STRATA

Identity Orchestration Buyer's Guide

Evaluating distributed identity solutions for multi-cloud and multi-identity systems

Overview

The move to multi-cloud has forced a fundamental change in how identity is managed across multiple, distributed platforms and identity systems. Identity orchestration is gaining momentum by building a distributed identity fabric that enables consistent access and identities across multiple platforms.

Multi-Cloud Identity Use Cases

There are three common use cases that illustrate the need to consider a distributed identity fabric:

1. Multi-cloud environments needing consistent identities and policies
2. Extending on-prem apps and identities to the cloud
3. Modernization - migration from legacy identity to modern identity systems

These uses will be covered in more detail in this guide.

Challenges

There are a number of challenges that must be overcome to achieve successful identity management that are unique to multi-cloud environments:

Multi-Cloud Introduces Identity Fragmentation and Silos. Organizations using more than one cloud must contend with more fragmentation from multiple silos, which makes managing secure access to apps difficult. Organizations need to manage identities and access consistently across platforms, regardless of vendor.

Multi-Cloud Needs Consistent Identities and Policies. Many organizations run apps on multiple cloud platforms and need to provide consistent access to users regardless of what cloud platform the app runs on. Further, each cloud has a built-in identity system that manages ***'local' user accounts.*** These local accounts must be consistent across platforms to support SAML and OIDC-based SSO. Further, **access policies** on these cloud platforms need to be consistent as well. Without consistent policies, it's difficult to manage access or demonstrate compliance.

Remote workers require secure access to on-premises apps and data. Accelerated by mandates to support remote workers, enterprises must quickly adapt to a massive rise in the number of employees working remotely. VPNs can only scale so much and what's needed is a way to extend access to on-premises apps to cloud-based users. These cloud-based users need secure and seamless access to apps and data that reside behind the firewall.

App-to-Identity Integration Results in Lock-In. For the past 10 years, enterprises have deployed identity infrastructure to support on-premises custom web apps. These apps were integrated directly with legacy identity infrastructure. Moving these apps to the cloud means rewriting apps to work with a new identity system, taking months of developer time. Apps that are hard-wired to a single identity vendor leads to more lock-in and limited choices. As a result, organizations are locked into legacy identity platforms, preventing apps from moving to the cloud.

Multi-Cloud Requires Higher Level of Scale. Consider managing identity and access for dozens of apps and thousands of identities, now multiplied by the number of cloud platforms in use. Factor in the rapid increase in the number of remote workers that need to access cloud-based apps and it's simply not possible to keep up with changes using manual effort.

Need Gradual, Agile Migration Capabilities. Enterprises have significant existing on-premises IT investments used to run the business. Moving these business-critical apps requires working with the dependencies between apps and infrastructure. It's not possible (or a good idea either) to do 'big bang' migrations or modernize everything all at once. What's needed is an agile approach that supports an app-by-app and user-by-user, incremental model that decreases risk and accelerates execution.

Need Zero-Trust Secure Architecture. With cloud deployments, networking plays a different role than on-premises as identity has become the new perimeter. As apps span multiple cloud platforms and on-premises platforms users must cross the open Internet to access them. What's needed in this case is to assume that all networks are hostile and implement mutual authentication and encryption for all communications. Further, managing access to system-level credentials is critical.

Core Multi-Cloud Identity Use Cases

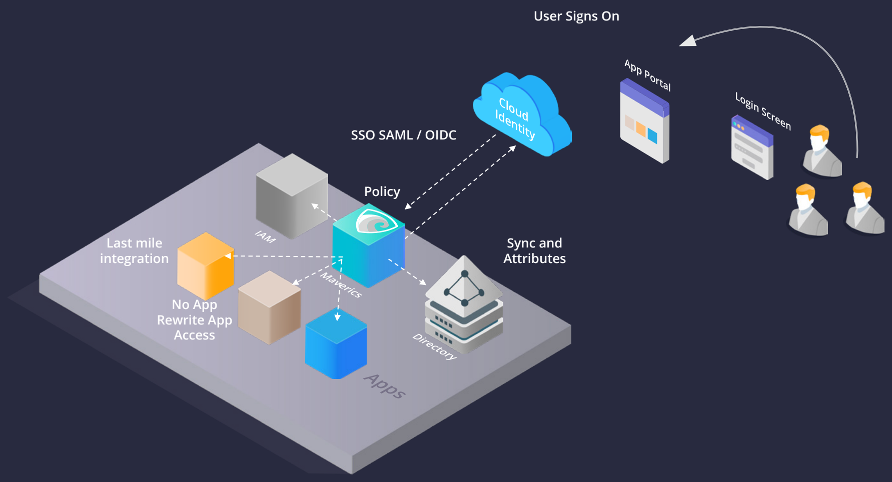
Multi-cloud platforms are the most common scenario involving two or more cloud platforms. Additionally, the mix of on-prem and public clouds adds new cloud and identity platforms. Some organizations use multiple cloud platforms to avoid vendor lock-in and spread their risk across several cloud vendors to keep their options open. They can also optimize their costs by moving workloads to the best-priced platform at any given time. Other organizations end up with multiple clouds through a merger or acquisition that historically used different technology stacks. And finally, some companies use multi-cloud for high availability and redundancy to dynamically failover to a redundant cloud provider, most likely on a different cloud vendor.

Core Multi-Cloud Identity Use Cases

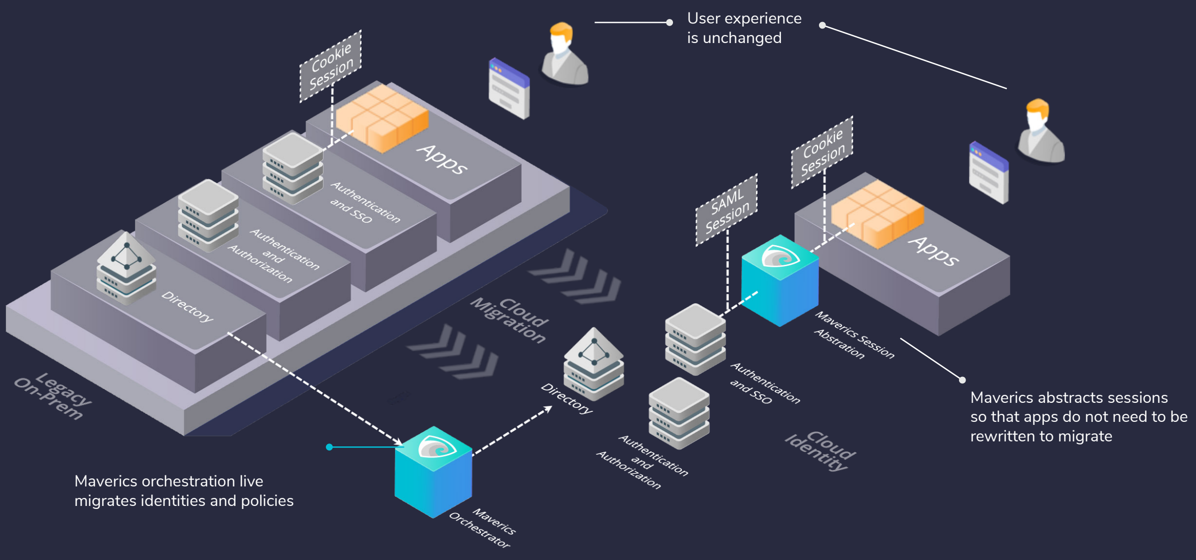
1. **Multi-Cloud Identity.** As companies adopt multiple cloud platforms they need consistent access to apps that run across those clouds. Composite apps with composite identity attributes from distributed identity systems also require an identity fabric. The following graphic illustrates the need to have consistent policies and identities across all cloud platforms.

Google Cloud
Microsoft Azure
Policy
Identity
amazon web services

2. Hybrid Extend Enterprise Apps to Cloud Users. The second use case requires that access to on-prem web apps be extended to cloud users. When identity policy is managed in the cloud, Single Sign On can be provided from the cloud through a proxy to the on-prem app. Also, maintaining consistency between on-premises and cloud identity is essential for hybrid use cases. The following illustration shows how on-prem apps can be extended to cloud-based users.



3. Migrate from legacy identity to modern identity. The third use case is providing a solution to migrate legacy identity to modern identity. This requires discovery of the legacy identities and then a legacy-to-modern live migration coupled with session abstraction. This approach allows you to migrate apps without rewriting them.



User experience is unchanged

Maverics abstracts sessions so that apps do not need to be rewritten to migrate

Maverics orchestration live migrates identities and policies

Detailed Solution Criteria

Let's take a closer look at the key criteria that you'll want to evaluate during your vendor review and product selection.

Multi-Cloud Policy

There are several criteria for cloud policy that are vital to ensure your existing identity policies can be managed across multiple clouds:

Discovery of policy	From which identity systems can the solution extract policy? (E.g., SiteMinder and OAM, Okta, Azure AD)
Types of policy	Does the solution manage policies for Authentication, Access, Auditing/Logging, Account Validation, MFA, Configuration and Conditional Access?
Mapping of policies across different platforms	Does the solution enable consistent access policies by filling functional and policy gaps between legacy and cloud systems?
Extending complex policies to work with simpler policies	Does the solution enable consistent access policies by filling functional and policy gaps between legacy and cloud systems?
Runtime execution of policies with callouts to custom APIs and services	Does the solution provide extensions (JavaScript, Golang) to incorporate custom logic or callout to APIs and services?
Declarative policy definition	Can you consistently manage policies and identities across multiple identity systems with a unified API?
Manage policy in version-controlled Git repository	Does the solution support declarative policy config as code using common source control tools? (Bitbucket, GitHub, etc.)
User Access Policy Standards	Is there support for IDQL?

Multi-cloud Identities

You'll also want to evaluate multi-cloud identity support with the following criteria in mind:

Support for provisioning and management of user data?	Does the solution manage local accounts, user profiles and user credentials across distributed identity platforms?
Support for real time migration of identities?	Does the solution support incremental and transparent migrations of users?
What sessions are managed?	Can the solution support HTTP headers, legacy SSO cookies, custom attribute providers, contextual access, custom authentication schemes and plug-ins?
Support for provisioning to the cloud and from the cloud?	Can the solution sync provisioning of users from cloud-to-prem and from prem-to-cloud?
Support for orchestrated profiles?	Can the solution assemble a users' profile from multiple identity systems?
Which identity standards are supported?	Does the solution support SAML, OIDC, OAuth, SCIM, and IDQL standards?

Identity Abstraction and Integration

Strata's Maverics provides an abstraction and integration layer that facilitates the three main use cases discussed earlier.

Support for a one-to-many integration model?	Does the solution support a one-to-many integration model, e.g. integrating apps with an Identity Fabric that integrates with many, easily interchangeable identity systems?

Support for no code integration?	Does the solution support No Code integration with Connectors and API abstraction?
Adding new integrations?	Does the solution allow for adding new connections without code? Are connectors deployable using configuration and continuously updated without separate redeployment?
How disruptive is your implementation and deployment?	Does the deployment require significant changes to the environment? Can the solution use "app gateways" to proxy access to apps and identity systems without changes?
How are identity and apps abstracted?	Does the solution support session abstraction and emulation technology that supports multiple session types including cookies, HTTP headers, OIDC, SAML, and others?
How is last-mile integration supported?	Can the solution extend cloud identity systems that only support SAML or OIDC to legacy apps that don't support those protocols?
How is custom logic for authentication, attributes and authorization supported?	Does the solution support a service extension framework (JavaScript, Golang) that makes runtime callouts to APIs or web services or to incorporate custom logic?
Is there support 'thick-client' apps?	Does the solution support thick client applications in addition to browser-based apps?

Enterprise-Grade Capabilities

Do you support Incremental roll-outs?	Does the solution support agile deployments using an app-by-app, live incremental migrations approach instead of an all-at-once or big-bang model?

Describe your security model	Is the solution secure from the core supporting a zero-trust deployment model, mutual authentication and data encryption, with customer-controlled secrets and key management using secure Vaults from HashiCorp and Azure? Does the configuration enforce a secure by default model? Can it protect against side-door access to apps via IIS or Apache sidecars?
Describe the Manageability of your solution	Does the solution support centralized management of distributed orchestrators, with APIs for configuring everything?
How DevOps Friendly is your solution?	Does the solution integrate with DevOps tool sets using APIs, CLIs, and YAML-based config?
What impact on our network, infrastructure and apps does your solution have?	Does the solution support a simple deployment with zero-touch implementation using DNS and proxies, no app-rewrite required?
Describe how your solution works with existing networking?	Does the solution support flexible networking topologies and work with F5, Palo Alto Networks, Cisco, Juniper, and other enterprise networking solutions?

Architecture

Does your solution run on-premises or in the cloud or both?	Does the solution have a natively distributed architecture that works across both on-premises and cloud and across multiple identity systems?
Describe the scalability of your solution	Does the solution have a cloud-scale proxy that can handle 25,000+ concurrent sessions and is horizontally scalable?
Describe the APIs of your solution	Does the solution provide APIs to do all configuration and policy management?
How is your solution configured and controlled?	Does the solution have a powerful CLI used to configure and operate identity orchestrators?
What operating system does the solution run on?	Does the solution run on common enterprise operating systems such as Windows and Linux?

What cloud-native architectures does the solution support?	Does the solution deploy in containers using container orchestration platforms (Kubernetes) and work with infrastructure as code (Terraform, Chef, Puppet) and CI/CD tools (Jenkins, Vagrant)?

Ease of Implementation

Describe your Proof of Concept process	Does the vendor offer an Express Proof of Concept (Express POC) in 3 Days or less?
How can Strata manage implementations, especially in quarantine conditions?	Can the vendor support remote implementations using only collaboration tools such as Zoom, Slack, and Microsoft Teams?
How much coding is required to deploy your solution?	Can the solution support Zero-code Connectors that deploy from a lightweight integrated packaged with configuration-only? Can the solution enable you to migrate apps from one identity system to another without any rewrites?
Describe the typical deployment?	Does the solution use App Gateways and Orchestrators that perform orchestration, and a Zero-touch implementation that drops in with minimal changes beyond DNS?
How long does it take to deploy?	Does the vendor offer 30 Day Get-to-Production services?

Pricing and ROI Justification

Describe your pricing model	Does the product use expensive user-based pricing or more cost effective per-app and infrastructure-based licensing?
What ROI do customers experience with the solution?	Does the solution deliver measurable ROI from retiring legacy, avoiding rewrite costs, and retiring infrastructure?

Conclusion

The adoption of multi-cloud platforms has created numerous challenges to securely manage identities and policies. Nearly every company that has adopted the cloud still has legacy applications that run behind their firewall. Each cloud provider has created silos of identities that must be managed separately with a greater cost of time and expense.

These three use cases are the key identity challenges faced by organizations today. Multi-cloud environments require consistent identities. Extending on-prem apps and identities to the cloud is also required so that users can securely access legacy applications. Modernization of identities includes the migration from legacy identity to modern identity systems.

The solution to these challenges is to use a natively distributed architecture in an enterprise-ready way, with easy implementation and strong ROI. Strata's Maverics Platform™ solves these key areas of identity: managing multi-cloud identities, extending access to on-prem apps and migrating legacy identity. To see how, please request your own demo at **www.strata.io.**