

Methodology and Technology Services

| [Home](#) | [Courses](#) | [Certification](#) | [Projects](#) | [Papers](#) | [Online Store](#) | [Contact Us](#) |

THE ENTERPRISE NEWSLETTER

Issue No 22:

WEB SERVICES FOR REMOTE PORTALS

[Printable PDF Version](#)

CONTENTS

- [Web Services for Remote Portals](#)
- [Previous Issues of TEN](#)
- [Glossary of Terms](#)
- [New Subscribers to TEN](#)

PERTH, AUSTRALIA: June 30, 2003: Earlier issues of TEN have separately addressed concepts of Enterprise Portals and also of Web Services. This issue bridges both topics and discusses the use of Web Services within Enterprise Portals through specification of Web Services for Remote Portals (WSRP).

For the following material I have drawn on the WSRP Specifications and also from the "WSRP Overview" presentation on the OASIS web site at <http://www.xml.org/>, as published by the OASIS WSRP Technical Committee. Their excellent work with WSRP is acknowledged here. With their permission, a number of figures from that presentation are included in this issue. Please note that these figures are sourced from OASIS and are all "Copyright 2002-2003 OASIS. All Rights Reserved."

Clive Finkelstein
TEN - The Enterprise Newsletter

WEB SERVICES FOR REMOTE PORTALS (WSRP)

[Home](#)
[Courses](#)
[Certification](#)
[Projects](#)
[Papers](#)
[TEN Archive](#)
[Contact Us](#)
[Search](#)
[Links](#)
[Online Store](#)

Many content providers publish their content live on the Internet using HTTP or FTP servers, or they provide client software that replicates and caches content via proprietary protocols. In each case, integrating content into an Enterprise Portal is a difficult task. While portals provide some portlets supporting particular content providers out-of-the-box, it is often necessary to develop and install additional portlets for the remaining content providers.

The organization that runs the portal spends much money and effort to integrate a rich set of content from different providers. This is not only a bad situation for portal owners, but also for content providers: it is relatively hard to include content in portals, and this limits business growth. It also limits the capability to exert some control over the way that content is rendered by the subscriber's portal.

With the definition and development of interoperable Web Services for Remote Portals, WSRP offers the promise of easily accessible Web Services to provide access to any data source, for easy incorporation in any Enterprise Portal product.

Web Services for Remote Portals (WSRP) is a standard specification for interactive, user-facing Web Services that are designed to be plug-and-play adapters for portals.

WSRP defines a WSDL interface description for invocation of WSRP services. It defines how to Publish, Find and Bind WSRP services and Metadata. WSRP specifies rules for using markup that is emitted by WSRP services, along with applicable security mechanisms, billing information and so on.

Many companies are involved in the definition of WSRP and development of portal adapters using WSRP. These include the following: BEA; Bowstreet; Divine; Epicentric; Factiva; France Telecom; Fujitsu; HP; IBM; Interwoven; Lexis-Nexis; Lotus; Moravia IT; Netegrity; Oracle; Peoplesoft; Plumtree; Silverstream; Stellent; SUN; Sybase; Tibco; WebCollage; SAP Portals; SeeBeyond.

To allow for easy integration of their content in portals, content providers can use WSRP to surface the content as remote portlets: publishing them as WSRP services in the public, global UDDI directory.

To provide this value-add to subscribers, the content provider serves remote portlets via the desired bindings in addition to the classical content server. Once the content provider has published a WSRP service in the UDDI registry, administrators of portals who wish to use content from the content provider can look up the content provider's business entry in the UDDI registry and bind to the WSRP service that offers the desired content. Portlets on the content provider's server are available without any programming or installation effort, and can be used immediately by the portal users as shown by Figure 1.

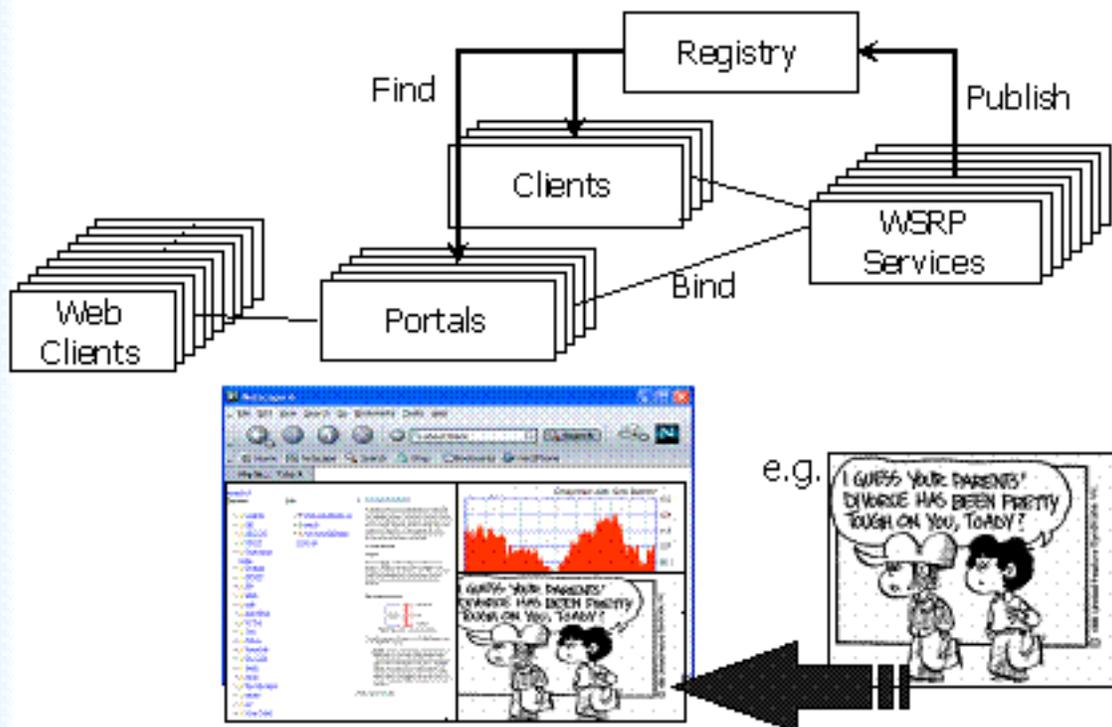


Figure 1: WSRP used for Plug-and-Play Portal Adapters (Source: OASIS)

Portal administrators can find and integrate the WSRP services they need with just a few mouse clicks. They can use their portal's administration facility to browse a registry for WSRP services and select them for automatic integration into the portal. This plug-and-play approach enables content providers to write one interface to their source content for Web Services and then describe that Web Service using WSDL and WSRP. Similarly portal vendors only have to develop one generic adapter that is designed to recognize WSRP and WSDL, then incorporate that generic adapter into their portal product. The goals that have been defined for WSRP so this can be achieved are to:

- Enable interactive, user-facing Web Services to be easily plugged into standards-compliant portals
- Let anybody create and publish content and applications as Web Services
- Let administrators browse directories for WSRP services to plug into their portals without programming effort
- Let portals publish portlets so that they can be consumed by other portals without programming

Make the Internet a marketplace of visual Web Services, ready to be integrated into portals



Figure 2: WSRP Allows Portlets to be Dynamically Added (Source: OASIS)

Figure 2 shows how WSRP adapters can be provided by many content providers as WSRP Producers. A portal vendor extracts from the WSRP markup fragments the information that it needs to incorporate the adapter into the Enterprise Portal. Portals can aggregate presentation from many WSRP services. Additionally the WSRP services can also be aware of portal context, such as the user's profile, desired locale and markup-type, and the user's device type from the portal.

Figure 3 shows a WSRP portlet that has been selected. Its WSRP wrapper describes the source content in a standard way, so the portal can incorporate that new adapter into the Enterprise Portal.



Figure 3: WSRP Allows Portlets to be Dynamically Added (Source: OASIS)

Portals are intermediaries between end users and WSRP services. Figure 3 shows how WSRP aggregates services from different content providers as follows:

1. The portal administrator of the content provider first publishes a portlet as a WSRP service to UDDI using the content provider's administration user interface.
2. The administrator of another Enterprise Portal finds a WSRP service using the portal's UDDI browser, and binds to it with a few mouse-clicks.
3. Users of this second portal can now select remote portlets like any local portlet and add them to their tailored portal pages.

4. The portal providing the portlet as a WSRP service, and the portal consuming the portlet, both adhere to the WSRP protocol and contracts just like any other WSRP service.

Portals offload significant traffic from content providers by caching content, thereby enabling content providers to serve a huge number of users with little IT infrastructure. By providing WSRP services, content and application providers can leverage portals as multiplying intermediaries to reach end users that they could not reach otherwise.

While portals traditionally have operated in isolation from each other, large corporations are now demanding cooperation between their portal installations. Very soon, Enterprise Portals will also need to cooperate with supplier or customer portals, so ultimately portals will need to cooperate over the Internet as well as within intranets.

Earlier we discussed a scenario where an employee portal consumes a service provided by the Human Resources function within a corporation. Now let us assume the Human Resources Department runs a portal that provides various HR related portlets. Some are only intended for use by HR staff like a Payroll Portlet or a Staff Record Portlet. However, there are other portlets that are of interest to all employees. For example, a Variable Pay Portlet calculates the variable pay based on current revenue and an HR Information Portlet providing HR related news.

Assuming that the enterprise has a corporate registry only accessible from the intranet, an HR portal administrator uses the portal publish function to create remote portlet Web Service entries for the Variable Pay Portlet and the HR Information Portlet in the corporate registry. Thus these portlets become available for integration in other portals in the corporation. For example, the administrator of the corporation's employee portal can find the remote portlets Web Services published by the HR portal using the portal's built-in registry browser, and integrate them into the portal with a single click.



Figure 4: WSRP enables Web Services in Applications (Source: OASIS)

Figure 4 shows that applications can also use WSRP Services with plug-in mechanisms such as standard SOAP and WSDL Web Services, or by using COM Components or ActiveX Controls. In this case, the client application plug-in adheres to WSRP protocol and contracts, just like a portal does.

It is expected that WSRP will lead to a more flexible market for Web Services. Much of the complexity in adding Web Services to portals or to applications is removed by the use of WSRP. Web Services will typically be charged on an annual or monthly subscription, and instead on a per-use basis. Some Web Services may only be needed occasionally. Figure 5 illustrates that Web Services can be added and removed dynamically using WSRP.

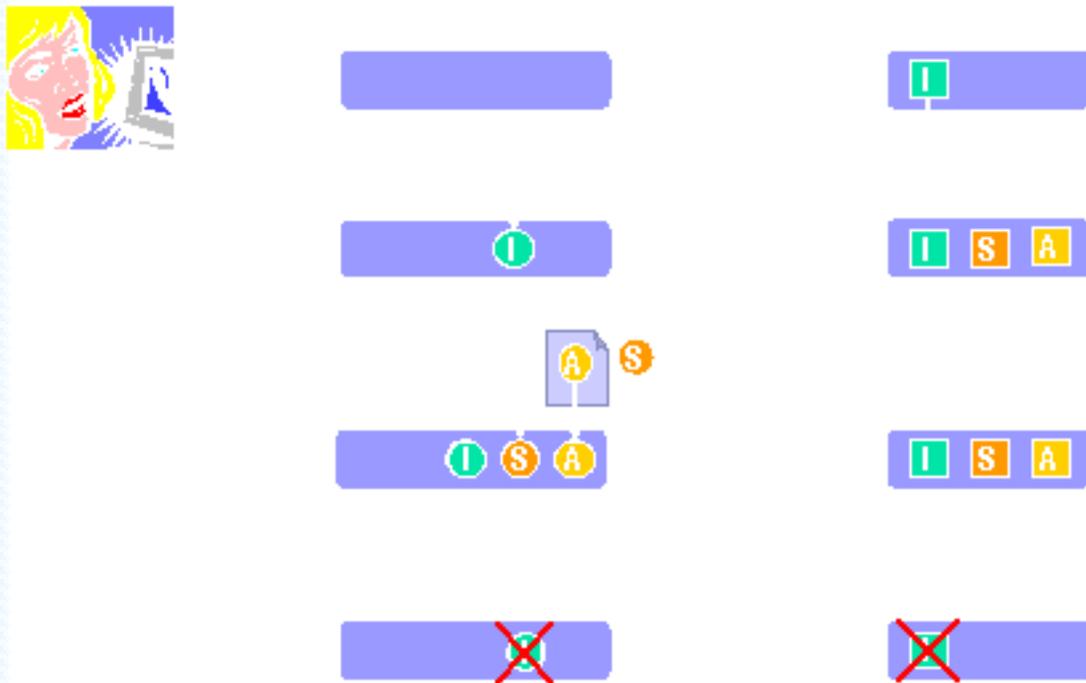


Figure 5: WSRP enables Web Services to be Added and Removed Dynamically (Source: OASIS)

When an end user dynamically adds a portlet to a page in the portal, the portal invokes the WSRP service. This specifies a new Portlet Instance (shown as "I" in Figure 5) that allocates a newly created portlet instance on the portal side.

When the user views the page containing the new portlet instance, the portal gets the portlet WSRP markup defining the fragment that is to be displayed. The returned markup may contain portlet action links "A" and/or a portlet session identifier "S" if the WSRP service wants to maintain the session state: to store details about the session for later reference. The portal may need to rewrite some action links to make them work in the final markup sent to the browser, and must store any returned session details to use for later reference with each subsequent request.

When the user clicks on a link in the markup, a request is sent from the browser to the portal. The portal intercepts the request to map it to the invocation of the WSRP service. It passes the session identifier to allow the WSRP service to look up the previously stored session details. The WSRP service then can restore those details and so change state back to what it was when the session was previously active.

When the Web Service has completed its operation, the portal refreshes the page and a new user-interaction cycle can start again. Alternatively, when an end user does not need a portlet instance anymore Figure 5 shows that it can be discarded from the portal page. The portal identifies the WSRP Web Service that is not needed and destroys the portlet instance. All related resources are released.

Figure 6 shows that WSRP fits into the context of the Web Services standards. It uses WSDL to formally describe the WSRP service interfaces, SOAP can be used for invocations of WSRP services. Furthermore, Figure 6 shows that WSRP has overlap with Web Services for Interactive Applications (WSIA) with which it shares a common base of interface and protocol definitions.

Figure 6: WSRP and Related XML Standards (Source: OASIS)

WSRP defines the notion of valid fragments of markup based on the existing markup languages such as HTML, XHTML, VoiceXML, cHTML, etc. For markup languages that support style definitions, WSRP also defines a set of standard style names to allow portlets to generate markup using styles that are provided by WSRP-compliant portals so that their markup fits nicely into the look and feel of the consuming portal.

Figure 7 is an example of a high-level portal architecture that may be employed for combined use of local and remote portlets, as well as making local portlets available via WSRP for use by other portals.

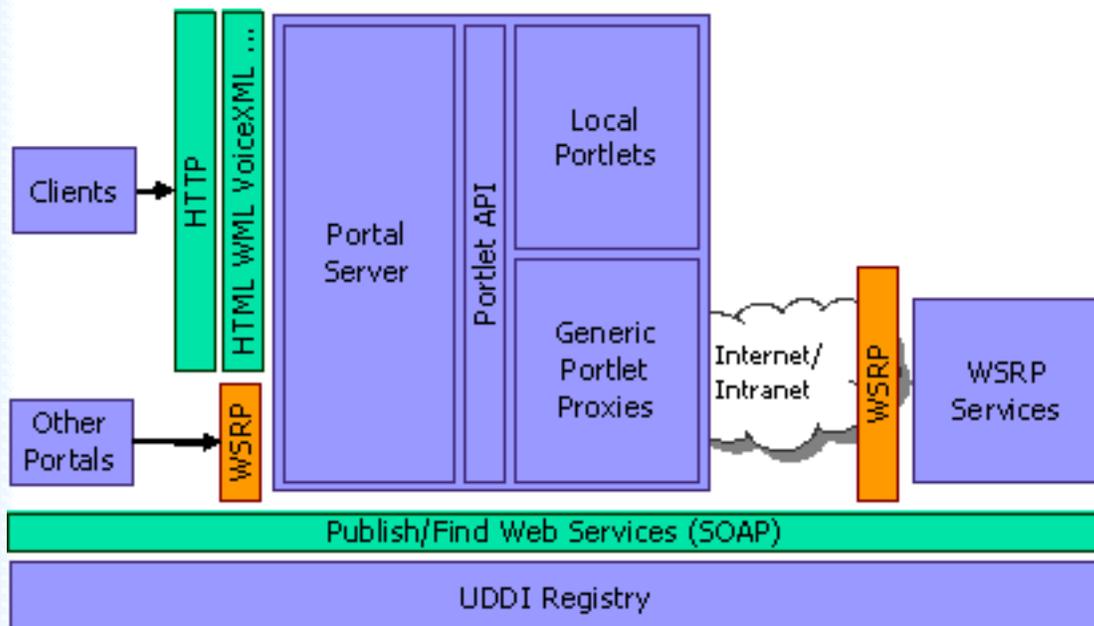


Figure 7: Enterprise Portal Architecture using WSRP (Source: OASIS)

Most portal clients access the portal via the HTTP protocol, either directly or through appropriate gateways like WAP phones or voice gateways. The mark-up languages used by these devices may be very different. WAP phones typically use WML, iMode phones use cHTML, voice browsers mostly use VoiceXML while the well-known PC web browsers use HTML.

When aggregating pages for end users as shown in Figure 8, the portal invokes all portlets that belong to a user's page through a local Portlet API. This may be a portal-vendor-specific API, or via WSRP or its Java counterpart: the Portlet API defined in JSR 168. This is a competing approach to WSRP. WSRP has been developed and refined through OASIS, while JSR 168 has been developed largely by SUN and submitted to the W3C. Further details are available from <http://www.w3c.org/>.

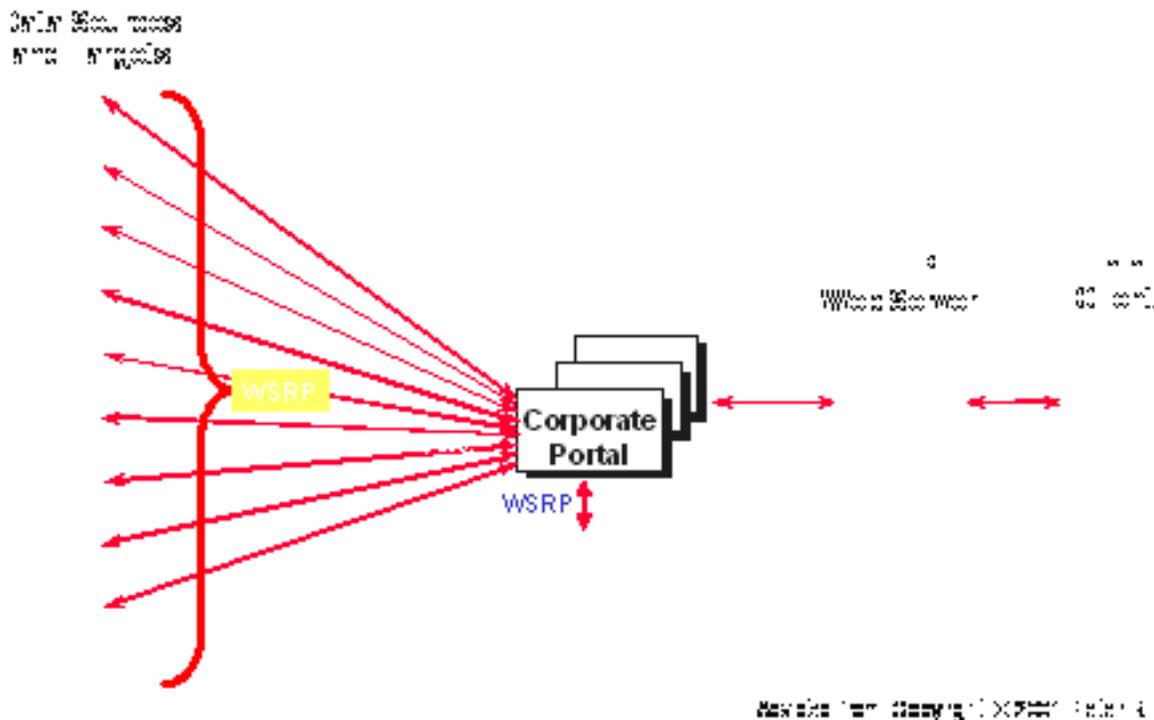


Figure 8: Enterprise Portal Adapters using WSRP (Source: Peter Mimno)

While local portlets can be expected to provide a large part of the base functionality for portals, the remote portlet concept allows dynamic binding of a variety of remote portlet Web Services without any installation effort or code running locally on the portal server. Also, portals may wrap local portlets and publish them as remote portlet Web Services for integration by other portals. Conversely also, remote portlet Web Services can be integrated into portals by wrapping them in a proxy written to the local portlet API.

So we see that Web Services can be used by Enterprise Portal vendors to provide ready access in real-time to related data sources, without having to build specific Gadgets or Portlets written for each data source format. This expands the access to different data sources that are available to the Enterprise Portal product. The Single Sign-On capability of the Enterprise Portal also provides central access control by Web Services to different data sources, with more effective and potentially more powerful central security control.

However, the greatest benefit that the use of Web Services offer over EAI in this situation is that real-time access to Web Services enables data - accessed from the Enterprise Portal - to also be changed in real-time to reflect relevant updates. This opens the opportunity of the Enterprise Portal used for initiation of application processing that updates relevant data sources with appropriate control and in real-time. Obviously, the performance implications and other control implications of this real-time update also must be taken into account.

In summary, Enterprise Portals benefit from WSRP Web Services as discussed next:

- Personalized, real-time access through Web Services to both structured and unstructured sources.
- Automatic content population from existing applications and ERP applications in real-time via Web Services.
- Content is dynamically presented based on the user's role, tasks, and security authorization.
- Dynamic execution, viewing and direct access to relevant information - with real-time update via Web Services, if appropriate.
- Integration with back-end content sources, such as SAP R/3, PeopleSoft, other ERP, CRM, and BI applications - via Web Services support in these products.
- Administrators can define access rights by group or role.
- Platform independence, multiple servers and load balancing.
- Support for alerts, exceptions, notifications, and subscriptions over the Internet, Intranet and Extranet.
- Single-Sign-On Authentication Server, LDAP support, Domain Server support, and Trusted User support.
- Security at the level of information objects, with external authentication using customizable drivers (LDAP, NT, or other standards).

Back to [Contents](#).

AUTHOR

Clive Finkelstein is the "Father" of Information Engineering (IE), developed by him from 1976. He is an International Consultant and Instructor, and Managing Director of Information Engineering Services Pty Ltd (IES) in Australia.

Clive Finkelstein's books, online interviews, courses and details are available at <http://www.ies.aust.com/cbfindex.htm>.

For More Information, Contact:

Clive Finkelstein, Managing Director
Information Engineering Services Pty Ltd
PO Box 246, Hillarys, Perth WA 6923 Australia

Details: <http://www.ies.aust.com/cbfindex.htm>

Web Site: <http://www.ies.aust.com/>

Phone: +61-8-9402-8300

Fax: +61-8-9402-8322

Email: cfink@ies.aust.com

| [Home](#) | [Courses](#) | [Certification](#) | [Projects](#) | [Papers](#) | [TEN Archive](#) |
| [Online Store](#) | [Contact Us](#) | [Search](#) |

(c) Copyright 2004-2006 Information Engineering Services Pty Ltd. All Rights Reserved.