



The evolution of auth, from users to AI agents

BSidesSLC 2025 | Apr 11, 2025

Maya Kaczorowski

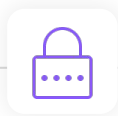
Maya Kaczorowski
Founder

OB/LIQUE



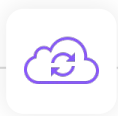
Evolution of authentication

Passwords



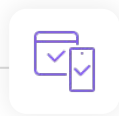
- Basic auth
- Password policies
- Password managers

Identity federation



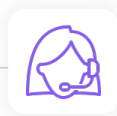
- All the acronyms: IdP, SSO, SAML, OIDC, OAuth, SCIM
- Social logins

Multi-factor authentication



- Hardware tokens
- Biometrics
- Passwordless
- WebAuthn
- Passkeys

AI agents



- Long tail of apps
- Distributed identities
- IdP discovery
- Delegation

Passwords



Email

✉ Enter your email

Password

🔒 Enter your password

Sign in

Passwords

- Introduced in 1960s at MIT
- Password hashing
- Password reuse



Email

✉ Enter your email

Password

🔒 Enter your password

🔗 [Forgot password?](#)

Sign in

Password complexity

- NIST SP 800-63
- Humans are predictable, not random
- Password rotation

Check out:

<https://neal.fun/password-game/>



Email

✉ Enter your email

Password

🔒 Enter your password

🔗 Fo Log in with my-email@example.com

Sign in

Password managers

- Users have 168 passwords on average
- Platform vs standalone password managers
- Cognitive burden

Identity federation



Email

✉ Enter your email

Password

🔒 Enter your password

🔗 [Forgot password?](#)

Sign in

or



Sign in with Microsoft



Sign in with Okta

Enterprise SSO

- Centralized identity stores (AD, LDAP)
- Federation protocols (SAML)
- Cloud identity providers (Okta, Ping)
- Authorization frameworks (OAuth)
- Authentication of delegated authority (OpenID Connect)



Email

✉ Enter your email

Password

🔒 Enter your password

🔗 [Forgot password?](#)

Sign in

or



Sign in with Microsoft



Sign in with Okta



Sign in with Google



Sign in with Facebook

Social login

- Improved both security and usability
- Let services depend on larger platforms for password management, including resets

Service authn: API keys

- API keys are like basic auth
- Secret managers
- OAuth authorization
 - Short-lived access tokens
 - Expiry & revocation
 - Permission scopes

Multi-factor authentication



Email

✉ Enter your email

Password

🔒 Enter your password

🔗 [Forgot password?](#)

Sign in

or



Sign in with Microsoft



Sign in with Okta



Sign in with Google



Sign in with Facebook

Two-factor authentication code

0

0

0

0

0

0

Multi-factor

- **1990s:** RSA SecurID

- **2000s:** SMS codes

- **2010s:** Authenticator apps

- **Mid 2010s:** Push notifications

- **Late 2010s:** Security keys



Email

✉ Enter your email

Password

🔒 Enter your password

🔗 [Forgot password?](#)

Sign in

or



Sign in with Microsoft



Sign in with Okta



Sign in with Google



Sign in with Facebook

Two-factor authentication code

0

0

0

0

0

0



Sign in with FaceID

Passwordless

● **2010s:** Magic links

● **2013:** TouchID

● **2015:** Windows Hello

● **2017:** FaceID

● **2019:** WebAuthn

Passkeys

The image shows a login interface for a service named 'iGen'. The top section, on a purple background, contains fields for 'Email' (with a placeholder 'Enter your email') and 'Password' (with a placeholder 'Enter your password' and a 'Forgot password?' link). Below these is a 'Sign in' button. A separator line with the word 'or' follows. Below that is a 'Sign in with Microsoft' button. The bottom section, on a white background, is titled 'Use a passkey'. It features a circular icon with a fingerprint symbol. Below the icon, it says 'Use your my-email@example.com passkey to sign in'. At the bottom of this section is another 'Sign in' button.

- **Promise:** phishing-resistant credentials that works across devices
- **Reality:** inconsistent, confusing experience across providers

AI agents

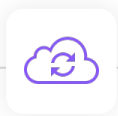
Evolution of authentication

Passwords



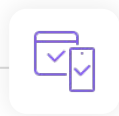
Human-generated
to
Random

Identity
federation



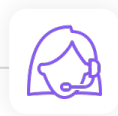
Decentralized
to
Centralized

Multi-factor
authentication



Static
to
Dynamic

AI agents



???

AI agent auth

Acting on behalf of

Looks like a user

- OAuth delegation
- User consent

MCP: OAuth resource server

Acting independently

Looks like a service

- OAuth scopes
- Expiry & revocation

A2A: any Open API auth (API keys, OAuth, OIDC)

Continuous, adaptive, and dynamic authentication

What should you do?

For an organization

- MFA as soon as possible
- OAuth use via your IdP

For a service

- Only SSO with WebAuthn — no passwords
- Build OAuth scopes

Thank you

Get the slides: <https://tinyurl.com/bsidesslc-auth>