

uPortal Project Survey

Respondent

John Lewis

Goal/Problem Space

uPortal is an enterprise portal framework focused on providing a single web location where users can easily find and use all the services available in a complex institutional enterprise.

It allows an enterprise to aggregate numerous user services into a single unified system, apply both institutional and individually controlled personalization to the experience of the system, while also providing rich access control over what services are available to each user.

Another way to think of the enterprise portal is as the primary User Interface for a truly Services Oriented Architecture (SOA).

Features

- Proven scalability to over a million users
- Pluggable API's throughout the framework
- Sophisticated layout management
- Validate user credentials against multiple backing credentials stores, or integrate easily with SSO frameworks like CAS, Shibboleth, or PubCookie
- ser attribute service draws data from multiple sources
- Powerful and flexible user group management system that can base groups dynamically on user attributes and that allows nested groups
- Layout system permits fragments and portlets to be presented to users in combinations determined by personal user attributes, preferences, group memberships, and the application of configurable and extensible layout component audience rules
- Powerful authorization system used to define and check each user who is permitted to take what actions in what contexts.
- Permissions for access to layout fragments, individual portlets, group administration, and channel publication.
- Ability for most institutions to easily skin, configure, and lightly edit the default theme in order to achieve their desired results rather than having to start from scratch or maintain a large list of local changes.
- Includes built-in support for channels presenting syndicated feeds, images, bookmarks, web proxies, inline frames, custom uPortal IChannels, and custom standards adherent JSR-168 portlets. Large and growing collection of community channels and portlets for a wide variety of functionality.
- Easily update or import channels, users, and layouts for migrations and upgrades.

Technology Stack

Java, Spring Framework, XML, XSLT, JSP, jQuery, Hibernate, Ehcache, Apache Pluto, Apache Tomcat, Apache Commons, Quartz, Maven.

Identity Services

In the context of uPortal, the portal framework serves as a place for portlets to execute. In considering these portlets as independent bits of enterprise capability, uPortal does produce/broker/convey identity services to the portlets that execute inside it and those portlets may further convey that identity information to other services it interacts with. But it does not provide them directly for services that are not executing within the portal.

Managed Information	Consume?	Produce?	Broker /Convey?
Privileges			
Roles			
Groups	X	X	X
Attributes	X	X	X
Identification	X	X	X
Defined Interfaces	Consume?	Produce?	Broker /Convey?
Authentication	X		X
Attributes	X		X
Permissions	X		
Provisioning			
Authorization	X		
Subjects	X		
Other	Consume?	Produce?	Broker /Convey?

Standards and Interfaces

uPortal does have direct support for using databases and LDAP as sources of identity information. It also comes bundled with CAS, which supports SAML. It can easily be fronted with other SSO frameworks like Shibboleth and CAS to provide any standard support they are capable of.

In collaboration with the University of Chicago, the Jasig Web Proxy Portlet is currently being updated to support SAML proxy authentication, allowing it to use a Shibboleth assertion to access other enterprise systems securely as the active user. This is being done with some new APIs in uPortal and a new standard library that will allow any portlet to use SAML proxy authentication as needed.

Issues and Challenges

uPortal currently does much of its own groups and permissions management internally, and in fact has a subproject called "Groups and Permissions" or GAP. While the services that GAP provides are sophisticated and have served uPortal well, the user interface for these services is bad and does not receive much developer attention since it is used by relatively few users. The functionality of GAP is a near complete duplication of the Grouper project and so it would be better to integrate with a project that receives more attention on the user experience of managing groups and permissions.

More Information

<http://www.uportal.org/>