WAM

Modernization and Migration Guide

okta

Safe Harbor Statement

This presentation contains “forward-looking statements” within the meaning of the “safe harbor” provisions of the Private Securities Litigation Reform Act of 1995, which include, but are not limited to, statements regarding our financial outlook, product development, business strategy and plans, and market trends, opportunities and positioning. These forward-looking statements are based on current expectations, estimates, forecasts and projections. Words such as “expect,” “anticipate,” “should,” “believe,” “hope,” “target,” “project,” “goals,” “estimate,” “potential,” “predict,” “may,” “will,” “might,” “could,” “intend,” “shall” and variations of these terms or the negative of these terms and similar expressions are intended to identify these forward-looking statements, although not all forward-looking statements contain these identifying words. Forward-looking statements are subject to a number of risks and uncertainties, many of which involve factors or circumstances that are beyond Okta’s control.

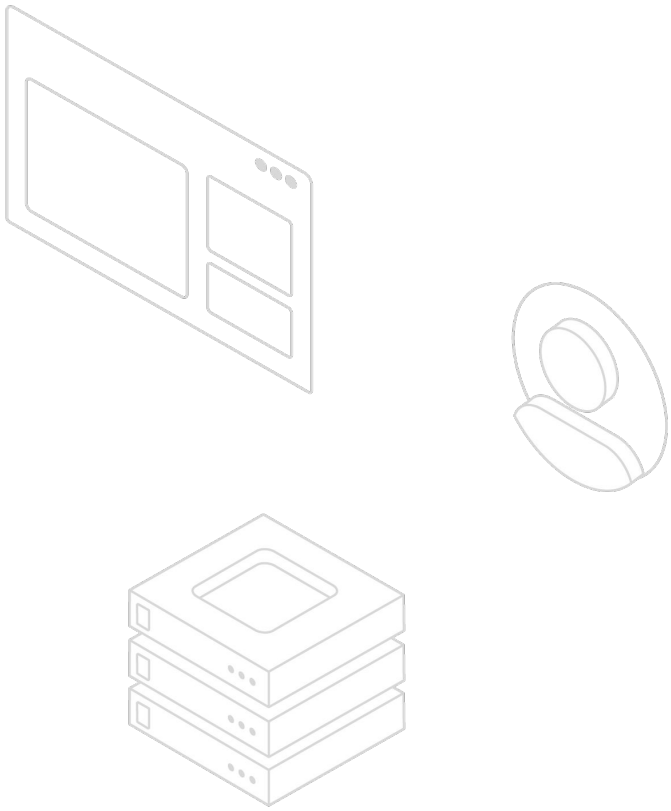
In particular, the following factors, among others, could cause results to differ materially from those expressed or implied by such forward-looking statements: the market for our products may develop more slowly than expected or than it has in the past; quarterly and annual operating results may fluctuate more than expected; variations related to our revenue recognition may cause significant fluctuations in our results of operations and cash flows; assertions by third parties that we violate their intellectual property rights could substantially harm our business; a network or data security incident that allows unauthorized access to our network or data or our customers’ data could harm our reputation, create additional liability and adversely impact our financial results; the risk of interruptions or performance problems, including a service outage, associated with our technology; intense competition in our market; weakened global economic conditions may adversely affect our industry; the risk of losing key employees; changes in foreign exchange rates; general political or destabilizing events, including war, conflict or acts of terrorism; our ability to successfully identify and integrate acquisitions, strategic investments, partnerships or alliances; our ability to pay off our senior convertible notes when due; and other risks and uncertainties. Past performance is not necessarily indicative of future results. Further information on potential factors that could affect Okta’s financial results is included in its Annual Report on Form 10-K for the year ended January 31, 2019 and other filings with the Securities and Exchange Commission that are posted on investor.okta.com.

Any unreleased products, features or functionality referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Product roadmaps do not represent a commitment, obligation or promise to deliver any product, feature or functionality. Customers who purchase our products should make their purchase decisions based upon features that are currently generally available.

The forward-looking statements contained in this presentation represent the Company’s estimates and assumptions only as of the date of this presentation. Okta assumes no obligation and does not intend to update these forward-looking statements whether as a result of new information, future events or otherwise.

This presentation contains estimates and other statistical data that we obtained from industry publications and reports generated by third parties. These data involve a number of assumptions and limitations, and you are cautioned not to give undue weight to such estimates. Okta has not independently verified the statistical and other industry data generated by independent parties and contained in this presentation and, accordingly, Okta cannot guarantee their accuracy or completeness. Expectations, estimates, forecasts and projections are subject to a high degree of uncertainty and risk. Many factors, including those that are beyond Okta’s control, could cause results or outcomes to differ materially from those expressed in the estimates made by the independent parties and by Okta.

Index



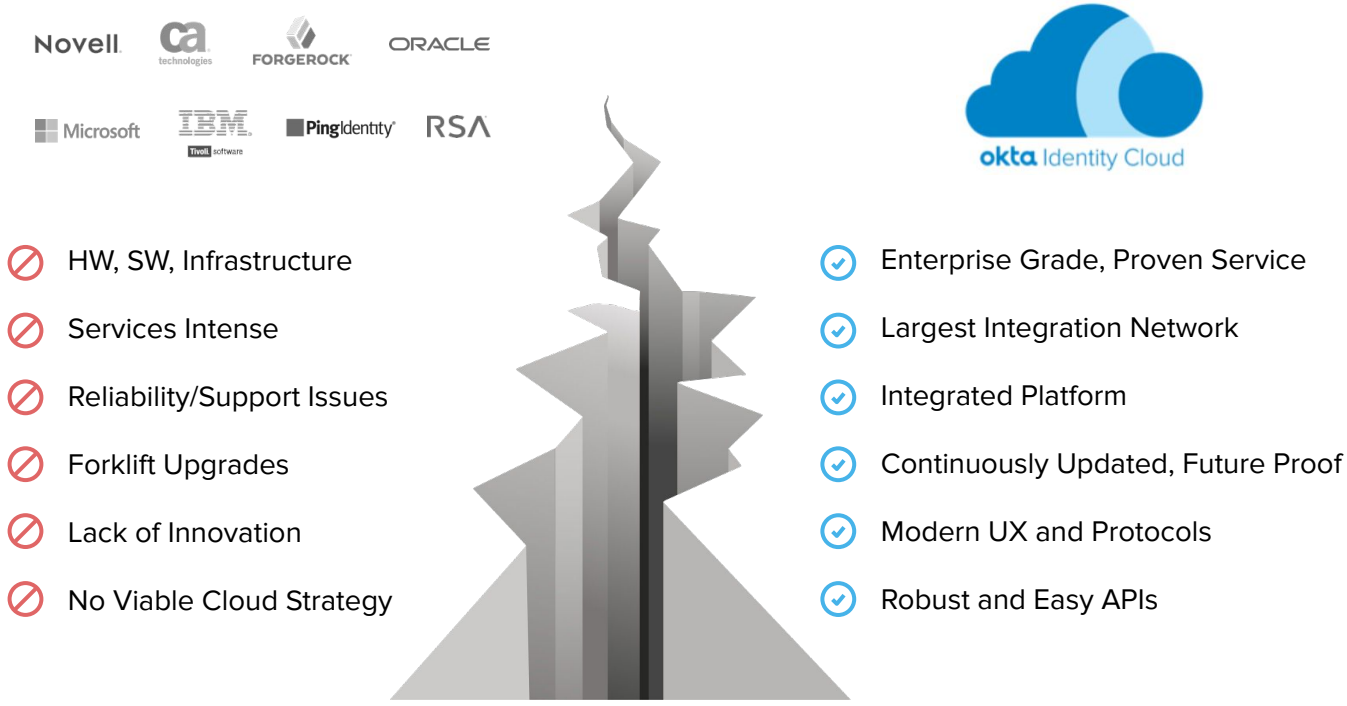
Introduction

Securely connecting users to applications is not a new problem, and to address this challenge, organizations traditionally adopted Web Access Management (WAM) solutions – CA SiteMinder, Oracle Access Manager, IBM Tivoli Access, Ping Access, ForgeRock AM, Microsoft Active Directory Federation Services (AD FS), and NetIQ/Novell Access Manager – to secure login and reduce friction for users when accessing web applications on-premises.

As technology evolved, companies adopted new services to secure access regardless of context, network, and location. From SaaS apps to mobile access and Infrastructure as a Service, enabling access to all systems and users broke the WAM security model, which depended on private networking to operate, could not be regularly updated to support new systems, and could not deliver cost-effective security.

This change in paradigm shook the WAM industry. Traditional WAM vendors deprecated their solutions by eliminating investments in new features and focused primarily on support, driving their customers to a tipping point.

Identity at a Tipping Point



> *"By 2022, IDaaS will be the chosen delivery model for more than 80% of access management deployments globally"*
>
> Gartner
>
> *Gartner's Magic Quadrant for Access Management, Worldwide 2018*

> *"The biggest benefit of using IDaaS compared to on-premises IAM solutions is a 30% to 35% lower ongoing maintenance rate"*
>
> FORRESTER
>
> *Forrester Wave: Identity-As-A-Service, 2017*

Okta Identity Cloud delivers modern identity as a service solution that addresses the requirements of today and tomorrow. Okta securely connects users to any web application, regardless of location and architecture: from cloud and mobile apps, to the enterprise apps traditionally protected by WAM solutions. Okta can be used both as a single solution for all your apps or in co-existence with your WAM solution.

This document describes the phases and steps for:

- Modernizing your identity stack, using Okta together with your WAM solution.
- Migrating your identity stack from a legacy WAM solution to Okta.

Most modernization and migration projects from WAM to Okta incorporate the best practices listed in this document. For guidance and a plan on modernizing and migrating from WAM to Okta tailored to your company, reach out to our team.

The big picture: drivers for Identity Modernization

Business Drivers	Identity Modernization	Business Requirements
Digital Transformation		Secure Omnichannel Experiences
Build 100% Cloud, Mobile IT		Lower Cost of IT Operations
Risk in Legacy Identity Providers		De-risking and Building Future Foundation
Higher User Expectations	**Security**	Modern Frictionless UX
Competitive Differentiation Into New Markets	**Agility** **Neutrality**	Time to Value and Rapid Adoption
New Security Exposure (Apps, Cloud, Speed)		Adaptive Digital Security Capability
Mergers, Acquisitions & Divestitures		Agility to Integrate Quickly, Securely, and Without Disruptions
High Cost and Complexity (Identity Silos)		Common Unified Identity Platform

Web Access Management: Recap and Limitations

> **Tip**: If you're already familiar with Legacy IdM and its limitations, skip to the next chapter.

This chapter describes how companies adopted Web Access Management (WAM) and user directories (LDAP) and why these solutions do not address the integration and security requirements of today.

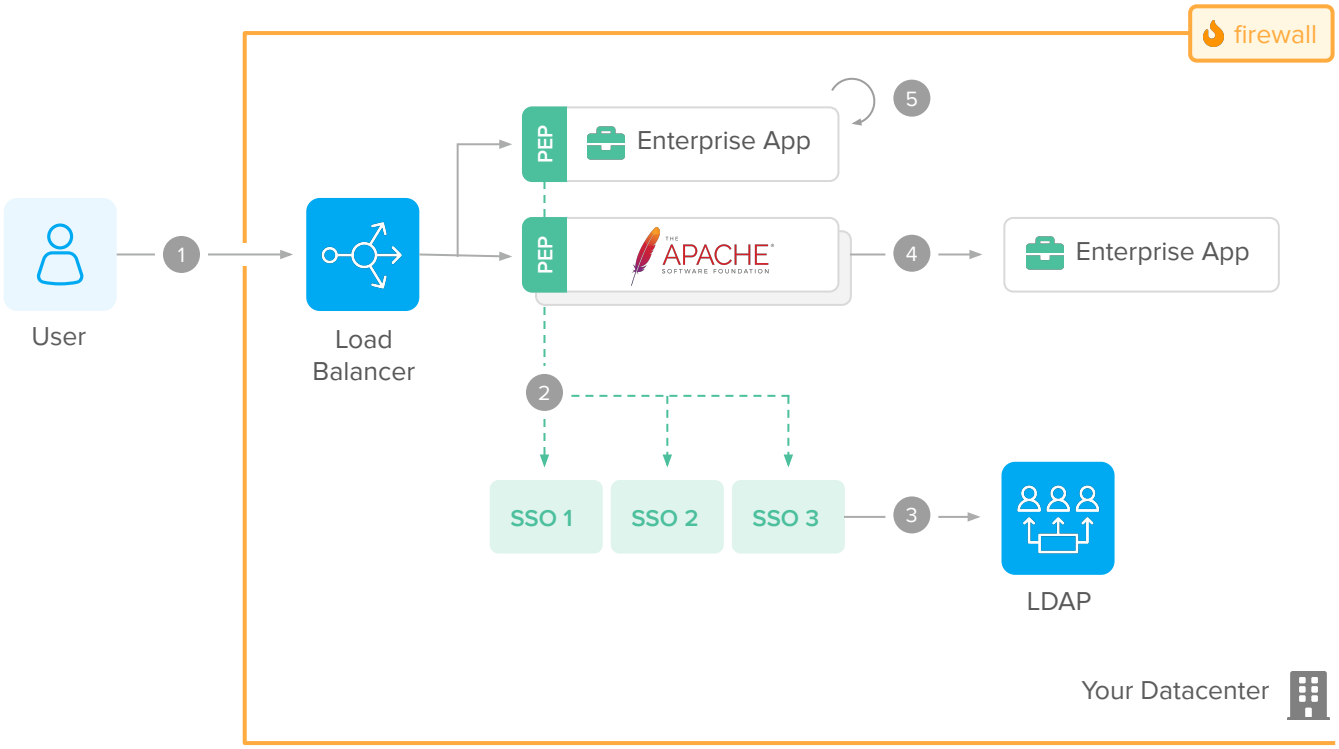
Legacy Identity Managers (IdM) - A Quick Recap

Securely authenticating and providing access to users is not a new problem. To solve this challenge, organizations adopted Web Access Management (WAM) solutions that include:

- **Legacy SSO servers** such as CA SSO (formerly known as CA SiteMinder), Oracle Access Manager (OAM), IBM Tivoli Access, Ping Access, ForgeRock Access Manager, Microsoft Active Directory Federation Services (AD FS), and NetIQ Access Manager (formerly known as Novell Access Manager).
- **Legacy LDAP directories** such as Oracle Directories (OID, OUD, and OVD), Sun Directory Server (Sun DS or ODSEE), Tivoli Directory Manager, OpenLDAP, and Novell eDirectory.

The Web Access Management (WAM) architecture relies on multiple server components and robust network tooling that includes firewalls, load balancers, and network segmentation – DMZ, Intranet, and data network zones – to secure access.

The WAM architecture work as follows:



Web Access Management: Conceptual Architecture (simplified)

Flow:

1. User accesses a web application. A Load Balancer routes the request to the proper server.
2. A plugin or agent integrated to the app or HTTP server (also known as Policy Enforcement Point – PEP) intercepts the request and validates the access with the SSO Server (also known as Policy Decision Point, or PDP) using a proprietary protocol (dotted green lines).
3. The SSO Server validates the user session and the resource (URL) requested. During authentication, it also validates the user credentials against an LDAP server.
4. Upon the SSO Server approval, the Agent complements the request context with information about the logged user – i.e., the user login – for the enterprise application. Agents embedded within the application pass the user information in session while agents used with HTTP servers send this information through HTTP Headers.
5. The enterprise application extracts the user information from the HTTP headers or the session, process the request, and returns a response to the end user.

Components

- **Load Balancer**
  Balances traffic between Applications and HTTP Servers.
  Examples: Big-IP F5, Apache with mod_proxy and mod_proxy_balancer, and Cisco ACE.

- **Policy Enforcement Point (PEP)**
  Agent or plugin that implements authentication on enterprise applications. Includes:

  - **Application/ERP/Custom Agents**
    These PEPs are installed directly in Application Servers, ERPs/CRMs, and as proprietary SDKs for direct use in applications.
    Examples: Oracle OAM Access SDK, Oracle AccessGate for eBusiness Suite, Oracle OSSO Plugin, OAM Authenticator Provider for WebLogic, CA Application Server Agent for Tomcat, WebSphere, JBoss, and WebLogic, SiteMinder Agent SDK, SiteMinder ERP Agents for SAP, PeopleSoft, Siebel, and Tivoli Manager for WebSphere.

  - **Web Agents or Web Gateways**
    These PEPs are provided as software appliances or as plugins for HTTP Servers such as Apache and IIS.
    Examples: Oracle WebGate, IBM WebSeal, Tivoli Apache Plugin, CA SSO Apache Plugin, CA SiteMinder Gateway, Novell Access Gateway.

- **HTTP Server(s)**
  Serves HTTP requests. Examples: Apache, IIS, and NGINX.

- **SSO Server or Policy Decision Point (PDP)**
  The SSO Server validates users, URLs, and URIs to authenticate and authorize access.
  Examples: CA SiteMinder, Oracle Access Manager (OAM), Tivoli Access Manager (TAM), and Novell Access Manager (NAM).

- **User Directory Store (LDAP)**
  A directory containing user credentials for authentication.
  Examples: Oracle Directories (OID, OUD, and OVD), Sun Directory Server (Sun DS or ODSEE), Tivoli Directory Manager, OpenLDAP, and Novell eDirectory.

- **Firewalls**
  Network solutions used for segmenting traffic between WAM components. The network segmentation is required to avoid unauthorized access to enterprise apps and to avoid traffic sniffing or spoofing.

- **Enterprise Applications**
  Applications protected by the IdM solution. Enterprise apps receive requests with the user information as session arguments or via HTTP headers.

WAM Architecture Limitations

WAMs are primarily built to provide password-based SSO to applications deployed in private networks (Intranets). Due to its architecture and the fact that you cannot have public SaaS apps behind firewalls, WAMs cannot cost-effectively support the IT and security requirements of today which include:

- Protect access to applications regardless of type and deployment locality.
- Protect all users – employees, contractors, partners, and customers, with a single solution (no IdM fragmentation)
- Support mobile apps and mobile access.
- Provide step-by-step instructions for integrating with popular SaaS services.
- Import and provision user accounts to SaaS applications.
- Support new types of MFA factors such as FIDO keys, push notifications, and Windows Hello.
- Implement behavioral access controls based geolocation, access from the darknet, and IP reputation.
- Update systems in real-time without service disruption.

Addressing these requirements in typical WAM architectures can be cost-prohibitive, requiring additional servers such as RSA and Oracle Adaptive Access Manager and integration with SMS gateways, or not possible – WAMs do not provide app catalogs and real-time updates.

Due to its limitations, traditional vendors have deprecated their WAM solutions by eliminating their investment in development and focusing primarily on support. Signs of retirement include, but are not limited to:

- Lack of integration wizards with applications such as Amazon AWS, Salesforce, Slack, and Office 365.
- Lack of automatic upgrades and security patches over the wire.
- Lack of support of modern federation protocols such as OpenID Connect with PKCE, OAuth 2.0, and SCIM.
- Lack of support of emerging security features such as MFA with FIDO 2, WebAuthn, and push notifications, and identifying Tor exit nodes (Darknet) and IP geo locations.

Okta Identity Platform: Overview

This chapter provides an introduction about the Okta Identity Platform, how it addresses the integration and security requirements of today, and how it delivers security to cloud, mobile, and enterprise applications from a single pane of glass.

Introduction and Benefits

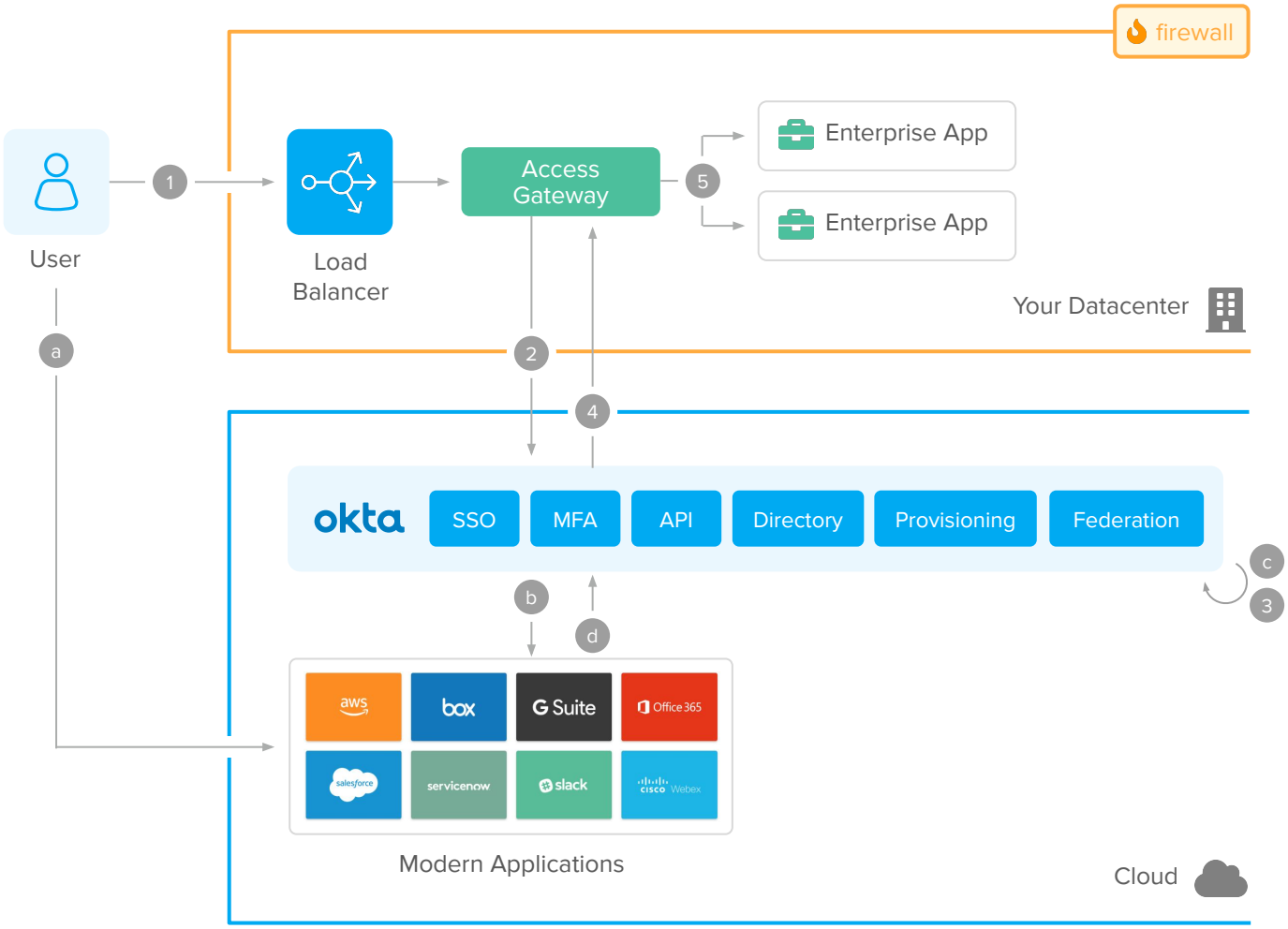
The Okta Identity Platform is an Identity as a Service (IDaaS) platform that provides Authentication, Adaptive MFA, Single Sign-On, Directory, Account Provisioning, and API Authorization. The Platform is:

- Provided as a cloud service.
- Globally available, 100% multitenant, stateless, and redundant.
- Regularly updated with security enhancements and new features.
- Built with a zero planned downtime architecture: the service is updated live, without scheduled downtime.

The Okta Identity Platform is capable of supporting both modern and enterprise applications:

- For integrating with modern applications, Okta provides the Okta Integration Network, a catalog of 6,000+ out-of-the-box integrations and extensive support for open patterns that securely connects your users to any technology while avoiding vendor lock-in.
- For integrating with your enterprise apps, Okta provides the functionality of multiple legacy IdM solutions combined – i.e., WAM, LDAP, RADIUS, SSO, MFA, and Provisioning – while mitigating infrastructure dependencies.

The Okta Identity Platform architecture works as follows:



Identity as a Service: Conceptual Infrastructure for Modern and Enterprise Applications

Modern (Cloud and Mobile) applications flow

A. User accesses a cloud or mobile application.

B. The app validates the session. If there's no session, it redirects the user for a federated authentication in Okta.

C. Okta authenticates the user using the native identity functions: Single Sign-On, User Directory Store, Adaptive MFA, and Federation.

D. Upon access approval, the user is redirected back to the Cloud of Mobile App. The app completes the federation process, establishes the user session, and responds to the user request.

Enterprise Applications flow

1. User accesses web application via HTTPS.
2. The Access Gateway intercepts the user request. If the session is invalid, it redirects the user for a federated authentication on Okta.
3. Okta authenticates the user. All the identity functions (Single Sign-On, User Directory Store, Adaptive MFA, and Federation) are built natively into the Identity Platform.
4. Upon access approval, the user is redirected back to the Access Gateway that completes the federation. Because the gateway is capable of working with different kinds of applications, it doesn't require native integration.
5. The enterprise app captures the user information, process, and responds to the request.

Components

- **Okta**
  Okta provides Single Sign-On, Adaptive Multi-Factor Authentication, API Authorization, User Directory, Account Provisioning, and Federation in the cloud for modern apps (via native standards-based integrations) and enterprise apps (via Access Gateway).
- **Cloud and Mobile Applications**
  Modern applications with direct access to the internet.
- **Access Gateway**
  The Okta Access Gateway bridges the gap between the Identity Platform and your enterprise applications. It leverages federated authentication for Single Sign-On and Adaptive MFA and connects to your enterprise application using WAM standards (such as security headers). Because the gateway supports different types of applications, it doesn't require native integration.
- **Firewalls**
  Network solutions used for segmenting traffic between components. The network segmentation is still required to avoid unauthorized access to enterprise apps and to avoid traffic sniffing or spoofing.
- **Enterprise applications**
  Applications protected by the gateway. Enterprise applications receive traffic via HTTP Server with the user information on the HTTP headers.

Okta Capabilities Beyond WAM

Okta delivers features beyond typical WAM solutions. You can take advantage of these features to deliver identity and access management in new use-cases, improve your security posture, and return of investment.

This section list projects you can address beyond the WAM capabilities:

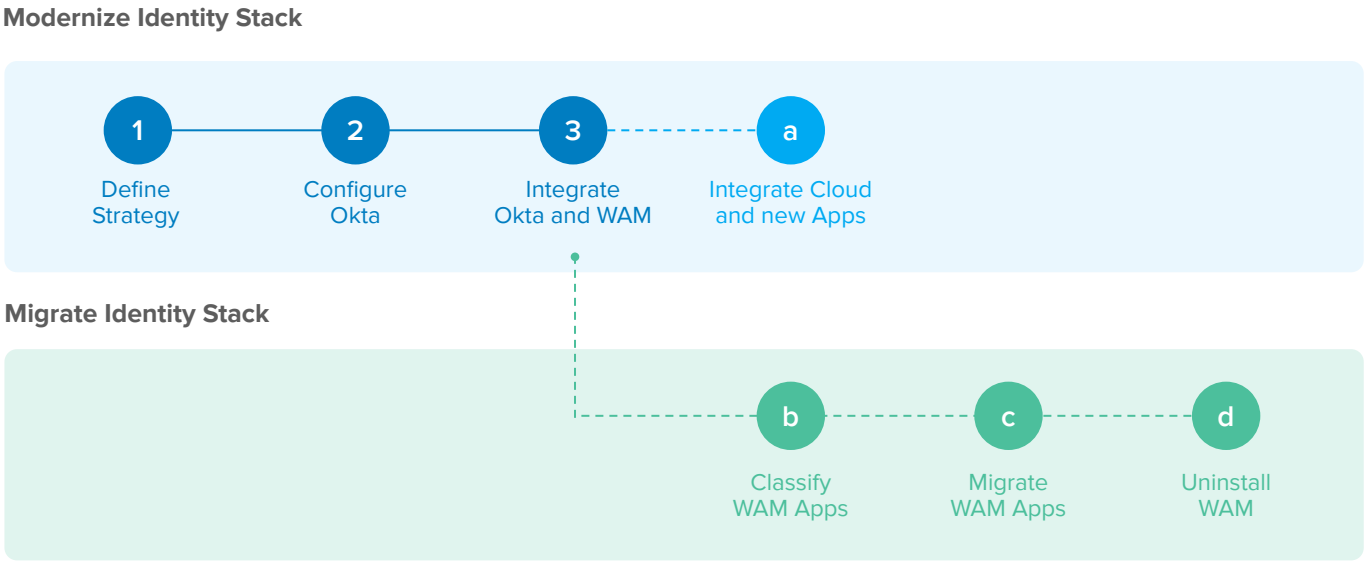
- **Accelerate Office 365 adoption** and avoid PowerShell complexity with Okta SSO, MFA, and Lifecycle Management.
- **Automate user onboarding & offboarding** to applications and AD/LDAP directories with Okta Lifecycle Management and HR integrations.
- **Consolidate AD domains and reduce AD footprint** with Okta Unified Directory.
- **Meet compliance and secure access to VPNs, Virtual Desktops, and Network Applications** with Okta MFA.
- **Reduce password-reset costs** with Okta Self-Service Password Reset for cloud applications, AD, and LDAP.
- **Secure Access to custom APIs**, with API Access Management and Okta's OAuth and OpenID Connect APIs and SDKs.
- **Secure access to cloud and on-premise servers**, by implementing Okta provisioning, SSO and MFA workflows to Linux and Windows servers over SSH and RDP.
- **Provide access and SSO to all users** – contractors, partners, and customers – from a single identity solution.

To learn more about the initiatives and projects you can accomplish with Okta, reach out to our team.

Modernization and Migration: Strategy and Phases

As a modern identity solution, Okta can operate in co-existence with your WAM solution or as a single solution for all apps.

Okta is implemented through the following phases:



Okta Implementation Phases across different scenarios.

The implementation phases are:

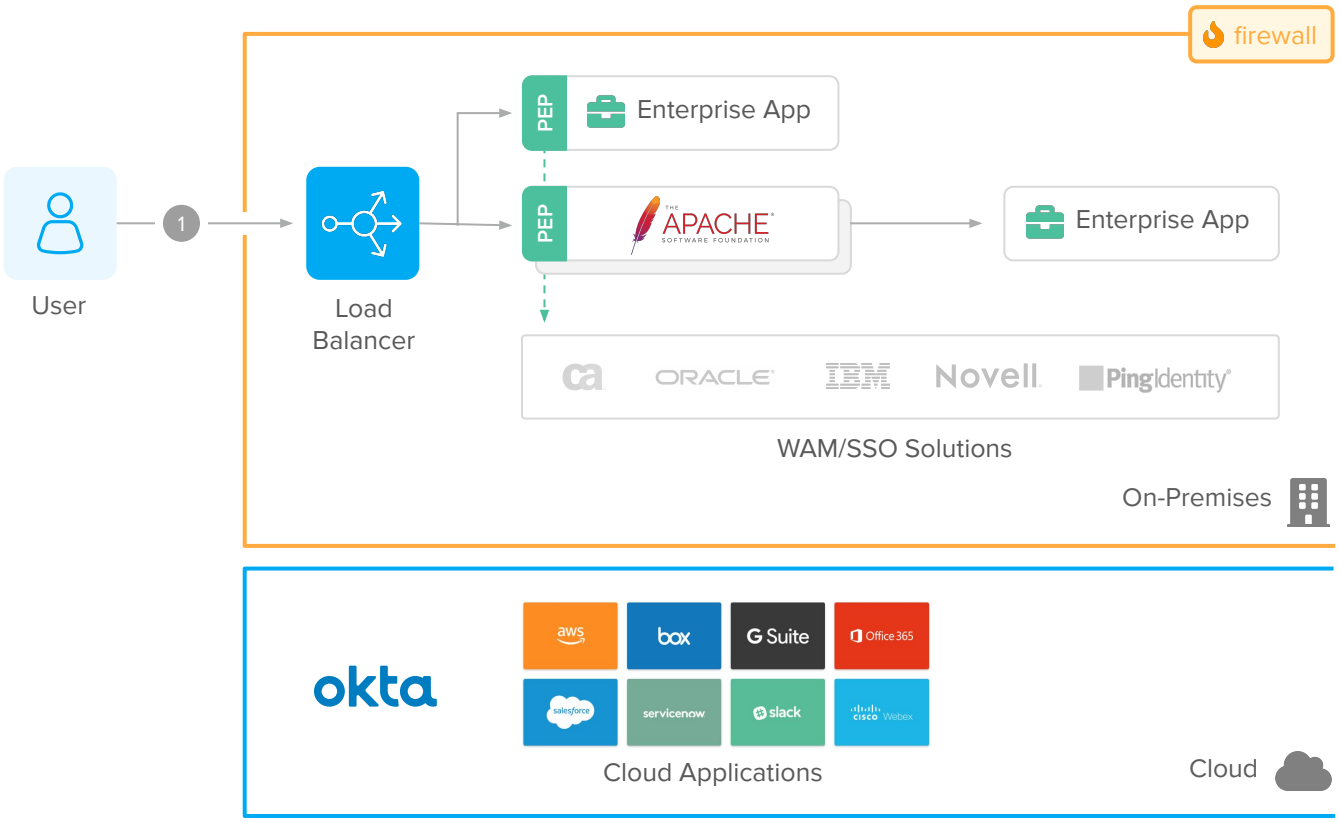
- **Modernize identity stack**
  Define how you use Okta, configure the Okta service, and integrate Okta with your current WAM solution as an Identity Provider. After this stage, all new applications and cloud (SaaS, PaaS, and IaaS) apps are integrated into Okta, deprecating the WAM use for new integrations. Also, Okta provides universal SSO and MFA for all applications, including your WAM solution. Your WAM trusts the Okta authentication to grant access to the WAM apps. Due to Okta's availability as a SaaS service and integration wizards, this phase is executed at a fast pace.
- **Migrate your identity stack**
  In this phase, you migrate your identity stack from WAM to Okta in 3 steps: 1) identify and classify your WAM applications, 2) migrate these apps to Okta, and 3) uninstall the legacy WAM service.

The Okta implementation phases are incremental. You can start with modernizing the identity stack and migrate your identity stack at your own pace. Within each scenario, you improve your security posture and user experience while reducing your footprint and improving your Return of Investment.

Phase 1: Modernize Identity Stack

In this phase, you configure Okta for initial use and integrate Okta with your current WAM solution – typically as a SAML Identity Provider. Steps include:

1. Define the initial strategy for using Okta
2. Configure the Okta service
3. Integrate Okta and WAM
4. Integrate new applications with Okta



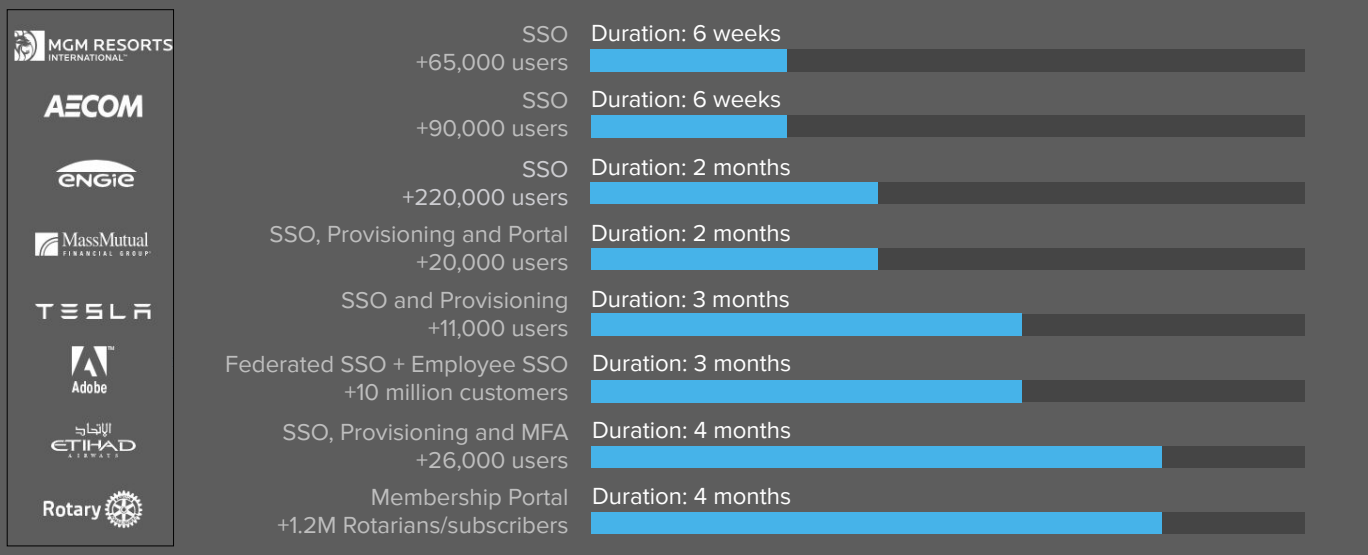
Okta and WAM integration: Conceptual Architecture

Benefits

After this stage:

- All new apps and SaaS services are integrated to Okta, deprecating WAM use for new assets.
- Okta provides universal SSO and MFA for all applications, including your WAM solution. Your WAM trusts the Okta authentication to grant access to the WAM apps.

Because Okta is provided as a cloud service, the initial phase is executed at a fast pace:



Examples of time to deploy and rollout Okta SSO, MFA, and Provisioning across different companies and industries

1. Define Initial Strategy

In this step, you list the requirements for your identity solution. Examples of requirements include:

- What users – i.e. employees, partners, customers, contractors – will use Okta?
- What systems – i.e. HR, AD, LDAP, or Okta – will act as the source of truth (or master) for each type of user?
- What are the authentication policies and account recovery requirements?
- What systems store and validate the user password?
- What MFA factors will be used?
- What applications will be integrated with Okta Single Sign-On?
- What applications will have accounts provisioned by Okta?

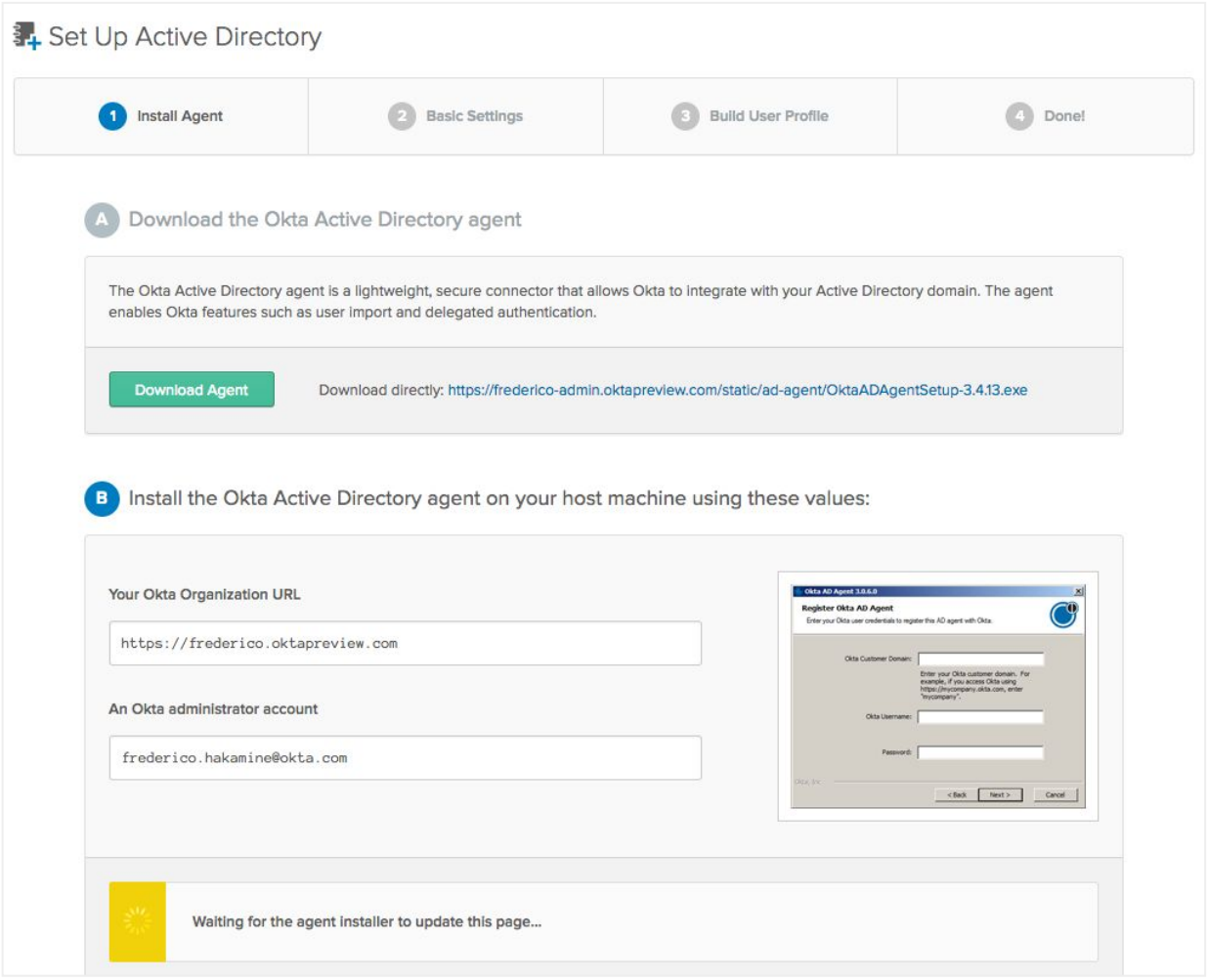
The requirements scope which services and settings are configured in your Okta tenant. Since Okta is a flexible subscription-based platform, you can change requirements as you go. Turning on additional features – such as Adaptive MFA – is a trivial task and doesn't require installations or manual integrations to existing modules, which saves money and reduces complexity as you embrace additional requirements.

2. Configure the Okta service

During the configuration step, you define your policies and settings in Okta. Deployments with a pre-existing WAM solution usually include:

- Install an LDAP or AD Agent to sync users and groups with Okta.
- Configure an initial authentication and password reset policy.
- Configure an initial policy for MFA enrollment and enforcement.

The initial Okta configuration is facilitated with default configurations and integration wizards:



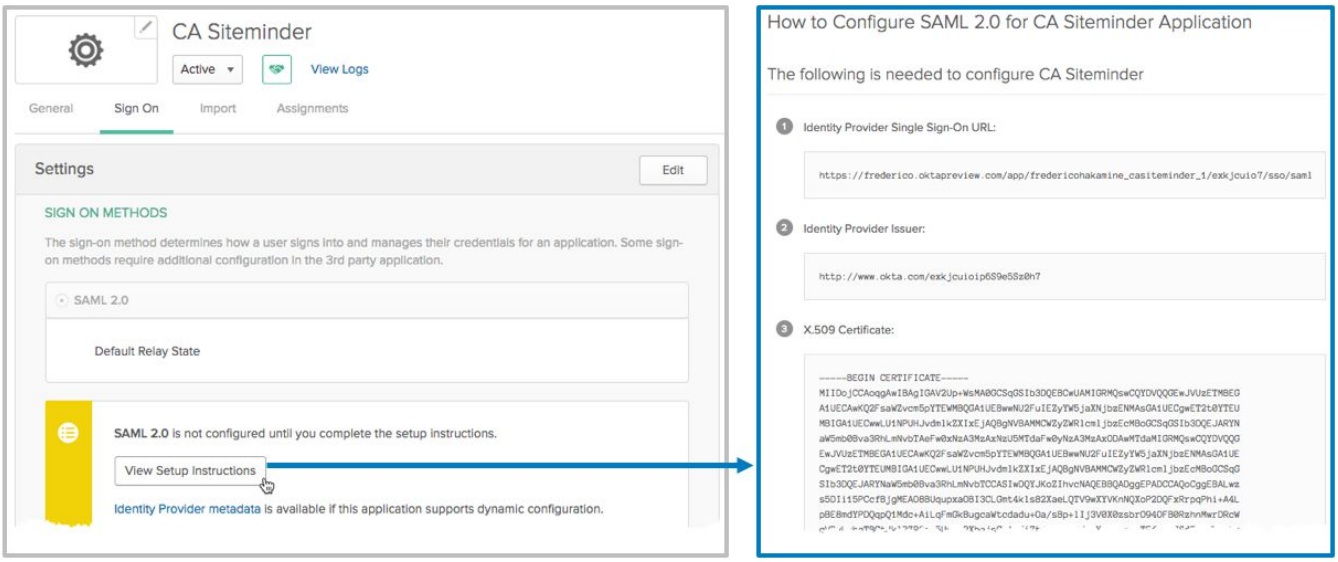
Active Directory Integration in four steps

By the end of the configuration steps, you should have users ready to access Okta with the same credentials they use to access on-premise systems.

3. Integrate Okta and your existing WAM

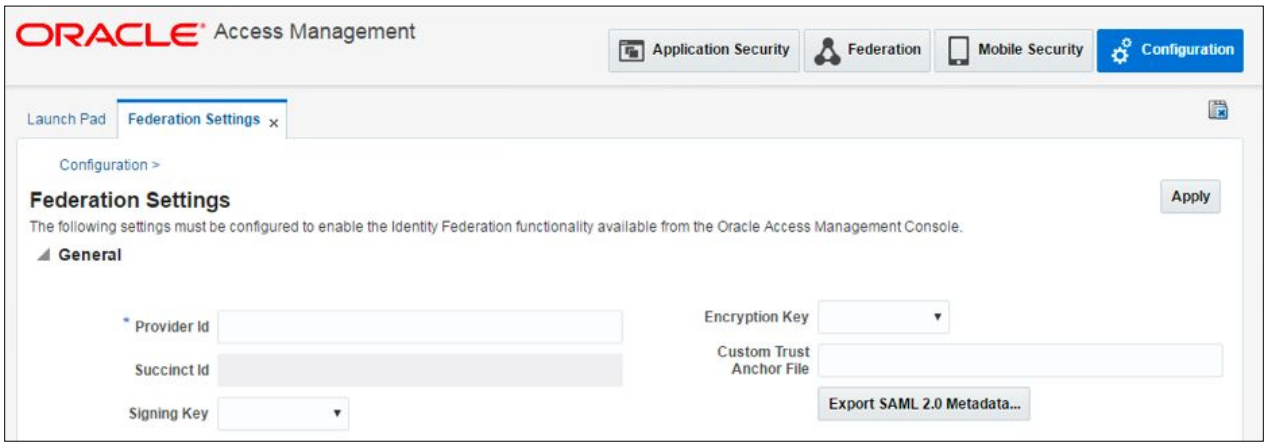
In this step, you configure Okta as a SAML Identity Provider for your current WAM/SSO solution. The configuration is performed on Okta and your current WAM solution.

For the Okta configuration, you use the Okta Application Integration Wizard (AIW). The AIW simplifies the configuration with any SAML or OpenID Connect partner (including your WAM server):



Okta Application Integration Wizard

On the WAM side, setup Okta as a SAML or OpenID Connect Identity Provider:



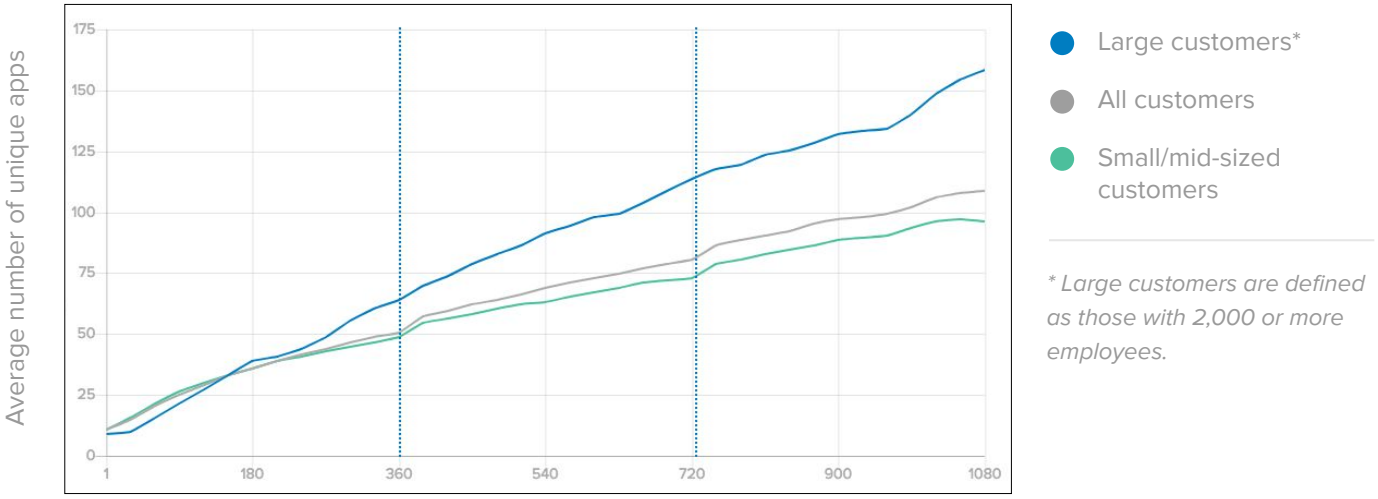
On your WAM server, configure Okta as an Identity Provider

WAM documentation references:

- **Oracle Access Manager**
  https://docs.oracle.com/cd/E27559_01/admin.1112/e27239/oif_srv_prvdr.htm#AIAAG6511
- **CA SiteMinder/Single Sign-On**
  https://docops.ca.com/ca-single-sign-on/12-52-sp1/en/configuring/legacy-federation/configure-a-saml-2-0-identity-provider
- **Novell Access Manager / NetIQ Access Manager**
  https://www.netiq.com/documentation/access-manager-44/admin/data/b1ax7qoc.html#b1jrguws
- **Tivoli Access Manager / IBM Access Manager**
  https://www.ibm.com/support/knowledgecenter/en/SSPREK_9.0.6/com.ibm.isam.doc/config/concept/con_fed_saml.html
- **Ping Federate / Ping Access**
  https://documentation.pingidentity.com/pingfederate/pf84/index.shtml#ssoIntegrationOverview/concept/serviceProviderIntegration.html
- **ForgeRock**
  https://backstage.forgerock.com/docs/am/5/saml2-guide/
- **Active Directory Federation Services (ADFS)**
  https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd727938(v=ws.10)

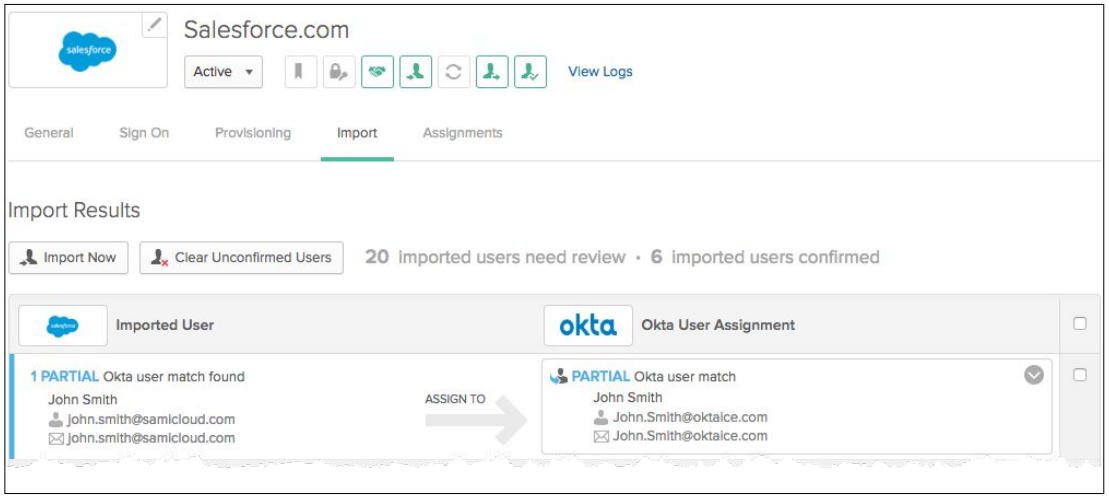
4. Integrate new applications with Okta

In the Application Integration step, you integrate Okta with existing cloud applications using the Okta Integration Network (OIN). A typical Okta deployment starts with 10 to 15 applications and scales to hundreds of applications fast:



Okta customer lifetime

Okta provides prescriptive integration guides for 6,000+ applications that support both net new and existing subscriptions. On existing apps, Okta is capable of importing and matching user records:

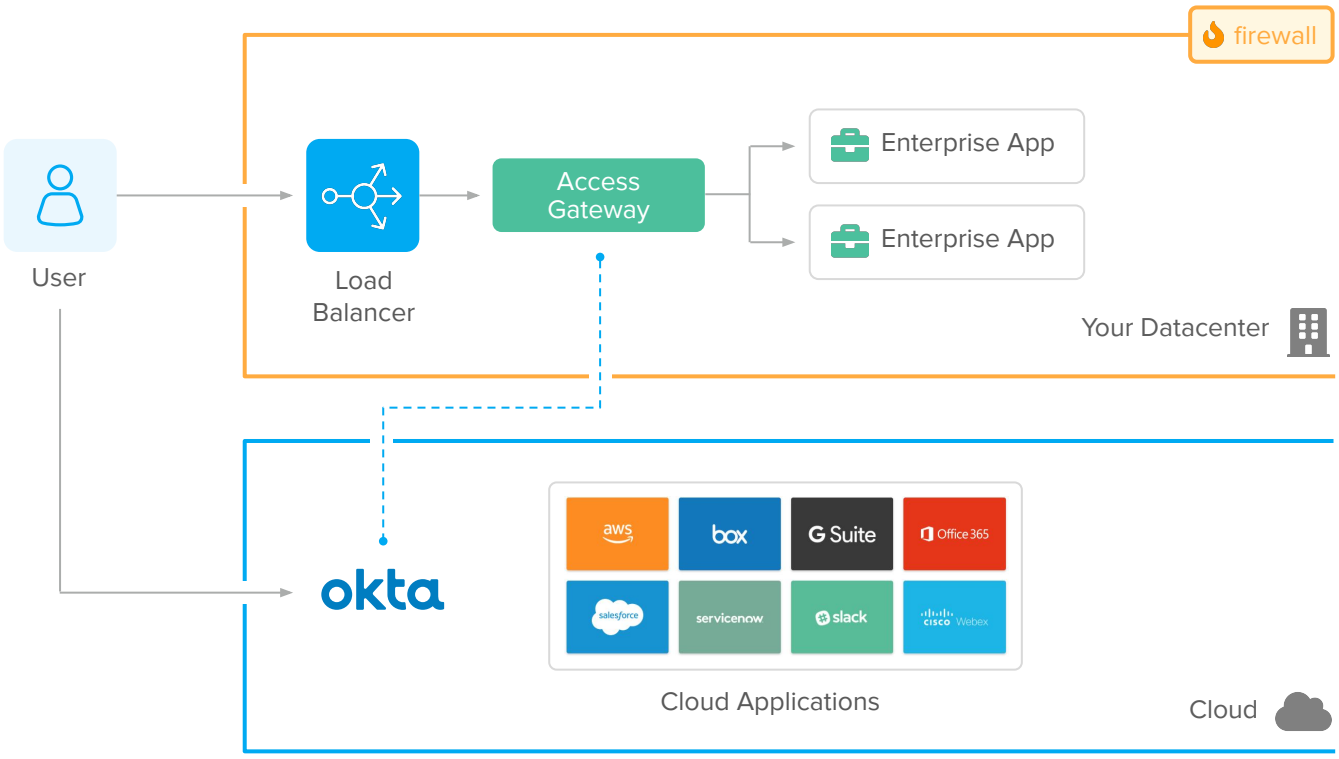


Okta Integration Network: Onboarding systems with pre-existing users

Phase 2: Migrate Identity Stack

In this phase, you migrate your identity stack from WAM to Okta in 3 steps:

1. Identify and classify your WAM applications
2. Migrate WAM applications to Okta
3. Uninstall the legacy WAM service



Conceptual Architecture after migrating WAM applications to Okta

Okta provides tailored migration options for WAM deployments. To learn more about Okta migration options and to get a migration tailored to your company, reach out to our team.

Benefits

After this stage, Okta acts as the single identity provider for all applications, improving your user experience and Return of Investments.

1. Classify your WAM applications

In this task, you take an inventory of apps currently protected by your WAM solution and then classify the applications based on the integration used. The integrations typically used on enterprise WAM deployments ranked by popularity are:

1. Header-based authentication
2. Agent-based authentication
3. Java Application Servers
4. SAML to COTS and SaaS apps
5. ERP Based (eBusiness Suite, Peoplesoft) apps

2. Migrate WAM Applications to Okta

In this step, you migrate applications from the existing WAM solution to Okta. The migration tasks include:

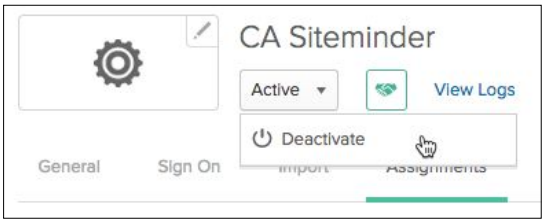
- Install the Access Gateway.
- Select in which order applications are migrated from WAM to Okta.
- Gradually migrate applications.

Ideally, your migration should start from low-risk applications and applications that share the same type of integration (e.g. header-based apps). Like most of our customers, your confidence in the migration procedure grows up to a point where you migrate applications of the same kind in bulk. The bulk migration expedites the process time while reducing costs.

3. Uninstall the legacy WAM service

After all apps are migrated from WAM to Okta, it's time to turn off the WAM service. Turning off the WAM service includes:

- Deactivate the integration with Okta (configured on 3: Integrate Okta and your existing WAM).
- Monitor access to the legacy WAM for few days to confirm the service is not being used in rogue apps (not identified in the classification process)
- Take final backups and uninstall the WAM service.



To learn more about the initiatives and projects you can accomplish with Okta, reach out to our team.

Appendix A:

Modernization/Migration FAQ

This chapter lists common questions around WAM modernization and migration to Okta.

What's the user experience during migration to Okta?

The migration from WAM to Okta doesn't impact the end-user experience significantly. During the Okta configuration, users are imported to Okta automatically – via the Okta LDAP and AD agents – and use the same credentials from the WAM/SSO system to access Okta. Depending on your deployment scenario and stage, users will see Okta as the new login page. The users' reaction to the new login page tends to be positive, mainly due to the page speed on the browser and the UI responsiveness on mobile access. Okta provides an End User Adoption toolkit that you can use for a successful launch to end-users.

Does the Access Gateway support URL-based Authorization?

Access Gateway supports the following authorization scenarios:

Authorization Complexity	Example
Public Assets (No authz)	intranet.org.com/public
App-level Authz	intranet.org.com/app1 and intranet.org.com/app2
App basic URI Authz	intranet.org.com/app1/admin and intranet.org.com/app1/home
App Deep Authz	intranet.org.com/app1/admin/x/a and intranet.org.com/app1/admin/a/t
Dynamic URI Authz	intranet.org.com/app1/{userid}/status

Can the Access Gateway replace proprietary SDK integrations such as the OAM C SDK or the CA Java SDK?

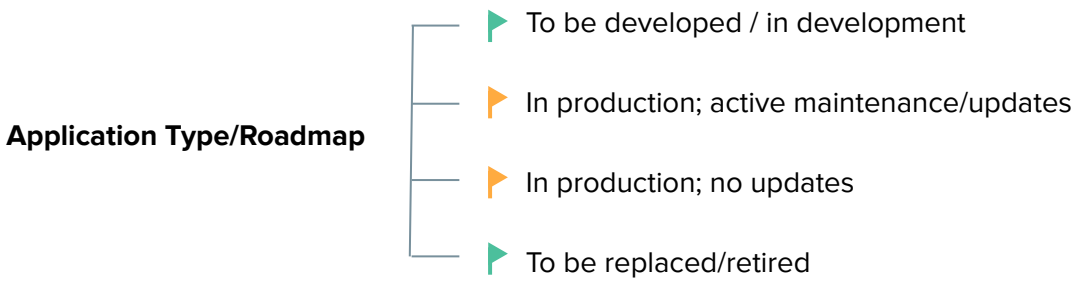
No. Okta integrates with applications via Access Gateway – HTTP request level – and via open standards – OpenID Connect, OAuth, LDAP. The use of proprietary WAM SDKs – e.g. OAM C SDK, and CA Java SDK – creates vendor lock-in with the WAM provider. Migrating these applications require changes in source code regardless of which new identity solution you adopt. The recommended action for apps with proprietary WAM SDKs is to replace the SDK code with open-standard integrations. The use of open standards allows you to adopt better solutions while avoiding vendor lock-in.

Does the Access Gateway support integrations to ERP systems?

The Access Gateway supports the following ERPs: eBusiness Suite, Peoplesoft, JD Edwards EnterpriseOne, and Agile PLM.

How to approach and application updates and future developments in environments with Okta and WAM?

Use this image as a guide:



- **Applications "To be developed / in development" should incorporate modern authentication using Okta**. This future proof the application and improve its support for API authorization and multi-cloud environments.
- **For applications "to be replaced/retired" before your legacy IdM**, consider waiting for the application retirement. For low-risk applications to be retired after the Legacy IdM uninstall, you can implement SWA.
- **For applications that are in production on the foreseeable future**, check its maintenance/update cadence. Applications with proper maintenance and constant updates usually offer better support for federated authentication.

What technical recommendations are applicable for when implementing the Access Gateway?

When implementing the Access Gateway, consider the following best practices:

- To avoid network conflicts, install the Access Gateway on the same subnet as your current Legacy IdM HTTP Server.
- To meet resiliency requirements, implement the same high-availability as your Legacy IdM HTTP Server and performance test your configuration.
- To avoid URL rewriting or re-bookmarks from users, try to keep the same application domains.
- Use your Load Balancer to gradually migrate traffic from legacy HTTP Agent/Gateway to the Access Gateway
    - Use the Load Balancer rules to direct/balance traffic between legacy WAM and Access Gateway.
    - Balancing strategies include network origin or round-robin with % distribution.
    - To adopt gradual migration via Load Balancer rules, make sure you have session stickiness/persistence. The persistence makes sure users that established a session in the Legacy HTTP Server are not routed to Access Gateway and vice-versa.
    - Document a sanity check script for testing each path (some Load Balancers allow you to determine your path through request headers) to help you confidently ramp-up the migration.
    - You can also use the Load Balancer policies to fallback traffic to the Legacy HTTP Agent in case you find problems during the Gateway adoption.

What technical recommendations are applicable for when uninstalling the legacy WAM?

Before Uninstalling the Legacy WAM, consider following the best practices:

- Consider monitoring the Legacy Solution for a period before uninstalling (so you can detect integration gaps).
- Start by cutting off traffic via Load Balancer. This allows you to easily fallback if a gap is discovered.
- Take a backup of your entire environment before uninstalling the system.

To learn more about the initiatives and projects you can accomplish with Okta, reach out to our team.

Appendix B:

Example of App migration from WAM to Okta

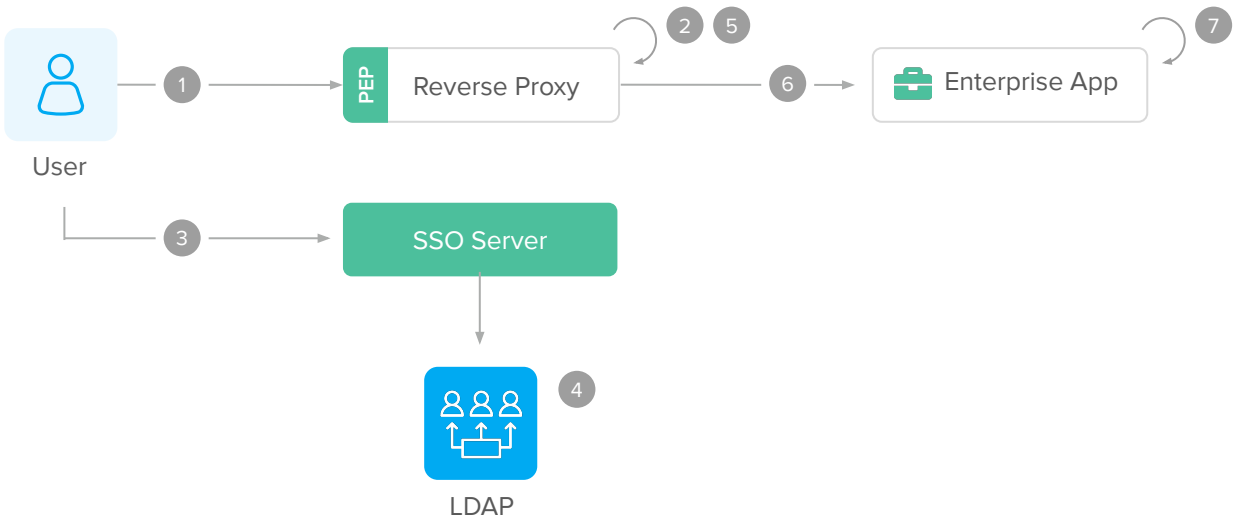
This chapter provides an example of how an application is migrated from legacy WAM to Okta.

Example: Header-based Authentication

The header-based authentication pattern is the most popular integration pattern used across different WAM vendors. This pattern wraps enterprise applications with Reverse Proxies – e.g. CA SiteMinder Proxy Service and Oracle WebGate – that validates user sessions and request authorization. Authorized requests are injected with HTTP header variables identifying the logged user, which are consumed by enterprise apps to set the user session.

Before migrating to Okta

Before Okta, the application used WAM and a reverse proxy for authentication:



Conceptual Architecture:
Header-based Authentication with traditional WAM solutions (SP initiated flow)

IdP-Initiated Flow

There is no IdP initiated flow for this pattern.

SP-Initiated Flow

The SP initiated flow is triggered when users try to access the enterprise applications served by the Reverse Proxy integrated to the WAM/SSO Server:

1. User tries to access an enterprise application URL via Reverse Proxy.
2. WAM/SSO plugin – installed in the reverse proxy – intercepts the requests and detects no SSO session.

   The user is redirected to the SSO login page for authentication.
3. User submits his/her credentials (usually username and password).
4. SSO server authenticates user credentials (usually against an LDAP or AD server).

   After login, the SSO server establishes an SSO session cookie and redirects the user back to the enterprise application URL (usually the same URL from step 1).
5. Reverse Proxy intercepts the request, validates and authorize the user session.
6. After authorization, the reverse proxy adds HTTP header variables – containing information about the logged user – to the request and allows the request to reach the enterprise application.
7. Enterprise app receives the request and reads the HTTP header variables to establish an app session.
8. Enterprise app processes the request and returns a page to the end-user.
9. The SSO session is reused on subsequent requests for authentication and authorization.

After migrating to Okta

In the migration to Okta, the reverse proxy serving enterprise apps changes from the WAM solution to the Access Gateway. The gateway acts the same way as the former reverse proxy and can deliver the headers expected by the enterprise app. Due to this, the enterprise application can operate with Okta without changes in source code.

Conceptual Architecture:
Header-based Authentication with Okta (SP initiated flow)

IdP-Initiated Flow

The IdP initiated flow is triggered when the user, from the Okta Dashboard, clicks on a shortcut to the enterprise application.

1. User clicks access an enterprise application from the Okta dashboard.
2. Okta redirects user to the enterprise application URL (protected by the Access Gateway).
3. Access Gateway receives the request and performs an initial SAML federation with Okta. This step is transparent to users already logged into Okta.
4. Access Gateway establishes a session cookie and authorizes the request URL.
5. After authorization, the Access Gateway adds HTTP header variables – containing information about the logged user –to the request and allows the request to reach the enterprise application. (The gateway uses the same header-based integration as the legacy WAM to avoid code updates in the enterprise app).
6. Enterprise app receives the request and reads the HTTP header variables to establish an app session.
7. Enterprise app processes the request and returns a page to the end-user.
8. The Access Gateway session cookie is reused for subsequent requests. The Access Gateway validates each request for SSO and authorization.

SP-Initiated Flow

The SP initiated flow is triggered when users try to access the enterprise applications served by the Reverse Proxy integrated to the WAM/SSO Server:

1. User tries to access an enterprise application URL via Access Gateway.
2. Access Gateway intercepts the requests and detects no session cookie.

   Access Gateway performs a SAML assertion with Okta.
3. If the user is not logged into Okta, a login page is displayed

   User submits his/her credentials (and optionally MFA) to Okta.
4. Okta authenticates the user credentials internally or via delegated authentication to LDAP/AD servers.
5. After the successful login, a SAML assertion is returned to the Access Gateway.

   Access Gateway establishes a session cookie and authorizes the request URL.
6. After authorization, the Access Gateway adds HTTP header variables – containing information about the logged user –to the request and allows the request to reach the enterprise application. (The gateway uses the same header-based integration as the legacy WAM to avoid code updates in the enterprise app).
7. Enterprise app receives the request and reads the HTTP header variables to establish an app session.
8. Enterprise app processes the request and returns a page to the end-user.
9. The Access Gateway session cookie is reused for subsequent requests. The Access Gateway validates each request for SSO and authorization.

To learn more about the initiatives and projects you can accomplish with Okta, reach out to our team.