Portals, Portlets and Web Services

Thomas Schaeck
Architect, WebSphere Portal Server
IBM
Presentation Goal

Learn about portal architecture, portlets, their relation to Web services, and emerging standards.
Learning Objectives

• As a result of this presentation, you will:
  – Gain insight in Portal Architecture
  – Understand Portlets and Portlet API
  – Know relation of Portlets and Web Services
  – Understand Web Services for Remote Portals
  – Know about emerging standards
Speaker’s Qualifications

• Initiated standards for
  – Java™ Portlet API
  – Web Services for Remote Portals (WSRP)

• Architect at the WebSphere Portal Server Development Team in Boeblingen, Germany

• Leads Portal standards activities at IBM

• 5 years experience with Java technology
Portals May Become the Center of the (IT) Universe …

… but only standards for dynamic integration of portlets and services will let portals reach their true potential:

Be a platform for quick integration, aggregation, and delivery of distributed applications and information through different channels and devices
Presentation Agenda

- Portal Architecture Introduction
- Portlets and Portlet API
- Web Services
- Portlets and Web Services
- Web Services for Remote Portals
- Demo
What Are Portals?

- Common access point to distributed information and applications

- Typical functions:
  - User registration
  - Authentication and Authorization
  - Pluggable portal components: Portlets
  - Personalization based on profiles/behavior
  - Customization of pages by users
  - Search
  - Content Management
Example of a Portal View
Example of a Portal Topology
Major Functional Components

User Registration/ Selfcare

Customization

Aggregation

Authentication

Authorization

User Information

Portlet Container

Portlet Proxy

Local Portlet

User’s Selections

Portlet Registry

Portlet Settings

Portlet Inst. Data

SOAP

Remote Portlet

Web Services
Typical Entities and Relations in a Portal

- User
  - Group
  - Page
  - Portlet Instance
  - Portlet
  - Defined by User
  - Defined by Administrator

12
Portlets and Portlet API
Portlets

- Components to be aggregated in portals
- Specialized servlets aware of portal context
  - User profile information
  - Per portlet-instance data stored by portal
  - Per portlet settings managed by portal
  - Portlet Window state (NORMAL, MIN, MAX)
  - Portlet Modes (VIEW, EDIT, CONF, HELP)
  - Portlet Events (action / message events)
Examples of Portlets

[Image of a webpage showing a news article about learning services and a search bar with options for BluePages, w3, and ibm.com.]
Invocation of Portlets

- Servlet API
- Portal Servlet
- Portlet API (Invocation)
- Portlet Container
- Local Portlet
- Local Portlet
- Local Portlet
- Portlet API (context)
- Portal Infrastructure, User Registry
  Persistent and Transient Portal State
- J2EE APIs
  - Connectors
  - Web Services
  - EJBs

- EJBs
Invocation of Portlets

Portlet API (Invocation)

Portal Servlet

Portlet Container

Local Portlet

Local Portlet

Local Portlet

Portlet API (context)

Portal Infrastructure, User Registry Persistent and Transient Portal State

Connectors

Web Services

EJBs

Servlet API

J2EE APIs
Invocation of Portlets

Portlet Container

Local Portlet

Local Portlet

Local Portlet

Portal Infrastructure, User Registry
Persistent and Transient Portal State

Servlet API

Portlet API (Invocation)

Portlet API (context)

Connectors

Web Services

EJBs

Portal Servlet

J2EE APIs
Invocation of Portlets

Portal Infrastructure, User Registry
Persistent and Transient Portal State

Portal Container

Portlet API (Invocation)

Local Portlet

Local Portlet

Local Portlet

Portlet API (context)

J2EE Pls

Connectors

Web Services

EJBs

Servlet API

Portal Servlet

Servlet API (Invocation)
Invocation of Portlets

Servlet API

Portal Servlet

Portlet API (Invocation)

Portal Container

Local Portlet

Local Portlet

Local Portlet

Portlet API (context)

Portal Infrastructure, User Registry
Persistent and Transient Portal State

Connectors

Web Services

EJBs
Invocation of Portlets
Portlet Action Handling

- Portlet Container
  - Action
  - Local Portlet
  - Local Portlet
  - Local Portlet

- Portlet API (Invocation)
- Portlet API (context)
- Portal Infrastructure, User Registry
- Persistent and Transient Portal State
- Servlet API
- J2EE APIs
- Web Services
- EJBs

- Connectors

- Portal Servlet
- Servlet API

Diagram highlights:
- Portlet Container
- Action
- Local Portlet
- Local Portlet
- Local Portlet
Portlet Action Handling

- Portal Infrastructure, User Registry
- Persistent and Transient Portal State
- Connectors
- J2EE APIs
- EJBs
- Web Services
- Local Portlet
- Local Portlet
- Local Portlet
- Action
- Portlet Container
- Portlet API (context)
- Portlet API (Invocation)
- Portal Servlet
- Servlet API
Portlet Action Handling

- Portlet Container
  - Action
  - Local Portlet
  - Local Portlet
  - Local Portlet

- Portlet API (context)

- Portal Infrastructure, User Registry
  - Persistent and Transient Portal State

- Servlet API
  - Portal Servlet

- J2EE APIs
  - Connectors
    - Web Services
    - EJBs

- Portlet Action Handling

- Portal API (Invocation)
Portlet Action Handling

- Servlet API
- Portal Servlet
- Portlet API (context)
- Local Portlet
- Action
- Local Portlet
- Connectors
  - Web Services
  - EJBs
- Portal Infrastructure, User Registry
  - Persistent and Transient Portal State
Portlet Action Handling

Portlet Container
- Action
- Local Portlet
- Local Portlet
- Local Portlet

Portlet API (context)

Portal Infrastructure, User Registry
Persistent and Transient Portal State

Servlet API
- Portal Servlet

J2EE APIs
- Connectors
- Web Services
- EJBs

Portlet API (Invocation)
Portlet Action Handling

Portlet Container
- Action
- Local Portlet
- Local Portlet
- Local Portlet

Portlet API (context)

Portal Infrastructure, User Registry
Persistent and Transient Portal State

Portal Servlet

Servlet API

J2EE APIs
- Connectors
- Web Services
- EJBs
Portlet API

- API defining interaction between portals and portlets
- Defines interfaces for interoperability between portals and portlets
- Based on the Java™ Servlet API
- Provides additional abstractions for portal context and functions
- Standardization initiated in the Java Community Process℠ program (JSR 168)
Core Classes and Interfaces: Example From WPS Portlet API

- Portlet base class
- PortletRequest/Response, PortletSession
- PortletData, PortletSettings
- Events: ActionEvent, MessageEvent
- Listeners: ActionListener, MessageListener
- User
Example of a Stock Quote Portlet

- Stock prices for user-selected list of stock symbols:
  - VIEW mode shows stock prices: doView
  - EDIT mode lets user change stocks: doEdit
  - HELP mode explains the portlet: doHelp
  - CONFIG mode lets administrator select stock
  - Quote source to use: doConfig
A Portlet Class: Example Using WPS Portlet API

- Portlets are similar to Servlets

```java
public class StocksPortlet extends Portlet {
    public void init(PortletConfig config) {
        ... initialize portlet ...
    }

    public void doView(PortletRequest req, PortletResponse rsp)
    throws PortletException, IOException {
        ... generate the portlet view ...
    }

    public void destroy(PortletConfig config) {
        ... destroy portlet ...
    }
}
```
Portlet View Mode: Example Using WPS Portlet API

- Use of PortletData, Beans and JSP™ components

```java
public void doView(PortletRequest req,
        PortletResponse rsp)
throws PortletException, IOException {
    // Get stock symbols from portlet instance data
    PortletData data = request.getData();

    String symb = (String) data.getAttribute("symbols");
    .. get prices for symbols from stocks service ..

    StockBean stockBean = new StockBean();
    .. put stock symbol/price pairs in stock bean ..
    req.setAttribute("stockBean", stockBean);

    getPortletConfig().getContext().include(
        "/WEB-INF/ViewStockQuotes.jsp", req, rsp);
}
```
Portlet Edit Mode:
Example Using WPS Portlet API

- Tie Actions to PortletURLs to process Forms

```java
public void doEdit(PortletRequest req,
                   PortletResponse rsp)
throws PortletException, IOException {
    // Create URI pointing to this portlet instance
    // and attach a portlet action
    PortletURI saveUri = rsp.createURI();
    PortletAction saveAction = new Action(SAVE);
    saveUri.addAction(saveAction);

    EditListBean editListBean = new EditListBean();
    // .. put saveURI and other data into the bean ..
    req.setAttribute("editListBean", editListBean);

    getPortletConfig().getContext().include(
        "/WEB-INF/EditSymbolListForm.jsp", req, rsp);
}
```
Web Services
Web Services

- Web services are platform and language independent
- Description of Web services in WSDL (Web Services Description Language)
- Invocation of Web services via SOAP (Simple Object Access Protocol)
- Publish and Find through UDDI (Universal Description, Discovery and Integration)
Service Oriented Architecture

Service Requestor

Service Registry

Service Provider
Service Oriented Architecture

Service Registry

Publish

Service Requestor

Service Provider
Service Oriented Architecture

Service Registry

Find

Service Requestor

Publish

Service Provider
Service Oriented Architecture
Using “Traditional” Web Services

- Data-oriented with service type-specific interfaces
  - Different WSDL definitions, different methods
  - Service-specific presentation must be provided
  - Programming effort for integration of each service
Visual, user-facing, Web services, common interface
- Same WSDL definition
- No service specific-presentation on portal side
- Pluggable, **NO programming effort**
- Once created many channels can be reached quickly
Portlets and Web Services
Portlets Invoking Web Services

- Portlets can invoke any Web service via a service interface-specific SOAP Proxy
- Tools can be used to automatically generate SOAP Proxies for the Java platform from Web service’s WSDL description
Example: Invoking Web Services from a Portlet

- Use of SOAP Proxy to invoke a Web service

```java
public void doView(PortletRequest req,
                   PortletResponse rsp)
    throws PortletException, IOException {
    // Get stock symbols from portlet instance data
    PortletData data = request.getData();

    String symbols = (String) data.getAttribute("symbols");

    // Get prices for symbols from stocks proxy
    float[] stockPrices =
        stocksProxy.getCurrentPrices(symbols);

    ... display result ...
}
```
Aggregation of Visual, User-Facing Web Services in Portals

- Common interface enables plug-n-play
- Use of generic Portlet Proxies as adapters
- No need for specific presentation in the portal
Web Services for Remote Portals
Web Services for Remote Portals (WSRP)

• Visual, user-facing Web services for integration and aggregation in portals
• WSRP interfaces include
  – Life-cycle methods: create/delete instances, sessions, bindings …
  – Methods for getting markup and processing actions
• WSRP technical contracts define
  – Action handling and embedding in URLs
  – Restrictions on markup produced
  – Allowed order of method invocation
• For Publish/Find WSRP defines
  – Information about WSRP services to be published
  – How to find WSRP services
Use of WSRP Services in Portals

- Portals can aggregate many visual WSRP services
- WSRP services optionally are aware of portal context
  - Remote user profile
  - Desired language and markup-type
  - User’s device type
Related Standards

- WSRP is based on existing standards
  - UDDI—Publish, Find, Bind
  - WSDL—Interface Description
  - SOAP—Invocation

- Coordinated with WSIA (Web Services for Interactive Applications)
Portal-WSRP Service Interaction

**User**
- Adds Portlet
  - Portal
    - createPortletInstance
    - Allocate New Instance
  - Web Service for Remote Portals
    - Generate Markup, (opt: create Session)

**User**
- Views Portlet
  - Portal
    - getMarkup
    - I S A
  - Web Service for Remote Portals
    - Generate Markup, (opt: create Session)

**User**
- Clicks Action
  - Portal
    - performAction
    - I S A
  - Web Service for Remote Portals
    - Action Handling, (opt: create Session)

**User**
- Removes Portlet
  - Portal
    - destroyInstance
    - Destroy Instance
Example: Portal Sharing Portlets

- Administrator publishes portlet as WSRP service to UDDI using portal’s admin user interface
- Administrator of other portal finds WSRP service using portal’s UDDI browser and binds to it with a few mouse-clicks
- Users of the second portal can select remote portlets like any local portlet and put them on their pages
Example: Portal Sharing Portlets

Portal 2
- Portal Administration
- Portlet Registry
- Portlet Aggregation
- Portlet Proxy Entry

Portal 1
- Portal Administration
- Portlet Registry
- Portlet Aggregation
- Portlet Entry

UDDI
- Portlet Entry
- Portlet Entry
- Portlet Entry

Find and Bind
- Portal 2
- Portal 1

Publish
- Portal 2
- Portal 1

Invoke (SOAP)
- Portal Proxy
- Remote Portlet

JavaOne™
Distributed Portal Systems

- Bank's Portal
- Stocks Web Service
- Account Portal
- Stocks Portlet
- Account Web Service
- Portlet Proxy
- Corporate Employee Portal
- Purchase Portlet
- Supplier Portal
- Portlet Proxy
- Corp. News Portlet
- Portlet Proxy
- Content Provider
- News Portlet
- Corporate News Web Service

Diagram showing relationships between different services and portlets.
Summary

- A standardized portlet model is key to interoperability between portal servers and portlets
- Portal servers should support both local and remote portlets
- Emerging standards:
  - Java™ Portlet API (JCP, JSR 168)
  - Web Services for Remote Portals (OASIS)
- Portlets can be published as WSRP services
- WSRP services can be wrapped in Portlets
Portals will have to dynamically integrate and aggregate local portlets as well as Web services across the Internet.

This can be achieved through the Java™ Portlet API combined with Web Services for Remote Portals.
Q&A