

WSRP Consumer Architecture

Table of contents

1 Protocol Handler.....	2
2 WSRP Object Model	2
3 PortletDriver	2
4 WSRP Stubs	2
5 ProducerRegistry	2
6 SessionHandler	2
7 URLHandler	3
8 PortletRegistry	3
9 UserRegistry	3

The Producer and Consumer are provided with a very modular architecture enabling an easy exchange of module implementations. All modules excel by interfaces based on the WSRP object model hiding the portal's object model and thus gaining independence of changes in the portal implementation or design.

Consumer Architecture

1. Protocol Handler

The Protocol Handler is a standalone Swing based application that implements the consuming portal and the browser functionality. It aggregates the integrated WSRP portlets and forwards all invocations together with relevant context and request information to the remote WSRP service. The SwingConsumer thereby uses the ConsumerEnvironment to collect all data required for a WSRP call.

2. WSRP Object Model

Most of the WSRP object model is being generated from the WSRP specification's WSDL types.

3. PortletDriver

The PortletDriver is the task oriented abstraction of the generated WSRP stubs for markup and action invocation.

4. WSRP Stubs

The stubs generated by the axis SOAP implementation. The stubs are also implementing the cookie handling - for more details see SessionHandler below.

5. ProducerRegistry

The ProducerRegistry stores and manages details about producer portals from which portlets were integrated or shall be integrated.

6. SessionHandler

The WSRP consumer implementation session handling is based on the SOAP stack's session handling. The generated SOAP stubs are doing the cookie handling and are caching the cookies in instance variables. This means that one stub object instance is equivalent to one HTTP session which we map to one WSRP session (initCookie wise). When a WSRP session

did time out an InvalidCookie exception is being thrown and a new session must be established.

7. URLHandler

The URLHandler is responsible for the URL rewriting of the WSRP URLs embedded in the markup received from the remote portlet.

8. PortletRegistry

The PortletRegistry stores and provides access to WSRP specific data and descriptions of a remote Portlet.

9. UserRegistry

The UserRegistry is in the Swing consumer case very simplistic. As the browser is integrated there is always only one "dummy" user being managed.