# Migrate An Existing User Database to Auth0

THE PROBLEM

Auth0 is commonly used to upgrade the existing Identity Provider (IdP) to one that provides modern features such as contextual multifactor authentication, federated logins for business partners, password-less authentication, biometrics, customer logins using their existing social accounts, Single Sign-on (SSO) and more with an enterprise-class, developer-friendly IdP. Auth0 can be deployed in such a way that it uses the existing user database, to minimize any disruptions or inconveniences for users, such as the requirement for them to reset their passwords. However, it is common for the existing user database to be in need of an upgrade as well to increase its scalability, performance, availability, security or scope. How can Auth0 be deployed to use the existing user database, ensure a smooth migration of all user data to its built-in, enterprise-class database, and avoid the hassle of password resets?

THE BIG PICTURE

Auth0 replaces the existing Identity Provider, and uses the existing external user database. The setting that determines which IdP existing applications should use is simply changed to be Auth0. Existing applications will then access Auth0 using the SAML, OpenID Connect, WS-FED or OAUth2 protocol. For new native, Web and mobile applications, the Auth0 Rest APIs are accessed through convenient, platform-specific SDKs to perform authentication using the code samples and customized step-by-step guidance that developers highly value from Auth0.
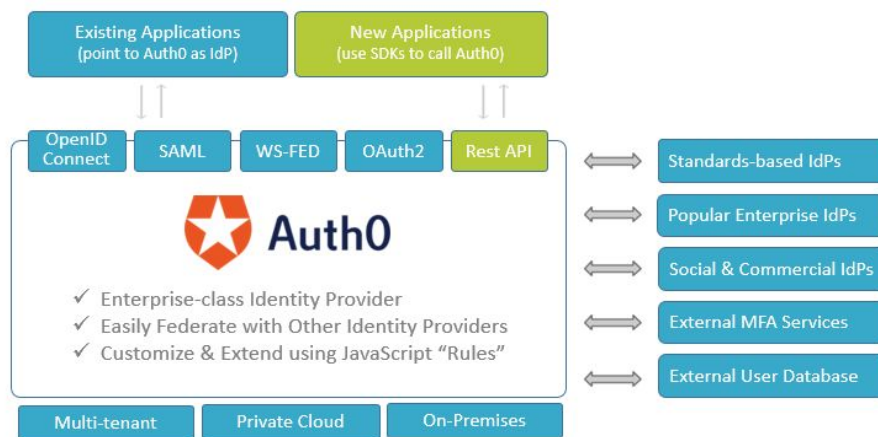

Auth0 is an Enterprise-class Identity Provider that federates and extends well

DATABASE MIGRATION

Auth0 utilizes a built-in, enterprise-class, highly scalable and available database that is ideal for keeping track of millions of users. The first time a user or device logs in to Auth0, they will not have a record in the built-in Auth0 database, so Auth0 will use its connection to the existing external user database to get the record, including any user data that is to be migrated into the Auth0 built-in database. As well as completing the authentication request, Auth0 adds the newly acquired user record to its built-in database. Over the course of a few weeks or months, a majority of the users will have been automatically migrated over without noticing anything has changed. The rest of the records can then be bulk imported into Auth0 at any time but they will

require password resets. Once the process is complete the existing external database can be retired. For more details see https://auth0.com/docs/connections/database/migrating.
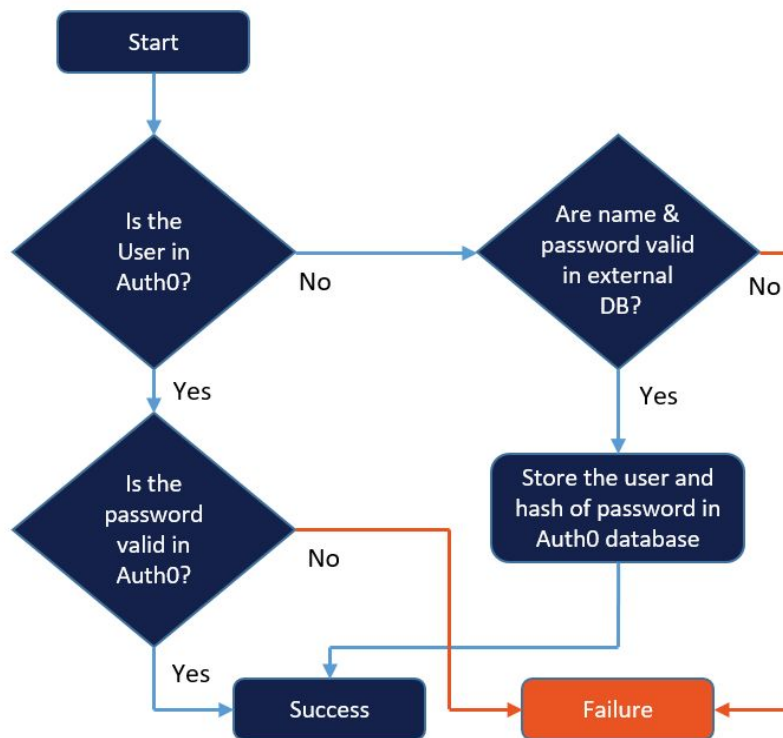

Logic used to migrate users to the Auth0 database

Using this algorithm means that the user will not have to reset their password. Once a user is migrated to the Auth0 user database, the legacy database records will no longer be used for them.

HOW IT WORKS

Applications are directed to use Auth0 as their Identity Provider (IDP). Auth0 either uses a direct connection to the original user database (https://auth0.com/docs/connections/database) or calls on a Web service that accesses the existing external database to authenticate the user and collect all of their user information, as depicted in the diagram below.
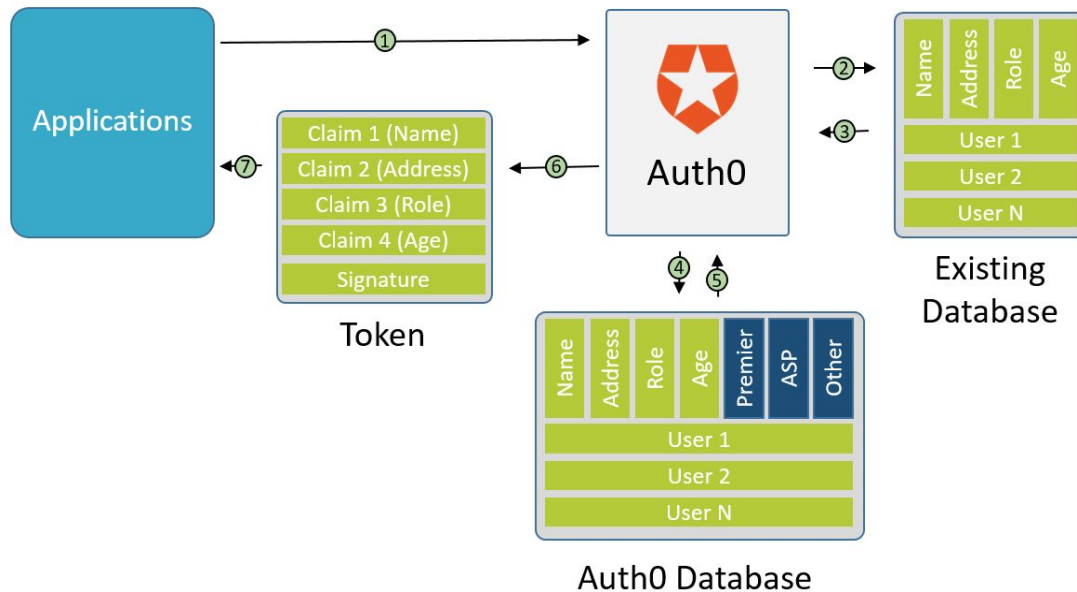
User identities in an existing external database are migrated on the fly to the built-in database

Auth0 provides the template Node.js code that it will run in a secure sandbox to perform the "Login" and "Get User Data" operations to the existing external user database or to a Web service so the data can be migrated to the Auth0 built-in database. This JavaScript template code is easily customized for the exact user data migration scenario.

Auth0 provides templates for most of the common databases such as: **ASP.NET Membership Provider**, **MongoDB**, **MySQL**, **PostgreSQL**, **SQLServer**, **Windows Azure SQL Database**, and for a Web service that will connect to any kind of database or Web service with a customized Node.js script and migrate its user data into the Auth0 user database easily and non-disruptively.

If desired, it's easy to gather user information from additional sources and add it to the record for each user as their user data is being migrated over to the Auth0 built-in user database.

ADVANTAGES

The database migration feature of Auth0 can accommodate any existing external database and can move the user data records to the Auth0 built-in user database, while enhancing the records with additional data if desired. The user records are migrated gradually over time to avoid adversely impacting operations, the last ones are bulk loaded into Auth0 so the old database can be retired. With Auth0, migrating user data to the Auth0 database to meets your scale, availability, performance or security goals is easier than you think.