

Identity Management in Liferay DXP

Overview and Best Practices

Table of Contents

Introduction	1	Other Third-Party Solutions	16
Identity Management Hygiene	1	Facebook & Google	16
Where Liferay Fits In	1	Additional Options	16
How Liferay Authentication and Authorization		Using Systems That Are Not Supported Out-of-	
Work	4	the-Box	17
Authentication	4	What about other IdPs?	17
Authentication Pipeline	4	Summary	18
AutoLogin	5	Disclaimer	18
A Filter Lesson	6	Moving Forward	18
Authorization	6	Liferay Enterprise Subscription	18
Role-Based Authorization Control (RBAC)	8	Liferay Global Services	18
What Are My Options?	8		
Out-Of-The-Box Liferay Authentication	8		
LDAP	9		
CAS SSO + LDAP	10		
NTLM	11		
OpenID	11		
Siteminder SSO + LDAP	12		
Other SAML- and OAuth-Based Solutions	12		
Included SAML Support	13		
Included OAuth Support	14		

Introduction

Identity Management (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.

IdM consists of mainly two things: Authentication and Authorization. While IdM does encompass those two concepts which focus on identity, it is more than just authentication (AuthN) and authorization (AuthZ). IdM is part of a federated identity management strategy, which requires a common set of policies, practices and protocols to be used for end users and devices across groups. Some systems are dependent upon an Identity Provider (IdP) to store its users. Liferay supports these IdPs and more: Single Sign On (SSO) servers, SAML, LDAP, Facebook, Google, OpenId, Open Authorization (OAuth), Shibboleth and others.

In Identity Management, the operative word is management. How should you manage identities for Liferay? Liferay Digital Experience Platform (DXP) 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 Hygiene

A solution or system that employs good IdM should practice 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 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.

A typical Liferay installation:

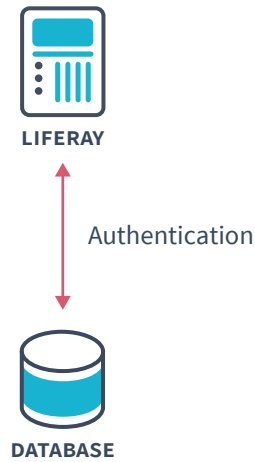


Figure 1

In Figure 1, the Liferay platform is relying on its own internal authentication mechanism, without a third-party IdP.

A typical Liferay installation with LDAP:

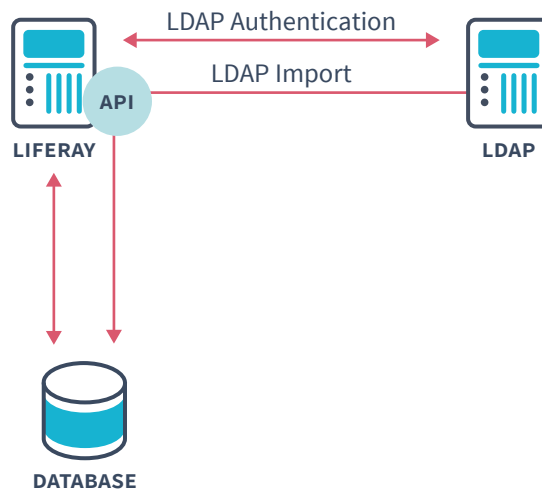


Figure 2

In Figure 2, the Liferay platform is using LDAP as its one true source of authentication. In this case, Liferay does not rely 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) 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 to do a two-way sync (export). The two-way sync allows users to change their information from within Liferay 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.

A typical Liferay Installation with SSO and LDAP:

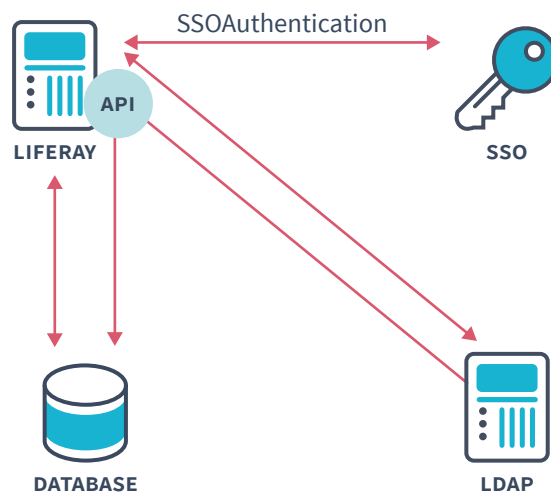


Figure 3

In Figure 3, the Liferay platform 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 Liferay 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 have many layers.

Authentication Pipeline

In its most basic configuration, Liferay uses either the sign-in portlet or sign-in screen of the Liferay web application to authenticate you.

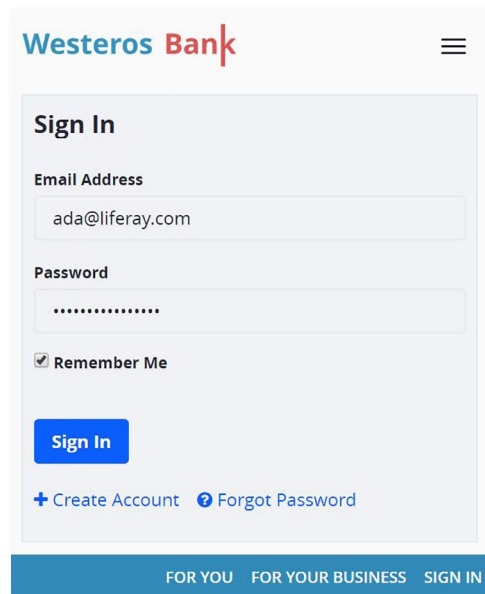


Figure 4.a

Administrators can specify if you login via:

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

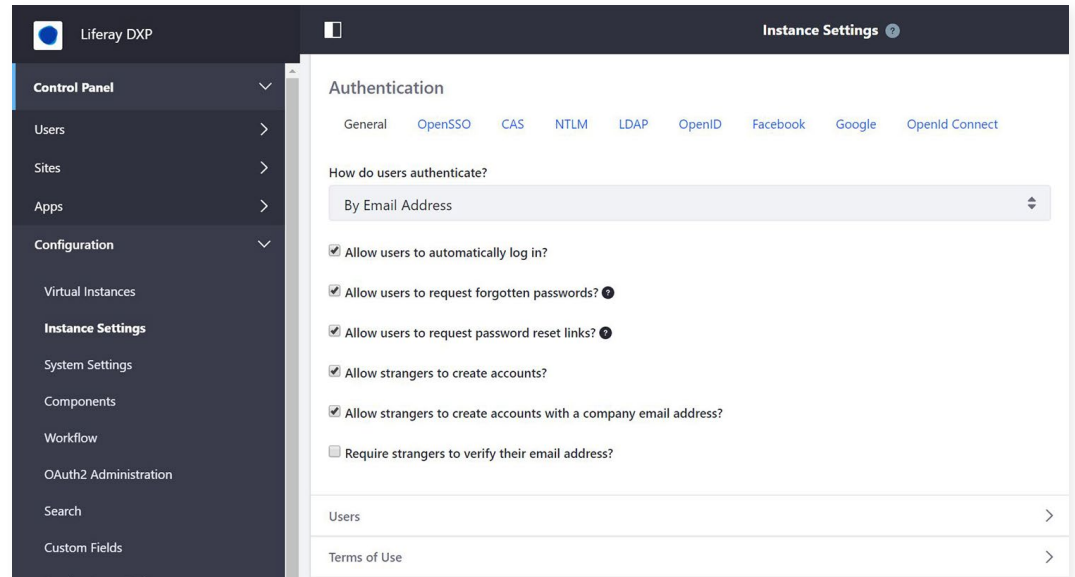


Figure 4.b

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.sso.cas.internal.  
auto.login.CASAutoLogin,com.liferay.portal.security.sso.facebook.  
connect.auto.login.FacebookConnectAutoLogin,com.liferay.portal.  
security.sso.ntlm.auto.login.NtlmAutoLogin,com.liferay.portal.  
security.sso.openid.auto.login.OpenIdAutoLogin,com.liferay.portal.  
security.sso.opensso.auto.login.OpenSSOAutoLogin,com.liferay.  
portal.security.auto.login.remember.me.RememberMeAutoLogin
```

The classes are called in the order that they are defined. The `AutoLogin` class attempts to automatically log in users. It is 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 provide data to automatically log in a user who has previously done so.

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 Liferay's `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 will it 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, update or 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, organizations,

Liferay instances). This is implemented in Liferay through Liferay Roles, while the assignment of these privileges is sometimes done from an outside system, such as LDAP using LDAP groups. Liferay synchronizes the LDAP groups and their membership 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*.

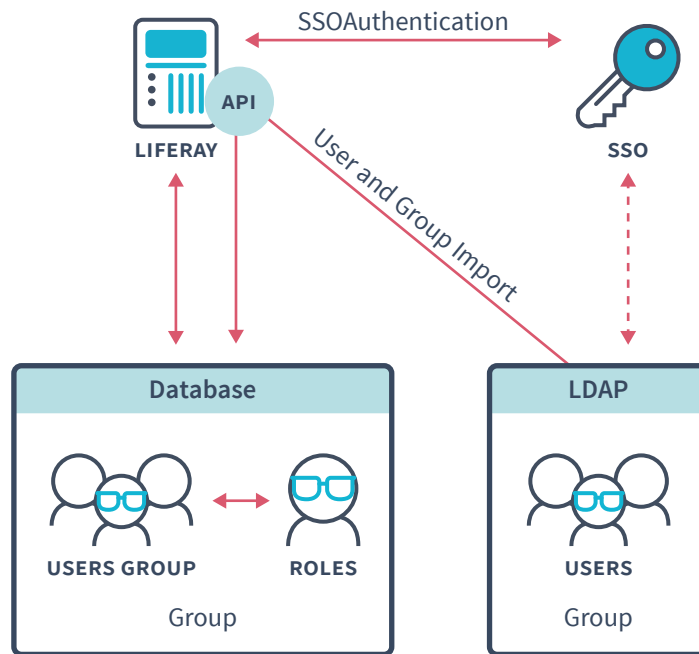


Figure 5

In Figure 5, Liferay 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 remove roles from individual users. In regards to Liferay roles, Liferay administrators create and define the privileges that make up those roles.

Role-Based Authorization Control (RBAC)

Liferay 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 Workspace to extend a current module or implementation, or create a new one altogether.

What Are My Options?

Out-Of-The-Box Liferay Authentication

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

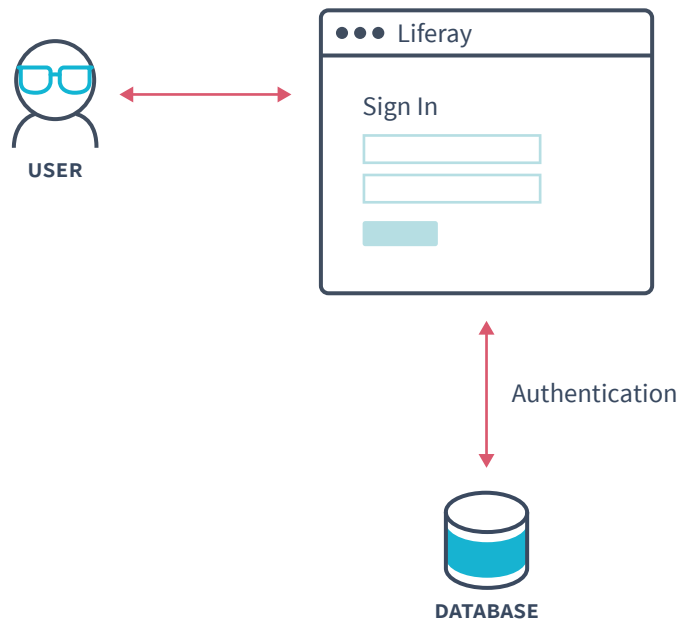


Figure 6.a

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

LDAP

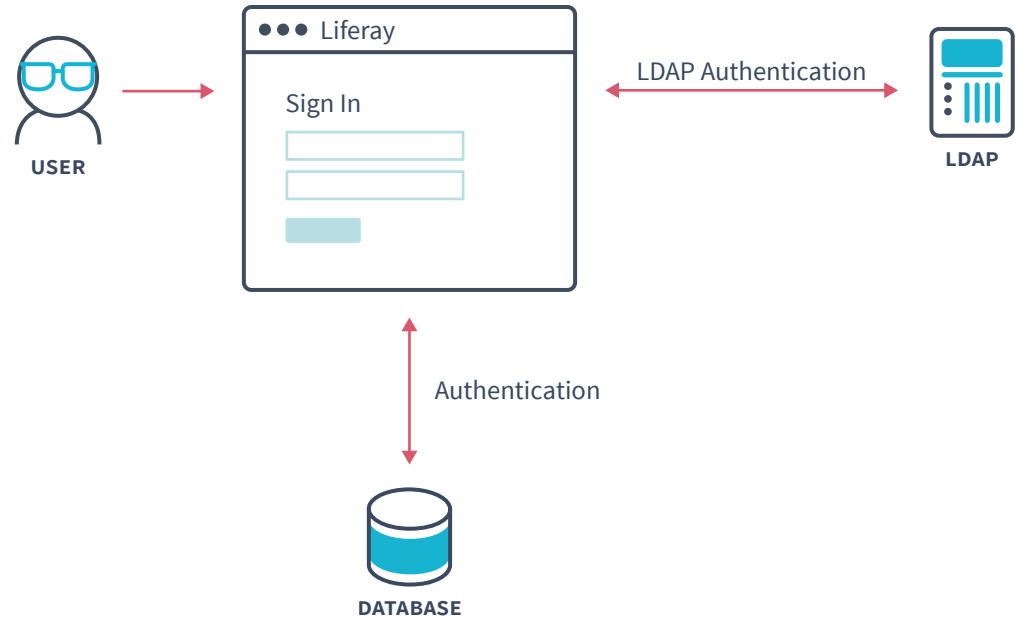


Figure 6.b

In Figure 6.b, after the user actively logs in through a sign-in screen or portlet, Liferay 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

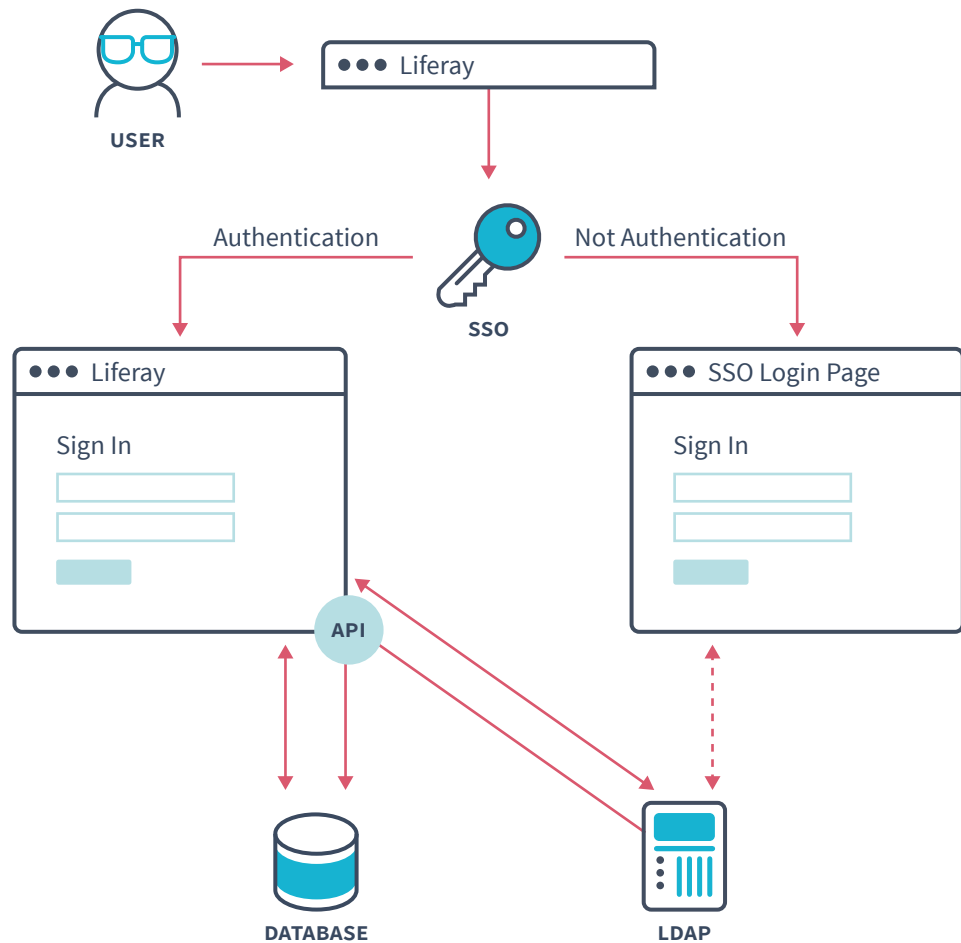


Figure 6.c

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 web application 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 now supports NTLM v2 authentication, which is more secure and has a stronger authentication process than NTLM v1.

OpenID

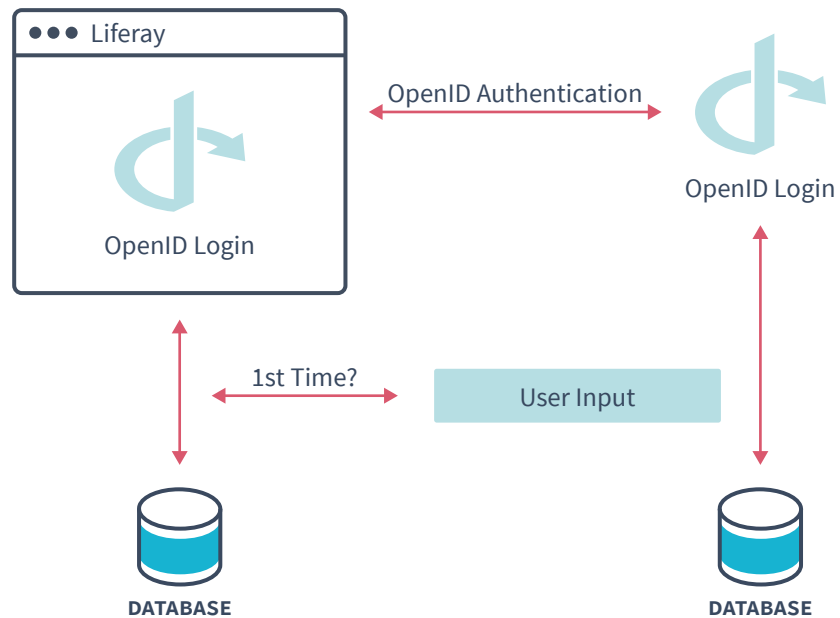


Figure 6.d

OpenID is a single sign-on standard implemented by multiple vendors. Users can register for an ID with the vendor they trust. The credential issued by that vendor can be used by all the web sites that support OpenID. Some high profile OpenID vendors are Google, Paypal, Amazon and Microsoft. Please see the OpenID site for a more complete list.

With OpenID, users don't have to register for a new account on every site which requires an account. Users register on one site (the OpenID provider's "site" and then use those credentials to authenticate to many web sites which support OpenID. Web site owners sometimes struggle to build communities because users are reluctant to register for another account. Supporting OpenID removes that barrier,

making it easier for site owners to build their communities. All the account information is kept with the OpenID provider, making it much easier to manage this information and keep it up to date.

Liferay DXP can act as an OpenID consumer, allowing users to automatically register and sign in with their OpenID accounts. Internally, the product uses OpenID4Java to implement the feature.

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.

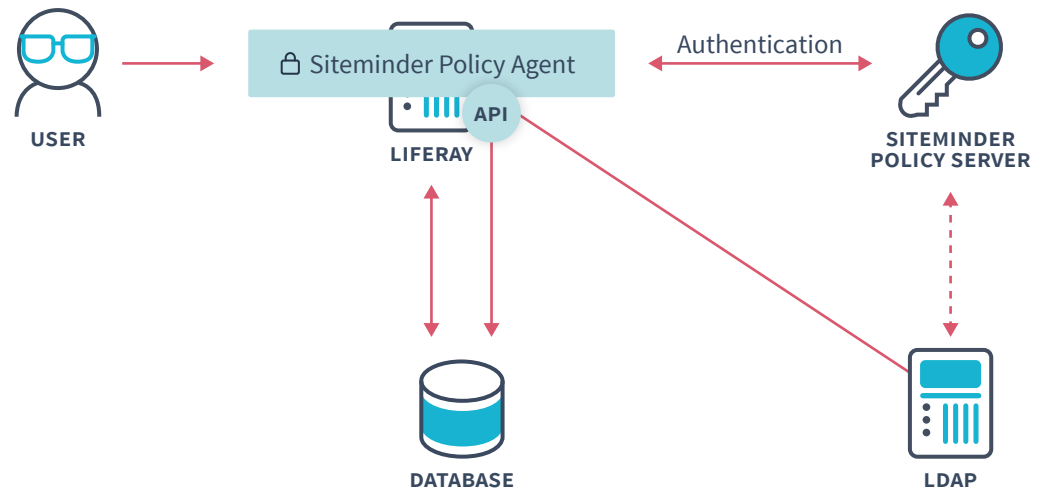


Figure 6.e

Other SAML- and OAuth-Based Solutions

For customers looking to define a federated identity management strategy, Liferay DXP can serve as a SAML Identity Provider. This provides added flexibility for customers looking to federate their DXP-based solution with applications like Salesforce.com and Workday.

SAML and OAuth 2.0: These are the most widely adopted security protocols for SSO and application sign in, supported through specific Apps that can be installed from [Liferay's Marketplace](#).

Liferay SAML 2.0 Provider: web.liferay.com/marketplace/-/mp/application/15188711

Liferay OAuth Provider: web.liferay.com/marketplace/-/mp/application/45261909

There is also support for SAML if you use products such as OpenSSO, CA SiteMinder, Okta, Oracle Access Manager, Tivoli Access Manager, OneLogin, Azure and others. Shibboleth Identity Provider supports SAML 1.1 and 2.0. In those situations, the SSO may serve as the Identity Provider and Liferay consuming those services. Liferay leverages the services of the SSO server. Those who wish to use SAML with another system may still do so, but Liferay recommends using a SAML plugin or add-on to your existing authentication server. SAML 2.0 is supported out-of-the-box, and Liferay can be used a SAML SP or IdP.

Included SAML Support

- GUI to configure endpoints
- Caching of Metadata
- Manual reload of Metadata
- HTTP based Single Logout
- SAML single logout
- Assertions containing user
 - Sites
 - Site Roles
 - User Group
 - Roles
 - Expando

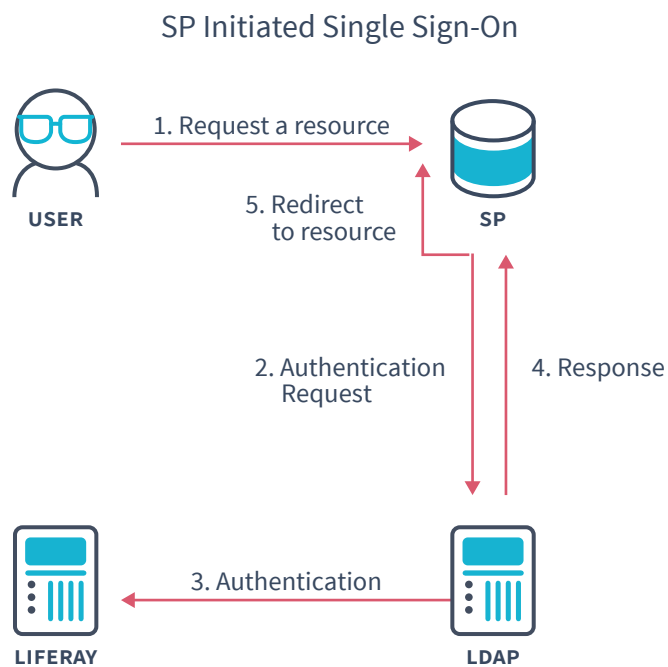


Figure 6.f

Included OAuth Support

Liferay can delegate an app or module's user authentication to a third party OAuth Service Provider, as well as act as an OAuth Service Provider. An OAuth-enabled plugin uses a token to prove it is authorized to access the user's third-party profile data and invoke authorized services. By implementing OAuth in your plugin, you get the best of both worlds—access to an outside service provider, and your users' trust that the plugin won't have access to their protected resources. When Liferay acts as an OAuth service provider, you can provide a means for your users to use their portal credentials to access other services that have OAuth configured. Liferay's OAuth Provider is available as an app from Liferay Marketplace. In today's modern web, the many Internet touchpoints users experience drive a need for scalable solutions based on OAuth, making OAuth implementations fairly ubiquitous. For example, a fairly common interaction requiring users to grant permission:

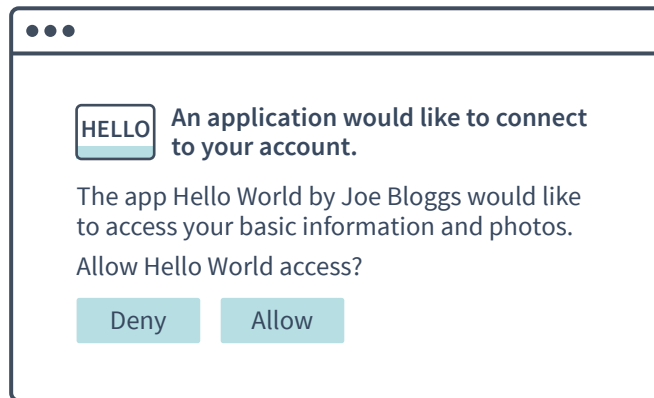


Figure 6.g

OAuth is a handshake mechanism where, instead of asking for personal information, Liferay redirects users to a service provider like Twitter or Facebook, where they can tell the Service Provider to allow Liferay to some limited access to their accounts. You wouldn't want a valet driver opening your glove box, storage spaces, hood and other personal compartments in your vehicle. You would only want the valet to access what is necessary to park your car. OAuth is based on this same idea: it gives a site just enough information to do what it needs and nothing more. This assures users that their personal information is safe, but gives them freedom to take advantage of valuable resources they typically use from the service provider's site.

A typical OAuth configuration flow:

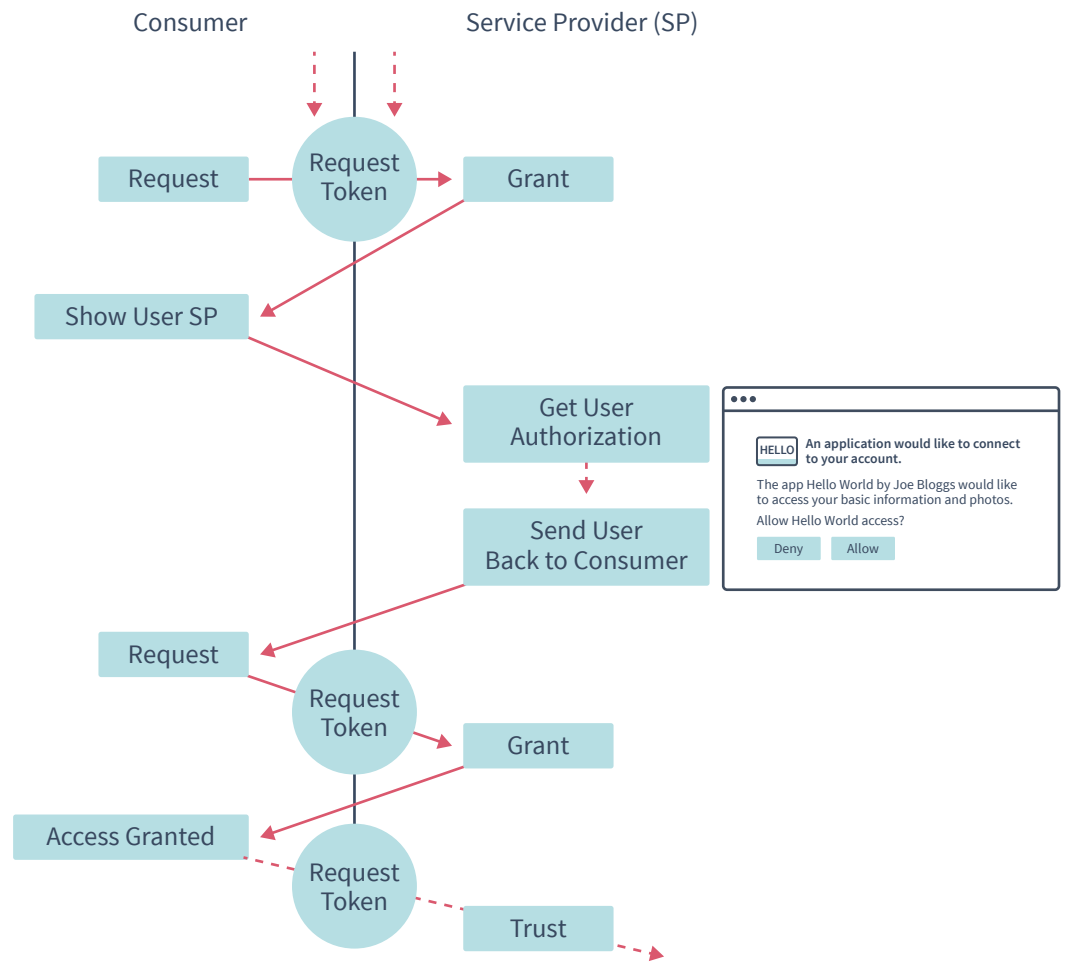


Figure 6.h

Liferay also supports user login through Facebook's OAuth authentication.

Other Third-Party Solutions

Facebook & Google

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

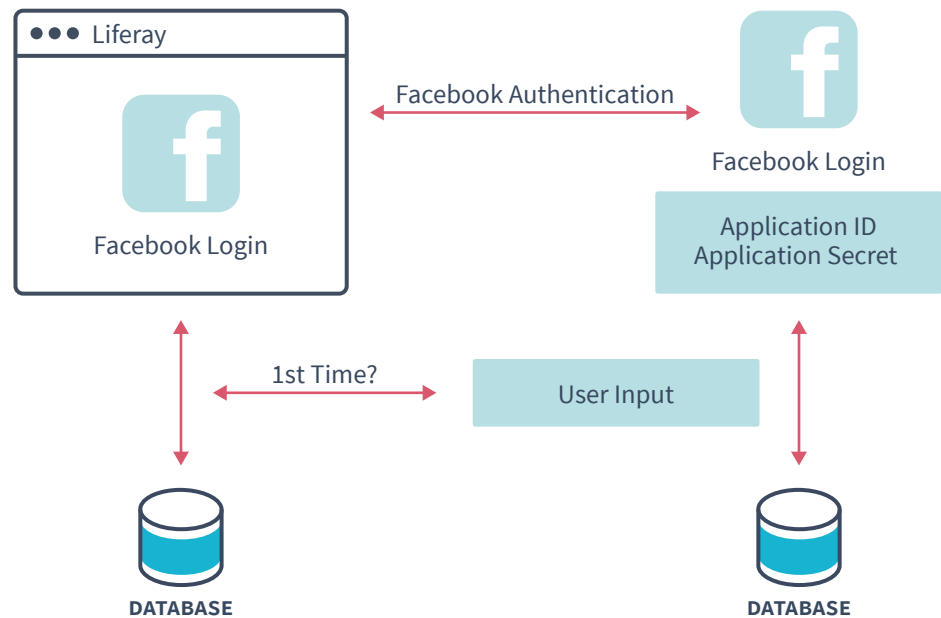


Figure 6.i

Facebook authentication mechanisms implement OAuth. Here are some examples of other Liferay Marketplace plugins:

- Google Login: web.liferay.com/marketplace/-/mp/application/31419296
- Social Login for Liferay: web.liferay.com/marketplace/-/mp/application/41130997

Additional Options

This is not a comprehensive list, but a sample of some of the more popular options used by Liferay customers:

- OpenSSO
- Okta
- Oracle Access Manager
- Tivoli Access Manager
- OneLogin
- Azure Active Directory (AD)
- PingFederate

Using Systems That Are Not Supported Out-of-the-Box

You have the ability to use systems that are not supported out-of-the-box. The Liferay Workspace 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?
- 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`. You can overwrite this property 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.MyCustomAccessManagerAutoLogin
```

A suggested quick start is to simply copy one of the existing AutoLogin classes from the source code and use that as a template for your custom class (e.g., `MyCustomAccessManagerAutoLogin`).

What about other IdPs?

There are many other IdPs available. DXP has integration with any SAML compliant Identity Providers by serving as a Service Provider through the Liferay SAML app. You also have the freedom to integrate with any IdP by developing a Liferay app or module, though some implementations may be easier than others. In some cases, Liferay recommends creating a new auto-login module. You may look in the source code to use an existing class (e.g., `OpenSSO`) as a template to create a new auto-login class, especially if it is very close to the implementation you need. With other IdPs with other types of implementations, you can use other existing classes as templates or starting points to get to where you need in your new custom authentication class. More information can be found in Liferay documentation on [writing a custom login](#).

Summary

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 give you an initial IdM recommendation based on best practices and the experience of professionals working with Liferay customers. System architects and business analysts must consider the scenarios and business requirements that your system will need to address, conduct the appropriate tests on systems before production deployment and identify areas of significant risks.

Please use this document and the [Liferay Developer Network](#) for more detailed information on how to configure Liferay.

Moving Forward

Liferay Enterprise Subscription

A Liferay Enterprise Subscription provides customers with Liferay DXP, an enterprise-ready platform that helps companies create, manage and deliver digital experiences. The subscription is packaged with support, maintenance, a commercial service level agreement, legal assurances and more. Find out how a subscription can benefit your business at liferay.com/services.

Liferay Global Services

Liferay Global Services can help you in the design, planning, implementation, deployment and overall architecture of your system. Performance tuning, high availability, and cloud infrastructure consultation is also available.

Please contact sales@liferay.com for more information.



Liferay makes software that helps companies create digital experiences on web, mobile and connected devices. Our platform is open source, which makes it more reliable, innovative and secure. We try to leave a positive mark on the world through business and technology. Hundreds of organizations in financial services, healthcare, government, insurance, retail, manufacturing and multiple other industries use Liferay. Visit us at [liferay.com](https://www.liferay.com).

© 2018 Liferay, Inc. All rights reserved.