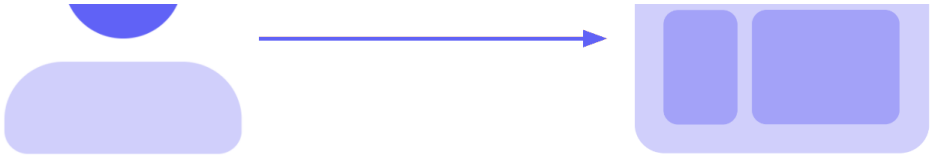
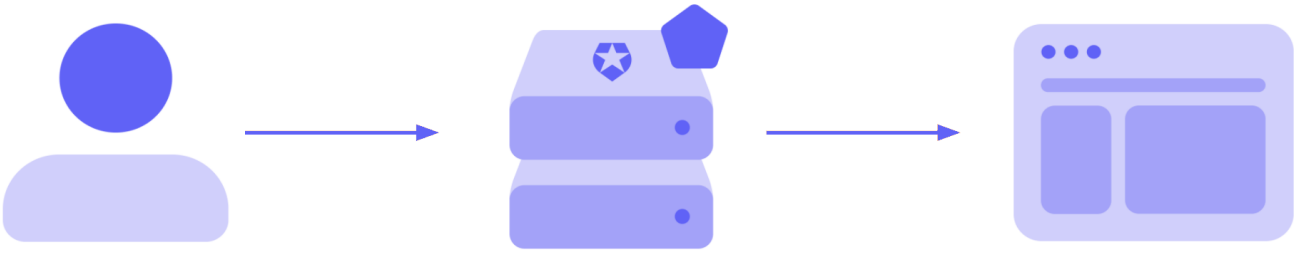


A user wants access to a resource.



Identity and access management verifies the user and controls their access to the resource.

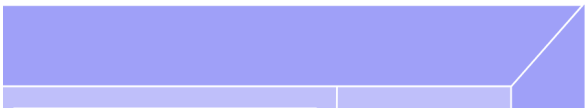
- **Authentication** is the verification of a digital identity. Someone (or something) authenticates to prove that they're the user they claim to be.
- **Authorization** is the process of determining what resources a user can access.

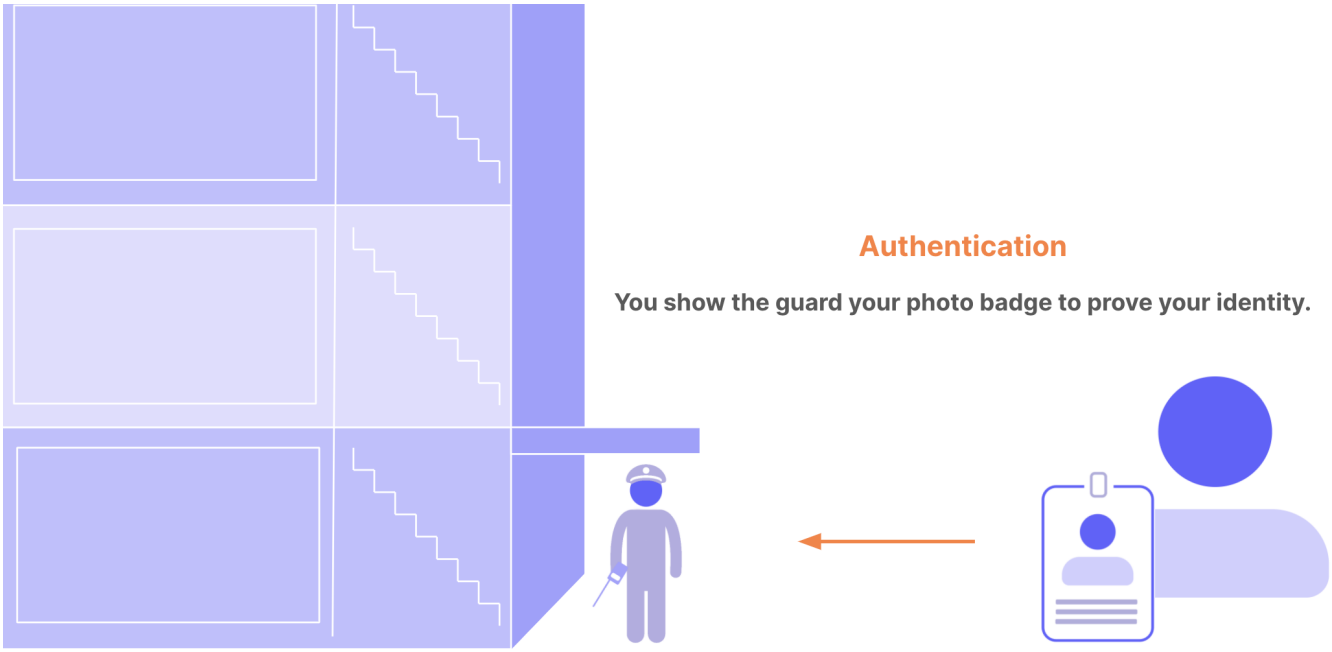
The difference between authentication and authorization

It's common to confuse authentication and authorization because they seem like a single experience to users. They are two separate processes: authentication proves a user's identity, while authorization grants or denies the user's access to certain resources.

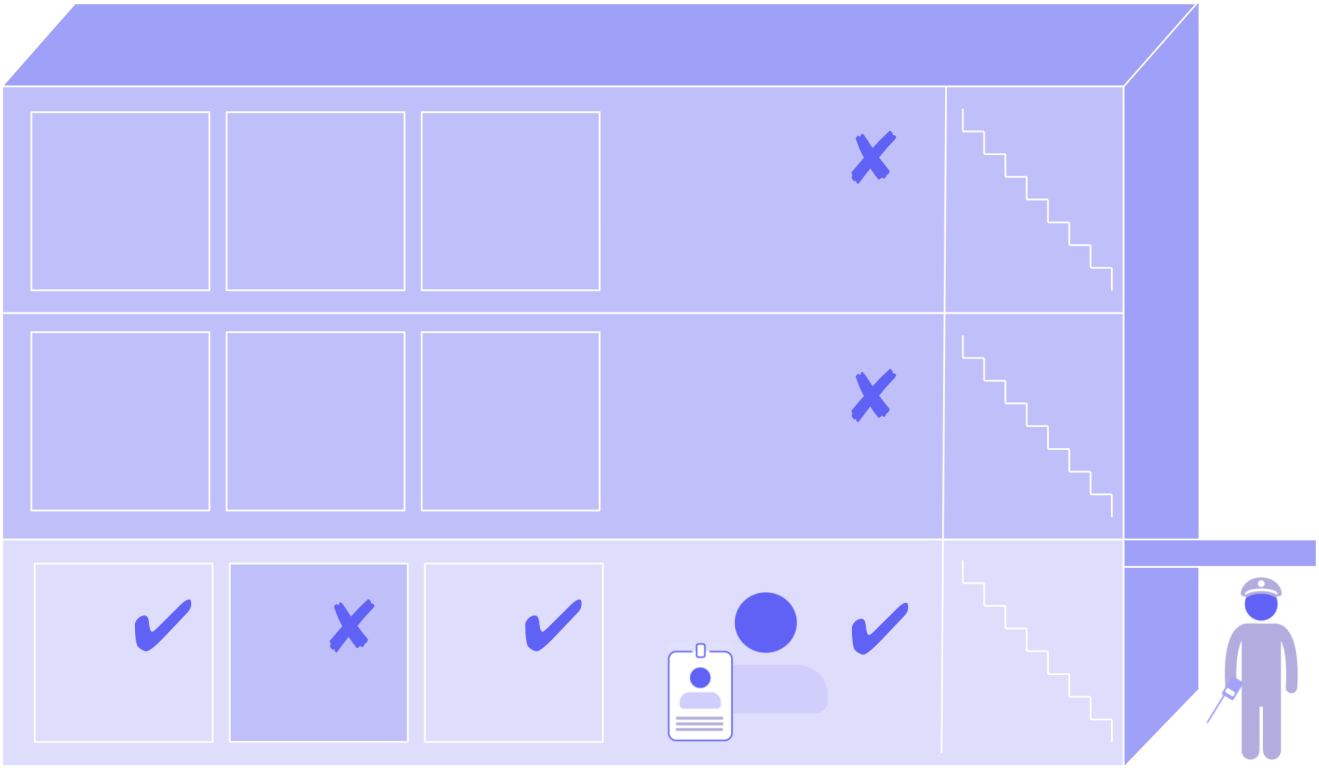
You can think of authentication and authorization as the security system for an office building. Users are the people who want to enter the building. Resources that people want to access are areas in the building: floors, rooms, and so on.

Authentication: When you enter the building, you must show your photo ID badge to the security guard. The guard compares the photo on the badge to your face. If they match, the guard lets you through the door to try to access different areas of the building. The guard doesn't tell you what rooms you can access; they only get proof that you are who you claim to be. This is authentication: confirming user identity.





Authorization: In this scenario, imagine the elevators and doorways in the building have key sensors for access. The chip in your badge gives you access only to the first floor, which your company occupies. If you swipe your badge to enter any other floor, your access is denied. You can access your private office but not those belonging to your colleagues. You can enter the supply room but not the server room. This is authorization: granting and denying access to different resources based on identity.



To learn more about authentication and authorization, read Authentication vs. Authorization.

What does IAM do?

Identity and access management gives you control over user validation and resource access:

- How users become a part of your system
- What user information to store
- How users can prove their identity
- When and how often users must prove their identity
- The experience of proving identity
- Who can and cannot access different resources

You integrate IAM with your application, API, device, data store, or other technology. This integration can be very simple. For example, your web application might rely entirely on Facebook for authentication, and have an all-or-nothing authorization policy.

Your app performs a simple check: if a user isn't currently logged in to Facebook in the current browser, you direct them to do so. Once authenticated, all users can access everything in your app.

It's unlikely that such a simple IAM solution would meet the needs of your users, organization, industry, or compliance standards. In real life, IAM is complex. Most systems require some combination of these capabilities:

- **Seamless signup and login experiences**—Smooth and professional login and signup experiences occur within your app, with your brand's look and language.
- **Multiple sources of user identities**—Users expect to be able to log in using a variety of social (such as Google or Linkedin), enterprise (such as Microsoft Active Directory), and other identity providers.
- **Multi-factor authentication (MFA)**—In an age when passwords are often stolen, requiring additional proof of identity is the new standard. Fingerprint authentication and one-time passwords are examples of common authentication methods. To learn more, read Multi-Factor Authentication (MFA).
- **Step-up authentication**—Access to advanced capabilities and sensitive information require stronger proof of identity than everyday tasks and data. Step-up authentication requires additional identity verification for selected areas and features. To learn more, read Add Step-up Authentication.
- **Attack protection**—Preventing bots and bad actors from breaking into your system is fundamental to cybersecurity. To learn more, read Attack Protection.
- **Role-based access control (RBAC)**—As the number of users grows, managing the access of each individual quickly becomes impractical. With RBAC, people who have the same role have the same access to resources. To learn more, read Role-Based Access Control.

Facing this level of complexity, many developers rely on an IAM platform like Auth0 instead of building their own solutions.

How does IAM work?

"Identity and access management" is not one clearly defined system. IAM is a discipline and a type of framework for solving the challenge of secure access to digital resources.

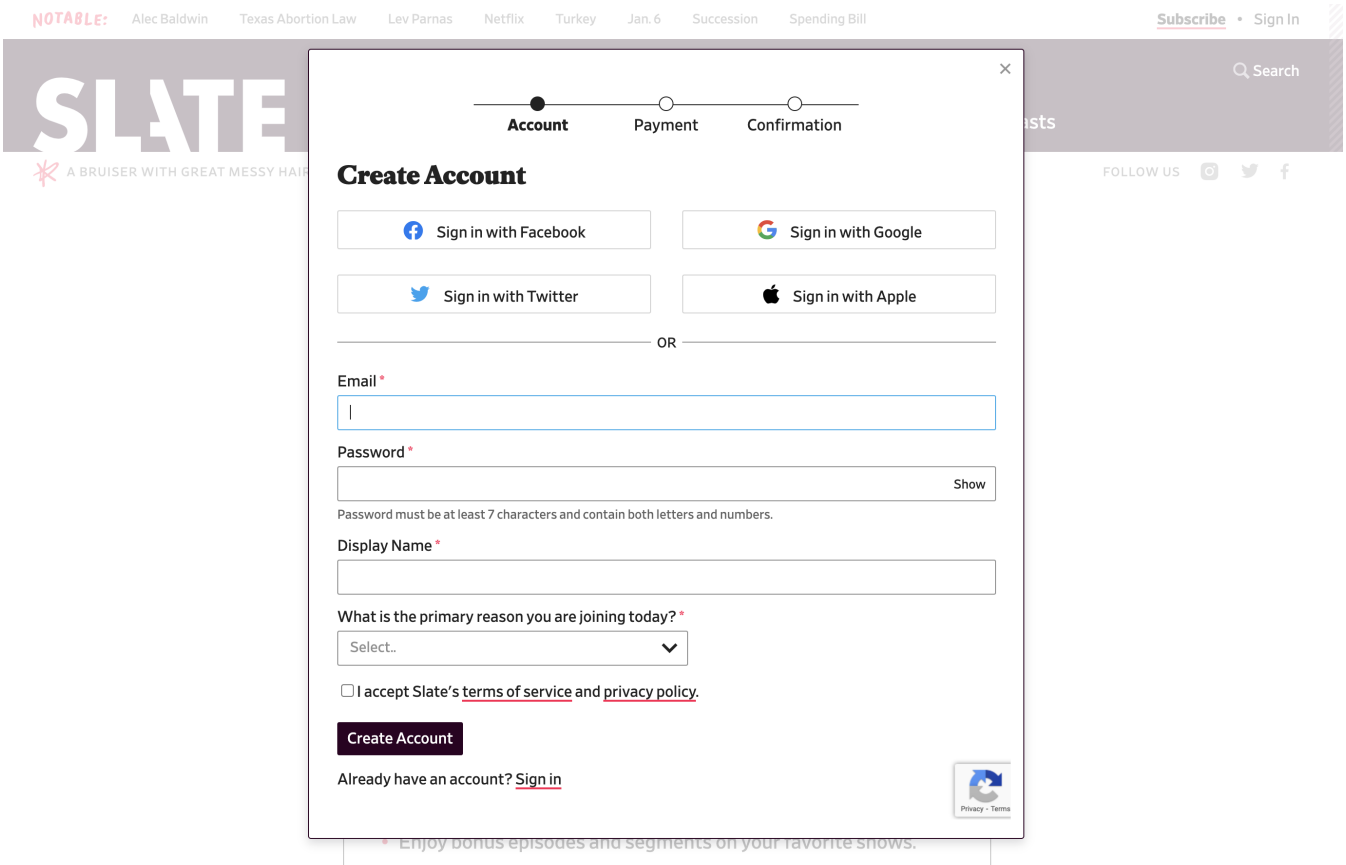
There's no limit to the different approaches for implementing an IAM system. This section explores elements and practices in common implementations.

Identity providers

In the past, the standard for identity and access management was for a system to create and manage its own identity information for its users. Each time a user wanted to use a new web application, they filled in a form to create an account. The application stored all of their information, including login credentials, and performed its own authentication whenever a user signed in.

As the internet grew and more and more applications became available, most people amassed countless user accounts, each with its own account name and password to remember. There are many applications that continue to work this way. But many others now rely on identity providers to reduce their development and maintenance burden and their users' effort.

An identity provider creates, maintains, and manages identity information, and can provide authentication services to other applications. For example, Google Accounts is an identity provider. They store account information such as your user name, full name, job title, and email address. Slate online magazine lets you log in with Google (or another identity provider) rather than go through the steps of entering and storing your information anew.



Identity providers don't share your authentication credentials with the apps that rely on them. Slate, for example, doesn't ever see your Google password. Google only lets Slate know that you've proven your identity.

Other identity providers include social media (such as Facebook or LinkedIn), enterprise (such as Microsoft Active Directory), and legal identity providers (such as Swedish BankID).

Authentication factors

Authentication factors are methods for proving a user's identity. They commonly fall into these basic types:

Factor type	Examples
Knowledge (something you know)	Pin, password
Possession (something you have)	Mobile phone, encryption key device
Inherence (something you are)	Fingerprint, facial recognition, iris scan

IAM systems require one or many authentication factors to verify identity.

Authentication and authorization standards

Authentication and authorization standards are open specifications and protocols that provide guidance on how to:

- Design IAM systems to manage identity
- Move personal data securely
- Decide who can access resources

These IAM industry standards are considered the most secure, reliable, and practical to implement:

OAuth 2.0

OAuth 2.0 is a delegation protocol for accessing APIs and is the industry-standard protocol for IAM. An open authorization protocol, OAuth 2.0 lets an app access resources hosted by other web apps on behalf of a user without ever sharing the user's credentials. It's the standard that allows third-party developers to rely on large social

platforms like Facebook, Google, and Twitter for login. To learn more, read OAuth 2.0 Authorization Framework.

Open ID Connect

A simple identity layer that sits on top of OAuth 2.0, OpenID Connect (OIDC) makes it easy to verify a user's identity and obtain basic profile information from the identity provider. OIDC is another open standard protocol. To learn more, read OpenID Connect Protocol.

JSON web tokens

JSON web tokens (JWTs) are an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. JWTs can be verified and trusted because they're digitally signed. They can be used to pass the identity of authenticated users between the identity provider and the service requesting the authentication. They also can be authenticated and encrypted. To learn more, read JSON Web Tokens.

Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML) is an open-standard, XML-based data format that lets businesses communicate user authentication and authorization information to partner companies and enterprise applications that their employees use. To learn more, read SAML.

Web Services Federation (WS-Fed)

Developed by Microsoft and used extensively in their applications, this standard defines the way security tokens can be transported between different entities to exchange identity and authorization information. To learn more, read Web Services Federation Protocol.

Why use an IAM platform?

Why do so many developers choose to build on an identity and access management platform instead of building their own solution from the ground up?

User expectations, customer requirements, and compliance standards introduce significant technical challenges. With multiple user sources, authentication factors, and open industry standards, the amount of knowledge and work required to build a typical IAM system can be enormous. A strong IAM platform has built-in support for all identity providers and authentication factors, offers APIs for easy integration with your software,

and relies on the most secure industry standards for authentication and authorization.

For those who haven't yet decided whether to build or buy an IAM solution, Build vs. Buy: Guide to Evaluating Identity Management is a useful resource.

Reports

Videos

Webinars

Case Studies

Podcasts

Contact Sales

Status • Legal • Privacy • Terms