Multiple Organization Architecture (Multitenancy): Overview

An Auth0 Integration Guide to Architectures that must Accommodate Application Instances for Multiple Organizations or Brands

Table of contents

Introduction

In the Business-to-Business (B2B) world, you are selling to businesses. Your users belong to different organizations that have signed up for your service. These users are often, and sometimes exclusively, employees of the different organizations that have signed up for your service. Whether you are architecting your integration with Auth0 or a developer looking for help designing an integration, this document should give you a high level overview of what common use cases we see with respect to multi-tenant applications.

Most B2B applications strive to create a pleasant user experience for the employees/users of the businesses they serve. To accomplish this most services in B2B add some branding to the service for each of the organizations that use the service. For example, let's say you work for AwesomeSaas (A SaaS software company) your company uses Human0 an HR application for managing benefits, etc. You would access your HR app at https://awesomesaas.human0.com when you log in you see the AwesomeSaas logo and it is customized to use AwesomeSaas colors.

The first thing you will need to consider is whether or not your customers (we will call them organizations) allow users from other organizations to log into their instance of the application. We need to know whether or not those users are shared between organizations or isolated to their organization.

Let's introduce a couple of examples of applications that will help highlight the differences.

First, let's introduce a fictional company: Travel0. When navigating this document, think of the company you work for as Travel0. If the application described has similar identity requirements to the product you produce, then you are in the right section.

Travel0 is a company that provides travel related services online. They have several applications, we will focus on the two applications that are marketed directly to organizations:

- **Travel0 Corporate Booking**: This application provides organizations with an online application where their employees can log into the application and book work related travel. Organizations that are customers of this application include:
    - **Hoekstra & Associates**: Small law office of just a couple of employees. They do not have an IT department and don't have the time or capacity to learn how to setup a corporate Identity Provider (IDP).
    - **Gupta & Smith Law**: Larger law office but they also do not have an IT department and don't have the time or capacity to learn how to setup a corporate IDP.
    - **MetaHexa Bank**: Large finance organization, they do banking and insurance. They have their own IDP.

    - **Many Student University (MSU)**: A large university with several campuses. Each campus has its own IDP.
- **Travel0 Adventure Management**: This application allows organizations to create and market adventures (white water rafting, horseback riding, zip line, etc). It allows guides (freelance or employees) to sign up for or be scheduled to lead adventures. Organizations that are customers of this application include:
    - **AdventureZ**: This is a large tour/event guide, they have their own IDP that they use for their employees. They rarely, if ever need freelancers because they just have enough guides on staff. Some of which only work during the busy times. They also do freelance work for other companies.
    - **Rocky Mountain High Adventures** => This is a new group, coming on the market for the first time. Just the co-founders run tours, and they mostly reach out to freelancers for help during busy times.
    - **Suzie's Rafting and Ziplines** => This company has been around for a long time. They have a staff of guides that handle most of their events, but will also reach out to freelancers when busy.

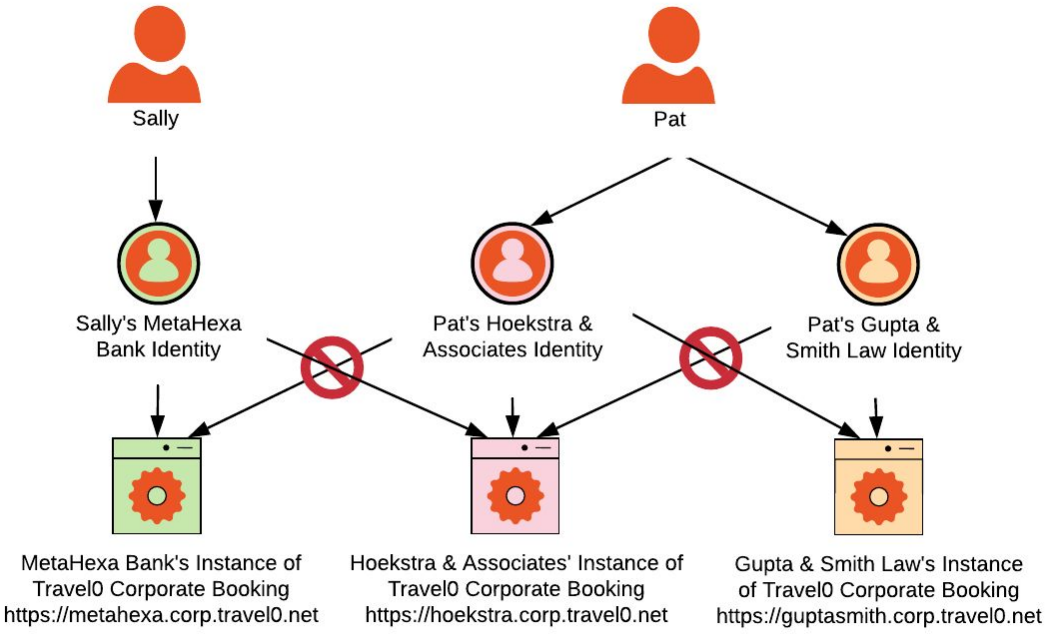
Terminology

Application Tenant => We will avoid using this term as much as possible to avoid confusion, but where necessary it will refer to a tenant in *your* application as opposed to the Auth0 Tenant. Instead of using this term, we will use "Organization".
Auth0 Tenant (Authorization Server) => This is the Auth0 tenant that you create in Auth0. It is your Authorization Server and represents a user domain.
Employee => A person who is part of your company. They likely have an account in your Identity Provider (IDP). They may need admin access to organization instances. NOTE: your customers may have users who are also employees, but we will refer to those as Organization Users as we don't know if they are employees or not. We will only refer to Employees of your company.
Identity Provider (IDP) => A service that manages authentication of users and optionally user profile information for an organization, company or group. Example Providers: Auth0, Azure AD, Google, Facebook, etc. NOTE: each of these providers will have an individual configuration instance per organization, company or group. Each individual instance would be considered a separate IDP, though many may be related or federate to each other.
Organization => This is a company that is a customer of yours. If you refer to organization instances of your applications as tenants, we will refer to them as organizations to avoid confusing the term with the Auth0 tenant. This is a replacement for the term Application Tenant to avoid confusion.
Organization User => The person who is logging into the application as a member of one of your organizations.

To Share or Not to Share

An organization should map directly to one of your business customers/partners. There are two different approaches to how to store your organization users. Pay close example to the users that need access to more than one organization to help you determine which approach more closely maps to your company's requirements.

- **Isolated to the organization** => Every user *belongs* to exactly one organization. It would not make sense for that user to be a part of more than one organization, and even if they were, it would make sense for them to have a separate "identity" for that other organization. See Users Isolated by Organization for more information. We will use Travel0 Corporate Booking for our example.

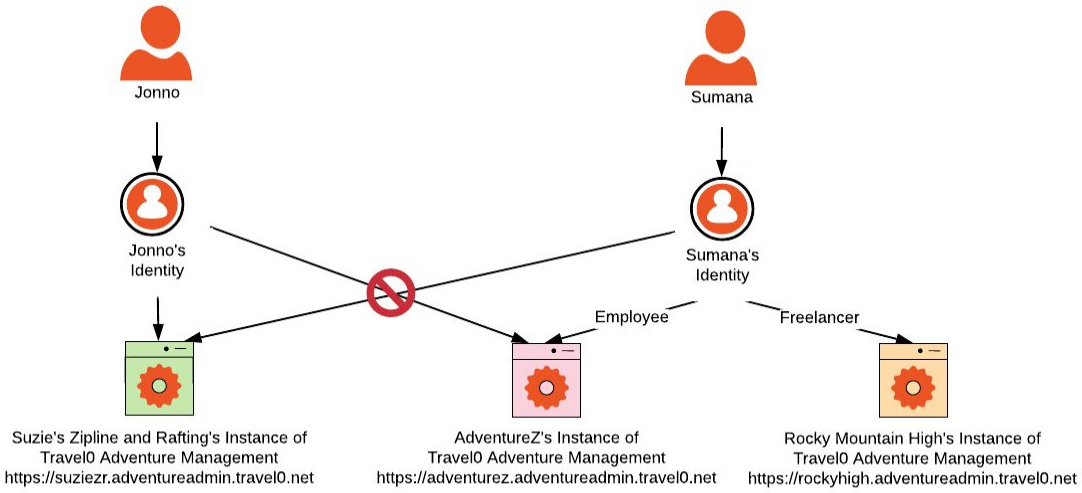


Sally is a typical user in this environment. Sally is an employee of MetaHexa Bank and she can only access the MetaHexa Bank's instance of Travel0 Corporate Booking.

Pat is atypical for this type of environment, either Pat doesn't exist for your company or is a rare user. We are including Pat as an example of the decision that is being made when isolating users to their organization. Pat is a freelance paralegal and does some work for both Heokstra & Associates and Gupta & Smith Law. Here is where your environment must make a decision. If you want users to be isolated to their organization, you are making the decision that Pat must create two separate users, one for accessing Heokstra & Associates's instance of Travel0 Corporate Booking and a separate user for accessing Gupta & Smith Law's instance of Travel0 Corporate

Booking. This makes sense in this scenario and probably reduces accidents to force Pat to create two separate personas, one for each law firm so that when Pat books travel it requires a login to the specific organization instance required to book.

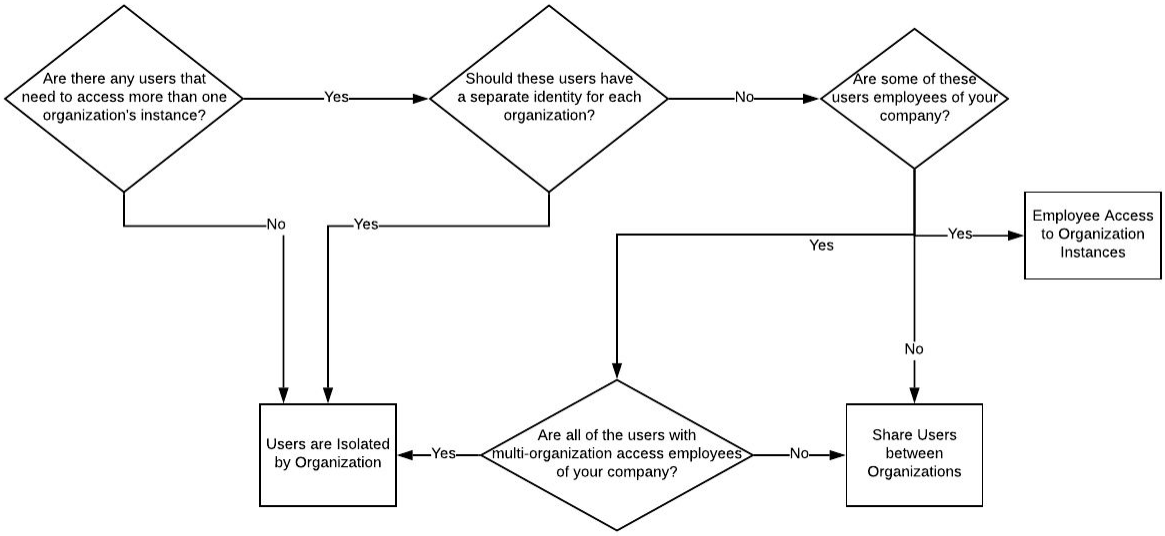
- **Shared between organizations** => In a case like this, users either create credentials in your company, or they can access other organizations' instances of your application using credentials from their own organization. A simple way to look at this is that one user may be authorized to access more than one organization's instance of the application. A user would understand that they can use the same credentials to access both instances of an application. See Some Users Shared between Organizations for more information. We will use Travel0 Adventure Management for our example.



Jonno is a typical user. Jonno is an employee of Suzie's Rafting and Ziplines. Jonno is only able log into Suzie's Instance of Travel0 Adventure Management to create and guide adventures.

Sumana is an employee of AdventureZ, but AdventureZ also coordinates freelance opportunities for the smaller guide companies during high peak times. Sumana has been invited by Rocky Mountain High Adventures to freelance. Sumana is authorized to log into both AdventureZ and Rocky Mountain's instances of Travel0 Adventure Management. However, since she has never been invited to guide for Suzie's Rafting and Ziplines, she is not authorized to access that instance. Sumana needs to have the same identity for both organizations because the guide system involves a rating system. Sumana's ratings need to carry over and be combined between organizations.

Now that we have defined two different approaches to user isolation, let's walk through how to make this decision. Here are the questions you need to ask yourself to determine which approach you will need to take:



*Even if only a small percentage of your users belong to more than one organization, you need to structure your system so users **can** belong to more than one if you want to support it for **any** of your users.*

Users are Isolated by Organization

Each organization has its own set of users and users can not and should not be able to access other organizations. If they attempt to, they should be rejected as unauthorized (see Users Isolated by Organization), keep in mind that you can choose to force your users create a separate account for each organization even if they belong to more than one as a person, they would be considered two different users.

Share Users between Organizations

A user may belong to more than one organization and it would be convenient if that user did not have to have a separate identity/account as they navigate from one organization to another (see Some Users Shared between Organizations). Organizations can still use their own IDP in shared user scenarios.

Employee Access to Organization Instances

Do your employees need to be able to log in to the organization's instances? If so, see Employees access to Organization Instance.