STRATA

What is Identity Orchestration?

Why You Need Identity Orchestration to Succeed with Multi-Cloud

strata.io

What is Identity Orchestration

Managing identity in the multi-cloud and hybrid cloud world requires a distributed identity model. Users must have consistent access to apps running on-premises or on multiple cloud platforms — whether from the cloud or within the enterprise network. Identity orchestration is the next generation of identity management software that makes this possible.

Identity Orchestration enables consistent identity and access to your apps regardless of where they run and which identity system you use.

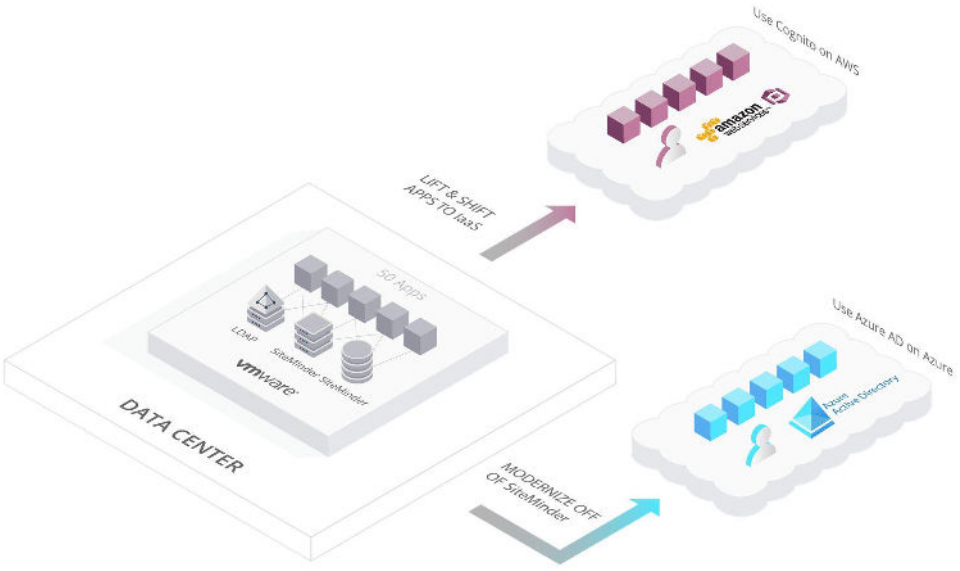
Identity Orchestration is a new way to do identity

Multi-cloud and hybrid-cloud adoption create new challenges for identity. Each cloud platform has its own identity system creating new identity silos during a cloud migration. Centralizing identities is not the answer, because companies need to leverage the unique capabilities of individual cloud platforms. This means multi-cloud now also means multi-identity.

Identity orchestration software creates a logical identity fabric that ensures identities and user access policies are consistent across disparate identity systems and multiple locations, both in the cloud and on-premises.

Imagine you need to transition 50 apps from on-premises to the cloud. Legacy, on-premises apps probably run on VMware (or similar technology) and are protected by SiteMinder as an identity infrastructure. The objective is to lift-and-shift legacy apps to run on Azure and AWS and use modern identity capabilities from Azure AD or AWS Cognito.

The location of the apps will change, but you don't want to change the user experience or recreate access policies in multiple identity systems. In other words, move the apps to multiple cloud platforms with the least amount of disruption, in the fastest, most cost-effective way, while never compromising on security.



Managing consistent identity and policies across multiple identity systems

Modern technical identity challenges to address

Multiple identity systems to manage

You use SiteMinder to protect your on-premises apps. You plan to use AWS Cognito to provide identity for AWS-hosted apps and Azure AD to provide identity for Azure-hosted apps. This means that in addition to managing the various systems that provide user attributes and MFA, you now also have three different identity systems to manage.

Apps require rewrites when modernizing identity systems

Your apps are hard-wired into SiteMinder using different types of HTTP headers and authentication mechanisms. Historically, moving these apps has meant rewiring from one identity system to another manually and at considerable cost and time.

It's essential to use standards like SAML and OIDC, but none of the apps support these protocols and, therefore, they require refactoring to accept SAML or OIDC for SSO. Furthermore, to use SAML and OIDC, each identity must have a local account to match up the SSO session with the identity profile. These local accounts don't exist in Azure or AWS. In the past, this meant that custom synchronization code must be developed (and maintained).

All users' passwords need resets during a big bang migration of identities

The identities on-premises are stored in LDAP, and they must be replicated from LDAP to Azure AD and AWS. Because of password hashing, it's impossible to copy and paste users from one directory to another because authentications will fail when compared to a hash. Historically, this has meant forcing users to do a password reset workflow as part of a "big bang" migration.

Legacy and modern identity systems need consistent policy enforcement

SiteMinder's sophisticated policy capabilities, including contextual access, active responses, and even the retrieval of attributes at runtime to personalize user experience at sign-on have been used for a long time. Azure AD does not support the same functional capabilities as SiteMinder. Historically, this has meant compromising and downgrading access policies to fit the lowest common denominator policy, and consequently weakening security protections.

Working within the company's complex environment

You have a robust, enterprise-grade networking infrastructure developed over many years that provides layers of defenses. It keeps bad actors out and selectively allows approved users in through VPNs. But, the VPN approach has collapsed under the rapid increase in the number of employees working remotely. You need to extend access to your on-premises apps securely at the edge of the data center. Historically, this has meant radical reconfigurations of enterprise networks at great expense and time delays.

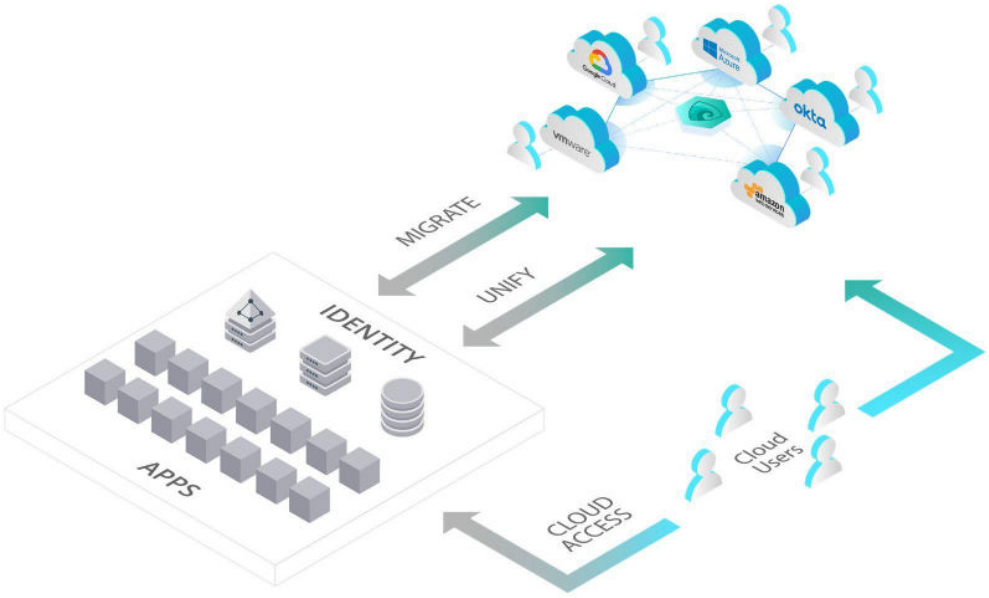
In addition to the technical challenges described, you are under pressure to cut costs in the near term to pay for the increase in spending to support remote workers. Timelines are severely compressed. What would have in the past taken several years to accomplish with custom code and complex migration projects, must now be done in months (or less) to keep pace with market changes.

Vendor lock-in

The rise of dominant cloud platforms from Amazon, Microsoft, and Google, brings a greater risk of getting locked in with these vendors. It won't come as a surprise that identity is one of the biggest sources of lock-in. You don't want to get locked into any platform — moving to multiple clouds is case-in-point. However, if not done correctly, you could find yourself moving from your legacy on-premises lock-in to cloud lock-in.

Identity Orchestration solves today's identity challenges with an Identity Fabric

An Identity Orchestration platform is a lightweight software that deploys in the cloud or on-premises. It runs as a service on a Linux server and reads its configuration from either a local or centralized YAML configuration file.



Modernizing apps without rewiring identity

Identity Orchestration uses connectors, workflows, and app gateways to orchestrate behavior across identity systems. It also creates an abstraction layer that applications use to integrate with any identity system without changing application code or modifying configurations. Identity Orchestration can move policies, configurations, and identities across any identity system. It also routes login requests to different identity providers or lookup-and-retrieve user attributes, groups, and other identity data from various identity stores.

The following foundational requirements define Identity Orchestration:

1. Natively distributed. Identity Orchestration must be engineered from the start to solve the distributed problems of multi-cloud and multi-identity use cases. Identity Orchestration will not succeed by retrofitting a centralized identity system to perform in a distributed way. This has been proven in compute where we have seen the rapid rise of Kubernetes due to its ability to natively manage distributed workloads using its distributed architecture.

2. Consistent identities. Identity Orchestration must enable consistent identities across multiple clouds and identity systems by programmatically automating identities into the various identity providers and creating a composite identity profile by building attributes from several identity providers in real-time.

3. Consistent policies. Identity Orchestration enables consistent user access policies by using an identity abstraction layer to normalize the definition of user access policies in a common syntax. It also must normalize enforcement of user access with meta-policies to fill functional gaps in policy that commonly occurs between different identity systems.

4. Identity abstraction layer. Identity Orchestration relies on an Identity Fabric that abstracts the various underlying identity infrastructures that an organization uses. This abstraction layer unifies different identity systems' APIs, data models, user access policies, and feature sets into a consistent Identity Fabric, saving the effort of learning multiple APIs and identity systems.

5. Distributed deployments. Identity Orchestration must be fit to deploy in many different configurations that reflect today's often complex environments. This means coexistence of on-premises and cloud. Identity Orchestration is also highly performant — it must handle all kinds of traffic and run as either a proxy or adjacent to apps through a sidecar model. All of which allow it to deliver elastic scale and rock-solid resilience.

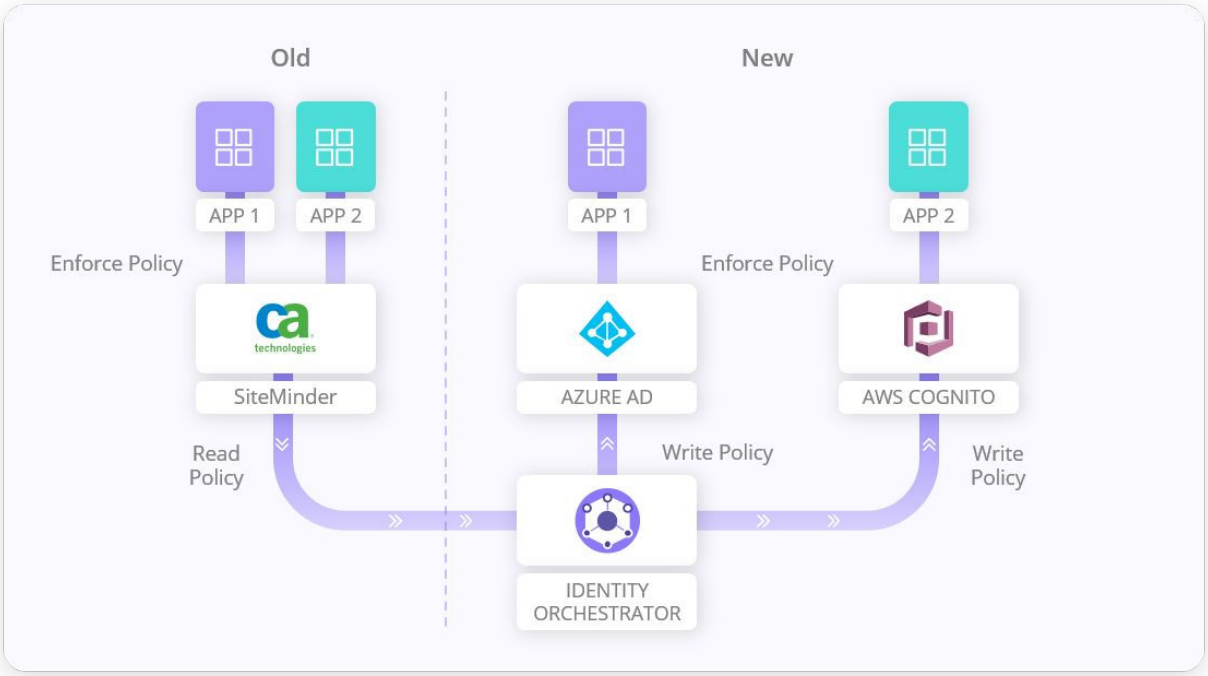
These foundational requirements are critical and define what Identity Orchestration must be capable of to support today's multi-cloud and multi-identity use cases.

Let's apply Identity Orchestration to the scenario above to understand how organizations can move to a cloud, multi-cloud, or hybrid-cloud while managing identities and access policies consistently.

Use Cases

USE CASE #1: Managing consistent identity and policies across multiple identity systems

To extend the policy defined originally in SiteMinder, Strata's Maverics Identity Discovery tool extracts policies from SiteMinder. Next, the Maverics Identity Orchestration software creates the system-specific access policy in Azure AD and AWS using their respective APIs. Next, Identity Orchestration is configured to present the identity to Azure AD and AWS Cognito with appropriate attributes so that Azure AD and AWS Cognito can enforce consistent access.

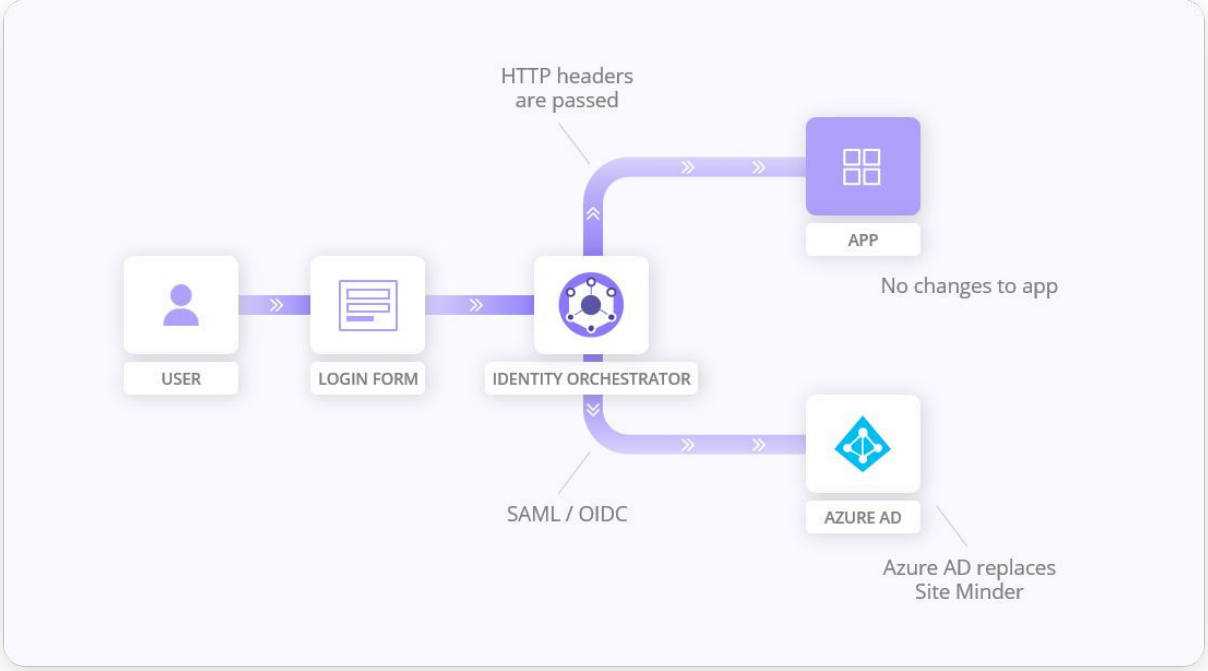


Managing consistent identity and policies across multiple identity systems

USE CASE #2:
Modernizing apps without rewriting identity

For incremental user migrations, Maverics proxies the login page and requests authentication from the legacy identity system. Then, it fetches additional attributes and creates a user account in Azure AD, assigning groups or roles and determining how to apply app access policies. This process takes place over several months, during which, the majority of active users are migrated.

The accounts that have not logged in during the period are considered dormant accounts and are bulk-migrated and put into a password reset process to re-validate the user identity. This eliminates the risk of dormant accounts being used by hackers to gain illicit access.

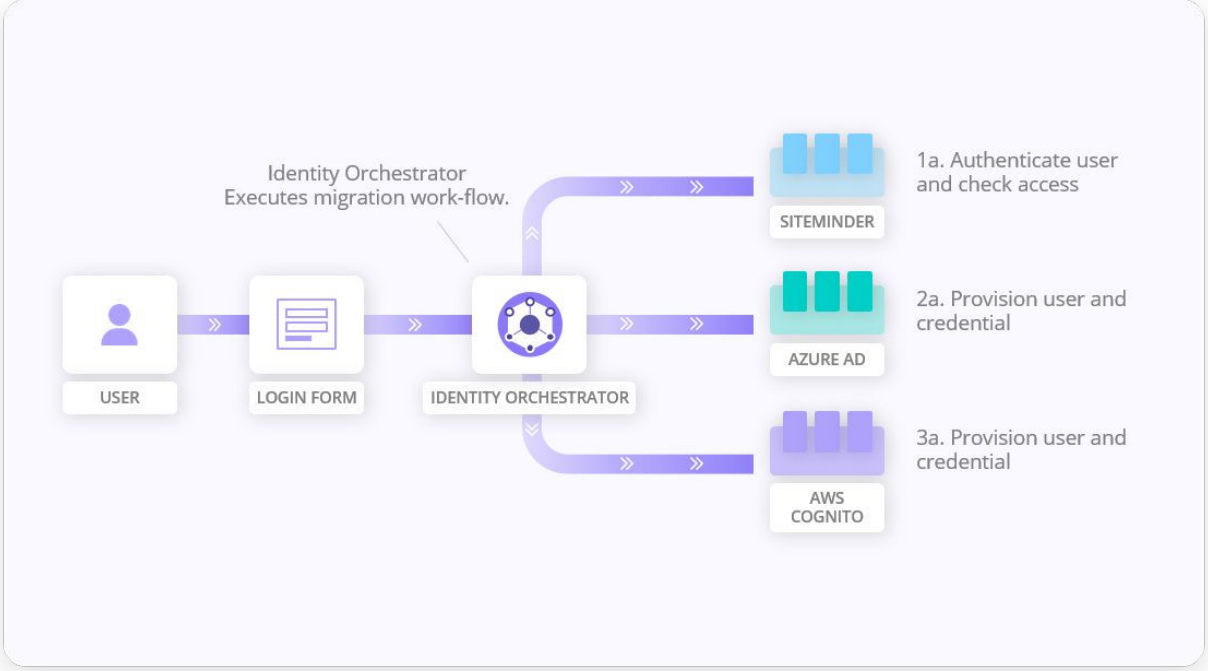


Modernizing apps without rewiring identity

USE CASE #3:
Avoiding big bang migration of identities

For incremental user migrations, the Identity Orchestrator proxies the login page and requests authentication from the legacy identity system. Then, the Identity Orchestrator fetches additional attributes and creates a user accounts Azure AD, assigning groups or roles and determining how to apply app access policies. This process takes place over several months where the majority of active users are migrated.

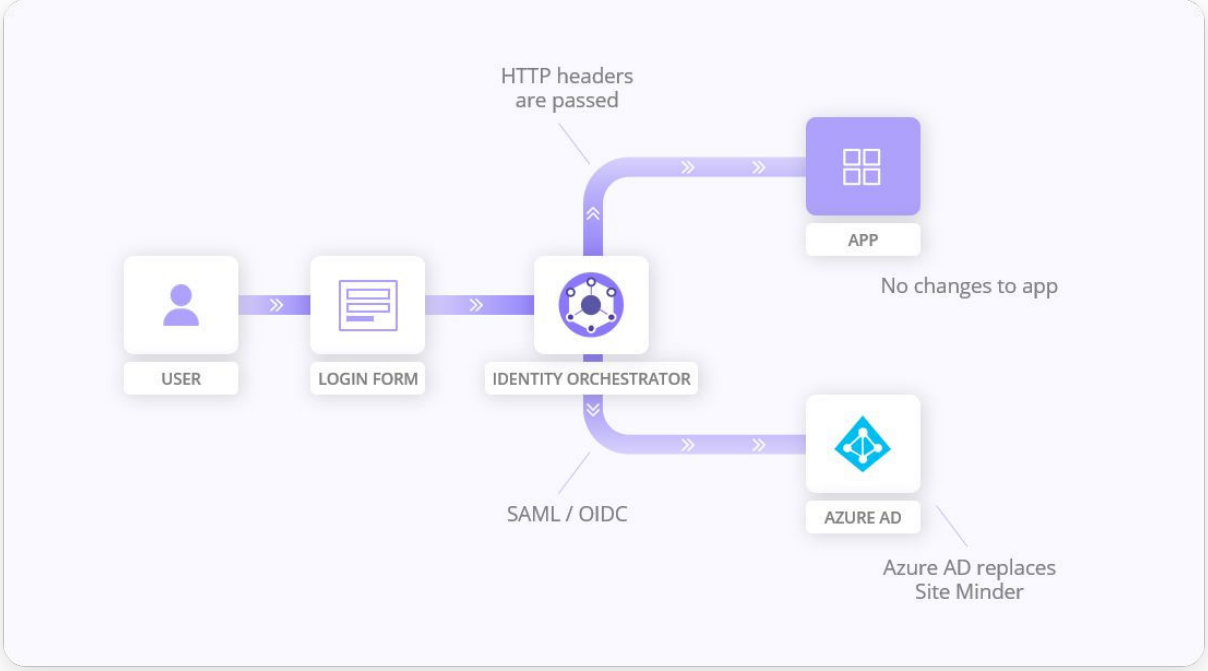
The accounts that have not logged in during the period are considered dormant accounts and are bulk-migrated and put into a password reset process to re-validate the user identity. This eliminates the risk of dormant accounts being used by hackers to gain illicit access.



Avoiding big bang migration of identities
with incremental migrations

USE CASE #4:
Extending SAML to apps that don't support SAML

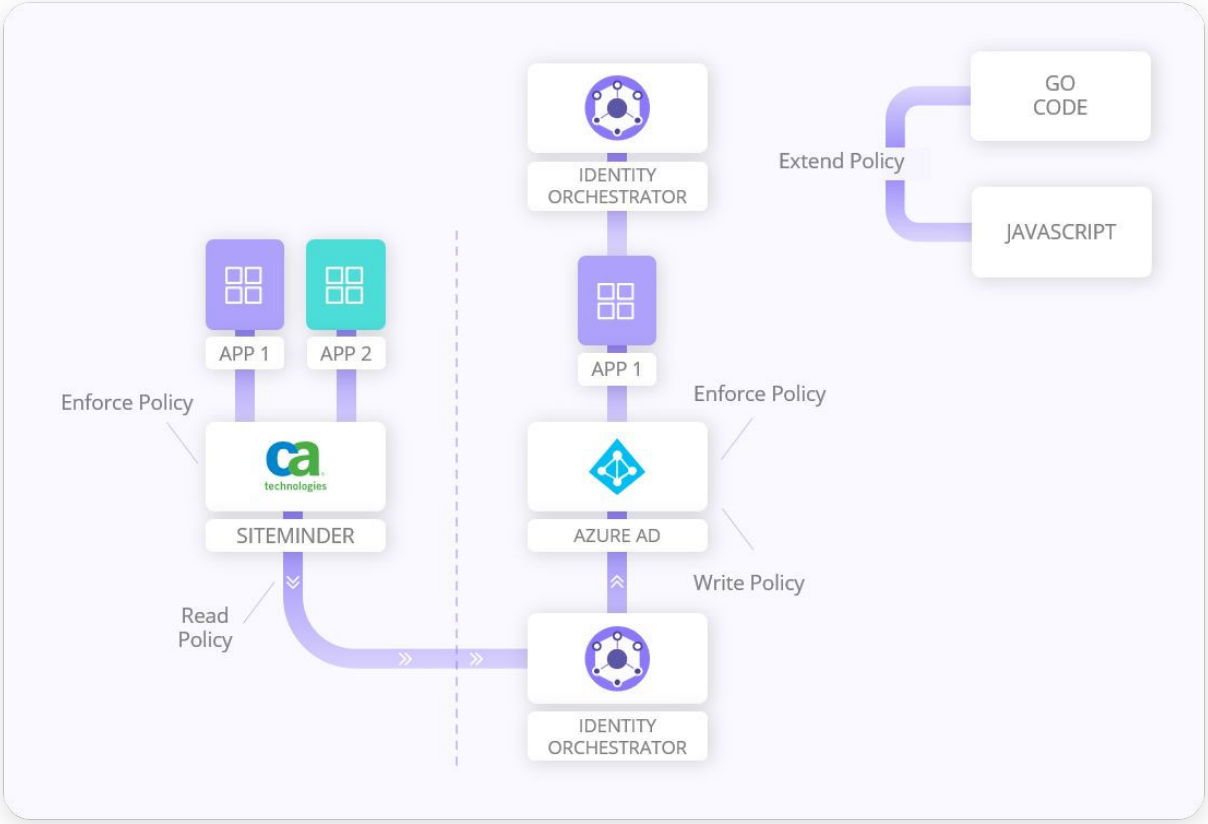
For legacy apps that don't support SAML or OIDC, Identity Orchestration handles the processing of federated identity. Upon success, it passes the user principal (user ID) to the web app the way it already consumes identity, through HTTP headers.



Extending SAML to apps that don't support SAML

USE CASE #5: Enabling consistent policy enforcement between legacy and modern identity systems

The Identity Orchestration platform bridges functional gaps in user access policies between complex policies enforced by legacy identity systems (such as contextual access and custom calculated attributes) and the simpler policies enforced by the cloud identity provider(s) — such as app-based MFA. To do this, Maverics uses JavaScript and Go service extensions that extend the built-in policies of Azure AD to provide near-infinite flexibility to enforce consistent user access policies for apps.

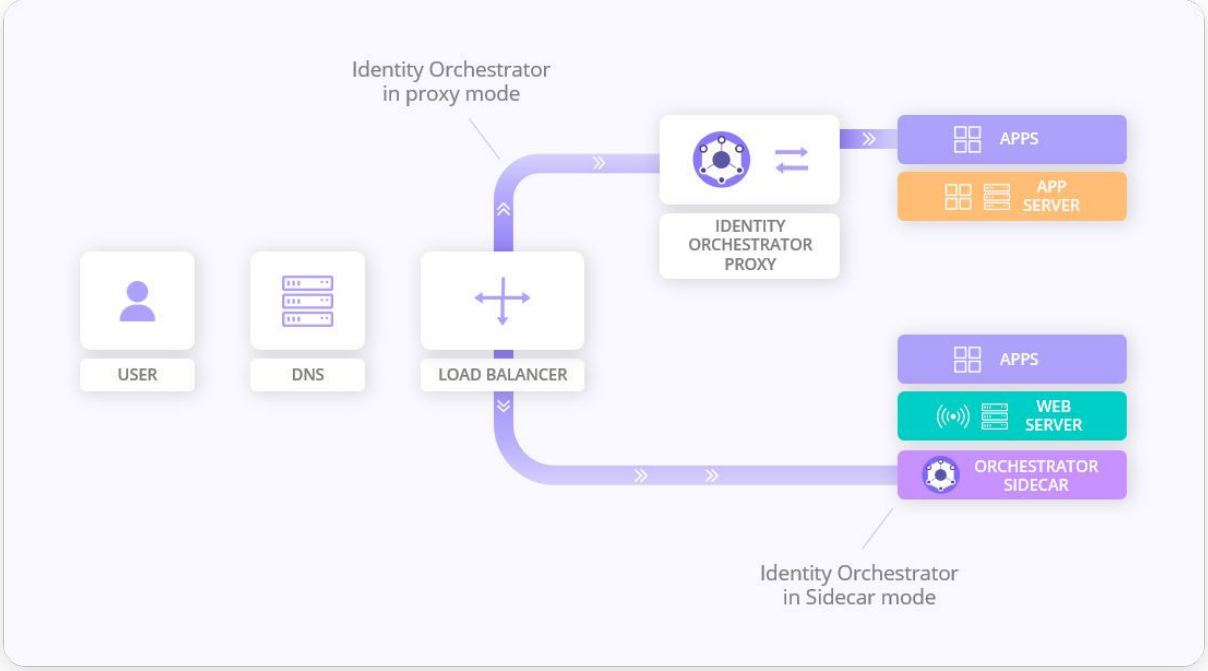


Consistent policy enforcement between legacy and modern identity systems

USE CASE #6:
Flexibility for complex networking environments

Maverics Identity Orchestration software deploys seamlessly into any networking topology, proxying the upstream apps and working without changes to load balancers, firewalls, and the web-tier hosting the apps.

To protect against 'side door' access, the Identity Orchestrator deploys as a 'sidecar' on web servers (such as IIS or Apache) and app servers. The Identity Orchestrator acts as a Zero Trust app gateway, securely extending access to apps whether those apps are on-premises or run in the cloud.



Flexibility for complex networking environments

Benefits of Strata's Identity Orchestration Platform

Strata's Maverics Identity Orchestration Platform enables organizations to confidently succeed in hybrid and multi-cloud environments using multiple identity systems, knowing there is a way to consistently manage access to apps running on multiple cloud platforms. Maverics saves you money and time by de-provisioning legacy infrastructures, avoiding the costs to refactor or rewire apps to work with modern identity, and turning complex migration projects into rapid modernization projects that happen in weeks instead of years. Finally, with Identity Orchestration, you eliminate the threat of vendor lock-in.

With a simple change in configuration, Strata improves agility by decoupling apps and identity to work with any identity system. Security is increased by moving your applications off end-of-service (EOS), outdated legacy identity systems, deploying modern security tools like MFA, identifying dormant accounts, and helping to enable Zero Trust for user access to apps.

Conclusion

The bottom line is that multi-cloud means multi-identity. This creates complex challenges to migrate and maintain identity and policies consistently across multiple clouds, unify on-premises identities, and modernize legacy identity. Deploying Strata Identity Orchestrator saves time and money while improving agility and security in these most crucial scenarios.

Take the next step by seeing a demo of Strata's Maverics software and requesting an Express Proof of Concept (POC). Contact sales@strata.io or visit us here: strata.io/demo

Connect with an Identity Orchestration expert SEE A DEMO

About Strata

Strata is pioneering the concept of identity orchestration for distributed, multi-cloud identity. The Maverics Platform enables enterprises to seamlessly unify on-premises and cloud-based authentication and access systems for consistent identity management in multi-cloud environments. Strata's distributed approach to identity enables organizations to break decades-old vendor lock-in that has prevented a broader transition of enterprise workloads to public cloud infrastructures. The company's founders co-authored the SAML open standard for identity interoperability, created the first cloud identity services, delivered the first open-source identity products, and are now building the first distributed identity platform.

Visit us at **Strata.io**

STRATA