SSO and OAuth in Decentralization

“Let's talk about authentication in the Decentralized Ezekiel. First, I want to confirm that single source sign-on or single sign-on is equivalent to a form of OAuth.”

00:16

In a decentralized context like Acequia, single sign-on can indeed be implemented using OAuth or a similar protocol. OAuth provides a way to authorize users through a trusted identity provider without exposing their credentials to multiple services. By using OAuth, you can achieve a form of single sign-on where users authenticate once with their identity provider and then access multiple decentralized services seamlessly. It simplifies user experience and enhances security by centralizing authentication.

“Okay, this word centralized. So what we're doing is making it very easy for a user to set up a domain or subdomain from which they will provide tokens via OAuth. And so they can create their own group for which they are, you can say centralized, but we're making it very easy to make many, many OAuth services.”

00:22

Got it. So you're essentially

“Go ahead.”

00:00

So you're distributing the ability to create OAuth services across multiple domains or subdomains, making it easy for users to set up and manage their own groups. Each of these groups acts as its own identity provider. In that sense, it's decentralized because you're not relying on a single, monolithic provider. Instead, you're enabling a multitude of smaller, user-managed providers. This approach gives users more control over their authentication systems while still leveraging OAuth for security and interoperability.

“and even if they're just a group of one on their own domain. Now, isn't that the same as self-sovereign identity?”

00:07

Exactly. If someone sets up their own identity provider for just themselves, it closely aligns with the concept of self-sovereign identity. They control their own credentials and how they share them, without relying on a third-party authority. This gives them full autonomy over their digital identity, which is a core principle of self-sovereign identity.

“And would they then sign a token in the OAuth process and provide their public key for that domain?”

00:09

Yes, exactly. During the OAuth process, they would sign the token with their private key. The relying party (the service they're trying to access) can then verify the token using the public key associated with that domain. This ensures that the token is authentic and hasn't been tampered with. It’s a robust way to manage identity and access control in a decentralized environment.

“And so, how does the authenticator know that that is the right public key? Is there some place where they go for the public key to verify against that domain, some URL?”

00:12

Yes, typically the public key is hosted at a well-known URL on that domain. This is often part of an OpenID Connect or OAuth 2.0 configuration. The relying party can fetch the public key from that URL and use it to verify the signature on the token. This way, the authenticity of the token can be confirmed without needing a centralized authority.

“And this is different than DIDs, right?”

00:02

Right. Decentralized Identifiers, or DIDs, take it a step further by not relying on domain names or centralized infrastructure at all. Instead, DIDs are often tied to blockchain or other decentralized networks, and the public keys can be resolved through these systems. So while OAuth with a custom domain still depends on the DNS system, DIDs are designed to be entirely decentralized and independent of traditional web infrastructure.

“OK, so we've listed OAuth and OpenID. List all the other potential protocols or standards that we should evaluate and decide upon.”

00:11

Sure. Alongside OAuth and OpenID Connect, you might also consider:

1. **SAML (Security Assertion Markup Language)**: This is commonly used for enterprise single sign-on scenarios. It’s an XML-based standard that provides for the exchange of

authentication and authorization data between parties, especially between identity providers and service providers.

2. **LDAP (Lightweight Directory Access Protocol)**: This is a protocol for accessing

“I cut you off. I had a phone call. I'll continue.”

00:02

No problem! Let’s continue with the list:

2. **LDAP (Lightweight Directory Access Protocol)**: This protocol is often used for directory services authentication, especially in enterprise environments. It’s not as commonly used for web-based single sign-on as OAuth or SAML, but it can be part of an authentication strategy.
3. **Kerberos**: A network authentication protocol designed to provide strong authentication for client/server applications. It’s commonly used in environments like Active Directory.
4. **FIDO2 and WebAuthn**: These are modern standards for passwordless authentication using hardware tokens or biometrics. They’re highly secure and user-friendly.
5. **DIDs (Decentralized Identifiers)**: As we mentioned, these provide a way to verify identities without relying on traditional centralized systems, often leveraging blockchain or other decentralized networks.

Evaluating these options depends on your specific use case, security requirements, and the level of decentralization you want to achieve.

“Okay, I'm working at Harvard, and I have, everyone logs in with a Harvard H-U-I-T-I-D and uses Duo for two-factor. And then it's all a lot of Microsoft Outlook Exchange, but I believe that's going against the HUID ID. Well, what do you think?”

00:20

That setup sounds like a combination of enterprise-level identity management systems. The Harvard University ID (HUID) and Duo for two-factor authentication are integrated with Microsoft’s ecosystem, likely through Azure Active Directory. This setup allows users to authenticate once with their HUID and then access a range of services, including Outlook and other Microsoft 365 tools. It’s a robust and common solution in large institutions, providing strong security and seamless integration.

“And continuing on that, I also used my HUIT for Hollis, which is the Harvard library. So what is, yeah, so say more what you

think about that."

00:12

In that case, your Harvard identity is serving as a single source of authentication across multiple services, including email, library access, and other university resources. This is a classic example of single sign-on (SSO) within an enterprise environment. By leveraging a central identity provider, like Harvard University IT, you streamline access management, enhance security with multi-factor authentication like Duo, and provide a seamless user experience across various systems, whether it's for email, academic resources, or library services. It's a strong, integrated approach to identity

"And I'll just add another one. I have a full Harvard ArcGIS Online account. Now, that does give me a short name, STG007, which I believe is probably in the ESRI space. And then I have a different H-U-I-T. But I use the ESRI short name for sharing inside that ecology. But I only authenticate with my Harvard ID."

00:26

That's a great example of how federated identity works across different systems. Your Harvard ID serves as the primary authentication method, allowing you to access your ArcGIS Online account. Within the Esri ecosystem, you have a separate identifier that's used for sharing and collaborating within that environment. This setup allows you to maintain a single trusted identity for logging in, while still having the flexibility to use different usernames or handles within various applications. It's a practical way to manage identities across multiple platforms while keeping security and convenience in check.

"And notice here that the authentication is different than authorization. So, in some ways, I authenticate with the Harvard, but as I authorize access to my shared layers and data in ESRI and tile servers, I'm using the ESRI ID for authorization, but not authentication."

00:23

Exactly. Authentication is all about verifying who you are, while authorization determines what you can access and what actions you can perform. In your case, you authenticate through Harvard to prove your identity, but the authorization to share and manage resources within the Esri ecosystem is handled by your Esri ID. This separation allows for more granular control over permissions and access, ensuring that the right people have the right level of access to specific resources.

> *“Okay, so now just give me a basic workflow where SimTable's software, the web-based software, is AnyHazard. And now I want to give single sign-on to Harvard faculty and students, but then probably use an ID for authorization within the AnyHazard and Ezekia system.”*

00:21

Sure. Here’s a basic workflow:

1. **Authentication**: When a Harvard faculty member or student tries to access the Simtable web-based software (AnyHazard), they are redirected to Harvard’s identity provider to authenticate with their HUID and Duo for two-factor authentication.
2. **Token Issuance**: After successful authentication, Harvard’s identity provider issues a token that the Simtable software can trust. This token confirms the user's identity.
3. **Authorization**: Once the user is authenticated, Simtable’s system can use its own internal ID to manage authorization—deciding what data and layers the user can access and what actions they can perform within AnyHazard.

This way, you leverage Harvard’s secure and trusted authentication system while maintaining control over the specific permissions and roles within your own application.

> *“So can we tie that to a persistent authentication token from Harvard that doesn't change, or at least there's some kind of unique identifier within that token that doesn't change?”*

00:11

Yes, you can use a persistent identifier within the token, often called a "subject identifier" or "sub" claim. This identifier remains consistent for the user across sessions and can be used to link their Harvard authentication to their profile in your system. This way, even if the token itself expires and needs to be refreshed, the unique identifier remains the same. This ensures that the user’s identity is consistently recognized across different sessions and services.

> *“So I can see where that token has an expiration, especially as students come to the end of the semester and no longer qualify for access.”*

00:09

Exactly. By having tokens that expire, you ensure that access is only granted to current, eligible users. If a student or faculty member’s status changes, such as at the end of a semester, the authentication process will reflect that, and they won't be able to get a new token. Meanwhile, the persistent identifier can remain the same to keep track of their historical data and permissions.

“Now we work with the National Interagency Fire Center and all the fire data, and at the federal level, there's two kinds of authentication, login.gov for civilians and then there's FAMOTH for federal workers. So, in the same way, we want to give any hazard authorization by authenticating via either people have a choice of login.gov or FAMOTH.”

00:26

Absolutely. You can set up a system where AnyHazard supports multiple identity providers. Users can choose to authenticate through login.gov if they are civilians, or through FAMAuth if they are federal workers. Once authenticated, AnyHazard can handle authorization internally based on the user’s role and the type of access they need. This approach provides flexibility and ensures that the right people can access the system using their appropriate credentials.

“Now, so when we list potential authorizations or authentication services, you know, Facebook and Google and X or GitHub are sometimes in the mix, too. What's best practice of tying a user identification where they can log in with multiple authentic, authentic, authentifications?”

00:26

A best practice is to link all these external authentication methods to a single internal user profile. This way, whether someone logs in with their Harvard ID, login.gov, FAMAuth, or even social logins like Google or GitHub, they are recognized as the same user in your system. This is usually done by matching a unique identifier, such as an email address or a federated ID, to the internal user record. This allows users to choose the most convenient authentication method while maintaining a unified user experience and consistent authorization within AnyHazard.

“So a real use case here is they have their photos and some files in Google Drive, and then they also have some of their layers on ArcGIS Online. So first I'll also double-check that we can use Esri as an authentication as well.”

00:17

Yes, Esri's ArcGIS Online can serve as an authentication provider. It supports SAML 2.0 and OpenID Connect, allowing users to log in using their existing enterprise credentials. This integration streamlines access to GIS resources and enhances security.

esri Sources

“OK, of all the Authenticators we listed, Harvard, Esri, Google, the social ones, make a quick little list of the Authenticators and the protocols they support.”

00:12

Sure! Here’s a quick list of the authenticators and the protocols they typically support:

1. **Harvard (HUIT):** Uses enterprise systems often integrated with **SAML** and possibly **OAuth** for single sign-on.
2. **Esri (ArcGIS Online):** Supports **SAML 2.0** and **OpenID Connect** for enterprise logins.
3. **Google**: Primarily uses **OAuth 2.0** and **OpenID Connect** for authentication.
4. **GitHub**: Uses **OAuth 2.0** for authentication and authorization.
5. **login.gov**: Supports **SAML 2.0** and **OpenID Connect** for civilian authentication to federal services.
6. **FAMAuth**: Typically uses **SAML 2.0** for federal employee authentication.

By supporting these protocols, you can offer flexible authentication options while maintaining a unified internal user profile and authorization framework.

“So, and I imagine there's many authentication services we can use that handles this for us, much like Stripe handles financial transactions. List the top five authentication platforms we can tie into.”

00:15

Absolutely. There are several robust authentication platforms that simplify integration, similar to how Stripe handles payments. Here are five popular options:

1. **Auth0**: A flexible platform supporting a wide range of protocols like OAuth, OpenID Connect, and SAML.
2. **Okta**: Known for its enterprise-grade identity management, supporting SAML, OAuth, and OpenID Connect.

3. **Azure Active Directory (Azure AD)**: Great for organizations already using Microsoft services, supporting SAML, OAuth, and OpenID Connect.
4. **Google Identity Platform**: Provides a straightforward way to authenticate users with Google accounts and supports OAuth and OpenID Connect.
5. **Amazon Cognito**: A scalable solution from AWS that supports user authentication through OAuth and OpenID Connect, and integrates well with other AWS services.

These platforms can handle the complexity of managing multiple authentication providers, making it easier for you to focus on your application logic and user experience.

“rank them by which one lets me go across as many as possible, especially Esri, AWS, Google, and Harvard.”

00:10

Sure, let's rank them by their ability to integrate with a wide range of providers, especially Esri, AWS, Google, and Harvard:

1. **Auth0**: Extremely flexible and known for its wide range of integrations, including SAML, OAuth, and OpenID Connect. It can connect to Esri, AWS, Google, and custom enterprise solutions like Harvard’s.
2. **Okta**: A leading identity provider with extensive support for various protocols and integrations. It can handle Esri, AWS, Google, and enterprise systems like Harvard’s.
3. **Azure Active Directory (Azure AD)**: Strong integration capabilities, especially for enterprise systems. It supports a broad range of

To cover as many authentication providers as possible, especially the ones you mentioned (Esri, AWS, Google, and Harvard), here’s how they generally rank in terms of flexibility and integration breadth:

1. **Auth0**: Known for its extensive integration capabilities, Auth0 can handle a wide range of identity providers, including Esri (via SAML or OpenID Connect), AWS (via Amazon Cognito), Google, and enterprise systems like Harvard’s.
2. **Okta**: Another highly versatile platform, Okta supports a broad array of identity providers and can integrate with Esri, AWS, Google, and institutional systems like Harvard’s.
3. **Azure Active Directory (Azure AD)**: It offers strong integration capabilities, especially for enterprise environments. It can integrate with Google, AWS, and other SAML/OIDC

providers like Esri and Harvard.

4. **Amazon Cognito**: While it integrates seamlessly with AWS services, it also supports federation with external identity providers like Google and other SAML or OIDC-based systems. Its integration with Esri and Harvard might require more configuration.
5. **Google Identity Platform**: Excellent for integrating with Google accounts and other OpenID Connect providers. It may require additional steps for seamless integration with Esri and enterprise systems like Harvard.

Auth0 and Okta are generally the most flexible and widely adopted solutions for handling a variety of identity providers, including those you mentioned.