EMAIL 1
Hello,
I'm new to SSO, and there are a lot of concepts I'm not educated on. To make my onboarding smoother, it'd help if you could provide me with some definitions of the following concepts:

- SAML
- IdP
- OAuth
- SDK

I'm also struggling when setting up my first enterprise customer. What is the "x509 Certificate"? Where does the user get redirected? Could you help me understand how it works?
Cheers,
James

EMAIL 1 RESPONSE

Hi James,

Happy to help out here! We offer a glossary that has definitions for these concepts and more that might pop up as you get more familiar with working with WorkOS. Let's dive into each of these points:

- SAML:
  A formal definition of Security Assertion Markup Language (SAML) is "an open standard XML protocol that allows users to sign in to multiple applications with one set of credentials."

  Web applications utilize SAML to persist authentication between the Identity Provider (IdP) and the Service Provider. You'll log in to your IdP to access your dashboard of applications and then have access to those services, such as AWS or Github. Okta dashboards are a good example, where you'll log in to Okta and then from the Okta dashboard access the various Service Providers that have been approved for use.

  SAML is what makes it possible to not have to log in to each of these Service Providers individually, instead passing the authentication to each service to streamline access.

  This cloudflare article is a great resource for learning more about SAML. They break it down particularly well:

  > *Think of SAML authentication as being like an identification card: a short, standardized way to show who someone is. Instead of, say, conducting a series of DNA tests to confirm someone's identity, it is possible to just glance at their ID card.*

Here are some additional resources that will help to gain a stronger understanding of SAML:

- https://www.onelogin.com/learn/saml
- https://auth0.com/blog/how-saml-authentication-works/
- https://duo.com/blog/the-beer-drinkers-guide-to-saml

- IdP:
  Identity Providers, mentioned a few times above, create, store, and manage digital identities. IdP's perform the authentication by checking usernames and passwords and granting access based on the user's roles and permissions that is logging in.

  Essentially, the IdP is what authenticates a user so that they can access the applications they need to. Some common IdPs you may have run into before are Google, Okta, and Microsoft Entra.

  Here are some additional resources to round out what an IdP is and how it works:
  - https://workos.com/docs/glossary/idp
  - https://www.cloudflare.com/learning/access-management/what-is-an-identity-provider/
  - https://www.okta.com/identity-101/why-your-company-needs-an-identity-provider/
  -

- OAuth:
  Formally, OAuth stands for Open Authorization and is a standard that applications can use to provide client applications with "secure delegated access". OAuth works over HTTPS and authorizes devices, APIs, servers, and applications with access tokens rather than credentials.

  OAuth allows you to provide consent so that one application can interact with another without having to reveal your password.

  This HackerNoon article has a great explanation in practical terms:

  > *Consider the most popular example, the valet key for your car. The vehicle owner's key can control everything in the car such as starting it, opening doors and windows, accessing the glove box, opening the trunk of the car etc. But the valley key can only be used to start the car and lock/unlock the doors. This is the concept behind OAuth. Providing a key with limited access rights.*

  Here are some additional resources as well for a deeper dive into OAuth:
  - https://developer.okta.com/blog/2017/06/21/what-the-heck-is-oauth
  - https://www.varonis.com/blog/what-is-oauth

- https://medium.com/@vikas.taank_40391/everything-that-you-need-to-know-about-oauth2-fb6a29b59e46

- SDK:
  SDKs, software development kits, are a collection of platform specific software building tools that are installable via one package. SDK's are used by developers to quickly ingest the tools they need so they can build out software and applications with the resources they need. Most SDK's offer documentation, libraries, sample code, and examples to help folks get started or give them direction. Since SDK's typically have pre built components it makes the building process quicker and more efficiently.

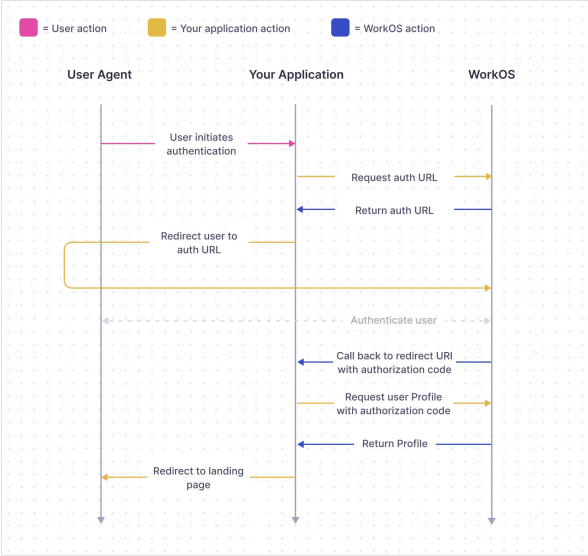
  Here are some additional resources to gain a better understanding of SDKs:
  - https://aws.amazon.com/what-is/sdk/
  - https://www.geeksforgeeks.org/what-is-software-development-kit-sdk/
  - https://getstream.io/glossary/api-vs-sdk/

> What is the "x509 Certificate"? Where does the user get redirected? Could you help me understand how it works?

An X.509 Certificate is a public key certificate used to authenticate SAML assertions. Sometimes referred to as Token Signature (AD FS). SAML signing certificates are X.509 certificates used in SAML responses to allow the Service Provider (SP) to verify the authenticity of a SAML response.

Here are some additional resources to gain a broader and deeper understanding about what a x509 Certificate is and how it works:
- https://emudhra.com/blog/x509-certificate
- https://www.sectigo.com/resource-library/what-is-x509-certificate
- https://darutk.medium.com/illustrated-x-509-certificate-84aece2c5c2e

The Redirect URI, a required, allowlisted callback URL, is the location to which the user gets returned to after successfully completing the authentication with their Identity Provider (IdP). You will set this up in your Redirects page of your WorkOS dashboard. There should be at least one redirect URI configured and selected as a default for a WorkOS Environment. How it works is that your user will sign in via their IdP or WorkOS service provider and go through authentication, upon confirmation of the authentication they will be redirected to the specified Redirect URI. This diagram may also help to shed some light on how it works:



Let me know if I can offer any more explanation, clarify anything further, or if I can help in any other way!

All the best,
James

EMAIL 2:

Hi,
I was checking out this blog post about “passport-saml”
https://medium.com/@athiththan11/saml-sso-with-passport-1690d2c38921
After reading it, I’m still not convinced that using WorkOS is the right approach. It may be because of my engineering background, but using an open source package seems like a better solution for us. This stuff doesn’t seem so hard for us to build.
Am I missing something here? Can you convince me of the opposite?
Talk soon!
Julia

EMAIL 2 RESPONSE:

Hi there Julia,

Thanks for writing in, and completely hear you on wanting to figure out what the best approach is here for you. Using an open source package can certainly have benefits, but may require that you spend more time building and maintaining your own solution rather than quickly getting up and running with an enterprise ready application. Passport.js is a good solution in early stages, but when demand for enterprise support grows, the manual provisioning of SSO connections with the customers IdP becomes unsustainable.

For example, it took Stack Overflow engineers roughly 3 months to build out an SSO solution. Instead of working on new features, squashing bugs, or optimizing your application you may find yourself spending just as much time working to maintain your inhouse SSO SAML solution.

If you do go ahead with Passport.js, but find yourself in a situation where you need to scale, are spending too many resources on maintaining the solution, or any other reason that may pop up, we have this helpful guide on migrating from Passport.js to WorkOS.

With workOS you can get up and running in a few minutes with minimal amount of code, and offers a friendly starting price point for developers depending on their needs. You’ll only need to integrate once and from there you can start supporting SAML with your chosen IdP. I’d recommend taking a read of our Developers Guide to refresh yourself with an overview of SSO and to choose the best way to add it to your app.

Here are some additional resources that may help you make your decision when it comes to choosing between an open source package versus using WorkOS:

- https://workos.com/blog/a-guide-to-organization-modeling
- https://workos.com/blog/the-developers-guide-to-user-management
- https://stackshare.io/stackups/passport-vs-workos

Additionally, I'd also recommend diving into the Customer Stories from users who have chosen to utilize WorkOS for their organization's needs. Lastly, with WorkOS you get access to our support team to help out with any issues that may arise so we can unblock you from problems as quickly as possible.

If there is anything else I can do to help make your decision easier do let me know. Also, if you have any other specific concerns about using WorkOS I'd be happy to address those as well.

Looking forward to hearing back from you about which direction you choose to go!

All the best,
James