okta

Managing Access to Legacy Web Applications with Okta

Okta Inc.
301 Brannan Street, Suite 300
San Francisco, CA 94107

info@okta.com
1-888-722-7871

Supporting Legacy Authentication Methods with Okta

At Okta, our customers are the most innovative, forward-leaning, and bold enterprises in their respective businesses. They look to Okta to securely connect their employees, partners, and customers to any technology and Okta is built to manage access to thousands of applications and resources, right out of the box. But no matter how innovative, organizations that have been around a while inevitably have legacy resources that rely on the technologies of the previous generation. Authentication is one area where older models persist; critical business applications use older approaches to authentication which are closed and inefficient. Enterprises need to enable seamless, secure access to every application or resource, so it must be able to support legacy and modern technologies. This whitepaper describes a best-of-breed approach to this problem, one that takes advantage of the best of Okta and the most effective legacy integration models, while simplifying the integration architecture.

How We Got Here

In the 1990s, many companies faced a problem: with the proliferation of web applications in the enterprise, end user access was difficult to manage and a poor user experience inhibited adoption. There was no ubiquitous standard for authentication that worked well for web applications, so many organizations turned to Web Access Management (WAM) solutions like CA SiteMinder, Oracle Access Manager, Tivoli Access Manager to control authentication and authorization to corporate resources. WAM tools provide single sign-on, centralized policy management and reporting and auditing capabilities for web applications.

In the late 2000s, two things happened: federated authentication standards like Security Assertion Markup Language (SAML) and later OpenIDConnect (OIDC) gained popularity, and SaaS, PaaS and IaaS started gaining traction in the enterprise. This is when Identity-as-a-Service (IdaaS) emerged as an alternate approach, with a cloud-based bridge to the cloud and lightweight directory integration. By leveraging the power of federation standards and the benefits of the cloud service model, IdaaS could provide a great user experience across web applications without the need for expensive infrastructure deployment or maintenance.

This shift continued, and now many organizations are beginning to centralize their Identity and Access Management (IAM) programs around IdaaS, moving the center of gravity of access control to the cloud. Of course, businesses still depend on legacy applications, so a modern IAM architecture cannot neglect them. Enterprises need to modernize on-premises applications, or implement solutions that enable more direct integration to IDaaS.

In this document, Okta will offer approaches that enterprises can employ to centralize access control and visibility across legacy and cloud applications and provide a great end user experience, while minimizing on-premises infrastructure.

Web Access Management, or WAM

There are generally considered to be two traditional WAM models: proxy-based, and agent- or plugin-based. The proxy-based approach routes all web traffic through network traffic manager, where HTTP requests can be denied or granted based on policies. This model introduced a single point of failure, but it offers protocol-level granular access control without installing any software. With agent-based approach, agents are installed on each app or web server. These plugins intercept HTTP requests, call out to the centralized policy server, and enforce access rules before responding. This approach removes the need to route all traffic through a proxy, which permits a distributed architecture, but carries the burden of having to install, update, and manage the agent software on every app server in your environment.

By contrast, modern standards like SAML and OIDC use a token-based approach. In this model, an identity provider supplies a token to the application (service provider), such as a JSON Web Token (JWT) or SOAP payload, with information about the user. With SAML, for example, the token is a SAML assertion, a SOAP-based Web Service message, signed by an identity provider, which contains claims about the user that application code can use to make access decisions. The token model uses the end user's encrypted browser context to exchange information between the identity provider (IdP) and the service provider (SP)—that is, the app. The nature of this model eliminates the need for the IdP and the SP to communicate directly, so networking changes are not required, no agents are required, and traffic need not be routed through a proxy. These benefits have contributed to SAML and OIDC's emergence and rise in popularity and traditional WAM models are now falling out of vogue.

However, applications need to be modified to support SAML or OIDC natively. Because modernizing these legacy applications competes with the enterprise's other priorities, that's not going to happen immediately or completely. The result: a disjoint architecture. New applications support modern standards, and older applications do not. IT Administrators must reconcile these two worlds with a single identity architecture to realize the potential of IAM.

Authentication Patterns

If your high-level goal is to manage identity and access across all of your apps, a good starting point is to understand the authentication patterns in use at your organization. At Okta, we've seen that web applications use one of the following methods to authenticate the end user:

Modern Patterns

Okta supports modern methods natively. Modern methods include:

- Forms-based Authentication—This pattern uses a custom page to capture the end user's username and password to authenticate the user. Okta supports Forms-based Authentication natively using our Secure Web Authentication plugin.
- SAML or WS-Fed-based Federation—This pattern allows end users to authenticate to an Identity Provider, which issues secure tokens that the end user can use to access other service and applications. Okta supports SAML and WS-Fed natively. You can read more about Okta and SAML on the Okta developer site.
- OIDC-based Federation—This pattern is a modern version of SAML. It allows end users to authenticate to service and provides a means to exchange identity information securely across services. Okta supports OIDC natively. You can read more about Okta and OIDC on the Okta developer site.

Legacy Patterns

- Okta does not support legacy patterns natively.
- No Authentication—This is also known as Anonymous Access. In this pattern, anyone can access a site without authenticating first. For web applications intended to be public, this is fine, but sometimes these pages require more security. In these cases, Okta recommends that customers improve security by forcing authentication, and allowing only authenticated users to access the app.
- Header-Based Authentication—A web access management system prompts the end user for authentication, then injects identity data into the HTTP Headers in the user's browser for consumption by the protected application. Common WAM systems include CA Siteminder, Oracle Access Manager and Tivoli Access Manager. Okta recommends migrating to a modern proxy-based architecture to accommodate this pattern.
- Client Certificate-based Authentication—This pattern utilizes a PKI certificate to authenticate the end user to an application. This is facilitated by most web servers natively, but can also be implemented using a WAM system. If the app cannot be modernized, Okta recommends leveraging a modern proxy-based architecture to accommodate this pattern.
- Windows Authentication—This pattern is also called Kerberos authentication (depending on the protocol used). This pattern silently logs the user using the active Windows domain session. This requires domain permissions and only works for internal users by default. Okta recommends integrating with a proxy-based architecture to provide remote access to these applications.

Bringing Legacy Apps into the Fold

Now that we understand the different patterns, let's talk about how to approach integrating all of your existing web applications into your IAM platform. We've used the following decision tree with customers and it's worked well. Here's the whole thing, and after the jump we'll step through it.

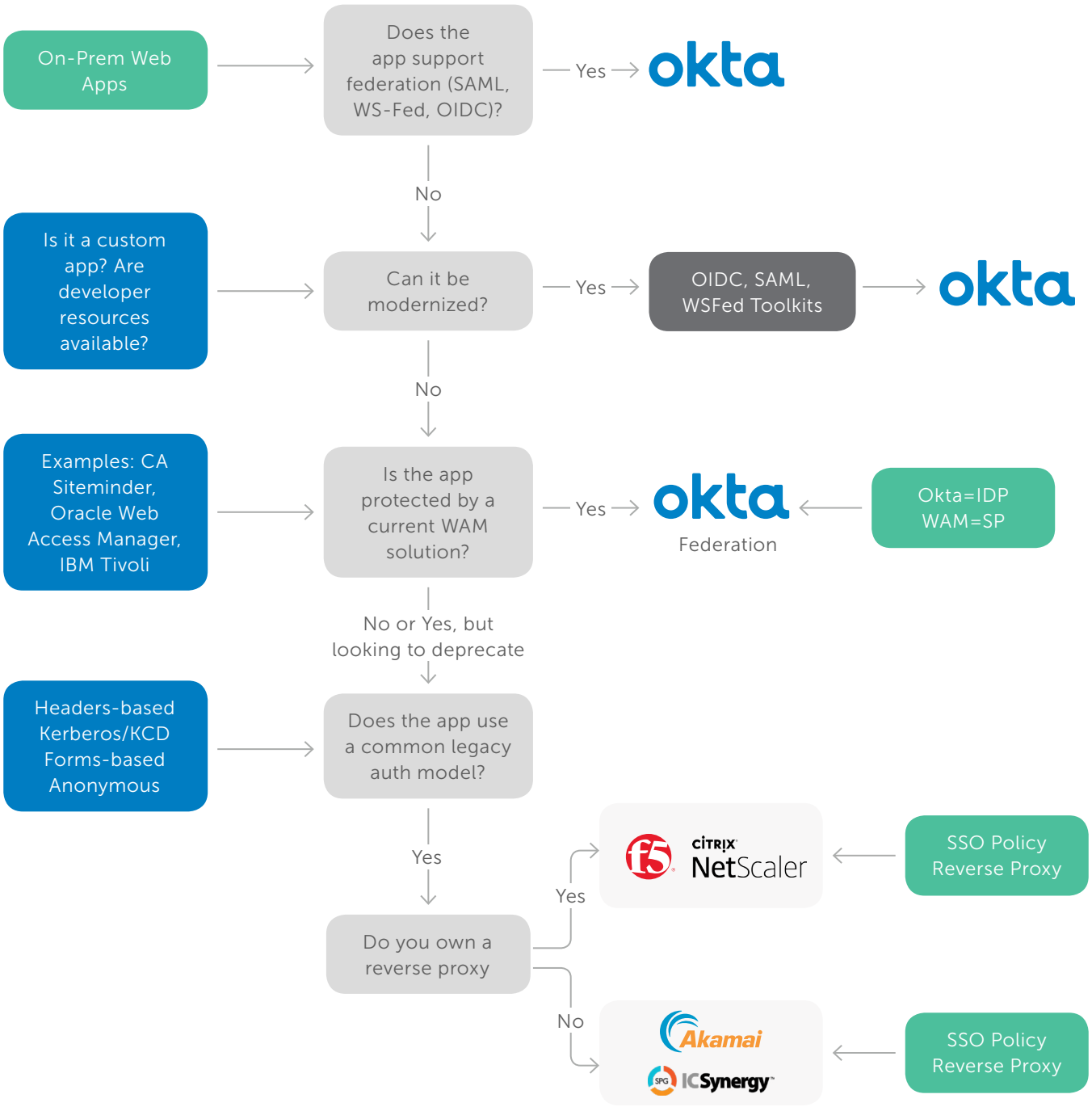

Figure 1. Decision Making Process for Integration Legacy Applications into a Modern Identity Platform

Decision Process—Step-by-Step

- **Does the app already support a modern pattern?** Determine whether the application supports SAML. Most enterprise-focused web applications have a built-in SAML capability, but the capability may need to be enabled, and sometimes an add-on needs to be purchased. Once the SAML capability is enabled on the application, use an Okta Application Network (OAN) pre-built SAML integration to connect the application rapidly. Okta have over 800 SAML application integrations, but if for some reason an integration is not available, create your own using a SAML 2.0 Template App in the OAN. (And make sure you let us know, so that we can add your app to the catalog.)

- **Can you modernize it?** This applies mostly to custom web applications. If you are able to modify the application, it's straightforward to add SAML or OIDC support to an existing web application. The implementation varies based on platform and development language so the Okta Developer site offers plenty of guidance across the most popular platforms. Modernizing applications take some time and effort, but it's worth it. This approach is low-cost to maintain, easy to integrate into Okta, simple to administrate, requires no extra hardware, and it's standards-based, which reduces lock-in.

- **Do you have a WAM deployed now?** You may already be using a WAM solution like CA Siteminder or Oracle Access Manager to protect applications that don't support modern standards. As we established above, though, not all WAM models are alike, and it matters whether you've deployed an agent-based model or a proxy-based model.

Authentication (AuthN) versus Authorization (AuthZ)

Authentication refers to the binding of a user to an account using some secure credential, like a password. Authorization refers to the enforcement of access control within the app, and there are two types, coarse-grained and fine-grained. Authentication restrictions, such as allowing authentication to the application only for certain groups, are a common way to implement coarse-grained authorization. Fine-grained authorization protects individual elements (e.g., pages, zones, or even individual DOM elements) within the application itself based on the authenticated user's attributes or roles.

Legacy WAM tools can provide very fine-grained authorization. They usually do so in proprietary ways, which makes it costly to replace a WAM solution in scenarios where fine-grained authorization is a requirement. That is, without replacing the app itself. So Okta's recommendation here is to figure that additional complexity into your cost analysis when determining if full WAM replacement is right for your organization right now.

Agent-based model: If you've deployed WAM in an agent-based model, we've seen customers use two approaches depending on their priorities:

i. Customers focused on business agility to onboard their new applications typically leave the legacy on-premises WAM in place with the idea that over time, most workloads and on-premises applications are moving to the cloud and this passive approach will reduce the WAM footprint automatically and at some point, in the future, they would be able to turn it off. This has the advantage of avoiding the up-front modernization costs, while still centralizing management in Okta, improving the end user experience, and introducing a migration vector.

> **Note:** From working with our customers, we've learned that if you've got more than about 25 applications that are currently integrated into a WAM solution, especially if they're using an agent-based model, this option is recommended. That's because the time, effort, and coordination involved in migrating that many applications or more can be significant and organizational delays are common.

ii. Customers focused on cost reduction and standardization on a single platform have aggressively migrated on-premises applications over to Okta either using federated proxies/agents or converting the applications to support federation. As described above, modernization requires up-front development work, but the benefits are often worth it.

Proxy-based model: If you've deployed WAM in a proxy-based model, you have two options: you can leave it in place and federate Okta to it, or you can eliminate redundancy and simplify your architecture by leveraging a traffic manager/reverse proxy like F5 BIG-IP, Citrix NetScaler, Akamai Enterprise Access, or ICSynergy SP Gateway. We'll talk more about the migration on the next step.

- **Do you currently own a traffic manager?** If you own one of the products listed above, you may not know this, but you can use that to facilitate a proxy-based WAM architecture. These products are typically deployed to manage network traffic and/or serve as a reverse proxy to make internal applications internet-accessible. If you currently own one of these products, Okta recommends leveraging the access management capabilities therein to extend Okta to the enterprise.

Putting It All Together

So, what does this best-of-breed architecture look like? Figure 2 below shows how everything fits together.

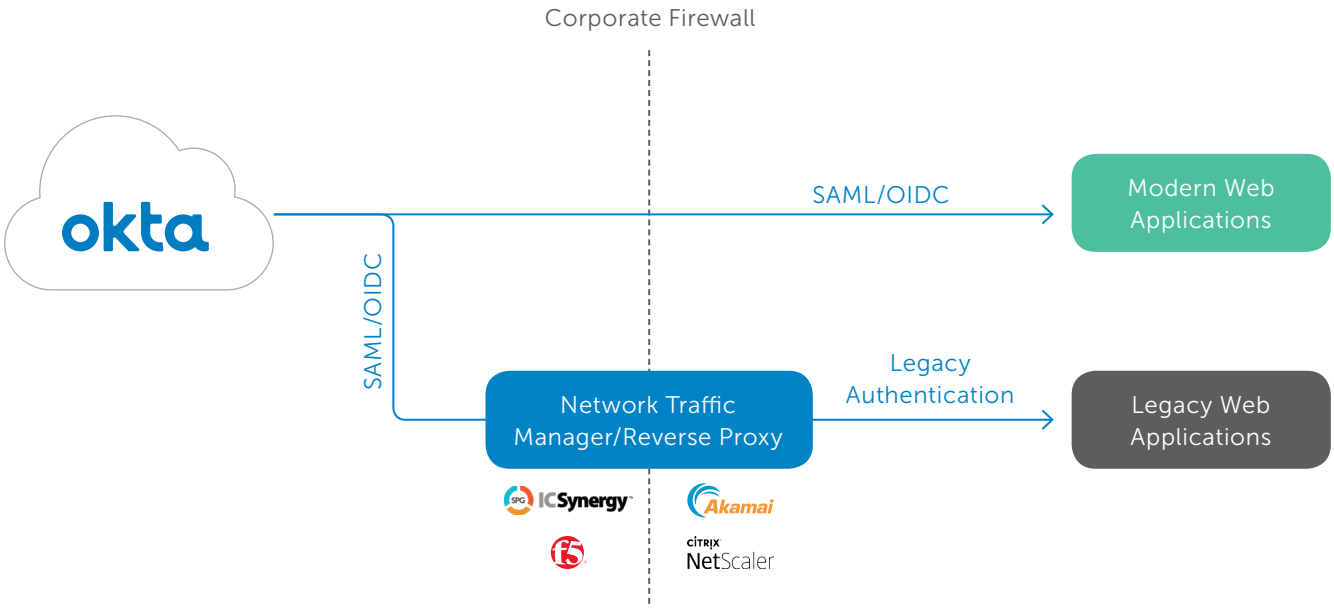

Figure 2. Reference Architecture for Best-of-Breed Approach to Legacy Application Access

This is a simple architecture, and that's really the point. The primary benefit is that it uses whatever's already in place.

By integrating with network traffic managers, Okta provides a seamless, single sign-on experience for end users whether they're accessing on-premises applications or SaaS solutions. Proxying and routing HTTP traffic, managing the load, enforcing security at the networking layer are all handled by best-of-breed technologies, which ensure a consumer grade experience for end users from anywhere in the world.

Other Common Use Cases

Secure Access to On-Prem Apps from Outside the Firewall

Enterprises typically use Okta for the 5,000+ integrations pre-built into the Okta Application Network. Okta also has full support for federation protocols for additional applications that support federation standards. Applications in the cloud with any kind of login form can, additionally, be easily added to Okta. When applications are behind the firewall, authentication is not enough. Users must gain network access to the application. This can be cumbersome with the standard VPN approach, requiring multiple steps for the end user.

With a leading reverse proxy integrated with Okta, end users can authenticate once into Okta and seamlessly access on-prem applications. This architecture extends Okta's authentication capability to applications that do not have native authentication mechanisms or support header-based authentication. Finally, a reverse proxy provides an additional layer of security for on-prem applications by securing all HTTP traffic to and from an application.

Contractor and Partner Access to On-Prem SharePoint Portals

It can be a challenge to expose SharePoint Server (on-prem) to external users such as contractors or partners. Okta can integrate to SharePoint for SSO via federation. However, in order to use certain SharePoint modules, such as SharePoint business intelligence features, users must have a Kerberos token. Network proxies support the key requirement of exchanging SAML assertions for Kerberos tokens, enabling use of the full set of functionality in SharePoint. Okta, paired with best-in-class network proxies, can manage contractor or partner identities and enforce multi-factor authentication.

Multi-Factor Authentication for Legacy Applications on IaaS

Enterprises that are moving on-prem servers to IaaS need to have a strategy for protecting access to those resources. One of the benefits of moving to IaaS may be that the service can be more easily reached from any network. Network Proxies play a key role in exposing these on-prem servers to the internet. Given the greater exposure, a good practice is to require multi-factor authentication to access these services. Okta can easily add multi-factor authentication with a soft token (iOS, Android or Windows Phone), SMS or voice as factors.

One End User Portal for All Applications, On-Prem and Cloud

The Okta end user portal is built to make it easy for end users to access all their applications from one place. The portal is customizable by end users, which drives a high level of user adoption. Typically, organizations using the Okta portal want all the end users' applications exposed and accessible through the portal. Integrating Okta with Network Proxies enables the user to log in once to Okta, and access all applications, cloud and on-prem, in one place.

Benefits of this Approach

This solution integrates best-of-breed technologies to do exactly what they're best at: Okta provides secure and seamless access to any application or resource, a traffic manager routes network traffic efficiently, and access policies on the traffic manager integrate with legacy applications and empower fine-grained authorization for legacy web applications. This solution offer benefits not possible with any single-vendor solution on the market today.

- **Embrace Standards**—Enterprises want to avoid vendor lock-in and using SSO standards like SAML for application access helps keep IT architecture resilient to change.
- **Eliminate Redundancy**—By fully utilizing the functionality of IdaaS and network traffic managers, legacy WAM providers can often be removed from the environment all together.
- **Scale and Perform**—Traffic Managers and Reverse Proxies are purpose-built to protect some of the largest applications in the world.
- **Consumer-Grade End User Experience**—Extend Okta's best-in-class SSO user experience to applications that IdaaS solutions are not optimized to integrate with.
- **Support All New Use Cases**—Integration with Okta IdaaS to position your organization to enforce multi-factor authentication and secure the mobile experience for every web app in your company.

Conclusion

By embracing the cloud, you will help your business to accelerate and gain critical advantages over your less agile competitors. Of course, the transition does not happen in the blink of an eye, so it's important to support your legacy systems for the foreseeable future. A modern identity management platform and a smart access management strategy will accelerate your IT evolution while bridging the gap between your trusty on-premises applications and the new technologies you're adopting.

Getting Started with Your Free Trial

To discover how easy it is to overcome identity and access management challenges in the cloud, visit www.okta.com/free-trial to get started with Okta.

About Okta

Okta is the foundation for secure connections between people and technology. By harnessing the power of the cloud, Okta allows people to access applications on any device at any time, while still enforcing strong security policies. It integrates directly with an organization's existing directories and identity systems, as well as 4,000+ applications. Because Okta runs on an integrated platform, organizations can implement the service quickly at large scale and low total cost. More than 2,500 customers, including Adobe, Allergan, Chiquita, LinkedIn, MGM Resorts International and Western Union, trust Okta to help their organizations work faster, boost revenue and stay secure.

For more information, visit us at www.okta.com or follow us on www.okta.com/blog.


okta