authentication methods like SSO, MFA, and OAuth2.

SSO, MFA, and OAuth2 are **distinct authentication and authorization methods**: **SSO** lets users log in once to access multiple services, **MFA** adds extra security layers by requiring multiple verification factors, and **OAuth2** is an authorization framework that lets third-party apps access data on behalf of a user without needing their password.

Single Sign-On (SSO)

What it is: A system that allows users to log in with a single set of credentials to access multiple applications and services.

How it works: An identity provider (IdP) authenticates the user once. This authentication is then securely shared with other applications (service providers), allowing access without a new login.

Examples: Logging into Google and gaining access to Gmail, Drive, and Calendar with a single login.

Multi-Factor Authentication (MFA)

What it is: A security process that requires users to provide two or more verification factors to gain access to a resource.

How it works: It combines different types of "factors" to verify identity, such as:

Knowledge: Something the user knows (e.g., a password).

Possession: Something the user has (e.g., a one-time code from a mobile app or a security key).

Inherence: Something the user is (e.g., a fingerprint or facial scan).

Benefit: Significantly enhances security beyond a single password by making it harder for unauthorized individuals to gain access even if they have a password.

OAuth 2.0

What it is: An open-standard authorization framework that grants a third-party application limited access to a user's data on a resource server without sharing the user's credentials.

How it works: Instead of sharing a password, the user grants permission for the third-party app to access certain data. The resource server then issues short-lived access tokens to the app, allowing it to perform specific actions.

Example: When you "Log in with Google," OAuth 2.0 is used to authorize the app to access specific information from your Google account without you giving the app your Google password.

Note: OAuth 2.0 is for authorization, not direct authentication. For a complete authentication process, it is often used with an identity layer like OpenID Connect (OIDC), which is built on top of OAuth 2.0

Revision #2
Created 29 October 2025 02:43:33 by AI API
Updated 1 December 2025 14:09:34 by AI Channel