JavaOne℠

Sun's 2004 Worldwide Java Developer Conference℠

Building a Portal and the Benefits of the Portlet Specification

A Case Study

java.sun.com/javaone/sf

Peter Moran

Senior Software Engineer

www.objectconsulting.com.au

Object Consulting

Java™

Sun microsystems

Goal

Learn how a real-world enterprise portal was built using Sun's Portal Server, and how the Java™ Portlet Specification meets the challenges encountered.

Agenda

Overview of the portal application

How the portal was implemented

Meeting the challenges

Migrating to the Java™ Portlet Specification

How the Portlet Specification helps deliver portals

Agenda

- Overview of the portal application
- How the portal was implemented
- Meeting the challenges
- Migrating to the Java™ Portlet Specification
- How the Portlet Specification helps deliver portals

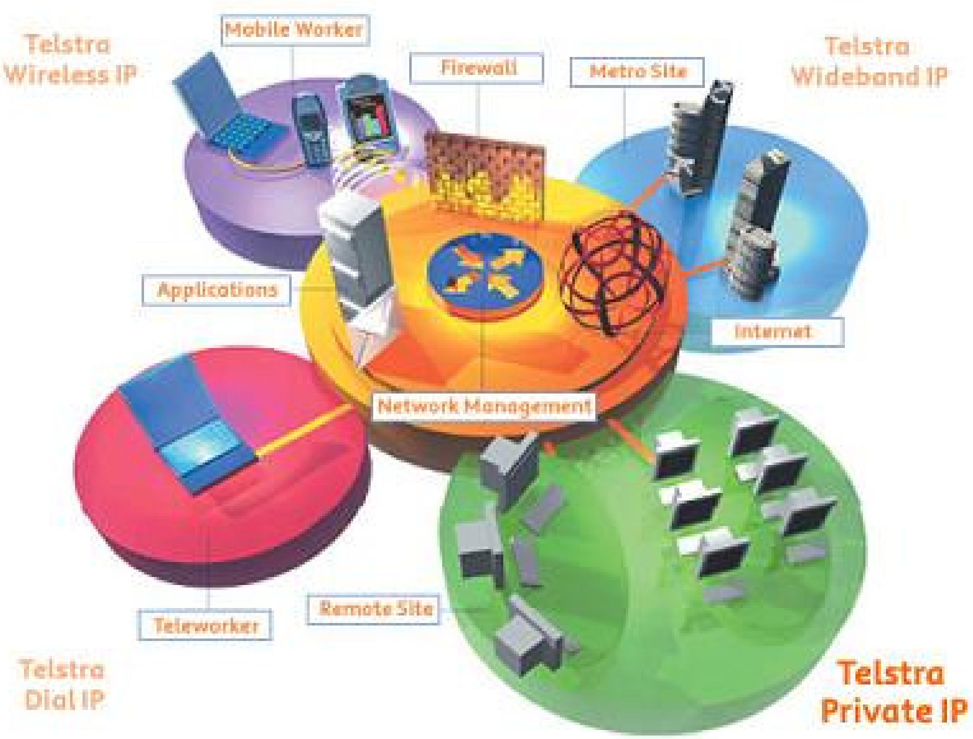
System Overview

The customer

- Telstra
  - Australia's leading tele-communications company

The product

- Telstra Private IP
  - A growing range of IP-based products and services



The Business Problem

- Disparate applications exist to manage products, services and users
- Current network management processes are manual
  - Front-of-house staff add new VPN users, change NAT rules
- Introducing new services is hard

Business Drivers

Why build a portal?

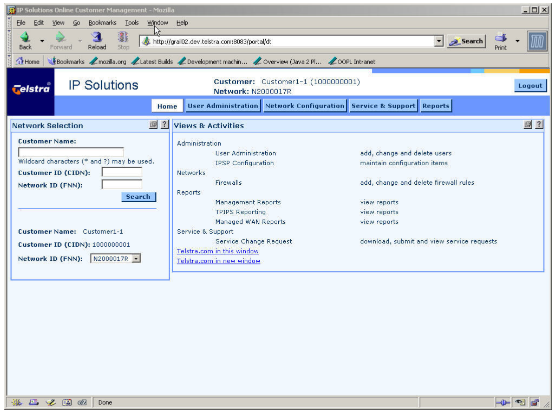
- To aggregate common applications
- Enable secure customer self-service of IP networks
- To provide a framework for quickly deploying new IP products, services and features

Key Features

Web Report Integration

Virtual Desktop

Service Change Requests



http://grail02.dev.telstra.com:8083/portal/dt

IP Solutions

Customer: Customer1-1 (1000000001)
Network: N2000017R

Network Selection

Customer Name:
Wildcard characters (* and ?) may be used.
Customer ID (CIDN):
Network ID (FNN):

Customer Name: Customer1-1
Customer ID (CIDN): 1000000001
Network ID (FNN): N2000017R

Views & Activities

Administration

Administration		
	User Administration	add, change and delete users
	IPSP Configuration	maintain configuration items
Networks		
	Firewalls	add, change and delete firewall rules
Reports		
	Management Reports	view reports
	TPIPS Reporting	view reports
	Managed WAN Reports	view reports
Service & Support		
	Service Change Request	download, submit and view service requests

Telstra.com in this window
Telstra.com in new window

Firewall Configuration

Management Reports

Network User Administration

Complex Application Integration

Framework services
Domain model
Logging
Auditing
Scheduling

Network Visualisation

Agenda

Overview of the portal application

How the portal was implemented

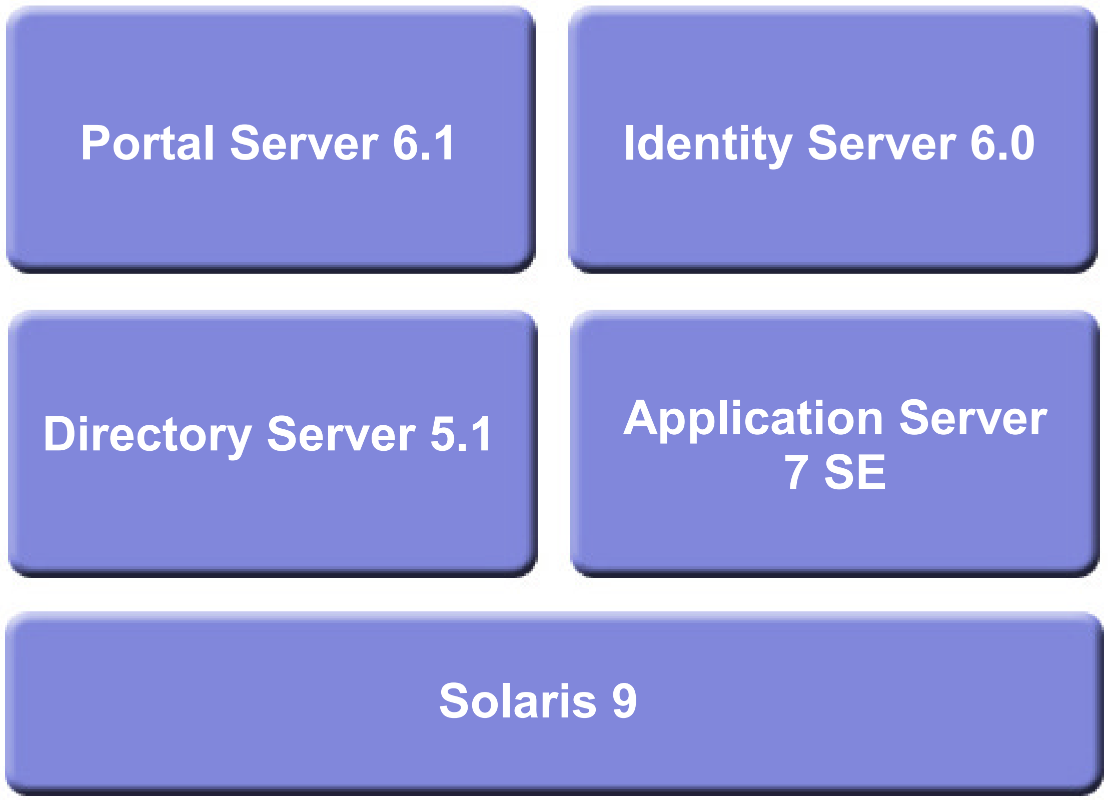
Meeting the challenges

Migrating to the Java™ Portlet Specification

How the Portlet Specification helps deliver portals

Software Platform

The Sun™ ONE stack



Portal Server 6.1

Identity Server 6.0

Directory Server 5.1

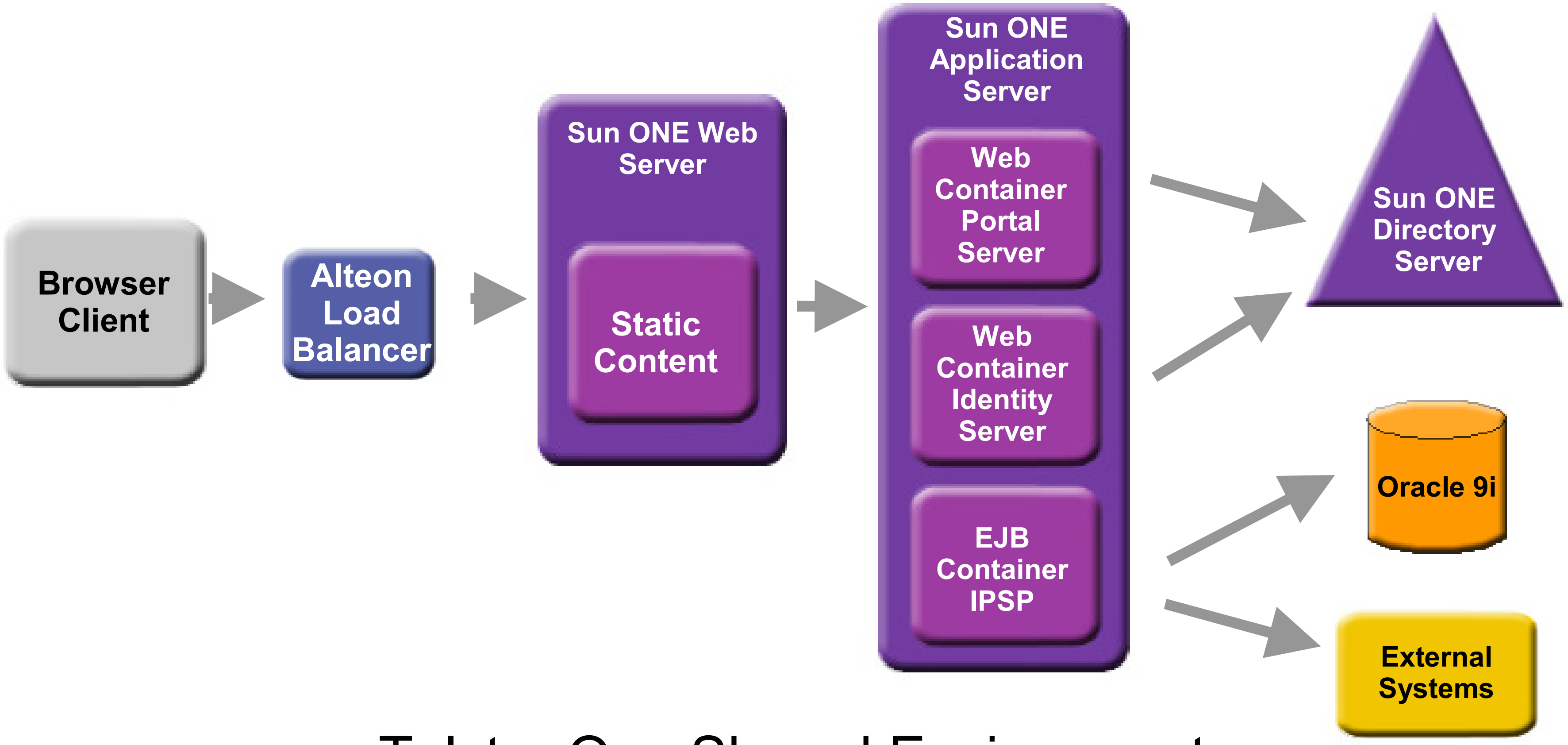
Application Server 7 SE

Solaris 9

Telstra.One Shared Environment

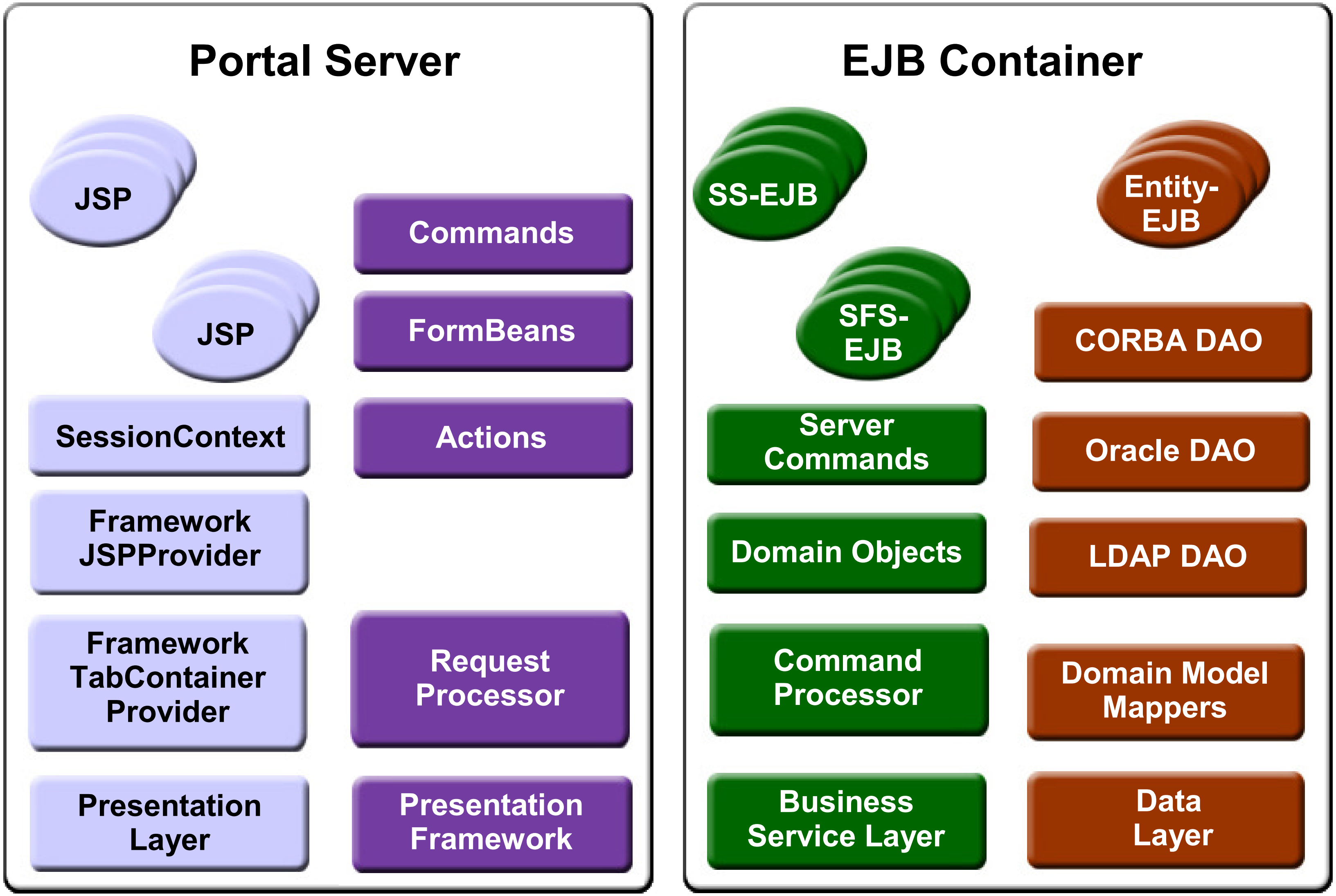
- Telstra.One is a realisation of Telstra's Online SOE
  - Hardware and software platform
- Supports a three-tiered architecture
  - Sun ONE J2EE™ software
  - Oracle DBMS
  - Documentum CMS
- IPSP is the first large-scale application deployed

Deployment Platform

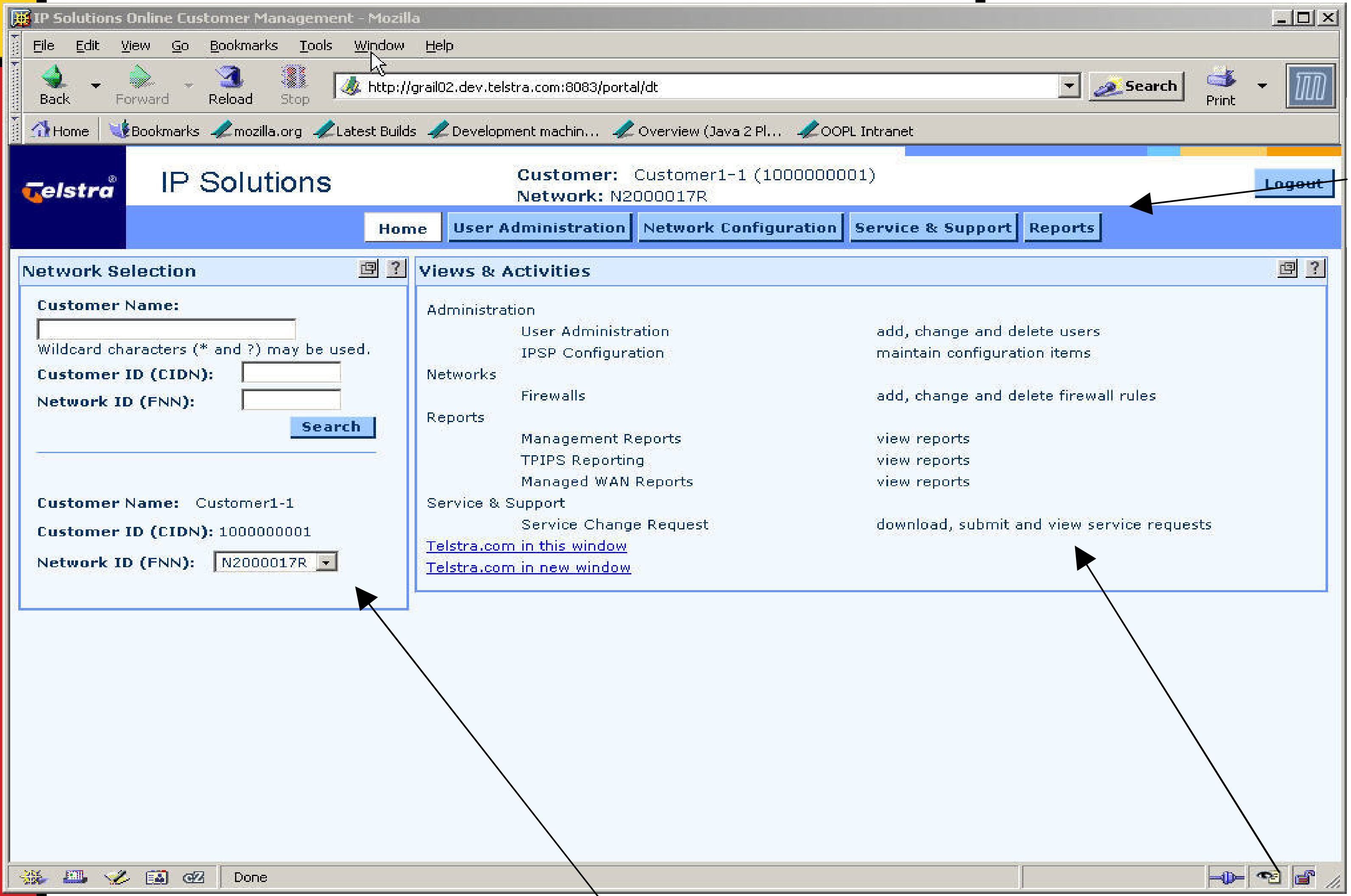


Telstra.One Shared Environment

IPSP Architectural Approach



Portal Server 6.1 Desktop


IP Solutions Online Customer Management - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
Search Print
Home Bookmarks mozilla.org Latest Builds Development machin... Overview (Java 2 Pl... OOPL Intranet
Telstra
Logout
Home User Administration Network Configuration Service & Support Reports
Done

http://grail02.dev.telstra.com:8083/portal/dt

IP Solutions

Customer: Customer1-1 (1000000001)
Network: N2000017R

Network Selection

Customer Name:

Wildcard characters (* and ?) may be used.

Customer ID (CIDN):

Network ID (FNN):

Search

Customer Name: Customer1-1

Customer ID (CIDN): 1000000001

Network ID (FNN): N2000017R

Views & Activities

Administration		
	User Administration	add, change and delete users
	IPSP Configuration	maintain configuration items
Networks		
	Firewalls	add, change and delete firewall rules
Reports		
	Management Reports	view reports
	TPIPS Reporting	view reports
	Managed WAN Reports	view reports
Service & Support		
	Service Change Request	download, submit and view service requests

Telstra.com in this window

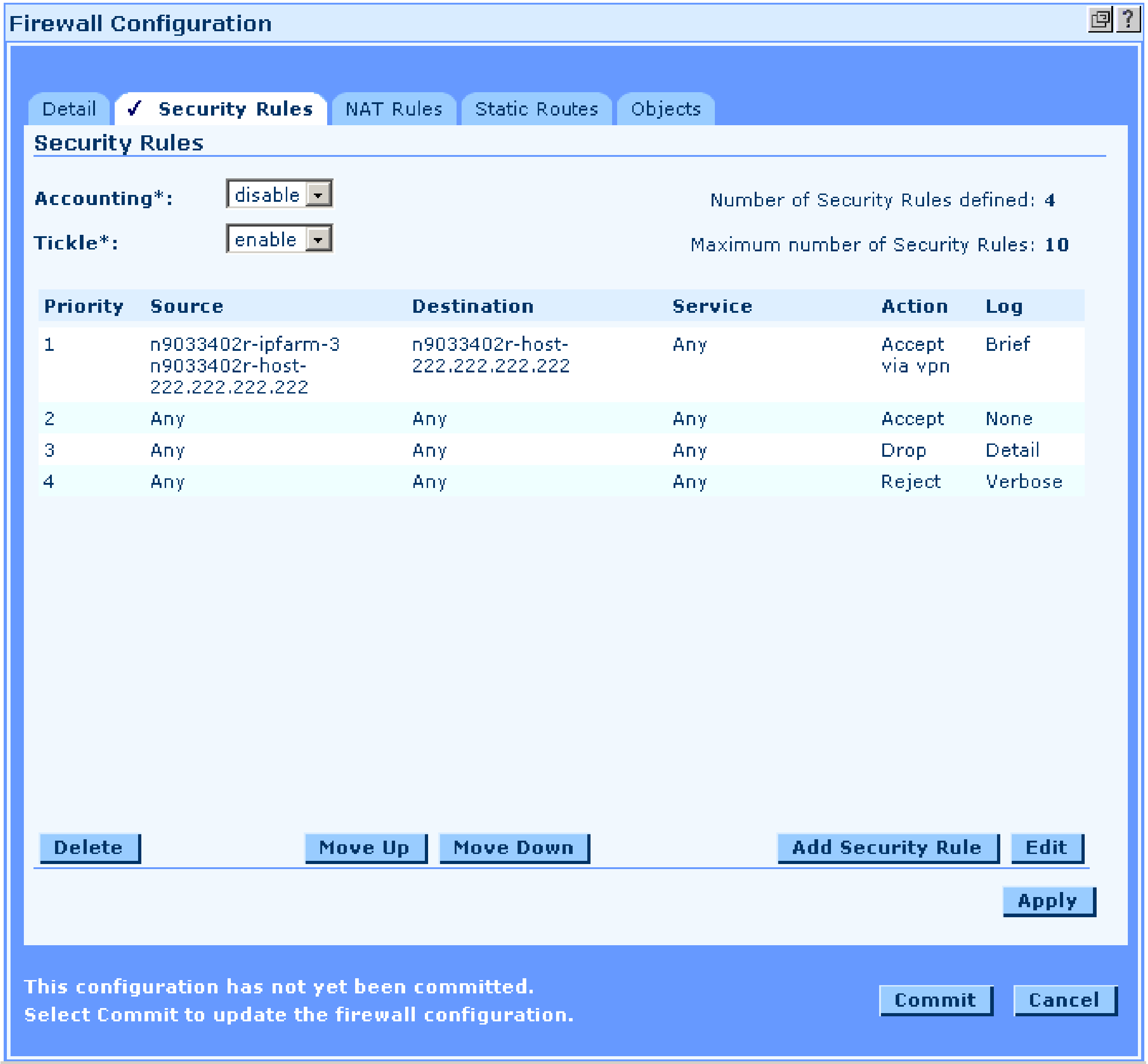
Telstra.com in new window

Custom tabs.jsp

JSPProvider-based channel

XMLProvider-based channel

Channel Page Flow



Firewall Configuration

Security Rules

Accounting*: disable

Tickle*: enable

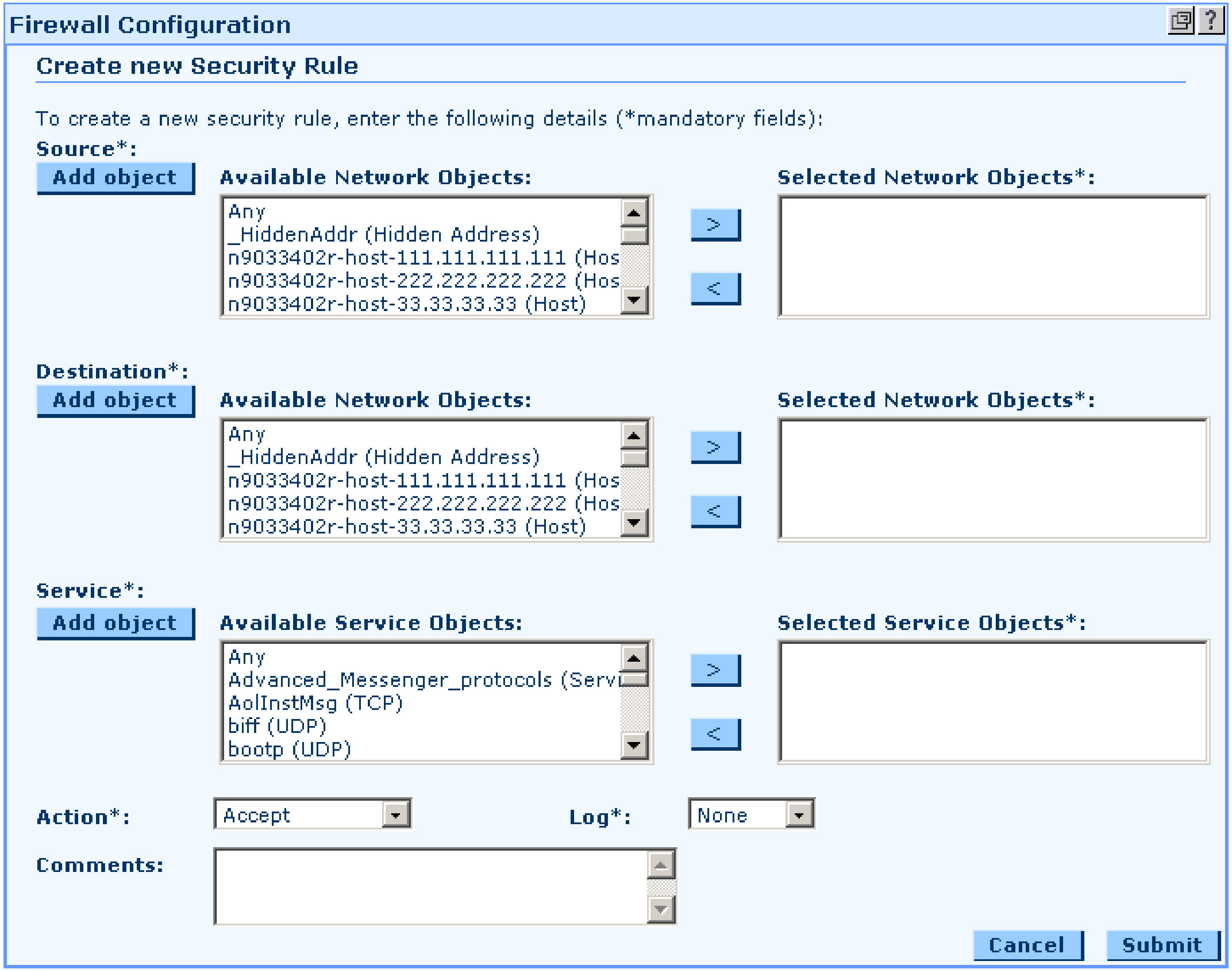
Number of Security Rules defined: 4

Maximum number of Security Rules: 10

Priority	Source	Destination	Service	Action	Log
1	n9033402r-ipfarm-3 n9033402r-host-222.222.222.222	n9033402r-host-222.222.222.222	Any	Accept via vpn	Brief
2	Any	Any	Any	Accept	None
3	Any	Any	Any	Drop	Detail
4	Any	Any	Any	Reject	Verbose

This configuration has not yet been committed.
Select Commit to update the firewall configuration.

Complex UI Components



Firewall Configuration

Create new Security Rule

To create a new security rule, enter the following details (*mandatory fields):

Source*:

Available Network Objects:

- Any
- _HiddenAddr (Hidden Address)
- n9033402r-host-111.111.111.111 (Hos
- n9033402r-host-222.222.222.222 (Hos
- n9033402r-host-33.33.33.33 (Host)

Selected Network Objects*:

Destination*:

Available Network Objects:

- Any
- _HiddenAddr (Hidden Address)
- n9033402r-host-111.111.111.111 (Hos
- n9033402r-host-222.222.222.222 (Hos
- n9033402r-host-33.33.33.33 (Host)

Selected Network Objects*:

Service*:

Available Service Objects:

- Any
- Advanced_Messenger_protocols (Servi
- AolInstMsg (TCP)
- biff (UDP)
- bootp (UDP)

Selected Service Objects*:

Action*: Accept

Log*: None

Comments:

Key Challenges

- Maintaining the desktop metaphor
  - Portal becomes the user's virtual desktop
  - Channels must be rendered within the desktop
  - There is no escaping the portal
- Hosted applications are complex
  - Supporting configurable navigation flows

Key Challenges

- Session management
  - Maintaining complex context state between channels
  - Supporting a portal-wide domain model
- Development
  - Developing channels in a MS Windows environment
  - Unit testing channels

Terminology

This	*Means*
Portal	The rendered desktop containing aggregated content
Channel	An individual piece of content delivered in a portal (Sun term)
Provider	A channel's backing Java class implementation (Sun term)
Portlet	Analogous to a channel
Sun™ ONE	What Sun Java System used to be called
Sun Java System	What Sun ONE is now called

Agenda

Overview of the portal application

How the portal was implemented

Meeting the challenges

Migrating to the Java™ Portlet Specification

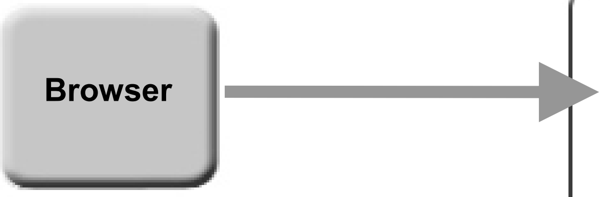
How the Portlet Specification helps deliver portals

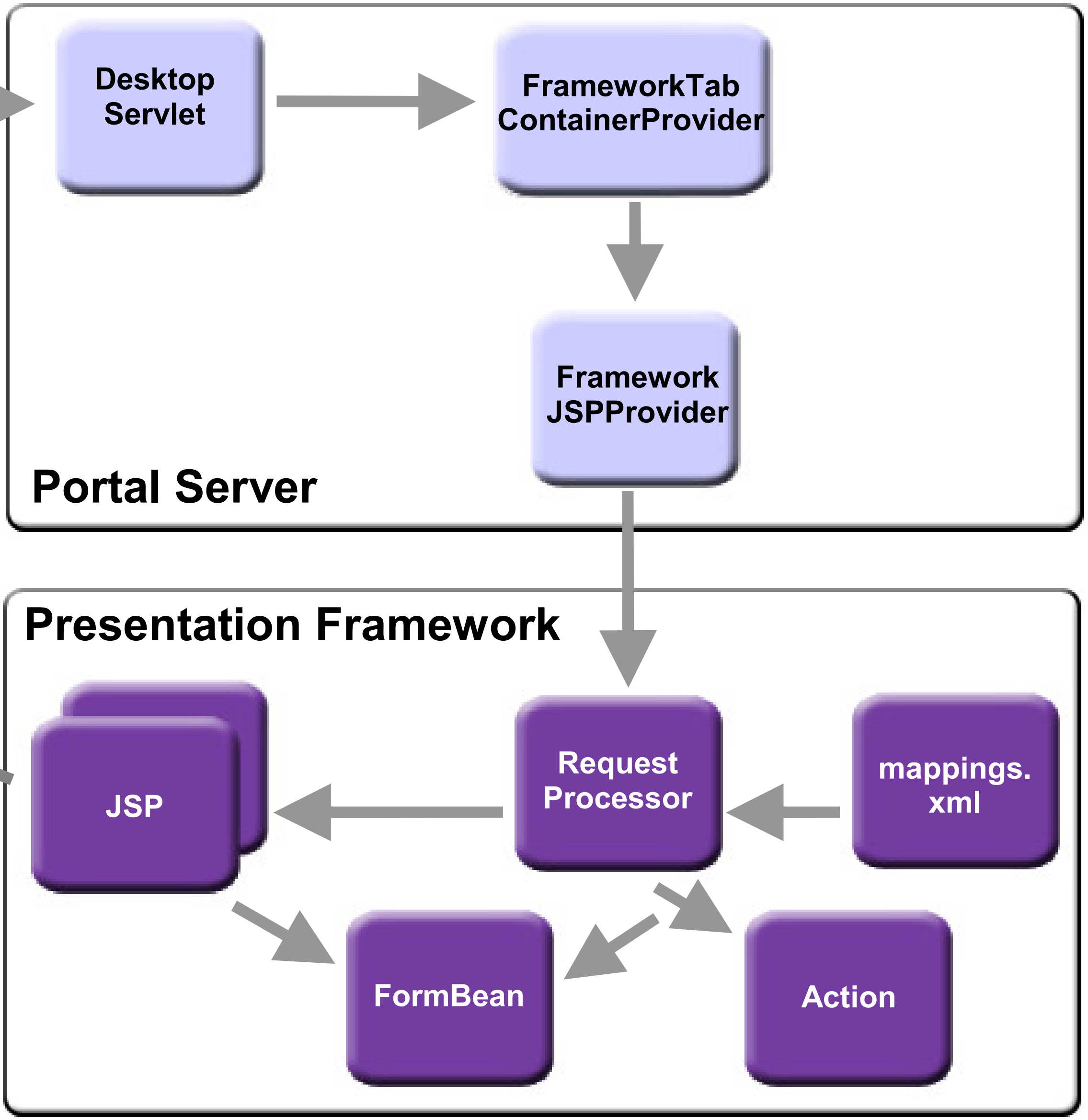
The First Challenge

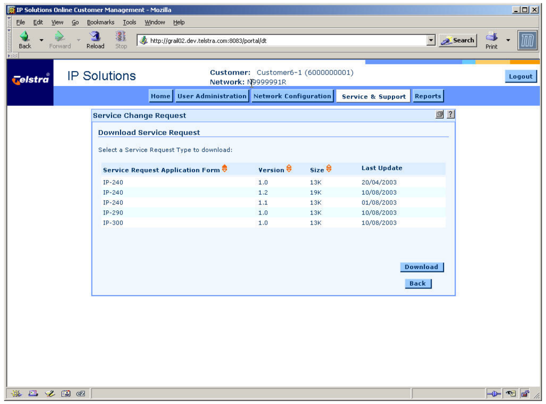
Handling complex navigation flows

- Portal Server building blocks are targeted at simple request-response interaction
  - Stock ticker, bookmarks, web services RPC
- Portal aggregates content, then delegates user actions
  - User seamlessly thrown to external applications
  - IPSP requires keeping the user in the desktop
- Developers are used to using MVC-based frameworks

A Portal Presentation Framework







http://grail02.dev.telstra.com:8083/portal/dt

IP Solutions

Customer: Customer6-1 (6000000001)
Network: N9999991R

Service Change Request

Download Service Request

Select a Service Request Type to download:

Service Request Application Form	Version	Size	Last Update
IP-240	1.0	13K	20/04/2003
IP-240	1.2	19K	10/08/2003
IP-240	1.1	13K	01/08/2003
IP-290	1.0	13K	10/08/2003
IP-300	1.0	13K	10/08/2003

Extending JSPProvider

Sample from FrameworkJSPProvider

```
public class FrameworkJSPProvider
  extends JSPProvider {
    private String next; // caches the next view
    private RequestProcessor requestProcessor;

    public StringBuffer getContent(
                    HttpServletRequest request,
                    HttpServletResponse response)
          throws ProviderException {
      ...
      next =
        requestProcessor.processRequest(request);
      storeSessionAttributes(request.getSession());
      ... // exception handling

      return super.getContent(request, response);
    }
}
```

Processing User Requests

Sample from RequestProcessor

```
public class RequestProcessor {
    private HashMap urlMappings;

    public String processRequest(
           HttpServletRequest request)
         throws PresentationFrameworkException {

      String selectedURL = getSelectedURL(request);
      return getNextPage(request,
        selectedURL, false);
    }
}
```

Generating the Next View

RequestProcessor (Cont.)

```
private String getNextPage(
        HttpServletRequest request,
        String selectedURL, boolean redirect)
          throws PresentationFrameworkException {

  URLMapping mapping = getURLMapping(selectedURL);
  FormBean formBean = initialiseFormBean(request,
    mapping, redirect);

  if (!formBean.validate()) {
    request.getSession().setAttribute("errors",
      formBean.getValidationErrors());

    // caller must determine what to do
    return null;
  }
  ...
```

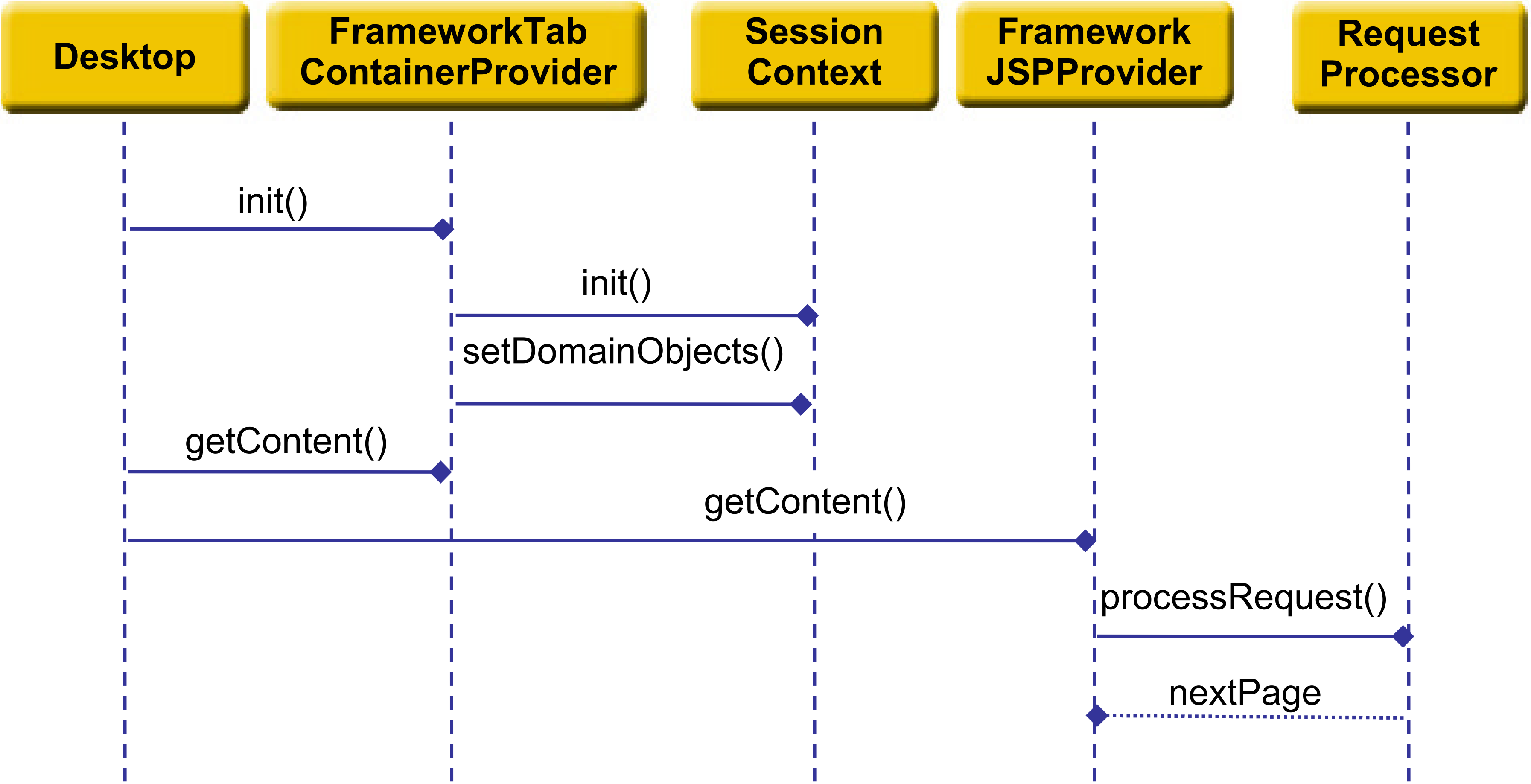
Generating the Next View

RequestProcessor (Cont.)

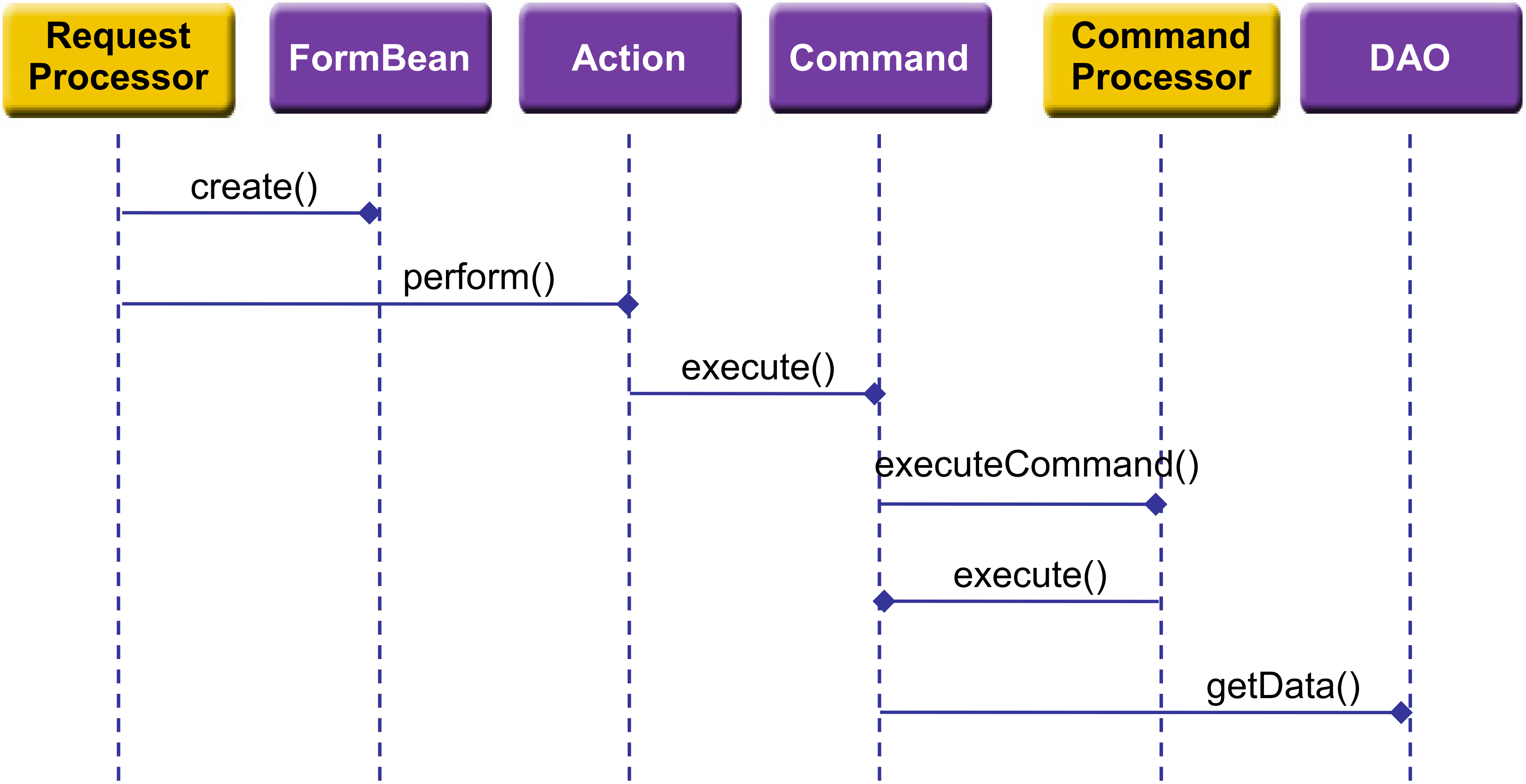
```
  Action action = getAction(mapping);
    ... // initialise action

  String nextPage = action.perform(request);
  return (nextPage == null ? mapping.getScreen() :
    getNextPage(request, nextPage, true));
}
```

Client Requests to Portal



Presentation Framework Takes Control



Portal Presentation Framework

Considerations

- Emulates Struts
  - Consideration given to integrating Struts
  - DesktopServlet is the front controller
- Moves navigation flow away from Portal Server
  - Makes the “next” page dynamic
  - Delegates to Actions
  - RequestProcessor is the single point of entry to flow logic
- Retains applications in the Portal context
  - Components share the same session context

The Second Challenge

Non-standard session management

- Web session is not made available to Providers
  - Portal substitutes the HttpSession
  - HttpSession interface is narrowed
- Each JSPProvider uses its own JSP™ engine
  - With its own session object
  - Enables channel hot deployment

Portal Session Management

Solution

- ProviderContext maintains session-scoped domain model
  - ProviderContext is shared by all channels
- Per-channel session attributes
  - Cached on FrameworkJSPProvider
- Custom tags and interfaces
  - Hide how session state is maintained

Hiding the Session Implementation

SessionContext interface

- Decouples framework from PAPI
  - Presentation Framework classes only know about SessionContext
- Delegates to ProviderContext
- Holds domain objects
  - User
  - Customer
  - VPN

Custom Tags

JSP pages using custom tags

```
<ipsp:useBean id=“formBean” scope="session"
  className="
com.telstra.ipsp.framework.context.ContextFormBean"
/>

<ipsp:getUser/>
<%
  boolean isSearchSuccessful =
    formBean.isSearchSuccessful();
  CustomerProfile currentCustomer =
    user.getCurrentCustomer();
%>
```

Session Management

Example from GetUserTag

```
public int doStartTag() throws JspException {
  FrameworkJSPProvider provider =
    (FrameworkJSPProvider)pageContext.getAttribute
      ("JSPProvider");

  SessionContext sc =
    (SessionContext) provider.getSessionAttribute
      (Constants.SESSION_CONTEXT);

  pageContext.setAttribute("user", sc.getUser());
  return SKIP_BODY;
}
```

Session Management

Example from FrameworkJSPProvider

```
private Map sessionAttributes; // instance cache

public Object getSessionAttribute(String key){

  if (Constants.SESSION_CONTEXT.equals(key)) {
      // shared scoped
    return sessionContext;
  }
    // provider scoped
  return sessionAttributes.get(key);
}
```

The Third Challenge

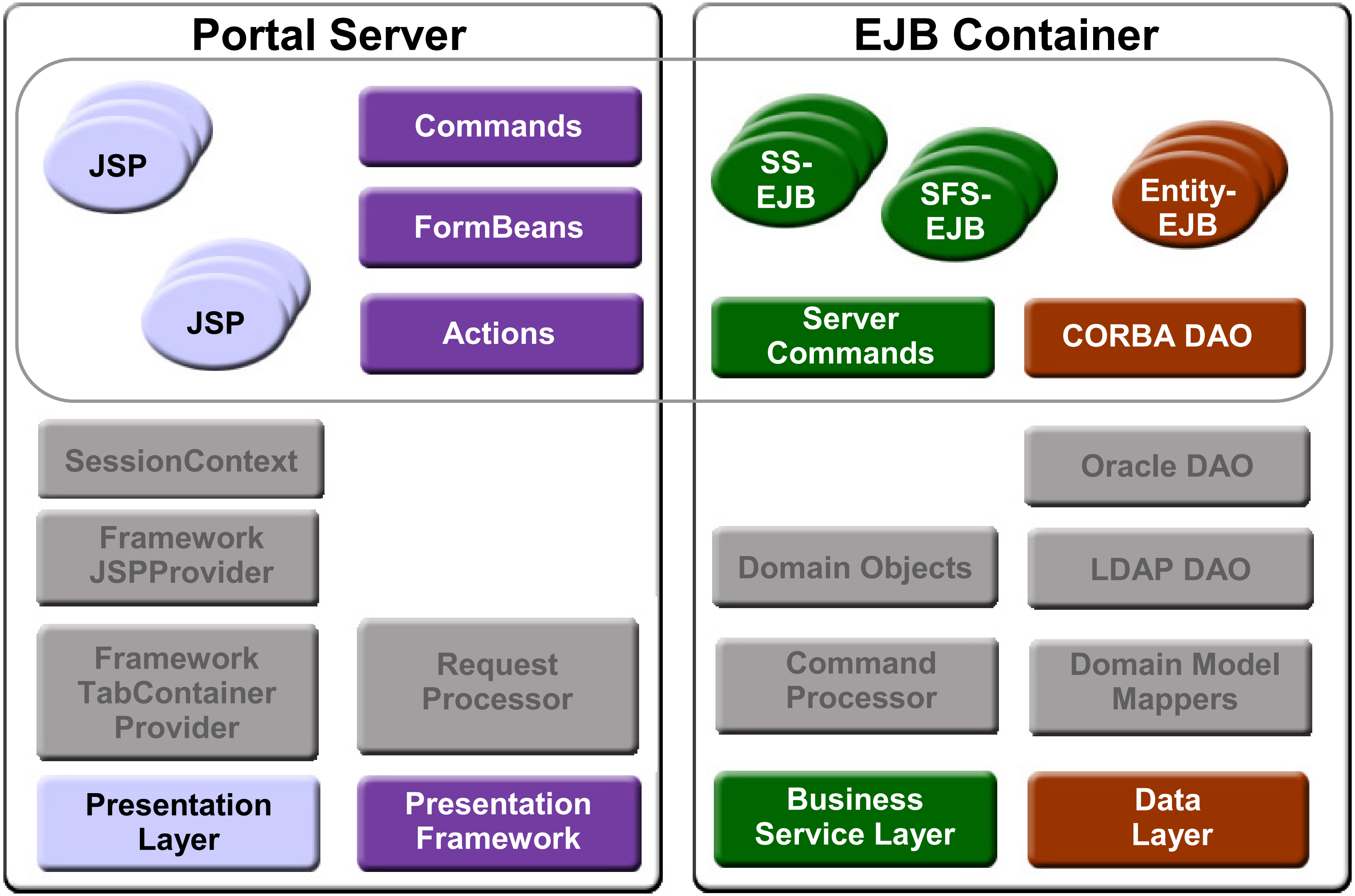
Developing and unit testing channels

- Portal Server only runs on Solaris
  - Dev environment MS Windows-based
- Portlet Builder
  - Version 1.0 could not be used with Sun ONE Studio 5
- ProviderServlet
  - Web application that emulates Portal Server environment
  - Deployable locally to Tomcat or Application Server
  - Sets up mock SessionContext, JSPProvider

Development Environment

- Channels can be developed as standard web apps
- Unit testing web tier components decoupled from Portal Server
- Use your tool of choice

Putting It All Together



Agenda

Overview of the portal application

How the portal was implemented

Meeting the challenges

Migrating to the Java™ Portlet Specification

How the Portlet Specification helps deliver portals

Portlets in Portal Server 6.2

Providers and portlets

- Portal Server 6.2 implements a portlet container
- A portlet is conceptually equivalent to a Provider
- Provides the backing class for a leaf channel
- New tools to deploy
  - pdeploy

Migrating to GenericPortlet

Sample from FrameworkPortlet

```
public class FrameworkPortlet extends GenericPortlet
{
  private RequestProcessor requestProcessor;
  private String startPage;
  private boolean initialiseHomePage;

  public void init() throws PortletException {
    PortletConfig pc = getPortletConfig();

    start = pc.getInitParameter("startPage");
    String s =
        pc.getInitParameter("initialiseHomePage");
    initialiseHomePage =
        Boolean.valueOf(s).booleanValue();
}
```

Migrating to GenericPortlet

Sample from FrameworkPortlet (Cont.)

```
public void processAction(ActionRequest request,
                          ActionResponse response)
            throws PortletException, IOException {

    HttpServletRequest wrapper =
        new RequestWrapper(request);
    String next =
      requestProcessor.processRequest(wrapper);
    request.getPortletSession().setAttribute(
        "next", next);
}
```

Migrating to GenericPortlet

Sample from FrameworkPortlet (Cont.)

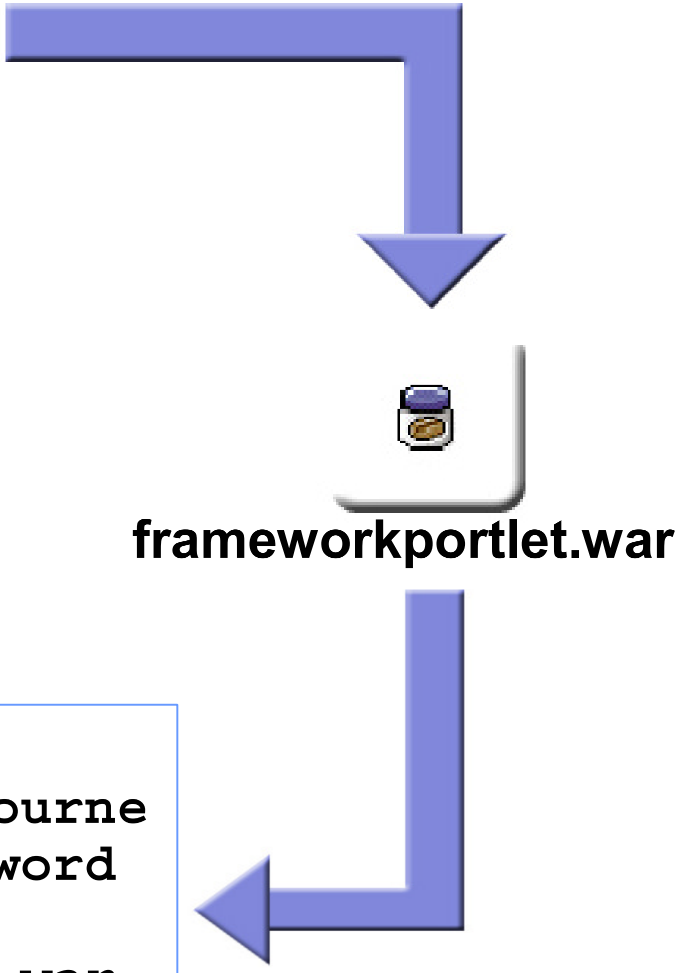
```
protected void doView(RenderRequest request,
                      RenderResponse response)
            throws PortletException, IOException {

  response.setContentType(
  request.getResponseContentType());
  String next = (String)
    request.getPortletSession().getAttribute("next");
    if (next == null) {
      // first render call
      next = startPage;
  }
  PortletRequestDispatcher rd =
    getPortletContext().getRequestDispatcher(next);
  rd.include(request, response);
}
```

Deploying the Portlet

portlet.xml

```
<portlet-app>
  <portlet>
    ...
  </portlet>
</portlet-app>
```

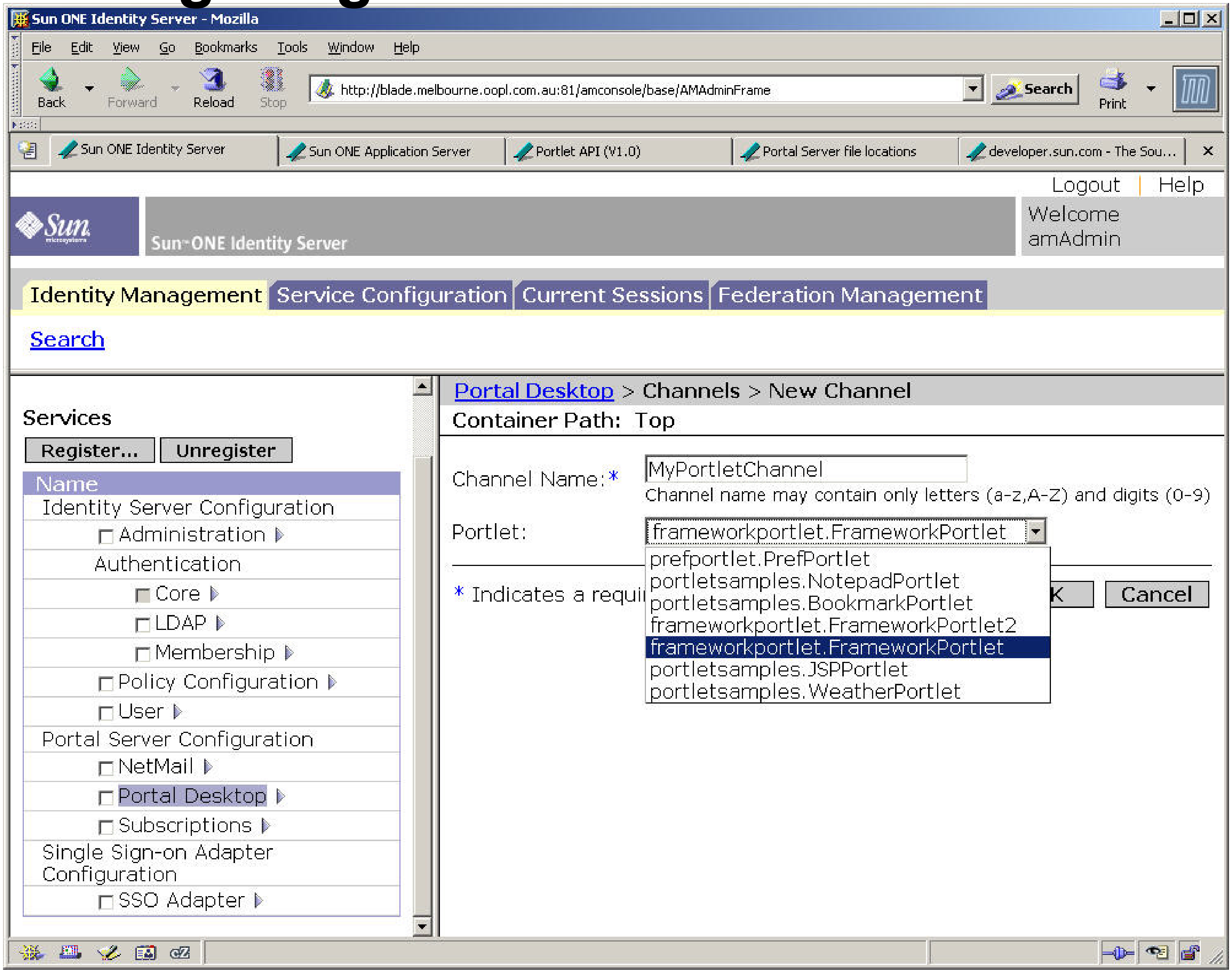


frameworkportlet.war

pdeploy

```
./pdeploy deploy -u
"uid=amadmin,ou=People,dc=melbourne
,dc=oopl,dc=com,dc=au“ -w password
-p password -g
/opt/portlets/frameworkportlet.war
```

Configuring the Channel



http://blade.melbourne.oopl.com.au:81/amconsole/base/AMAdminFrame

Welcome amAdmin

Services

Name
Identity Server Configuration
□ Administration
Authentication
□ Core
□ LDAP
□ Membership
□ Policy Configuration
□ User
Portal Server Configuration
□ NetMail
□ Portal Desktop
□ Subscriptions
Single Sign-on Adapter Configuration
□ SSO Adapter

Portal Desktop > Channels > New Channel

Container Path: Top

Channel Name:* MyPortletChannel

Channel name may contain only letters (a-z,A-Z) and digits (0-9)

Portlet: frameworkportlet.FrameworkPortlet

- prefportlet.PrefPortlet
- portletsamples.NotepadPortlet
- portletsamples.BookmarkPortlet
- frameworkportlet.FrameworkPortlet2
- frameworkportlet.FrameworkPortlet
- portletsamples.JSPPortlet
- portletsamples.WeatherPortlet

* Indicates a requi

Considerations for Migration

What to watch out for

- PortletRequest does not extend HttpServletRequest
  - IPSP Presentation Framework expects HttpServletRequest
- No standard portlet page flow
  - Is there a need for standardising portlet MVC?

Portlet Lifecycle

Providers and portlets have different lifecycles

- Providers live and die with ProviderContext
  - Session-based lifecycle
- Portlets depend on portlet container
  - Long-lived
  - Service multiple clients
  - Container-based lifecycle

Considerations for Migration

- Portlets do not have access to ProviderContext
  - No direct support for passing data between providers and portlets
  - No direct API access to PAPI
- Portability works!
  - FrameworkPortlet was built for Portal Server 6.2
  - Seamlessly deployed to Pluto

Agenda

Overview of the portal application

How the portal was implemented

Meeting the challenges

Migrating to the Java™ Portlet Specification

How the Portlet Specification helps deliver portals

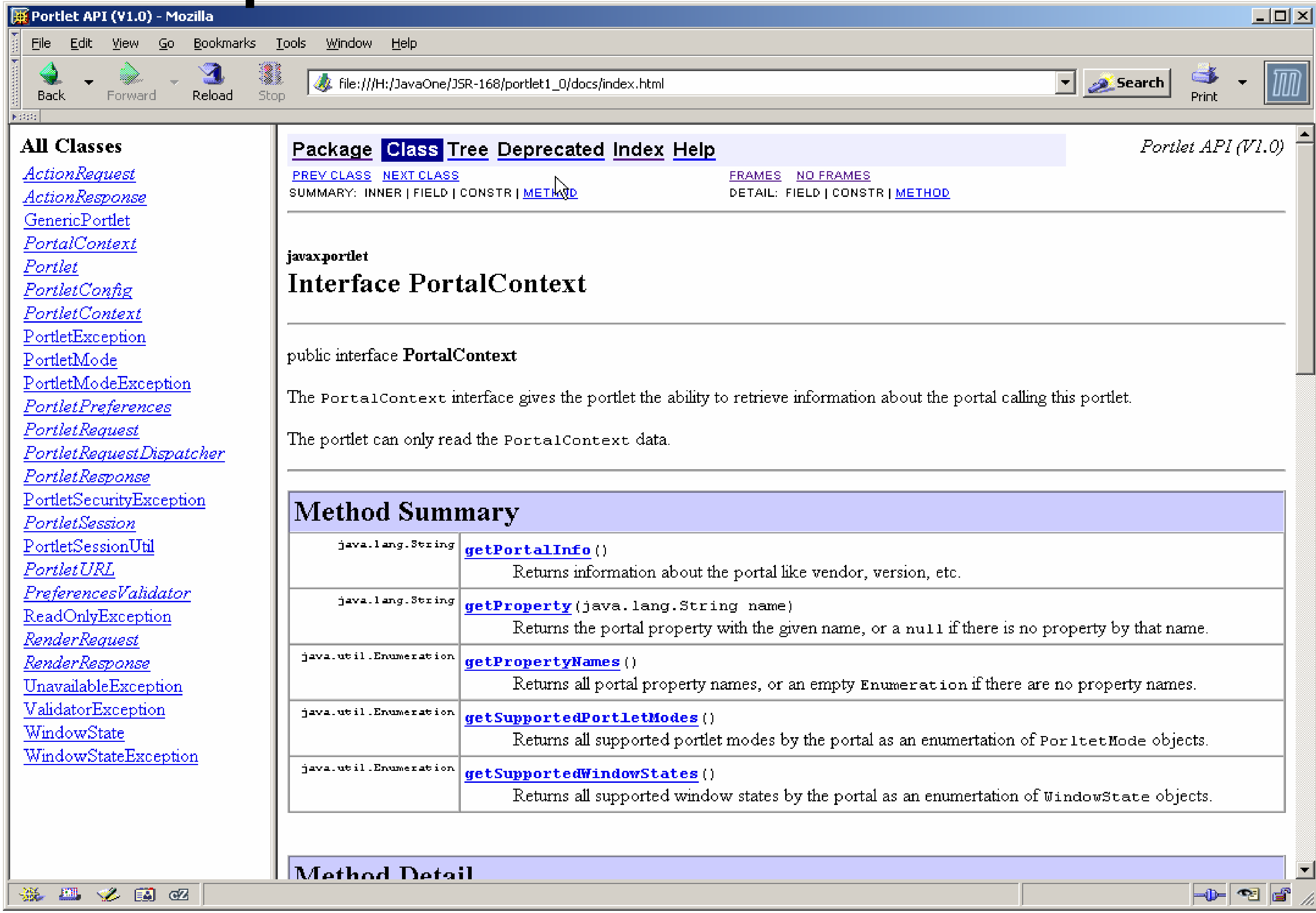
A Java Portlet Is a Web Application

- Portlets are J2EE web components
  - Flat learning curve for web developers
  - Natural fit for MVC-based applications
- Portlets can include servlets, JSP pages and HTML
  - PortletRequestDispatcher allows content to be rendered in-portlet

Session Management

- Portlet session APIs follow servlet specification
- All portlet components share the web session
- Session data can be isolated
  - `PortletSession.APPLICATION_SCOPE`
  - `PortletSession.PORTLET_SCOPE`
- Developers are freed from proprietary session management

A Simple API


Portlet API (V1.0) - Mozilla
File
Edit
View
Go
Bookmarks
Tools
Window
Help
Back
Forward
Reload
Stop
file:///H:/JavaOne/JSR-168/portlet1_0/docs/index.html
Search
Print
Package
Class
Tree
Deprecated
Index
Help
PREV CLASS
NEXT CLASS
FRAMES
NO FRAMES
SUMMARY: INNER | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD

All Classes

ActionRequest
ActionResponse
GenericPortlet
PortalContext
Portlet
PortletConfig
PortletContext
PortletException
PortletMode
PortletModeException
PortletPreferences
PortletRequest
PortletRequestDispatcher
PortletResponse
PortletSecurityException
PortletSession
PortletSessionUtil
PortletURL
PreferencesValidator
ReadOnlyException
RenderRequest
RenderResponse
UnavailableException
ValidatorException
WindowState
WindowStateException

Portlet API (V1.0)

javax.portlet

Interface PortalContext

public interface **PortalContext**

The `PortalContext` interface gives the portlet the ability to retrieve information about the portal calling this portlet.

The portlet can only read the `PortalContext` data.

Method Summary

`java.lang.String`	`getPortalInfo()` Returns information about the portal like vendor, version, etc.
`java.lang.String`	`getProperty(java.lang.String name)` Returns the portal property with the given name, or a `null` if there is no property by that name.
`java.util.Enumeration`	`getPropertyNames()` Returns all portal property names, or an empty `Enumeration` if there are no property names.
`java.util.Enumeration`	`getSupportedPortletModes()` Returns all supported portlet modes by the portal as an enumertation of `PorltetMode` objects.
`java.util.Enumeration`	`getSupportedWindowStates()` Returns all supported window states by the portal as an enumertation of `WindowState` objects.

Method Detail

Packaging

- Defined by Servlet Specification 2.3
- Familiar WAR structure
  - Integrate with existing tools
  - Standard ant builds
- Familiar deployment descriptors
- Standard classloader semantics

Pluggability and Portability

- Simplifies portlet development
  - Promotes choice of development environment
  - Remember to deploy early and often
- Use the reference implementation
  - Pluto is in early days, but provides a simple testing/prototyping environment

Lessons Learned

- Align the portal metaphor
  - Between business requirements and the technology
- Separation of presentation from business tiers is crucial
  - Ensure controller and business components can be developed and tested standalone
- Proprietary protocols constrain application design
  - Specifications ease the pain

Summary

- Telstra IP Solutions Portal
  - Aggregates Private IP products and services
- Extend Portal Server 6.1 to meet the challenges of complex applications
- Providers can easily be migrated to the Java Portlet Specification
- JSR-168 lets us treat portlets as J2EE components

Conclusion

The Java Portlet Specification lets good web developers become expert portlet developers.

For More Information

- Telstra Private IP
  - http://www.telstra.com.au/privateip/index.htm
- Java Portlet Specification
  - http://www.jcp.org/en/jsr/detail?id=168
- Javaworld portlet article
  - http://www.javaworld.com/javaworld/jw-08-2003/jw-0801-portlet.html
- Pluto portlet reference implementation
  - http://jakarta.apache.org/pluto/
- Sun Java System Portal Server 6.2
  - http://wwws.sun.com/software/products/portal_srvr/home_portal.html

Q&A

JavaOne

Sun's 2004 Worldwide Java Developer Conference

Building a Portal and the Benefits of the Portlet Specification

A Case Study

java.sun.com/javaone/sf

Peter Moran

Senior Software Engineer

www.objectconsulting.com.au

Object Consulting