

BEAWorld2005

Federated Portals

Subbu Allamaraju
Staff Engineer
BEA Systems Inc

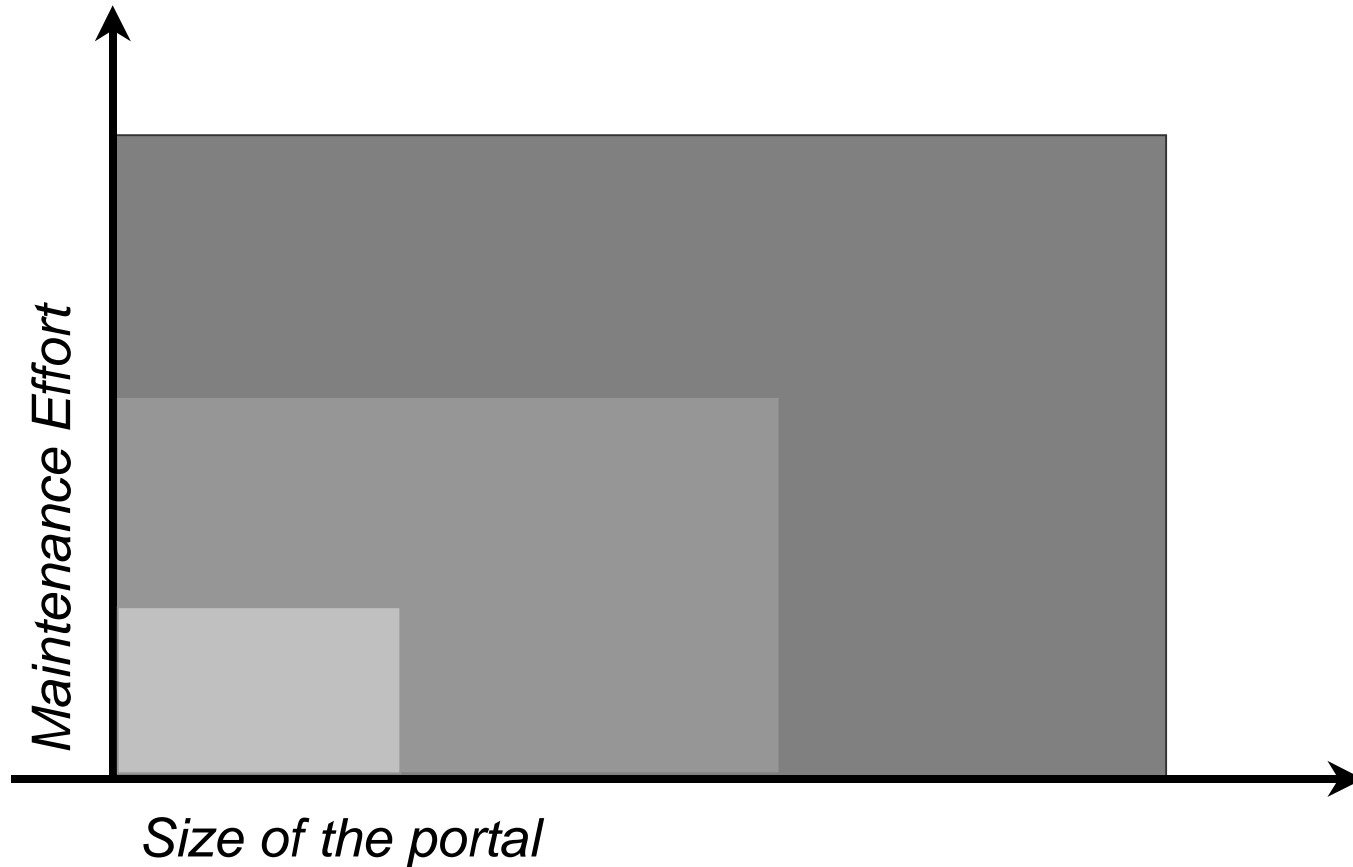


Agenda

- Federated portals and SOA
- Federated portals internals
- Best practices
- Upcoming features
- Q&A



Non-Federated Portal Maintenance



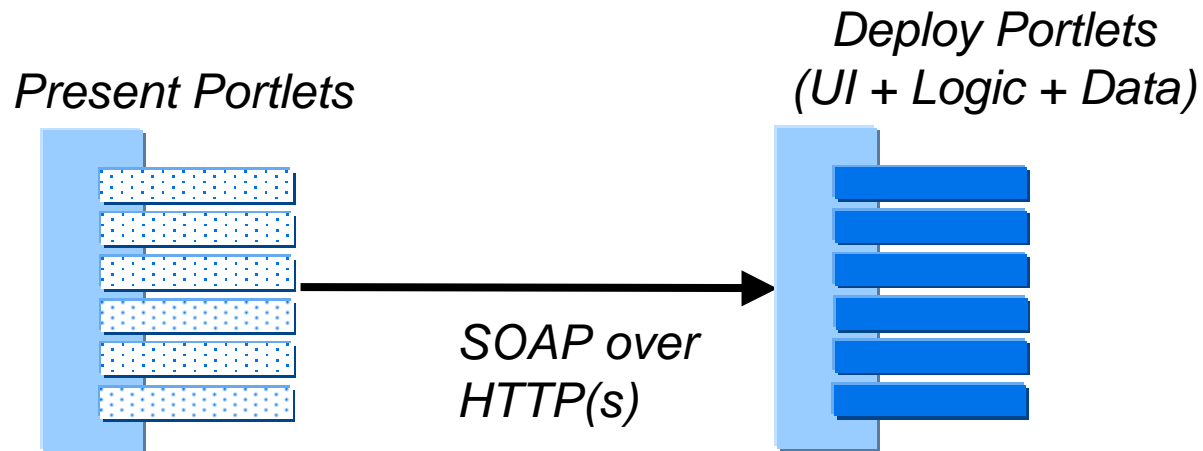
Maintenance effort: Bug fixes, enhancements, adding new portlets, testing, moving the portal from dev to staging to production etc.

Federated Portals

- Federated portals

- ▶ Aggregate user facing applications (portlets) from across the enterprise into a portal
 - Distributed
 - Decoupled
 - Collaborating
- ▶ Plug-and-play SOA
 - Find and use – no programming
- ▶ Standards based
 - SOAP, WSDL
 - **Web Services for Remote Portlets (WSRP)**
 - WS-Security, SAML
 - UDDI

Key Actors



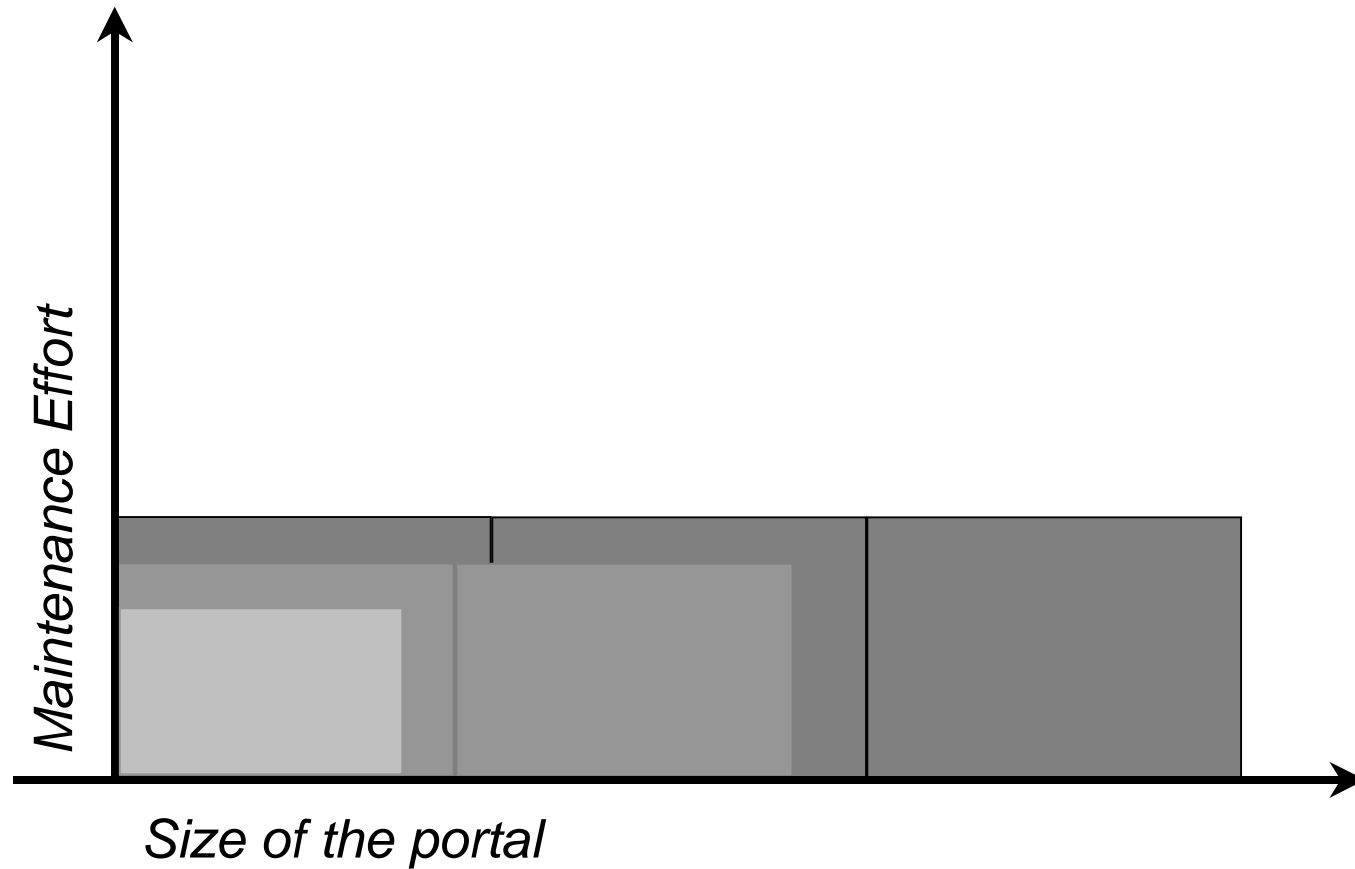
Consumer

- *A runtime for portals and proxy portlets*
- *Offers a Web front-end*
- *Administration-centric*
- *A web app in WLP*

Producer

- *A container for portlets (UI and business logic)*
- *Offers a SOAP front-end*
- *Application-centric*
- *A web app in WLP*

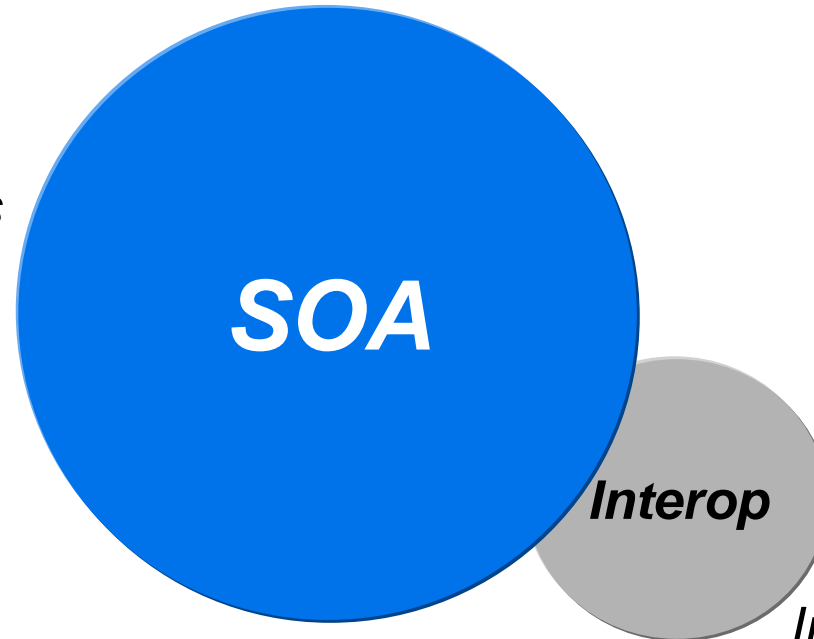
Federated Portal Maintenance



Maintenance effort decentralized

Federated Portals and SOA

Build loosely-coupled portals



Interoperate with other portals

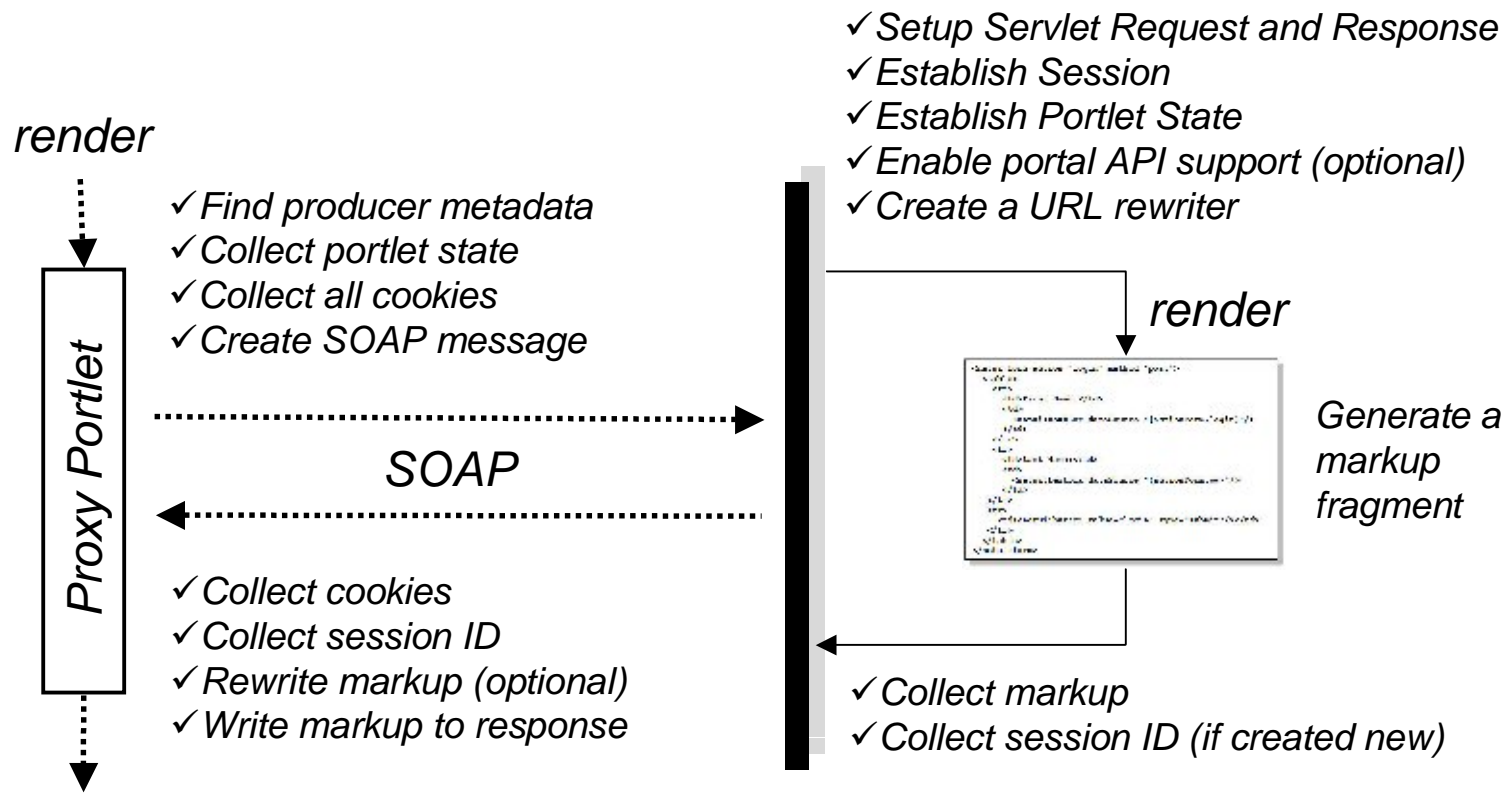
Service-oriented, Loosely-coupled, Standards-based

Goals of the Session

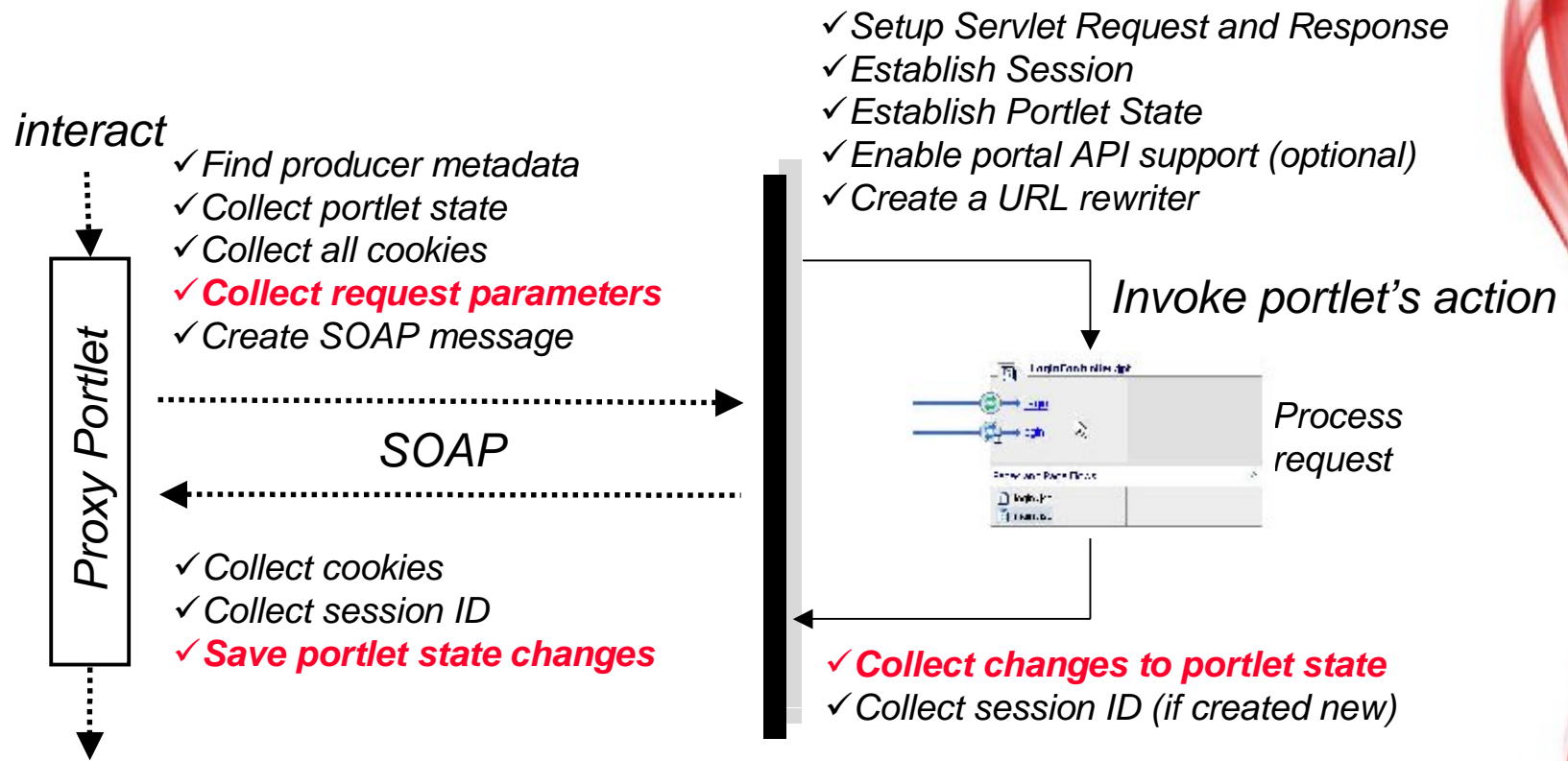
- Federation Internals
 - ▶ Portlet life cycle stages
- How to make federation succeed?
 - ▶ Best Practices
 - ▶ Guarantee loose-coupling
- Upcoming features
 - ▶ Brief overview of new features



Rendering a Remote Portlet



Interacting with a Remote Portlet



Rendering vs. Interaction

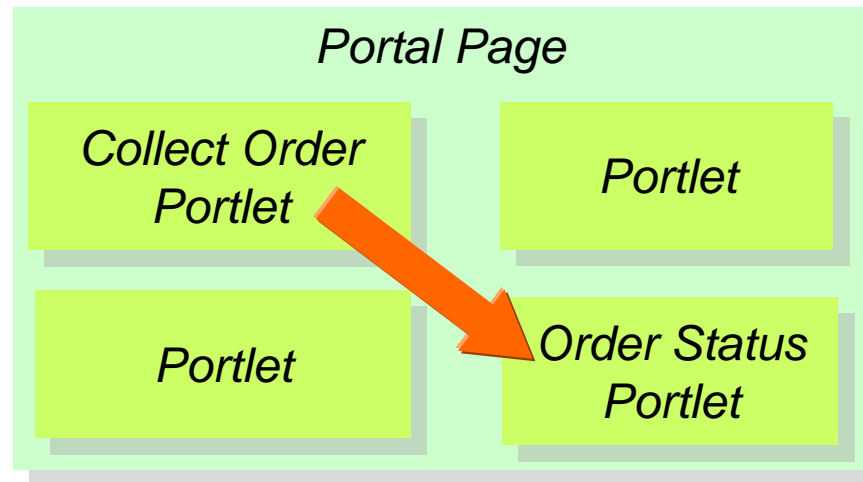
Render Phase		Interaction Phase	
Focus on presentation (view)		Focus on business logic (controller)	
May be replayed several times		Driven by user interaction	
Idempotent	No state/mode changes	Non-idempotent	Can ask for mode/state changes
	No changes to portlet preferences		Can change portlet preferences
	Can not redirect user		Can redirect user
Generate markup		Can optionally generate markup	



Decoupling Lesson I

- Portlets will not get the same servlet request for the interaction and render phases
 - ▶ Not even for the first rendering after an interaction
 - ▶ True for local portlets too
- Decouple rendering from interaction processing
 - ▶ Portlet should be able to render itself as many times as needed without depending on the user's interaction directly.
- Solution
 - ▶ Store interaction changes for future rendering
 - WLP stores state automatically for Page Flows and Struts apps
 - Use render parameters for Java Portlets
 - Use HTTP session for backing files

Dependencies Across Portlets



- **Scenarios**

- ▶ When a user submits an order in Collect Order Portlet, update the Order Status Portlet with the status of the new order

- **Result**

- ▶ Portlets become tightly coupled to each other

- **Solution**

- ▶ Use event based inter-portlet communication

Inter-Portlet Communication with Events

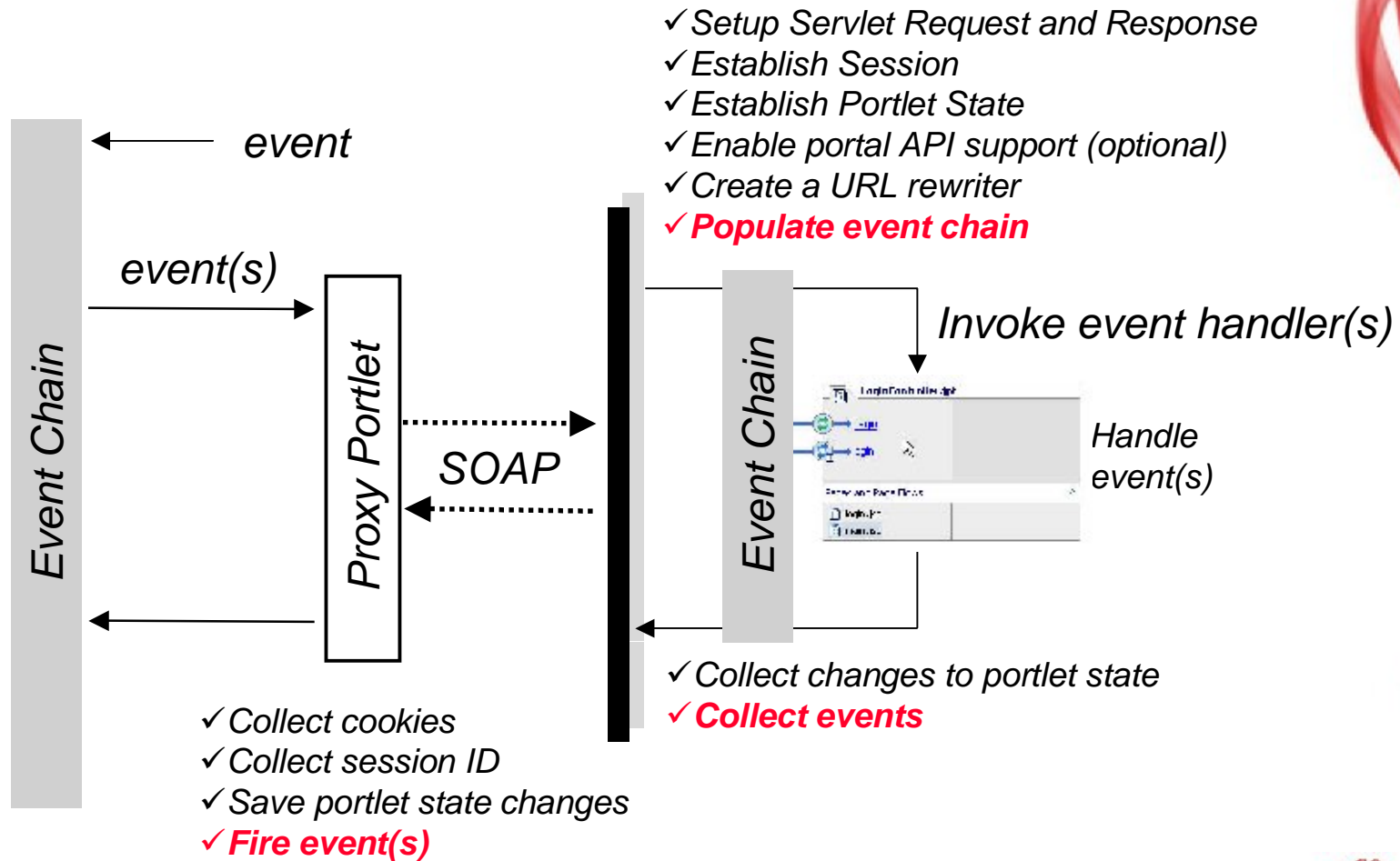
~~When a user submits an order in Collect Order Portlet, update the Order Status Portlet with the status of the new order.~~



When an *OrderPlaced* event occurs, update the Order Status Portlet with the status of the new order.

- Loosely couple with events
 - ▶ Hide dependencies behind an event name and payload

Inter-Portlet Communication with Remote Portlets

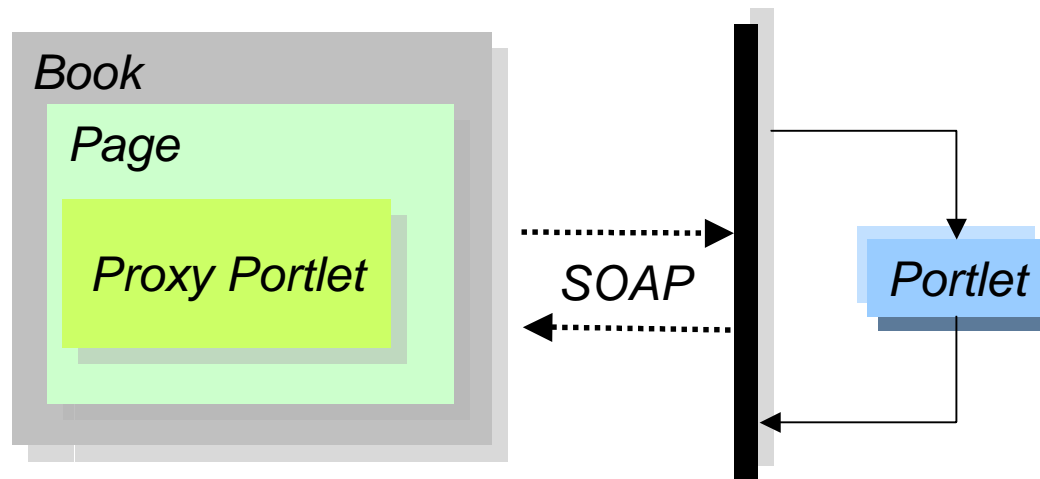


Decoupling Lesson II

- Depending on other portlets
 - ▶ Strongly couples portlets
 - ▶ Difficult to maintain for upgrades, bug fixes etc.
- Solution
 - ▶ Use events
 - ▶ Use event names for dependencies
 - ▶ Avoid using sourceDefinitionLabels on events



Depending on the Portal Layout



- Use cases

- ▶ Displaying a list of portlets on a page
- ▶ Finding whether another portlet is included in a page or book

- Problem

- ▶ Producers are decoupled from the Consumer (portal)
- ▶ Producers don't know about pages and books on the Consumer

Decoupling Lesson III

- Depending on the portal, books, and pages
 - ▶ Strong coupling
- Solution
 - ▶ Avoid when possible
 - ▶ When required, deploy those portlets locally on the consumer

Coupling by URLs

```
String url = "http://mydomain.com/portal/portal.portal?";  
url = url + "myportlet_actionOverride=login";  
url = url + "...";
```

```
  
<a href="/docServlet?docId=9999">Download</a>
```

- Reverse-engineering URLs
 - ▶ Links to Page Flow actions, images, files etc.
 - ▶ Directly adding parameters to URL strings
 - ▶ Getting a Page Flow action name from outer request
- Problem
 - ▶ Links don't work when federated
 - ▶ Remote portlets don't follow the same URL structure

Decoupling Lesson IV

- Depending on internal structure of URLs
 - ▶ URLs fail when federated
- Solution
 - ▶ Let WebLogic Portal create URLs
 - Use netui tags
 - Use struts tags
 - Use render tags or GenericURL
 - ▶ These tags go through URL writers

Summary of Best Practices

Decouple rendering from interactions (interaction free rendering)	Better refreshability
Don't assume that the same request is used across portlet life cycle	Portlets well-behave when federated
Use events to avoid explicit dependencies between portlets	Portlets can be federated, and be swapped
Don't depend on portal framework constructs like books, pages, desktops etc.	Portlets can be consumed remotely
Let WebLogic Portal create URLs	URLs generated to account for federation



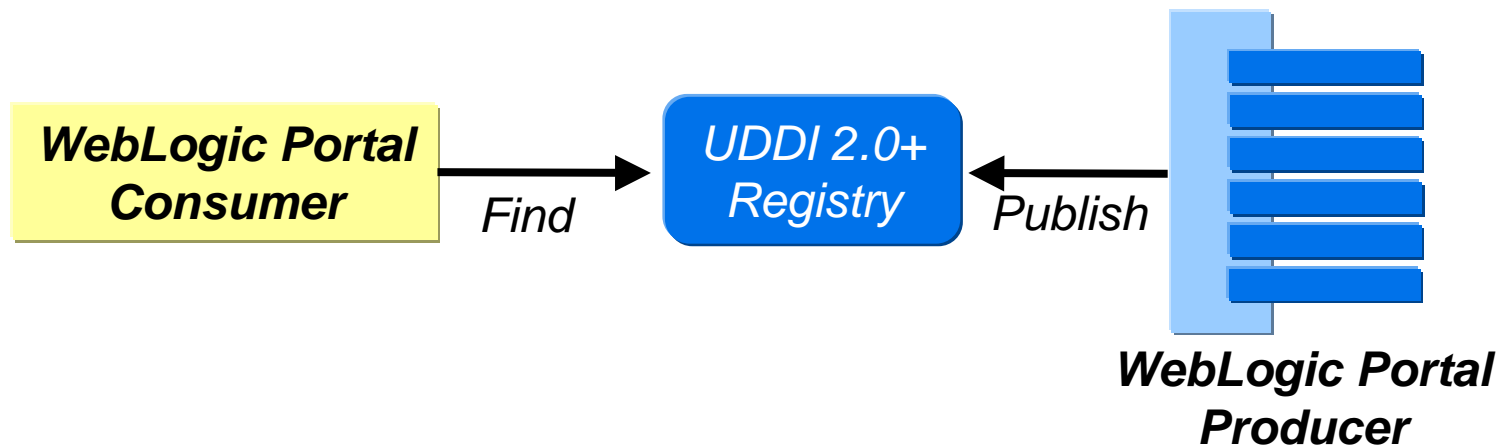
Upcoming Features



Security Enhancements

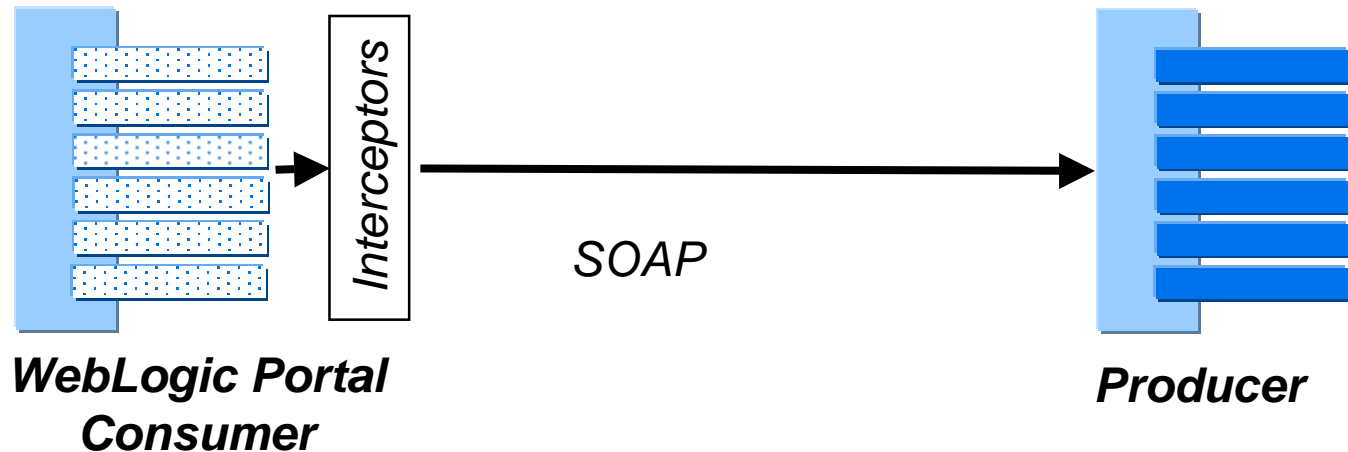
- Standards based
 - ▶ WS-Security 1.0, SAML 1.1, WS-Policy
- Uses WSDL policy attachments
- User identity
 - ▶ SAML Token Profile – identity based on assertions
 - ▶ Username Token Profile – identity based on credentials -
- Message security
 - ▶ Signatures
 - ▶ Encryption
- Pluggable and open

Portlet Registry



- ▶ Publish producers and portlets in UDDI registries
 - Automatic publishing
 - Multiple Producers can publish to a centralized registry
- ▶ Standards based
- ▶ Simplify tasks for users
 - No need to know WSDL URLs
 - Use keyword search to find producers and portlets

Interceptors



- ▶ Intercept requests and responses
- ▶ Influence the behavior of the consumer
- ▶ Dynamically add, swap, remove interceptors
- ▶ Numerous applications – error/fault recovery, caching, security, auditing, markup filtering, etc.

References

- WebLogic Portal

- ▶ Inside WSRP

http://dev2dev.bea.com/pub/a/2005/03/inside_wsrp.html

- ▶ Product docs - <http://e-docs.bea.com/wlp/docs81/wsrp/index.html>

- WSRP 1.0

- ▶ Specification - [http://www.oasis-](http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf)

[open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf](http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf)

- ▶ Primer - <http://www.oasis-open.org/committees/download.php/11177>

BEAWorld2005



Q&A

Subbu Allamaraju
subbu_at_bea_dot_com

BEAWorld2005