



Title: Portal Standards support in WebSphere Application Server and WebSphere Portal (JSR 168 / 286, WSRP 1.0 / 2.0)

Speaker: Andreas Brunnert (brunnert@de.ibm.com)
WebSphere Portal Development
IBM Germany R&D Labs, Boeblingen

Objectives

- Learn what's new with JSR 286 and WSRP 2.0
- Learn what you can do with portlets in WebSphere Application Server
- Get to know the relationship of the WebSphere Application Server portlet runtime and WebSphere Portal 6.1
- Learn how to implement isolation scenarios using the WebSphere Application Server WSRP producer

Agenda

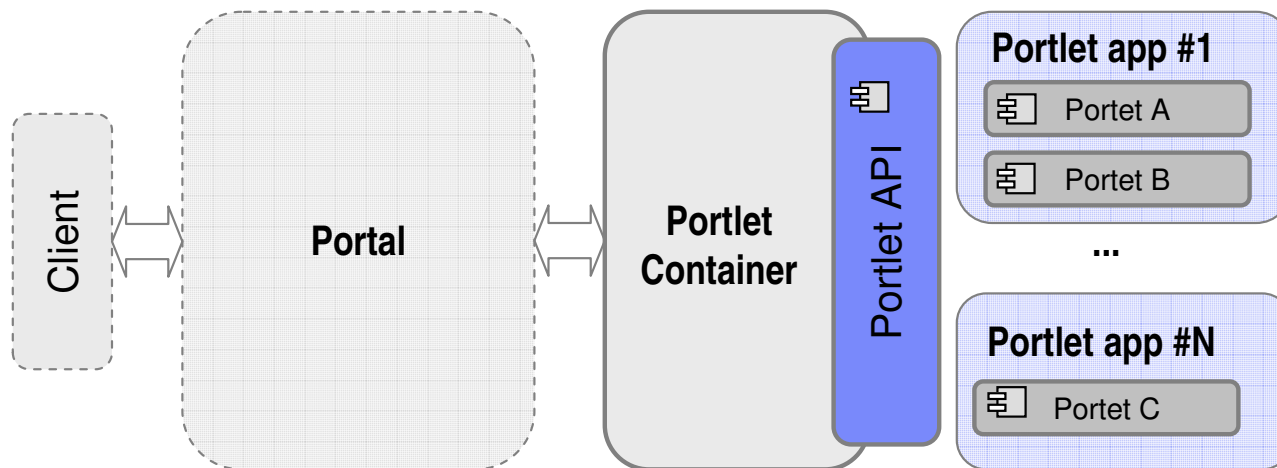
- Portlet Standards growth
- Portlet support in WebSphere Application Server
- Relationship with WebSphere Portal 6.1
- WebSphere Application Server WSRP producer
- Migrating to WebSphere Portal
- Outlook
- Summary

Agenda

- Portlet Standards growth
 - ▶ Portlet specification 1.0 (JSR 168) / 2.0 (JSR 286)
 - ▶ Webservices for Remote portlets (WSRP) 1.0 / 2.0
- Portlet support in WebSphere Application Server
- Relationship with WebSphere Portal 6.1
- WebSphere Application Server WSRP producer
- Migrating to WebSphere Portal
- Outlook
- Summary

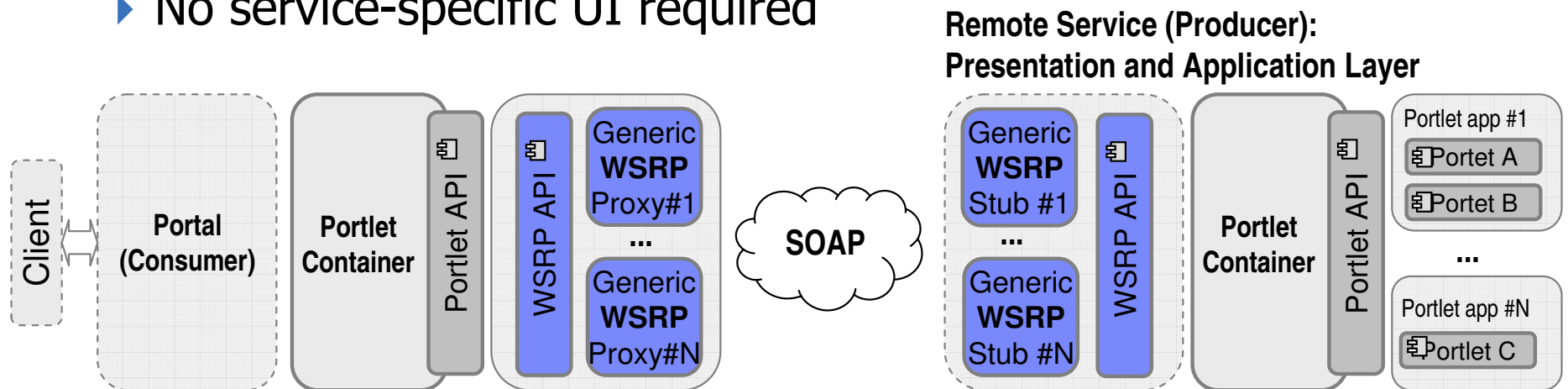
Scope of the Portlet Specification (JSR 168 / JSR 286)

- Portlet API and portlet container
- Contract between the API and the container
- Deployment unit: portlet application
- Not:
 - ▶ Aggregation, layout management
 - ▶ Page personalization and configuration engines
 - ▶ Portal administration and configuration...



Scope of Webservices for Remote Portlet (WSRP)

- Enable the sharing of portlets over the Network with a common interface
 - ▶ Presentation-oriented WebServices
 - ▶ Cross vendor publishing and consuming of applications and content (also cross language: .NET, Java...)
- Pluggable, NO coding required to integrate new services
 - ▶ Proxy and stub are coded once or generated automatically
 - ▶ No service-specific UI required



Where do we want to go from 1.0?

- 1.0 design goal
 - ▶ Provide the programming model for **standalone**, pluggable UI application components
 - ▶ Portlet communication via common session
 - For portlets bundled into same web application
 - ▶ Applies to WSRP 1.0 & Java Portlet API 1.0 (JSR168)
- 2.0 design goal
 - ▶ Define how portlets may be **coordinated** and react as a whole
 - Allows building composite applications based on portlet components
 - ▶ Allow for a better user experience using AJAX patterns
 - ▶ Portlet communication cross web application / producer boundaries
 - ▶ Applies to WSRP 2.0 & Java Portlet API 2.0 (JSR286)

Portlet specification 2.0 (JSR 286) – Coordination Features

■ Events

- ▶ A portlet can declare events it wants to receive and events it wants to emit
- ▶ The portal / portlet container will act as broker and distribute the events accordingly
- ▶ Allows wiring of portlets at runtime
- ▶ Used for complex data types – action semantic
 - New processEvent lifecycle method

■ Public Render Parameter

- ▶ Extension to the portlet navigational state, managed by the Portal
 - Shares render parameters between portlets
 - Enables portlets to react in a coordinated manner
- ▶ Simple types – no extra lifecycle

Portlet specification 2.0 (JSR 286)

- Resource Serving
 - ▶ Primarily an AJAX driven feature to serve any content within the Portlet and have access to the full portlet state at the same time
 - ▶ Portlets will not be rendered as markup fragments and therefore control the output stream themselves
 - ▶ New `serveResource` lifecycle method
- Extended Request Dispatcher Capabilities:
 - ▶ Better support of web frameworks on top of portlets
 - ▶ Request dispatching is now allowed for all lifecycle methods
 - No markup can be returned for action / event
 - ▶ Request dispatcher forward is now allowed for all lifecycle methods
 - Delegate to servlets for action handling
 - Delegate to JSP's for complete markup generation

Portlet specification 2.0 (JSR 286) - Miscellaneous

- Caching
 - ▶ New API allows to get and set cache settings
 - ▶ Shared cache entries
 - ▶ Validation based caching
- PortletFilter
 - ▶ Define filters to intercept the portlet Invocation or the URL creation
- Extended runtime Id's
 - ▶ Portlet can now access the portlet window ID at the request
- PortletURL now accepts a writer
- CC/PP (JSR 188) support
- Restricting the custom window states for a given markup
- Lots of small clarifications and clean up...

WebServices for Remote Portlets (WSRP) 2.0

- Aligned with JSR 286
 - ▶ Supports Coordination Features:
 - Eventing
 - Public Render Parameter
 - ▶ Supports resource serving
 - ▶ ...
- WSRP specific new features:
 - ▶ Leasing
 - Improve Producer resource management by providing lifetimes for portlet instances

Agenda

- Portlet Standards growth
- Portlet support in WebSphere Application Server
 - ▶ Motivation
 - ▶ Capabilities
 - ▶ Architectural Overview
 - ▶ Access Portlets
 - ▶ Demo
- Relationship with WebSphere Portal 6.1
- WebSphere Application Server WSRP producer
- Migrating to WebSphere Portal
- Outlook
- Summary

WAS Portlet Support - Motivation

- Encourage a consistent Programming model
 - ▶ IBM recommendation:
 - Use portlets as UI components
 - Use servlets to implement services
- Makes this programming model available for single components
 - ▶ Easy portlet integration into web applications
 - ▶ Same programming model can be used to define business mashups (aka composite applications) within WebSphere Portal
- Make the Portlet Programming Model as easy to develop and access as Servlet Programming Model

WAS Portlet Support - Capabilities

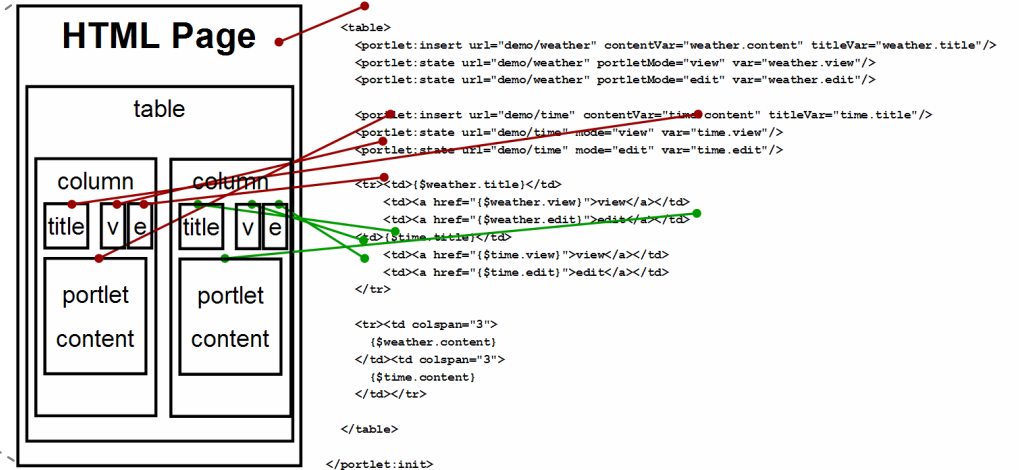
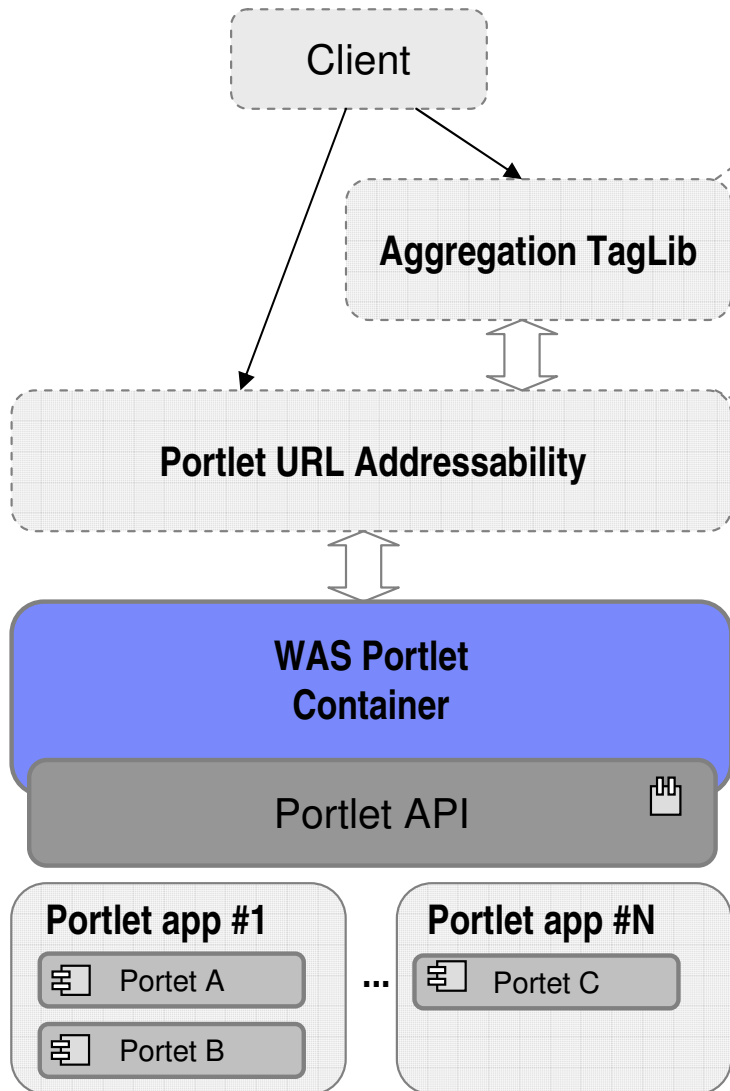
- Basic portlet support has been introduced with WAS 6.1
 - ▶ JSR 168 compliant portlet container
 - ▶ Two Basic Portals: „Url Addressability“ and the „aggregation Taglib“
- Extended support for JSR 286 with WAS 7.0
- Manage Portlets and the PortletContainer
 - ▶ Using ISC (admin console) as known from servlets and the webcontainer
- Access Portlets
 - ▶ Direct „URL Addressability“ of portlets
 - Portlets are accessible via URL as known for Servlets
 - `http://<host>:<port>/<context-root>/<portlet-name>`
 - ▶ Aggregation of portlets using the „aggr. TagLib“

WAS Portlet Support - Capabilities

- Performance Measurement
 - ▶ Request Metrics
 - Log all portlet lifecycle invocations
 - ▶ Performance Measurement Infrastructure (PMI)
 - Displays average performance indicators for portlets
- Portlet Caching
 - ▶ Leverages Dynacache
 - ▶ Defined using cachespec.xml and the portlet.xml cache configuration
 - ▶ Fragment Caching
- Security
 - ▶ Define security constraints within the portlet.xml / web.xml
 - ▶ Standard J2EE security

WAS Portlet Support - Architectural Overview

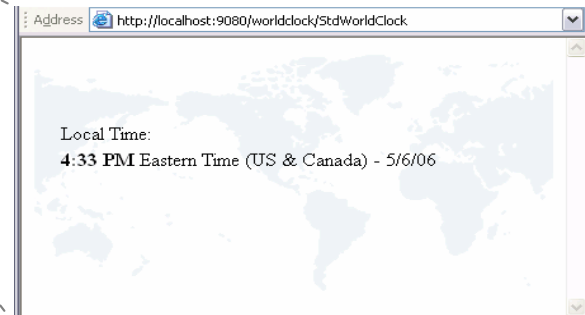
-> Aggregate Portlets in jsp pages using a simple JSP tag library



```

<portlet:init portletURLPrefix=" http://localhost/hello/framework/" portletURLSuffix="/something"
portletURLQueryParams="p1=v1&p2=v2">
</portlet:init>
<table>
<tr>
<td colspan="3">
($weather.content)
</td>
<td colspan="3">
($time.content)
</td>
</tr>
</table>
</portlet:init>

```



-> access Portlets as known from Servlets using direct URL „addressability“
.../<context>/<PortletName>

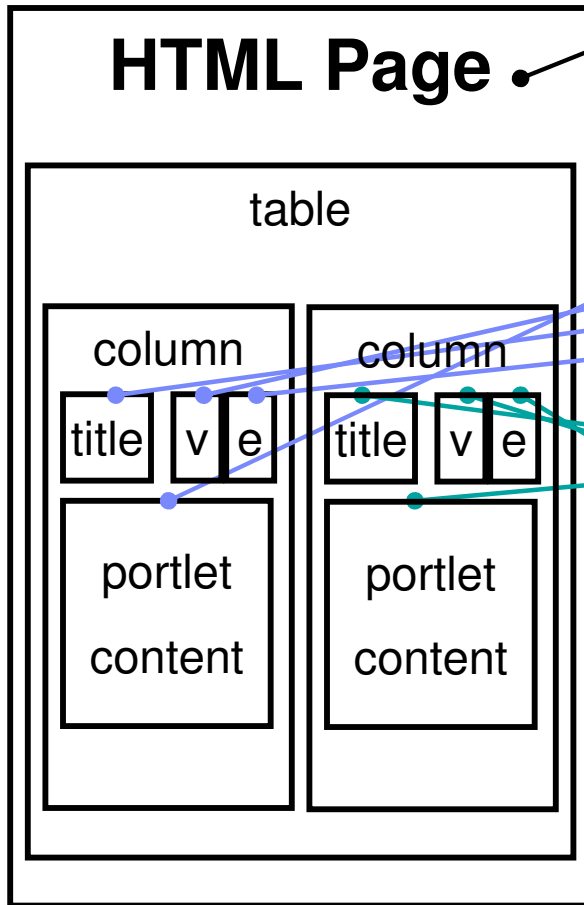
-> Implements the Portlet Standards and enables Portals build on top to exploit those features by implementing defined extension points
 -> it is up to the Portal implementation what is supported and how a specific feature works e.g. Preference Persistence, URL handling...
 -> provides capabilities for portlet and portletcontainer management using common WAS configuration means

WAS Portlet Support - Access Portlets - URL Addressability

- Complete portlet „URL Addressability“ – URL pattern with example

http://<host>:<port>	http://localhost:9080
/<context-root>	/worldclock
/<portlet-name>	/StdWorldClock
/<portletwindow>	/default
/<ver>	/ver=2.0
/resource/<id>	/resource/id=image.jpg
/action	/action
/<mode>	/mode=edit
/<state>	/state=maximized
/<rparam>	/rparam= timezones=UTC=MEZ
<query parameter>	?timezone=UTC

WAS Portlet Support - Access Portlets - aggregation Taglib



```
<portlet:init portletURLPrefix=" http://localhost/hello/framework/" portletURLSuffix="/something"
portletURLQueryParams="pi=v1&p2=v2">

<table>
  <portlet:insert url="demo/weather" contentVar="weather.content" titleVar="weather.title"/>
  <portlet:state url="demo/weather" portletMode="view" var="weather.view"/>
  <portlet:state url="demo/weather" portletMode="edit" var="weather.edit"/>

  <portlet:insert url="demo/time" contentVar="time.content" titleVar="time.title"/>
  <portlet:state url="demo/time" mode="view" var="time.view"/>
  <portlet:state url="demo/time" mode="edit" var="time.edit"/>

  <tr><td>{$weather.title}</td>
    <td><a href="{ $weather.view }">view</a></td>
    <td><a href="{ $weather.edit }">edit</a></td>
  </tr>
  <tr><td>{$time.title}</td>
    <td><a href="{ $time.view }">view</a></td>
    <td><a href="{ $time.edit }">edit</a></td>
  </tr>

  <tr><td colspan="3">
    {$weather.content}
  </td><td colspan="3">
    {$time.content}
  </td></tr>

</table>

</portlet:init>
```

WAS Portlet Support - Access Portlets - Details

- Address portlets directly via URL request
 - ▶ Returning portlet output as HTML document, by default
 - ▶ Example: `http://localhost:9080/worldclock/StdWorldClock`
- Include portlets as fragments by any Servlet via RequestDispatcher
 - ▶ Returning portlet fragment only
 - ▶ Including portlets by portlets is NOT supported !
 - ▶ Example:
`servletRequestDispatcher.include(„/worldclock/StdWorldClock“);`
- Access remote portlets via Remote Request Dispatcher
 - ▶ Normal Servlet Remote Request Dispatcher can detect this URL and render the portlet remotely (works only for JSR 168!)
 - ▶ allows the invocation of portlets outside of the current JVM within a Network Deployment single core group environment

WAS Portlet Support - Access Portlets - Details

- PortletPreferences are stored within cookies
- Using resource Serving to have complete control over the markup (WAS 7.0)
- Aggregate multiple portlets on a page via Aggregation Tag Library
 - ▶ The URL contains only the state of one portlet on the page
 - ▶ The page state (state of all portlets) is managed via session
 - NO bookmarkability or back-button support
 - ▶ Supports Public Render Parameter (WAS 7.0)
 - Share the navigational state between portlets by defining a page scope using a specific JSP tag
 - ▶ NO Eventing support

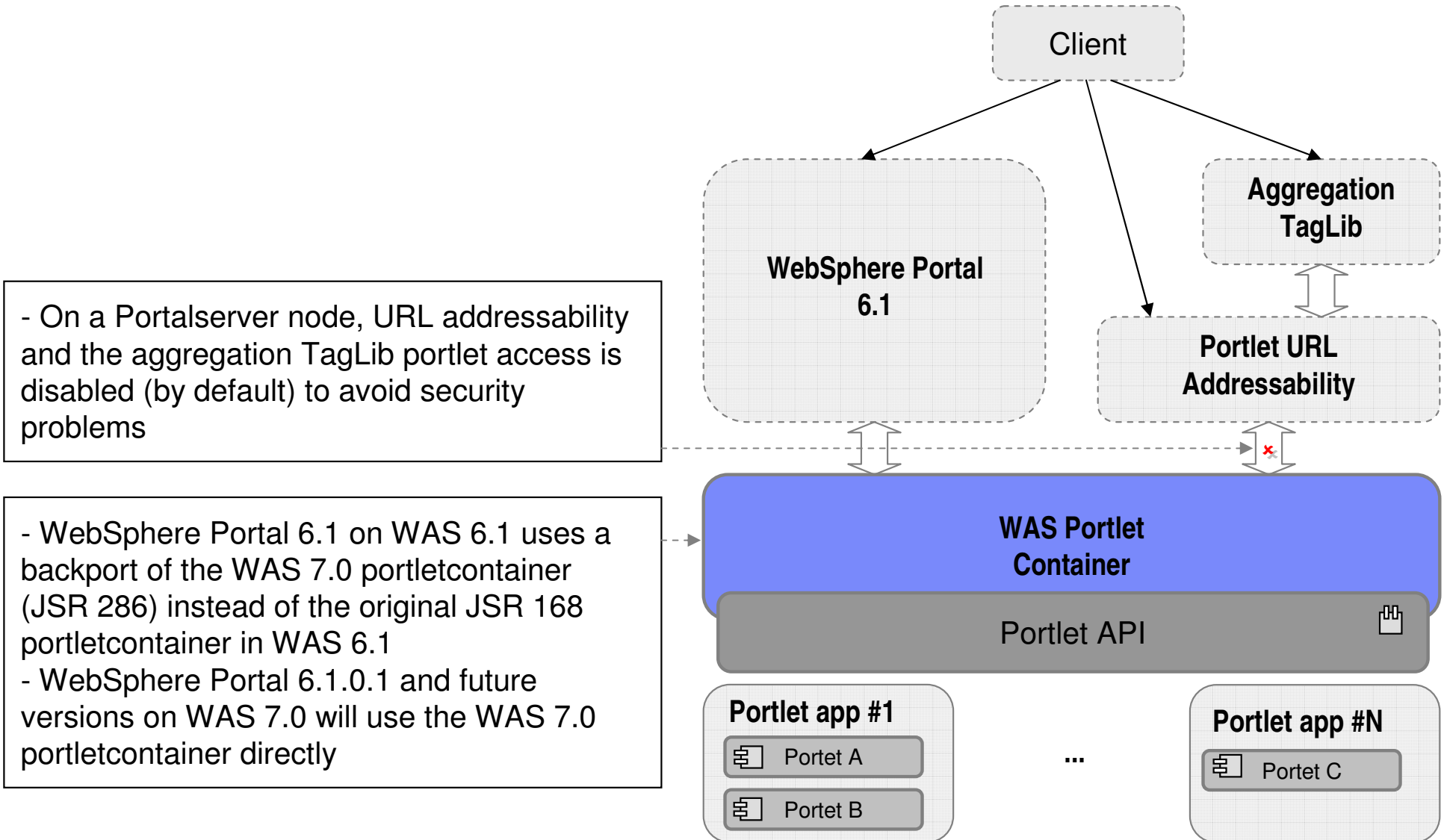
WAS Portlet Support - Demo

- Installation
- URL addressability
- Aggregation
 - ▶ Public Render Parameter
- (PMI)

Agenda

- Portlet Standards growth
- Portlet support in WebSphere Application Server
- Relationship with WebSphere Portal 6.1
 - ▶ Architectural Overview
 - ▶ Features
- WebSphere Application Server WSRP producer
- Migrating to WebSphere Portal
- Outlook
- Summary

Relationship with WebSphere Portal – Architectural Overview



Relationship with WebSphere Portal – Standard Features

Feature	WebSphere Portal 6.1	WAS 6.1	WAS 7.0
JSR 168	✓	✓	✓
JSR 286	✓	✗	(✓)
Eventing (JSR 268)	✓	✗	✗
Public Render Parameter (JSR 286)	✓	✗	✓
Resource Serving (JSR 286)	✓	✗	✓
WSRP 1.0 Producer	✓	✓	✓
WSRP 1.0 Consumer	✓	✗	✗
WSRP 2.0 Producer	✓	✗	(✓)
WSRP 2.0 Consumer	✓	✗	✗

Relationship with WebSphere Portal – Pitfalls

- Portlet URL length management differs
- User management and security aspects
 - ▶ Embed backend-systems via WebSphere Member Manager
- Portlet Preferences and modes
 - ▶ Persistency guaranteed by use of databases instead of Cookie or Session persistence
- Customizations and default values defineable via corresponding modes
- Portlet Fragment Caching
- Portlets on a portlet page must take care about namespacing themselves
- No Access to WebSphere Portal programming model extensions
 - ▶ Property Broker, Credential Vault, Content Access Service ...

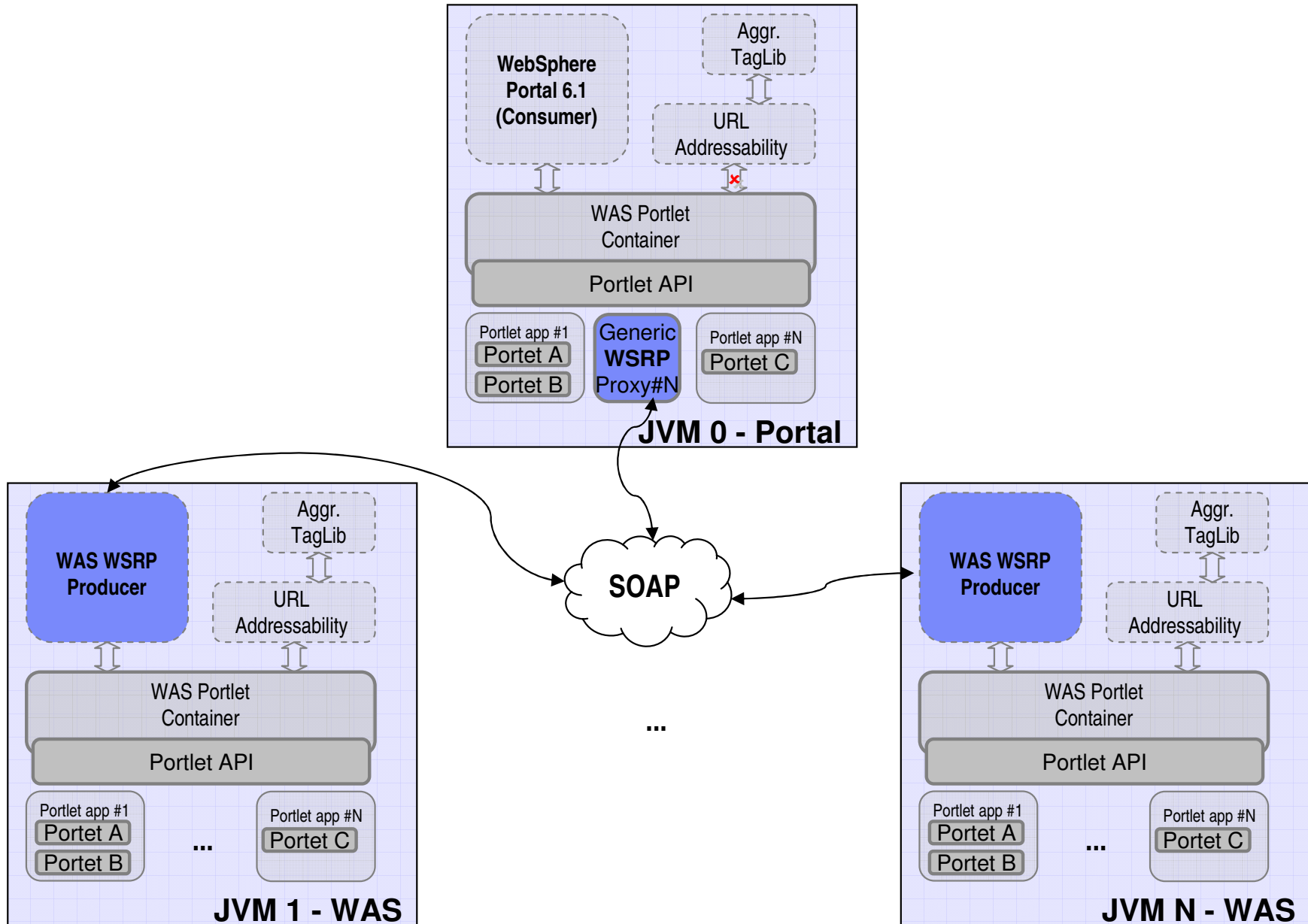
Agenda

- Portlet Standards growth
- Portlet support in WebSphere Application Server
- Relationship with WebSphere Portal 6.1
- WebSphere Application Server WSRP producer
 - ▶ Architectural Overview
 - ▶ Usage Scenarios
 - ▶ Demo
- Migrating to WebSphere Portal
- Outlook
- Summary

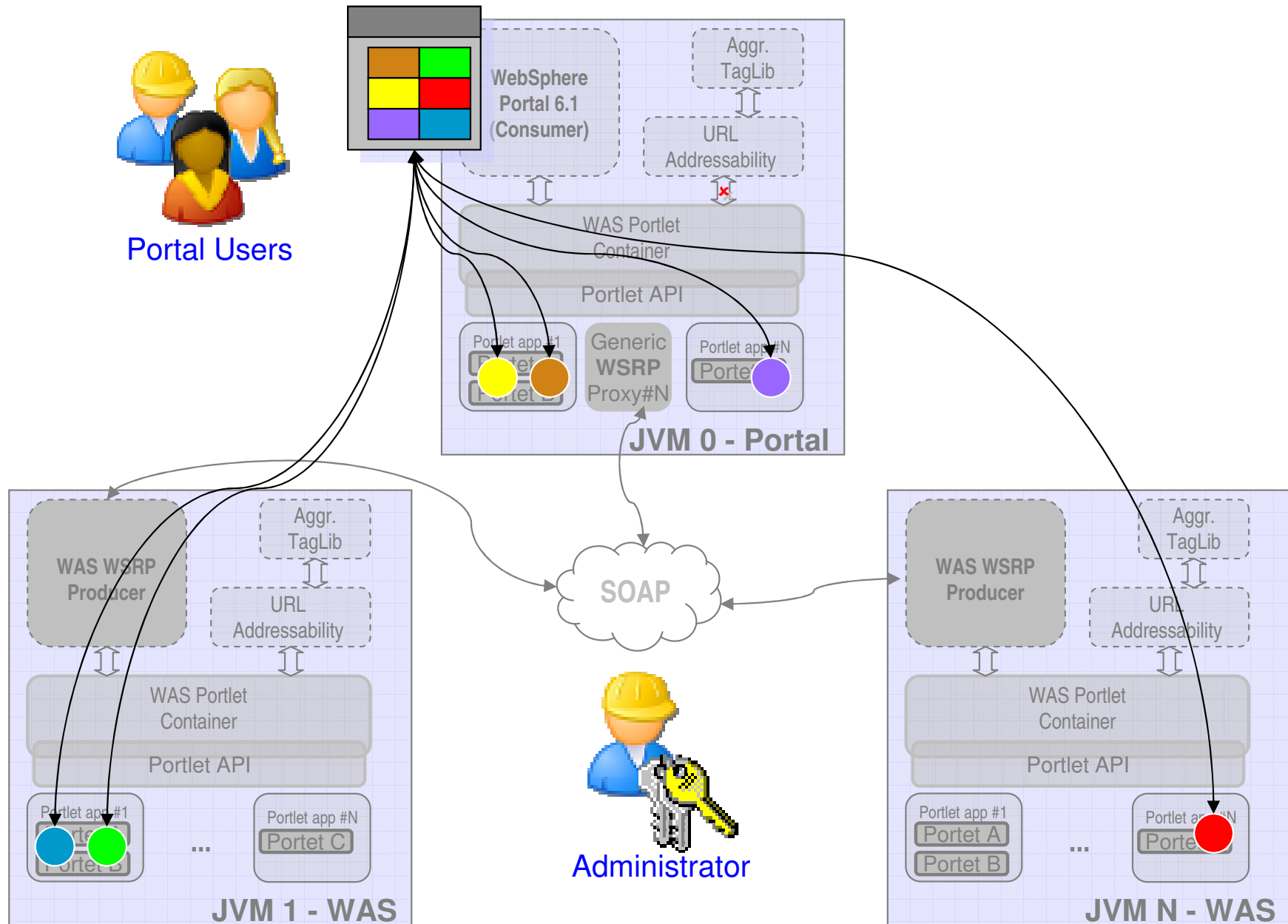
WAS WSRP Producer – What is it?

- Lightweight IBM WebSphere Application Server WSRP 1.0 Producer
 - ▶ Builds on top of WAS Portlet Container
 - Exposes JSR168 portlets as WSRP services
 - Enables all WSRP 1.0 Consumers to interact with WAS portlets
- “Silent” integration
 - ▶ Easily deployable (just drop an EAR file on WAS)
 - ▶ Seamless integration with JSR168 container in WebSphere Application Server
 - ▶ No new admin UI
 - Directly integrated into existing WebSphere Application Server admin console
- Supported on all platforms and available as **free** Catalog download
- Covers the same WSRP 1.0 protocol spectrum as WebSphere Portal 6.1

WAS WSRP Producer – Architectural Overview



WAS WSRP Producer – Architectural Overview



WAS WSRP Producer – Usage Scenarios

- Lightweight means to integrate content into Enterprise Portals
- Provide access to JSR168/JSR 286 (soon) portlets deployed on WebSphere Application Server
- Exploit WebSphere Application Server features from portlets and integrate them to WebSphere Portal
- JVM Isolation & Workload Distribution
 - ▶ Distribute portlets to different JVMs
 - ▶ Protect Enterprise front-end portal from “malicious” portlets tearing down the JVM
 - ▶ Spread out load to Producer servers
 - If many applications need to be served
 - Applications using much memory
 - Applications causing high cpu load on central server

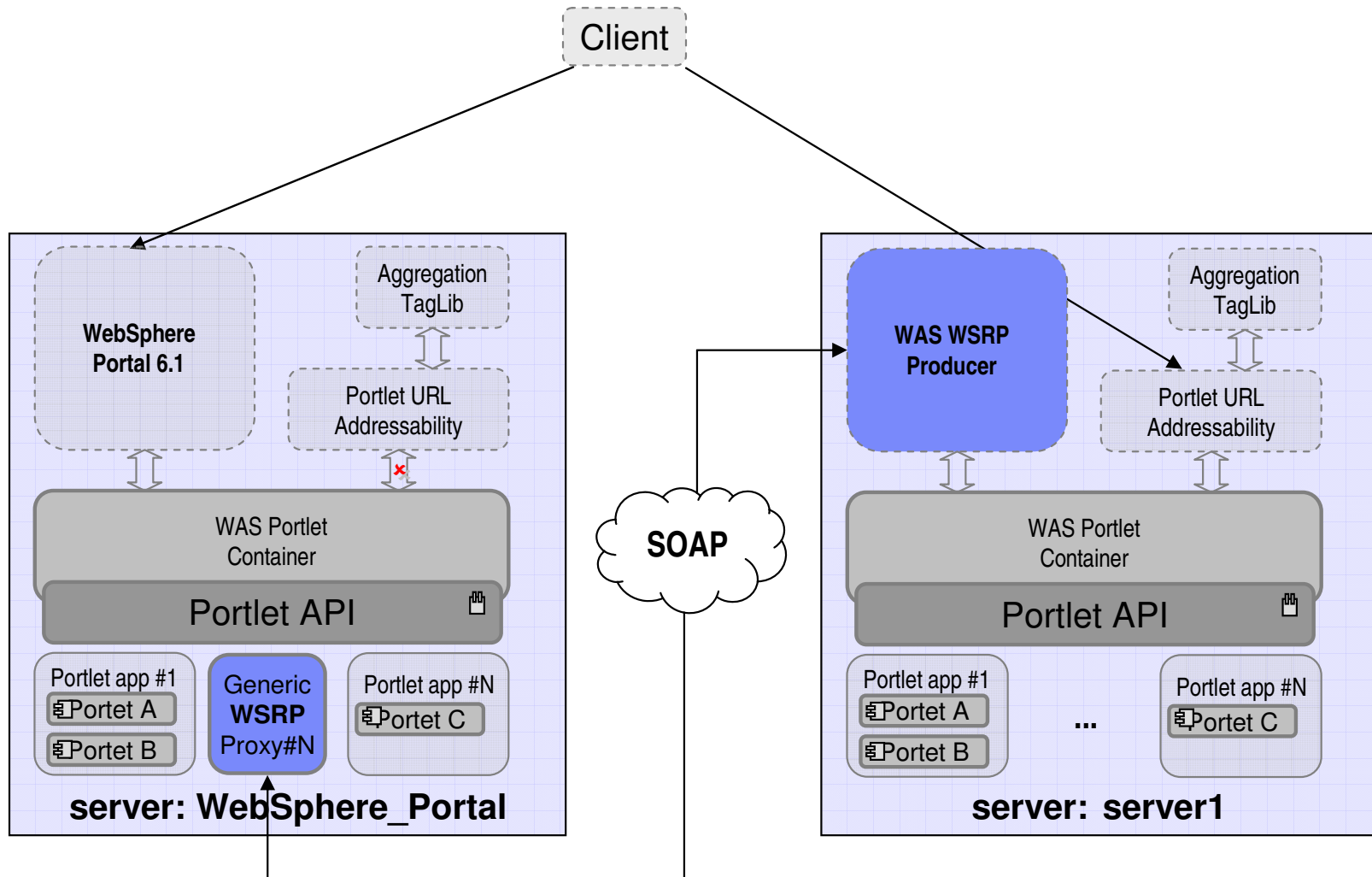
WAS WSRP Producer – Exposing Portlets

- By default all portlets installed into WebSphere Application Server are exposed as WSRP remote portlets
 - ▶ WSRP Consumers can access portlets right out-of-the-box
 - ▶ No additional administrative tasks necessary
- Administrator can control access to these portlets
 - ▶ Portlet exclude list
 - Portlets can be excluded from being provided over WSRP in general
 - Controlled via WAS admin console
 - ▶ Access Control checks
 - Tied to JavaEE security
 - Can be enabled/disabled by admin

WAS WSRP Producer – Security

- Authentication, Integrity, Confidentiality, Non-Repudiation
 - ▶ Realized by WS stack (WS-Security)
 - ▶ And/or transport level (SSL)
- Supports SSO using WS-Security identity assertion
 - ▶ LTPA token forwarding
 - ▶ Signed UsernameToken
 - ▶ UsernameToken for asserted identity + UsernamePasswordToken for trust identity
 - ▶ In general everything WebSphere Application Server Supports with WS-Security
- Access Control handled by WSRP 1.0 Producer & WebSphere Application Server Security
 - ▶ Authorization checks against JavaEE security role specified by the portlet

WebSphere Application Server WSRP Producer - Demo



WAS WSRP Producer – How does it compare to Portal?

	WebSphere Portal 6.1 Producer	WAS WSRP 1.0 Producer
WSRP 1.0 conformance	✓	✓
Bridge JSR 168 portlets to WSRP	✓	✓
Cluster support	✓	✓
Persistent State Management	✓	✗ Pushed to Consumer
Session Management	✓	✓
Remote config mode	✓	✗
Edit defaults support	✓	✗
WSRP caching support	✓	✓
SSO using WS-Security	✓	✓
Authorization	✓	✓
Granular control (by mode)	✓	✗
Personalization (WSRP P3P profiles)	✓	✓
Puma User support	✓	✗
WebSphere Application Server 6.1	✓	✓

Agenda

- Portlet Standards growth
- Portlet support in WebSphere Application Server
- Relationship with WebSphere Portal 6.1
- WebSphere Application Server WSRP producer
- **Migrating to WebSphere Portal**
- Outlook
- Summary

Migrating to WebSphere Portal

- No direct migration from „URL Addressability“ or „aggr. TagLib“
 - ▶ reuse the portlets directly
- Install Portal on top of an existing WAS with portlets
 - ▶ Use the XML Access „predeployed“ app task
 - ▶ Mind the Pitfalls:
 - Caching
 - Security
 - ...
- Integrate portlets running on WAS in a portal node (using the WAS WSRP producer)
 - ▶ no changes in the portlets
 - ▶ Isolation to run your applications on multiple JVMs
 - ▶ Step by step migration

Agenda

- Portlet Standards growth
- Portlet support in WebSphere Application Server
- Relationship with WebSphere Portal 6.1
- WebSphere Application Server WSRP producer
- Migrating to WebSphere Portal
- Outlook
- Summary

Outlook (subject to change)

- Portlet and Portletcontainer management will be more aligned with the standard WAS configuration means as in previous Portal versions
- WebSphere Portal profile and configuration management will be easier
 - ▶ e.g. Use the Dmgr for your cluster deployment directly
- WAS WSRP 2.0 producer
 - ▶ On WAS 7.0
- WAS 7.0 will support all component model features of JSR 286
 - ▶ Portlet filters
 - ▶ Resource Serving
 - ▶ Validation based Caching
 - ▶ Public Render Parameters
 - ▶ ...

Summary

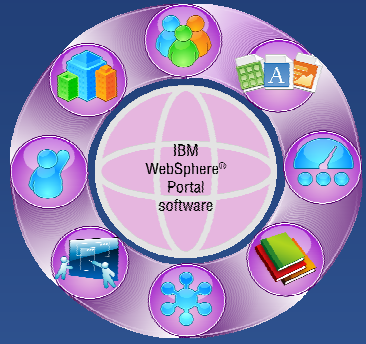
- The major theme of the second version of the Portlet Standards is Coordination and building composite applications based on portlet components
- Use the WAS portlet support for easy developing and testing your applications
 - ▶ WAS 6.1 supports JSR 168
 - ▶ WAS 7.0 supports JSR 286 (partially)
- WebSphere Portal 6.1 builds upon the WAS internal portlet support instead of shipping its own container
 - ▶ Enables advanced Portal features as aggregation, personalization...
- Use the WAS WSRP producer for Isolation and Migration scenarios
 - ▶ WAS WSRP 1.0 producer available (can be used on WAS 6.1 and WAS 7.0)
 - ▶ WAS WSRP 2.0 producer will be available soon (WAS 7.0)

Additional Information and Resources

- „Exploiting the portlet runtime in WebSphere Application Server 6.1“ developer works article series:
 - ▶ Part 1: Introduction
http://www.ibm.com/developerworks/websphere/library/techarticles/0607_hesmer/0607_hesmer.html
 - ▶ Part 2: Extended Capabilities
http://www.ibm.com/developerworks/websphere/library/techarticles/0607_hesmer/0607_hesmer2.html
 - ▶ Part 3: Performance Measurement
http://www.ibm.com/developerworks/websphere/library/techarticles/0608_hesmer/0608_hesmer.html
 - ▶ Part 4: Migrating to WebSphere Portal
http://www.ibm.com/developerworks/websphere/library/techarticles/0610_hesmer/0610_hesmer.html

Additional Information and Resources

- “Leveraging J2EE roles in JSR 168 portlets running in WebSphere Portal”
 - ▶ http://www.ibm.com/developerworks/websphere/library/techarticles/0703_hesmer/0703_hesmer.html
- Q&A on developing portlets in WAS 6.1
 - ▶ http://www.ibm.com/developerworks/websphere/library/techarticles/0705_rick/0705_rick.html
- WAS WSRP producer download
 - ▶ <http://www-01.ibm.com/software/brandcatalog/portal/portal/details?NavCode=1WP1001BA>
- Register predeployed portlet applications in Portal
 - ▶ <http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1m0/index.jsp?topic=/com.ibm.wp.ent.doc/admin/j2ee.html>



WebSphere User Group UK Meeting, Edinburgh, 2008

Thanks for your attention!
Questions?

Andreas Brunnert

Software Engineer

*WebSphere Portal
Development*



*IBM Deutschland
Research & Development
GmbH*

*Schoenaicher Strasse 220
71032 Boeblingen*

*Phone: +49-7031-16-2159
brunnert@de.ibm.com*

© IBM Corporation 2008 All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

IBM, the IBM logo, WebSphere, Lotus, Lotus Notes, Domino, Quickplace, Sametime, Workplace and Quickr are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

All references to Renovations Inc. refer to a fictitious company and are used for illustration purposes only.

