

What does Okta do?

Identity theft is not a joke, Jim



Justin ✓

🔒 Dec 21, 2020

♡ 15

💬 4



The TL;DR

Okta is an enterprise-focused identity provider: they take care of managing usernames, passwords, and permissions so you can focus on your apps.

- If you've ever used "sign in with Google" to log into an app, you've used **Single Sign On (SSO)** - a way to outsource identity to third-party providers
- When you build SSO into your app, you use **Identity Providers (IdPs)** like Google or Okta to take care of the actual authentication
- Beyond just basic authentication, Okta also helps with provisioning user accounts, mobile device management, and directory sync
- Enterprises (really big companies) have **tight security requirements** – they're often not allowed to sign contracts with vendors that don't support SSO

Okta's share price has grown by almost 15x (1,500%) to \$270 since their late 2017 IPO. As more enterprises move to cloud apps, Okta [has been printing money](#) – 50% YoY revenue growth over the past few years, to be exact.

Single Sign On and IdPs

Every app you use requires authentication – you have an account, and every now and then you need to prove that you are who you say you are, so you can use that account. The most popular method of *implementing* that is username and password, stored on company servers. But for reasons that are obvious to anyone sharing a Netflix account, this auth method is *not* ideal: you need a different username/password for *every service you use*, and passwords are very hackable.

In other words, most apps work with a **decentralized identity model**:

- Each app has its own database of usernames and passwords implemented differently
- Adding security features like multi factor authentication requires custom work

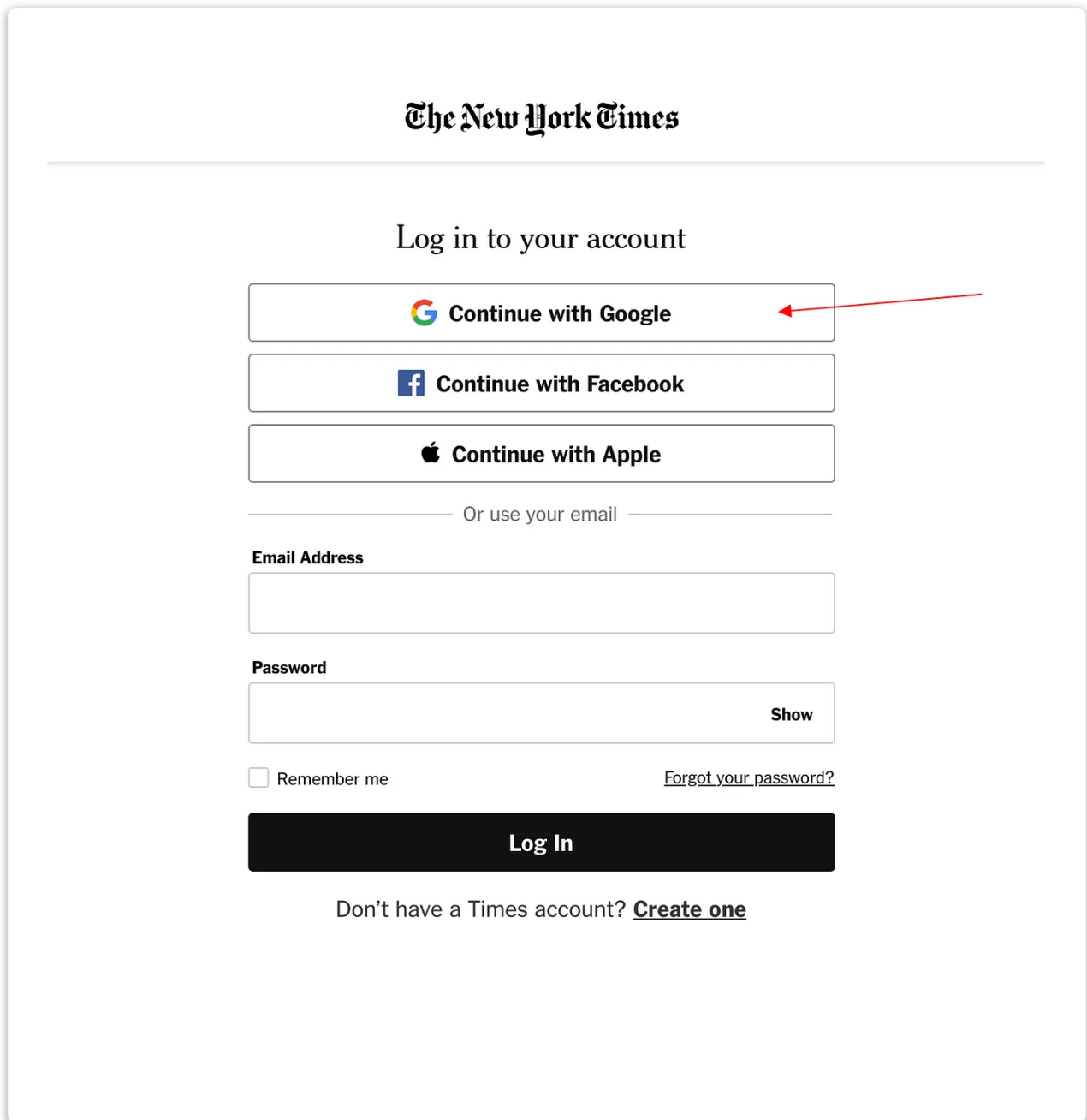
The problem with this model is, well, that's it's decentralized - every company needs to re-invent the wheel and implement security features independently.

⌘⌘ Related Concepts ⌘⌘

This, by the way, is where companies like [Auth0](#) make their money - they provide tooling that helps companies build authentication in house.

⌘⌘ Related Concepts ⌘⌘

Single Sign On solves this problem by letting companies outsource their authentication to a third party, like Google or Okta. When you're trying to log into the New York Times and click "sign in with Google," the NYT is using Google as an outsourced identity provider: you log in with your Google login, and that proves that you are who you say you are. The whole process of checking your username/password against what exists in the database happens on *Google's servers*, not the NYT's.



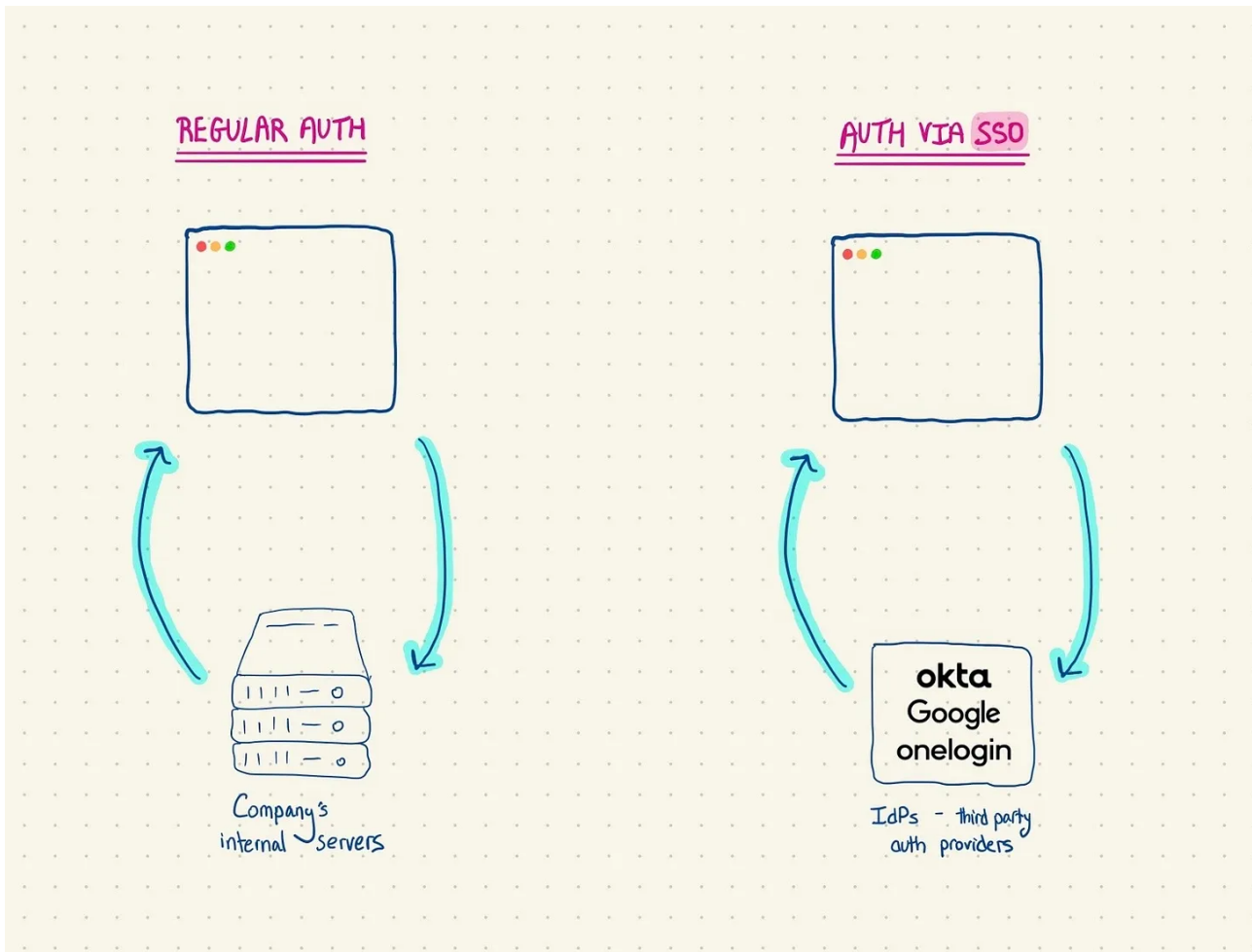
The screenshot shows the login page for The New York Times. At the top is the "The New York Times" logo. Below it is the heading "Log in to your account". There are three buttons for social login: "Continue with Google" (with a red arrow pointing to it), "Continue with Facebook", and "Continue with Apple". Below these is a link "Or use your email". Underneath are input fields for "Email Address" and "Password". The password field has a "Show" button. Below the password field is a checkbox for "Remember me" and a link "Forgot your password?". At the bottom is a large black "Log In" button. Below the button is a link "Don't have a Times account? [Create one](#)".

To understand the basics of SSO, you need to know the terminology for both sides of the ecosystem:

- **Service Provider (SP)** – the app you're building that needs authentication. In our example, the New York Times.
- **Identity Provider (IdP)** – the service your app is using for outsourced authentication. In our example, Google (more accurately, GSuite).

When a user attempts to log into the New York Times site (our SP) with SSO, the NYT site redirects them to Google's servers (our IdP) to authenticate. If the user successfully signs on with Google, Google sends a message back to the NYT site

saying “hey, all’s good, let this guy in.” If this sounds simple, it’s because it is! The hard part is in actually implementing this stuff, but thankfully that’s not your job (yet).



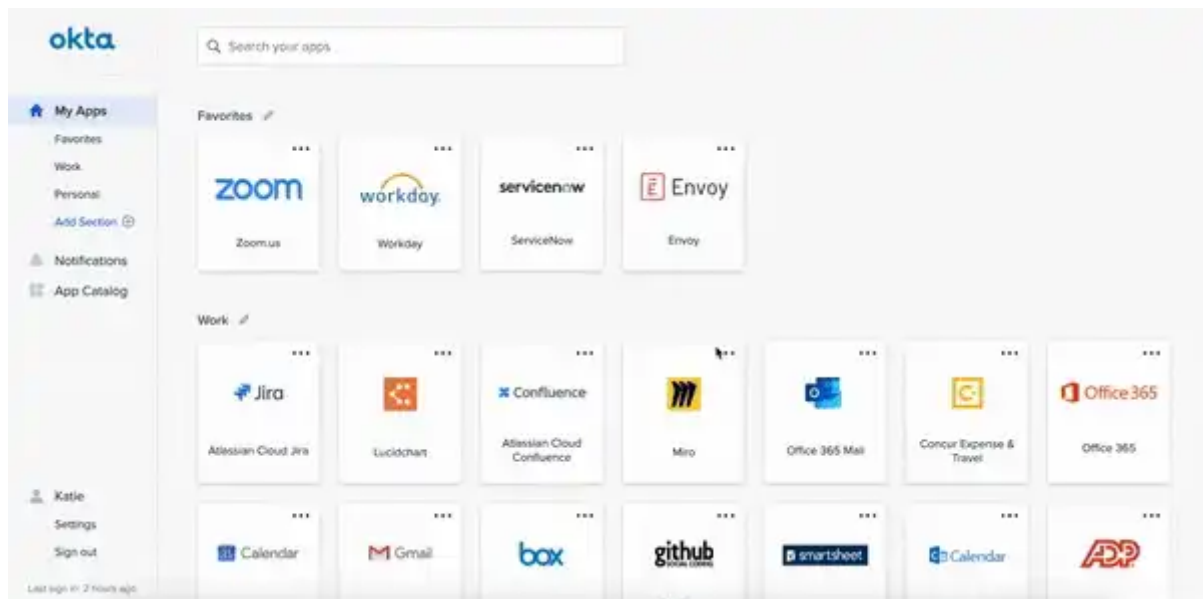
What Okta actually does

Okta is an identity provider, so they’re one of the companies that other companies outsource their identification to. The easiest way to understand this is through the perspective of the user: let’s imagine today is your first day working at FedEx corporate ([an Okta customer](#)). FedEx uses a bunch of tools that you need access to: Looker for data, Salesforce for a CRM, and Gmail for email.

If FedEx is using SSO (they are), instead of a username and password for each of those tools, you have **one Okta login** that you use for all of them. When you load the Looker app, it will prompt you to sign in via Okta, and you’ll get redirected to

an Okta login screen. If you authenticate with Okta successfully, you'll get sent back to Looker and can get started.

This type of flow we just described is called **SP-initiated**, because you started the sign in process from the app you want to log into (Looker, the service provider), not from the identity provider (Okta). But you can also do **IdP-initiated** login: Okta provides a portal of all of the apps you have access to, and you can launch the app from there *already authenticated* instead.



Deeper Look

The hard engineering work here is integrating your system with Okta and the various other IdPs that your customers are going to want to use (like [OneLogin](#)). The most popular open source protocol for doing this is called [SAML](#) (security assertion markup language), and it's notoriously frustrating to work with.

Deeper Look

SSO is Okta's bread and butter, but over the years they've taken advantage of their position in the ecosystem (a central authority for identity) to build a strong supporting feature set. Like any enterprise-focused website, understanding these features is near impossible. Here are a few broken down:

1. **Multi factor authentication**

You've seen enough public breaches to know that username and password alone isn't particularly secure: authentication through *multiple touch points* (that security code sent to your phone, or the authenticator app) is getting more and more popular. Okta lets you require this for all of your users, and provides a mobile app (it has 2 stars in the app store!) to generate 2FA codes.

2. Directory sync

Large organizations have hundreds of internal user groups with different permission levels – an engineering manager might have admin permissions to databases, while junior engineers can only read from those databases. Okta allows you to build a directory of these users and roles, and sync them with other apps like Github automatically.

3. Lifecycle management

User Lifecycle Management (ULM) is the process of creating and deleting user accounts as people join your organization. Back to our FedEx example, when you join, your IT Admin will need to create accounts for you in Looker, Gmail, and the like – and when you leave, those accounts need to get deactivated so you can't access company data. Okta provides tooling to do this automatically based on the directory settings you've configured.

Deeper Look

The open source **SCIM** protocol (system for cross-identity management) is the most popular option for building these "directories" of permissions and automating user lifecycles. An example: if an employee leaves your company, you can set up SCIM and Okta to *automatically* de-provision their accounts.

Deeper Look

Terms and concepts covered

Single Sign On (SSO)

Identity Provider (IdP)

Service Provider (SP)

Security Assertion Markup Language (SAML)

System for Cross-Identity Management (SCIM)

Further reading

- For a more in depth introduction to SSO and SAML, I wrote a beginner's guide for developers [here](#)
- Okta covered SCIM in a beginner's introduction [here](#), and you can find the actual spec (for the technically inclined) [here](#)

Questions from the stands

→ *What does Databricks do? (Evan @ True Search)*

Ah, Databricks - a classic "we sell to the enterprise so our website doesn't say what our product actually does" company. The long story short is that they provide a managed service for running intensive data workloads on an open source technology called [Apache Spark](#). Spark lets you run distributed "jobs" - think training a big machine learning model, or a big ETL pipeline - in Python. But it's [notoriously frustrating to set up](#) and run on your own infrastructure, so Databricks takes care of that for you (along with a nice UI for writing these jobs). Over the past few years, they've expanded into a couple of other frameworks like [MLFlow](#).

→ *What's the difference between a data lake and a data warehouse? (Nandu @ Canaan)*

A **data lake** is an unstructured place where you just plop all of your data, whatever form it may be in. It's typically optimized for low storage costs. A **data warehouse** (Snowflake, BigQuery, Redshift), on the other hand, is a structured environment where you store transformed, organized data that's already in a format you can use to answer important business questions. AWS actually has a great explainer between the two [here](#).

Note that as warehouse storage has gotten cheaper, the distinction between lakes and warehouses has been getting more cloudy, because teams are putting their raw source data directly into the warehouse instead of a data lake. This is what people mean when they say ETL (extract -> transform -> load) is moving towards ELT (extract -> load -> transform).

→ *What does Fastly do? (Zevi @ Houlihan Lokey)*

Fastly is a CDN, which stands for Content Distribution Network - they've built a network of servers across the globe that you can use to get your web pages to your customers faster. Like [Cloudflare](#) (post coming soon), they also provide security for DDoS attacks against your site.

4 Comments



Write a comment...



wam Writes YOLO Dec 22, 2020

THIS -> "Like any enterprise-focused website, understanding these features is near impossible."

♡ 3 Reply Collapse ...

1 reply



Jeff Dec 22, 2020

Thank you for the explainer! Outside the scope of Technically, but I wonder what their sales motion is? Who are they selling to into the org? You highlight their revenue growth,

but I don't immediately understand why a CIO would want a whole vendor dedicated to this when they get the same capabilities from MSFT. Likely related to the additional features you highlight.

And this "objection" (why not just get these services in a bundle) could apply to like seemingly all enterprise SaaS so what do I know.

 Reply Gift a subscription Collapse ...

2 more comments...

© 2023 Justin · [Privacy](#) · [Terms](#) · [Collection notice](#)
[Substack](#) is the home for great writing