# JSR-286: Portlet Specification 2.0

## Upcoming enhancements and new features for Portal and Portlet Developers

Ate Douma
JSR-286 Expert Group

Software Architect
Hippo – Open Source Content Management Software
ate@apache.org / a.douma@hippo.nl

# Preamble

*This presentation only represents the current thinking of the JSR-286 Expert Group.
The final specification may differ from the content of this presentation.*

# Agenda

- A short summary of JSR-168: Portlet Specification 1.0
- What is missing from JSR-168
- JSR-286: Enhancements and new features for V2.0
- Questions and Answers

# JSR-168 – A short summary

PLT.2.1 What is a Portal?

*“ A portal is a web based application that -commonly- provides personalization, single sign on, content aggregation from different sources and hosts the presentation layer of Information Systems.”*

# JSR-168 – A short summary

## PLT.2.2 What is a Portlet?

*“A portlet is a Java technology based web component managed by a portlet container that processes requests and generates dynamic content.*
*Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems.”*

# JSR-168 – A short summary

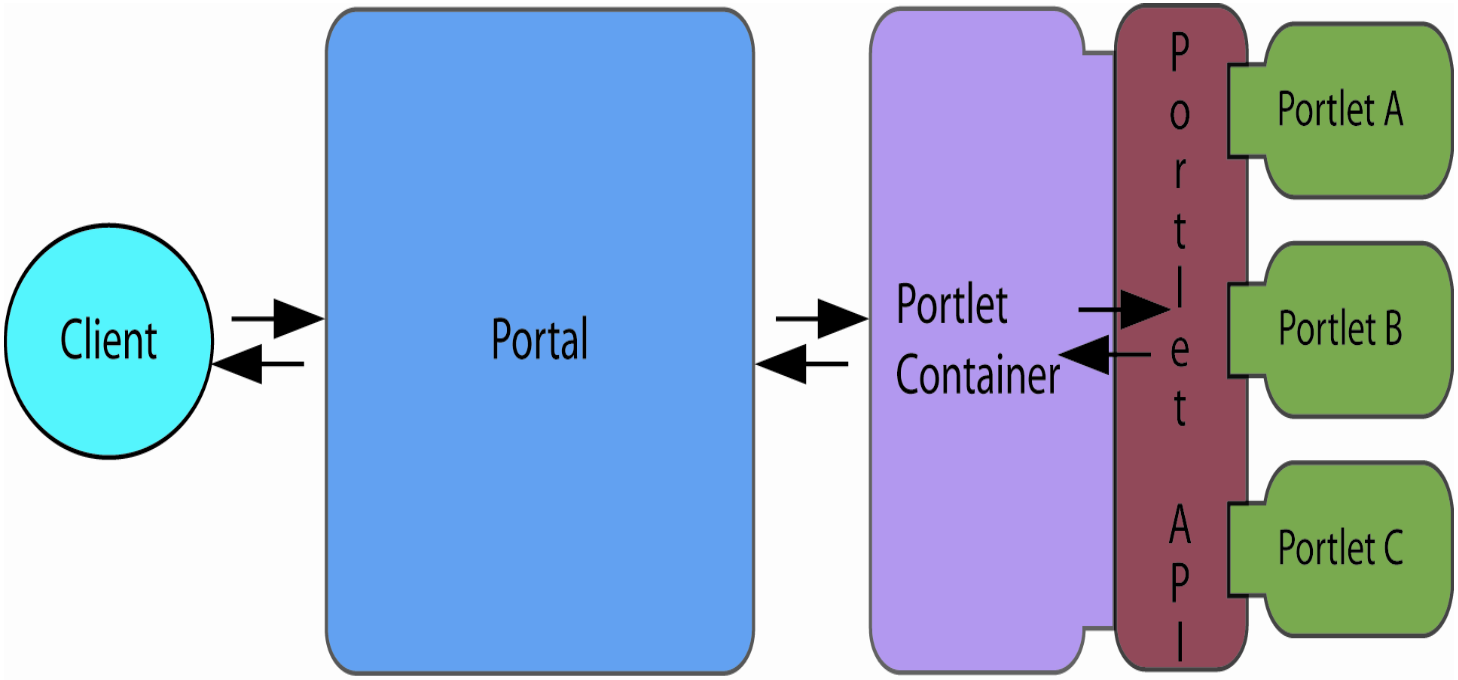
## PLT.2.3 What is a Portlet Container

*“A portlet container runs portlets and provides them with the required runtime environment. A portlet container contains portlets and manages their lifecycle.”*

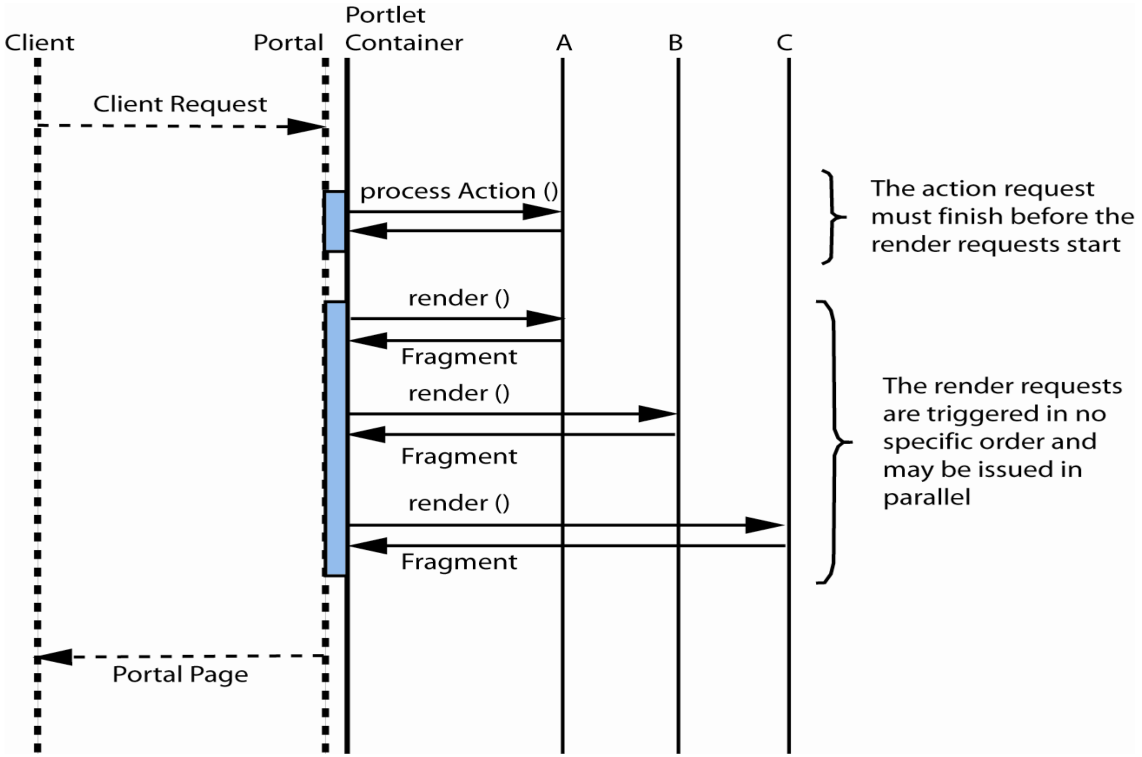
...

*“A portlet container receives requests from the portal to execute requests on the portlets hosted by it.”*

# JSR-168 – A short summary



# JSR-168 – A short summary


Client
Portal
Portlet Container
A
B
C
Client Request
process Action ()
render ()
Fragment
render ()
Fragment
render ()
Fragment
Portal Page
The action request must finish before the render requests start
The render requests are triggered in no specific order and may be issued in parallel

*Request processing*

http://www.hippoportal.org/display/PORTAL/JSR-286

# JSR-168 – What is missing?

- inter-portlet communication
- serving non-html resources (pdf, doc, images etc.)
- contributing javascript or css to <head>, using cookies
- proper support for common web frameworks
- portlet filters

# JSR-168 – What is missing?

Inter-portlet communication

- only supported within the same portlet application using session attributes
- target portlets will only “see” messages during next render request
- portlets cannot (should not) update their state during a render request: “event” handling not really possible

# JSR-168 – What is missing?

Serving non-html resources

- A portlet can only render html fragments
- Have to fallback/delegate to the servlet container
- Requires coordination between portlet and servlet

# JSR-168 – What is missing?

Contributing to <head>, setting cookies

- javascript or css can only be embedded withing the content markup; no body onLoad handling hooks
- API forbids adding cookies: only client side setting of cookies using javascript is possible

# JSR-168 – What is missing?

Proper support for common web frameworks

- Most web frameworks are Servlet API only
- Servlet dispatching not supported from processAction
- Needs Portals Bridges or similar solutions
- JSTL support very limited:

```
<c:out value=”<%= ((FooBean)renderRequest.getSession()
                        .getAttribute("fooBean",PortletSession.PORTLET_SCOPE))
                           .getBeanValue() %>"/>
```

# JSR-286 - Coming up

- Portlet Specification 2.0
- Expert Group started January 2006
- 1st Early Public Draft released August 2006 covering most, but not yet all, planned features
- 2nd Early Public Draft November 2006 covering all features
- 1st Public Draft December 2006
- Final Release May 2007
- RI will be done under Apache Pluto umbrella with help from a group at University of Jena

# JSR-286 - Coming up

- Binary compatible with JSR-168
- Alignment with J2EE 1.4, WSRP 2.0
- Portlet coordination
  - Public render parameters
  - Shared session state across applications (maybe)
  - Portlet events
- Resource serving
- AJAX support
- Portlet filters
- Extended cache support
- Improved support for common web frameworks
- ... and more ...

# JSR-286 – Coming up

## Public render parameters

- Allow coordination of render state with other portlets across web applications
- Limited to String values
- Defined in portlet.xml:

```
<shared-render-parameter>
  <name>myfoo</name>
  <name>foo</name>
</shared-render-parameter>

<portlet>
  <portlet-name>Portlet-A</portlet-name>
  ...
  <supported-shared-render-parameter>
    <name>myfoo</name>
  </supported-shared-render-parameter>
</portlet>
```

# JSR-286 – Coming up

## Shared session state

- Share user session data independent of navigational state across web applications
- Managed by the portal, not the portlet container
- Defined in portlet.xml:

```
<shared-application-session-attribute>
  <name>com.foo.bar</name>
  <name>com.acne.foo.bar</name>
</shared-application-session-attribute>

<portlet>
  <portlet-name>Portlet-A</portlet-name>
  ...
  <shared-portlet-session-attribute>
    <name>com.my.foo.bar<name>
    <java-class>com.foo.bar.FooBar</java-class>
  </shared-portlet-session-attribute>
</portlet>
```

# JSR-286 – Coming up

Shared session state (continued)

- Must be serializable and have a valid JAXB 2.0 binding
- HttpSessionBindingListener supported

*but:*

- Difficult to implement and use reliable
- Still under investigation if really needed
- Maybe dropped again as Portlet events might already provide enough functionality

# JSR-286 – Coming up

## Portlet events

- A portlet declares the events it wants to receive or emit in portlet.xml:

```
<event-definition>
  <name>com.foo.bar</name>
  <java-class>java.lang.String</java-class>
</event-definition>

<portlet>
  <portlet-name>Portlet-A</portlet-name>
  ...
  <supported-processing-event>
     <name>com.foo.bar<name>
  </supported-processing-event>
  <supported-publishing-event>
    <name>com.foo.bar<name>
  </supported-publishing-event>
</portlet>
```

- Emitting dynamic non-declared events is allowed too

# JSR-286 – Coming up

Portlet events (continued)

- Allows wiring of portlets at runtime
- The portal / portlet container will act as broker
- The portal or portlet container can also emit events
- Non-reliable, i.e. no guarantee of delivery
- New 3rd lifecyle phase: before rendering
- API:

```
EventPortlet.processEvent(EventRequest req,EventResponse res)
StateAwareResponse.setEvent(String name,Object value)
StateAwareResponse.setEvents(Map events)
```

- Can be created from processAction and processEvent
- State changes are allowed during processEvent

# JSR-286 – Coming up

## Resource serving

- New type of PortletURL and request handler
- API:

```
PortletResponse.createResourceURL()
ResourceServingPortlet.serveResource(ResourceRequest, ResourceResponse)
```

- Extends the render phase, NOT a new lifecycle phase
- Cannot change state
- Additional URL parameters are specific for the request
- Full control over request and response headers
- Can be used for binary data or readonly AJAX

# JSR-286 – Coming up

AJAX support allowing state changes (TBD)

- Still under discussion
- Several issues need to be addressed
    - AJAX request identification
    - synchronizing state changes to the client for other portlets on the page
    - XMLPortletRequest or plain XMLHttpRequest
    - integration with popular AJAX toolkits

# JSR-286 – Coming up

## Portlet filters

- Allow on the fly transformations of information in both the request to and the response from the portlet
- Modelled after Servlet filters
- Defined in portlet.xml:

```
<filter>
  <filter-name>Event Filter</filter-name>
  <filter-class>com.acme.EventFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>Event Filter</filter-name>
  <portlet-name>SamplePortlet</portlet-name>
  <lifecycle>EVENT</lifecycle>
</filter-mapping>
```

- Targetted at all lifecyle calls (*) by default

# JSR-286 – Coming up

## Extended cache support

- Allow public cached content for multiple users

```
<portlet>
  ...
  <expiration-cache>
    <expiration-time>300</expiration-time>
    <scope>public</scope>
  </expiration-cache>
</portlet>
```

- Support validation based caching using ETAG
- API:

```
response.getCacheControl()
  .getExpirationTime(), .setExpirationTime()
  .getPublicScope(), .setPublicScope(boolean)
  .getETag(), .setETag(String)
  .useCachedContent(), .setUseCachedContent(boolean)
```

# JSR-286 – Coming up

Improved support for web frameworks (TBD)

- Allow servlet dispatching during all lifecycle calls:
  - processAction
  - processEvent
  - render
  - serverResource
- Extended JSP tag library
- <defineObjects/> support for JSF

# JSR-286 – Coming up

... and more ...

- PortletRequest.getWindowID()
- Java 5 annotation support for events in GenericPortlet:

```
@ProcessEvent(Retention=RUNTIME, name=<event name>)

void <methodname> (EventRequest, EventResponse)
  throws PortletException, java.io.IOException;
```

- TBD
  - Contributing to HTTP and HTML headers
  - JSR-188: Composite Capability/Preference Profiles

# More information

- JSR-286: Portlet Specification 2.0
  http://jcp.org/en/jsr/detail?id=286
- Online current early draft (non-official)
  http://ipc658.inf-swt.uni-jena.de/spec/
- Apache Pluto
  http://portals.apache.org/pluto/
- RI development
  http://svn.apache.org/repos/asf/portals/pluto/branches/1.1-286-COMPATIBILITY/

# Questions and Answers