## PING ACCESS:

## Ping Access

An alternative to WAM (Web access management) systems

* To protect sites, APIs, and other resources using rules and other authentication criteria
* Already have a federated access management solution
* Enables both internal and external users to access applications securely using WAM products

## Why to use Ping Access ?

* Applications that don’t support SAML or need granula / finer access control
* Ping Access can be deployed as a reverse proxy or gateway solution for all traffic to the application. After authentication by PingFederate, Ping Access can provide session data to the application through HTTP headers, JWT tokens or proprietary token injection. This removes the need for integration kits and reconfiguring applications to enable SSO.
* Ping Access contacts PingFederate for token mediation, providing applications protected by WAM agents with the appropriate token.
* When combined with PingFederate, Ping Access enhances security by applying resource policies (URL) and transaction-specific policies with a wider range of contextual data used for authorization.
* Ping Access can validate authentication tokens by communicating with PingFederate in predetermined time intervals.
* Ping Access reduces risk by requiring applications to accept only encrypted, scope-specific session tokens. If an incorrect session token is presented, Ping Access can redirect the user to PingFederate for validation or re-authentication.

## Ping Access Overview

* Protects Web Applications and APIs by applying security policies to client requests
* Access requests are either routed through proxy gateway or target sites
* Policies applied to access requests for the target Application are evaluated, and Ping Access makes a policy-based decision to grant or deny access to the requested resource. When access is granted, client requests and server responses can be modified to provide additional identity information required by the target Application.

## Virtual Hosts

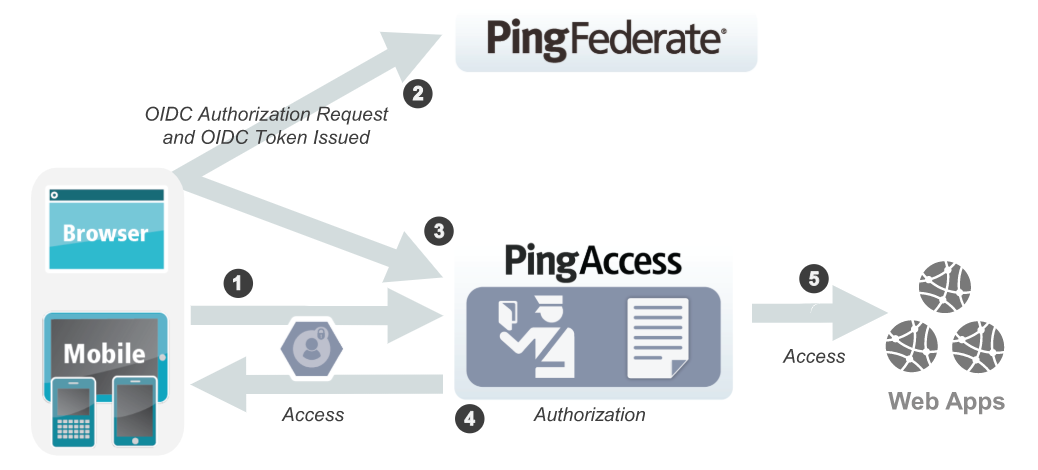
* It enables you to host multiple server or domain names. This allows one server to share resources without requiring all sites on the server to use the same host name. For example, you may want to use multiple names on the same server so that each site name reflects the services offered rather than the actual server name where those sites are hosted.

## Agents

* Agents are web server plugins that are installed on the web server hosting the target application
* Application and resource evaluation
* Applications represent Web applications or APIs to which a request is sent. They are defined by a context root and virtual server which must be unique. The context root is the first part of the URL path, starting with a slash (/) and can be arbitrarily long or deep - that is, it can contain any number of slashes. No wildcards are allowed in the context root.
* Resources represent parts of the application URL space beyond the context root that have distinct security requirements. All applications have the default Root Resource which corresponds to all URLs not handled by other resources. Any number of additional resources can be defined. Each resource can specify an arbitrary number of URL path prefixes which may contain wildcards.

## WAM session initiation

Once a user authenticates, Ping Access applies the application and resource-level policies to the request. Once policy evaluation is passed, any required token mediation between the back-end Site and the authenticated user is performed. The user is then granted access to the Site.



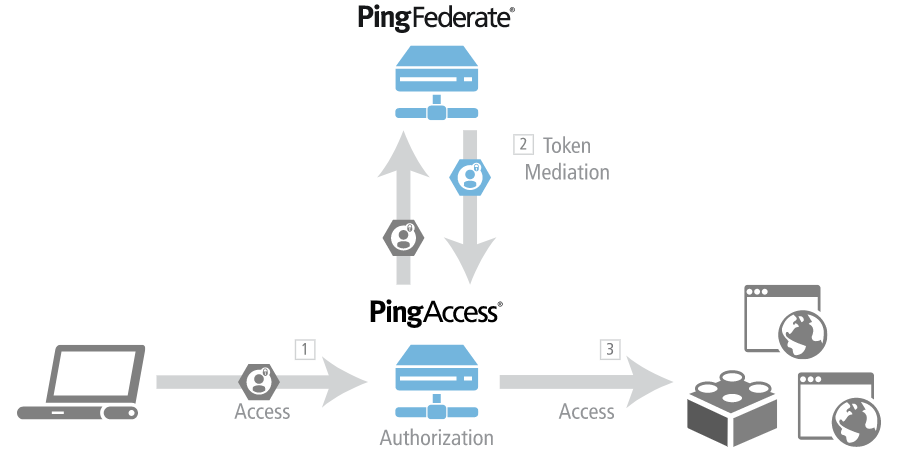
* When a user requests a Web resource from Ping Access, Ping Access inspects the request for a PA Token.
* If the PA Token is missing, Ping Access redirects the user to an OpenID Connect Provider (OP) for authentication.

Note: Oauth must be configured.

* The OP follows the appropriate authentication process, evaluates domain-level policies, and issues an OpenID Connect (OIDC) ID Token to Ping Access.
* Ping Access validates the ID Token and issues a PA Token and sends it to the browser in a cookie during a redirect to the original target resource. Upon gaining access to the resource, Ping Access evaluates application and resource-level policies and optionally audits the request.
* Info: Ping Access can perform [Token Mediation](https://documentation.pingidentity.com/pingaccess/pa40/pa_c_Token_Mediation.html) by exchanging the PA Token for the appropriate security token from the PingFederate STS or from a cache (if token mediation occurred recently).
* Ping Access forwards the request to the target site.
* Ping Access processes the response from the site to the browser (step not shown).

## Token mediation

* When planning a Ping Access deployment, it is necessary to take stock of existing applications and their authentication requirements and mechanisms. When an existing token-based authentication mechanism is in use, retrofitting that mechanism may not always be desirable or cost-effective.
* Token Mediation allows a Ping Access gateway to use a PingFederate token generator to exchange the PA Token or an OAuth Bearer Token for a security token used by the foreign authentication system. The access request is transparent to the user, allowing Ping Access to transparently manage access to systems using those foreign tokens. The request is also transparent to the protected application, which handles the access request as if it came from the user directly. Once token mediation has occurred, the token used for accessing the application is cached for future use during the session.



Processing steps:

* A user requests a Resource from Ping Access with a PA Token or OAuth Bearer Token.

Info: This example assumes the user has already obtained a PA Token or OAuth Bearer Token. See [Web Access Management](https://documentation.pingidentity.com/pingaccess/pa40/pa_c_Web_Access_Management.html) or [Using the OAuth Authorization Server](https://documentation.pingidentity.com/pingaccess/pa40/pa_c_Using_the_OAuth_Authorization_Server.html) for details on how users authenticate with PingFederate and obtain a PA Token or OAuth Bearer Token.

* Ping Access evaluates resource-level policies and performs token mediation by acquiring the appropriate security token from the PingFederate STS specified by the Site Authenticator.
* Ping Access sends the request to the Site (Web application) with the appropriate token.
* Ping Access returns the response to the client (not shown).

## Choosing Between an Agent or Gateway deployment

Ping Access can be deployed using Agents, as a Gateway (or reverse proxy), or using a combination of both. Before deciding on a deployment, it is important to understand the pros and cons of each deployment scenario and determine how they impact your strategy.

**Gateway**

Pros:

* Fewer number of deployed components that require maintenance
* Independent of target application platform
* No impact on web/app server processing and performance
* Able to work with existing security token types (such as creating 3rd party WAM tokens)

Cons:

* Requires networking changes
* Requires strategy for securing direct access to backend web/app servers (network routing or service level authentication)
* Depending on the application, may require content / request/response rewriting
* Another layer that requires HA/DR planning

Agents

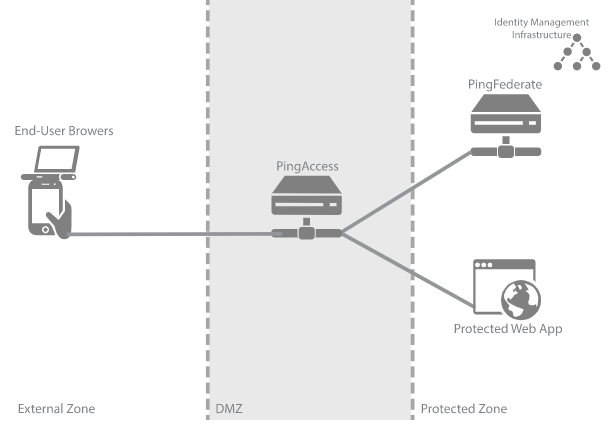
Pros:

* No networking or server level authentication changes required
* Tight integration with web server handling requests
* Scales with application

Cons:

* High cost of ownership when many agent instances deployed, although should be upgradable/patchable independently of Ping Access (policy) server
* Policy evaluation is cached; although it is periodically flushed/re-evaluated (for new sessions, updates to session token, etc.) it isn't quite is "real time" as proxy
* Tight dependency on web server version & platform

## WAM gateway



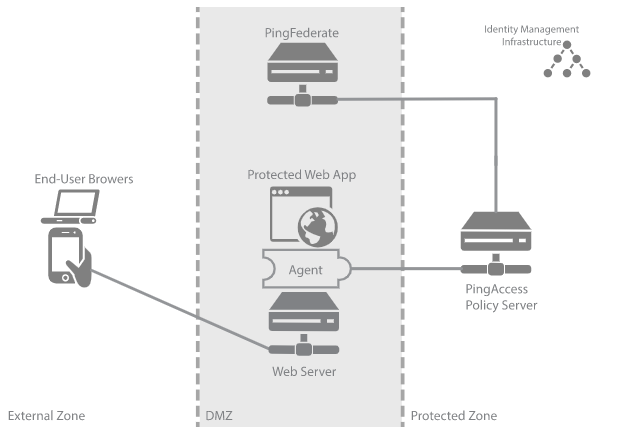
End User (abc.xyz) -> Load Balancer (abc.xyz points to PA servers IP) -> PA (abc.xyz Virtual Host) -> Processes the request -> Site (abc.applb.xyz) -> Load Balancer (abc.applb.xyz points to App servers)

End User (abc.xyz) -> Load Balancer (abc.xyz points to PA servers IP) -> PA (abc.xyz Virtual Host) -> Processes the request -> Site (server IPs) -> App server

| Zone | Description |
| --- | --- |
| External Zone | External network where incoming requests for Web applications originate. |
| DMZ | Externally exposing segment where Ping Access is accessible to Web browsers. Ping Access is a standalone instance in this environment, serving as both a runtime and an administrative port. |
| Protected Zone | Back-end controlled zone in which Sites hosting the protected Web applications are located. All requests to these Web applications must be designed to pass through Ping Access. PingFederate is accessible to Web browsers in this zone and is a standalone instance in this environment, serving as both a runtime and an administrative port. PingFederate requires access to identity management infrastructure in order to authenticate users (depicted by the icon in the diagram). |

For production- load balancer sits in DMZ as well as in Protected Zone

## WAM Agent



| Zone | Description |
| --- | --- |
| External Zone | External network where incoming requests for Web applications originate. |
| DMZ | Externally exposed segment where application Web server is accessible to Web clients. Ping Access Agent is deployed as a plugin on this Web server. The agent interacts with Ping Access Policy Server in the Protected Zone. PingFederate is deployed as a standalone instance in this environment because during user authentication clients interact with PingFederate. PingFederate requires access to Identity Management Infrastructure in order to authenticate users. |
| Protected Zone | Back-end controlled zone with no direct access by Web clients. Ping Access Policy Server is deployed in this zone. Ping Access interacts with PingFederate in the DMZ Zone. Identity Management Infrastructure is deployed in this zone. |

In Production, load balancer sits in DMZ as well as in Protected Zone