Single Sign-On (SSO)

Single sign-on (SSO) is a session and user authentication service that permits a user to use one set of login credentials -- for example, a username and password -- to access multiple applications. SSO is built on a concept called federated identity.

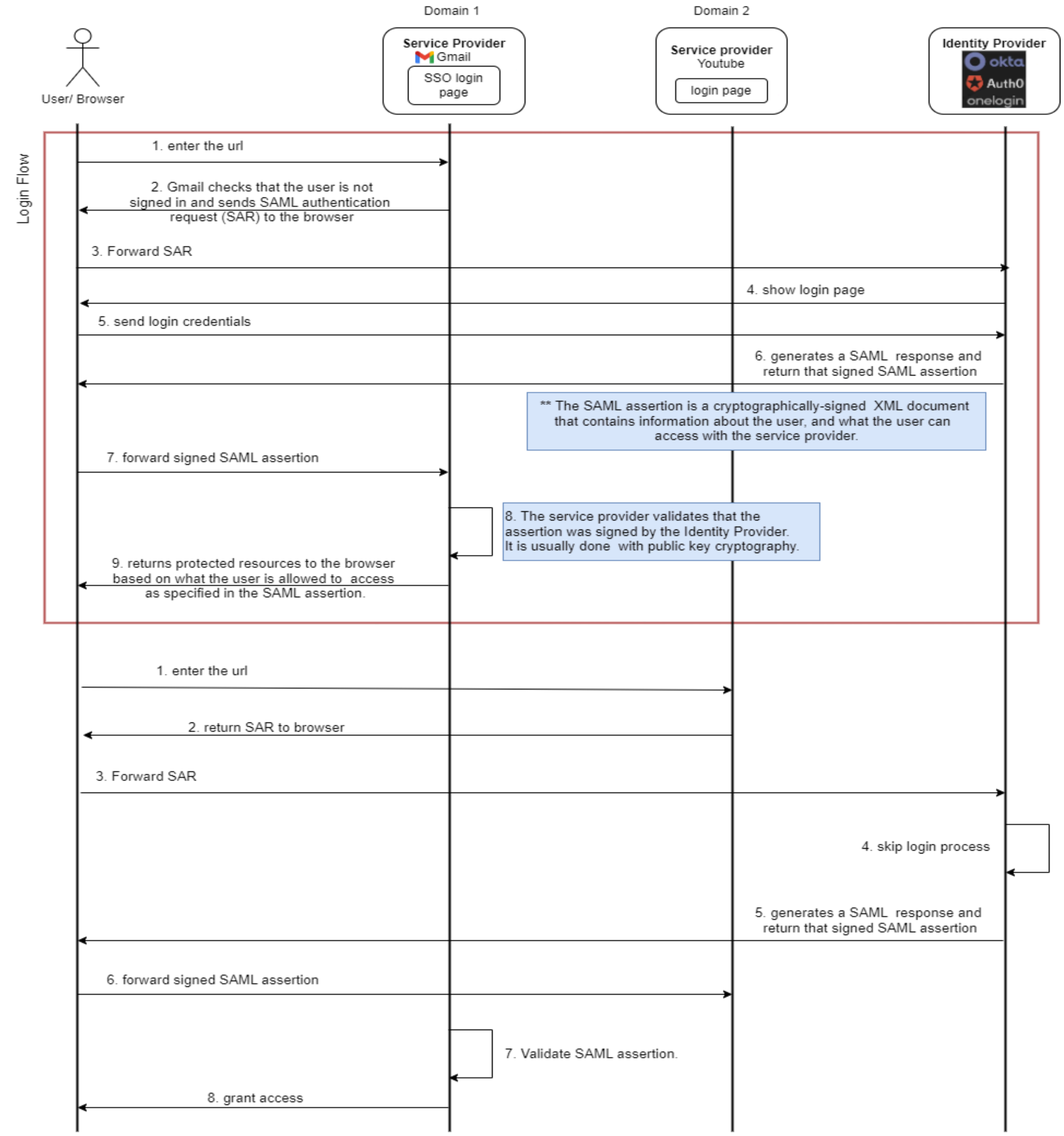
Federated identity management (FIM) is an arrangement between multiple enterprises or domains that enables their users to use the same identification data (digital identity) to access all their networks. It can be an organization, a business unit, a smaller subsidiary of a larger organization, etc.

There are two common protocols for this authentication process.

SAML, or Security assertion markup language, is an XML-based open standard for exchanging identity information between services.
The other common protocol is **OpenID Connec**t. It uses JWT, or JSON Web Token, to share identity information between services.

How does single sign-on work?

First, let’s focus on SAML. Lo


Domain 1
Domain 2
User/ Browser
Service Provider
Gmail
SSO login page
Service provider
Youtube
login page
Identity Provider
okta
Auth0
onelogin
Login Flow
1. enter the url
2. Gmail checks that the user is not signed in and sends SAML authentication request (SAR) to the browser
3. Forward SAR
4. show login page
5. send login credentials
6. generates a SAML response and return that signed SAML assertion
7. forward signed SAML assertion
9. returns protected resources to the browser based on what the user is allowed to access as specified in the SAML assertion.
1. enter the url
2. return SAR to browser
3. Forward SAR
4. skip login process
5. generates a SAML response and return that signed SAML assertion
6. forward signed SAML assertion
7. Validate SAML assertion.
8. grant access

** The SAML assertion is a cryptographically-signed XML document that contains information about the user, and what the user can access with the service provider.

8. The service provider validates that the assertion was signed by the Identity Provider. It is usually done with public key cryptography.

The OpenID Connect flow is similar to SAML, but instead of passing signed XML documents around, OpenID Connect passes around JWT. The implementation details are a little bit different, but the overall concept is similar.

Which one of these SSO methods should we use?

Both implementations are **secure**. For an enterprise environment where it is common to outsource identity management to a commercial identity platform, the good news is that many of these platforms provide strong support for both.
The decision then depends on the **application** being integrated and which protocol is easier to **integrate** with. If we are writing a new web application, integrating with some of the more popular OpenID Connect platforms like Google, Facebook, and Github is probably a safe bet.

SSO security risks

Although single sign-on is a convenience to users, it presents risks to enterprise security. An attacker who gains control over a user's SSO credentials is granted access to every application the user has rights to, increasing the amount of potential damage.

To avoid malicious access, SSO should be coupled with identity governance. Organizations can also use two-factor authentication (2FA) or multifactor authentication with SSO to improve security.

SSO advantages and disadvantages & SSO vendors

https://www.techtarget.com/searchsecurity/definition/single-sign-on

Web Security

https://portswigger.net/web-security/all-topics
You'll get necessary topics in this link. Just go through definitions, use case/example, and how to prevent these attacks.

Server-side Topics:

- SQL Injection
- Authentication
- Server-side Request Forgery
- API Testing

Client-side Topics:

- CSRF
- CORS
- XSS

CSRF Attack

https://www.blackduck.com/glossary/what-is-csrf.html#:~:text=A%20CSRF%20attack%20exploits%20a,a%20user%20without%20their%20consent.
https://owasp.org/www-community/attacks/csrf

XSS Attack

https://www.blackduck.com/glossary/what-is-cross-site-scripting.html#:~:text=Cross%2Dsite%20scripting%20(XSS)%20is%20an%20attack%20in%20which,a%20trusted%20application%20or%20website.
https://owasp.org/www-community/attacks/xss/

Differences:
https://www.wallarm.com/what/what-is-the-difference-between-csrf-and-xss#:~:text=XSS%20is%20a%20two%2Dway,while%20CSRF%20is%20HTTP%2Dbased.

HTTPOnly Flag

Facebook oi cookie ta HTTPOnly kore dile cookie ta facebook server baade onno kothao theke access kora jabe na. Also kono javascript code diyeo access kora jabe na, bcz javascript client side e run kore.

HTTPOnly kore dile cookie ta facebook server chara onno kono server jehetu access korte parbe na, tahole amra facebook e onno kichu diye login korte parbo na, like login with google.

Secure Flag

HTTPOnly te same domain na hoile cookie pathaboi na. Kintu ekhane same domain hoileo pathabo na, jodi server er HTTPS ba SSL certificate na thake.