

Enterprise Portal Computing

Next Steps in Evolution & Key Technology Enablers

Madhu Seshadri

Dec 2003

The information contained in this document represents the views of the author on the topics discussed. This information should not be interpreted as Cognizant's Solution service offering or the company's opinions on the discussed topics. Cognizant or the Author does not guarantee the accuracy of this information.

Table of Contents

Abstract.....	3
Enterprise Portals – Constituent Evolution.....	3
Next Generation Portal – Federated Portals	6
Technology Standards.....	7
WSRP – Web Services for Remote Portlets	7
Identity (Single Sign-On) Federation	8
SAML – Security Assertion Markup Language.....	8
Solutions Market – Off the Shelf System Landscape	9
Perspectives	10
Conclusion	11

Abstract

It would not be an overstatement to say that the Internet Portals like Yahoo, MSN and others have tamed the Internet, so to speak, and have become a sort of personalized content, collaboration and application delivery platform to the consumer. These portals have evolved from being pure gateway interfaces to a more personalized content, application aggregation infrastructure and collaborating media.

Enterprises were quick to see value in these Internet portals and were quick to adopt the same concepts to their enterprise information systems. Over the years, enterprises have built several organization and department level and practice centric portals.

The portal computing trends indicate that portals are no longer looked upon merely as content delivery channels but as e-business application integration platforms to business users, aiding infrastructure consolidation of user facing ebusiness applications. This trend is not without reason. Consolidation would help businesses save costs and enable faster time to launch of new ebusiness systems along with personalized user experience. The term ebusiness or web application is used in this paper to refer to any system that uses browser-based deployments using Internet, Intranet, and Extranet infrastructures.

More than ever, the enterprises are looking at the next steps of portal computing evolution. Composite and Federated portals are concepts that are emerging and gaining industry momentum. The concept of federation demands for standards. As a response, the technology industry has been busy working on several standards and some have gained huge market adoption.

This paper aims at providing portal computing constituent evolution; emerging concepts for next generation portals, discuss key standards that have emerged, their market adoption and perspectives.

Enterprise Portals – Constituent Evolution

Gartner calls the next generation, fourth generation portals as federated portals. Looking back, the portal computing has gone through an immense metamorphism, with each generation adding new concepts and technologies. First generation portals could be called systems that provided interface web pages to web applications in an enterprise. Second generation portals formed ways to publish targeted content and applications. They also adopted personalization concepts to personalize content to the users. Third generation portals evolved as integrated end user platform for launching enterprise applications and have become collaborating media. They also entered the foray of identity management and declarative access web security models for all applications integrated with the portal.

A third generation portal computing solution was a harmonious coexistence of a) Personalized Web Desktop and Application Integration Frameworks, b) Access Security (Single Sign-on) and Identity Management Solutions c) Enterprise Content or Knowledge Management (ECM) & Search Tools and d) Asynchronous Collaboration Tools.

Personalized Web Desktop and Application Integration Frameworks

A personalized system would deliver what the user wanted from the enterprise network to his desktop. Such a system would need to be user centric and controlled by rules of delivery. Industry coined a term “Personalization” to refer to this concept.

Personalization has to be driven by rules. These rules became enterprise policies to control content delivery to the user. Personalization rules are built around the user’s role in an organization. User profile can be defined as a set of attributes that describe his/her relationship with the enterprise.

Portal servers, a class of off-the-shelf systems emerged and provided frameworks to build personalized user web desktops like “My Yahoo”. Most of these systems used directory based user personalization store. These systems allowed setting up administrative rules for personalization at the organization, department and user level. Integration Frameworks provided a way for applications to be integrated and create portal desktops. Web applications were created based on the off-the-shelf proprietary frameworks and used the personalization store to control delivery.

Directories are hierarchical, read optimized databases accessed via network protocols like LDAP. Replication and synchronization technologies allowed scaling to a very large enterprise user base. The hierarchical way of organizing people directories provided a way to do easy delegation of personalization rules allowing individual organizations to control their content.

Identity Management & Single Sign-On Solutions

The cost of managing user identities was becoming higher and higher. Typically, businesses have employees, partners, customers who need access to a variety of the enterprise’s resources, in short, content and applications. Each user identity (user security information to access resources) was separately created and administrated inside each application that the user has access to. Each system used its own identity lifecycle management tools, resulting in distinct data-sources of identities and access control tools to provide access security. Lifecycle of identities tend to change frequently resulting in heavy administrative actions. This resulted in individual maintenance of these applications, even application customization, resulting in huge costs.

Portal became the user platform to enterprise content and applications. Naturally they became a key driver to have single user identity, allowing individual web applications not to worry about lifecycle management of user identities. Declarative security policy administration is part of identity lifecycle management. Single identity management allowed creation of single access security systems for all applications. Access security systems become the policy enforcers and provided single login for all applications. They were commonly referred to as Web Single Sign-On (Web-SSO) solutions.

Managing the lifecycle of partner organizations identities is a daunting and high cost task because of involved help-desk interactions. Web self-service model became an answer to this challenge, resulting in the birth of delegated identity management systems. This allowed departments to delegate the lifecycle management of identities. Examples for application of this delegation model could be dealer identity in case of the automotive industry, agent identity in case of insurance industry, franchises user identity in case of restaurant businesses, and branch user identity in case of retail banking.

Delegated Identity management system had to be driven by organizational hierarchy resulting in them using directory server technology for user stores. Self-service identity management off-the-

shelf system emerged as packaged applications allowing extensive customization and workflow capabilities over an organization user data maintained in a directory store.

For Web-SSO, several technologies became prevalent. Two technologies that became very common were referred as “Agent” and “Reverse Proxy” models.

In an agent model, each web application installs an agent or custom component that would provide access security by validating every user request to any secured web application with a central or distributed authentication and authorization server. Authorizations are enforcements of security policies. An example of this solution would be Oblix Netpoint or Netegrity Siteminder.

In a reverse proxy security model, a proxy server sits between the portal and browser providing security and session management. This idea is similar to the way a large corporate has adapted to allow access from intranets to Internet via the proxies. In this case, server is prox-ied, hence the technology is called reverse proxy. IBM Tivoli Access Manager is example of such model. Proxy server controls authentication and authorization to all requests to secured applications.

Content Management & Search Engines

Portals became the content delivery platform. Enterprise content management off-the-shelf systems addressed these needs of static content like news, articles and bulletins lifecycle management. They became part of the portal solution. Lifecycle management involved creation to publish and archival-purge of the content.

ECM solution provided the following,

1. Secure browser interface for content managers to create web content
2. Workflow engines to create, review, approve and publish content
3. Meta-data provisioning allowing target personalization and search
4. Multi-lingual support for content creation
5. Secure deployment over the Internet across multiple servers
6. Secure meta-data deployment to servers for personalization engines
7. Content Archival tools
8. Single sign-on strategy for the content managers to do their work

Any content management solution is incomplete without ability for the user to search for the content. Internet Search technologies were based web-based crawling engines that crawl Internet sites and create index repositories. A class of off-the-shelf systems addressed the enterprise content search needs with similar technologies. Autonomy and Sun One Portal server search module are examples of these systems.

Asynchronous Collaboration Tools

Portal provides an ideal and immediate platform for providing collaboration tools. Asynchronous collaboration tools could be document management, virtual team rooms and integration to enterprise email and calendaring solutions.

ECM and Asynchronous collaboration tools are types of ‘application’ that needs to be aggregated by the portal. Most of third generation portals had some form of these.

Next Generation Portal – Federated Portals

Outlook and trend prove the business value of portals as application delivery platforms. These days, the portals are looked at as the way any user would do business with the enterprise. In reality, user's businesses don't confine to the way the portals have been built today. In most organizations, portals were built at the departmental levels or for practice levels. In some cases, the user would need to work with other organization portals or even third party external applications.

In order to provide the user a single business platform for all enterprise resources, the user should be able access content across all portals in the enterprise. To achieve this, portals have to be integrated in such a way that content is location-transparent to users without users having to remember numerous URLs and identities for access. Industry analysts are predicting that this would be the next generation evolution for portal computing. Gartner refers to such portals as federated portal. In essence, such federated portals would need to provide federated content, identity and single-sign-on.

Federation is not possible without standards. The industry has been working on several standards to realize this; some have gained huge market adoption in 2003. This paper discusses three key relevant standards JSR 168, WSRP and SAML, enablers for content and security federation. WSRP and SAML rely on several other webservices infrastructure standards like SOAP, WSDL, UDDI, and secure XML messaging standards.

Composite portals could become mere gateway models once content and security federation become reality. Composite models could even create an opportunity for browser delivered rich clients targeted at the type of the user. Since we are in the era of broadband, such technologies could have faster adoption.

Another concept that has to come in vogue is "Vortals". This industry term refers to portals that are practice centric. Some of ERP, CRM off-the-shelf systems have adopted portal computing principles in their solution offerings. These systems are referred as Vortals in this article. Examples of this could be My Siebel or My SAP.

On one side, the need for intra-enterprise portal federation is evident; at the same time, there is also need for inter-enterprise federation. An example of this would be that of a third party 401K, insurance integrated with enterprise user portal providing employee's 401K management tools, insurance details tailored for the user based on the company policies. Project liberty (www.projectliberty.org) is a key standard that has evolved to address the need of inter-enterprise federation. Microsoft has developed a service called Passport addressing this need. (www.passport.net).

Technology Standards

More than ever, the current portal industry is in need of technology standards for security and content integration as portal by itself means the co-existence of a variety of solutions implemented using an array of technologies. Following sections discuss some industry efforts that have become standard and that have gained huge adoption by the off-the-shelf solution markets.

Content (Application) Federation

JSR 168

JSR 168 is outcome of Java community process, an open forum for the development of Java standards, reference implementations and TCKs (Test Compatibility Kit).

JSR 168 is very similar to Servlet specifications, the request-response presentation framework part of J2EE, the java enterprise edition middleware standard. JSR community made a decision to create a new specification for portlets. JSR 168 defines a standard for creating application markups (aggregate-able application interfaces) called portlets. It also defines a run-time environment service for portlets called portlet container and a protocol API (application programming interface) for portlet and portlet container handshake protocol.

Portal container provides lifecycle management for portlets, processes the user requests and generates dynamic content to form the user desktop. The portlet container to portlets is analogous to servlet container to servlets. Portal container leverages servlet technology and can be built on top of the servlet container. Portlet container differs from servlet container by adding presentation constructs with portlets. Presentation constructs enable the container to produce an aggregated user desktop and allows the container to send information to portlets on user actions. Presentation constructs include predefined portlet modes and window states. Since the portlets need to be personalized, container services include a declarative way of providing user information or attributes to the portlets.

JSR 168 provides declarative security model for the portlets but does not favor any single sign-on standards. This specification follows J2EE role based declarative security model for portlets. It also includes standard deployment constructs and conformance toolkit for portlet container services.

This standard achieves several milestones in portal computing. It provides

- a) A standard API for creating portlets or dynamic application content markup for portals
- b) A standard container service model – Container services can be provided by as off-the-shelf system vendors (portal servers), allowing developers to work business applications
- c) No vendor lock-in – portlets could be run in any containers that conforms to this standard
- d) Hot deployment of new portlets and hence no down-time for new application launch
- e) Support for localization and internationalization and
- f) Support for multiple types of client

WSRP – Web Services for Remote Portlets

WSRP is the outcome of joint efforts of two Oasis (www.oasis-open.org, a non-profit ebusiness standards organization) technical committees, WSIA – Webservices for Interactive Applications and WSRP itself. WSRP has been approved as standard in August 2003.

WSRP addresses the issue of location transparency for portlets. WSRP leverages webservices infrastructure for accessing the portlets hosted in remote containers. Location transparency is not

new to the computing world. Several technologies were invented to do this, starting with RPC remote procedure calls. During component Object oriented paradigm revolution, DCOM, RMI, CORBA and EJB were invented to address this. Webservices is becoming the latest in this line with its advantages to work over the Internet.

WSRP defines a producer-consumer-client architecture for getting applications to user desktop. Client refers to the browser that acts as a user desktop. Consumer refers a portal server or aggregation engine that aggregates from several producers to form the user desktop. Producer can be application interfaces (other portals serving portlets or stand-alone applications) that produce markups. WSRP defines producer constructs for service producers; consumption constructs for consumers and a protocol for consumer-producer communication.

The following section provides some more details on the WSRP inner workings.

Portlet state in the consumer is part of transport and hence consumer doesn't have to maintain this state but producers can be mandated to consumer to handle session persistence. Consumer communicates to producer via SOAP (Simple object access protocol). Producer publishes the interfaces services or applications via WSDL (Webservices definition language). Consumer can lookup, register with the producers and invoke their services. Consumer maintains session with client. WSRP defines a Portlet management interface allowing the consumers to customize the portlets and refers them as consumer configured portlets. This allows producers to cater different set of consumers by allowing themselves to be customized by the consumer consuming it. WSRP also address URL rewriting since consumer aggregates content from producers (portlets) that can be remote behind a firewall. An example of this could be third party services provider acting as WSRP producer. In this case, remote portlets may have produced URLs that point back to itself and since they are remote, the consumer (portal) would need to rewrite the client URLs. WSRP delegates the security implementation between the consumer and producer to webservices security standards.

Identity (Single Sign-On) Federation

JSR168 provides standardized local portlet integration to portal desktops. WSRP allows them to be location transparent enabling applications to be shared across portals. Next issue to address is federation of declarative security systems built for portals called as Web-SSO (Web Single Sign-On).

Web-SSO model has to evolve into Federated SSO model solutions allowing user to navigate between different portal domains without having to re-authenticate. SAML exactly addresses this topic of access security inter-operability between different security systems, essentially portals.

SAML – Security Assertion Markup Language

SAML, currently in its version 1.1, specification from Oasis technical committee is an XML framework for exchanging assertions between security systems. Assertions are verification of certain facts about subjects; subject being a user or a system.

Specification has four major components,

- a) assertion and request-response protocol data formats called SAML Core
- b) bindings and profiles called SAML Bind
- c) conformance program for SAML called SAMLConform
- d) security and privacy considerations called SAMLSecure

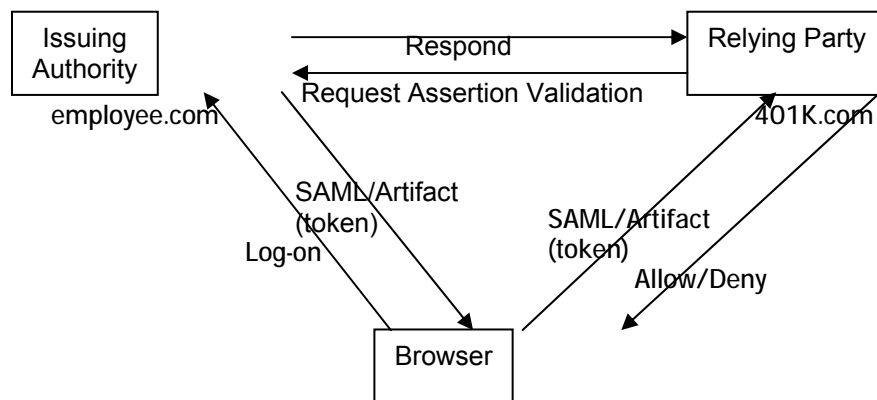
SAML Core covers three types of assertions,

- a) Authentication – Was the subject authenticated by a particular means at a particular time?
- b) Attributes – Is the subject associated with supplied attributes?
- c) Authorization – Can the subject access a specific resource?

SAML Core defines a request-response protocol called SAML protocol between assertion artifact (token) issuing party and the relying party. Relying party asserts with Issuing party about user log-ins, attributes and authorization credentials.

Specification is designed in such a way that this request-response protocol can be used in several scenarios and leverage transport protocols for communication. In SAML terms, scenarios are referred to as SAML Profiles and transport protocols for communication are referred to as SAML Bindings.

SAML Bindings and Profile specification defines a profile for web-browsers single sign-on. It mandates SOAP over http bindings. The SAML browser profile can be represented as follows,



Issuing party issues an SAML token as part of URL to browser and when the user accesses a URL for relying party, the browser posts the SAML token to the relying party. Relying party de-references the issuing party from the SAML token. It acts as an SAML requester, requesting assertions for user authentication validity. Issuing party acts as the SAML responder and provides the relying party with requested assertions allowing the relying party not to re-authenticate the user.

Other two specifications of SAML, Security and Privacy addresses the privacy of subject attributes and various security considerations. Conformance specification provides a toolkit to ensure SAML compliance verification. Since SAML can have different profiles and bindings, conformance defines terminology to express SAML compliant system compliance levels. It also defines test cases for ensuring the actual compliance.

It is important to note that SAML authority is not the authentication authority and it does not provide any PKI solution. Assertion decisions - policies for authentication, attributes and authorizations can be maintained by separate systems. SAML only defines the type of assertions and a protocol between requester and responder. SAML resorts to SSL, TLS, and Digital Signatures for requester-responder authentication, message integrity and confidentiality.

Solutions Market – Off the Shelf System Landscape

Portal market is one of the more fiercely competitive markets. The vendor market ranges from pure desktop framework personalization vendors, security vendors to complete solution sets. A subset of vendors is provided below,

Taxonomy	Vendors
Portal Servers	Oracle, Sun, IBM, BEA, ATG, Plumtree, Epi-centric, Microsoft
Content Management	Vignette, Interwoven
Identity Lifecycle Management Solutions	Oblix, IBM Tivoli, Netegrity
Portal Access Security (SSO)	BEA, IBM Tivoli, Oblix, Sun, SiteMinder, Securant, Evidian, Entrust, Netegrity, RSA Security, Baltimore technologies, Sigaba, Verisign
Pure Search	Autonomy, Inktomi
Collaboration/Document Management	Open Text, Quickplace, Documentum
Vortals	Broadvision, SAP, Siebel etc

Perspectives

JSR 168 will allow application developers to write portlets allowing integration with any portal server providing container services. It will also create a market for portlets. Vortals, ECM will create portlets for easy integration to portals.

JSR 168 is the first but a strong step towards standardizing the way application content could be created for integration with portal desktops. JSR 168 adoption has almost become unequivocal. All major portal players Sun, BEA, Oracle, Plumtree, Vignette have adopted them and provide portlet containers. Vortals would soon adopt them and there could be an influx of portlets written to access legacy information systems. Apache reference implementation for JSR 168 is named as Pluto and can be found at <http://jakarta.apache.org/pluto/>.

WSRP adoption is emerging. IBM has initiated WSRP4J (<http://ws.apache.org/wsrp4j/>) under Apache's ASF. Plumtree announced that it is supporting WSRP standard in its latest release in Sept 2003. Weblogic released a developer preview WSRP kit in Dec 2003. Oracle has a demo site for showing WSRP/JSR168 hosted portlets.

Most of the portal servers also provide remote container services (WSRP) producer and act as (WSRP) consumer allowing cross sharing of portlets between portals.

WSRP and JSR 168 is set to enable Content federation.

Adoption of SAML by identity market is evident as vendors have rush to implement this. IBM Tivoli, Oblix, Sun, SiteMinder, Securant, Evidian, Entrust, Netegrity, RSA Security, Baltimore technologies, Sigaba, Verisign are some to name a few.

SAML browser profile compliant Access Security System is set to address Security Federation

Examining the relations between these standards, JSR 168 does not mandate WSRP but refers it to remote portlets. WSRP does not mandate JSR 168 portlets but JSR 168 fit right as portlet standard in the java world. Since WSRP does not mandate JSR 168, application (markups/portlets) can be aggregated between both .NET and Java. WSRP not directly need or mandate SAML, but the need for SAML arises in the case of federated single sign-on. SAML does mandate SOAP over HTTP in its implementation.

Taking a closer look, missing standard is federated SSO for webservices. WSRP rely on webservices standards and hence federated SSO would be necessary to consume a WSRP producer produced portlet, protected by a different Web-SSO system. Example of this could be employee.com portal (WSRP consumer) accessing a portlet on a 401K.com (WSRP producers) and both the portals have their own Web-SSO systems.

SAML was designed smart to allow applications with different scenarios. Webservices security model standards could use SAML for achieving this federation. WS-Security, security language of webservices created by IBM and Microsoft defines a SAML profile for SOAP messages. WS-Security is not discussed in detail in this paper.

WS-Security is yet to become an open standard but under consideration. WS-Security SAML profile can provide federated-SSO for webservices.

Conclusion

Portal has evolved as a user-centric application integration platform. It has become a strategy rather than solution implementation.

Federated Portals is poised to become the next generation concept for portal computing. Content and Identity federation will become the key technology enablers for federated portals. Without the standards, this will become a pipe dream. JSR 168, WSRP and SAML are set to achieve these goals. Co-existence with federated content and identity standards will become one of the key selection criteria for all off-the-shelf web solutions market like Vortals.

In order to reach to the next generation, businesses would have to conduct assessments to identify federate-able portals. The value of consolidating infrastructures for application and content delivery has proven in case of portals. Federation follows its footsteps at the enterprise level avoiding multiple user identities even between portals, eliminating the need for same application launches on different portals. In some cases, such federation model can even become business opportunities. An example of this could be an enterprise participating in the federated ebusiness eco-system acting as a service provider.

Portal Implementations are challenging because of involved integration complexities. Planning, Choosing, Aligning and Governing were the key elements of a successful portal implementation. In case of federated portals, challenges are no longer at the departmental levels, but at the higher level, enterprise-wide.

References

JSR-168 – <http://www.jcp.org/aboutJava/communityprocess/review/jsr168/>
SAML – http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
WSRP - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
WS-Security - <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>
Web Logic Portal 8.1 - <http://www.bea.com/products/weblogic/portal/index.shtml>
WebSphere Portal - <http://www-4.ibm.com/software/webservers/portal/>
SunOne Portal: http://www.iplanet.com/products/iplanet_portal/home_portal.html
Oracle: Oracle 9i Portal <http://www.oracle.com/ip/ deploy/ias/portal/index.html>
Oblix – www.oblix.com
IBM Tivoli – www.tivoli.com
Netegrity – www.netegrity.com
RSA – www.rsasecurity.com
Developer Works - <http://www-106.ibm.com/developerworks/webservices/library/ws-wsrp/>
Developer Works - <http://www-106.ibm.com/developerworks/xml/library/x-samlmyth.html>