**

```
response.setRenderParameter(RENDER_PARAMETER_CONTENT_ID, (String) contentIdObject);
```

- Allows a bookmarked Content Viewer page to open original article again in all portlets (i.e. in Related Content too)





# *Demo 11*





Questions?

**Lotus** software



CREATED WITH LOTUS® SYMPHONY™

