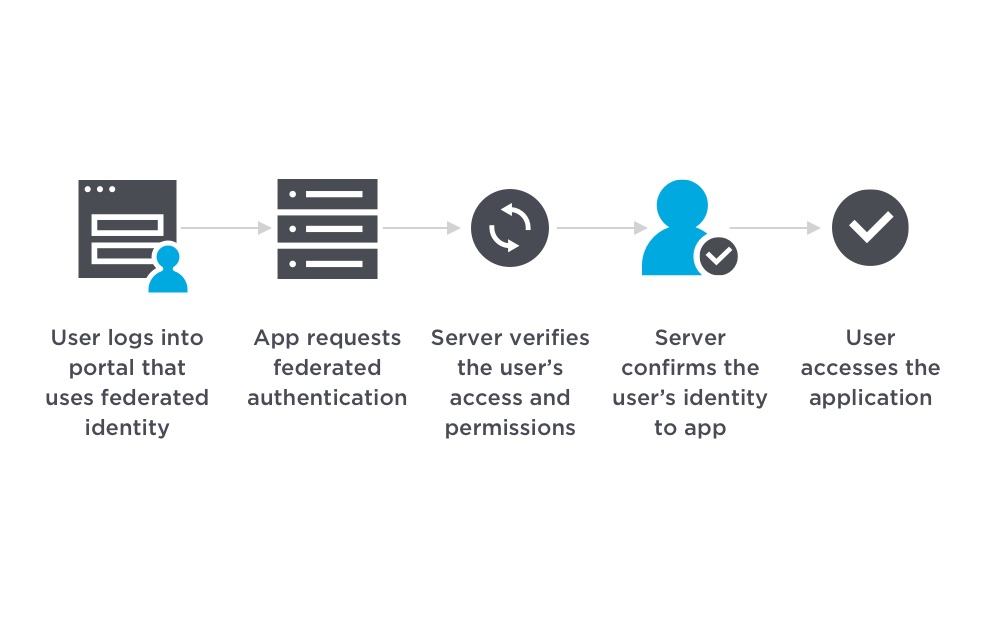
**Federated Identity & Single Sign On (SSO)**

**Federated identity:** also known as Federated Identity Management (FIM) – works on the basis of mutual trust relationships between a Service Provider (SP) such as an application vendor and an external party or Identity Provider (IdP). The IdP creates and manages user credentials and the SP and IdP agree on an authentication process. Multiple SPs can participate in a federated identity agreement with a single IdP. The IdP has mutual trust agreements with all these organizations.



**Single Sign On (SSO)** is an authentication scheme where users can securely authenticate and gain access to multiple applications and websites by only logging in with a single username and password.

For example, logging in to your Google account once will allow you to access Google applications such as Google Docs, Gmail, and Google Drive.

Diagram

Description automatically generated

Without SSO solution, the website maintains a database of login credentials – username and passwords. Each time the user login to the website, it checks the user’s credentials against its database and authenticates the user.

With the SSO solution, the website does not store login credentials in its database. Instead, SSO makes use of a shared cluster of authentication servers where users are only required to enter their login credentials once for authentication. With this feature of one login and multiple access, it is crucial to protect login credentials in SSO systems.  
Hence it is highly recommended to integrate SSO with other strong authentication means such as smart tokens or one-time passwords to achieve multi-factor authentication.

**Working of SSO**

1. User enters login credentials on the website and the website checks to see if the user has already been authenticated by SSO solution. If so, the SSO solution would give the user access to the website. Otherwise, it presents the user with the SSO solution for login.
2. The user enters username and password on the SSO solution.
3. The user’s login credentials are sent to SSO solution.
4. The SSO solution seeks authentication from the identity provider, such as an Active Directory, to verify the user’s identity. Once the user’s identity is verified, the identity provider sends a verification to the SSO solution.
5. The authentication information is passed from the SSO solution to the website where the user will be granted access to the website.
6. Upon successful login with SSO, the website passes authentication data in the form of tokens as a form of verification that the user is authenticated as the user navigates to a different application or web page.

Diagram, timeline

Description automatically generated

**Advantages of SSO :**

These are advantages for users, for businesses.

**For Users –**

* Risk of access to 3rd party sites are mitigated as the website database do not store the user’s login credentials.
* Increased convenience for users as they only need to remember and key in login information once.
* Increased security assurance for users as website owners do not store login credentials.

**For Businesses –**

* Increase customer base and satisfaction as SSO provides lower barrier to entry and seamless user experience.
* Reduce IT costs for managing customer’s username and passwords.

**Disadvantages of SSO :**

* Increased security risk if login credentials are not securely protected and are exposed or stolen as adversaries can now access many websites and applications with a single credential.
* Authentication systems must have high availability as loss of availability can lead to denial of service for applications using a shared cluster of authentication

**SAML Authentication**

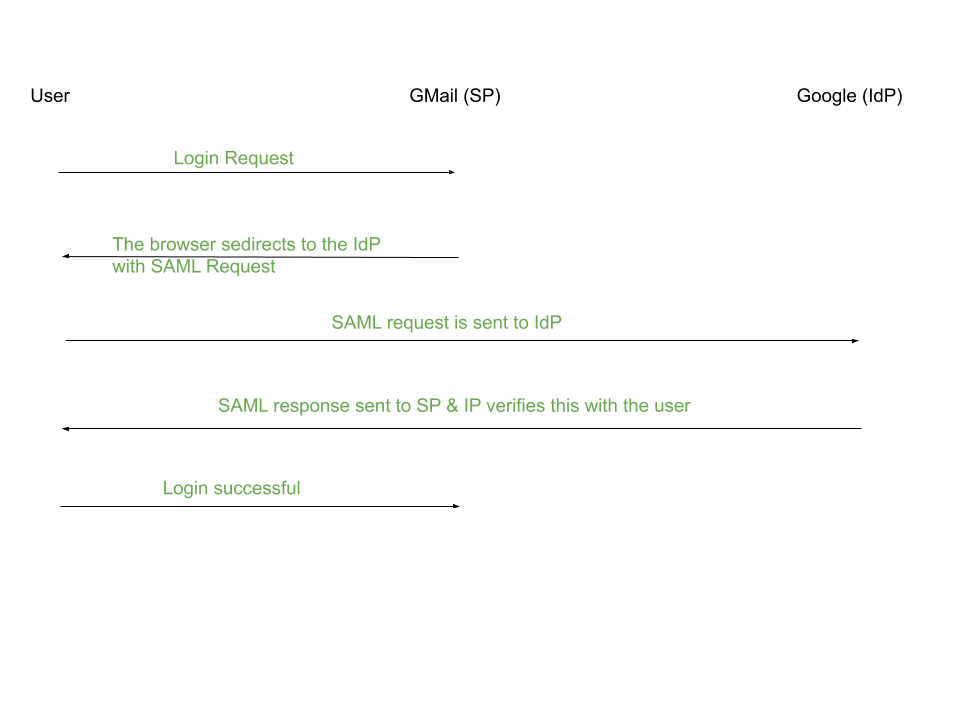
SAML is an XML based framework that stands for **Security Assertion Markup Language**. Let us see how SAML is used to enable SSO (Single-Sign-On). SSO is a term used for a type of login method where a company configures all of its web apps in such a way that the user can log in to all of these apps by just signing in once.

**Example –**When one logs in on gmail.com, they can visit YouTube, Google Drive, and other Google services without having to sign in to each service separately.

The SAML authentication flow is based on two entities –

1. **Service Providers (SP) –**The SP receives the authentication from the IdP and grants the authorisation to the user.
2. **Identity Providers (IdP) –**The IdP authenticates a user and sends their credentials along with their access rights for the service to the SP.

In the example given above, SP will be Gmail and IdP will be Google. SAML enables SSO, and as it is explained above, a user can log in once and the same credentials will be used to log into other SPs.  
  
**SAML Authentication Workflow –**

[](https://media.geeksforgeeks.org/wp-content/uploads/20200803122354/SAML.png)

1. A user tries to log in to Gmail.
2. Gmail generates a SAML request.
3. The SAML request is sent to Google by the browser, which parses this request, authenticates the user and creates a SAML response. This SAML response is encoded and sent back to the browser.
4. The browser sends this SAML response back to Gmail for verification.
5. If the user is successfully verified, they are logged in to Gmail.

**SAML Request:**

Some of the important terms in the SAML request are defined below –

1. **ID –**Identifier for a particular SAML request.
2. **Issuer –** The name of the service provider (SP).
3. **NameID –** The username/email address or phone number which is used to identify a user.
4. **AssertionConsumerServiceURL –**The SAML URL interface of the SP where the IP sends the auth token.

**SAML Response:**

A SAML response consists of two parts –

1. **Assertion –**  
   It is an XML document that has the details of the user. This contains the timestamp of the user login event and the method of authentication used (eg. 2 Factor Authentication, Kerberos, etc.)
2. **Signature –**  
   It is a Base64 encoded string which protects the integrity of the assertion. (If an attacker tries to change the username in the assertion to the victim’s username, the signature will prevent the hacker from logging in as the user).

**Key Generation:**

The Identity Provider (IdP) generates a private key and a public key. It signs the assertion with the private key. The public key is shared with the Service Provider (SP) which uses it to verify the SAML response and then log the user in.