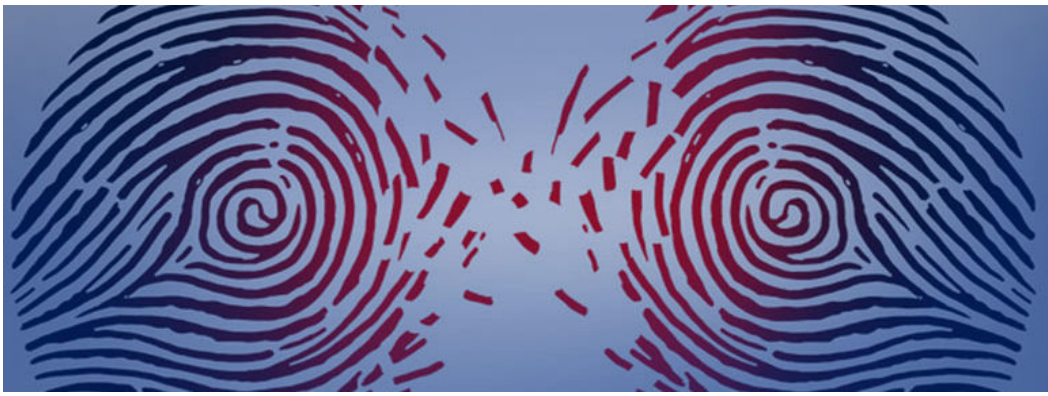
Security Boulevard

July 29, 2021

Pushing the Limits of IDaaS with AMaaS



by Canming Jiang, CEO Datawiza

Concern over secure access to data has led to significant adoption of cloud identity management solutions, specifically identity-as-a-service (IDaaS), to ensure that people accessing applications are who they say they are—that is, to authenticate their identity.

However, IDaaS solves only half the problem. Privacy regulations require that we ensure only the right people have access to sensitive data, such as personally identifiable information (PII), at the right time. This requires comprehensive, up-to-the-minute access control of each authenticated user's right to access information down to the cell level (e.g., access to the last four digits of SSNs versus entire SSNs).

Current IDaaS solutions also place significant demands on development teams to integrate applications with them, requiring significant engineering resources and slowing time-to-value. Furthermore, today's hybrid multi-cloud environments may rely on multiple authentication scenarios, causing confusion for IT and governance teams and a poor end-user experience.

Access management-as-a-service (AMaaS) is an emerging strategy for adding distributed authentication and authorization for assets (e.g., applications and APIs) across the hybrid multi-cloud infrastructure, but with centralized management of those access policies.

The Limitations of IDaaS

Authorization: Some IDaaS solutions provide limited authorization capabilities but are not fine-grained enough (e.g., URL level) to ensure compliance with emerging data privacy regulations, including the EU General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA).

Integration effort: Integrating applications and APIs with IDaaS solutions requires development teams to learn modern authentication and authorization protocols, such as OAuth, OIDC and SAML, as well as different platforms' SDKs and APIs. Developers also need to write integration code for each app they want to integrate with the IDaaS solution. For companies with hundreds or thousands of applications, this is time-consuming and expensive even if the development teams are fully trained.

Migrating legacy apps to modern IDaaS requires significant application rewrites and multiple error-prone manual configuration steps. If all this work is not guided by a security expert, security vulnerabilities could be introduced that are as bad as what the company is trying to eliminate by moving to IDaaS.

Finally, once the IDaaS solution is in place, bolting on an access management solution requires even more work. Legacy on-premises gateway solutions (like web access managers) are expensive and difficult to install and make it hard to onboard apps. The design of the local control planes of these legacy gateway solutions also makes them painful to manage. Rewriting a custom authorization system with modern protocols is also time-consuming and expensive.

Hybrid multi-cloud: Applications running in different on-premises locations and public clouds often employ different identity access management (IAM) solutions. This makes consistent enforcement of IAM policies across the distributed architecture extremely difficult, which can easily lead to security gaps and violations of privacy regulations.

AMaaS: A Centralized Authorization Strategy

The goal of AMaaS is to provide a

Continued >

cloud-delivered and cloud-managed access management solution that seamlessly integrates with the IDasS solution to enable single sign-on (SSO) and multi-factor authentication (MFA) for applications and APIs and enforce fine-grained conditional access across the entire hybrid multi-cloud infrastructure.

AMaaS solutions usually embrace the Gartner cybersecurity mesh architecture, including two major components:

- Distributed policy enforcement points (PEPs) in the form of gateways or agents deployed close to the assets (e.g., applications or APIs) across the entire infrastructure, which are used to enforce authentication and authorization for a particular set of assets.
- A centralized management console, usually in the cloud, to manage the PEPs distributed with the assets. This console can manage configurations and access policies, collect access logs and generate reports and dashboards. Some advanced platforms can provide advanced AI-based threat detection and prevention.

The principles underlying any AMaaS solution should include:

- **Support for modern security protocols, such as OAuth/OIDC and SAML.** These protocols are today's standards for SSO and provide a simpler and more convenient experience for users logging in to many different applications. They also increase security.
- **The ability to define access policies based on detailed user and device attributes, including group, role, IP, or browser.** As companies centralize authorization management, they must manage policies for an ever-growing number of complex scenarios. For example, access to a customer database that includes PII may be limited by an employee's role or group in the company, as well as the current location of the person (i.e., they are allowed to access some PII while working in a particular country but not while traveling outside that country).
- **Support for hybrid multi-cloud environments.** It should not matter where an application or dataset resides; on-premises, in a public or private cloud or in multiple public clouds. The distributed agents or gateways should be deployed with those assets no matter where the assets reside. And the agents or gateways should be managed by a centralized console, usually in the cloud.
- **Minimal coding.** To the extent possible, a solution should minimize the need for significant coding and security expertise. This is necessary to accelerate migrations, manage costs and ensure security.

Operationalizing AMaaS

AMaaS solutions can help overcome two common challenges. First, consider a mid-sized company with limited IT resources that relies on a mix of on-premises legacy applications and cloud-based SaaS applications. Since the SaaS apps (e.g., Workday, ServiceNow) support modern protocols (e.g., OIDC), the company can easily integrate an IDaaS solution (e.g., Azure AD) with the SaaS apps. But employees cannot leverage the simple, easy-to-use SSO and MFA of the IDaaS for their legacy applications and must log in via a legacy identity system. An AMaaS solution would enable this company to easily migrate legacy applications to modern IDaaS platforms, enabling a true SSO scenario that lets employees work more efficiently.

In another example, a Fortune 500 financial institution still has most of its applications on-premises. It is starting to adopt modern IDaaS platforms but is still using several legacy systems (e.g., on-premises AD and LDAP stores) for storing user identities and attributes. AMaaS would enable the company to connect those different identity systems (modern IDaaS or legacy) to enable unified access control by combining different user attributes scattered in different systems and enforcing centralized managed access policies for applications in a complex distributed hybrid environment. This would require only minimal code tweaks, saving months of development time and enabling IT and security teams to focus on other challenges.

To summarize, a hallmark of today's enterprise infrastructure is that it is filled with a variety of ever-changing solutions. A company may use a mix of IDaaS solutions (e.g., Microsoft Azure AD, Okta and Auth0) across its hybrid multi-cloud environment. As such, it is essential for an AMaaS to support the widest range of complementary solutions to enable each organization to reach its security and user experience goals.