

SEPT/OCT 2004

WLDJTM

WWW.WLDJ.COM

**THE LEADING
INDEPENDENT
MAGAZINE
FOR WEBLOGICTM
PROFESSIONALS**

SOA-Based Applications

TIME TO GET ON
THE BUS ...6

PLUS...

Canonical Message Formats ...10

So You Want an SOA with
Web Services ...16

Service on Demand Portals ...22

Server Configuration ...26

A Service-Oriented Management
Approach for Service-Oriented
Architecture ...34



\$9.99US \$9.99CAN



TRANSACTIONS:
*Transactions for the
Next Generation*

page 32

PLATFORM:
*A Real-World Business
Process Model PART 2*

page 40

INTEGRATION:
*Building Business
Processes, PART 2*

page 46



PORTALS

Service on Demand Portals

A PRIMER ON FEDERATED PORTALS



BY RAJUL RANA & SAI KUMAR

AUTHOR BIOS...

Rajul Rana is a senior architect with Mphasis Corporation (www.mphasis.com), a global IT services organization. Rajul leads the portal practice at Mphasis, and has architected several large, successful e-business and portal projects for Fortune 500 companies, built on a variety of products. His areas of interest span J2EE technologies, portal, content management, and SOA.

Dr. Sai Kumar is a senior architect with Mphasis Corporation, where he heads the Web services and SOA practice. He has provided strategic consulting in SOA to various companies and has architected several enterprise solutions for financial, health care, and retail industries.

CONTACT...

rajul.rana@mphasis.com
saikumar@mphasis.com

Enterprises are moving towards a highly collaborative environment to achieve higher competitive advantage.

Availability of the right information across the enterprise at the right time has become a key capability to provide such an advantage. Though this was a well-understood objective, various architectures that evolved to manifest such an enterprise information delivery infrastructure were not elegant, intuitive, or aligned to governance and organizational dynamics.

With the emergence of service-oriented architecture (SOA) and Web Service for Remote Portals (WSRP), the enterprise information bus (EIB) topology has evolved to be the infrastructure for service on demand portals.

This article, which is the first of two, provides a comprehensive introduction to federated portals, WSRP, and EIB.

Federated Portals

A few years back, in a spree to go "online," organizations started building individual Web sites that evolved into siloed portals. Time to market, inadequate product maturity, and budgetary constraints are just a few reasons why organizations find themselves with a large number of siloed portals.

Today, enterprises are becoming more complex, distributed with increasing lines of businesses, business processes, and siloed portals, while at the same time having the need to portray a unified face, in the form of an enterprise portal, to their customers. Though this means consolidation, they still want to protect their investments in existing portal-based applications and leverage them.

The consolidation of different portals, information islands, and business processes is not only a challenge, it is also simply unmanageable. It is practically impossible to collate information and data from different lines of business portals into a centralized enterprise portal for a variety of reasons. Rationalization and consolidation of a multitude of portals into a true enterprise portal, which is a common platform for delivery of user-interactive applications, is a task that remains high, but elusive on the enterprise wish list today.

The challenge of realizing a true enterprise portal becomes manifest when it is important to consider the governance, information ownership models, and trust within an enterprise for information life-cycle management.

Yesterday's portal proliferation and the need to leverage that, combined with today's need for a unified enterprise face, poses unique challenges in creating an information delivery framework for large enterprises. The solution lies in fabricating a highly distributed yet unified portal framework that can accomplish:

- Standards based information interoperability
- Loose coupling with manageable integration

The availability of the right technology and standards is important to realize truly distributed portal information domains and yet provide a seamless integration and delivery of information across enterprise.

Recent Standards in the Portal Space

Two complementary industry standards that are emerging in the portal space are:

- **JSR 168:** An industry standard that defines a standard way to develop portlets. It allows portlets to be interoperable across portal ven-



dors; e.g., portlets developed for, let's say, BEA WebLogic Portal, can be interoperable with IBM Portal. This will allow organizations to have a lower dependency on the portal product vendor.

- **WSRP (Web Service for Remote Portals):** Allows remote portlets to be developed and consumed in a standard manner and will facilitate federated portals.

JSR 168 complements WSRP by dealing with local rather than distributed portlets. As shown in Figure 1, a portal page may have certain local portlets which are JSR 168 compliant and some remote, distributed portlets that are executed in a remote container.

With JSR 168 and Web Service for Remote Portals (WSRP) maturing, the possibility of true enterprise service on demand portals has become a reality.

WSRP has combined the power of Web services and portal technologies and is fast becoming the major enabling technology for distributed portals in an enterprise. WSRP will lead to a new era that will allow enterprises to provide one face to the users resulting in consistent user experience, with unified information delivery. At the same time, WSRP will allow all various applications to remain distributed.

Introducing WSRP

Briefly, WSRP is a standard that enables a "portal based" Web application to easily consume services from any number of distinct providers on behalf of its end users (portal users), and to present this information to them with minimum integration effort. It allows dynamic binding to remote portlets without any installation or code running locally on the portal server

WSRP offers the following benefits:

- **Reusable Presentation Tier:** Presentation delivered with Web service, not just data
- **Interoperability:** Widely accepted standard supported by large players in the industry
- **Portability:** Can potentially be used with both J2EE and .NET platforms (although Microsoft has yet to formally announce their WSRP implementation).

WSRP builds upon the original Web services vision and uses the same underlying technologies, such as SOAP, WSDL, and UDDI. WSRP services can be published, found, and bound in a standard way. The main difference between WSRP and Web services is:

- Web services are data oriented, whereas WSRP is presentation oriented and interactive by nature.
- Web services return "raw" XML-based data, often quite complex, which the application has to transform into the HTML pages and additionally maintain state in a multipage (multistep) interaction. WSRP returns ready-to-use HTML markup fragments that the consumers can embed directly, without any further processing.

Consider using WSRP when:

- Integrating user-facing, presentation-oriented and visually rich functionality
- Integrating interactive application with complex flow: Multistep, multipage user interaction
- You want to reuse a user interface and look-and-feel tier
- Host services in a central environment best suited for execution, allowing maintained control over format and presentation of the content.
- Sharing code at the portlets level across different areas within the organization: reimplementation of the presentation layer on each portal is avoided.

Federated Portal Architecture

Let's consider that an organization has several existing departmental portals. Each of them could be on different technology and products such as Vignette Portal, WebLogic Portal, or even .NET (SharePoint) technology. Some of them may not even be on a portal product. Each portal may or

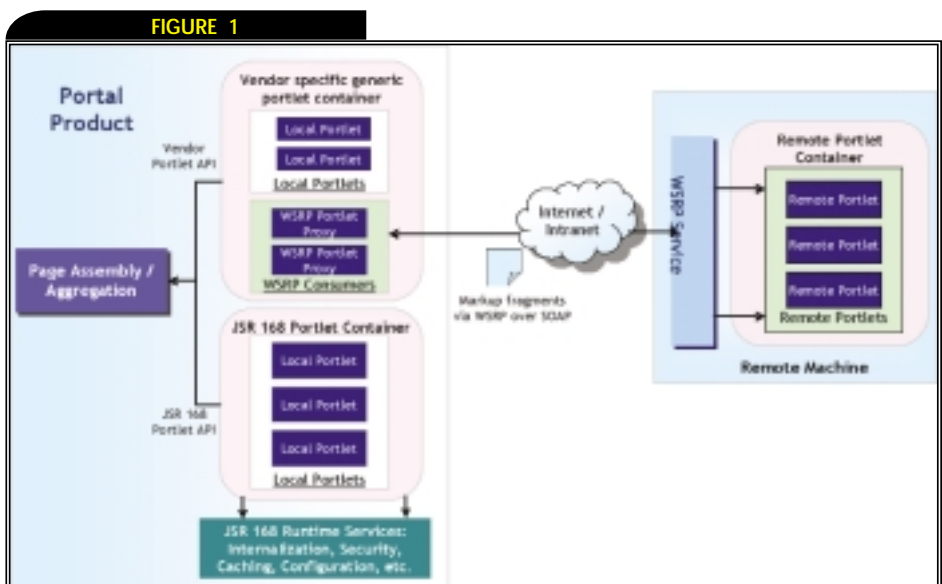
may not have same look and feel or navigation, but all departmental portals participate under a central SSO umbrella. Each departmental portal is self-contained and provides unique functionality specific to its product or services.

Now, let's consider that the organization wants to provide a centralized face to the users, instead of users having to access a multitude of applications. There are main two architectural options:

1. **Monolithic enterprise portal model:** Migrate the existing applications and create one large main portal that has all the features. This is practically impossible, as it is costly and time-consuming. Also, there could be a technology limitation in migrating heterogeneous applications to a common technology platform.
2. **Federated portal model:** Create a main portal that serves as an entry point or gateway to the enterprise. The main portal is a thin layer that sits on top of the individual departmental portals and leverages them.

A federated portal allows organizations to provide a common entry point, but, at the same time, independence, to individual departments to develop, maintain, control the release schedules, etc. Initial implementation of federated portals integrated the main portal with departmental applications or portals using traditional presentation layer-based integration techniques, such as:

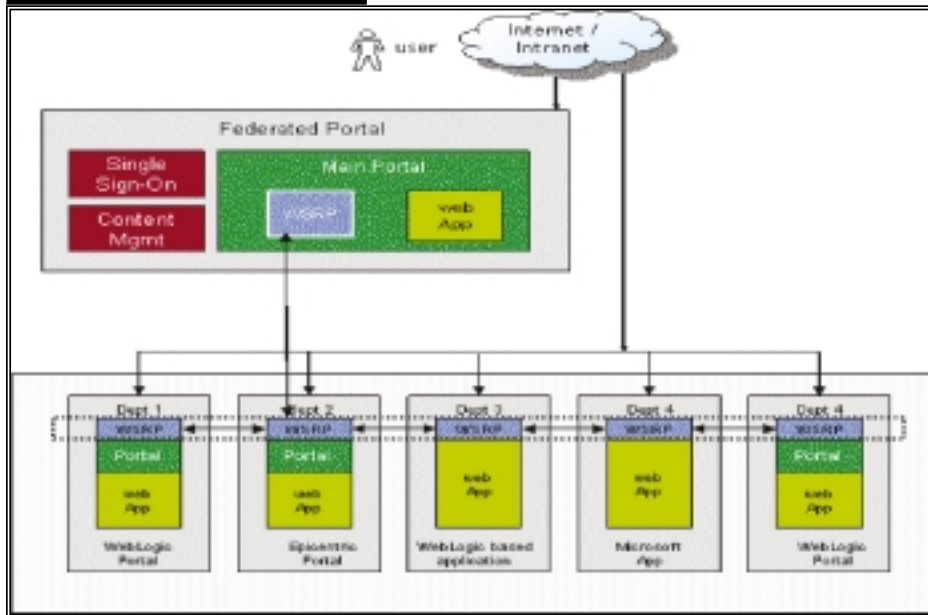
- **New browser window:** A link can be provided on the main portal and, on click-



Portal page with JSR-168-compliant portlets



FIGURE 2



Federated portal architecture

ing to this link, the target application can be opened up in a new window.

- **Frame-based integration:** Target application can be integrated as an HTML frame into the portal.
- **Screen scraping:** In this case, applications can be integrated into the portal by screen scraping the target Web application.

These techniques, however, present the following challenges and limitations:

- **Application integration issues**
 - Session Management issues (timeout synchronization, etc.)
 - Propagating logoff requests to all active applications
 - Security around application launch (as departmental applications need to be directly accessible from the DMZ)
 - Window management (closing of windows user logs off)
- **Management issues**
 - Complicated new on-boarding application process; will have to develop custom application registry.
 - Each application will be required to have its own set of Web servers (and DMZ) and application servers.
 - Cost, administration, and version issues
 - Each Web server is required to have an agent to participate in an SSO environment.
- **Over time, managing (version upgrade, installation, etc.) a large number of agents can be problematic.**

WSRP presents a more elegant, service-oriented approach to implementing federated portals. Service-oriented architecture provides flexibility and reuse of core service components. Service components traditionally are reusable business and infrastructure components in nature. A truly federated portal needs to integrate with transactional services and information across the enterprise, provided by various specialized portals and service infrastructures. With WSRP as the underlying technology, portlets are designed, built and exposed as reusable Web services on an enterprise information bus (EIB).

WSRP natively supports portlet concepts, such as modes (view, edit, help, preview, and custom), window states (maximized, minimized, solo, custom), portlet preference management, etc. This makes it much easier to integrate applications in a portal paradigm. Further, WSRP addresses some of the common integration issues (many of the above issues), such as transparent session management, timeout handling, caching, support for different markups, etc. It integrates applications as services rather than fragile links. Moreover, since WSRP is built over the SOAP stack, it can leverage enterprise Web service management infrastructure, including service metering, routing, prioritization, and lifecycle manageability, e.g., versioning, monitoring, and upgrade. With services and information exposed as Web services, they

can also be discovered from a UDDI registry and secured using Web service security standards. For transport-level security, WSRP can be used with SSL.

All of these features, put together in an industry-standard manner, make WSRP an ideal technology for building federated portals.

Federated Portal Model Using WSRP

The main portal acts as a facade (see Figure 2), serving up a common logon and home page. It authenticates users against the central Single-Sign-On (SSO) infrastructure to display an entitlement-driven home page. The home page is composed of multiple portlets that display information from various departmental applications. Each portlet acts as a WSRP consumer that interacts with the WSRP producer on the other end. Thus, departmental services are integrated seamlessly into the main portal, using WSRP. To achieve this, the departmental application services are exposed as WSRP services. This requires adding a WSRP layer (specifically WSRP Producer) on top of existing application, which is much easier than rebuilding the application in the main portal.

The responsibility of the main portal (WSRP consumer) is limited to:

- Content/application service assembly.
- Portal (top Level) personalization and customization.
- Authentication.
- High-level entitlement: The main portal controls the access to the portlets on the home page and therefore access to the services offered by the departmental portal.

The departmental portals (WSRP Producers) continue to provide:

- Business logic
- Content rendering
- Portlet-specific customization and personalization
- Entitlement checks: Each producer should also check entitlements for access control, fine-grained entitlement check.

Another level of sharing between multiple portals is to integrate the touch points, such as directory, security, personalization profiles, metadata and portal components, which isn't easy. Issues to consider in this area:

-continued on page 38

capacity, availability, and performance in order for them to be humanly consumable and then to enable automated and “standardized” monitoring across tiers of infrastructure.

Figure 3 is not meant to be “anatomically correct,” but rather to illustrate that a cohesive, end-to-end CAP monitoring strategy is not only possible but also necessary as infrastructure stacks become more complex and dynamic. Like the SOA concept of abstracting fine-grain application functions into coarse-grained services, end-to-end application and infrastructure stack component measurements (Web server, application server, database, etc.) could be abstracted into higher-level measurements of CAP. New service-oriented management systems will leverage these standardized measurements and provide a means to aggregate and correlate them to the services that they provision. Imagine being able to categorize and find a capacity or performance bottleneck down to at least the level of an element or component. How are application server performance measurements impacted by network performance measurements? What impact do they have on the service layer? In my experience, most IT shops do not have this down to a science, but it is possible – and in an SOA-enabled enterprise, where services are providing business differentiation, service quality will be increasingly important.

Bridging the Gap Between Web Services and Application and Infrastructure Management


Having standardized measurements of both Web services and the supporting application stack makes it possible to trigger downstream infrastructure stack analysis and alerting to the measurements of Web service quality, as described earlier. If the Web service layer measurements and monitors detect a performance problem, they can automatically trigger downstream analysis in order to determine like-kind (performance) infrastructure problems that may have been occurring at the time the service degraded. Figure 4, a high-level diagram of an actual project to provide end-to-end proactive monitoring for a BEA WebLogic Integration 8.1 SOA plat-

“The promise of enhanced flexibility, adaptability, and agility in the context of “everything services” will win in the end”

form, depicts a Web services-centric operational management diagram along with a simple SLA-based analysis workflow for correlated problem detection. BEA made it easier than usual by publishing Web services statistics via JMX and also publishing performance statistics of Web services that were organized into business processes via their Workshop product.

Summary

The SOA trend – already pronounced across the industry – will, in my view, only accelerate over the next several years. The promise of enhanced flexibility, adaptability, and agility in the context of “everything services” will win in the end. However, the complete value of SOA will be fully realized only when all parties involved in IT service delivery, and service support of the entire application life cycle, work together with the common goal of designing, coding, testing, deploying, and managing services from the common objectives of business.

This is an exciting time for IT developers and operations. In an SOA world, they are both seated at the head table as trusted advisors to the business and as critical partners for any key revenue-generating or cost-reduction objectives. By taking a unified, service-oriented approach to designing, deploying, and managing business services, they have a wonderful opportunity to get it right. 

Service on Demand Portals

–continued from page 24

- **User experience:** All participating applications (producers) should conform to a common look-and-feel standard. Navigation should also be consistent across them.
- **Single Sign-On:** Since the departmental portals are no longer Web applications, but are services similar to Web services. The WSRP spec does not address security directly but encourages leveraging standards such as WS-Security, SAML, XML Signatures, and XML-Encryption. Consumers should be able to provide SAML assertions to authenticate and propagate user identity to the Producer. BEA WebLogic’s WSRP Identity Assertion Provider, used on the producer end, validates user identity through the SAML assertions.
- **Entitlements:** High-level entitlement-based access to the main portal can be accomplished by setting role-based entitlements on WSRP consumer (proxy) portlets

at the consumer end. This indirectly allows/blocks access to the WSRP producer, provided they are not directly available. It is recommended that the producers also enforce certain authorization checks and not rely completely on the consumer to enforce that. Furthermore, low-level entitlements can be independently enforced by the producer. In order to do this, the producer needs to get the user identity and information via SAML assertions.

- **User Management:** All portal products have their own user repository to store user customizations and personalization profile elements. Ideally, a central user directory should be leveraged as much as possible for user identity and basic profile information.
- **User profile information:** Though defined in the WSRP specs, BEA’s current implementation (WebLogic Portal 8.1 SP3) does not support the passing of user infor-

mation in the initial request. One option is for the Producer to directly access the user repository and fetch the user profile.

- **Sharing content management infrastructure:** Content management (CMS) can be a centrally deployed, shared service that is leveraged by all the portals. Though the CMS is on a centralized infrastructure, individual portals can have their own sandboxes with independent release schedules.

Additionally, with the federated approach, each departmental portal’s services and information can be freely shared and used to support other portals. It allows loose coupling between portals into a federation of shared services where any portal can leverage information from any other portal.

• • •

Next month we’ll look more deeply at federated portals, WSRP, and EIB. 