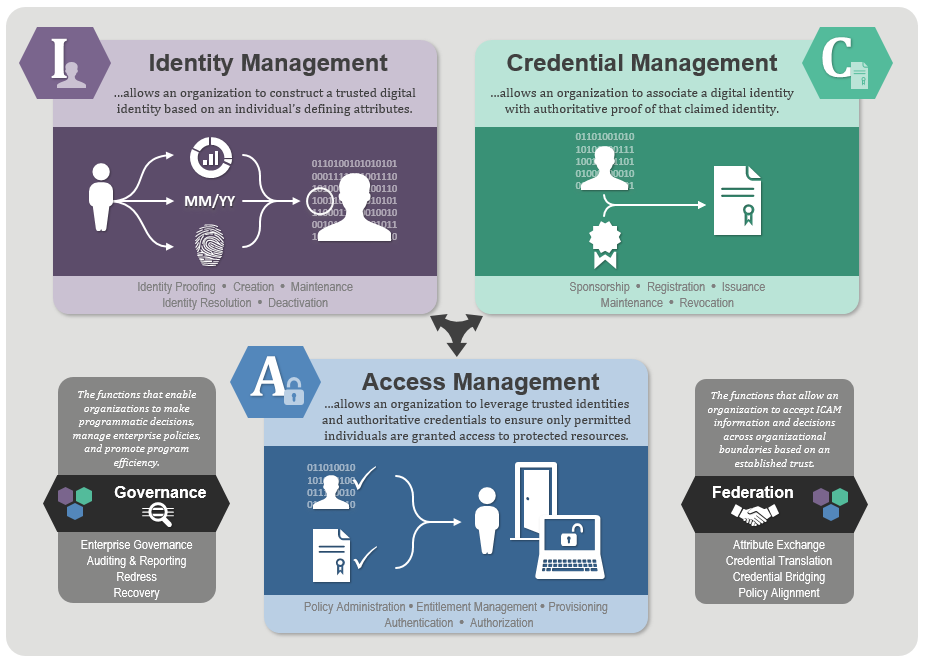
# Identity and Access Management

Identity and Access Management Overview  
The purpose of Identity and Access Management within any organization is to properly construct a trusted digital identity of an individual (whether it is an employee or an external customer) based on individual’s defining attributes such as online credentials, physical finger print, eye retina scan, or personally identifiable information such as birth day, social security number or mother’s maiden name.

The diagram below shows how Identity and Access Management fits within the overall digital security architecture. As noted in the diagram, the functions of Identity and Access Management include:

* Identity Management - provides a process to vet users (identity proofing) and manage the lifecycle of digital identities providing VA applications with a level of assurance that users are who they claim to be.
* Credential Management - provides proof of identity for users that enable them to be uniquely distinguished by VA applications.
* Access Management - provides capabilities that support authentication of users and authorization of user access enabling VA applications to safeguard information resources



Within an API Management Platform, Identity and Access Management platform(s) acts as an “External Identity Provider” used for Identity Management, Credential Management and Access Management. An API Platform does not act itself as an Identity and Access Management platform unless it has built-in IAM components. Typically, however, the IAM function is outsourced to a dedicated Identity and Access Management platform, hence it is treated as an “External” Identity Provider for the API Management Platform.

Examples of the best-of-breed Identity Management platforms include Oracle Access Manager, Ping Identity, Okta, Microsoft Active Directory, CA and IBM

When reviewing the capabilities of any API Management Platform, it is important to consider the ability a given API gatewat has to externalize IAM functionality and what standards are supported for that externalization. For example, goes the given API gateway support authentication via an external identity provider using SAML.

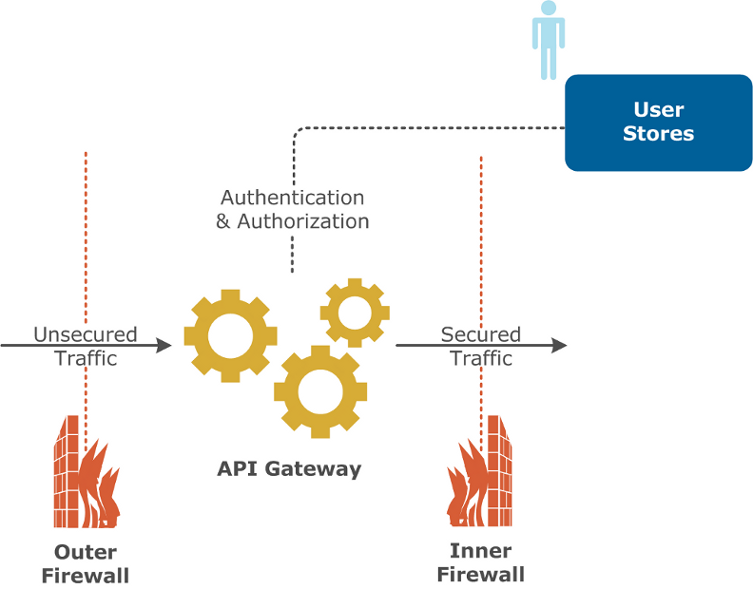
API Gateway Integration with Identity and Access Management

An enterprise API Gateway should be able to support a wide variety of identity and access management user stores including LDAP, Active Directory where the Identity and Access Management platform would provision and maintain end user identities. The Gateway should also be able to integrate with a number of Access Control products such as CA SiteMinder and Oracle Access Manager using industry standards (e.g. SAML, OAuth). Identity and Access Management user stores contain some of the most valuable information in an organization.

The diagram below shows how a unsecure traffic hits the API Gateway on inbound API requests, to be protected and secured by the API Gateway and Identity and Access Management User Stores integration. An API Gateway will receive all unsecure traffic requests for a given API, validate the identity of the requestor, and pass the authorized and secured traffic downstream to the APIs. The process of validating and authenticating an identity of the API requestor requires an API Gateway to interact with the Identity and Access Management platform to authenticate and authorize the identity or to validate an existing secure session or token.

For example, if an API request already contains a SAML assertion, Open ID Connect profile, or JWT or OAuth Bearer token, then the function of the API Gateway at this point is to request the External Identity Provider to validate the token, this may include making a request to the Identity and Access Management platform for such token validation.

However, if the API requestor hasn’t yet been authenticated and authorized by the Identity and Access Management platform, the function of the API Gateway is to request the External Identity Provider to authenticate, authorize and establish a secure session for the requestor, before passing the original API request downstream to the API Management platform.

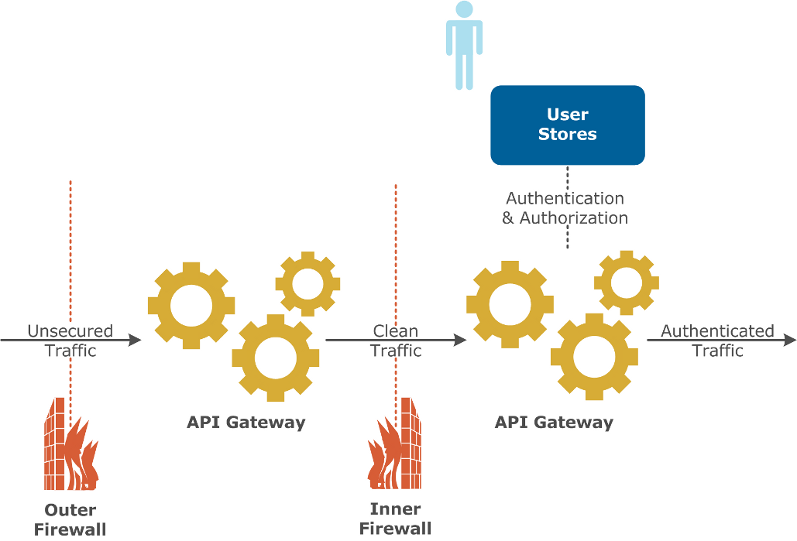


In both scenarios, an unsecure request for an API gets validated by the API Gateway before the request is passed downstream.

A more secure solution would have two API Gateway placements:

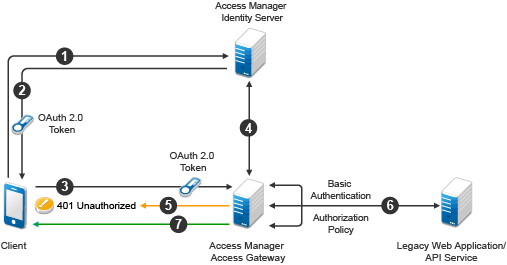
* One API Gateway in the DMZ to act as a Web Application Firewall (WAF) to protect internal resources from DDoS, JSON, XML or SQL Injection attacks
* The second API Gateway is placed inside the internal network zone protected by the Inner Firewall. The purpose of that second API Gateway is to interact with the Identity Management user stores and to request identity validation and authorization.

The diagram below depicts this two-API Gateway deployment which is a more secure and scalable solution than having only single API Gateway placed in the DMZ:



#### Identity and Access Management Flow

An API Gateway is the gatekeeper of incoming API traffic to ensure that all requests get validated with the External Identity Provider. The below diagram shows an example of how the identity of a requestor gets verified, a security token issued by the Identity and Access Management Platform, an API request with the token get passed through the API Gateway and finally gets validated and passed downstream to the actual API.



**Step 1** – An API Requestor (marked as “Client”) authenticates via Access Manager by providing end user credentials or some other identity required attributes

**Step 2** – Once Access Manager validates the identity using its connection to the User Store (e.g., LDAP store), it issues an OAuth 2.0 token in response to the Client. This can be also an Open ID Connect token, a JSON Web Token (JWT) or a SAML 2.0 assertion.

**Step 3** – The Client then makes an API request passing the security token or SAML assertion in the request

**Step 4** – The API Gateway (or Access Gateway) requests the Identity Management Access Manager to validate this token (if necessary)

**Step 5** – If the token or assertion is expired or invalid, an API Gateway throws back an HTTP 401 Unauthorized response back to the Client

**Step 6** – If, however, the token or assertion is validated successfully, the API Gateway passes the request (which is now considered a secured request) downstream to the API Service

As a conclusion, it is important to consider API Gateway capability of having a native integration with one or many Access and Identity Management products, to minimize the effort of building the above flow manually. Typically, such integration that comes out of the box with Mulesoft, Apigee, Azure, AWS, Kong, 3Scale or other API platforms include the following:

* Out-of-the-box pre-built, reusable and configurable security policies which can be easily added to one or more API
* Native connectors or adapters to the best-of-breed IAM platforms including Ping Identity, Okta, Oracle, or CA SiteMinder
* Native integration with a number of User Stores such as LDAP and Active Directory stores
* Support for open standards security protocols such as OAuth 2.0, SAML 2.0, Open ID Connect, and JWT
* Enabled security logging and monitoring
* Certification for secure management of PII, PHI, and PCI end user data