Cloud Federation and Federated Access Control

Cloud Strategy Partners, LLC

Sponsored by: IEEE Educational Activities and IEEE Cloud Computing
This IEEE Cloud Computing tutorial has been developed by Cloud Strategy Partners, LLC. Cloud Strategy Partners, LLC is an expert consultancy firm that specializes in Technology and Strategy relating to Cloud Computing.
In this tutorial, we will discuss access control basics including defining terms and illustrating generic authentication and authorization models. Next we will review federated access control and identity management models and mechanisms with a focus on OAuth2.0, Shibboleth, OpenID, and SAML. This is followed by a review of cloud federation models that include both client side federation and provider side federation. We will also discuss AWS Identity and Access Management (IAM), Microsoft Azure Active Directory (AAD) and OpenStack Keystone.
Outline

In this lesson we will cover

- Access Control basics
- Terms definition
- Generic Authentication and Authorization models; Federated Access Control and Identity
- Management models and mechanisms
- OAuth2.0, Shibboleth, OpenID, SAML Cloud federation models
- Client side federation and Provider side federation; AWS Identity and Access Management (IAM); Microsoft Azure Active Directory (AAD); OpenStack Keystone

Access Control Terms

Access Control is a fundamental part of Cloud Computing. One cannot have a secure application running on a Cloud unless one has a secure Access Control system to the Cloud stack supporting it.

First let us look at Access Control Terms

Authentication ("AuthN") is the process of identifying a user or an access subject, based on identity credentials which examples are username and password, digital certificates, one-time-tokens, etc. Authentication refers to the confirmation that a user/subject who is requesting services is a valid user of the resources or services requested.

Authorization ("AuthZ") is the process of granting or denying a user access to cloud or network resources once the user has been authenticated. Authorization is typically defined by the authorization enforcement policy which is managed by the resource owner.

Access Control includes two based services: Authentication and Authorization, and is typically complemented by the Identity Management service

AAI (Authentication, Authorization Infrastructure) refers to the basic access control services Authentication, Authorization and supporting them additional services such as identity and attribute management service, namespace and trust management

AAA (Authentication, Authorization, Accounting) Framework: Provides a basis for intelligent/integrated access control, accounting and auditing services providing information necessary for services protection, management and billing

Federated AAI (FedAAI) multi-domain and multi-provider infrastructure that uses federation mechanisms to allow users' access to resources in one domain (foreign domain) based on authentication in another domain (home domain) - FedAAI enable Single Sign On service for the whole federation
Identity Management describes the management of identities of individual principals, their attributes and privileges within or across system and enterprise boundaries.

Security Credential is an assertion of a characteristic or a property of a subject/entity that can be used for Authentication or Authorization. Security credential must be digitally signed to protect its integrity and optionally encrypted to protect confidentiality.

**Accessing/Requesting Remote Service: Transport Level Security**

A fundamental requirement of Access control is Transport level security.

Transport level security provides secure channels over insecure network for services communications and data exchange. It begins with Authentication between communicating systems. This is done as a part of TLS/SSL or HTTPS protocols. SSH protocol is used from creating a security channel between two host or end points; other protocols can communicate over this channel.

An HTTP header can carry Authentication credentials that can be in a form of security tokens or browser cookies. The HTTP body then may contain a payload message; this is the normal mode for the SOAP or REST protocol. The slide contains an illustration which shows how Transport level security works with this variety of protocols.

**Authentication (“AuthN”) by User Home Organization Service**

A continuing part of the total Access Control picture is Authentication.

This slide contains an illustration showing how a typical AuthN system would be implemented.

The idea is that the User’s home organization would authenticate the User, and assert a set of attributes for that User. This would be done by an attribute authority in the User home organization.

In order for that User of that Organization to access a Cloud Service Provider, there are two alternatives. One alternative is completely manual, where the User Home Organization has no pre-arranged mechanism to access the Cloud Service Provider using the User Domain AuthN which they already have established. In this case they access the Cloud Service Provider as any other individual would.

In the case where automated (Single Sign On) access to the Cloud Service Provider is desired, a gateway would be set up with a Trust/Federation relationship to the User Organization. This way the User Home Organization has a pre-arranged mechanism to access the Cloud Service Provider using the User Domain AuthN which they already have established.
As illustrated, this Gateway can then derive the Authorization (from the Authentication) needed to access the Cloud Service Provider. Please follow the steps through on the Illustration.

**User or Service Request Authentication**

In the real world, often the Authentication occurs because there is a pending User or Service Request. In this case there are Two options to perform authentication:

- By Service Provider (SP) Authentication is requested after receiving Service Request. SP AuthN service request user identity and attribute information from user HO.
- By user Home Organization (HO) User obtains AuthN and Attribute assertions in advance from HO and include it in the Service Request. SP AuthN service validates AuthN and Attribute assertions.

The User identity information can be extracted from the Service Request. This identity can come is several ways:

- Username, Password as a simplest option PKI based message authentication (digitally signed message).
- And In the case of a positive authentication AuthN assertion (AuthnAssert) is issued by Identity Provider (IDP).
- Attribute assertion (AttrAssert) is issued by Attribute Authority Service (AAS).

**PKI Based Authentication**

Let us now look at obtaining User identity information

Everyone knows about Username, Password as a simplest option. So lets explore PKI based message authentication (digitally signed message).

Prerequisites: User obtains X.509 Public Key Certificate (PKC) from the trusted Certification Authority (CA).

CA acts as a Trusted Third Party (TTP) to make assertion about User ID.

Private key (privkey) is stored on user computer.

Messages based Authentication

Service Request message is signed with privkey PKC is included/attached with the message.

Server/receiver checks signature if it is valid and checks if PKC is trusted.
Remote login PKI based Authentication
Client gets from the service a challenge number e.g. in a form of nonce (number used once)
Client (modifies and) signs a challenge/nonce with privkey and attach PKC
Server verifies signature and checks PKC

Authorization
Authorization adds “capabilities” and “permissions” types of information to the now
Authenticated User. These information types are driven by “Policy”.

The Generic Authorization model includes 3 basic functions or component which take part in
the Policy mechanism:
Policy Enforcement Point (PEP),
Policy Decision Point (PDP)
Policy Authority Point (PAP)

Policy Enforcement Point receives AuthZ related information from AuthZ/Security gateway.
Policy Enforcement Point creates Authorization Request and sends it to Policy Decision Point
Policy Enforcement Point checks validity of the provided AuthN and Attribute assertions
(AuthN context) Policy Decision Point retrieves policy from Policy Authority Point and
evaluates request, and makes decision –Permit, Deny or conditional Permit/Deny Policy
Decision Point typically has standard implementation and uses specific policy language, e.g.
XACML

Policy Enforcement Point in case of Permit
Either simply relies on the ServReq or creates AuthZ assertion containing AuthZ context

General Federated Access Control Model
Now that we’ve gone through most of the concepts, lets put it together into a General identity
federation and federated access control set of sequences and trust relations:
Trust relation exists between a Service Provider and an Identity provider in the same domain
For federated access control and identity provisioning, there must trust relations established
between Identity Providers in different domains. Trust relations in federation mean that that
the federation members trust each other assertions and will accept each other
security/authentication tokens as valid.
The Slide contains an Illustration showing the steps for identity federation and federated
access control
Corresponding to the Illustration, the following steps usually take place during the federated
access control:
1 – Client/user requests access to a resource or service
2 – Service Provider replies with the list of trusted IDPs and/or Authentication services (AuthN)
3, 4 – Client authenticates to the trusted IDP and obtains an Authentication assertion (AuthnAssert) or a Secure Token (ST); for web browser based client ST may be provided in a form of web cookie that is stored in the browser.
5, 6 – Client presents AuthnAssert or ST/cookie to the Service Provider's Authorization service that validates presented credentials and evaluates the request against the access control policy. As a result of positive evaluation, it authorizes the Client to do a requested action or access requested resource. Authorization service may issue an Authorization assertion (AuthzAssertion) or simply forward request to the Resource.
7, 8 – Client presents AuthzAssertion to the Resource and gets access to it.

Example Sequence SSO + IDP (multi-domain): Shibboleth Operation with WAYF Service

Next let's look at an example sequence of Single Sign On and multi-domain Identity Providers. Shibboleth represents a generic federated access control model and was actually the first implementation of the privacy enforced federated access control for web applications. The slide contains an illustration showing the Shibboleth Operation with a “Where Are You From” (WAYF) Service.

The Numbers in the illustration show the sequence of events. Please follow the flow through the illustration. Shibboleth is an architecture and open-source implementation for federated identity-based authentication and authorization infrastructure based on Security Assertion Markup Language (SAML)

Federated identity allows for information about users in one security domain to be provided to other organizations in a federation. It allows for cross-domain single sign-on and removes the need for content providers to maintain user names and passwords. Identity providers (IdPs) supply user information, while service providers (SPs) consume this information and get access to secure content

OAuth2.0 – Open Authorization Protocol

A standard for Authorization is the Open Authorization Protocol, called OAuth2.0. Functionally OAuth2.0 is an Authorisation and Delegation Protocol, eg, OAuth is a standard for authorization of resources. It does not deal with authentication. Initially proposed by Facebook, currently also implemented in Windows 8 Standardized in RFC6749 The OAuth 2.0 Authorization Framework
The Slide contains an illustration of how OAuth2.0 works. Please follow the sequence as enumerated on the Slide.

**OpenID Connect**
OpenID is another authentication standard. It serves a different purpose than other authentication models. OpenID is a way to specify one identity for multiple sites so you don't need to register over and over again. OAuth (just discussed) is a way to allow one application access to one account without giving said application your account login information. So actually OpenID and OAuth can be used in conjunction with each other.

OpenID Connect 1.0 is a new specification and a new version of OpenID Authentication (known as OpenID 2.0) that has identifier format what means that relying parties need to migrate those user identifiers to continue serving their users.

OpenID Authentication 2.0 is a popular authentication federation protocol through which the relying party can obtain the user's verified identifier from the OpenID Provider (OP) to which the user was authenticated.

**OpenID Connect Protocol**
This slide contains two illustrations showing how OpenID works.
The First diagram illustrates the full OpenID protocol sequence according to specification
The Second diagram provides more visual illustrates when user accesses web application.

**SAML Assertions and Protocol**
SAML (Security Assertion Markup Language) is an umbrella standard that encompasses profiles, bindings and constructs to achieve Single Sign On (SSO), Federation and Identity Management.

It is an OASIS standard. The OASIS SAML specification defines SAML Security Assertion and SAML protocol to exchange security assertions during Authentication and Authorization.

SAML defines 3 types of assertions: Authentication, Authorization, Attribute SAML Protocol defines binding to HTTP and SOAP protocols SAML assertions can be communicated as browser cookie The slide contains an illustration which shows how SAML works.

**Cloud Federation – Scaling Up and Down**
Scalability is one of the main features in cloud. We look how it done/related to the cloud service model.
We look at two typical cloud use cases: so-called Cloud burst is when enterprise offload excessive load temporary to cloud (public or private), and cloud services migration or replication.

Scaling up means provisioning new identities or in a new location. This may also require populating session context of some ongoing processes to new location. Scaling down means de-provisioning identities and cleaning or invalidating session context. Scaling can be initiated by the provider or by the customer.

### Cloud Federation: Actors and Roles

A number of actors and roles are defined for cloud federation as defined by a number of standardization bodies.

We can mention NIST Cloud Computing Reference Architecture, OASIS Cloud Identity Federation model, IETF and EGI Federated Cloud TF.

It is important to mention that in cloud we need to distinguish between customer that may be an organization entity and user which is typically end user and a member of organizational customer.

A Cloud Service Broker is widely defined as either simply working as a 3rd party or also providing services composition and provisioning to customer or user.

An Identity Provider can be a part of CSP service/infrastructure or IDP is created as a part of the provisioned cloud infrastructure. We will look at different options later.

### Cloud Federation Models

The proposed models reflect different use cases User/customer side federation

1.1 cloud services are entirely provisioned in clouds
1.2 customer organization has own IDM and want to use user home IDs to access cloud services
1.3 this a case when customer wants that their users can use 3rd party ID provider, like OpenID, Facebook, LinkedIn, iTunes, Microsoft.com, etc.

Provider (resources) side federation
2.1 a use case for resources federation between cloud providers

Goal for defining these use cases and models is to create a basis for protocols and API's definition.
Basic Cloud Federation model (1.1) – Federating users/HO and CSP/cloud domains (no IDP-HO)

We can start with a simple model when customer organization doesn’t have own ID management service and all accounts are separately/specifically provisioned for cloud based services.

The Home Organization admin manages all resources and accounts in the cloud.

In this case The Identity Provider IDP-Xa is a virtualized service of the CSP Identity Provider.

Basic Cloud Federation model (1.2) – Federating HO and CSP domains (IDP-HO1 and IDP-CSP)

In this second case the Identity Provider IDP-Xa can be implemented as instantiated service of the CSP IDP, essentially in the cloud.

Basic Cloud Federation model (1.3) – Using 3rd party IDP for external users

In this third case, the 3rd party Identity Provider provides user identities. Virtualised IDP-Xa needs to trust identities provided by this IDP. This trust relation is established via CSP’s IDP that may play a role of identity gateway or identity broker.

Basic Cloud Federation model – Combined User side federation

This slide provides consolidated view of all 3 customer side federation model

Basic Cloud Federation model (2.1) – Federating CSP’s/multi-provider cloud resources

This diagram illustrates federation relations in cloud resources sharing and outsourcing between CSPs. Resources provided to customer by one CSP can actually belong to other resource providers.

To allow secure and trusted infrastructure all resources need to be federated.

This means that resources or actually cloud providers owning resource need to trust service requests from the main/frontend provider and further from customer and user.

Cloud Federation Model - Combined

This is another consolidated diagram that includes both customer side and provider side federation.
Intercloud Federation Infrastructure
This slide illustrates a possible end-state called Intercloud Federation. This is where Trust Brokers and Resource Brokers (also called Exchanges) can be seen as part of the public cloud infrastructure. There are also specific functions such as resource directory and federated Identity Provider Through the use of a Gateway which interfaces each cloud to this infrastructure, participating clouds can provide resources to requesting clouds which need them.

Basic AuthN and AuthZ services using Federated IDPs – For additional Credentials validation
Federated Identity Providers are a core part of many of these cloud federation schemes. This illustration in the Slide shows the main components of AAI (Authentication and Authorization Infrastructure) in federated multi-cloud environment. It explains what processes and interaction take place when a user access a resource in the provisioned cloud infrastructure or service.
There are two basic models:
Push – authentication credential is obtained by user requesting ID credentials and authentication before accessing service
Pull – authentication is done by the resource (i.e. authentication service of the resource provider)
In multi-provider environment, IDP’s are federated what means that if presented ID is not known to local IDP (i.e. running as a part of the provisioned cloud infrastructure) or Authentication service when they request information or confirmation from the home IDP from the user or resource origin.

Basic AuthN and AuthZ services using Federated IDPs – Federation/Trust domains
This illustration on the slide shows 3 federation or trust domains:
Domain 0 –a user home domain Domain 1 or cloud services domain where all federation and AAI services are run Domain R –a resource domain, actually CSP domain.
This establishes relation between CSP domain, virtual cloud service domain and user home domain.
Interaction between services has been explained on previous slide.
AWS Identity and Access Management (IAM)

It is helpful to look at examples of different Identity and Access Management capabilities of several commercial cloud offerings. First we will consider AWS IAM.

AWS IAM provides functionality to securely control access to AWS services and resources for individual users and groups by defining individual and group permissions and policies. It also provides:

- Fine-grained access control to AWS resources
- Manage access control for mobile applications with Web Identity Providers
- Integrate with your corporate directory
- Multi-Factor Authentication for highly privileged users.

The slide provides additional detail on this service.

Examples of using AWS IAM

Here are even more detailed Examples of using AWS IAM.

- Fine-grained access control to AWS resources
- Manage access control for mobile applications with Web Identity Providers
- Integrate with your corporate directory
- IAM can be used to grant your employees, and applications federated access to AWS Management Console and AWS service APIs
- Multi-Factor Authentication for highly privileged users

Top Ten IAM Best Practices

AWS has a list of best practices to help IT professionals and developers manage access control without losing flexibility or resiliency.

1. Users – Create individual users
2. Groups – Manage permissions with groups
3. Permissions – Grant least privilege
4. Password – Configure a strong password policy
5. Multi Factor Authentication – Enable MFA for privileged users
6. Roles – Use IAM roles for EC2 instances
7. Sharing – Use IAM roles to share access
8. Rotate – Rotate security credentials regularly
Transcript

9. Conditions – Restrict privileged access further with conditions

10. Root – Reduce/remove use of root

---

**Example: Multi Factor Authentication-Protected API Access Sequence**

Multi Factor Authentication-protected API access simply requires users to enter a valid MFA code before using certain functions designated by account administrators. The illustration in the slide details how the process works in the programmatic use case.

Because the AWS Management Console calls AWS service APIs, you can enforce MFA on APIs regardless of access path, either from programmatic API calls or via the console user interface.

---

**Microsoft Azure Active Directory (AAD)**

In Microsoft Azure, the central component for IAM is Active Directory, called Azure Active Directory (AAD)

Microsoft Azure Active Directory is a modern cloud service providing Identity Management and Access Control capabilities to cloud applications.

Provides Identity and access management in the cloud
Can be integrated with on-premises AD
Supports Integration with cloud applications

Microsoft Azure Active Directory provides four basic services
Microsoft Azure AD Access Control (ACS)
Microsoft Azure AD Directory
Microsoft Azure AD Graph
Microsoft Azure Authentication Library (AAL)

---

**Windows AD Server vs Microsoft Azure AD**

Microsoft Azure AD is different from Active Directory Server
Windows Active Directory
AD Domain Services AD Lightweight Directory Services AD Federation Services AD Certificate Services AD Rights Management Services Corporate environment Kerberos, LDAP, DNS

Azure Active Directory Azure Access Control Service AD RMS (Preview) Cloud based and cloud oriented REST, SAML-P, WS-Federation, OAuth, Graph API
**Microsoft Azure Active Directory (AAD)**

This slide has an illustration of how Microsoft Azure Active Directory (AAD) works

For Identity Challenges for the:

User
Doesn’t want to use different identity for every app

Developer
Doesn’t want to write code to support multiple identity providers

Administrator
Wants to easily grant access to apps to Active Directory identities

AAD provides Identity Solution: Cloud SSO with Access Control for the:

User
Can use his preferred IDP

Developer
Writes one set of code to work with multiple Identity Providers

Administrator
Grants access to all AD users by establishing trust between AD and AAD

**Microsoft Azure AD Access Control**

A cloud federation service for your cloud applications and services

Federates on-premises and cloud identity services

Prerequisites
Demands federated authentication
AD on-premises and AAD on cloud synchronization
Supports multiple identity providers
Facebook, Google, Microsoft, Windows Server AD FS, Yahoo!
Supports multiple protocols
WS-Federation, WS-Trust, OAuth 2.0 (draft 13)
Supports multiple tokens
JWT, SAML 1.1/2.0, SWT

**The Azure Graph API**

Azure AD Graph service

A Developer Restful API for the cloud directory supporting Oauth, JSON and Odata

An enterprise social graph service used to build enterprise social application (for production or business purposes)

Provides a way for applications to query the Directory and other sources for identity information and relationships, to provide a richer experience for users

The OData Protocol is an application-level protocol for interacting with data via RESTful web services.
The protocol supports the description of data models and the editing and querying of data according to those models.

**Example: AAD based Access Control and Federation**

The slide contains an illustration which shows the first of these two options, where Azure Active Directory is all that's required.

As the figure shows, Azure AD is a multi-tenant service. This means that it can simultaneously support many different organizations, storing directory information about users at each of them.

In this example, a user at organization A is trying to access a SaaS application. This application might be part of Office 365, such as SharePoint Online, or it might be something else -non-Microsoft applications can also use this technology. Because Azure AD supports the SAML 2.0 protocol, all that's required from an application is the ability to interact using this industry standard. (In fact, applications that use Azure AD can run in any datacenter, not just an Azure datacenter.)

The process begins when the user accesses a SaaS application (step 1). To use this application, the user must present a token issued by Azure AD.

This token contains information that identifies the user, and it's digitally signed by Azure AD. To get the token, the user authenticates himself to Azure AD by providing a username and password (step 2). Azure AD then returns the token he needs (step 3).

This token is then sent to the SaaS application (step 4), which validates the token's signature and uses its contents (step 5). Typically, the application will use the identity information the token contains to decide what information the user is allowed to access and perhaps in other ways.

**OpenStack Identity Service - Keystone**

Keystone provides a single point of integration for OpenStack policy, catalog, token and authentication. Keystone handles API requests as well as providing configurable catalog, policy, token and identity services.

Identity service provides validation of users authorization credentials, Roles, Tenants and associated metadata Token service validates tokens that are used by users or tenants for authentication Endpoint discovery and endpoint registry services are provided by the Catalog
service Rule based authorization is provided by the Policy service. The Keystone service can use various formats of credentials and storages such as file, SQL, PAM or LDAP. Each Keystone function has a pluggable backend which allows different ways to use the particular service. Most support standard backends like LDAP or SQL, as well as Key Value Stores (KVS). Most people will use this as a point of customization for their current authentication services.

OpenStack Identity (Keystone) provides a central directory of users mapped to the OpenStack services they can access. It acts as a common authentication system across the cloud operating system and can integrate with existing backend directory services like LDAP. It supports multiple forms of authentication including standard username and password credentials, token-based systems, and Amazon Web Services log in credentials such as those used for EC2.

Additionally, the catalog provides a query-able list of all of the services deployed in an OpenStack cloud in a single registry. Users and third-party tools can programmatically determine which resources they can access. The OpenStack Identity Service enables administrators to:

- Configure centralized policies across users and systems
- Create users and tenants and define permissions for compute, storage, and networking resources by using role-based access control (RBAC) features
- Integrate with an existing directory, like LDAP, to provide a single source of authentication across the enterprise

The OpenStack Identity Service enables users to:

- List the services to which they have access
- Make API requests
- Log into the web dashboard to create resources owned by their account

**OpenStack Keystone Interaction**

Keystone provides services to all Openstack services and functional components. The slide contains an illustration showing the various modes of OpenStack Keystone Interaction.

Keystone Identity Service and Service Catalog
Keystone creates project, and tenants assign their attributes
Create Users, Roles and Groups Map Users to Projects via Roles
Other OpenStack components (Nova, Glance, etc) look at the roles users have for a particular project and then perform their own policy based authorization based on the role information.

Keystone creates and manages Domains
Define the administrative boundaries for management of Keystone entities. A domain can represent an individual, company, or operator owned space.

Keystone acts as a Service Catalog to let other OpenStack systems know where relevant API endpoints exist for OpenStack Services.

---

**Keystone Identity Service – Sequences Server Deployment**

The illustration in this slide shows the Sequences for how Keystone is used.

1. Alice wants to launch a server: Alice sends credentials/authenticates to Keystone and gets a temporary security token ST

2. Alice requests from Keystone a list of her tenants or projects in which she has accounts. Keystone returns a list of tenants

3. Alice sends credentials to access desirable tenant/project. Keystone returns a tenant token to access available services.

4. Service verifies Alice’s token (internally via Keystone) and check if a token is authentic and valid for the requested services

5. Keystone upon successful authentication provides additional attribute or information and sends request to the service

6. Service validates request against service’s own policy and executes request, in particular creates a new Server VM

7. Service reports back to Alice about fulfilling the request

---

**Recent Keystone Functionality Improvement (OpenStack2014.1 Icehouse)**

With new improvements in the Icehouse version of 2014 the OpenStack Keystone became a Federation solution of choice for enabling customizable ID federation service private and community cloud. These are listed in the slide.
Summary and Take Away

In this Lesson we have covered:

Cloud federation is a common trend in building multi-cloud infrastructures and applications that allows integrating heterogeneous cloud platform.

Federated access control is effective in providing access control to cloud based resources and services for enterprise users where the users can use their home identities to access cloud services.

Two general cloud federation models: client side and provider side, provide a framework for user access federation and inter-cloud resources sharing.

AWS IAM services provides a solution for federated access control and Identity Management in AWS and federates.

Microsoft Azure Active Directory is a powerful platform for FedIDM and federated access control that naturally integrates with enterprise Active Directory services.

OpenStack Keystone provides flexible Identity Federation Framework that currently supports most of popular protocols.