CILogon VO Management, Computing Cluster, and CLI Integration Patterns

Version 1.1

Overview

CILogon leverages COmanage Registry (COmanage) for identity registry, group registry, and user lifecycle management for virtual organizations (VOs). COmanage is a platform for enrolling and managing federated identities a user brings to a VO.

As an identity registry COmanage manages information about users including name, email, usernames and other identifiers. As a group registry COmanage manages group memberships and ownerships. In addition to these standard registry objects, COmanage includes two additional objects useful for managing access to computing resources:

1. **Authenticators**: A COmanage Authenticator is used to prove a user's identity to an application or service. Although COmanage is designed for consuming and then managing external, federated identities, there are a number of use cases where it makes sense for VO managed credentials to be used to access resources. Examples of COmanage Authenticators include SSH keys, passwords, X.509 certificates, and MFA tokens.

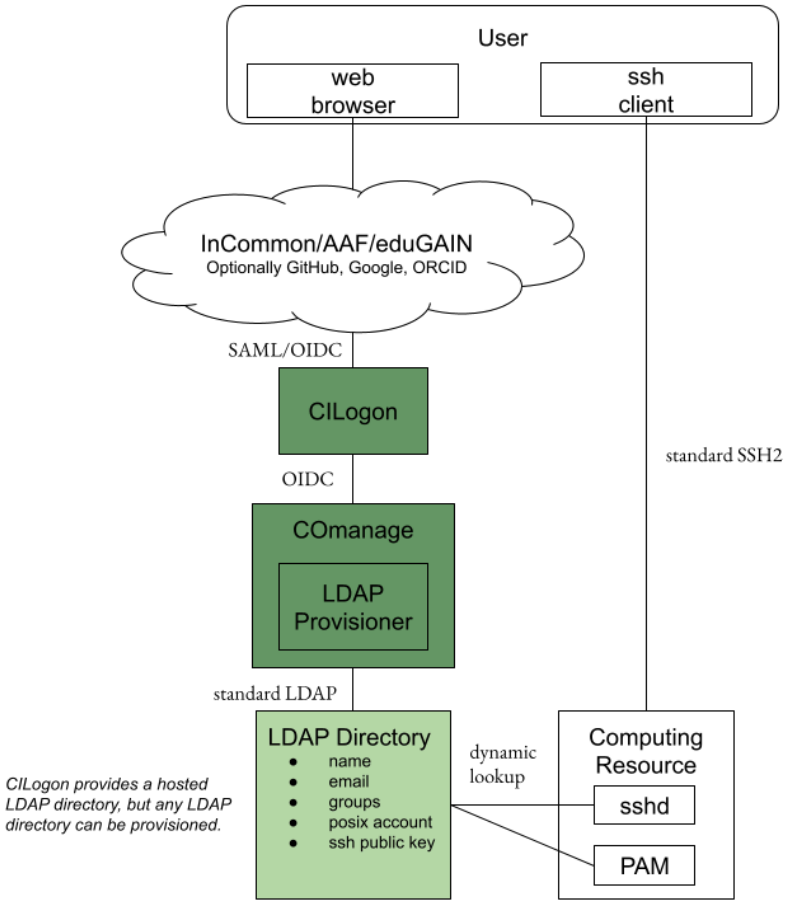
2. **Clusters**: A COmanage Cluster represents a user's accounts within a given application or service. The typical use case is managing accounts on one or more computing servers using the Unix Cluster Plugin.

A user's Authenticator and Cluster objects, along with name, email, usernames, and other identifiers may be provisioned out of COmanage for consumption by and integration with external resources including computing resources.

Conversely, existing identity information about users can be *consumed* by COmanage. This is especially useful when integrating with existing computing resources with their own identity management systems and processes. Each integration is unique and addressed using a combination of Organization Identity Sources, Job Plugins, and Enrollment Flow Plugins.

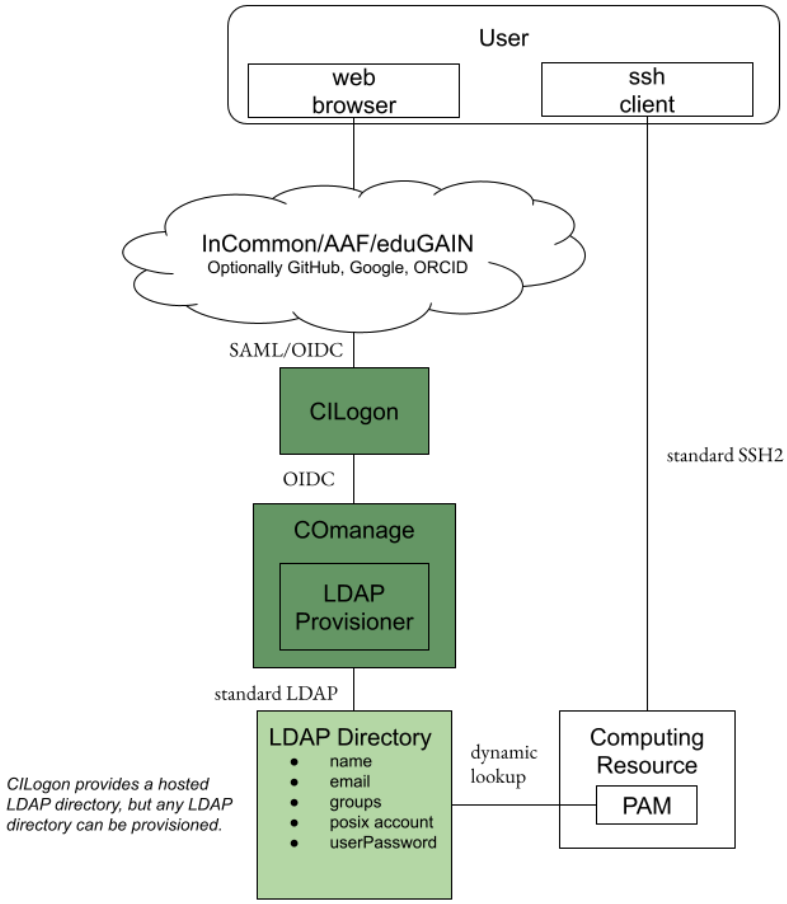
CILogon Integration Patterns

1 SSH Keys and Posix Account Using an LDAP Directory



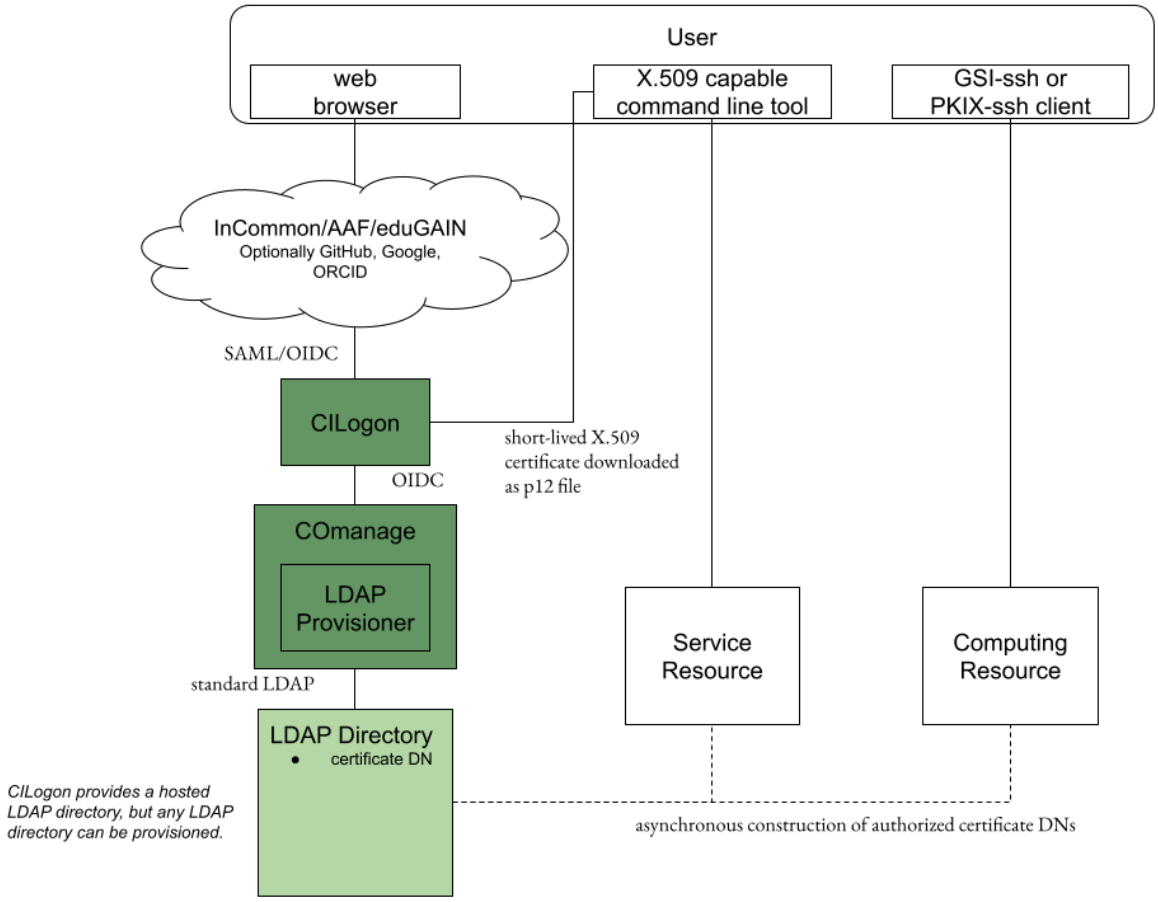
1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. Either during or after enrollment the user uploads one or more SSH public keys.
3. COmanage provisions the user record including name, email, group memberships, username and other posix account details, and the SSH key(s) to one or more LDAP directories. CILogon provides a hosted LDAP directory but any standard LDAP directory may be provisioned.
4. Services on the computing resource, often sshd and PAM, query the LDAP directory dynamically to authenticate the user and obtain account information.

2 Password and Posix Account Using an LDAP Directory



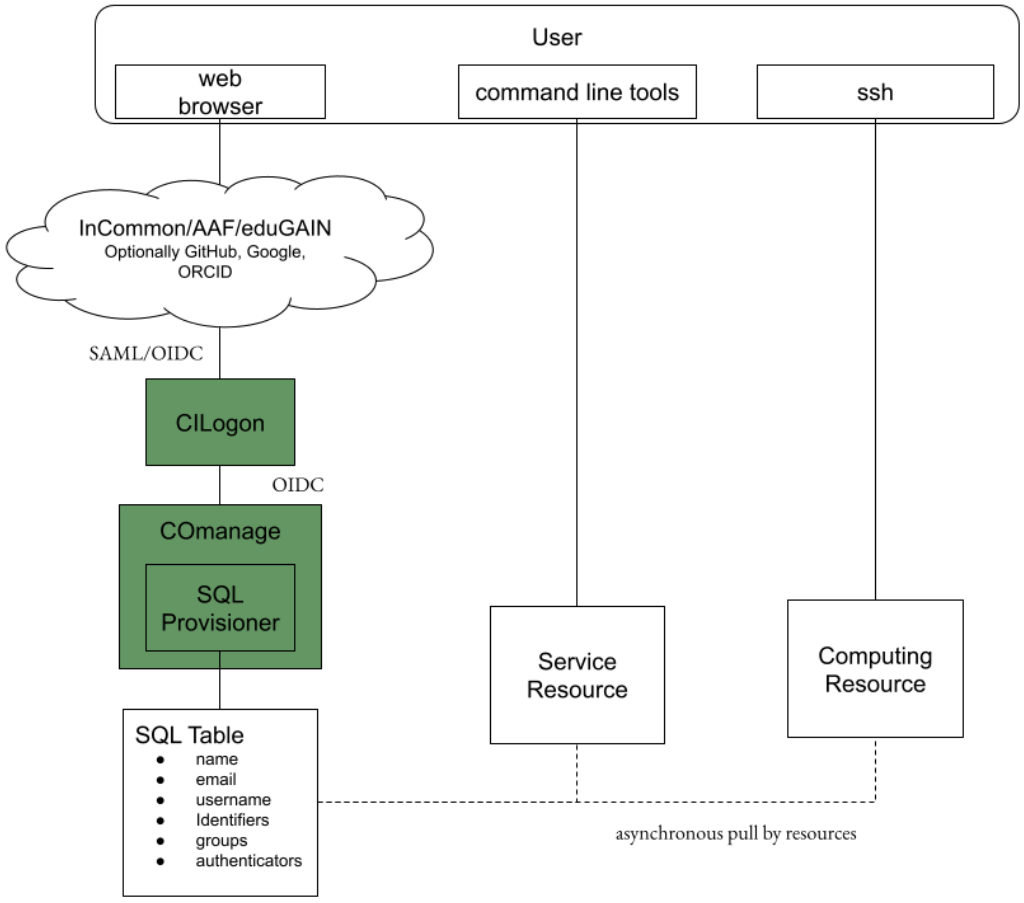
1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. Either during or after enrollment the user sets a Password Authenticator in COmanage.
3. COmanage provisions the user record including name, email, group memberships, username and other posix account details, and the userPassword attribute to one or more LDAP directories. CILogon provides a hosted LDAP directory but any standard LDAP directory may be provisioned.
4. PAM on the computing resource is configured to query the LDAP directory dynamically to authenticate the user and obtain account information.

3 Legacy X.509 Certificates



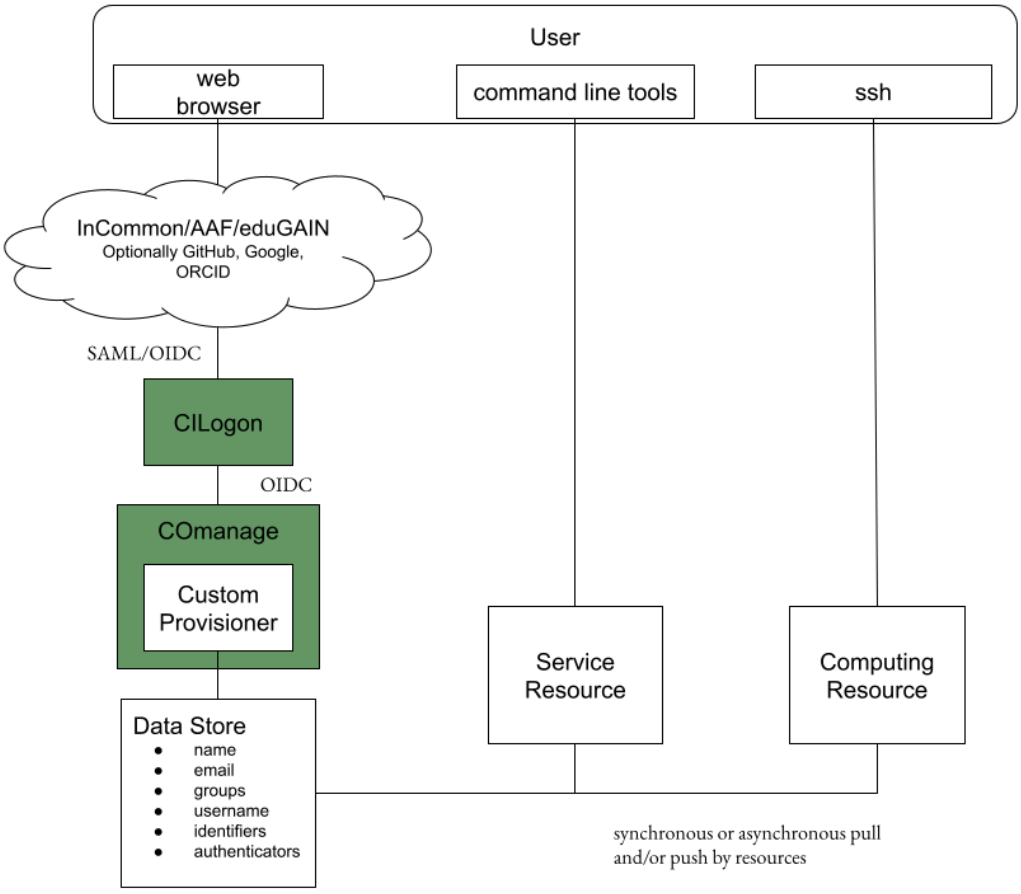
1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. During enrollment COmanage creates a Certificate Authenticator object and records in it the certificate DN the CILogon certificate authority (CA) will issue for the user.
3. COmanage provisions the DN along with other user information to the user record in one or more LDAP directories. CILogon provides a hosted LDAP directory but any standard LDAP directory may be provisioned.
4. Later a user again authenticates using a federated identity to CILogon and downloads a short-lived, password-protected, X.509 certificate in p12 format. The certificate is downloaded to the user's desktop as a file using the web browser.
5. Services and computing resources query the LDAP directory to construct the list of authorized certificate DNs. Non-standard SSH clients including GSI-SSH and PKIX-SSH may use the short-lived X.509 certificates to authenticate to capable sshd daemons.

4 Using a Staging Table



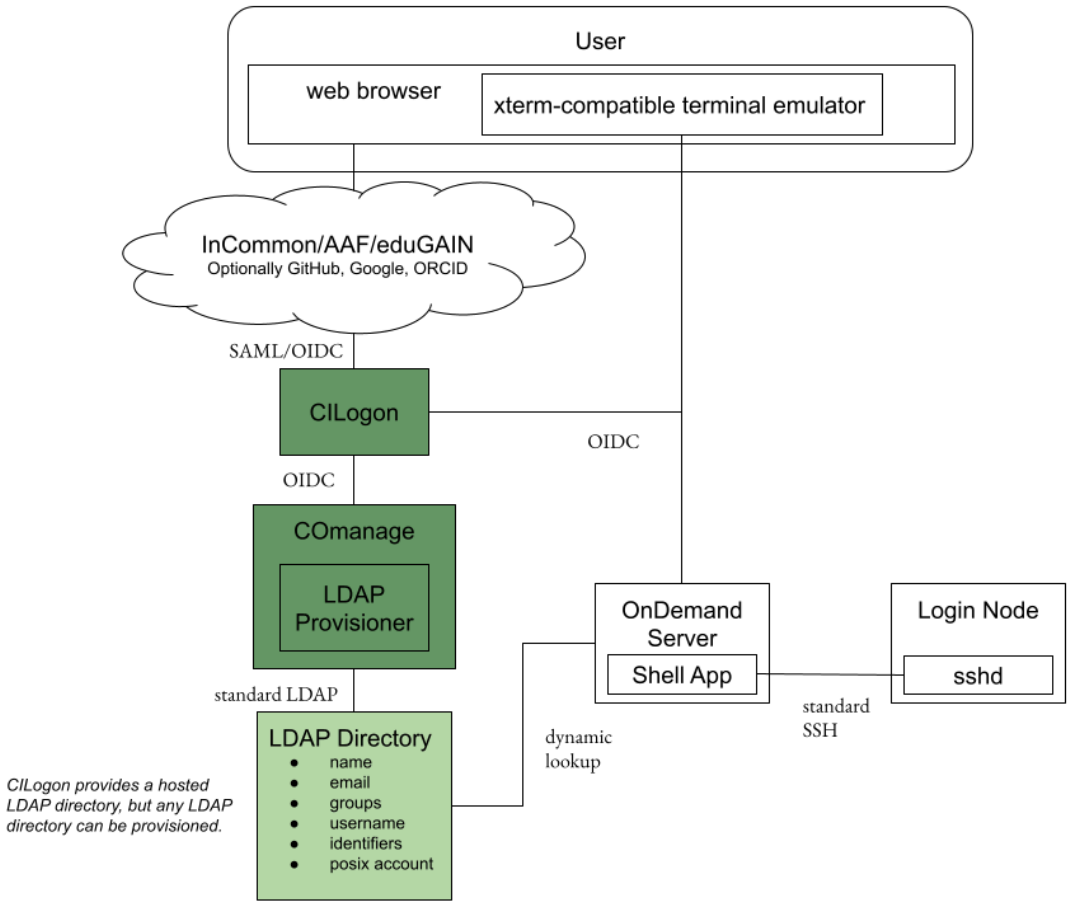
1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. Either during or after enrollment the user sets a Password Authenticator in and/or uploads an SSH key to COmanage
3. COmanage provisions the user record including name, email, group memberships, username, identifiers, and authenticators to one or more SQL tables.
4. Services and computing resources asynchronously pull user records from the SQL table and provision accounts and authorize access as necessary.

5 Custom Provisioner and Data Store



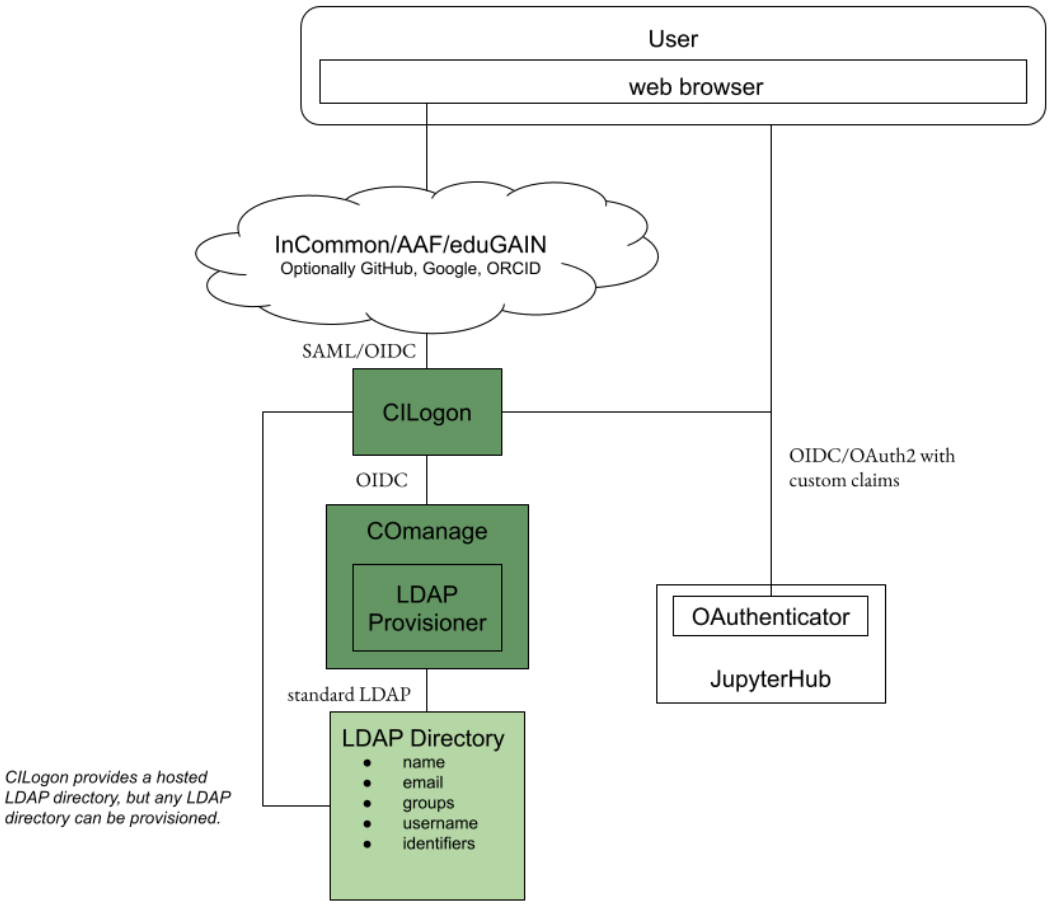
1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. Either during or after enrollment the user sets a Password Authenticator in and/or uploads an SSH key to COmanage.
3. One or more custom COmanage Provisioning plugins provision the user record including name, email, group memberships, username, identifiers, and authenticators to one or more data stores supported by the custom provisioners. Provisioning plugins are written in PHP. Common data stores include key-value databases such as Redis and LMDB, service infrastructures including Google Workspace and Microsoft 365, and other provisioning engines.
4. Services and computing resources synchronously or asynchronously push and/or pull user records from the data store and provision accounts and authorize access as necessary.

6 OPEN OnDemand



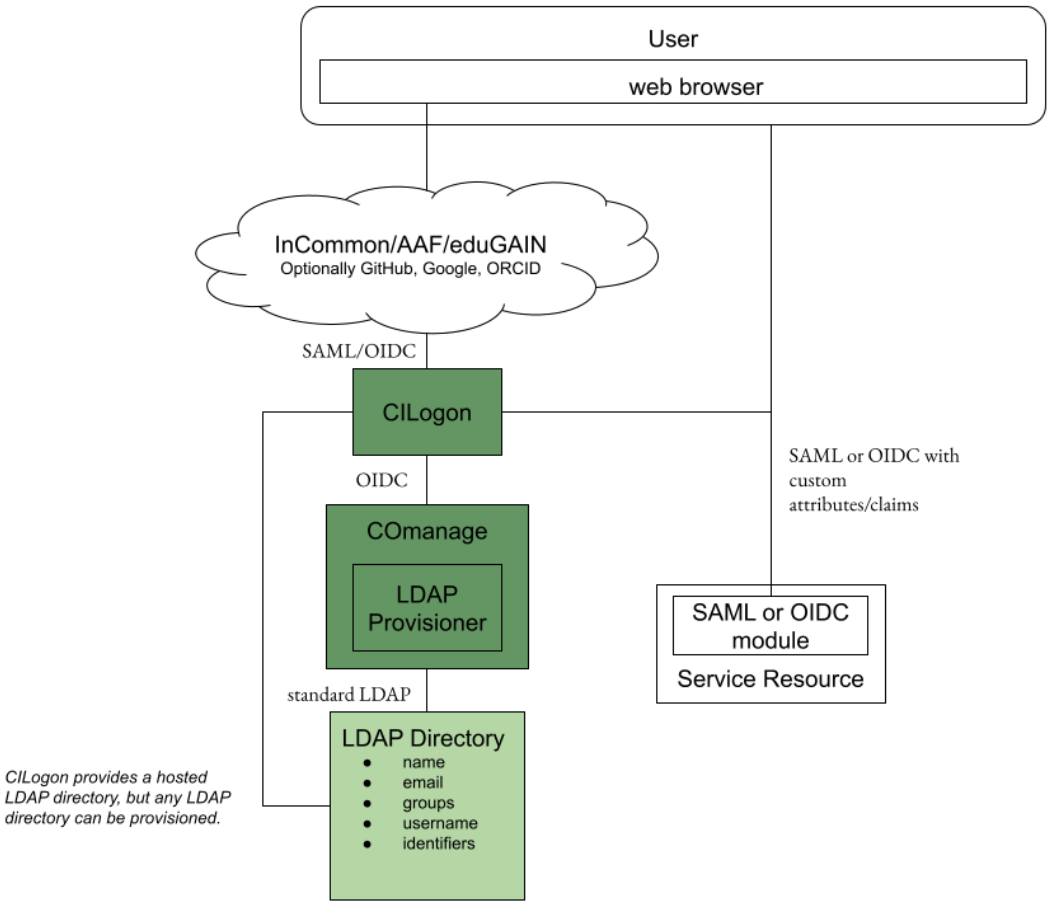
1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. COmanage provisions the user record including name, email, group memberships, username, identifiers, and posix account details to one or more LDAP directories. CILogon provides a hosted LDAP directory but any standard LDAP directory may be provisioned.
3. After enrollment the user browses to the Open OnDemand Server and authenticates using the registered federated identity and OIDC (perhaps experiencing SSO, depending on session timing).
4. The user accesses the OnDemand Shell App using an xterm-compatible terminal emulator written entirely in JavaScript (using the Google client hterm).
5. The OnDemand Server dynamically queries the LDAP directory after authenticating the user for account details. The ShellApp forks an ssh process that connects to the server specified by the OnDemand administrator (typically a login node).

7 JupyterHub Authenticator Using OAuth2



1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. COmanage provisions the user record including name, email, group memberships, username, and identifiers to one or more LDAP directories. CILogon provides a hosted LDAP directory but any standard LDAP directory may be provisioned.
3. After enrollment the user browses to the JupyterHub server configured with the OAuthenticator and authenticates using the registered federated identity and OIDC (perhaps experiencing SSO, depending on session timing).
4. During the OIDC flow to the JuptyerHub server the CILogon proxy queries the LDAP directory for user details including custom identifiers and groups and includes them as custom claims. The user details from the LDAP directory may augment or override claims asserted by the upstream campus identity provider.

8 Generic SAML and OIDC



1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. COmanage provisions the user record including name, email, group memberships, username, and identifiers to one or more LDAP directories. CILogon provides a hosted LDAP directory but any standard LDAP directory may be provisioned.
3. After enrollment the user browses to a service with SAML or OIDC authentication capabilities and authenticates using the registered federated identity and either SAML or OIDC (perhaps experiencing SSO, depending on session timing).
4. During the authentication flow to the service the CILogon proxy queries the LDAP directory for user details including custom identifiers and groups and includes them as custom SAML attributes or OIDC claims. The user details from the LDAP directory may augment or override attributes and claims asserted by the upstream campus identity provider.

Future and Ongoing Integration Patterns

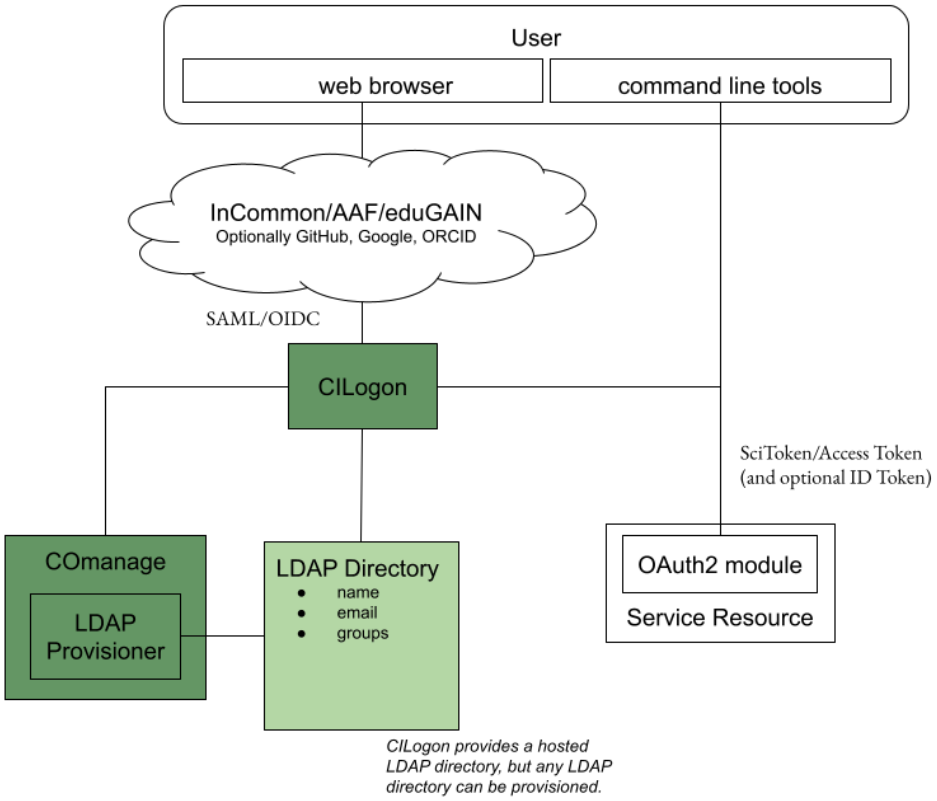
The following are forward looking integration patterns based on ongoing research and collaborations undertaken by members of the CILogon team and external collaborators. Please contact CILogon at help@cilogon.org if you are interested in exploring these integration patterns with CILogon.

9 SciTokens

SciTokens are signed JSON Web Tokens (JWT) issued by an authorization server that tools, including command line tools, can present to a service as a bearer token. The SciToken focuses on capabilities the bearer should have at the service rather than the identity of the bearer. This frees services from needing to duplicate authorization policies based on identity mapping.

CILogon is collaborating with Fermilab and the Laser Interferometer Gravitational-wave Observatory (LIGO) to implement SciToken access to resources via the command line.

The diagram below details the specific use case for using a SciToken with a capable command line tool and the OAuth2 Device Flow to access a service.



1. Using a web browser a user authenticates using a federated identity and completes an enrollment flow to onboard into COmanage Registry.
2. COmanage provisions the user record including name, email, and group memberships to one or more LDAP directories.
3. After enrollment the user uses a capable command line tool to begin an OAuth2 device flow.
4. CILogon responds with a URL communicated to the command line tool that the user is to use with a web browser to authenticate using the enrolled federated identity.
5. After the user authenticates, CILogon completes a policy and authorization check by querying the LDAP directory for user group membership information.
6. If the user belongs to the correct authorization group(s), CILogon returns a web page with a simple, short-lived code the user enters into the command line tool.
7. The command line uses the code to obtain a SciToken from CILogon, and then POSTs the SciToken to the service.
8. The service verifies the signature on the SciToken and inspects it for the capability(ies) needed for access to the service resource(s). Optionally the SciToken may include an ID Token if the service requires details about the user in addition to capability authorization. The use of refresh tokens is not depicted.

10 GA4GH Passports

A GA4GH Passport is a GA4GH-compatible access token conforming to the GA4GH AAI specification. Much like a SciToken, a GA4GH Passport is a signed JSON Web Token that includes claims that signal to consuming services (Claim Clearinghouses and Data Holders) that a user is authorized to access specific resources, and in particular data sets.

CILogon is interested in collaborating with researchers and infrastructure providers to map the CILogon VO Collaboration Management suite of services onto the GA4GN AAI specification and understand how the existing CILogon infrastructure for managing access to resources through SciTokens can be adapted to managing access to data through GA4GH Passports.