



DINI Workshop Personalisierte Webportale 


Standards für Portalkomponenten

Java Portlet API und WSRP

Oliver Köth
IBM Deutschland Entwicklung GmbH




© 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

Agenda


- Portals, portlets and standards
- Content of existing portlet standards
- Evolution of portlet standards
- New web technology – portals and AJAX
- Outlook and wrap-up




2 | © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Portal, portlets and standards




© 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

Some definitions in the context of the standards

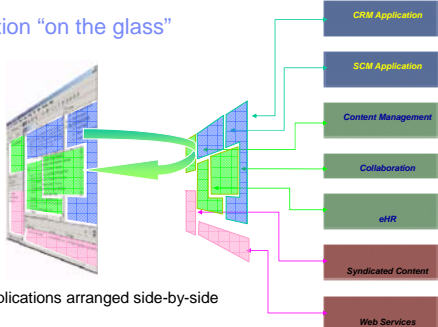
- Portal: “**Web-based application that aggregates content from different sources**”
- Portlet: “**An application that provides a specific piece of content (information or service) inside a portal**”
 - Portlets are not just content but interactive web applications
 - The part of a portal framework that manages and interacts with portlets is referred to as “portlet container”




4 | © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 


Integration “on the glass”



Multiple applications arranged side-by-side




5 | © 2007 IBM Corporation

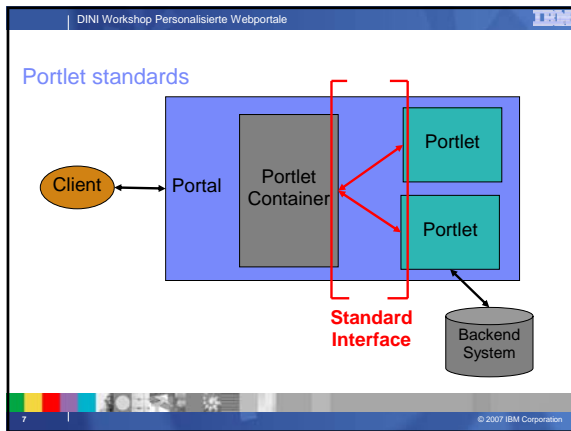
DINI Workshop Personalisierte Webportale 

Why portlet standards?

- Decouple portal framework from portlets
- Allow mix and match between portlets from different sources
- Content providers can easily make information available on different portal platforms
- Customers developing significant numbers of portlets are not locked onto a particular platform



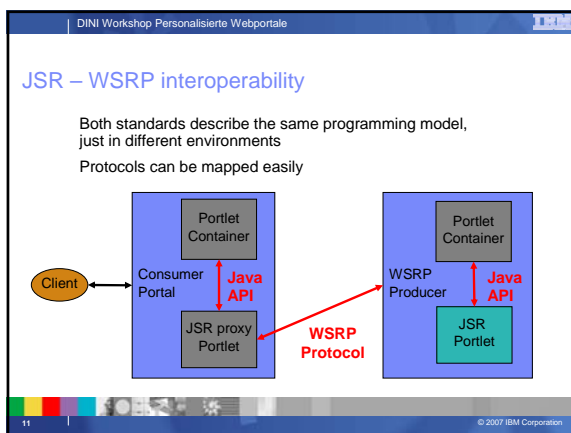
6 | © 2007 IBM Corporation



- DINI Workshop Personalisierte Webportale
- ### Two standard interfaces for portlets
- **“Java Portlet Specification”**
 - Standard for Java portlets that are running “inside” the portal framework
 - Based on Java EE standards family (web applications)
 - defined in the Java Community Process (JCP)
 - **“Web services for remote portlets” (WSRP)**
 - Standard for “remote” portlets to which the portal framework connects over the network
 - Based on web services and SOAP standards family
 - defined by the OASIS consortium
- © 2007 IBM Corporation

- DINI Workshop Personalisierte Webportale
- ### Java Portlet Specification
- **Portlet standard for local Java portlets**
 - First version defined as JSR 168, finished Oct. 2003
 - Co-chaired by IBM and Sun
 - <http://jcp.org/en/jsr/detail?id=286>
 - **Reference implementation available at Apache, donated by IBM**
 - <http://portal.apache.org/pluto>
 - **Compliance test suite available from Sun**
 - **Wide market adoption**
 - supported by many commercial and open source portals
 - BEA, IBM, Oracle, Sun, Tibco, Vignette, ...
 - Apache, eXo, JBoss, Liferay, uPortal, ...
- © 2007 IBM Corporation

- DINI Workshop Personalisierte Webportale
- ### Web services for remote portlets
- **Defined at OASIS**
 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
 - **Chaired by IBM**
 - **Sample implementation available at Apache, provided by IBM**
 - <http://ws.apache.org/wsrp4j>
 - **Conformance test suite available at sourceforge**
 - <http://sourceforge.net/projects/wsrptk>
 - **Wide market adoption**
 - supported by many commercial and open source products
 - Apache, BEA, IBM, Netunity, Oracle, Sun, SAP, Tibco, Vignette, ...
- © 2007 IBM Corporation



DINI Workshop Personalisierte Webportale

Content of existing portlet standards

© 2007 IBM Corporation

DINI Workshop Personalisierte Webportale IBM

What do the standards define?

- **Contract between the portlet and the portlet container**
 - specifies invocation of portlets and services for portlets
 - for local portlets: Java API
 - for remote portlets: WSRP markup interface
- **Integration of new portlets into a portal**
 - for local portlets: packaging format for portlet applications
 - for remote portlets: WSRP service description interface

13 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale IBM

Standards content

14 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale IBM

Outside the scope of the standards

- **All features of the portal front-end**
 - aggregation and layout management
 - navigation between pages
 - personalization and customization
 - administration, access control
 - ...
- **There can be huge functional differences between two standards-compliant portals**

15 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale IBM

Portlet interaction model

- **Basic interaction follows the web request/ response pattern**
 - portlets need to live together on a common page as “well-behaved citizens”
 - this requires a refined interaction model
- **Unlike stand-alone web apps, the interaction model distinguishes two types of requests**
 - **action request:** user interaction with a portlet, changes internal state – no output
 - **render request:** no state change – just generate markup for the current state

16 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale IBM

Request flow in the Java Portlet API


17 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale IBM


Some features of the programming model

- **Persistent configuration (portlet preferences)**
- **User information**
- **Portal support for different portlet modes**
 - e.g. configuration screens with access protection
- **Caching of portlet markup**
- **Portal-managed navigational state (render parameters)**
 - state is tracked in the URL → on the client, no server sessions
 - support expected web browser behavior with interactive applications (back button, bookmarking)


18 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Evolution of portlet standards




© 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

Evolution of portlet standards

- **Both existing standards aimed at a "60%" target**
 - support most common use cases
 - agree on basic functionality quickly
 - defer advanced functionality to later standard versions
- **Java Portlet Specification 2.0 (defined by JSR 286)**
 - <http://jcp.org/en/jsr/detail?id=286>
- **WSRP 2.0**
 - <http://docs.oasis-open.org/wsrp/v2/wsrp-2.0-spec.html>
- **Remainder of this presentation will refer to "V1.0" and "V2.0" standards**




© 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

JSR 286 details

- **Chaired by IBM**
- **All major portal players in the EG**
 - Apache, BEA, eXo Platform, JBoss, Novell, Oracle, SAP, Sun, TIBCO, Vignette,
- **Reference implementation will be provided at Apache**
- **Test Compliance Kit will be available for free**
 - both developed at Jena University in cooperation with IBM




© 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

JSR 286 continued

- **Schedule expectations**
 - First draft with WSRP 2.0 features: July 06
 - Public final draft: May 07
 - Final version: Aug 07
- **Continuously updated public drafts**
 - <http://jpc658.inf.swt.uni-jena.de/spec/>
- **Outlook: include Java portlet API in Java EE**
 - over time, supersede servlet API as "normal" programming model for web components




© 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

WSRP 2.0 details

- **Chaired by IBM**
- **Members of the TC**
 - BEA, R. Brooks, CA, Microsoft, NetUnity Software, Novell, Oracle, Tibco, SAP, Sun, Vignette, ...
- **Schedule**
 - Public draft: June 06
 - Final draft: Dec 06
 - Approved: Feb 07
- **Alignment with JSR 286**
 - Considering updates to include missing JSR 286 features



© 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Portlet coordination

- **Top complaint about the V1.0 standards**
 - V1.0 only allows information exchange between portlets packaged in one application (via session sharing)
- **V2.0 standards add two more mechanisms for information exchange**
 - Shared navigational state – portal managed URL information that is accessible to multiple portlets
 - Event-based communication – send and receive events via a publish-subscribe model



© 2007 IBM Corporation

DINI Workshop Personalisierte Webportale

Shared navigational state in portal URLs

<http://portal.com/...docid=document2...>

 <http://portal.com/...docid=document3...>

Portal Page

Navigator Portlet

- document1
- **document2**
- document3

Content portlet

content of document2

link to select document3

Portal Page

Navigator Portlet

- document1
- document2
- **document3**

Content portlet

content of document3

25 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale

Shared navigational state

- **No new programming concept for JSR portlets**
 - Reading and writing render parameters is already part of JSR168 – simply share this information between portlets
- **But a simple and powerful concept**
 - Typical use cases for coordinating portlet view state can be solved with this model
 - e.g. navigator and content display portlet share the same parameter "content ID" → automatic synchronization
- **Portlet coordination is limited to synchronizing simple view state information**
 - Portlets that want to transfer complex data or react to messages with code need a more powerful concept: **events**

26 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale

Events in JSR 286 and WSRP 2.0

- **Again, standards describe the protocol between portal and portlets**
 - How local Java portlets publish and receive events
 - How events are transported between the portal and remote portlets
 - Allows complex types for event payload with XML mapping
- **Event distribution is not part of the standard**
 - portlets never call each other, control always flows through the portal
 - different portals may use different ways to match published events to potential receivers
 - match by event name, explicitly defined wiring, add type transformations etc.

27 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale

Request flow with events

User

Portal

Portlet container

Portlets A B C

Action on B

Do action on B

processAction

setEvent(X)

processEvent(X)

Render A, B, C

render

Change state

Change state

28 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale

Side note: "ad-hoc" applications

- **Portlet standards allow web components from any source to co-exist on a page**
 - allows portal frameworks that can assemble components into pages using customization tools
- **Combine this with loosely coupled coordination mechanisms**
 - allow to interconnect portlets that were not originally intended to work together
- **You get the ability to assemble components into simple "ad-hoc" applications on the fly**
 - This is exactly what web 2.0 terminology calls a "Mashup"


29 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale

Mode programming model additions


- **Support better integration with existing web frameworks**
 - filters, request dispatching
- **Caching extensions**
 - more flexibility through validation-based caching
- **Resource serving**
 - serve stand-alone content from a portlet
 - e.g. for binary formats (PDF, SVG) or pop-up windows
 - can be used for AJAX support
- **Improvements for a lot of minor inconveniences**

30 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

New web technology – portals and AJAX


© 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

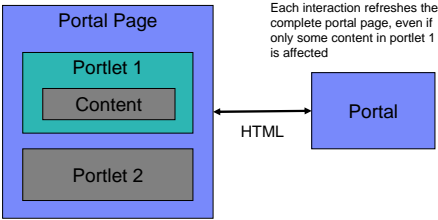
What is AJAX?

- Stands for “Asynchronous JavaScript And XML”?
- Dynamic web-pages that can refresh parts of their content dynamically
 - interacting with a web application does not require reload of the entire page
 - instead, clicking a link will only refresh the part of the UI that is affected
 - behind the scenes, a number of DHTML techniques can be at work
 - not always asynchronous, not always XML
- What is the advantage of AJAX?
 - web interfaces become more responsive and powerful
 - another swing in the client/server paradigm shifts

32 © 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

Without AJAX – classic portal

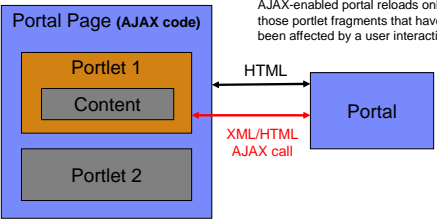


Each interaction refreshes the complete portal page, even if only some content in portlet 1 is affected

33 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 


AJAX flavor 1 – client side aggregation



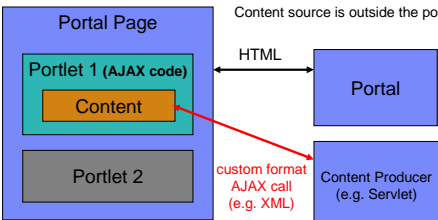
AJAX-enabled portal reloads only those portlet fragments that have been affected by a user interaction

Implemented entirely by the portal. No special portlet programming needed – but possible trouble with JavaScript in portlets

34 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 


AJAX flavor 2 – today



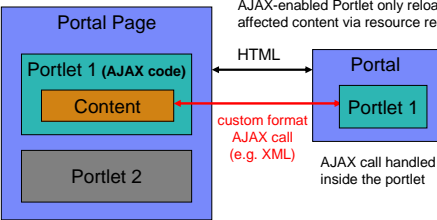
AJAX-enabled Portlet only reloads affected content
Content source is outside the portal

Can be implemented with V1.0 functionality. AJAX call cannot use portlet APIs or affect portlet (except via servlet session sharing)

35 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 


AJAX flavor 3 – resource request



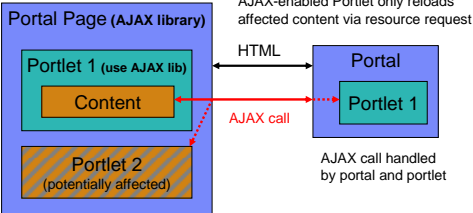
AJAX-enabled Portlet only reloads affected content via resource request

Supported with V2.0. AJAX call can use a limited set of portlet APIs, because the portal is not involved


36 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

AJAX flavor 4 – under discussion...



Planned for V2.0. AJAX call should be able to use the full set of portlet APIs including coordination features


37  © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Outlook





© 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Portals as integrated applications


- **Portal customers often want more than just content pieces "side-by-side"**
 - Portal should behave like a single integrated application
- **Portlets should influence portal environment and each other**
 - Link from portlet to other pages
 - Link from portal "frame" or external applications into portlets (e.g. search box)
 - Launch dynamic pages from portlets, dynamically add/remove portlets (e.g. task processing scenario)
 - Add portlet-specific items to the portal navigation
 - ...


39  © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Portals as integrated applications


- **Standards do not (yet?) cover many integration scenarios**
 - Coordination features are a first step
- **Creating standards for "close integration" use cases is difficult**
 - Portal vendors have different concepts and features
 - Need to create standards for portlet interfaces without restricting implementation range for portal frameworks
- **The V2.0 standards are not the end...**


40  © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Summary


- **Standards for portal components: Java Portlet API and WSRP**
- **Standards allow to write portable componentized webapps**
 - first generation addresses many common use cases
 - offers some extra functionality beyond stand-alone webapps
 - targeted at "on-the glass" integration of individual applications
- **Standards are evolving**
 - next generation will be completed soon
 - will offer significant functional enhancements
- **Portals/Portlets can support AJAX technologies in a number of ways**

41  © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Thank you!

- **Questions?**
- <mailto:okoeth@de.ibm.com>

42  © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Backup




© 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Portlet standards and SOA – the wider picture




44 © 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

Portlet standards and SOA

- **Service Oriented Architectures**
 - SOAs try to decompose enterprise IT infrastructures into decoupled services
 - part of this is to separate business processes, business services and presentation logic
 - business processes need to remain flexible
 - use standards-based interfaces between components
- **Portlets are a good match for the presentation layer in a SOA**
 - small units for data input and presentation that can be assembled flexibly into a portal framework
 - business processes can drive the presentation layer “behind the scenes” rather than being hardcoded in the UI




45 © 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

Features of the V1.0 programming model

- **Persistent configuration (portlet preferences)**
 - without requiring to code a DB layer
- **User information**
 - simple standardized access to user repository information
 - to be replaced by a more complete Java EE standard in the future
- **Caching of portlet markup**
 - expiration based caching in the portal framework




46 © 2007 IBM Corporation


DINI Workshop Personalisierte Webportale 

Features of the V1.0 programming model

- **Portal-managed navigational state (render parameters)**
 - state tracking in the URL → on the client
 - without requiring a server-side session
 - support expected web browser behavior with applications (back button, bookmarking)
- **Portlet modes**
 - allow the portal to react on different states of the portlet
 - e.g. protect configuration view by portal access control




47 © 2007 IBM Corporation

DINI Workshop Personalisierte Webportale 

Java portlet features inherited from Java EE

- **Portlets are web components and have many common characteristics with servlets**
- **Session management**
 - separate scopes for each portlet occurrence on a page
- **Use of Java Server Pages for markup generation**
- **Bridges to many common servlet MVC frameworks**
 - JSF, Struts, Spring, ...



48 © 2007 IBM Corporation

Enhanced web framework support

- **Better integrate with existing web frameworks**
 - Make transition from servlet to portlet application as easy as possible
 - Based on feedback from lead developers of frameworks like JSF, Struts, Spring, WebWork
- **Following features were added**
 - Call into servlet artifacts (request dispatch) from all portlet request phases – JSR considered this only for rendering
 - Introduce filters for all phases to handle bridging logic

Validation-based caching

- **V1.0 supported only expiration-based caching**
 - minimal calling overhead
 - only usable if the content validity period can be determined in advance
- **V2.0 adds validation-based caching**
 - now portlets can specify that they want to be queried if the content is still valid
 - more flexible, at the cost of more calling overhead