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

Joost van Dijk

NLUUG conference 29 Nov 2022

yubico

Overview

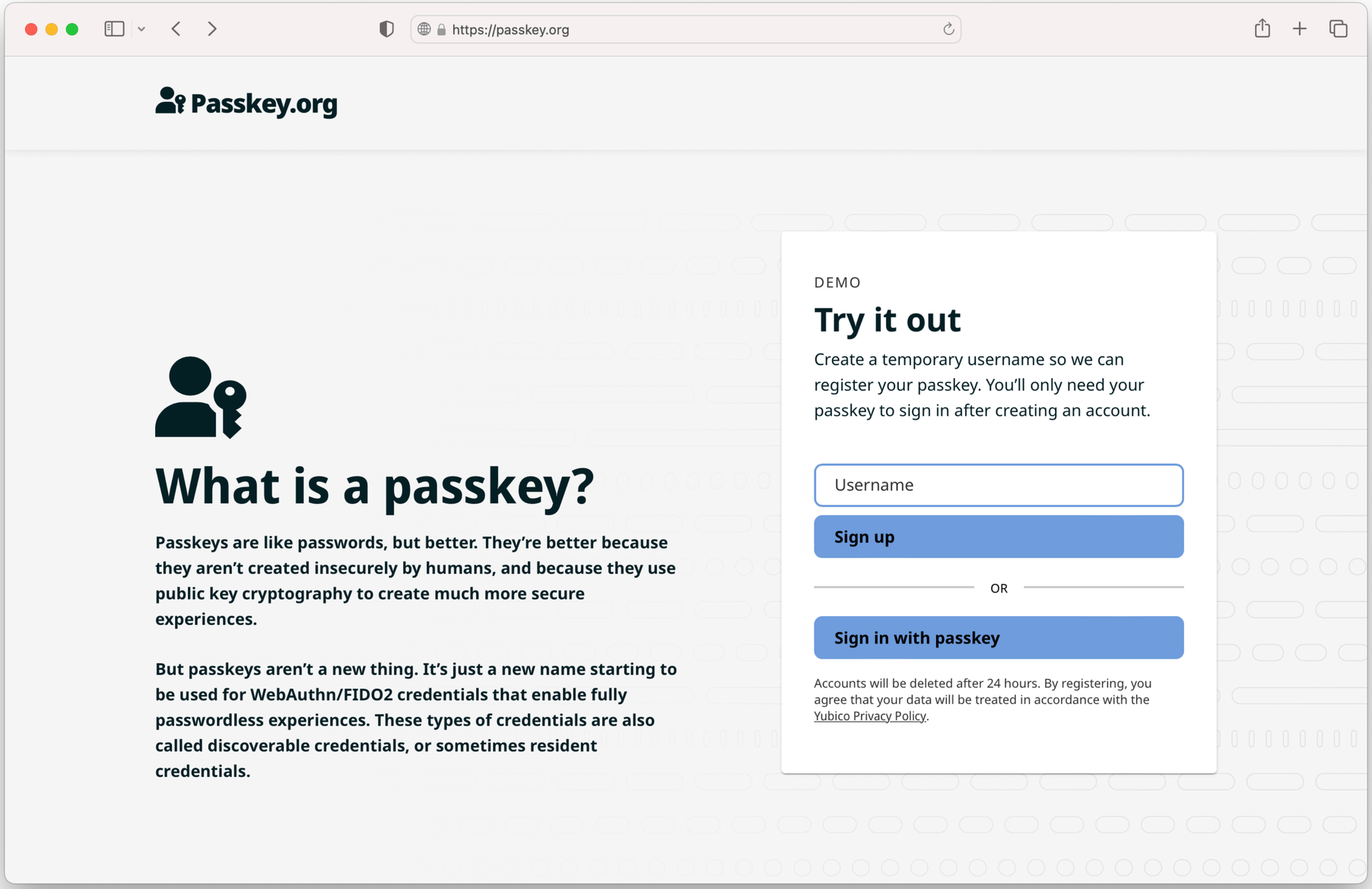
- What is a passkey?
- Single-device vs Multi-device credentials
- Device attestation and the FIDO metadata service
- Migrating from passwords to passkeys
  - Conditional mediation
- Cross-device authentication

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

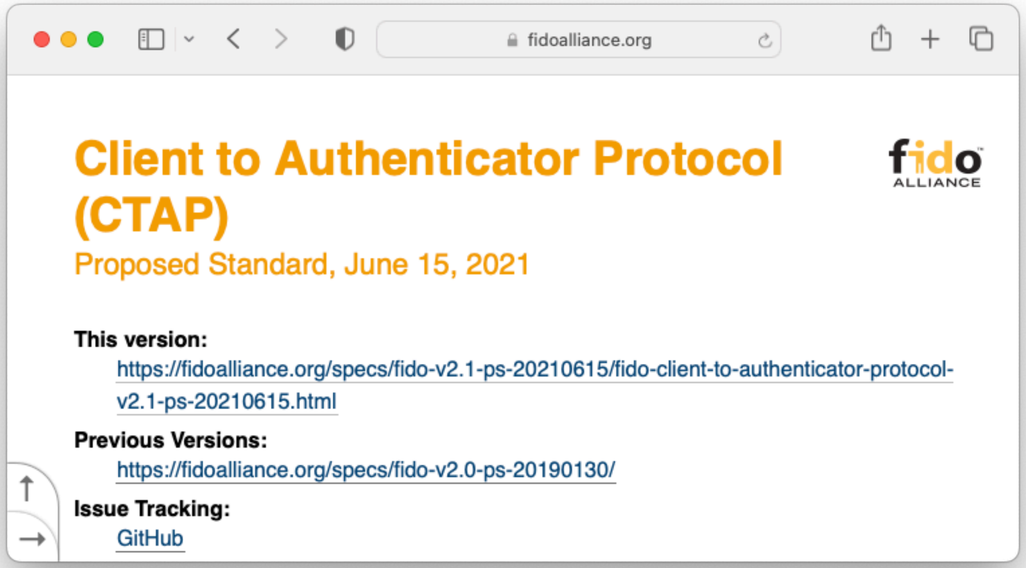
Demo



yubico

FIDO2

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



yubico

CTAP vs Webauthn

User
Verification

fido alliance
CTAP

USB/NFC/BLE

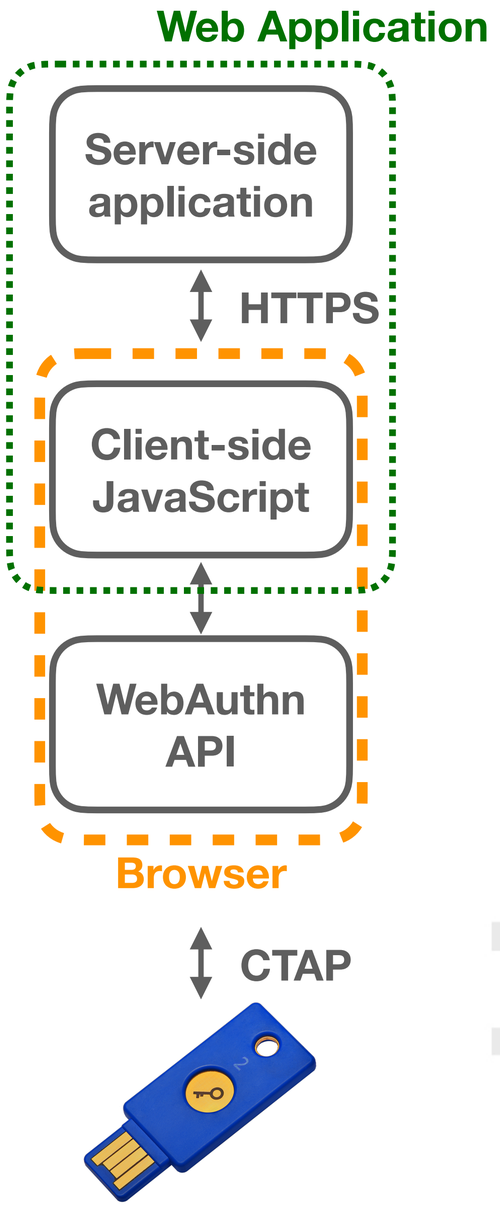
HTTPS

W3C®
WebAuthn API

Web Server

Webauthn: JavaScript API

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



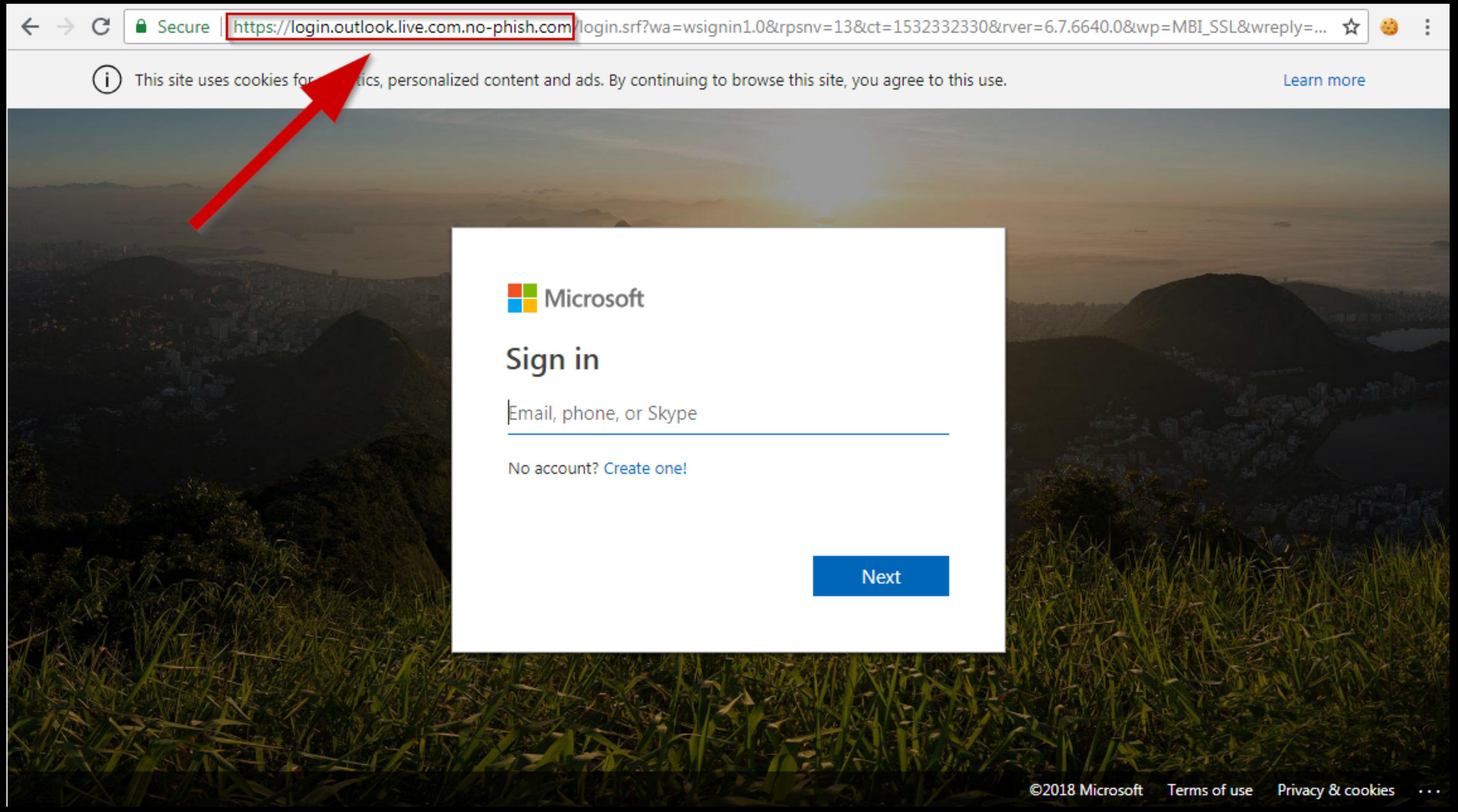
yubico

Challenge/Response authentication

(not FIDO)

response = sign(k, challenge)

result = verify(p, response, challenge)



https://breakdev.org/evilginx-2-next-generation-of-phishing-2fa-tokens/

Phishing resistance

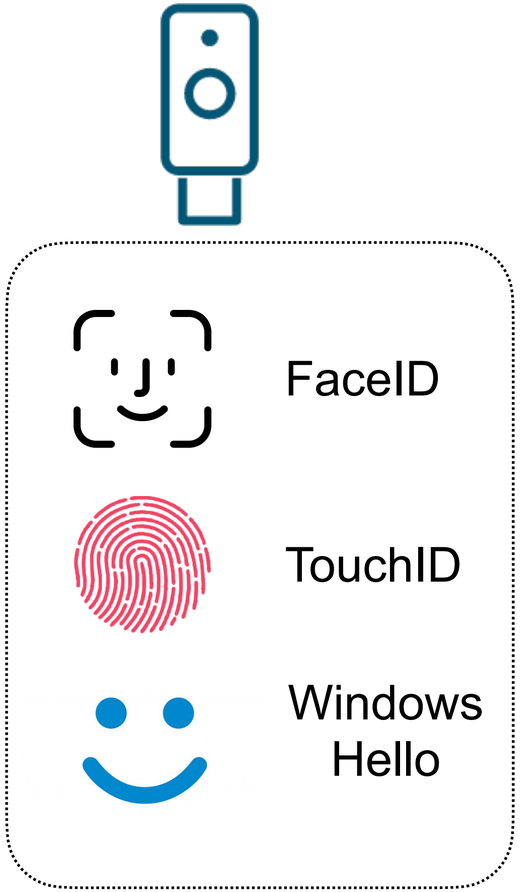
(Simplified)

response = sign(k, challenge)
+origin

result = verify(p, response, challenge)
+origin

Authenticators

- *Roaming* Authenticator
  example: a USB security key

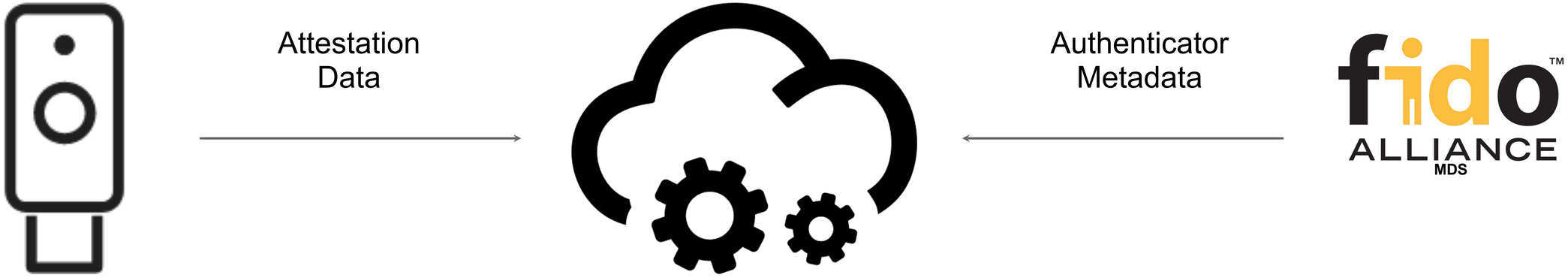


- *Platform* Authenticator
  example: a built-in fingerprint sensor

- Authenticators can store multiple FIDO credentials (passkeys)
- Roaming authenticators can use different *transports*: USB, NFC, BLE

Attestation and Metadata

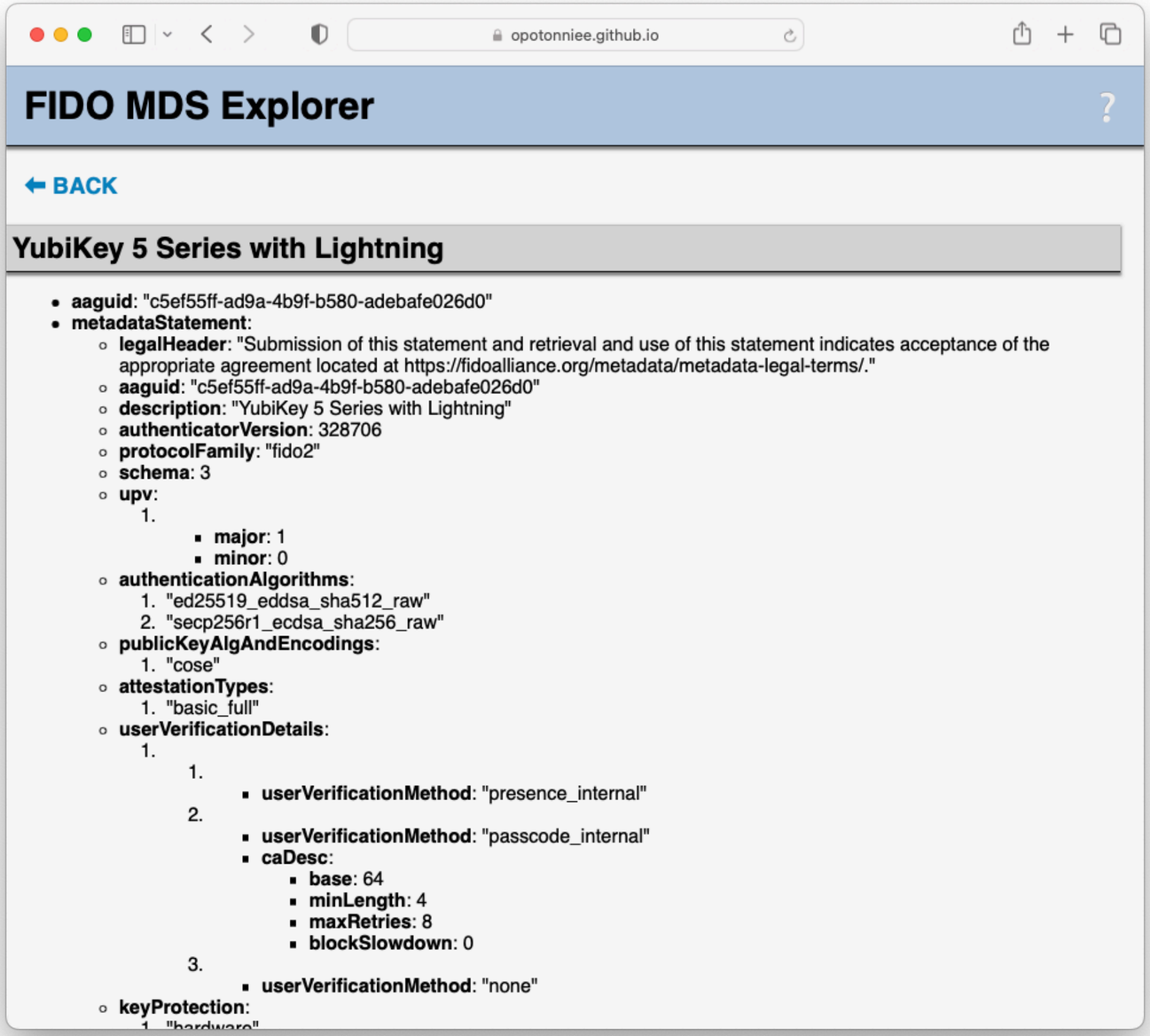
- Attestation provides verifiable evidence as to the authenticator’s origin
- Based on a hardware attestation key and certificate
- Use FIDO Alliance Metadata Service to determine provenance
- Implement Allow/Deny lists to filter Authenticators
- Typically used in high-assurance (enterprise) use cases



yubico

Metadata example

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



yubico

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

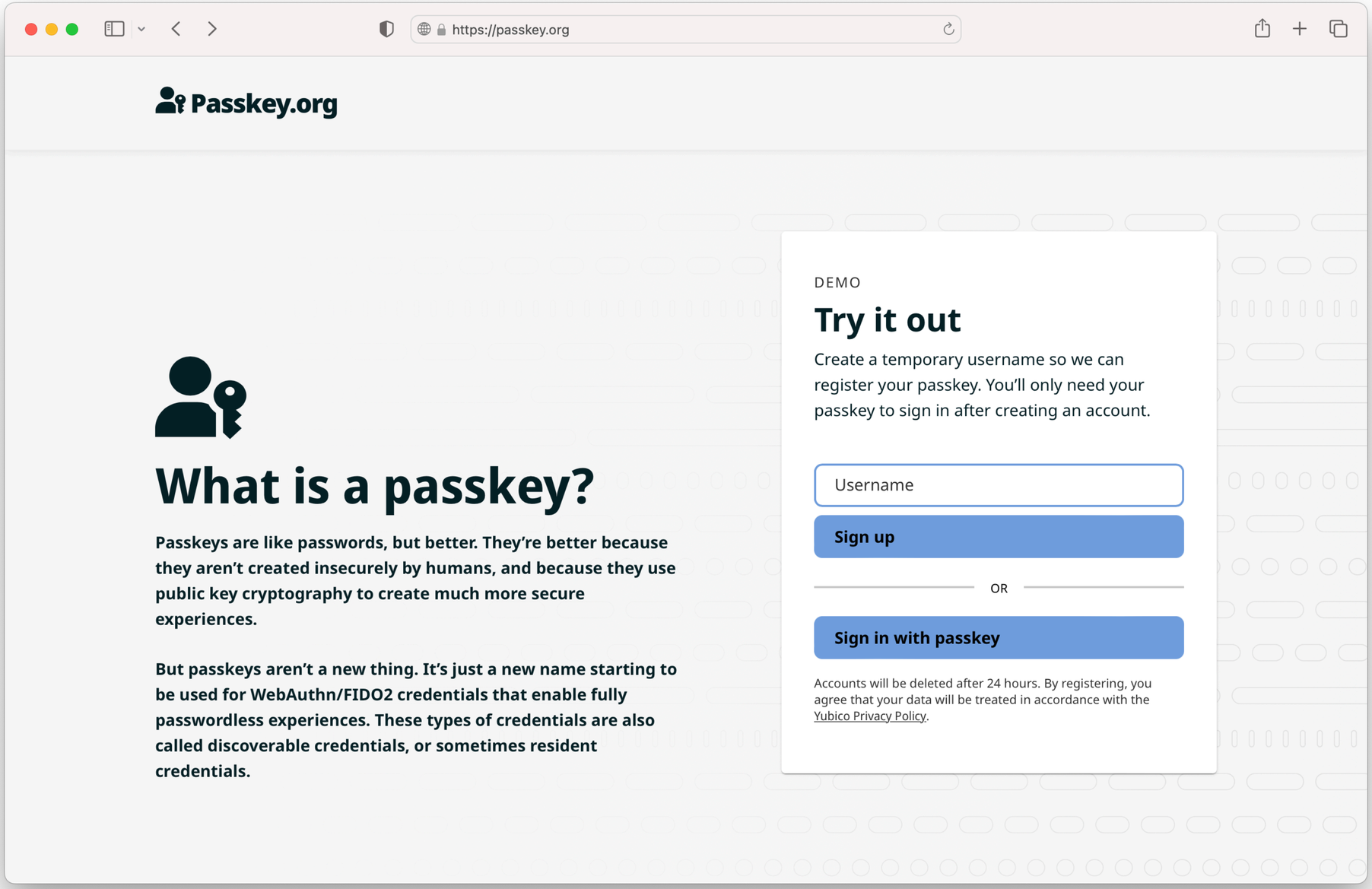
- No need to re-enroll a new device on every account!
- Synced across *devices* but not across *ecosystems*
  (i.e. a passkey synced via iCloud is not available on Android)

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



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

yubico

Deployment considerations

Low assurance

High assurance

Password

OTP

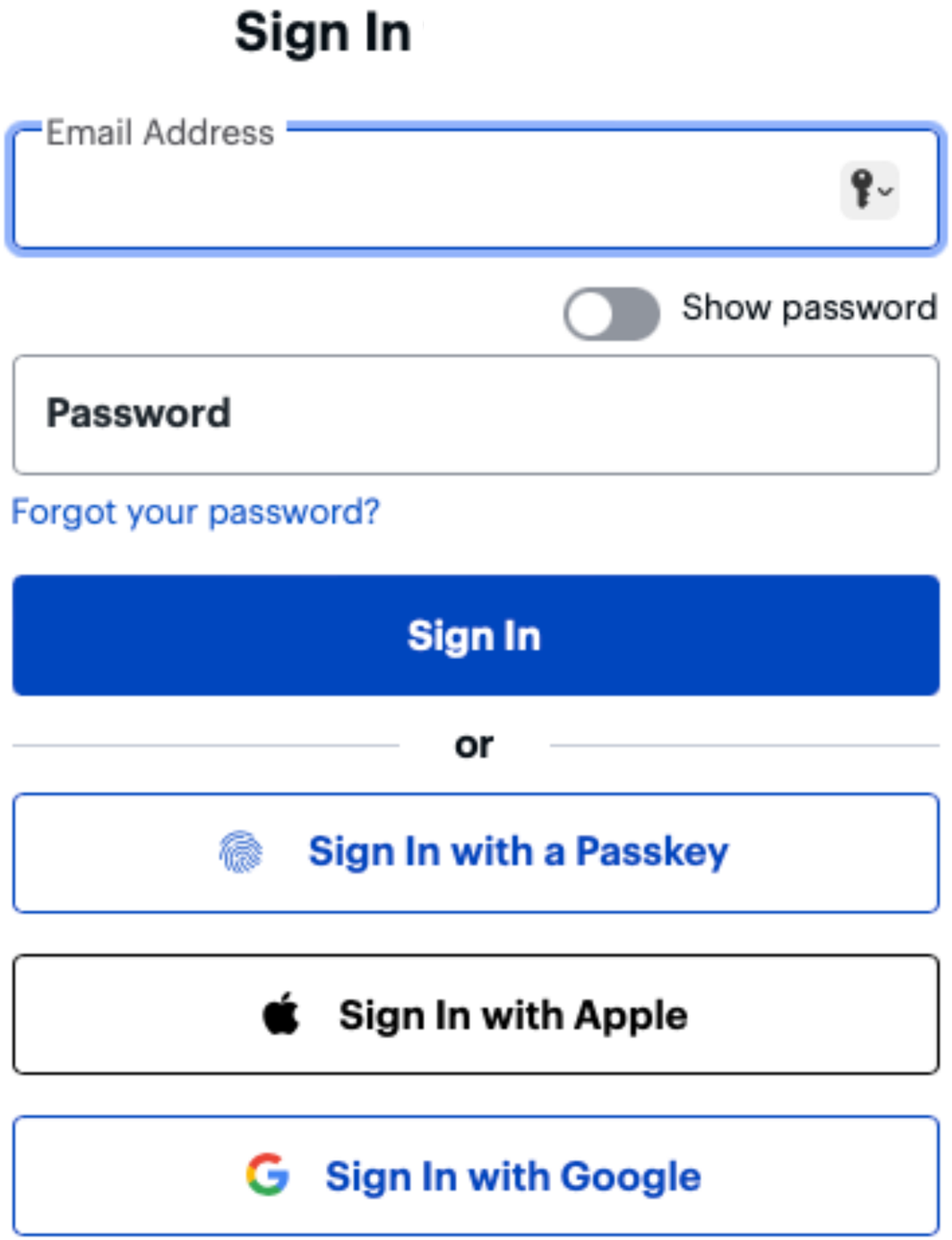
Copyable passkey

Hardware bound passkey

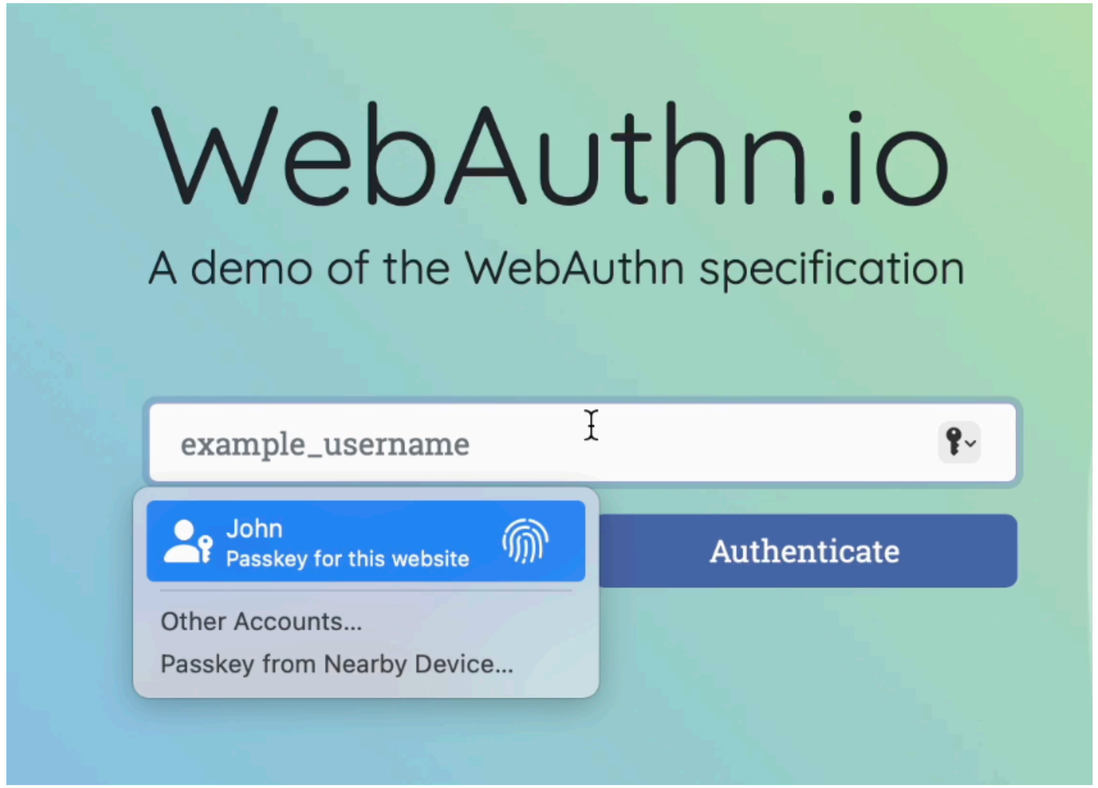
yubico

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?



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" />
```


yubico

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)

yubico

Hybrid (caBLE)

Decode

```
FIDO:/
284322846269769345345259231502556365370064829063443718468807301286316100222224786341523013544215784712235817551315763050072315363400833845326346315498242109321447142660
```

Decode

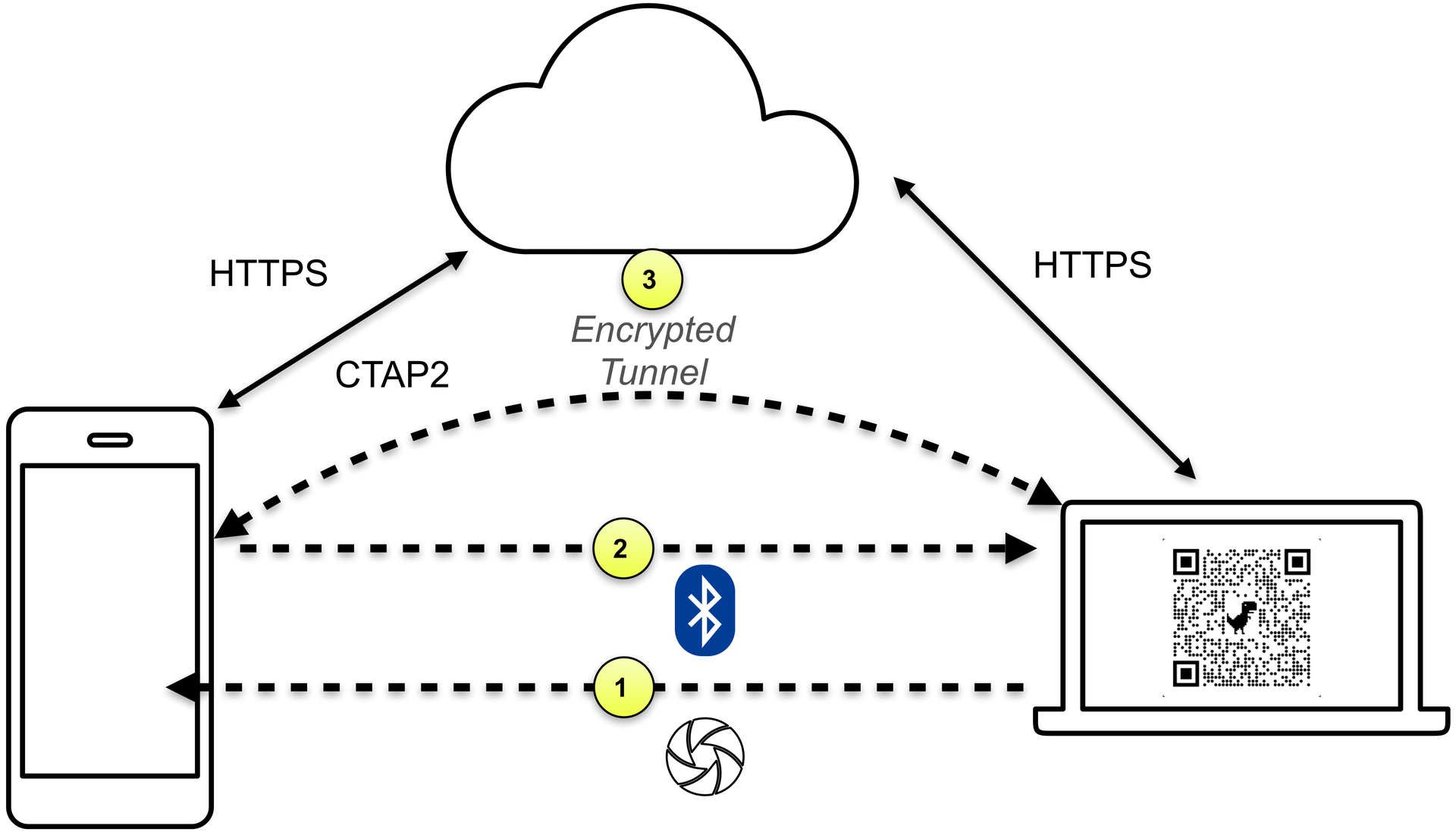
```
{
  0: h'0303659bc84279d9e6bd13a11445ba3de8f7a0237c5b40ff77e497ab32b9d8082d',
  1: h'b2c251f13fcc397bc753121d7953b491',
  2: 2,
  3: 1662126241_2,
  4: true,
  5: "mc",
}
```

Public key

Random secret

yubico

Hybrid (caBLE)

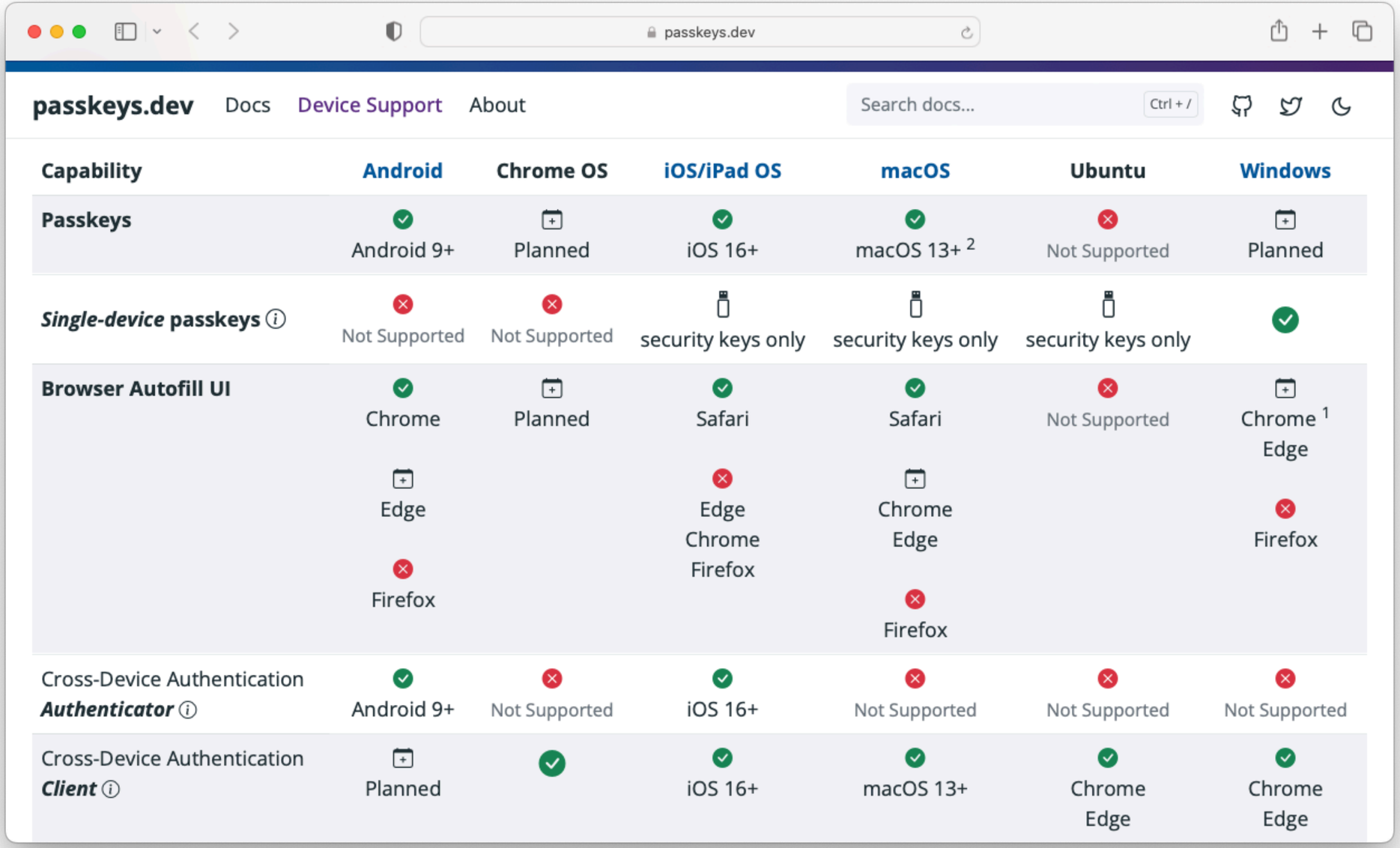


yubico

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



Capability	Android	Chrome OS	iOS/iPad OS	macOS	Ubuntu	Windows
Passkeys	Android 9+	Planned	iOS 16+	macOS 13+ [2]	Not Supported	Planned
***Single-device* passkeys** ⓘ	Not Supported	Not Supported	security keys only	security keys only	security keys only	
Browser Autofill UI	Chrome Edge Firefox	Planned	Safari Edge Chrome Firefox	Safari Chrome Edge Firefox	Not Supported	Chrome [1] Edge Firefox
Cross-Device Authentication ***Authenticator*** ⓘ	Android 9+	Not Supported	iOS 16+	Not Supported	Not Supported	Not Supported
Cross-Device Authentication ***Client*** ⓘ	Planned		iOS 16+	macOS 13+	Chrome Edge	Chrome Edge

yubico

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/
- Open Source Software:
  - https://github.com/Yubico/java-webauthn-server/
    (v2.2 support for passkeys, released Nov 24)
  - https://github.com/YubicoLabs/WebAuthnKit

yubico

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)

yubico

Questions?