



Identity Management in Liferay Overview and Best Practices

LIFERAY PORTAL 6.0 EE

Table of Contents

INTRODUCTION 1

 IDENTITY MANAGEMENT HYGIENE 1

WHERE LIFERAY FITS IN 2

HOW LIFERAY AUTHENTICATION AND AUTHORIZATION WORK 4

 AUTHENTICATION 4

 AUTHENTICATION PIPELINE 4

 AUTOLOGIN..... 5

 A FILTER LESSON 5

 AUTHORIZATION 5

 ROLE-BASED AUTHORIZATION CONTROL (RBAC) 6

WHAT ARE MY OPTIONS?..... 7

 OUT OF THE BOX 7

 LDAP 7

 CAS SSO + LDAP..... 8

 NTLM..... 8

 OPENID..... 9

 FACEBOOK..... 10

 SITEMINDER SSO + LDAP 10

USING SYSTEMS THAT ARE NOT SUPPORTED OUT OF THE BOX 11

 WHAT ABOUT SAML? 11

 WHAT ABOUT SHIBBOLETH?..... 11

 WHAT ABOUT OTHERS?..... 11

SUMMARY 12

DISCLAIMER 12

MOVING FORWARD 12

 LIFERAY ENTERPRISE EDITION SUPPORT 12

 LIFERAY PROFESSIONAL SERVICES 12

Introduction

Identity management, or IdM, is a broad administrative area that deals with identifying individuals in a system (e.g., country, network, or organization) and controlling access to the resources in that system by placing restrictions on the established identities of the individuals.

Liferay Portal provides a robust authentication and authorization framework that allows you to manage users as desired by using the built-in mechanism or plugging into other identity and authentication sources.

Identity Management consists of mainly two things: Authentication and Authorization. While IdM does encompass those two concepts (focusing on Identity), it is more than just authentication (AuthN) and authorization (AuthZ). Some systems are dependent upon an Identity Provider (IdP) to store its users. These IdPs may include LDAP servers (including Microsoft Active Directory), SSO servers, Facebook, OpenId, and others. In Identity Management, the operative word is *management*. How should you manage identities for Liferay?

IDENTITY MANAGEMENT HYGIENE

A good IdM solution or system that employs good IdM practices good IdM hygiene. In other words, for IdM to be reliable, it must be able to obtain an identity from an authoritative source. Another good practice is to ensure that the creation of that identity is monitored and audited. The identity should also be locked down in such a fashion that only the true entity has claim on it and prevents any other entity from assuming or altering it.

Where Liferay Fits In

You may be wondering where Liferay fits into your existing ecosystem. The key thing to remember is this: Liferay is a Java web application, running on a standard Java servlet container or application server. Since the choice of app server does not matter for IdM, we will use Liferay on Apache Tomcat as an example.

Here is a typical Liferay installation:

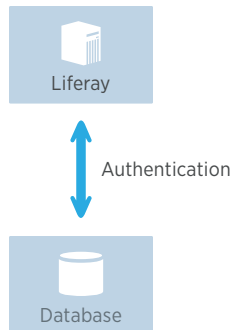


Figure 1

In Figure 1, it is apparent that Liferay thrives in this ecosystem. In this example, Liferay is relying on its own authentication mechanism.

Here is a typical Liferay installation, with LDAP:

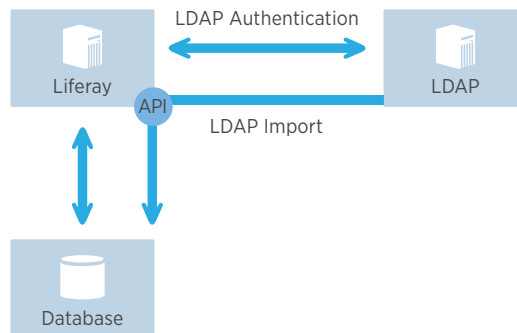


Figure 2

In Figure 2, Liferay is using LDAP as its one true source of authentication. The portal no longer relies on the user information in the Liferay database but on the user credentials stored in the LDAP server. For Microsoft Active Directory, Figure 2 applies as well. In this setup, the user ID and passwords are managed from the LDAP server along with the tools that are associated with it. Many organizations either have a separate password reset or administration page or need to call their IT departments to reset or administer their password. In either case, Liferay does not know anything about these tools. It simply binds to the LDAP server and verifies whether the user has provided valid credentials in the portal.

By default, once authenticated and the user's session is created, the user's basic credentials (username, password, email address, first name, last name, title) are all imported into the Liferay database in a one-way sync. This happens every time a user performs a login. There is also the option to configure Liferay Portal to do a two-way sync (export). The two-way sync allows users to change their information from within the portal and have it sync outwardly to the LDAP server. The synchronizations can happen real-time or be set to a time interval. If you wish to have the LDAP server be the central point of administration, it is recommended that you do not turn on the two-way sync, allowing the LDAP server to remain as the one true source of authentication at all times.

Here is a typical Liferay Installation, with SSO and LDAP:

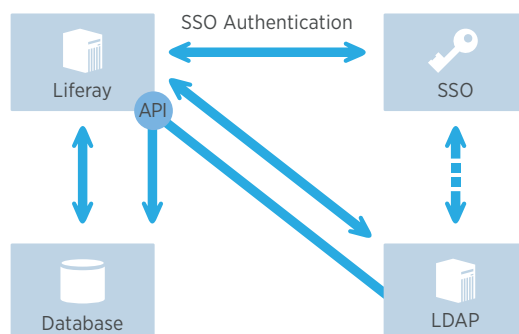


Figure 3

In Figure 3, Liferay Portal is relying on the SSO server to broker the authentication, with LDAP storing the user data. Figure 3 demonstrates that even though the SSO server is responsible for authentication, that user's identity is still stored in the LDAP server. The SSO server will use one of a variety of authentication mechanisms. These include cookies, tokens, and agents. Whatever implementation is used, the SSO server will simply authenticate and allow you a session. It will not give complete user identity to the portal. In Figure 3, the user is authenticated by the SSO server, but after authentication, the user's information is imported into the portal from LDAP. In this sort of system, other applications can leverage your SSO server without requiring LDAP on those systems. In Figure 2, if you wanted to use LDAP as your central IdM server, this would require that all your web applications are able to connect to your LDAP server.

How Liferay Authentication and Authorization Work

Authentication and authorization are separate functions, but they both fall under the umbrella of IdM.

AUTHENTICATION

Liferay authentication can be very simple, or it can have many layers.

AUTHENTICATION PIPELINE

At its most basic setting, Liferay uses either the sign-in portlet or sign-in screen of the portal to authenticate you.

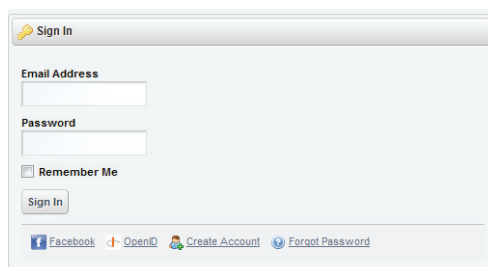
A screenshot of the Liferay 'Sign In' form. It features a title bar with a key icon and the text 'Sign In'. Below the title bar are two input fields: 'Email Address' and 'Password'. A 'Remember Me' checkbox is located below the password field. A 'Sign In' button is positioned below the checkbox. At the bottom of the form, there are links for 'Facebook', 'OpenID', 'Create Account', and 'Forgot Password'.

Figure 4.a

Administrators can specify if you login via:

- Email address (default)
- Screen name
- User Id (primary key in the User_ table)

A screenshot of the Liferay 'Settings' page, specifically the 'Authentication' section. The 'General' tab is selected. Under 'How do users authenticate?', the 'By Email Address' option is chosen. Several checkboxes are visible: 'Allow users to automatically login?' (checked), 'Allow users to request forgotten passwords?' (checked), 'Allow users to request password reset links?' (checked), 'Allow strangers to create accounts?' (checked), 'Allow strangers to create accounts with a company email address?' (checked), and 'Require strangers to verify their email address?' (unchecked). A sidebar on the right contains links for 'LIFERAY', 'liferay.com', 'Configuration', 'Authentication', 'Users', 'Mail Host Names', 'E-mail Notifications', 'Identification', 'Addresses', 'Phone Numbers', 'Additional Email Addresses', 'Webhooks', 'Miscellaneous', 'Display Settings', and 'Google Apps'. 'Save' and 'Cancel' buttons are at the bottom right.

Figure 4.b

If you specify an LDAP server, the user would use the same UI mechanism to sign in. However, on the back end, the LDAP server will be used to authenticate instead of the Liferay database. The behavior above is defined in the Liferay Authenticator class and is governed by the Liferay Authentication Pipeline. See the *portal.properties* file for more on the Authentication Pipeline.

AUTOLOGIN

This class is a filter that is called when a user has not been authenticated or their previous session has timed out. The AutoLogin classes are defined in `auto.login.hooks` property:

```
auto.login.hooks=com.liferay.portal.security.auth.CASAutoLogin,com.liferay.portal.security.auth.  
FacebookAutoLogin,com.liferay.portal.security.auth.NtlmAutoLogin,com.liferay.portal.security.auth.  
OpenIdAutoLogin,com.liferay.portal.security.auth.OpenSSOAutoLogin,com.liferay.portal.security.auth.  
RememberMeAutoLogin,com.liferay.portal.security.auth.SiteMinderAutoLogin
```

The classes are called in the order that they are defined. AutoLogin class attempts to “automatically” log in users. It appears to be automatic because the user does not necessarily have to interact (e.g. enter login information). For example, if the user has already authenticated to a Single-Sign-On (SSO) server, the AutoLogin class may be able to access data from the SSO server to log the user automatically into Liferay. Another example is the RememberMeAutoLogin module that uses a cookie that will automatically log in a user who has previously logged in.

A FILTER LESSON

Filters are an underappreciated feature of the Java servlet platform, ideal for writing components that can be added transparently to any web application. A filter is like a lightweight servlet that, instead of generating its own content, plugs into the request handling process and executes in addition to the normal page processing.

Filters might record information about requests, convert content to a different format, or even redirect access to a different page. Filters can be applied to any resources served by a servlet engine, whether it is flat HTML, graphics, a JSP page, servlet, or the like. They can be added to an existing web application without the filter or the application being aware of one another. Filters are essentially a server plugin that works with any servlet container compliant with version 2.3 or later of the servlet specification.

A filter implements the interface `javax.servlet.Filter` and is configured in the web application `web.xml` file, where the URLs it will process are defined. For each request, the servlet container decides which filters to apply and adds those filters to a chain in the same order they appear in `web.xml`. Each filter has its `Filter.doFilter()` method called, which triggers the invocation of the next filter in the chain or the loading of the final resource (HTML page, servlet, etc.).

For authentication in Liferay, filters are used to “protect” the Liferay app with whatever IdP you have configured. For example, if you have CAS set up, the `CASAutoLogin` class will achieve login for you, but it will not protect the app so that it redirects to your CAS login page nor pass along the correct data to be used by the AutoLogin class. Instead, the `CASFilter` will provide that functionality.

AUTHORIZATION

Authorization involves a user’s rights and privileges to view, edit, or update and have general access to different components of the portal. These components can be as small as a fragment of HTML or piece of web content, or as big as a portlet application, whole pages, or entire sets of pages (e.g., communities, organization, Liferay instances). This is implemented in Liferay Portal via Liferay’s fine-grained permissioning system. The administration of the *implementation* of the actual privileges is done in the portal itself via Liferay roles. The administration of the *assignment* of these privileges can be done from an outside system, such as LDAP using LDAP groups. Liferay synchronizes the LDAP groups and the membership of these groups from the LDAP server to Liferay Portal. The LDAP users are imported into Liferay users, and the LDAP groups are imported into Liferay user groups. Also, if the LDAP user is a member of the LDAP group, the Liferay user will be a member of the corresponding Liferay user group. *In other words, the Liferay users and user groups mirror the LDAP users and groups.*

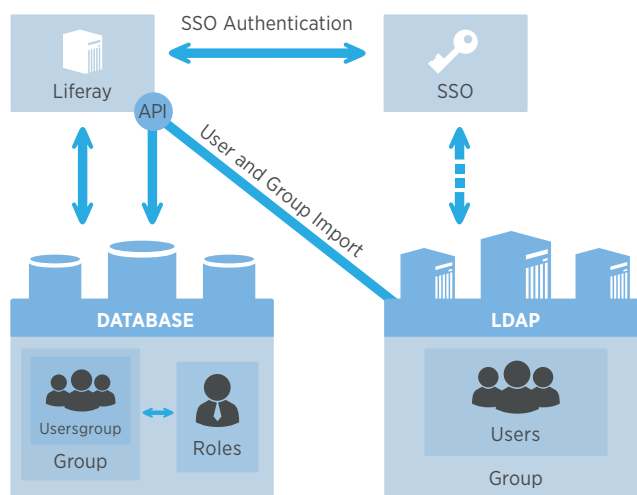


Figure 5

In Figure 5, Liferay Portal is deferring the management of the authentication to the SSO server and the administration of the assignment of user groups and roles to the LDAP server. Once all of the users and user groups exist in the portal, Liferay roles that have been assigned to each Liferay user group will control what access each individual user has to various areas of the portal. In this way, by virtue of placing an LDAP user in an LDAP user group, an administrator can assign or take away roles to individual users. In regards to Liferay roles, Liferay portal administrators create and define the privileges that make up those roles.

ROLE-BASED AUTHORIZATION CONTROL (RBAC)

Liferay 6.0 uses RBAC permissions out of the box, so no extra effort is required to use this implementation other than creating your own custom roles and defining their privileges, if desired. You can map to these roles from something like an LDAP group, maintaining the single point of administration on the IdP. You can also map from other entities on your IdP using the Liferay Plugins SDK to extend a current module or implementation, or create a new one altogether.

What are My Options?

OUT OF THE BOX

Liferay does not require LDAP, SSO, or any other external authorization mechanism or server.

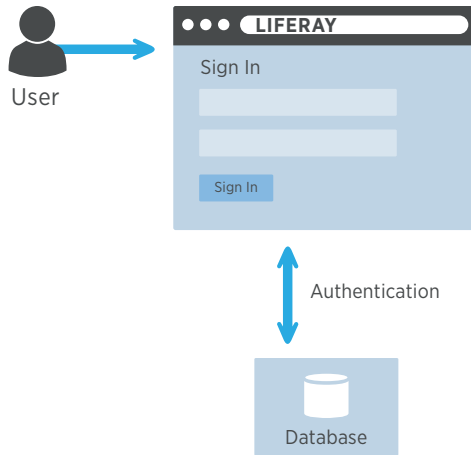


Figure 6.a

In Figure 6.a, the user actively logs in via a sign-in screen or portlet and is authenticated against the Liferay database.

LDAP

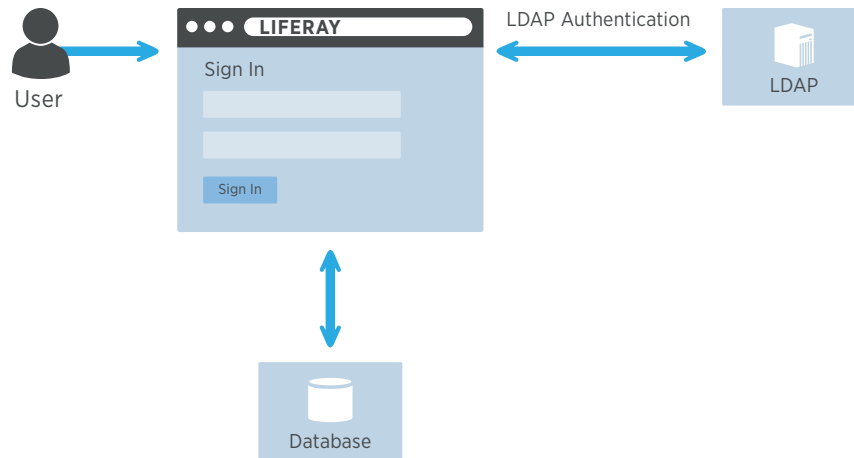


Figure 6.b

In Figure 6.b, after the user actively logs in via sign-in screen or portlet, Liferay Portal binds to the LDAP server and uses the input provided by the user to authenticate against the credentials in the LDAP server.

CAS SSO + LDAP

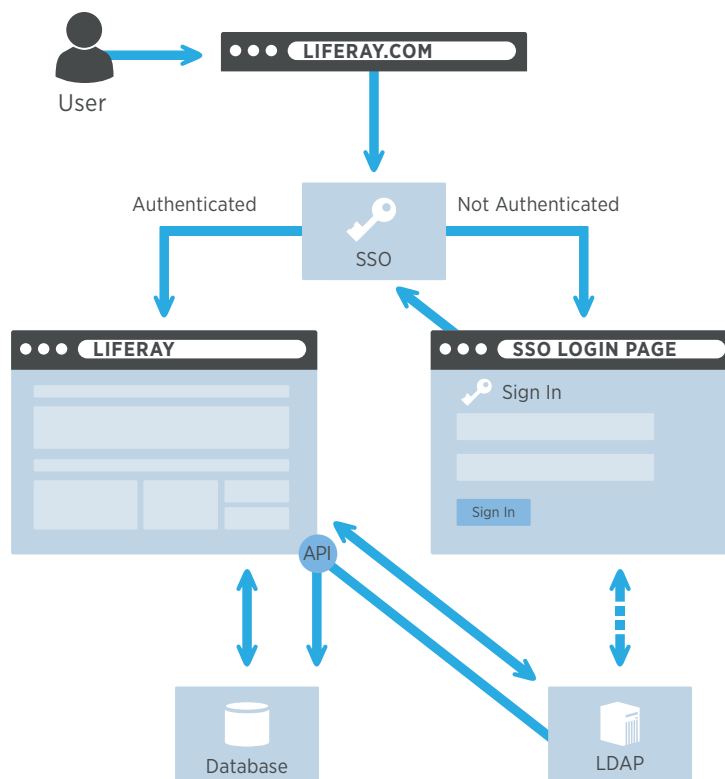


Figure 6.c

In Figure 6.c, there may not be any explicit login that occurs. The reason is that the user may have already provided their credentials to the SSO server via some other web application. If they have already been authenticated, the Liferay auto-login hooks (which includes CAS) will kick in when a portion of the portal that requires authorization is accessed. A check for CAS authorization will occur. If the user is authentic the page will render. If there is no authorization, then the user will be redirected to the CAS login URL for authentication. After authentication, the user will be redirected to the original URL that was attempted.

NTLM

NTLM is a Microsoft protocol that can be used for authentication through Microsoft Internet Explorer. Although Microsoft has adopted Kerberos in modern versions of Windows' server, NTLM is still used when authenticating to a workgroup. Liferay Portal now supports NTLM v2 authentication, which is more secure and has a stronger authentication process than NTLM v1.

OPENID

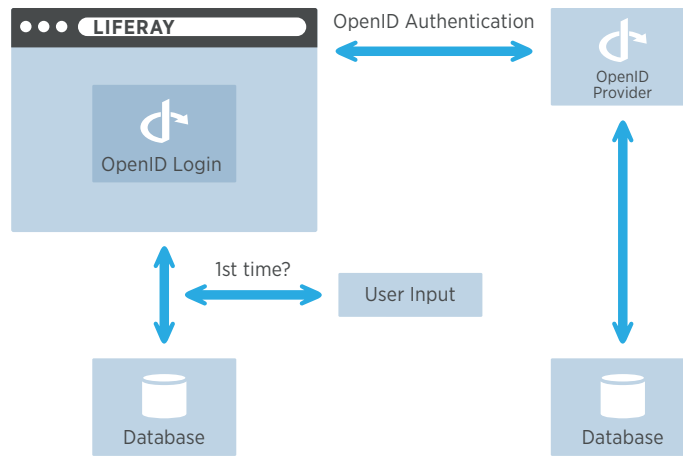


Figure 6.d

OpenID is a single sign-on standard that is implemented by multiple vendors. The idea is that multiple vendors can implement the standard and users can register for an ID with the vendor they trust. The credential issued by that vendor can be used by all websites that support OpenID. Some high profile OpenID vendors are AOL (<http://open-id.aol.com/screenname>), LiveDoor (<http://profile.livedoor.com/username>), and LiveJournal (<http://username.livejournal.com>). Please see the OpenID site (<http://www.openid.net>) for a more complete list.

The main benefit of OpenID for users is that they no longer have to register for a new account on every site in which they would like to participate. Users can register on one site (the OpenID provider's site) and then use those credentials to authenticate to many websites that support OpenID. Many website owners often struggle to build communities because end-users are reluctant to register for several different accounts. Supporting OpenID makes it easier for site owners to build their communities because the barriers to participating (i.e., the effort it takes to register for and keep track of many accounts) are removed. All of the account information is kept with the OpenID provider, making it much easier to manage this information and keep it up to date. Liferay Portal can act as an OpenID consumer, allowing users to automatically register and sign in with their OpenID accounts. The first time you log in via OpenID, the portal will ask you for a first name, last name, email address, captcha verification, and verification from the OpenID provider. Thereafter, the user's data will be stored in the Liferay database and the user's OpenID will be sufficient to authenticate.

FACEBOOK

Liferay can leverage Facebook accounts and use the data from Facebook accounts to authenticate.

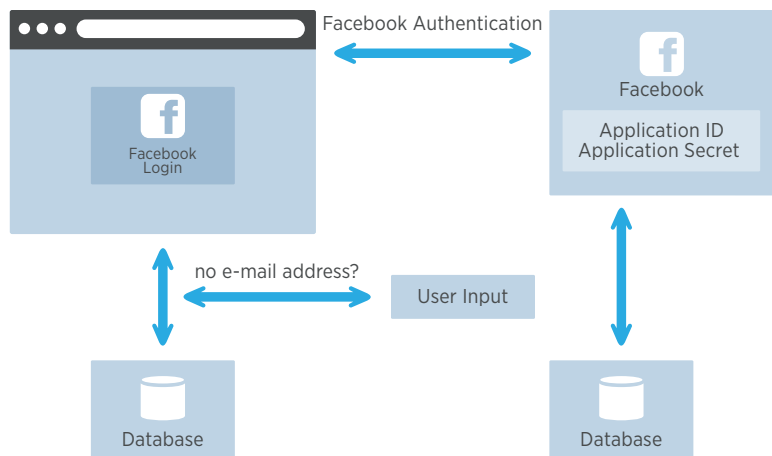


Figure 6.e

Facebook SSO works by taking the primary Facebook email address and searching for the same email address in Liferay’s User_ table. If a match is found, the user is automatically signed on (provided that user clicked “Allow” from the Facebook dialog). If no match is found, the user is prompted in Liferay to add a user from Facebook. Once selected, a new user is created by retrieving four fields from Facebook (first name, last name, email address, and gender).

SITEMINDER SSO + LDAP

SiteMinder is another common SSO provider that uses an agent instead of a token or cookie to allow authentication. SiteMinder uses a custom HTTP header to implement its single sign-on solution.

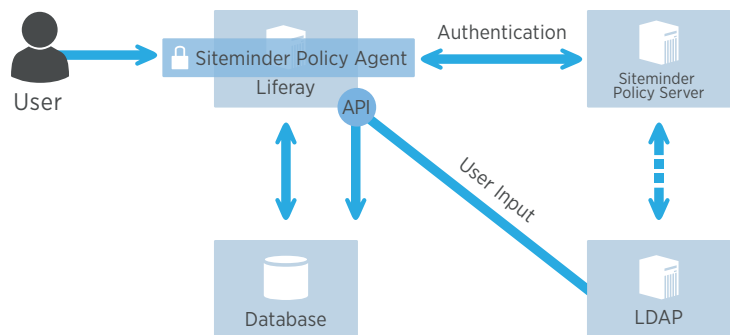


Figure 6.f

Using Systems That are Not Supported Out of the Box

Indeed, you have the ability to use systems that are not supported out of the box. The Liferay Plugins SDK can be used to extend Liferay with extra Authenticator or AutoLogin classes. But how do you know which class to implement? That depends on these factors:

- Do you require the user to always sign in from Liferay Portal?
- Do you require the user to auto login after already signing in somewhere else?

Often, Liferay customers already have a proprietary or closed-source sign-on server already existing in their ecosystem. If you find that yours is not supported out of the box, it is a relatively simple development task to add another AutoLogin class to the chain that should already exist in `portal.properties`. Actually, you can overwrite this property via `portal-ext.properties` to only implement the AutoLogin filters that you use, including any custom ones you have created. For example:

```
auto.login.hooks=com.liferay.portal.security.auth.RememberMeAutoLogin,com.abc.portal.security.auth.  
SunAccessManagerAutoLogin
```

A great method for starters is to simply copy one of the existing AutoLogin classes and use that as a template for your custom class.

WHAT ABOUT SAML?

For Liferay 6.0EE, there is already support SAML for SSO if you use something such as OpenSSO, CA SiteMinder, or Oracle Access Manager. In those situations, the SSO serves as the Identity Provider; Liferay consumes those services. One could say there is no “direct” integration for SAML because Liferay leverages the services of the SSO server. Those who wish to use SAML with another system may still do so, but Liferay recommends that, if possible, you find a SAML plugin or add-on to your existing authentication server.

For Liferay Portal 6.1, SAML 2.0 is supported out of the box, as it is able to use Liferay Portal itself as a SAML identity provider. In such cases, other applications can use Liferay Portal directly as the SAML IdP.

WHAT ABOUT SHIBBOLETH?

Shibboleth 2.0 builds on SAML 2.0 standards. The IdP in Shibboleth 2.0 has to do additional processing in order to support passive and forced authentication requests in SAML 2.0. The Service Provider (SP) can request a specific method of authentication from the IdP.

Liferay 6.0EE does not have out-of-the-box support for Shibboleth, but the Liferay Plugins SDK could be easily used to achieve such integrations.

WHAT ABOUT OTHERS?

There are many other IdPs available. You have the freedom to integrate with any desired IdP via a Liferay Plugin, though some implementations may be easier than others. For example, Sun (now Oracle) Access Manager is based on OpenSSO. Liferay integrates with OpenSSO out of the box. Depending on the version of Sun Access Manager (SAM) being used, there may be nuances with Sun Access Manager that are not compatible with the out-of-the-box OpenSSO implementation. In this case, Liferay recommends creating a new SAM auto-login module. You may use the existing OpenSSO class as a template to create your new SAM auto-login class, especially since it is very close to the implementation you need. Likewise, with other IdPs, you can use existing classes as templates or starting points to get to where you need in your new custom authentication class.

Summary

In this paper, we outlined how Liferay authentication and authorization can be used in conjunction with your existing IdM systems. All of the administration of the actual user identities can still occur on your existing IdM system. Liferay implements RBAC with its own role-based permissioning system, and these roles can be mapped from groups that exist outside of the portal, such as an LDAP system.

Disclaimer

Liferay can only give you an initial IdM recommendation based on best practices and the experience of professionals working with Liferay customers. Ultimately, it is your responsibility as system architects and business analysts to come up with the scenarios that your system will need to address and to run the appropriate tests on your system before production deployment, so that you can identify significant circumstances and other unforeseen system and network issues.

Please use this document and the Liferay Portal Administrator's Guide (<http://www.liferay.com/documentation/liferay-portal/6.0/community-resources>) for more detailed information on how to configure Liferay Portal settings.

Moving Forward

LIFERAY ENTERPRISE EDITION SUPPORT

Liferay Enterprise Edition ensures stability and reliable technical support for your Liferay Portal installation and your organization's team, including a customer portal, product bulletins, security alerts, and support from over 60 partners worldwide.

LIFERAY PROFESSIONAL SERVICES

Liferay Professional Services can help you in the design, planning, and implementation of your system. Performance tuning consultation is also available.

Please contact sales@liferay.com for more information.



LIFERAY, INC. is the provider of leading enterprise open source portal and collaboration software products, used by major enterprises worldwide, including Allianz, AutoZone, Benetton Group, Cisco Systems, Lufthansa Flight Training, The French Ministry of Defense, and the United Nations. Liferay, Inc. offers professional services, technical support, custom development and professional training to ensure successful deployment in the most demanding IT environments.

© 2011, Liferay, Inc. All rights reserved.