

# From passwords to passkeys: What's new with FIDO?

Joost van Dijk

NLUUG conference 29 Nov 2022



# Overview

- What is a passkey?
- Single-device vs Multi-device credentials
- Device attestation and the FIDO metadata service
- Migrating from passwords to passkeys
  - Conditional mediation
- Cross-device authentication

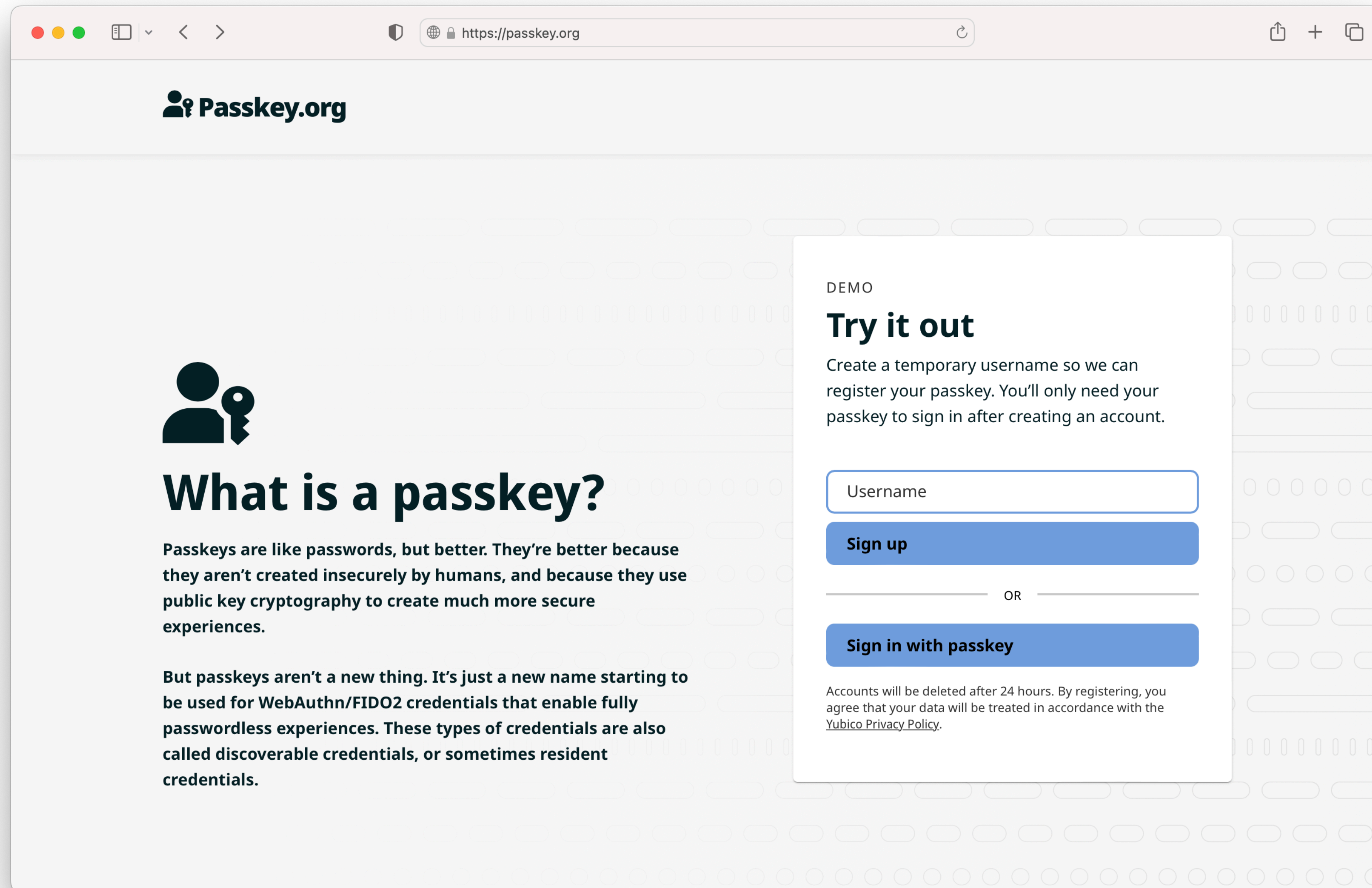
# What is a passkey?

- Passkeys are a more secure alternative to passwords
- More secure, because:
  - Passkeys are resistant to phishing
  - Passkeys have no secrets that can be leaked from servers
  - Passkeys are generated automatically, never reused
- Also easier to use:
  - “Sign in with your face, your finger, or your PIN”
  - Optionally, automatically backed up and synced
- Technically, a passkey is a *FIDO2 credential*



yubico

# Demo



The screenshot shows a web browser window with the URL <https://passkey.org>. The page header features the Passkey.org logo. The main content area is titled "What is a passkey?" and includes a sub-header "Try it out" with a "Sign up" button. A secondary "Sign in with passkey" button is also visible. The background of the page has a faint pattern of passkey icons.

**Passkey.org**

## What is a passkey?

Passkeys are like passwords, but better. They're better because they aren't created insecurely by humans, and because they use public key cryptography to create much more secure experiences.

But passkeys aren't a new thing. It's just a new name starting to be used for WebAuthn/FIDO2 credentials that enable fully passwordless experiences. These types of credentials are also called discoverable credentials, or sometimes resident credentials.

DEMO

### Try it out

Create a temporary username so we can register your passkey. You'll only need your passkey to sign in after creating an account.

Username

**Sign up**

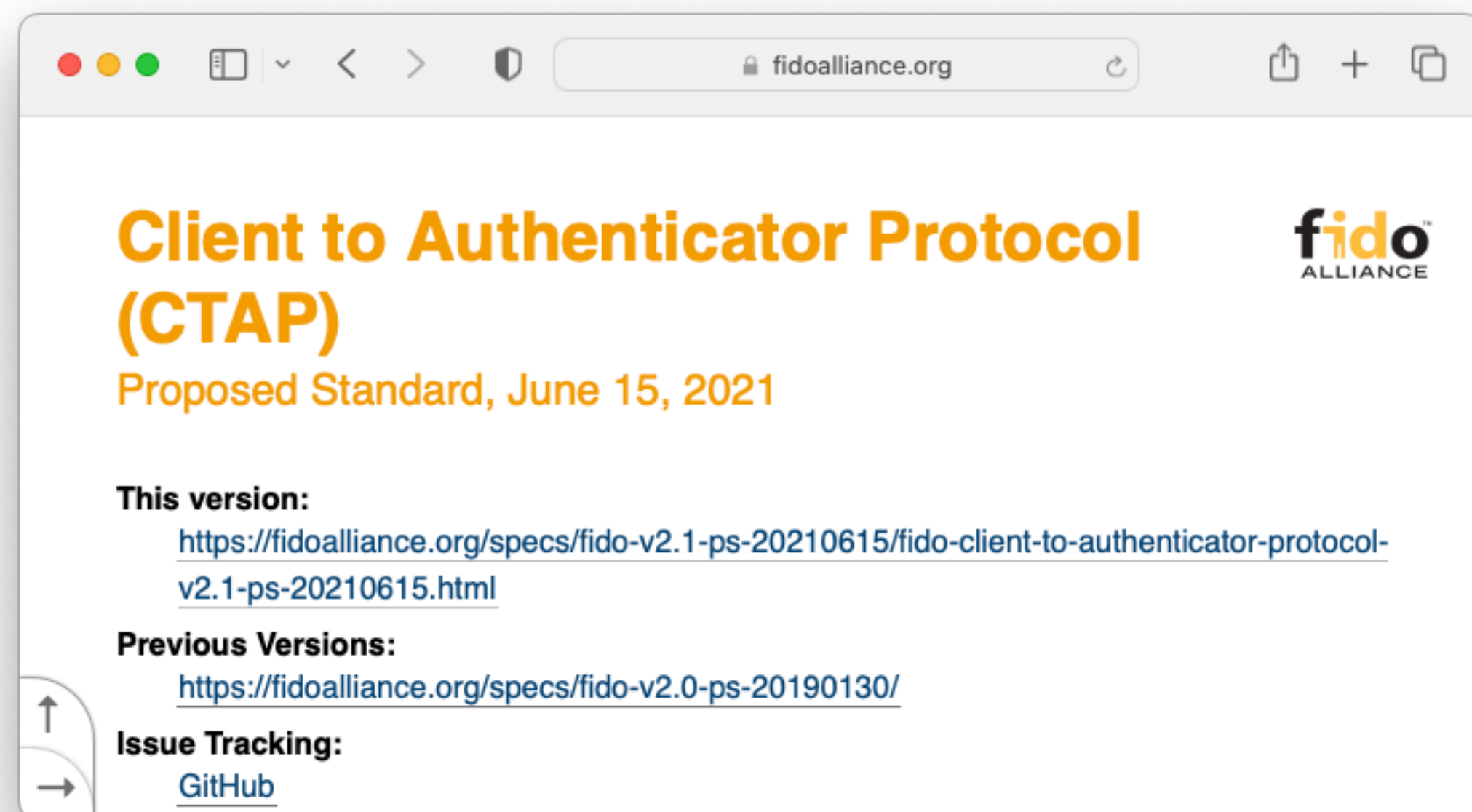
OR

**Sign in with passkey**

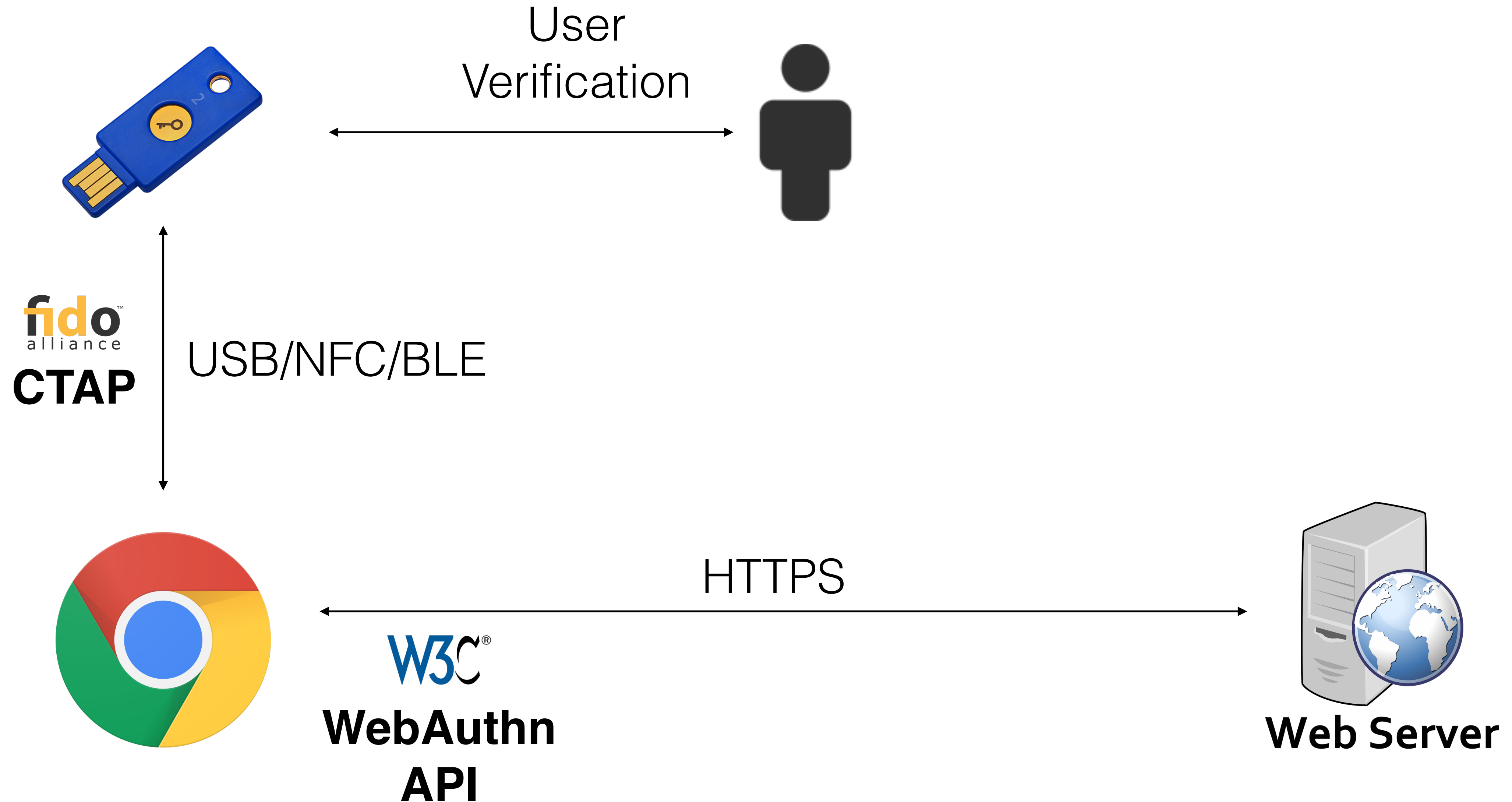
Accounts will be deleted after 24 hours. By registering, you agree that your data will be treated in accordance with the [Yubico Privacy Policy](#).

# FIDO2

- Specifications:
  - CTAP - using a FIDO authenticator from a client (e.g. a browser)
  - Webauthn: API for using FIDO credentials in web applications

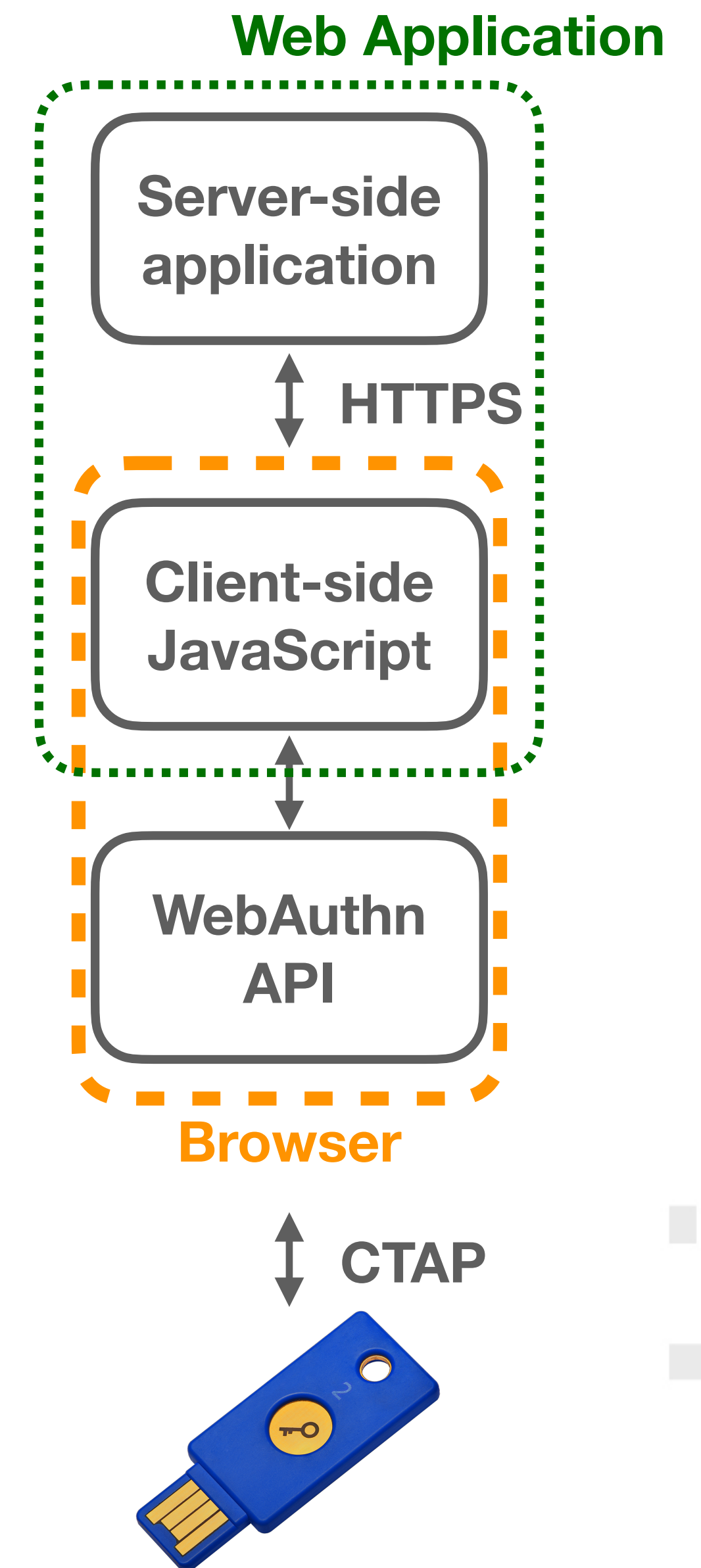


# CTAP vs Webauthn



# Webauthn: JavaScript API

- `navigator.credentials.create()`  
register new FIDO credential
- `navigator.credentials.get()`  
authenticate using a previously registered credential



# Challenge/Response authentication

(not FIDO)



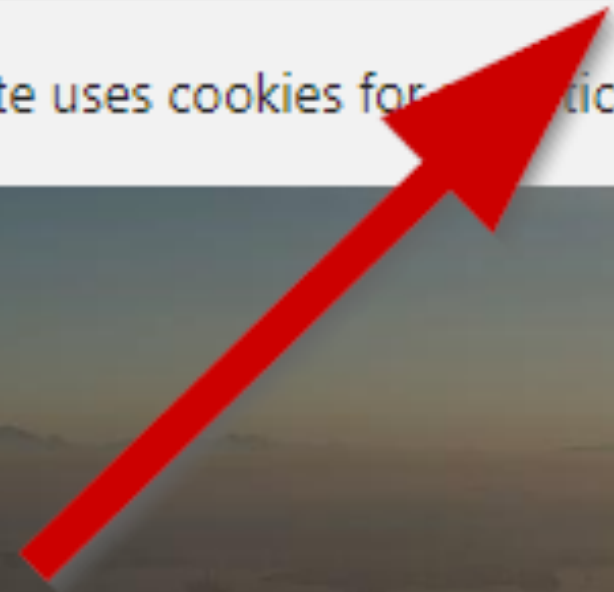
$\text{response} = \text{sign}(k, \text{challenge})$

$\text{result} = \text{verify}(p, \text{response}, \text{challenge})$

yubico



This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. [Learn more](#)



## Sign in

Email, phone, or Skype

---

No account? [Create one!](#)

Next

# Phishing resistance

(Simplified)

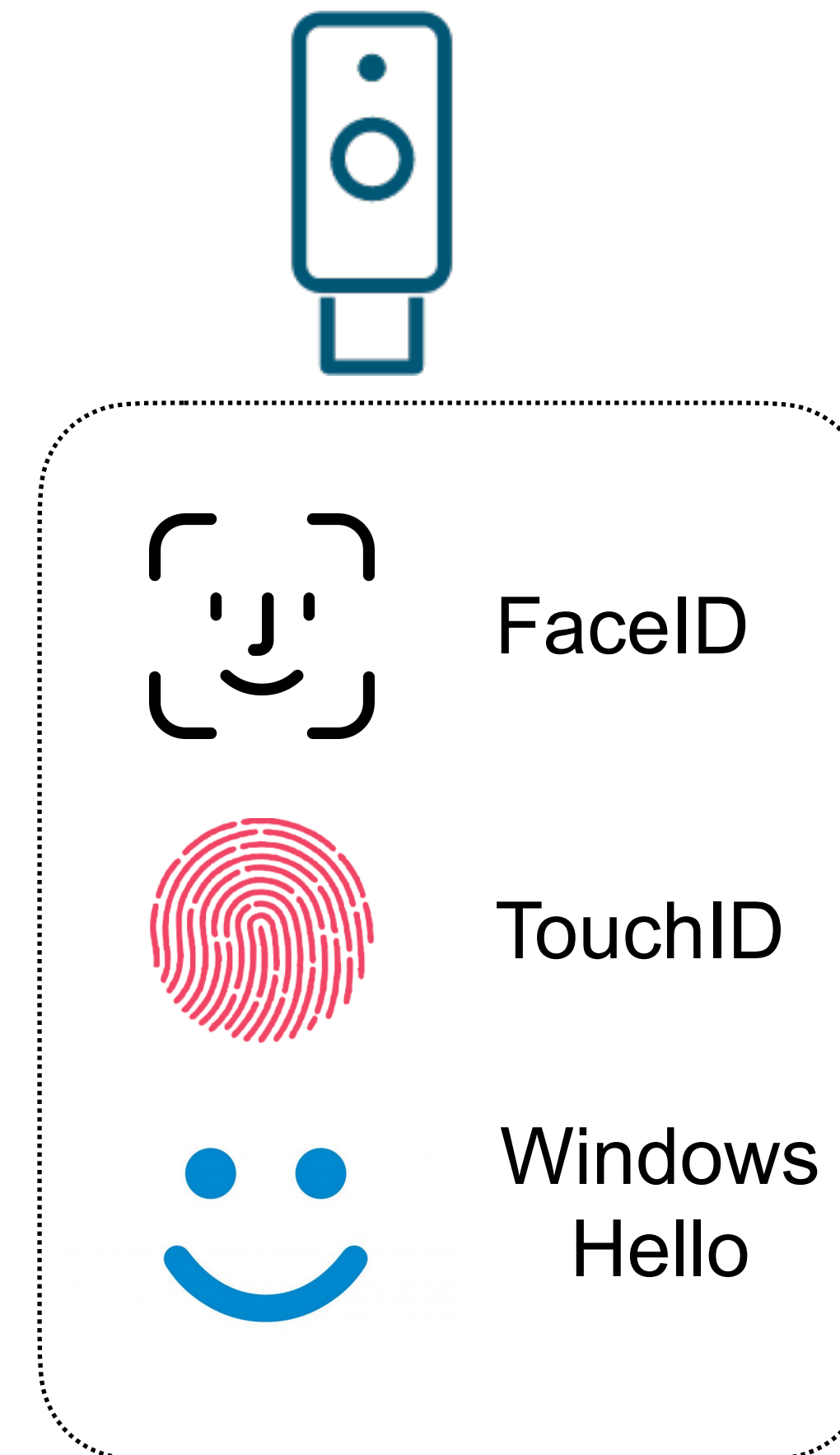


$\text{response} = \text{sign}(k, \text{challenge})$   
**+origin**

$\text{result} = \text{verify}(p, \text{response}, \text{challenge})$   
**+origin**

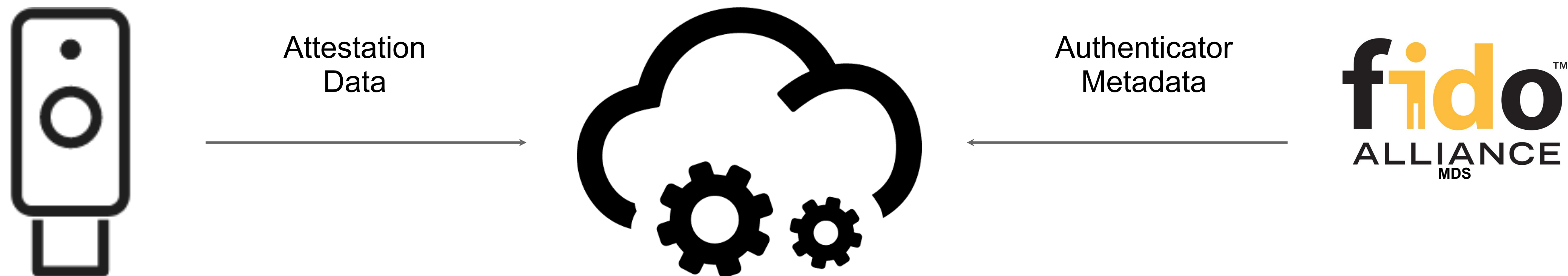
# Authenticators

- *Roaming Authenticator*  
example: a USB security key
- *Platform Authenticator*  
example: a built-in fingerprint sensor
- Authenticators can store multiple FIDO credentials (passkeys)
- Roaming authenticators can use different *transports*: USB, NFC, BLE



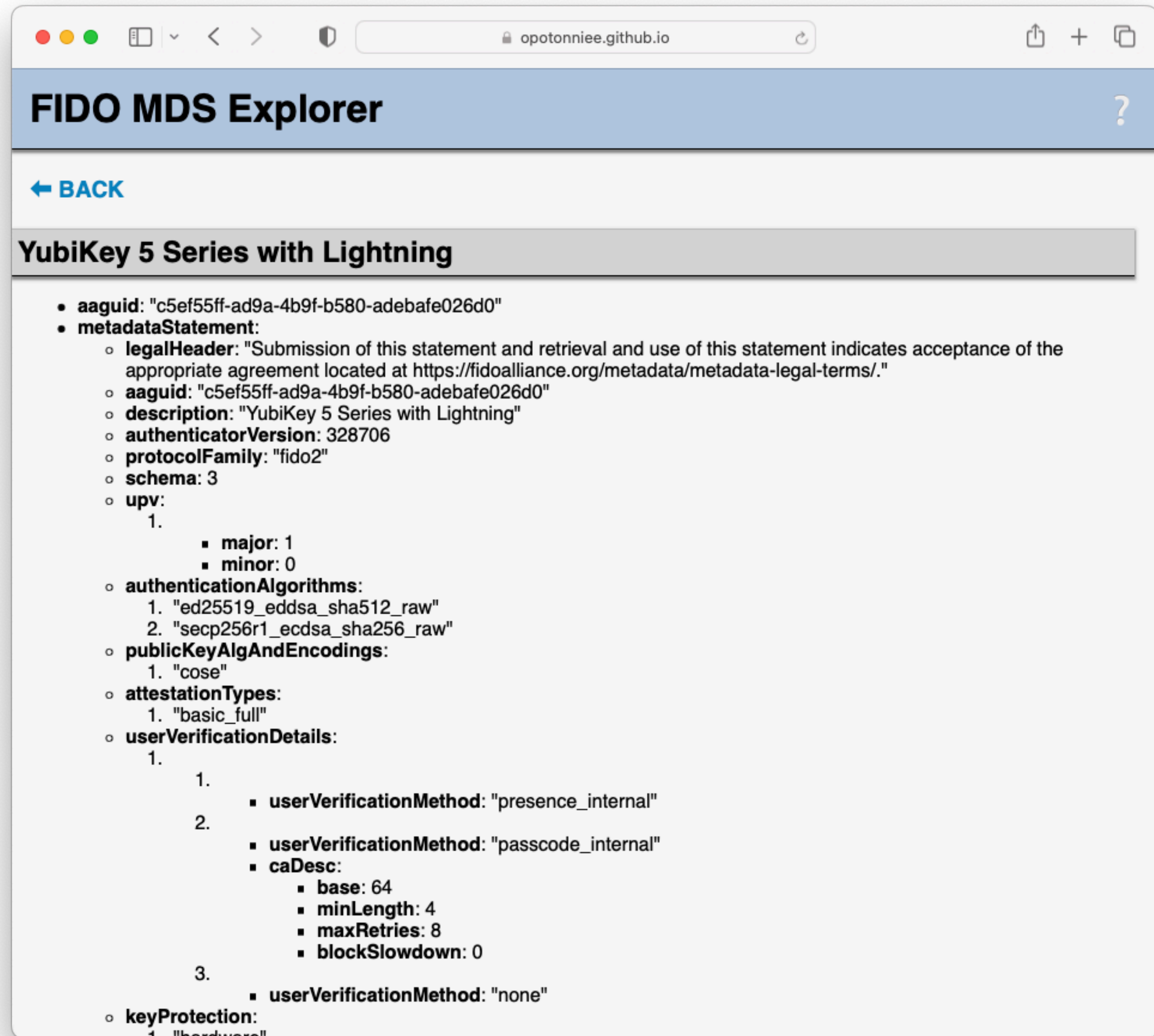
# Attestation and Metadata

- Attestation provides verifiable evidence as to the authenticator's origin
- Based on a hardware attestation key and certificate
- Use FIDO Alliance Metadata Service to determine provenance
- Implement Allow/Deny lists to filter Authenticators
- Typically used in high-assurance (enterprise) use cases



# Metadata example

- `aaguid`  
(Authenticator unique ID)
- `keyProtection`  
e.g. `secure_element`
- `transports`  
e.g. `usb`
- `status`  
(certification level)



The screenshot shows a web browser window with the address bar containing `opotonniee.github.io`. The page title is "FIDO MDS Explorer". Below the title is a "BACK" button. The main content area displays the metadata for a "YubiKey 5 Series with Lightning". The metadata is structured as follows:

- `aaguid`: "c5ef55ff-ad9a-4b9f-b580-adebaf026d0"
- `metadataStatement`:
  - `legalHeader`: "Submission of this statement and retrieval and use of this statement indicates acceptance of the appropriate agreement located at <https://fidoalliance.org/metadata/metadata-legal-terms/>."
  - `aaguid`: "c5ef55ff-ad9a-4b9f-b580-adebaf026d0"
  - `description`: "YubiKey 5 Series with Lightning"
  - `authenticatorVersion`: 328706
  - `protocolFamily`: "fido2"
  - `schema`: 3
  - `upv`:
    1.
      - `major`: 1
      - `minor`: 0
  - `authenticationAlgorithms`:
    1. "ed25519\_eddsa\_sha512\_raw"
    2. "secp256r1\_ecdsa\_sha256\_raw"
  - `publicKeyAlgAndEncodings`:
    1. "cose"
  - `attestationTypes`:
    1. "basic\_full"
  - `userVerificationDetails`:
    1.
      - `userVerificationMethod`: "presence\_internal"
    2.
      - `userVerificationMethod`: "passcode\_internal"
      - `caDesc`:
        - `base`: 64
        - `minLength`: 4
        - `maxRetries`: 8
        - `blockSlowdown`: 0
    3.
      - `userVerificationMethod`: "none"
  - `keyProtection`:
    1. "hardware"

# FIDO2 UX issues

1. How to recover from device loss?
2. How to simplify sign in during transition from passwords to passkeys?
3. How to bootstrap a new device?

# Recover from device loss

- Buy a backup security key
  - And register twice for every RP?
- Leave it to the RP to recover
  - Often fallback to password (or other knowledge-factor)
- New approach:
  - Instead of a backup security key, backup FIDO credentials in the cloud: *Multi-Device Credentials*

# Multi-device credentials

- Passkeys that can be synchronised via a cloud provider
- Backed up and synced across devices
- Similar to a cloud-synced password manager
  
- No need to re-enroll a new device on every account!
- Synced across *devices* but not across *ecosystems*  
(i.e. a passkey synced via iCloud is not available on Android)

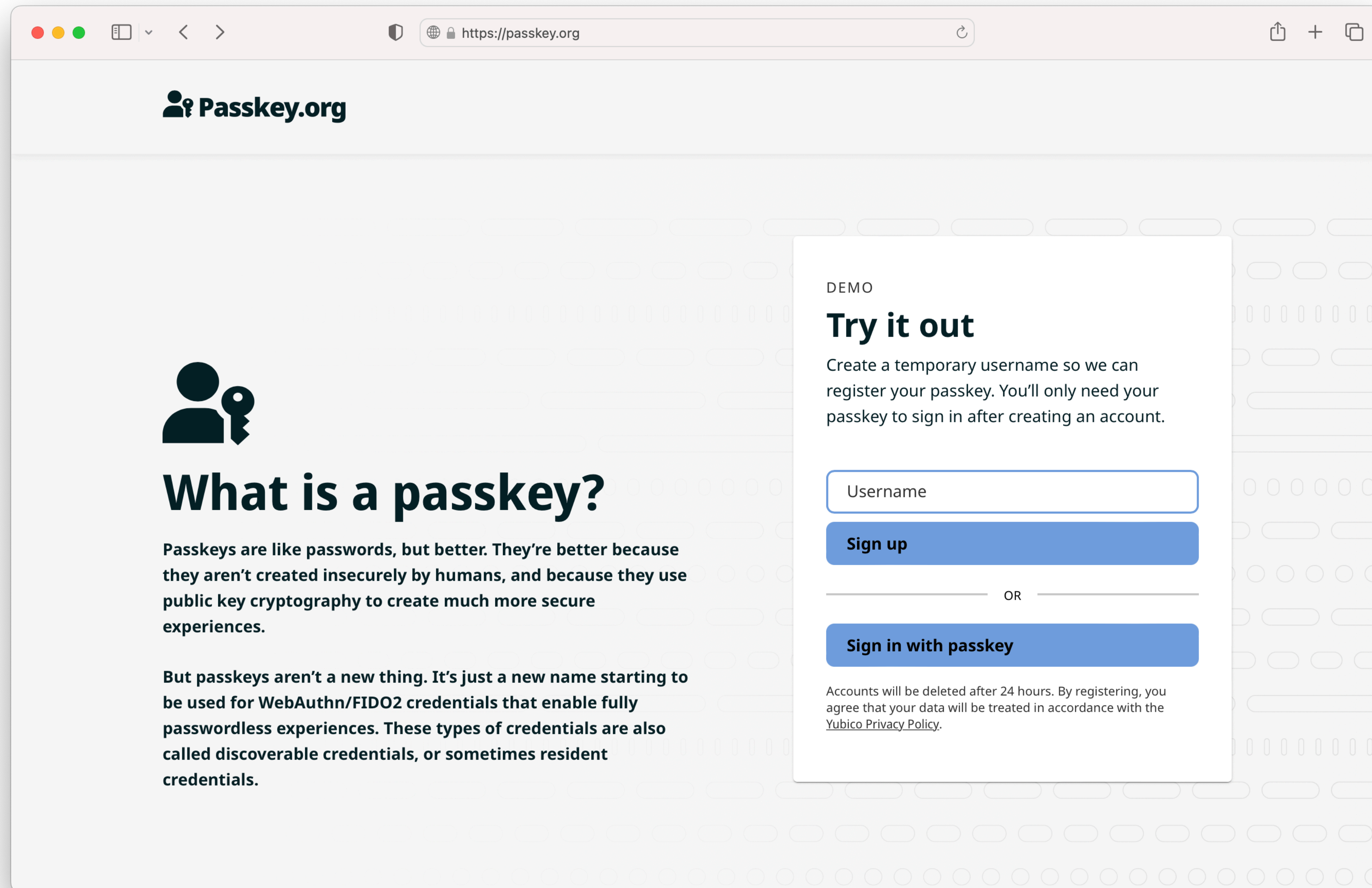


# Single vs Multi device passkeys

- Single-device passkey
  - Hardware-bound
  - Hardware attestation
  - Ideal for high assurance use cases
  - Example: FIDO2 credential stored on a security key
- Multi-device passkey
  - Copyable
  - Syncable
  - Ideal for low assurance use cases
  - Example: FIDO2 credential stored on an iOS device



# Demo



The screenshot shows a web browser window with the URL <https://passkey.org>. The page header features the Passkey.org logo. The main content area is titled "What is a passkey?" and includes a sub-header "Try it out" with a "Sign up" button. Below the sign-up button is an "OR" separator and a "Sign in with passkey" button. A disclaimer at the bottom of the form states that accounts will be deleted after 24 hours and that users agree to the Yubico Privacy Policy.

**Passkey.org**

## What is a passkey?

Passkeys are like passwords, but better. They're better because they aren't created insecurely by humans, and because they use public key cryptography to create much more secure experiences.

But passkeys aren't a new thing. It's just a new name starting to be used for WebAuthn/FIDO2 credentials that enable fully passwordless experiences. These types of credentials are also called discoverable credentials, or sometimes resident credentials.

DEMO

### Try it out

Create a temporary username so we can register your passkey. You'll only need your passkey to sign in after creating an account.

Username

**Sign up**

OR

**Sign in with passkey**

Accounts will be deleted after 24 hours. By registering, you agree that your data will be treated in accordance with the [Yubico Privacy Policy](#).

yubico

# Apple support for passkeys

- iOS 16, released September 12, 2022
- iPadOS 16.1, released October 24, 2022
- MacOS 13 (Ventura), released October 24, 2022
  
- Stored in *Apple iCloud*
- Apple lets users share passkeys via AirDrop
- [About the security of passkeys](#)

# Google support for passkeys

- Google Play Services (November 2022)
- Required Android 9 or newer
- Stored in *Google Password Manager*
- Google currently does not let users share passkeys
- [Security of Passkeys in the Google Password Manager](#)

# Deployment considerations

**Low assurance**

**High assurance**




# Conditional Mediation

- When transitioning from passwords to passkeys, need support for both
- Users may not remember if they registered a passkey before
- Users need to choose:  
*password or passkey flow?*

**Sign In**

Email Address




Show password


**Password**


[Forgot your password?](#)

**Sign In**

or

 **Sign In with a Passkey**

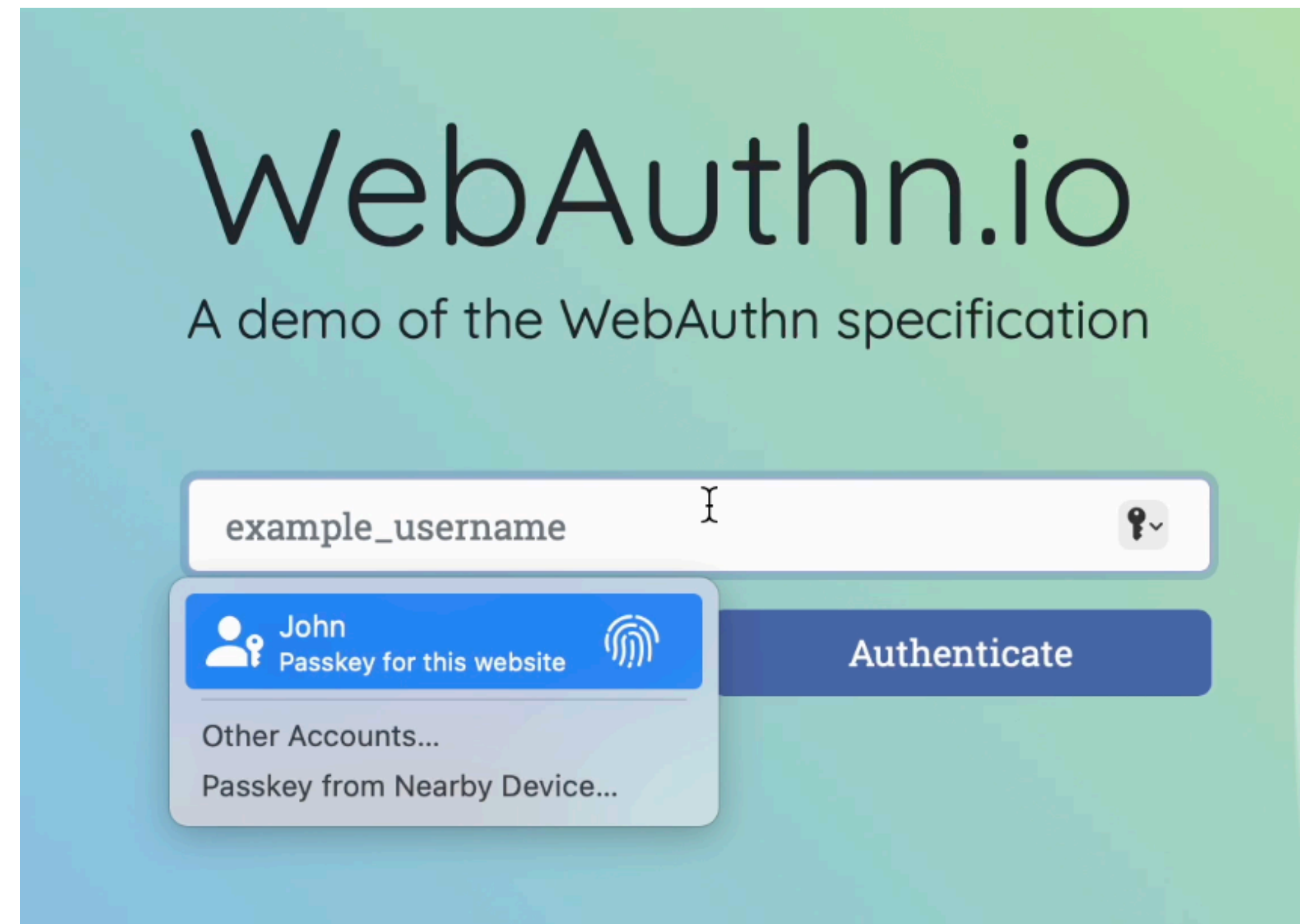
 **Sign In with Apple**

 **Sign In with Google**

# Conditional Mediation

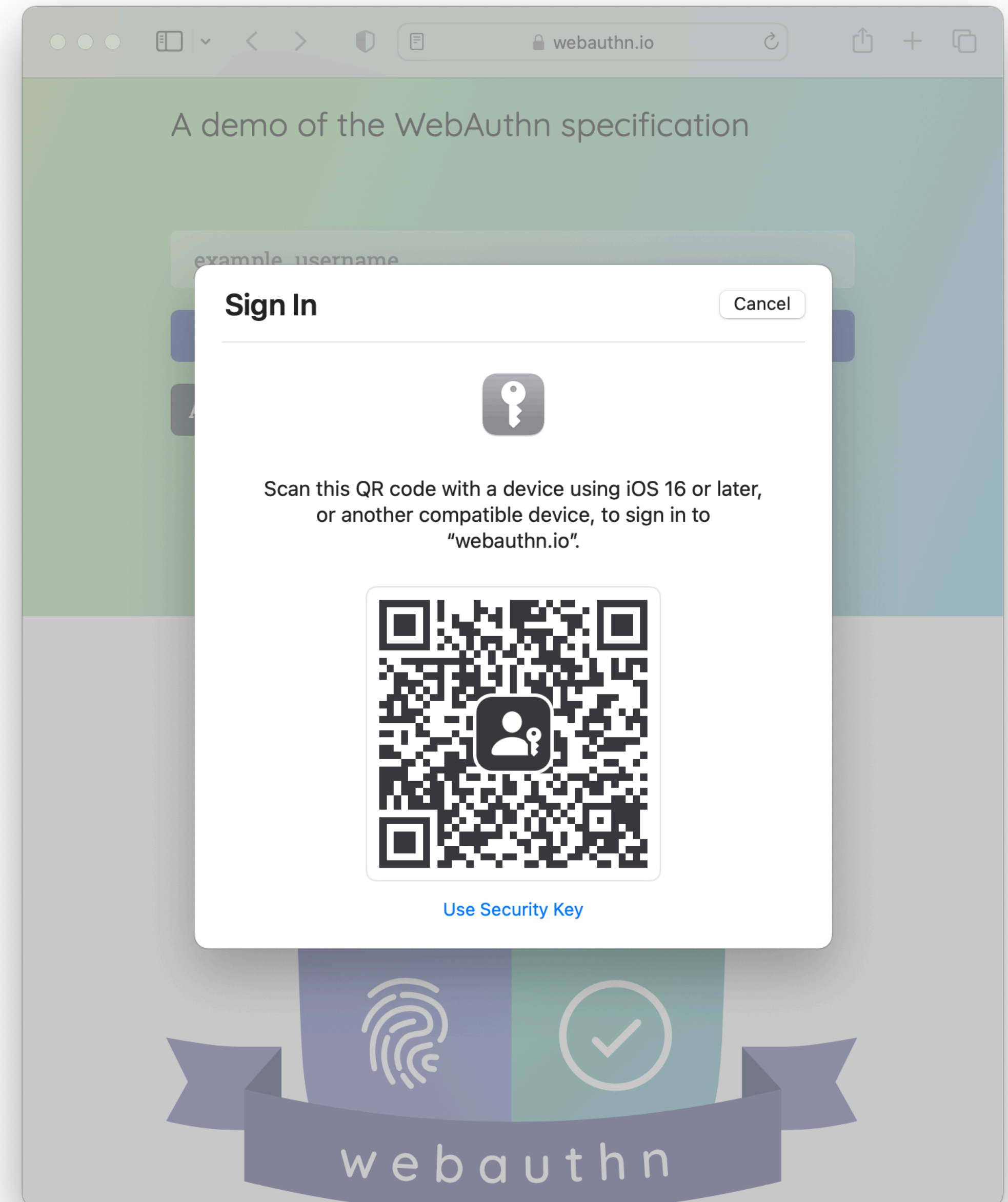
- Help users decide whether to use a password or a passkey
- Similar to a password manager popup
- A.k.a. “*conditional UI*” and “*Autofill UI*”
- Not implemented everywhere yet!
- Detect support with:  
`PublicKeyCredential.isConditionalMediationAvailable();`
- Use autocomplete attribute on your username/password form:

```
<input type="text"  
      id="username-field"  
      autoComplete="username webauthn" />
```



# Cross-device authentication

- Use case:
  - Bootstrap a new device with a passkey on a different device
  - After sign in: register passkey on new device
- Examples:
  - Use your Android phone to sign in on macOS using Safari
  - Use your iOS phone to sign in on Windows using Chrome
- Protocol: Hybrid  
formerly caBLE (Cloud Assisted BLE)

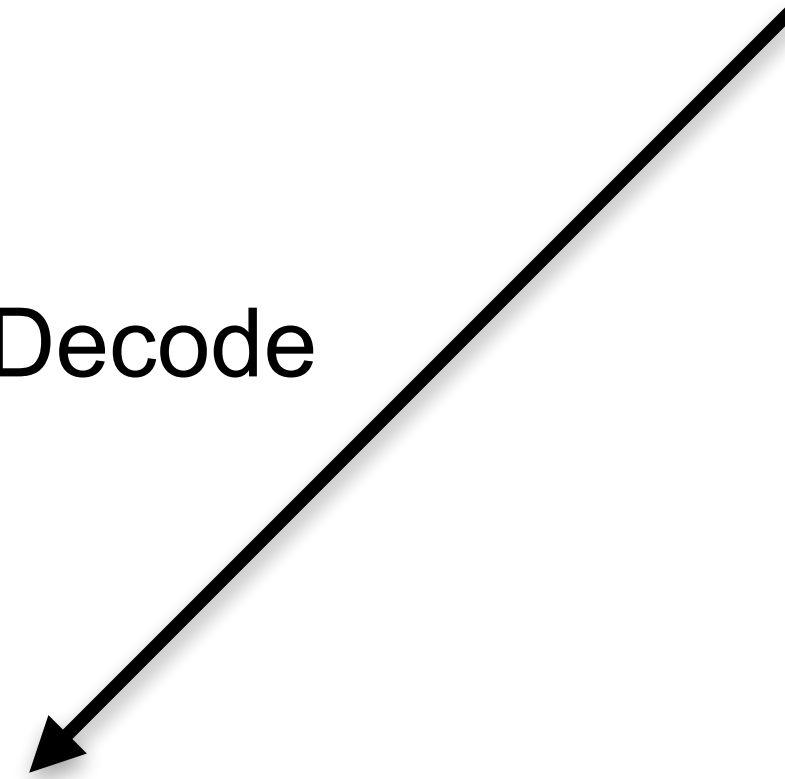




# Hybrid (caBLE)



Decode



```
FIDO:/  
2843228462697693453452592315025563653700648  
2906344371846880730128631610022222478634152  
3013544215784712235817551315763050072315363  
400833845326346315498242109321447142660
```

Decode



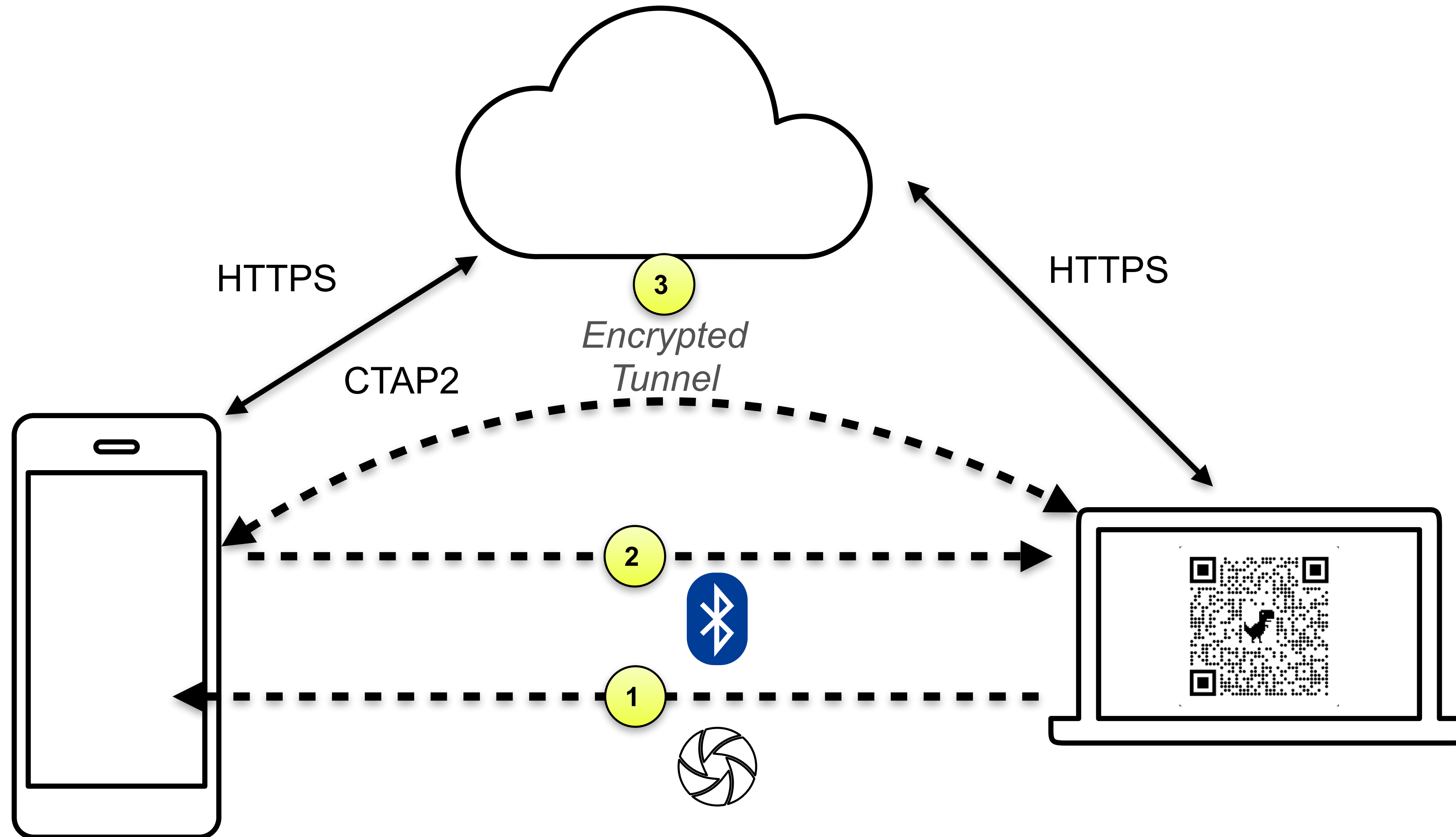
```
{  
  0: h'0303659bc84279d9e6bd13a11445ba3de8f7a0237c5b40ff77e497ab32b9d8082d',  
  1: h'b2c251f13fcc397bc753121d7953b491',  
  2: 2,  
  3: 1662126241_2,  
  4: true,  
  5: "mc",  
}
```

Public key

Random secret



# Hybrid (caBLE)



# Device Public Key extension

- (Draft) Webauthn extension
- Per RP device-bound key (in addition to passkey)
- Implemented by platform authenticators
- Signal to RP: “new device” or “same device as last time”
- Example use case:  
Require extra factor when a synced key is used for the first time.
- currently: Android beta

# Platform/browser Support

The screenshot shows the 'passkeys.dev' website with a navigation bar containing 'Docs', 'Device Support', and 'About'. A search bar is present with the text 'Search docs...' and a 'Ctrl + /' shortcut. The main content is a table with columns for 'Capability', 'Android', 'Chrome OS', 'iOS/iPad OS', 'macOS', 'Ubuntu', and 'Windows'. The table lists support for 'Passkeys', 'Single-device passkeys', 'Browser Autofill UI', 'Cross-Device Authentication Authenticator', and 'Cross-Device Authentication Client'. Support is indicated by green checkmarks, red X marks, or calendar icons for 'Planned' status. Some entries include version requirements like 'Android 9+', 'iOS 16+', and 'macOS 13+'.

Capability	Android	Chrome OS	iOS/iPad OS	macOS	Ubuntu	Windows
<b>Passkeys</b>	✓ Android 9+	📅 Planned	✓ iOS 16+	✓ macOS 13+ <sup>2</sup>	✗ Not Supported	📅 Planned
<b>Single-device passkeys</b> ⓘ	✗ Not Supported	✗ Not Supported	📱 security keys only	📱 security keys only	📱 security keys only	✓
<b>Browser Autofill UI</b>	✓ Chrome	📅 Planned	✓ Safari	✓ Safari	✗ Not Supported	📅 Chrome <sup>1</sup> Edge
	📅 Edge		✗ Edge Chrome Firefox	📅 Chrome Edge		✗ Firefox
	✗ Firefox			✗ Firefox		
<b>Cross-Device Authentication Authenticator</b> ⓘ	✓ Android 9+	✗ Not Supported	✓ iOS 16+	✗ Not Supported	✗ Not Supported	✗ Not Supported
<b>Cross-Device Authentication Client</b> ⓘ	📅 Planned	✓	✓ iOS 16+	✓ macOS 13+	✓ Chrome Edge	✓ Chrome Edge

# Resources

- More Information
  - <https://fidoalliance.org/passkeys/>
  - <https://passkeys.dev/>
  - <https://passkey.org/> (launch Nov/Dec)
  - <https://github.com/herrjemand/awesome-webauthn>
- Developer documentation:
  - <https://developers.yubico.com/Passkeys/>
  - [https://developers.yubico.com/WebAuthn/Concepts/Passkey\\_Autofill/](https://developers.yubico.com/WebAuthn/Concepts/Passkey_Autofill/)
- Open Source Software:
  - <https://github.com/Yubico/java-webauthn-server/>  
(v2.2 support for passkeys, released Nov 24)
  - <https://github.com/YubicoLabs/WebAuthnKit>

# Resources

- Passkey backup security:
  - <https://support.apple.com/en-us/HT213305>
  - <https://security.googleblog.com/2022/10/SecurityofPasskeysintheGooglePasswordManager.html>
- FIDO Metadata Service
  - <https://fidoalliance.org/metadata/>
  - <https://opotonniee.github.io/fido-mds-explorer/>
- Multi-Device-Credential support
  - <https://github.com/passkeydeveloper/discussions/wiki/Known-Passkey-Support> (passkey support using best practices)
  - <https://passkeys.directory/> (passkey support but some entries with issues)

# Questions?