Understanding and Selecting Identity and Access Management for Cloud Services

Version 1.0
Released: June 14, 2013

Securosis, L.L.C. 515 E. Carefree Blvd. Suite #766 Phoenix, AZ 85085 T 602-412-3051 info@securosis.com www.securosis.com

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Contributors

The following individuals contributed significantly to this report through comments on the Securosis blog and follow-on review and conversations (in alphabetic order):

Gerry Gebel

Jackie Gilbert

Tom Kemp

Nishant Kaushik

Nirav Mehta

Darren Platt

Chris Wraight

Licensed by Symplified

symplified.

About Symplified

Symplified enables IT organizations to simplify user access to applications, regain visibility and control over usage and meet security and compliance requirements. Symplified provides single-sign-on, identity and access management, directory integration, centralized provisioning, strong authentication, mobile device support and flexible deployment options. Symplified is headquartered in Boulder, Colorado, and can be found online at www.symplified.com.

Copyright

Table of Contents

Introduction

We have been hearing about federated identity and Single Sign-On (SSO) services for the last decade, but the fundamental *need* for these features has only fully manifested in the last few years. Companies originally wanted to integrate internal applications and services with central identity management systems to reduce management effort, but those challenges now seem trivial. It is cliché to speak of cloud computing and mobile devices as disruptive innovations, but these advances really *have* forced a complete rethink of how we accomplish Identity and Access Management (IAM) — *and that is a very good thing*! Managing users on cloud and mobile resources *outside* your corporate network — on third-party systems outside your control — is not just a simple change in deployment models. We are presented with new risks, stemming both from the change in the way services are offered, but the way users wish to access those services. Cloud computing forces a fundamental shift in how we handle authentication, authorization, and provisioning. Enterprises want to extend capabilities to their users across low-cost cloud service providers — while maintaining security, policy management, and compliance. But they cannot simply use the same enterprise IT tool, deployed under a network perimeter security model, to cloud scenarios. Making use of cloud services *as if* they were your own in-house systems is the goal, but extending identity and access management capabilities requires new mental and technical models for successful transitions.

If you want to understand emerging Identity and Access Management (IAM) architectures, it's best to start by forgetting what you know. The directory services we use today (most often LDAP and Active Directory) were designed in the client-server age, and their implementations generally presuppose a closed system. Third-party cloud services, and to a lesser extent mobile computing, have forced a fresh approach that embraces decentralization. We liken the change from in-house directory service to Cloud IAM as that of moving from an Earth centric view of the universe to a Sun centric view: it's a complete change in perspective. We are talking about the fusion of multiple identity and access management capabilities — possibly across multiple cloud services — for computers and devices not fully under your control. We are developing the ability to authorize users across multiple services without distributing credentials to each and every service provider.

Our principal goal for this research paper is simple: Present the trends in IAM in a clear fashion so that security and software development professionals understand the new services at their disposal. We will show how cloud computing is driving extensible architectures and standardization of identity protocols, and how identity and authorization is orchestrated across in-house IT and external cloud services. Changes to IAM architectures provide the means to solve multiple challenges; additionally, external service providers offer commoditized integration with the cloud and mobile devices — reducing development and management burdens.

Right now, behind the scenes, new approaches to identity and access management are being deployed — often seamlessly — into cloud services we already use. Classic enterprise IAM is largely about provisioning users *and* resources into a common directory, such as Active Directory or RACF, where the IAM tool enforces access policy. The cloud changes this model to a chain of responsibility, so a single IAM instance **cannot** completely mediate access policy. A cloud IAM instance has a *shared* responsibility, for example, for assertion and/or validation of identity. Carving up this set of shared access policy responsibilities is a game changer for the enterprise. The risk and complexity of mapping identity to public or semi-public infrastructure are reduced, while retaining sufficient flexibility to take full advantage of *multiple* cloud service and deployment models.

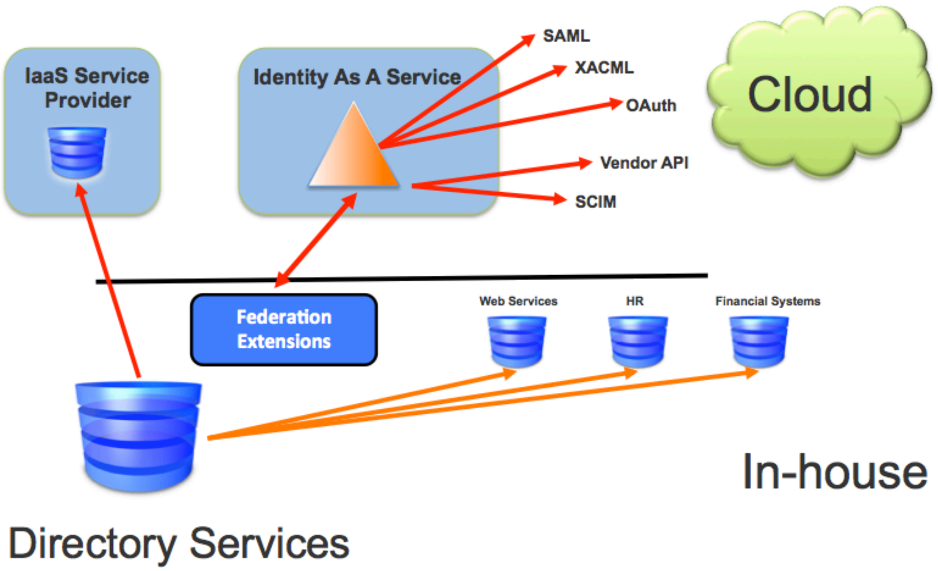
This report will illuminate these changes, and how the underlying technologies have evolved to satisfy these new demands. Cloud services — be they SaaS, PaaS, or IaaS — are not just new environments in which to deploy IAM tools you are familiar with. It is not just how IT resources are deployed in the cloud and how consumers interact with them that have changed — the changes are driven by major economic shifts in efficiency and scale. In order to take advantage of cloud services — characterized by their elastic, on-demand, web-enabled nature — for heterogenous clients you need to take advantage of new approaches to managing trust and identity.

Solution Space

Cloud computing excels at providing enterprises with applications and data on demand, regardless of user location and client-side capabilities. Employees can access corporate data and applications without using corporate servers or 'dialing-in' through an VPN connection. But how can we manage identity information with third-party providers when we remove applications from the control of on-premise identity management systems? Companies are trying to figure out how to retain control of identity management while taking advantage of the cloud. The goal is to unify identity management for internal and external users, across both traditional IT and third-party cloud services.

It is possible to manage user access to cloud computing resources in-house, but the architecture must take integration complexity and management costs into account. Most organizations — particularly enterprises — find these inconveniences outweigh the benefits. For a variety of common reasons (including on-demand service, elasticity, broad network access, reduction in capital expenditures, rapid deployment and total cost) companies adopt cloud computing services to replace in-house services and leverage third-party cloud services to manage identity and access.

Managing identity was much simpler under the client-server computing model, when users were mostly limited to a single desktop PC with perhaps another set of credentials to access a handful of servers. Set up the Access Control Lists (ACLs), sprinkle on some roles, and *voila*! But as servers and applications multiplied, the 'endpoint' shifted from fixed desktops to remote devices and servers were integrated into other server domains (never mind ACLs and roles — what realm are we in?), we used directory services as a single identity management repository and propagated identity across the enterprise. Now we have an explosion of external service providers: financial applications, cloud storage, social media, workflow, CRM, email, collaboration, and web conferencing, to name just a few. These 'extra-enterprise' services are business critical but don't link directly into traditional directory services.



The diagram above illustrates the change in architecture and deployment for identity, with the first step cloning directory services, moving to outsourced Identity as a Service (IDaaS) that extend in-house capabilities to the cloud:

Cloud computing services turn traditional identity management on its ear. The shift comes in four main parts:

1. IT no longer owns the servers and applications the organization relies upon.
2. Cloud provider capabilities are not fully compatible with existing internal systems.
3. Network centric security model, with clearly defined concepts of 'insider' and 'outsider', are gone.
4. The ways users consume cloud services are radically different when you consider mobile devices and mobile apps. An employee may consume corporate cloud services without ever touching in-house IT systems, and mix personal and professional services.

Just about every enterprise leverages one or more Software as a Service (SaaS) providers, and many are taking their first steps with Platform and Infrastructure as a Service (PaaS and IaaS, respectively) as well — each with its own approaches to Identity and Access Management. Extending traditional corporate identity services outside the corporate environment is non-trivial — it requires integration of existing IAM systems with cloud service providers. Most companies rely on *dozens* of cloud service providers, each with a different set of identity and authorization capabilities, as well as different programmatic and web interfaces. The time, effort, and cost to develop and maintain links with each service provider can be overwhelming.

Cloud Identity Solutions

Ideally we would like to *extend* existing in-house identity management capabilities to third-party systems, minimizing the work for IT management while delivering services to end users with minimal disruption. We want to maintain control over user access — adding and removing users as needed and propagating new authorization policies without significant latency. We also want to collect information on access and policy status to help meet security and compliance requirements. And rather than build a custom bridge to each and every third-party service, we want a single — and simple — management interface that extends our controls and policies to various third-party services.

Features and benefits common to most cloud identity and access management systems include:

- **Single Sign-On (SSO) Authentication:** This core service consists of a) authentication of user credentials, and b) provision of user access to multiple internal and external services without further supplying credentials to each service. Offering SSO to users is, of course, just about the only time anyone is happy to see the security team show up — so make the most of it!
- **Identity Federation:** Federated identity collects identity and authorization settings from *multiple* identity management systems, enabling different systems to define user capabilities and access. Identity and authorization are a shared responsibility across multiple authoritative sources. Federated identity is a superset of authentication and single sign-on. Federation has become much more relevant as a conveyance engine for SSO and Web Services. Its uptake in the cloud has been substantial because its core architecture helps companies navigate one of the thornier cloud issues: retaining in-house control of user accounts while leveraging cloud applications and data.

- **Granular authorization controls:** Access is typically not an 'all-or-nothing' proposition — each user is allowed access to a subset of functions and data stored in the cloud. Authorization maps instruct applications as to which resources to provide each user. How much control you have over each user's access depends on the capabilities of both the cloud service provider and the IAM system. The authorization industry and the cloud are both evolving to focus on finer-grained access control, removing access policy from code as much as possible. In a nutshell, roles are **necessary** but **not sufficient** for authorization — you need attributes too. You also do not want to spelunk through millions of lines of code to define, review, change, or audit them so rules should be configurable and data-driven. Authorizations standards are available, but maturity and adoption are nascent.

- **Administration:** Administrators generally prefer a single management pane for administering users and managing identity across multiple services. The goal of most cloud IAM systems is to do just that, but they need to offer granular adjustments to authorization across different applications while pulling data from different identity authorities. This requires central administration to link with different identity authorities, as well as with services that consume identity information.

- **Integration with internal directory services:** Cloud IAM systems rely on integration with in-house LDAP, Active Directory, HR systems, and other services to replicate existing employee identities, roles, and groups into cloud services. It's leveraging in-house sources that reduce the amount of work to enable cloud services, and provides a trusted starting point for Cloud IAM. In-house IAM services remain the central authority for in-house identity, but delegate responsibility for cloud access management to the cloud IAM service.

- **Integration with external services:** One of the core benefits of cloud IAM providers is that they offer connectors to common cloud services so you don't need to write your own integration code. The interface to communicate with Salesforce is different than the interfaces to Box, Evernote, Dropbox, Amazon Web Services, and Rackspace. Not even the largest companies want to write custom interface code to every external cloud service they use. Leveraging third party 'glue' to connect to common SaaS, PaaS, and IaaS vendors makes integration easier and faster.

Key Issues with Identity Services

Additional capabilities of various services include multi-factor authentication, mobile user integration, and support for multiple user personas. These features help tie traditional identity management into cloud services. But as vendors attempt to solve these issues, cloud IAM creates new problems that don't get as much attention.

- **APIs:** While IAM vendors offer connectors to the most common cloud services, they are unlikely to provide all the connectors you need. And cloud providers change APIs on a regular basis, sometimes breaking their customers applications. You will likely need to either provide your own integration, leverage tools from IAM service providers for building custom connections, or contract your IAM vendor to build these for you. This raises the additional challenge of on-boarding third-party developers and giving them appropriate access rights.

- **Authorization Mapping:** There are many possible ways to specify authorization rules, such as by role vs. by attribute. Existing access rules are likely to need rewriting for a cloud service provider, after all the objects (such as URLs and data) you are granting access to are provided by the cloud provider.

- **Audit:** In-house systems can be linked with log management and SIEM systems to produce compliance reports and provide monitoring and detection of security events. Audit logs from cloud service providers remain problematic — the multi-tenant model prevents most providers from providing full logs because they could disclose data on *other* customers.

- **Privacy:** Users, user attributes, and other information are often pushed outside your corporate network and into one or more cloud data repositories. The security and privacy controls for these external repositories are not fully under your control, so you need to explore what data your vendor copies, what they use 'in-place', and determine both IAM vendor and cloud service provider security controls on copies that reside in the cloud.
- **Latency:** Propagating rule changes from internal IAM to cloud IAM can take some time. For example, if an employee is terminated or has their access rights reduced, there may be a lag between the internal change and when the cloud service enforces the change. Latency is a subject to discuss with both your IAM provider and cloud service provider.
- **Privileged User Management:** This has been a problem for a long time, and the cloud adds a new wrinkle. Historically privileged users were all employees, and if things went pear-shaped you could handle it as an HR event. In the cloud that breaks down.
- **App Identity:** Once you have the user logged in you might still need to verify the application they are using — or perhaps there is no user at all, just middleware. But where did the request come from? The sad truth is that as long as you know how to call the service, many applications today do not verify the client at all. Like the previous point, this has been an IT problem for a long time.
- **Mobile**: Security teams are still absorbing the cloud's implications but technology does not wait around — we have a whole new paradigm to tool up for. Mobile security is particularly relevant *not just because* for the cloud because it's viewed by users as a convenient — and often principle — way to consume cloud services. It's because mobile is part of the cloud; SaaS, IaaS and PaaS form the service-side of cloud computing, mobile devices form the consumer side. As we mentioned in the introduction, mobile connections to cloud services occur outside of the boundaries of normal IT operations. This means yet another domain to manage, one which is loosely coupled from in house directory and authorization systems. While we cannot possibly cover mobile IAM in sufficient detail in this paper, there are several trends worth mentioning: The first is the concept of conditional authorization. In essence this means that users rights may be limited if they are outside the corporate network or using mobile applications. The context of what a user is trying to do, and under what conditions (geo-location, device type), are factored into authorization by the application platform. The second trend is the need for two-factor authentication (2FA), and not just relying on the assumption your user is in possession of a mobile device. Requiring additional verification that a user is who they say they are makes it harder for attackers to simply steal devices — or identity tokens — and act as the user across dozens of different cloud services. Once you appreciate that a smartphone is essentially a multi-tenant smart card, you can see that mobile security is fundamentally an identity problem, and why these trends are the early indicators that mobile becomes a convergence point for identity and security.
- **Identity Store Location**: If companies are moving their applications and data to cloud services, will they also move existing identity stores? Some firms view this as a logical evolution, while others have security and governance requirements that require identity and authorization data be maintained in-house. As the concepts of 'inside' vs. 'outside' continue to erode, and the differentiation between 'enterprise' and 'cloud' applications blur, both the risks and benefits of moving identity stores will be at the center of debate in the coming years.

Use Cases

Cloud computing poses (sometimes subtly) different challenges and requires rethinking IAM deployments. The following use cases illustrate the principal motivators mentioned by organizations moving existing applications to the cloud — for both internal and external deployments — and how they integrate with third party cloud services.

IAM architecture often feels quite abstract — describing its traits is a bit like postulating how many angels can dance on the head of a pin or whether light behaves more like a particle or a wave. And then there are standards — lots and lots of standards. But use cases are concrete — they show the catalyst, the activity, and the value to the enterprise and users. Start with use cases and *then* look for identity technologies and standards that fit, rather than the other way around.

To help understand why cloud computing requires companies to rethink their Identity and Access Management strategies we provide a handful of cases to illustrate common problems. These cases embody the drivers of IAM deployment structure, and the need for new protocols to propagate user privileges and establish identity in distributed environments.

Key IAM Concepts

Before we get to the use cases themselves, we need to describe the types of IAM actors. There can be numerous different roles in a cloud IAM system, but the following are found in most deployments:

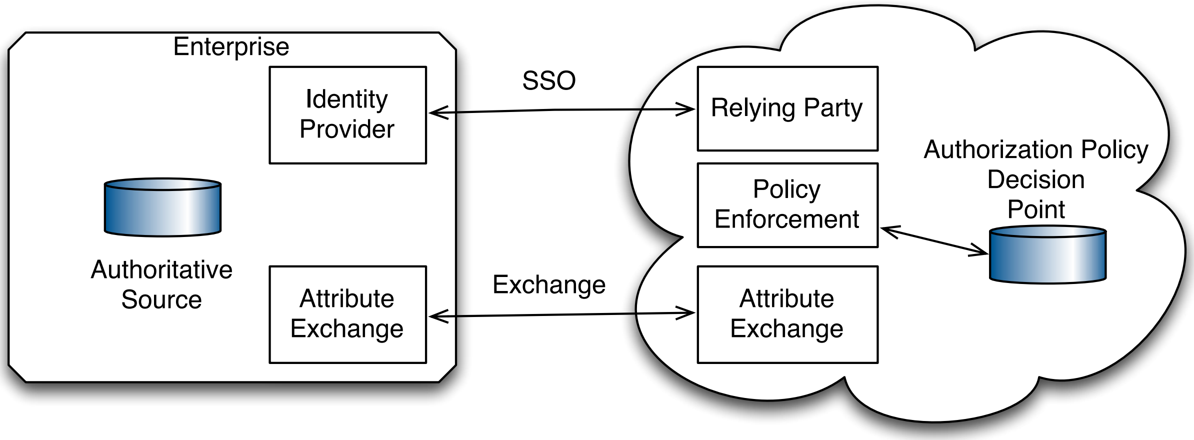
- **Identity Provider:** Consulted at runtime, the IdP is an authoritative source of information about users. This is often an Active Directory or LDAP server — which in turn provides attributes the IdP uses to mint tokens which represent user identities. Cloud computing architectures often include more than one IdP. The IdP is usually implemented 'in-house' on the enterprise system.

- **Relying Party:** An RP is an application that relies on identity information asserted by an Identity Provider (or an Attribute Provider, see below) to establish identity. The relying party validates tokens as genuine and from an authoritative identity provider, and then uses them to assert user identity. RPs are usually implemented on the cloud provider's system. In some cases an RP looks to leverage the 'network effect' of multiple trusted network services help validate a user is who they say they are.

- **Attribute Provider:** An AP either has access to, or directly stores, the fine-grained attributes that define user capabilities. The AP differs from an IdP in that it uses attributes associated with a user as opposed to passwords. The AP uses things like phone number, address, credit card number as a consumer oriented way of identifying users. Permissions may be role-based, attribute-based, or both. The value is in enabling dynamic data-driven access control. This information is critical — it defines application behavior and gates user access to functions and data. How it provides attribute information and integrates with the application varies greatly.

- **Authoritative Source:** This is the authority on identity and provisioning settings. The AP is typically the HR system which stores master identity records, used as the source of truth for account status. This system has rights to add, edit, and disable accounts from other systems — typically via a provisioning system. For legal and compliance requirements these systems keep detailed transaction logs.

- **Policy Decision Point:** A PDP handles authorization decisions by mapping each access request to a policy. This may be performed in application code or within a separately configured policy.



There may be other IAM roles in your deployment but this is the core set for cloud IAM. The key is that identity and access management roles are broken down into simple functions and each one is distributed to one or more in-house or cloud platforms. The location of each service varies depending on the deployment model you select. Each role may be fulfilled by the cloud provider and/or the enterprise, but these roles distribute the fundamental responsibilities for cloud identity services.

Use Cases

Most cloud deployments address some combination of these three IAM use cases.

Single Sign-On

Single Sign-On is the single greatest motivation for companies to look at new IAM technologies to support cloud computing. And for good reason — during our careers in security we have seen few occasions when people are glad to see security features introduced. Single Sign-On (SSO) is a happy exception to this rule because it makes every user's life easier. Supply your password *once*, and you automagically get access to every site you use during the course of the day. Adding new cloud applications (Salesforce, Amazon Web Services, and Dropbox, for example) only makes SSO more desirable. Most security does not scale well, but SSO does.

Behind the scenes SSO offers other more subtle advantages for security and operations. In an SSO deployment the Identity Provider provides a central location for policies and control. The user store behaves as the authoritative source for identity information, and by extending this capability to the cloud — through APIs, tokens, and third-party services — the security team can avoid some of the worry about discrepancies between internal and cloud accounts. The Identity Provider effectively acts as the source of truth for cloud apps. Please note that for this use case we focus on SSO for cloud services; while we recognize that SSO for mobile is an equally compelling use case, we cannot fully do that subject justice in the scope of this paper.

But while we have mastered this capability with traditional in-house IT services, extending SSO to the cloud presents new challenges. There are many flavors to SSO for the cloud, some based on immature and evolving standards, others proprietary and vendor-specific. Worse, the mechanisms by which identity is 'consumed' vary, with some services 'pulling' identity directly from other IT systems while others require information to be 'pushed' to them. Finally, the protocols used to accomplish these tasks vary as well: SAML, OAuthv2, vendor APIs, and so on. Fortunately SAML, as the agreed-upon standard, is offered by major SaaS and IaaS providers, and is becoming increasingly common with PaaS vendors who previously focused on passwords provided through web APIs.

Another challenge to cloud SSO is the security of the identity tokens themselves. As tokens become more than just simple session cookies for web apps, and embody user capabilities for potentially dozens of applications, they become more attractive as targets. An attacker with an SSO token gains all the user rights conveyed by the token — which might provide access to dozens of cloud applications. This is less of an issue if all the protocols adequately protect tokens communicated across the Internet, but some do not. So SSO tokens must be protected by TLS/SSL on the wire, with a protection regime for token access and storage by applications.

SSO makes life easier for users and administrators, but for developers it is only a partial solution. The sign-on process is simplified but the granularity of attributes and hooks into the Relying Party authorization code still require careful design and development.

Finally, we have the inverse problem of SSO: SLO (Single Logout). "What do we do when a user logs out?" If the user is logged into many applications, do we terminate just one session or all of them? Even if you want all sessions to close, most SSO systems are unable to clean up running sessions. You will need to review session management and activity timeouts in conjunction with SSO system capabilities. These must be tested to ensure the applications themselves do not keep sessions open despite policy. SLO is almost universally mentioned in company RFIs and RFPs, but in practice is rarely implemented. Instead companies rely on automatic session timeouts and other techniques for cleaning up SSO sessions.

Provisioning

As we add new cloud applications and services, how can we enable user access to them? This is called provisioning, and is how we provide user account information to cloud applications. There are several possibilities, and customers may employ one of more, typically combined into a provisioning process:

I. **Account registration:** Create a new account in the cloud service, through user sign-up or possibly seeding information from an authoritative source such as the HR System.

II. **Account propagation:** Synchronize or replicate accounts to the cloud provider, usually from directory services.

III. **Account management:** Update account status, attributes, groups, roles, and other account information.

IV. **Account disable:** 'De-provision' or disable access to functions or applications.

V. **Audit:** Track the end-to-end activities of users and management of access rights. This includes both ongoing activity monitoring and periodic policy review for concerns such as separation of duties.

This lifecycle can be managed within the company, with synchronization to the cloud provider. Or it can be performed entirely in the cloud if the additional risk of having a complete set of user credentials within the provider's (likely multi-tenant) environment is acceptable. Handling *all* provisioning remotely in the cloud generally requires greater effort. The third option is outsourced Identity Management as a Service — cloud services for IAM. A handful of vendors can bridge internal systems with external cloud providers, taking care of the synchronization and integration of cloud services. A harbinger of how much work is moving to the cloud, vendors now offer the IDM lifecycle via dedicated IDM clouds with associated management UI.

Attribute Exchange

Traditional enterprise IT and cloud provider systems need coordinated data management. Information may be held by one or more parties so synchronization is key. Data exchange occurs through gateways, web services portals, Enterprise Service Buses (ESBs), or application identity — systems that make decisions based on application identity rather than user identity. You can think of this as a form of provisioning, but how data is provided, and how it is used by the application, is fundamentally different.

Application identity is one area where cloud providers are often ahead of standard practice in IT, because each host is typically assigned its own account rather than being managed through an *ad hoc* process. These identities — whether ESB, gateway, application server, or other — tend to be longer-lived, so different protocols may be in play. Application identity *may* be managed with “old school” x.509 certificates — as with Amazon’s AWS — but these attribute exchanges lend themselves to protocols designed to handle such issues, such as SCIM for provisioning and SAML for SSO. For finer-grained operations or ‘fresher’ data, x.509 is often not good enough. Security Token Service (STS) may be used to broker identity interactions. One factor to consider in this use case is the role of the client. In some patterns such as API security, WS-Trust, and OAuth, the client is responsible for performing the push and pull legwork of setting cookies and session data and navigating the protocol dance between client and server. In patterns such as SAML and browser-based security, the client may be passive, with the server initiating operations and the client receiving and responding to messages.

Whether old or new school, Attribute Exchange use cases are the duct tape of identity: they cover a wide variety of integration scenarios from sharing identity on the latest smartphones to backend legacy messaging systems. Identity is the management of sharing, and attribute exchange helps to scope interactions. No single product or protocol can achieve full end-to-end coverage, but it is essential for the architect to factor in the full chain of responsibilities to maintain the quality and integrity of identity data.

Integration with Cloud Providers

The term "the cloud" is so misapplied and overused that it no longer has meaning without context. When discussing cloud identity and access management, it is important to keep in mind the relevant cloud context(s): Infrastructure, Platform, and/or Software as a Service. Each of these models (SaaS, PaaS, and IaaS) presents its own unique challenges for IAM because each model promotes different approaches and each vendor offers their own unique flavor. The cloud service model effectively acts as a set of constraints which the IAM architect must factor into their architecture.

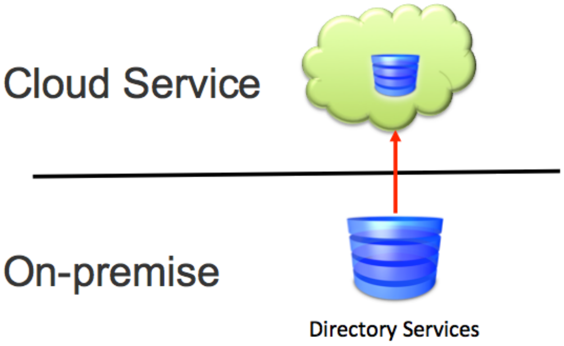
With the SaaS model most enterprises look to federated identity, meaning the enterprise uses federation capabilities to gate access to cloud applications, keeping account provisioning control internal. This approach is both simpler and offers better security policy control than the primary alternative of replicating accounts into the SaaS provider — copying big chunks of user directories into the cloud. A middle road has emerged lately, where account management is through an Identity as a Service (IDaaS) cloud provider; we will discuss it later.

With IaaS identity federation is an option as well, but the need is not as great because everything is managed above the Infrastructure level. Infrastructure providers offer some built-in identity management capabilities, but since you control most of the user-visible infrastructure, extending your existing capabilities into the cloud is a more natural progression. IaaS vendors such as Amazon have offered limited support for federation over the years, but the quality and depth of their functionality demonstrates that they largely expect customers to handle identity. PaaS, as usual, is somewhere in between. Most PaaS service providers offer fairly robust capabilities, so federation is a first-class choice in most major platforms today.

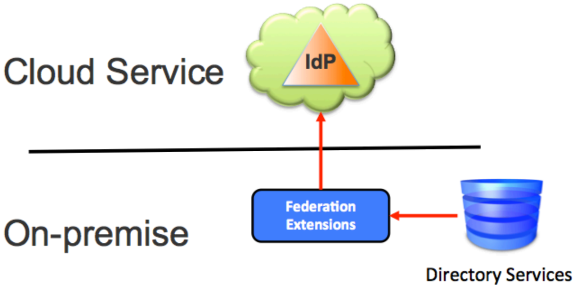
Cloud Identity Deployment Models

IAM deployments for the cloud generally use some combination of the following approaches:

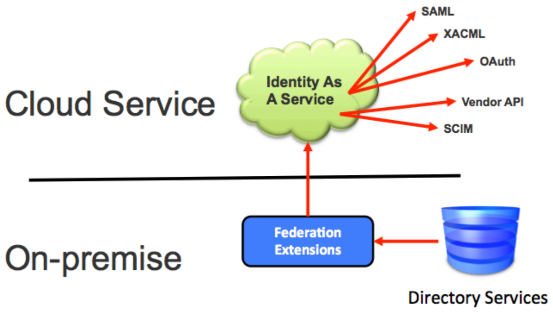
- **Store Accounts in the Cloud:** This is exactly what it sounds like: replicate or synchronize accounts into the cloud environment. This is conceptually simple, and most IT departments are already familiar with directory replication so it is the easiest way to get started. It is also a model which creates security problems when you *replicate* — and potentially expose — sensitive information including keys, passwords, and other access control data. Remember, "the cloud" is naturally a multi-tenant environment, shared by other customers and administrators. Role changes, as well as account removal or disabling, can lag unacceptably before the internal change is effective at the cloud provider.



- **Identity Federation:** The next option is federation, where user identities are managed locally but identity assertions can be validated through tokens presented to the cloud service interface. The enterprise retains local control and management, typically via Active Directory, and requests are typically made dynamically. This avoids storing secret data such as passwords in the cloud. Federation lets the enterprise leverage existing IDM processes, possibly across multiple systems, which simplifies management and provisioning.



- **IDMaaS:** Identity Management as a Service is an emerging architecture to watch. This is effectively a hybrid of the first two approaches. A separate cloud manages identity, usually run by a different cloud service provider, who links directly to internal on-premise systems for policy decision support. One of the major advantages of this approach is that the IDMaaS provider then links you to various cloud services, handling all their technical idiosyncrasies behind the scenes. The IDMaaS provider effectively glues everything together, providing identity federation and authorization support. However, be careful that your provider is actually offering IDMaaS and not simply copying all your data into their cloud environment, in essence becoming an outsourced data center.



Vendor Services

Most cloud deployments today are intended to leverage federated identity, but this is where the commonality stops. The ways identity is used, and types of IAM services available, cover a broad range of possibilities.

- **Authentication:** This deceptively simple concept is devilishly hard in practice. As IT managers and programmers we understand in-house authentication — but what about cloud applications, middleware, and proxies? What about privileged users? Where and how is each session controlled? How are authentication events audited? Authentication is one area most enterprises think they have nailed down, but edge cases and external interfaces add complexity and raise questions which have not yet been fully addressed.

- **SSO:** Single sign-on is table stakes for any cloud provider, but its delivery varies between providers and service models. SSO is a subset of federated identity and provides the seamless integration users demand. Seek out cloud providers with open-standards-based SSO to reduce complex and costly integration work.

- **Federation:** Federation of identity is another *sine qua non* for SaaS, and a basic capability of most cloud service providers. Key success factors include integration for the federation servers to cloud consumer and provider.

- **Authorization:** Identity defines “who is who”, and authorization then defines what identified users can do. Traditionally users have been assigned roles which define what they can do and access. This streamlines administration because a single role change can update many users. But roles are not good enough any more: applications need attributes — granular characteristics — provided by an authoritative source. The question is where are they stored and maintained? Cloud side, enterprise side, or both? Policy-based authorization via XACML is steadily becoming more common. Today

XACML is more likely to factor into PaaS and IaaS deployments, but it is likely to be the *de facto* authorization standard in the future.

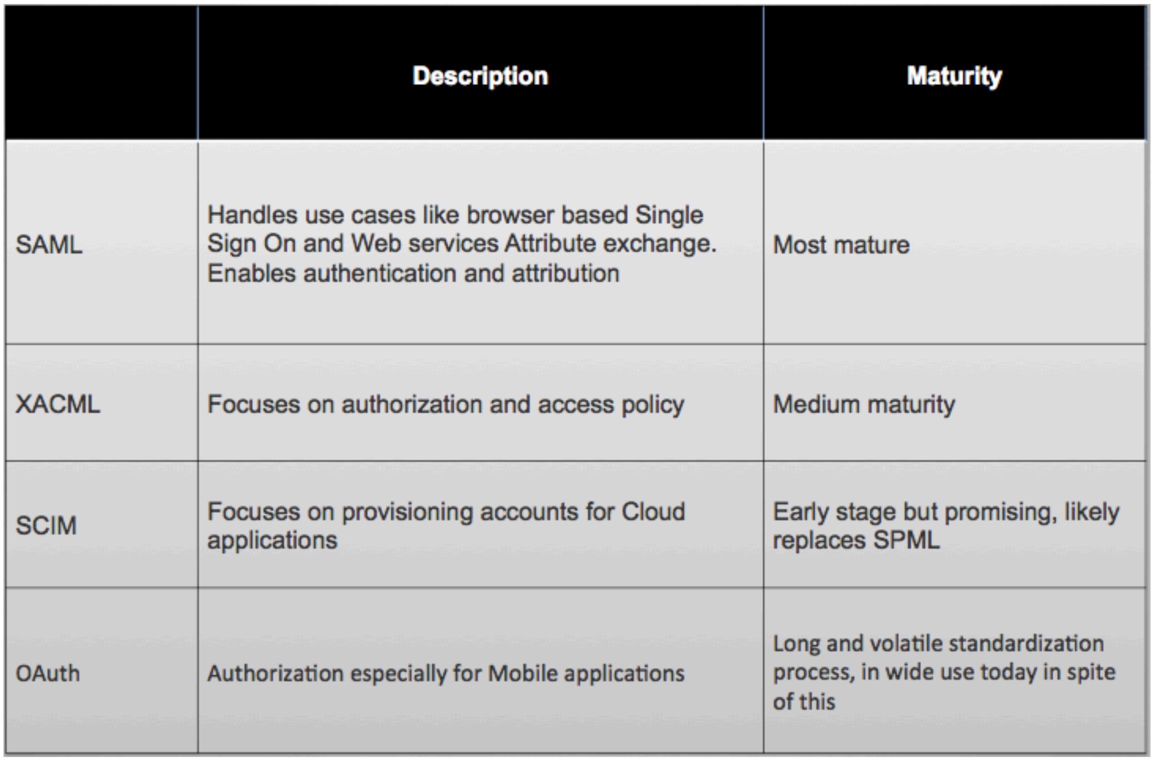
- **Provisioning:** Account and policy lifecycle management — including user account creation, propagation, and maintenance — is primarily an enterprise function. Cloud applications are just another endpoint for the provisioning system to communicate with to. SCIM is a proposed standard for provisioning, but some people use SAML beyond its intended scope to effectively replicate SCIM's provisioning functions.
- **Audit:** Cloud applications, including identity management offerings, need to provide APIs to extract information so reviews of access history can be reviewed. This is an issue and a major point of differentiation between cloud IDMaaS vendors. Some cloud deployments don't offer API level access, and those that do don't always offer a complete picture given issues of log files containing data from other cloud tenants. A handful offer full capture of authentication and user activity. In many cases, authentication and authorization services need a way to publish events to an audit repository, which is generally Log Management of SIEM located on the enterprise side. This is another 'must-have' for enterprises with corporate governance, but current auditing capabilities are neither ubiquitous nor complete. Unfortunately there are no identity audit standards to date, and we expect piecemeal solutions on a vendor by vendor basis for the foreseeable future.
- **Governance, Risk and Compliance**: Cloud services pose a challenge to most enterprises. Cloud providers want to be a simple vending machine; offer commodity services on demand. But like a vending machine, handling special requests and exposing *how* services are delivered is not part of the deal. Enterprises, on the other hand, need to fully verify security controls, and demonstrate the compliance with contractual and governmental requirements. Issues like segregation of duties, employee vetting, certification, audit logs and enforcement of privacy in multi-tenant environments are all critical areas enterprise customers want to verify. Opacity of cloud provider internal systems does not mesh with transparency requirements of corporate governance. These two conflicting approaches limit the ability for some large enterprises to adopt cloud services in general, and IDMaaS specifically.

Standards

Identity is the plumbing of the web. Identity standards are the building blocks of cloud identity, and they help connect all the parts together. But there are many different moving parts, each with a different job, so there are many different standards. The good news is that each standard excels at an aspect of identity propagation. The bad news is that there are many identity standards to choose from, so it is hard to know which standard does what, or which is the right one for you. If you are building your own solution choose carefully — cloud IAM architecture should be standards-based, but the standards should not drive the cloud IAM architecture. One of the advantages of working with third party IdaaS vendors is that they already have this working, which saves time and effort. You integrate with the IdaaS vendor, who provides connection glue to other cloud services on your behalf.

When thinking about identity challenges it is helpful to separate the task to be performed from the IAM standard that does the work. As an example, the following is one possible set of standards to accomplish a full complement of IAM tasks:

But not every standard enjoys wide adoption. In fact many ‘standards’ are in their infancy, with only limited adoption even among cloud and identity service providers. Further, it requires both parties in a federated relationship to support the same approach to SSO/Federation for it to work. The following chart shows relative maturity of common standards:



	Description	Maturity
SAML	Handles use cases like browser based Single Sign On and Web services Attribute exchange. Enables authentication and attribution	Most mature
XACML	Focuses on authorization and access policy	Medium maturity
SCIM	Focuses on provisioning accounts for Cloud applications	Early stage but promising, likely replaces SPML
OAuth	Authorization especially for Mobile applications	Long and volatile standardization process, in wide use today in spite of this

Integration

Unlike most IT functions relating to identity, the enterprise is still responsible for quite a lot.

- **Process Integration:** Provisioning and policy management systems are less technology-centric than business-process-centric. Making these cloud-ready requires updating processes to account for cloud providers’ roles, responsibilities, and limitations.

- **First Mile Integration:** To establish federation, the enterprise identity provider must be integrated with an authentication and/or attribute source. For IDMaaS this is usually an on-premise appliance or piece of software that provides a communication tunnel to the service provider. Most often this is custom code or federation extensions atop existing directory services.

- **Last Mile Integration:** The cloud provider’s systems must be able to consume identity and bootstrap users into cloud applications, and the identity’s Relying Party must have adapters to integrate with the cloud provider. The communication protocols may be industry standards such as SAML or proprietary APIs.

IAM Architecture & Design

It is time to discuss architecture and deployment models for identity, as well as access management for cloud services. This is a complex subject — we have to cover three different cloud service models (SaaS, PaaS, & IaaS); in three different deployment scenarios (public, private, & hybrid); with a variety of communication protocols to address authentication, authorization, and provisioning. The Cloud Security Alliance has cataloged many different identity 'standards', but the fact that we have dozens to choose from illustrates how unresolved this whole field is. It is *not* one size fits all — work is required to determine which standard can best solve *your* problems. Worse, cloud providers' standards support is often incompatible with others in the field, so you are likely to need custom code to connect and share identity information.

The point is that discussion of IAM 'standards' is often a *starting point* for companies considering cloud identity. But **standards should not drive architecture** — projects are much better driven by use cases and risk. Our goal is to define an overall architecture which fits your organization and then fill in appropriate communication standards. To help disentangle design from implementation standards we offer design patters to describe the architecture. A design pattern is a universal model that both abstracts and simplifies the structure from underlying environmental complexities. For each use case we describe a design pattern that address the core challenges of propagating identity information across multiple services. Then we can discuss how IAM technology standards fit those models.

As previously discussed, there are three core cloud IAM use cases: Single Sign-On (SSO), Provisioning, and Attribute Exchange. Delivering on these use cases requires a number of architectural decisions and workarounds for various issues.

SSO Architecture and Design: Learning from the Pin Factory

> *"One man draws out the wire, another straights it, a third cuts it, a fourth points it, a fifth grinds it at the top for receiving the head: to make the head requires two or three distinct operations: to put it on is a particular business, to whiten the pins is another ... and the important business of making a pin is, in this manner, divided into about eighteen distinct operations, which in some manufactories are all performed by distinct hands, though in others the same man will sometime perform two or three of them." — Adam Smith, 1776*

SSO is often implemented using a 'federation' model, under which each user's identity and associated attributes are stored across multiple distinct identity management systems. Which identity management repository within the larger federated group determines whether to validate a user is determined dynamically at request time. Federated identity is tailor-made for the cloud because it cleanly separates responsibilities between the enterprise and the cloud provider. As in Adam Smith's pin factory, each participant can specialize in the areas they are best able to handle, with the identity protocol establishing modes of exchange between participants.

SAML has been the dominant standard in this area, used by enterprises and cloud providers to coordinate SSO. SSO architectures implement one or more Identity Providers (IdP) to act as authoritative sources for account information. The IdP is generally on the enterprise side but may also reside in a separate *external* IdP cloud. SAML is the runtime identity protocol for SSO exchanges and the IdP is the linkage point between the service provider (the external application) and the provisioning services which manage the accounts (typically the HR database).

Relying Parties (RP) generally reside at the cloud provider; their task is to consume and verify identity assertions and require proper access rights to cloud applications. The agreement between the IdP and RP defines the identity protocol, how it is initiated (from enterprise and/or cloud provider side), the schema, which attributes are sent, and any additional details.

Federated identity enables the enterprise, as the party with the freshest and most accurate user information, to control and manage accounts. The cloud provider controls the application side and can consume and use assertions from the enterprise without the burden of user management.

Federation enables Single Sign-On for an open, interactive application architecture. But that requires coordination of in-house and cloud systems, across Internet connections that may not be 100% reliable. Any standard used for federation must be resilient in the face of unreliable networks, as well as privacy and integrity attacks, at both browser and protocol layers.

Provisioning Architecture and Design: Process Automation

Provisioning systems are architected very differently than the SSO and federated systems discussed above. The focus is less on architecture and more on process, particularly how and when systems communicate. These systems don't need real-time synchronization — they often run in batch mode, perhaps hourly or even daily. Provisioning systems function as back-office support applications so design requirements center around integration — largely to accommodate the byzantine protocols necessary to communicate with directories and vendor systems. The good news for security is that these services are less exposed than other systems, requiring neither browser integration nor direct exposure to users.

Provisioning processes, such as 'on-boarding' new users, updating accounts, and managing users, are highly automated. They must be to accommodate thousands of users and scale across dozens of cloud services. The back-end processes to update and synchronize data are critical in traditional on-premise IAM systems, to ensure users don't gain unwarranted access to data or retain access rights longer than appropriate. Extending these functions beyond the corporate IT perimeter is inherently difficult. Like football referees, these systems are only visible to users when they fail.

The unique aspect of provisioning cloud applications is its focus on process automation across a least two companies — the enterprise and the cloud provider. We don't hear many success stories of process automation across multiple companies. Fortunately the account handoff to the cloud provider is relatively simple.

There are three key architecture decisions for provisioning systems, but in the end they all come down to the same thing: location, location, and location.

1. Where will the authoritative source reside? In the cloud or inside the enterprise?
2. Where will the target (IdP) reside?
3. Who manages the provisioning system and related policies? Which interfaces are used?

Provisioning has not historically been a focus of standardization efforts. SAML is a well-used standard but SPML (Service Provisioning Markup Language) has not achieved broad adoption and appears unlikely to ever do so. SCIM (System for

Cross-domain Identity Management) has largely filled this gap, and appears well positioned to provide a standard that satisfies cloud requirements.

SCIM was designed for use across domains, making it ideal for cloud services. Schemas cover users and groups so authentication and authorization information can be updated. Basic operations are available through a simple REST API — including creation, reading, replacement, deletion, updating, searching, and bulk updates of users and groups. That helps map existing processes into these operations and extend them to cloud applications.

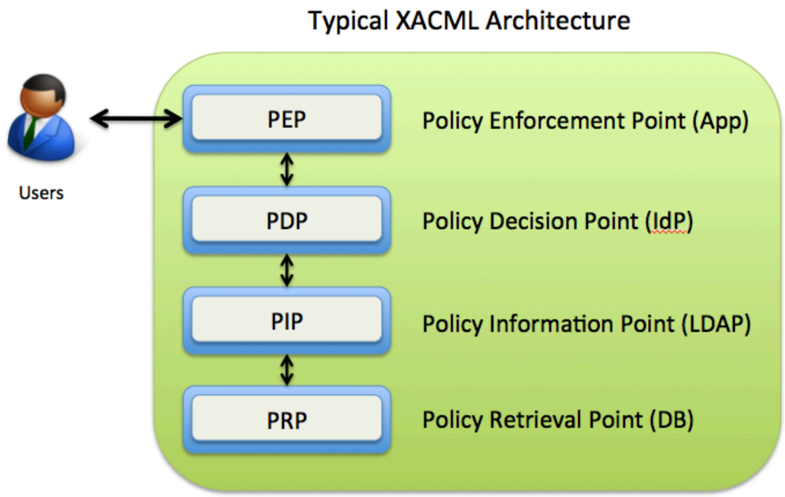
Attribute Exchange Architecture and Design … and Everything Else …

Of course there is more to identity than SSO and provisioning. Attribute exchanges are used to provide information about users and applications and to make attribute-based access control decisions. For example an attribute might be a unique user identifier, an indicator of group membership, or an indication of policies which apply to the user session.

This is where our three use cases combine to deliver value. Provisioning systems provide fresh and accurate information from the authoritative source (directory services). Identity protocols such as SAML move it around within the system, ferrying data to cloud applications. But this is all for naught unless and until those attributes are used to enforce access control decisions. User identity and authorization maps must actually *do* something, and attribute exchanges are how we set these policies. We don't want attributes hard-coded into applications — especially because SaaS providers serve many different customers. Instead it should be quick and easy to change policies without recoding applications or services.

The beginning of this journey is the Policy Enforcement Point (PEP) that enforces policy by first gathering attributes and passing them to a Policy Decision Point (PDP), and the PDP that maps target rules to answer yes/no access questions. The PDP then provides the PEP with the information it needs to authorize use. These two building blocks form the simple producer/consumer design pattern for central policy creation and management with distributed enforcement.

XACML (eXtensible Access Control Markup Language — recently standardized to v3.0) provides an end-to-end standard for authoring authorization rules and policies. Chief among its advantages is removal of authorization logic from code. This makes policies easier to change and audit, as well as more dynamic, so authorization can be data-driven rather than hard coded.


Typical XACML Architecture
Users
PEP
Policy Enforcement Point (App)
PDP
Policy Decision Point (IdP)
PIP
Policy Information Point (LDAP)
PRP
Policy Retrieval Point (DB)

Unlike the SSO and provisioning architectures, the majority of the access control runtime is likely to reside at the cloud provider. The enterprise can serve as policy author and expose a policy information point (attributes used for access control decisions) but for performance and integration reasons it almost always makes the most sense to place policy decisions and enforcement on the cloud provider side.

Where does this leave the enterprise?
The XACML standard alone cannot provide everything needed. To take advantage of it enterprises must be able to write

and update policies to drive authorization on the cloud provider side. This can be managed at several different levels of granularity, from coarse-grained roles (RBAC) to fine-grained dynamic Attribute Based Access Control. The common thread across all these approaches is a defined entry point which exchanges attributes to drive the access control workflow.

The point of standards is not compliance for its own sake, but to make things work. The purpose of identity standards is to establish proven and reviewed protocols that avoid vendor lock-in to any cloud or identity vendor, at low cost. We mention this because we often hear companies say “standard XYZ is our identity architecture.” But that misses the point: the *use case* should drive architecture — standards should play a supporting role. Think of them as the lumber or concrete you build a house with, not the design of the house.

Implementation Roadmap

IAM projects are complex, encompassing most IT infrastructure, and can take years to fully implement and roll out. Trying to do everything at once is a recipe for failure, so we offer an implementation roadmap to help reach your goals without biting off more than you can chew. We will center our discussion around how to build an architectural schema for your particular organization, based on the cloud service and deployment models you selected. Then we will create different implementation roadmaps, depending on your project goals and most critical business requirements.

Previously we described three common use cases for Cloud IAM: Single Sign-On, Provisioning, and Attribute Exchange. The good news is that the process of creating a deployment roadmap is largely the same regardless of which use case you choose. But every customer's environment and priorities differ, so delivering on these use cases requires a slightly different implementation and plan for every customer.

Strategy First

Implementation roadmaps start with a system design and then describe the series of steps needed to deploy the solution in phases. The roadmap should begin with the assumption that there will be a lot of catch-up to play because few organizations have cohesive identity strategies. Unfortunately dedicated identity teams are rare — much less a VP-level position managing IAM as a critical function. This work is largely left to unlucky souls who zigged when they should have zagged, and as a reward got the title "IAM Architect" tacked onto an existing laundry list of responsibilities. These people, overwhelmed by complexity, tend to punt and outsource the problem to consultants. The predictable result is a patchwork of partially implemented tactical solutions.

We don't want to be too negative but we started this section in Debbie Downer mode because a) you are unlikely to successfully solve the problem without appreciating its magnitude, and b) your plans need to take the current state of IAM in your company into account. With these considerations in mind you can realistically decide which problems to address first — taking into account the available organizational, process, and technology support. Try not to think of cloud IAM as yet another point IAM solution. The total rethink of IAM prompted by cloud computing offers more flexible and effective solutions than were available before. So we urge you to adjust your thinking, consider where identity solutions will be useful, and figure out how one of the cloud architectures we have described can extend your capabilities.

Based on your architecture, select set a list of priorities, and then list which capabilities are present within your existing infrastructure.

Roles As a Process Guide

Let's drill into our use cases and focus specifically on the 'actor' roles, mapping how they interact with each other. We touched on several common roles: Identity Provider, Relying Party, Attribute Provider, Authoritative Source, and Policy

Decision Point. A good first step for outlining your strategy is to figure out which servers will fill these roles. Second, determine how the parties will communicate and what information they need to exchange. This process map should provide a good understanding of how all the pieces work together, which feature will be important, and what data must be available. Your map should include constraints imposed by these system actors — for example, the cloud application Relying Party likely accepts a limited set of identity tokens. Understanding limitations early is just as important as knowing the feature requirements.

Each role needs to communicate with other roles. Communications are often taken for granted — but It's that Internet / cloud thing, so it must all be HTTP, right? Mostly, but not always. Communication could be via API calls, or HTTP communications might rely on supplementary SSL/TLS for security. Each party needs to establish mutual trust to avoid being duped by malicious parties. To avoid surprises and last minute fire drills over firewall rule changes, trace out the necessary end-to-end communications and protocols. Often requirements for non-HTTP protocols are buried deep beneath the surface — this is particularly common for provisioning. Security issues crop up due to information leakage, session security, spoofing, and other concerns, so it pays to examine the dialogue between parties and specify secure communications during the design phase.

Filling the Gaps

As we alluded earlier, the state of play in IAM is frequently a hodgepodge of stuff, with various components bolted on to solve specific problems that arose over time. This forces many IAM projects to burn considerable calendar time on data cleanup and transformation. Again, the starting point is a schema for identity and accounts used for cloud access decisions. It is critical to understand what work must be performed and identify the most difficult integrations.

From there the order of implementation is heavily influenced by how much of a mess you need to clean up. We caution that simple is best — do not try to build a be-all end-all über-identity-schema. Even when schema definition is straightforward, enforcing it across multiple backends rarely is. It is important to review data sources, ensure they work with the identity schema, and establish a process for cleaning up and dealing with conflicts. Realistic expectations are essential — be conservative about what can be achieved and don't get too aggressive out of the gate. Do not copy your feature list from a vendor's capabilities document and assume everything will "just work". Be conservative — less is more.

One final word on building your schema: You need to understand not only how things work, but also what happens when they *don't* work. Identity and access have ugly failure modes; when they break people notice and you get the blame. Plan for failures at each node within your schema and understand the side effects when each service goes down; are there interesting complications if two or more go down at the same time, or in the worst possible sequence? Can your system withstand periodic brief outages? You need sufficient testing to discover serious issues *before* production deployment. But most security and QA tools are not well suited to testing IAM. So for each use case you deploy, build a set of test cases (both positive: this should work; and negative: this should fail) to ensure that what you are promoting works end-to-end. These tests can influence deployment timeline as problems are discovered — better to find and fix problems now than while or after moving the system into production. Most users don't care about *your* technical issues — they just need this stuff to work, every time.

Operational planning should also include building a runbook: documentation of the installation, configuration, logging, and administrative tasks needed to keep the IAM system running. For cloud applications this requires careful planning and coordination because some roles and responsibilities are new, and roles are shared between cloud vendors and the enterprise. You need to understand which roles you manage and which are handled by your cloud service provider. For some cloud deployments (IaaS, private cloud, and some PaaS) you can configure the infrastructure to ensure logging, system configuration, and administrator roles are fully defined before launching any instances. With SaaS and IDaaS you need verification from your cloud vendor.

Buyers Guide

How can you decide which approach is right for you? We present a list of common questions to help define what you need from cloud identity services, and the type of deployment model that will best serve you. No two customer environments or lists of requirements are the same, but key decision criteria help narrow down the field to suitable platforms. We will provide questions to help determine which vendors offer solutions that fit your architecture, as well as a set of criteria to measure the appropriateness of a vendor solution to your design goals and help you work through the evaluation process.

Your mileage WILL vary

Spoiler alert — there is no such thing as plain vanilla IAM. You may need a solution for customers as well as users. You may or may not need to include mobile devices. You may need fine-grained authorization controls for external applications, and you'll likely need detailed audit trails as well. There are far too many variables in play for IAM evaluation to be fully quantifiable. But to help you weed out some players, to align your needs with product function, and to give you a handle on major product differentiators, we created the table below. Make sure any products fit your goals. Even IAM suites that can do everything on paper have their own strengths and weaknesses, so make sure you know them before leaping in.

As we have discussed, prospective buyers should start with understanding their use cases and governance processes before analyzing the IAM marketplace. That said, here is a proposed checklist for beginning to analyze IAM products:

Capability	Evaluation Criteria	Notes
Product Architecture	What are the key capabilities in the product?	
	What are the internal data models and formats for identity?	
	What are the external data models and formats for identity?	
	How does the product scale? Vertically or horizontally?	

Capability	Evaluation Criteria	Notes
Development	What is the development interface for implementing and extending the product?	
	Is development typically done by the company or third-party consultants? Answer for both development and maintenance phases.	
	What integration is available for source code and configuration management?	
	What languages and tools are required to develop wrappers, adapters, and extensions to extend the product?	
Interoperability	What IAM standards are supported and where are they supported in the product? Which interfaces are standards based and which are proprietary?	
	What directories are supported — Active Directory, LDAP, etc.?	
	What application servers are supported - WebSphere, IIS, Tomcat, SAP, etc.	
	Are cloud applications supported? Which ones?	
	Are mobile platforms supported? Which ones?	

Capability	Evaluation Criteria	Notes
Product Security	What is the product security model?	
	Is Role-Based Access Control supported? Where?	
	How is access audited?	
Use Case Support	How does the product support Provisioning use cases?	
	Describe the product's support for Single Sign-On use cases?	
	Describe the product's support for attribute exchange use cases?	
	What user self-service capabilities does your product support?	
Cost Model	How is the product licensed?	
	Does the cost scale based on number of users, number of servers, or something else?	
	What is the charge for adapters and extensions?	

This checklist provides a starting point for analyzing IAM products. As the evaluation unfolds it is key to remember what matters: integration, standards, and cost. The buying process works much better if the initial step includes an inventory of IAM sources and targets: where identity is used and the authoritative sources. You will need an understanding of both what identity sources you have today and what external applications you are likely to use within the next 24 months. So ask, "What IAM processes exist currently, and which are desired in the future?"

POC FTW

Securosis highly recommends a Proof-of-Concept (POC) as a final step for IAM buyers. PowerPoint does not crash much but new implementations do. There is nothing like seeing a product working in your own environment.

If there is more than one vendor in play — and there usually is — then bake-offs can be useful to determine the best fit. But we generally do not recommend bake-offs with more than two vendors. Many vendors take widely different conceptual approaches to IAM problems, and in-depth evaluations are too demanding to perform more than once or twice. Start with an initial review against our checklist to weed out unsuitable candidates. Then use a proof of concept to test viability and/or a bake-off to compare a couple similar candidates.

Closing Comments

Identity and Access Management is a tremendously complex subject — and we have just scratched the surface to give you an overview of current trends and solutions. You could spend your entire career studying the nuances of how all these pieces work together. It is likely that IAM is only *part* of your job, so we don't expect you to master everything we have discussed immediately. We hope you find this paper useful and know many of you will have questions on features, technologies, and implementation specifics. If you have any questions or want to discuss your particular situation, feel free to send us a note at info@securosis.com.

About the Authors

Adrian Lane, Analyst and CTO

Adrian Lane is a Senior Security Strategist with 25 years of industry experience. He brings over a decade of C-level executive expertise to the Securosis team. Mr. Lane specializes in database architecture and data security. With extensive experience as a member of the vendor community (including positions at Ingres and Oracle), in addition to time as an IT customer in the CIO role, Adrian brings a business-oriented perspective to security implementations. Prior to joining Securosis, Adrian was CTO at database security firm IPLocks, Vice President of Engineering at Touchpoint, and CTO of the secure payment and digital rights management firm Transactor/Brodia. Adrian also blogs for Dark Reading and is a regular contributor to Information Security Magazine. Mr. Lane is a Computer Science graduate of the University of California at Berkeley with post-graduate work in operating systems at Stanford University.

Gunnar Peterson, Associate Analyst

Gunnar Peterson is a Managing Principal at Arctec Group. He focuses on distributed systems security for large mission critical financial, healthcare, manufacturing, and insurance systems, as well as emerging startups. Mr. Peterson is an internationally recognized software security expert, frequently published, an Associate Editor for IEEE Security & Privacy Journal on Building Security In, a contributor to the SEI and DHS Build Security In portal on software security, a Visiting Scientist at Carnegie Mellon Software Engineering Institute, and an in-demand speaker at security conferences. He maintains a popular information security blog at http://1raindrop.typepad.com.

About Securosis

Securosis, L.L.C. is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services.

Our services include:

- The Securosis Nexus: The Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics, telling you exactly what you need to know, backed with industry-leading expert advice to answer your questions. The Nexus was designed to be fast and easy to use, and to get you the information you need as quickly as possible. Access it at <https://nexus.securosis.com/>.
- Primary research publishing: We currently release the vast majority of our research for free through our blog, and archive it in our Research Library. Most of these research documents can be sponsored for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements and conform with our Totally Transparent Research policy.
- Research products and strategic advisory services for end users: Securosis will be introducing a line of research products and inquiry-based subscription services designed to assist end user organizations in accelerating project and program success. Additional advisory projects are also available, including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessment.
- Retainer services for vendors: Although we will accept briefings from anyone, some vendors opt for a tighter, ongoing relationship. We offer a number of flexible retainer packages. Services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluation, and merger and acquisition assessment. We maintain our strict objectivity and confidentiality. More information on our retainer services (PDF) is available.
- External speaking and editorial: Securosis analysts frequently speak at industry events, give online presentations, and write and/or speak for a variety of publications and media.
- Other expert services: Securosis analysts are available for other services as well, including Strategic Advisory Days, Strategy Consulting Engagements, and Investor Services. These tend to be customized to meet a client's particular requirements.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Additionally, Securosis partners with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis. For more information about Securosis, visit our website: <http://securosis.com/>.