Resisting The Temptation: Is It Ever The Right Choice To Use APIs To Integrate Identity Providers With Your Business Processes?

BY AIDAN TWOMEY

Single Sign On (SSO) is an authentication process that allows a user to gain access to multiple applications, such as signing on to a Google account to access other websites. SSO has become an essential tenet of usability, giving employees, or members, the ability to log in once, with one set of credentials, to get access to all corporate or brand apps, websites, and data for which they have permission.

To maximize its effectiveness SSO requires a central directory, such as Okta's Universal Directory or Azure AD, to store and access the identities of those belonging to an organisation. This allows their authentication request to be validated, regardless of the individual access point.

The directory could also be used to automatically create the third-party accounts required by a valid user, and then to easily close those accounts when that user leaves the organisation. This automated provisioning can allow companies to efficiently allocate expensive licenses and optimise costs. So, a new starter in the marketing department could start using Salesforce on her first day, a software developer could sign on to the app they are developing, and the company would allocate the licences without any manual intervention.

But… life is sometimes messy, and it may be that a company has more than one list of users that it would like to authenticate.

For example, a business might use an on-premises Active Directory (AD) to authenticate workers in head office but rely on a cloud-based identity provider to authenticate remote workers.

Choosing either one of these as a single source of truth would have drawbacks:

- consolidating all users into the AD might lead to unnecessary licences costs for users who don’t require the benefits it provides, or
- moving everyone to the cloud-based provider may require disruptive account migration for little discernible benefit.

The ideal solution could be to integrate both types of user with a single Identity-as-a-service provider for a unified sign on experience that defers authentication to the AD only for those that need it.

Vendor products are so good at handling these standard scenarios that we can now consider ***Authentication and Access*** as a commodity. The model of out-of-the-box configuration, complemented by custom code that calls APIs for idiosyncratic cases, is very common. Presented with this scenario we might add business specific attributes on an account and then use the API to populate them. We could even use a web hook to call into some of our own code that resets a password or deactivates a user completely. However, we should take care not to try to lead the products where they don’t want to go. They may well defer to another authenticator but will likely block a mix-and-match approach that tries to invent a unique authentication process.

Let’s say we configure Okta to hand control of authentication over to Active Directory, allowing organisations to leave password-reset rules in the hands of AD. It is important to realise that once we have done this, Okta will limit the lifecycle operations it will allow from API calls, and any calls to activate or deactivate the user in Okta will be rejected. Okta will accept these status updates only from AD.

Similarly, the natural home for personal information is the organisation’s HR system. If the authentication system requires emails or phone numbers – in order, for example, to perform multi-factor authentication – we should source them from the appropriate system rather than duplicate the information. Access management platforms like Adaptive or Okta have standard connectors with common HR packages, making mapping between them easily configurable. Again, once they have handed control over to another system, they take care not to allow intermediate states.

Sample Case:
Okta can map values from the popular HR system Workday and allow it to be a profile master. It will also synchronise the user lifecycle with Workday, as it’s appropriate for the HR system to determine when people have joined and left the organisation.

However, following such a configuration, any customisation via the API will likely be difficult. SSO to applications after users have left the organization, for example, presents a problem if we try to do this through the authentication system rather than the HR system.

Once Workday has terminated a user, Okta will treat this as a message from God. Even if we have configured Okta to do nothing, attempts to re-establish the connection with Workday will be ignored.

The core purpose of automating access management is to optimise costs and simplify license provisioning.

It is always tempting to regard an individual organisation as a special case and believe that the ability to tweak processes using APIs permits bending access management systems to existing processes. Sometimes there are unique requirements intrinsic to a business but responding to these always raises the risk of butting up against the reasonable restrictions the vendor systems have imposed.

It is better to remember that identity and access management has become a commodity, that standardisation brings with its efficiency and compliance, and that it is ultimately cheaper and more effective to transform business processes than to rely on APIs to transform systems.

Aiden Twomey is a Consultant at Amido