**POC for Customer “Fictional” Inc for Multi-protocol SSO:**

This document briefly provides the Design and working of a Proof-of-concept which demonstrates how Auth0 can help centralize the user authentication for employees of Fictional Inc. to access multiple applications which currently use different Identity Providers.

This POC is to show how two applications which currently use different Identity Providers can be integrated with Auth0 and facilitate Single Sign-On across the two applications.

**Solution Overview:**

I have three applications setup in my Auth0 tenant.

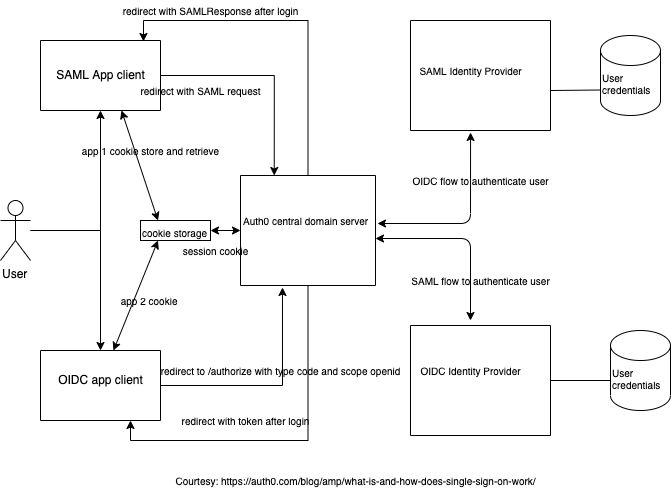
1. NodeExpress-App1: This is a regular server-side web application using the sample code provided by the quickstart in Auth0. This is a Node and Express web application which receives id\_token from Auth0
2. Python\_Emp\_Portal2: This is also another server-side web application using the sample code provided by the quickstart in Auth0. This is a Python web application which receives id\_token from Auth0
3. SAML NodeJS App: This is the same codebase as provided in Auth0 samples in github to have an application which is a SAML client (or receives SAML assertions from Auth0), It uses Node and Express and Passport JS

All applications are configured in Auth0 to use the default out-of-the-box configuration and to use the Classic Universal Login behavior provided by Auth0 and this is a key element of the solution for browser-based Single Sign-On functionality to work. Auth0 uses the OIDC protocol and OAuth framework to perform the authentication for the users accessing the portal applications. Auth0 acts as an Authorization server and sits between the applications and the existing Identity Providers used by the company.

Two Auth0 Connections are configured in the tenant and enabled for the two applications

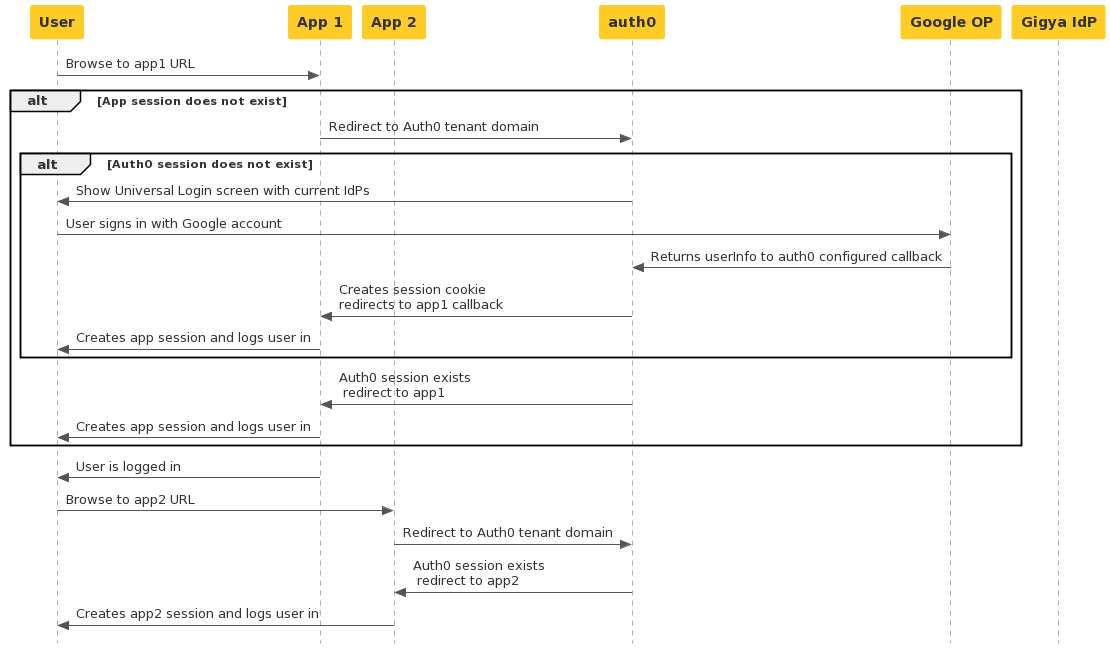
1. Google (google-oauth) as **OpenID connect provider** using my google cloud account (and not the dev keys)
2. Gigya or SAP Customer Data Cloud as **SAML IdP** using a test and training tenant I have access to

Single Sign On works with Auth0 where users are redirected to for login (Auth0 tenant acts as a central domain). Auth0 serves as a bridge to the Identity providers for the different applications (configured in connections) and users authenticate directly with the Identity Provider with their credentials and the configured protocols (SAML or OIDC). Users are redirected back to Auth0 central domain after successful authentication and Auth0 receives the user details. Users are then redirected back to the application with the id\_token (issued by Auth0) there are two sessions which are created; Application session and the Auth0 session cookie which allows for Single Sign-On when the user navigates to another app and tries to login, it redirects to the Auth0 central domain and it checks that a session cookie exists and it will redirect back to the application with the user details without the user having to login again.



The user can login once with one of the Identity Providers which they use and consent to sharing the requested scopes/claims for the application/portal they are using, and they can now navigate to any other application without having to enter credentials.

A sequence diagram with user choosing one of the Identity providers(google) is shown below



**Account Linking:**

With the use of and enablement of an Auth0 built-in extension “Auth0 Account Link”, users who are associated with multiple Identity Providers will be prompted to link their account when logging in with a different Identity Provider then they used previously, so there will be one user profile instead of multiple for the same user (identified with their email address). The users can choose not to link their accounts and have multiple user profiles in Auth0.

**Demo:**

**Scenario 1:**

1. A user with email [testfullreg@mailinator.com](mailto:testfullreg@mailinator.com) logs into App A (SAML node js app) which uses the SAML Idp provider Gigya
2. In the same browser session user navigates to another app App B (Python OIDC app) an is signed in without having to enter credentials

**Scenario 2:**

1. Another user with email [pratik.scm@gmail.com](mailto:pratik.scm@gmail.com) logs into App B (Python OIDC app) which uses OIDC IP provider Google OAuth
2. Same user goes to App A and is signed in without having to enter credentials

**Scenario 3:**

1. Delete user [pratik.scm@gmail.com](mailto:pratik.scm@gmail.com) as there is an issue with account linking with SAML
2. With App A (SAML node js app) signup new user [pratik.scm@gmail.com](mailto:pratik.scm@gmail.com) by choosing SAML IdP provider
3. Close browser session
4. In new session, User with same email [pratik.scm@gmail.com](mailto:pratik.scm@gmail.com) goes to App B and chooses the Google IdP to sign in
5. Account will be prompted to link with the existing account created with SAML connection and after authentication, the account will be linked.

Issues which could not be resolved:

1. Suggested Account Linking with Account Link Extension does not work with SAML Web app addon, code driven account linking may be needed after the user is logged in is shown the accounts which could be linked and then linked later
2. SAML app using the OIDC IdP fails