

Modern Authentication with WebAuthn



Jade Philippe

AustinRB Meetup
July 9, 2024

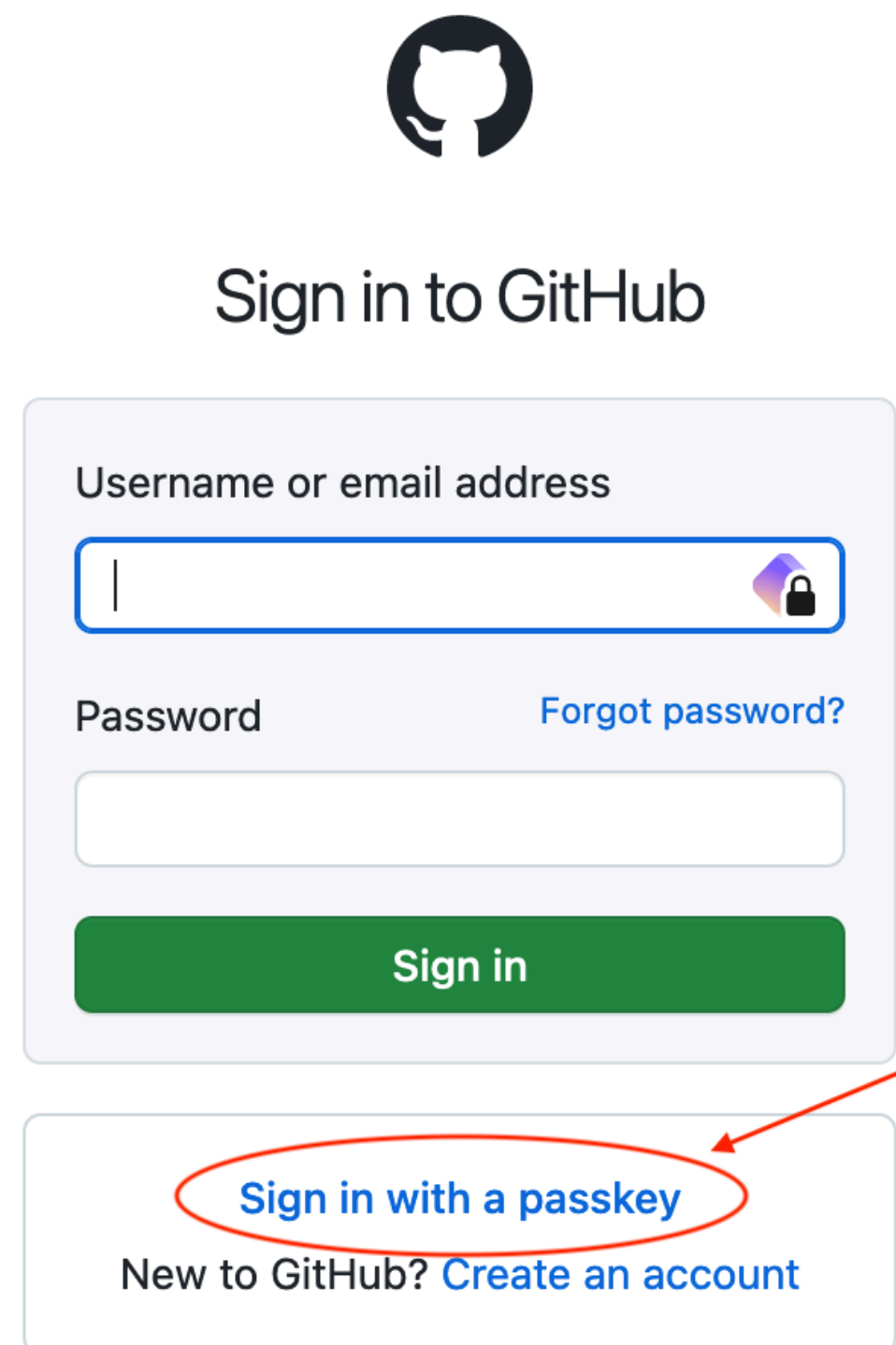
What is WebAuthn?

- WebAuthn = Web Authentication
- Standard written by W3C and FIDO
- Goal: register and authenticate users using public key cryptography

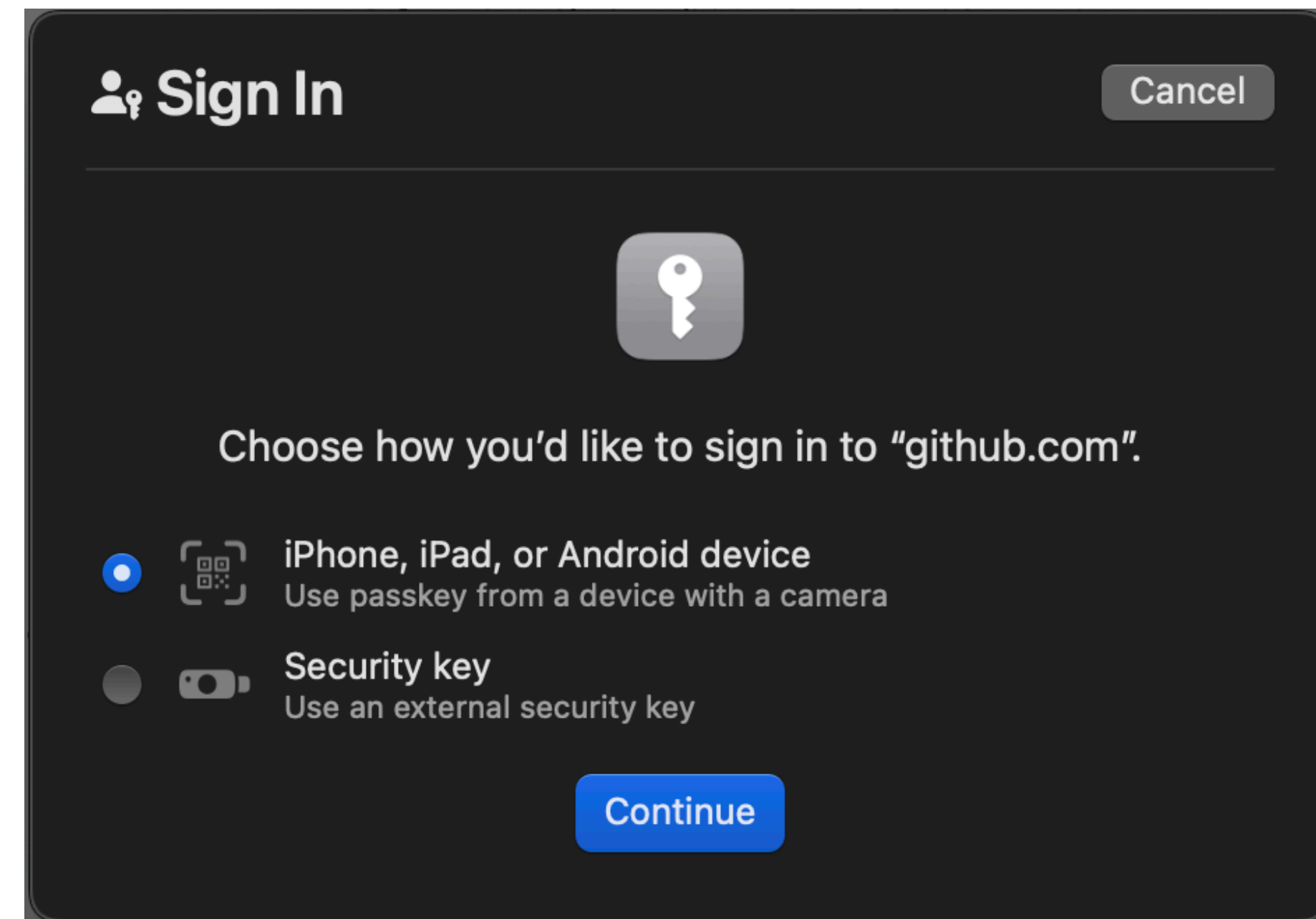
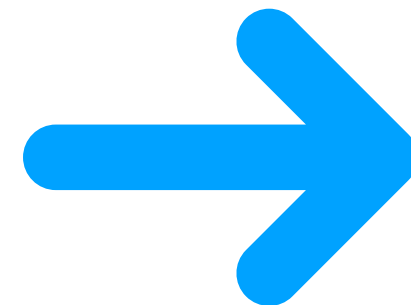
Why use WebAuthn?

- Do away with passwords. They are vulnerable because they are shared secret
- When using WebAuthn the server holds no secrets, only a public key 
- The private key is stored on a user's device 

WebAuthn vs passkeys



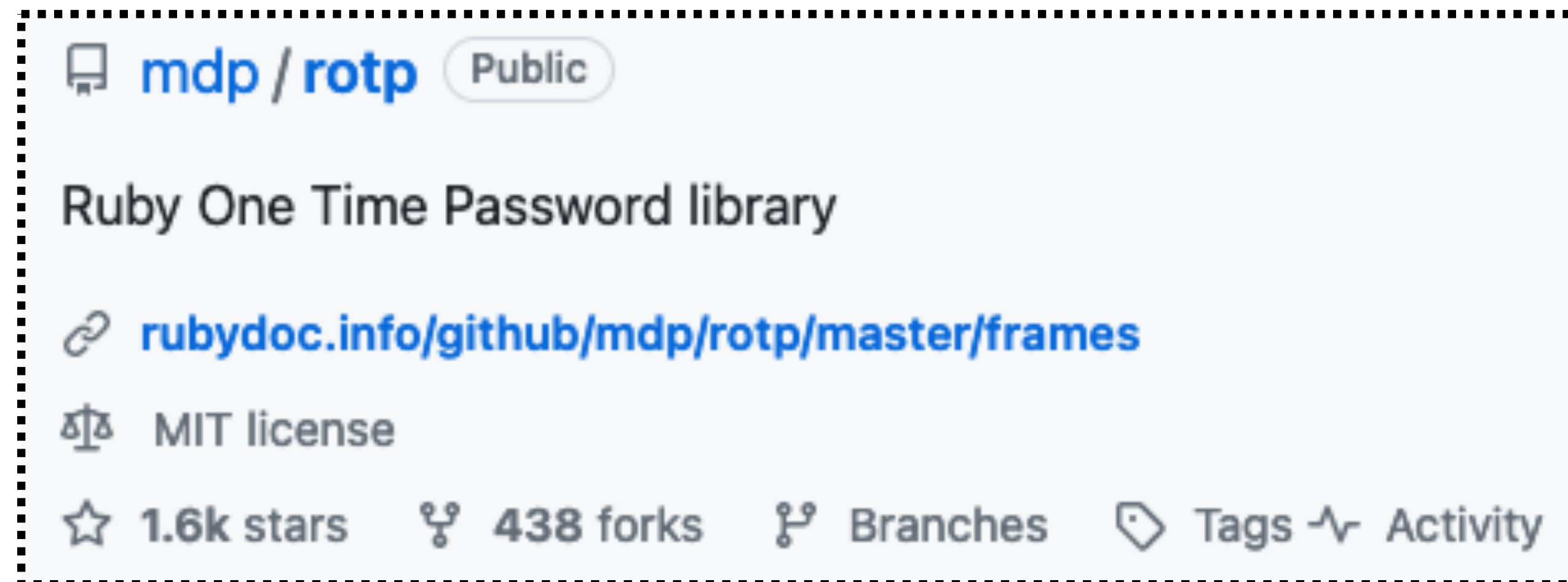
The image shows the GitHub login interface. At the top is the GitHub logo. Below it is the text "Sign in to GitHub". The login form has two input fields: "Username or email address" and "Password". There is a "Forgot password?" link next to the password field. A green "Sign in" button is below the password field. At the bottom of the form, there is a link "Sign in with a passkey" which is circled in red. Below this link is the text "New to GitHub? Create an account".



The image shows a "Sign In" modal. At the top is the text "Sign In" and a "Cancel" button. Below this is a key icon. The text "Choose how you'd like to sign in to 'github.com'." is displayed. There are two options: "iPhone, iPad, or Android device" with a camera icon and the text "Use passkey from a device with a camera", and "Security key" with a key icon and the text "Use an external security key". A blue "Continue" button is at the bottom right.

Passkeys refer to the passwordless implementation of WebAuthn

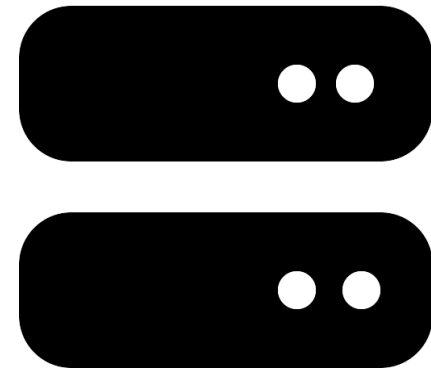
What about OTP?



*“Although this library will continue to be maintained, if you're implementing a 2FA solution today, you should take a look at **Webauthn**. It doesn't involve shared secrets and it's supported by most modern browsers and operating systems.”*

Three ingredients for WebAuthn

Relying Party



Server for application

User Agent



Web browser

Authenticator



Face ID, security key,
fingerprint, PIN, etc.

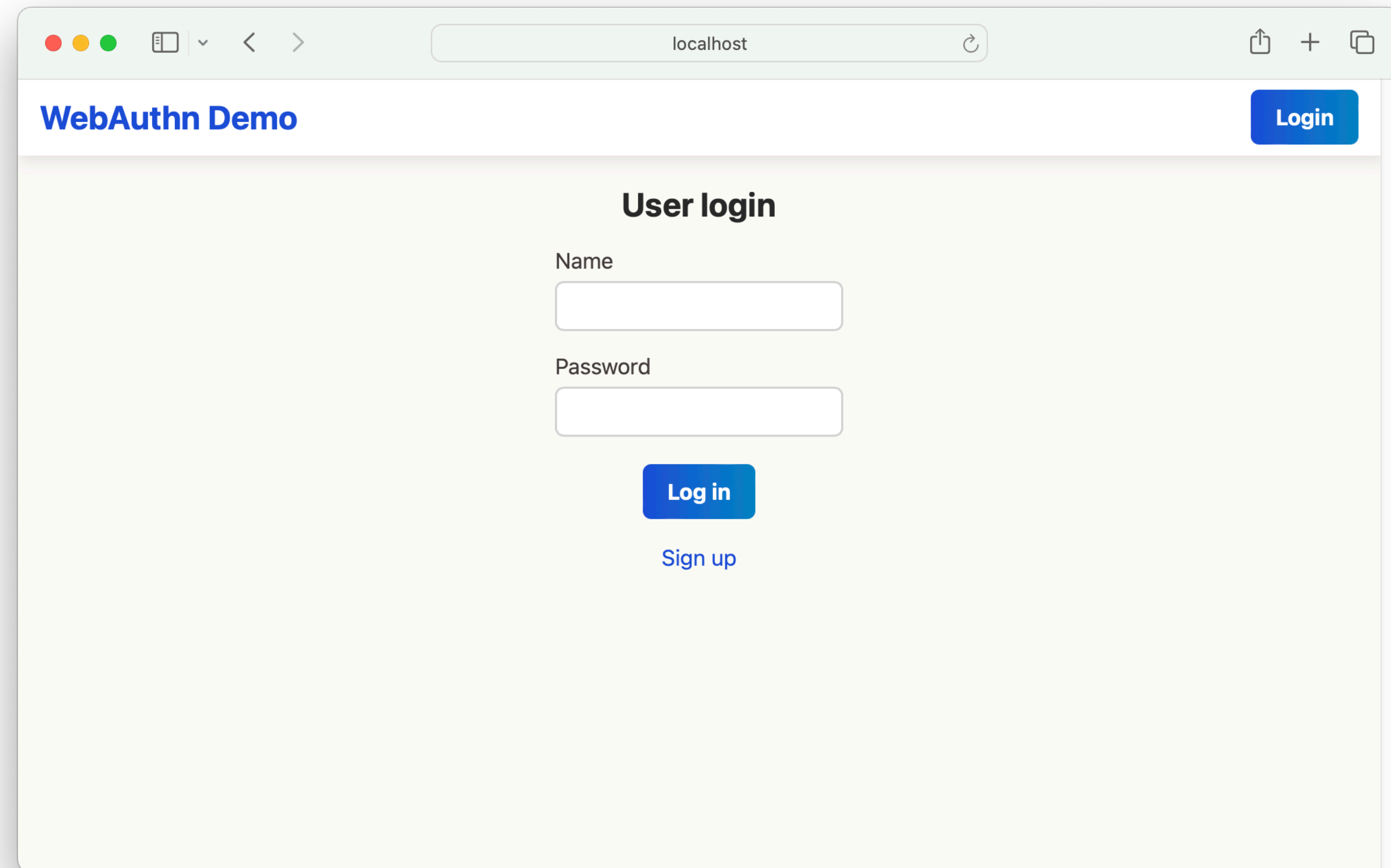
What is a security key?

- Hardware authentication device
- Can be used as replacement for SMS/email one-time passwords or authenticator apps
- More secure than SMS, email or authenticator apps
- Don't lose them! Have a backup



YubiKey security keys by Yubico

Demo time!



The screenshot shows a web browser window with the address bar set to 'localhost'. The page title is 'WebAuthn Demo'. In the top right corner, there is a blue button labeled 'Login'. The main content area has a light yellow background and is titled 'User login'. It contains two input fields: one for 'Name' and one for 'Password'. Below these fields is a blue button labeled 'Log in' and a link labeled 'Sign up'.

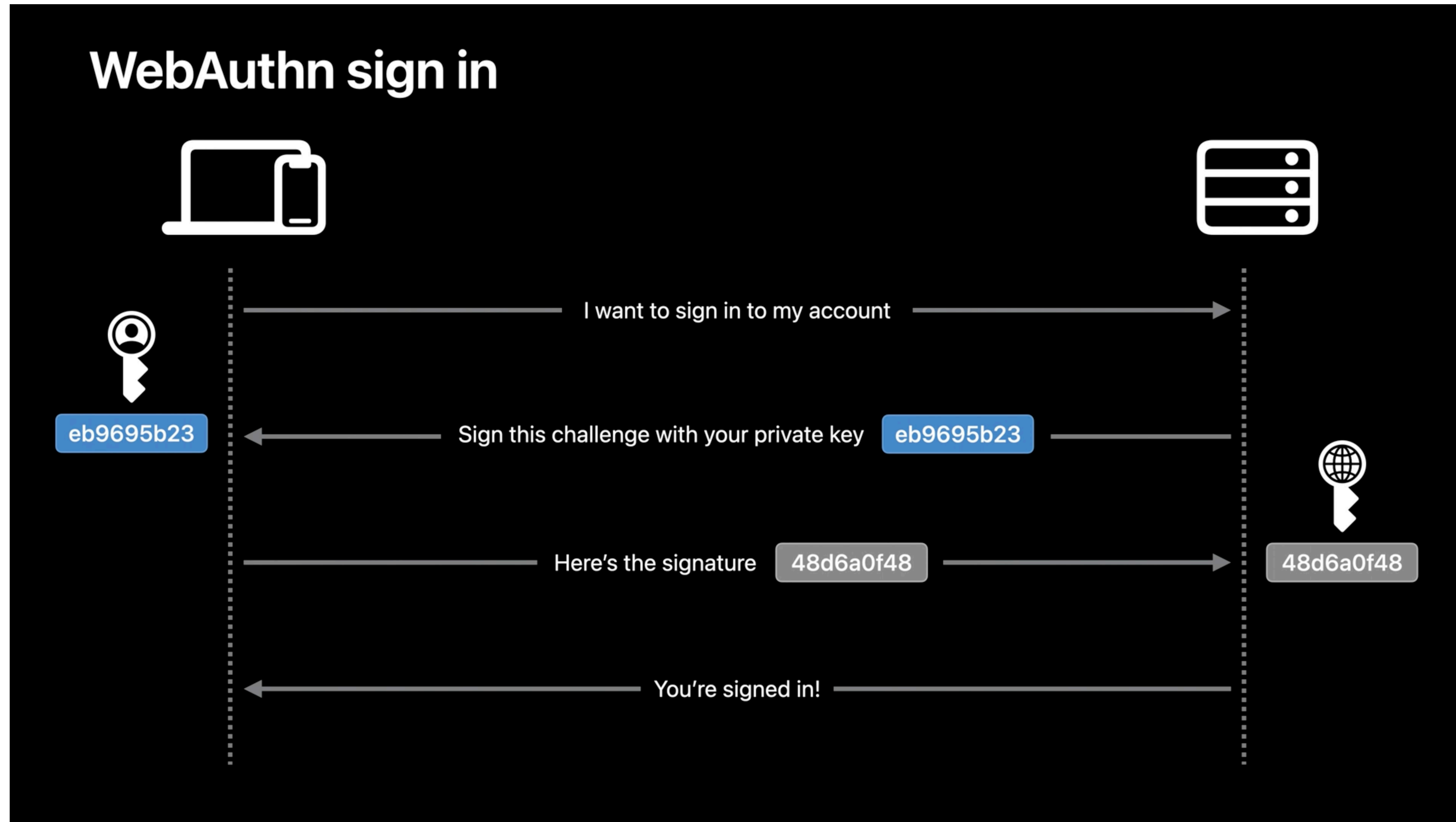
<https://github.com/jp524/webauthn-demo>

How it works

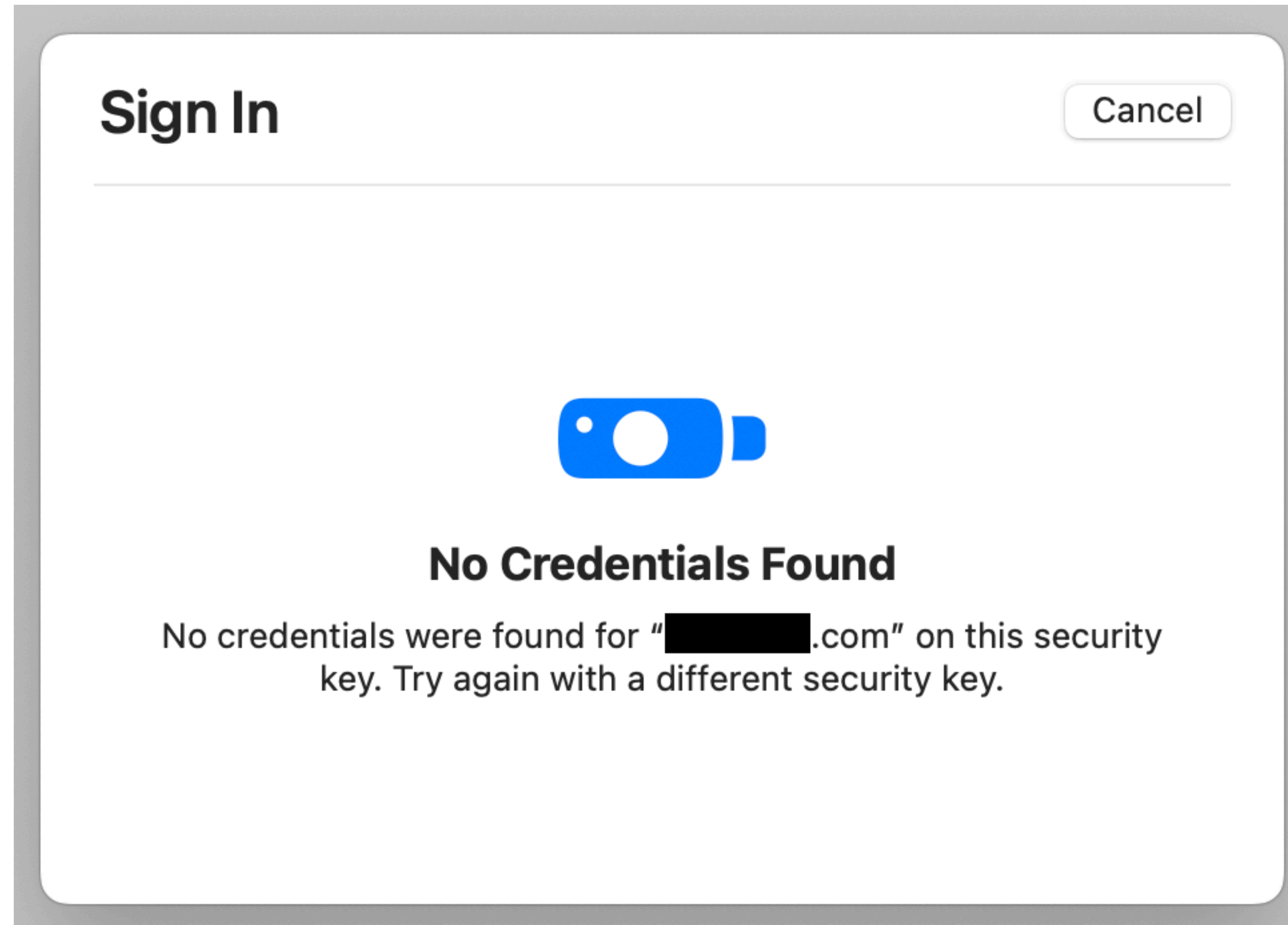
WebAuthn account creation



How it works



A brief note about security



Implementation in Rails

- **webauthn** gem
 - Turns Rails server into Relying Party
 - Rails controllers
- **@github/webauthn-json** NPM package
 - Wrapper for the WebAuthn API
 - Stimulus controllers

Resources

- Introduction to WebAuthn and passkeys
 - <https://webauthn.guide/>
 - <https://fidoalliance.org/passkeys/>
 - https://developer.mozilla.org/en-US/docs/Web/API/Web_Authentication_API
- Introduction to WebAuthn and passkeys from Apple's WWDC: <https://developer.apple.com/videos/play/wwdc2021/10106/>
- Implementation of passkeys for Shopify's Shop Pay: <https://shopify.engineering/supporting-passkeys-in-shop-authentication-flows>

Resources

- WebAuthn gem: <https://github.com/cedarcode/webauthn-ruby>
- WebAuthn JSON package: <https://github.com/github/webauthn-json>
- Implementing WebAuthn in Rails: <https://www.honeybadger.io/blog/multi-factor-2fa-authentication-rails-webauthn-devise/>
- Passwordless demo project by CedarCode: <https://github.com/cedarcode/webauthn-rails-demo-app>