Presenting Prettier Passkeys (for RPs)

Matthew Miller
Senior Technical Lead @ Cisco Duo

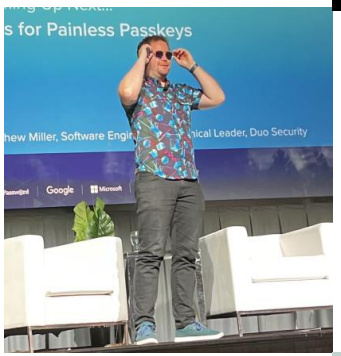
authenticate 2025
THE FIDO CONFERENCE

Signature Sponsors
Google | Microsoft | VISA | yubico

Who am I?

- **WebAuthn SME** with an eye on the Relying Party developer experience
- Author of **SimpleWebAuthn** and **py_webauthn** libraries
- Current steward of **webauthn.io**
- FIDO Alliance **Board Member** and W3C **WAWG Editor**

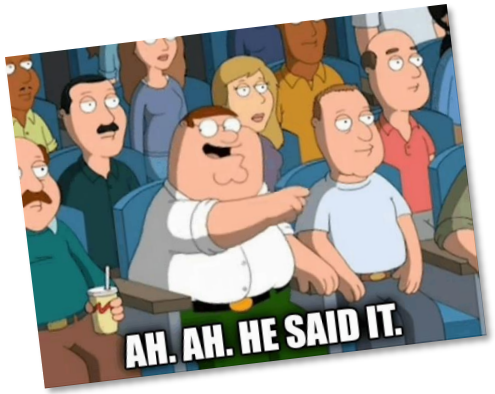
Authenticate 2022



Authenticate 2023

Authenticate 2024

Agenda



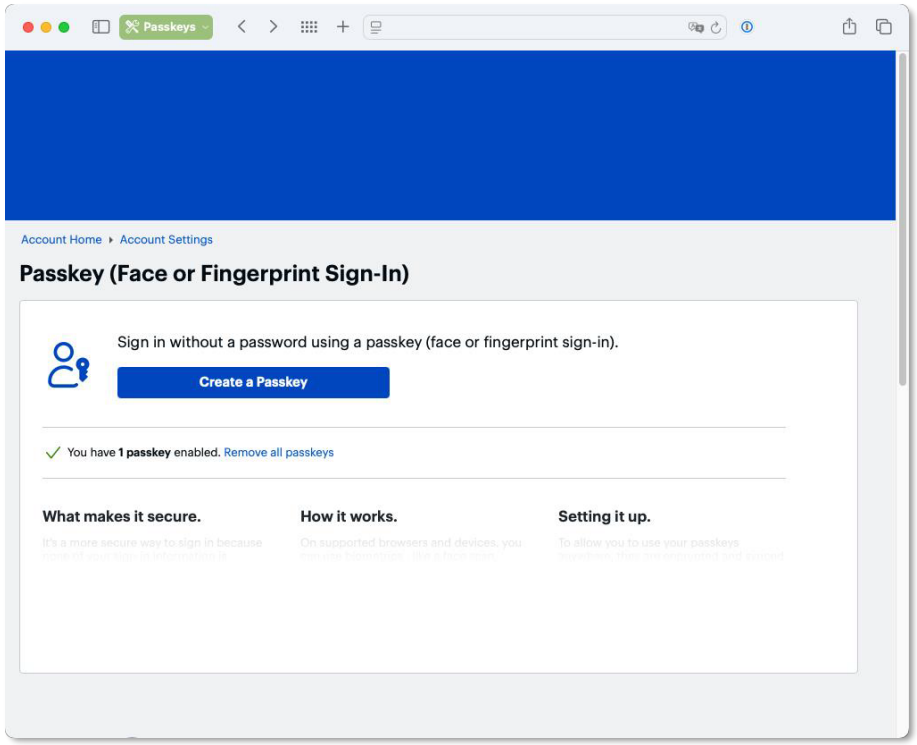
- Passkey management antipatterns
- Presenting prettier passkeys
- Taking things one step further
- Q & A

Passkey management antipatterns

(Names have been changed to protect the innocent)

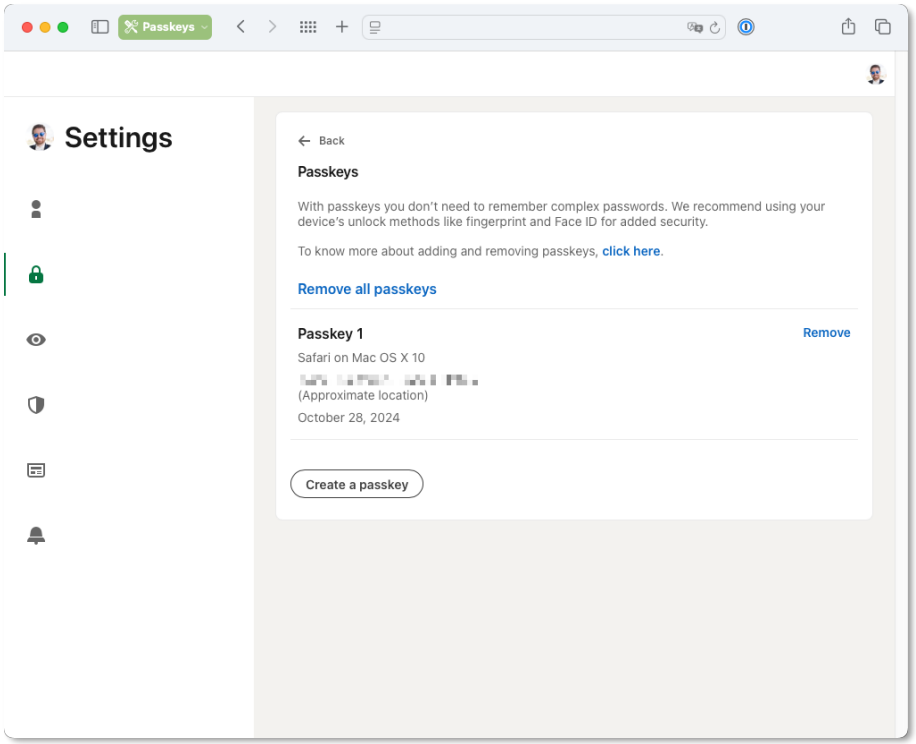
Example 1: GreatPurchase.com

- Only shows the number of passkeys the user has
- Management consists of a single option to “remove all passkeys”



Example 2: ConnectedOn.com

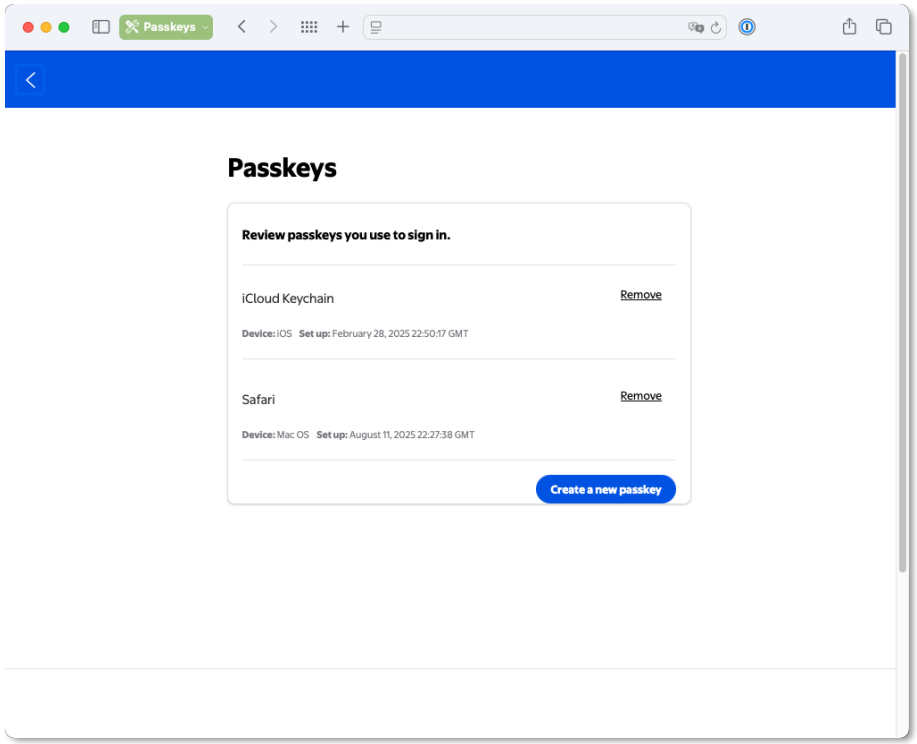
- Passkeys are all listed, but only as “Passkey 1”, “Passkey 2”, etc...
- Provider-identifying information is not displayed anywhere



Example 3: VerticalStore.com

- Passkey registration and management (were) only offered on mobile

- Passkeys get nicknames that capture where it was ***registered***, but not where it ***resides***



Presenting prettier passkeys

Elevate your passkey management UX

List their passkeys

Name their passkeys

Show familiar provider iconography

Introducing FIDO Convenience MDS

A subset of FIDO Metadata Service (MDS) data to easily look up **provider name** and **iconography** by **AAGUID**

ETA: Coming Soon™

More Info:

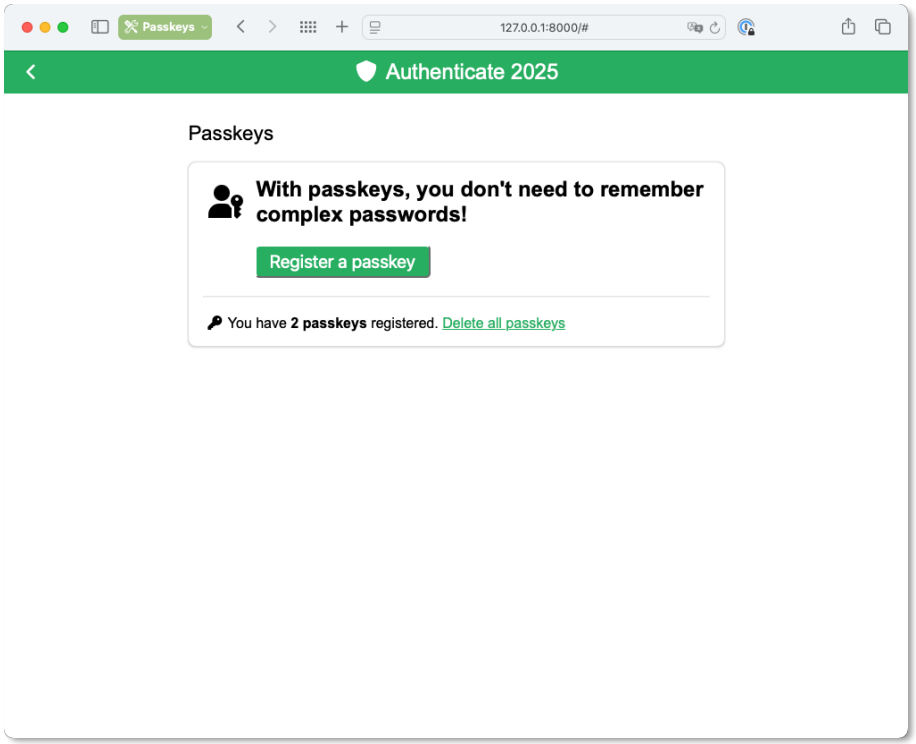
https://fidoalliance.org/metadata

```
dictionary ConvenienceDetails {
  FriendlyNames friendlyNames;
  DOMString     providerLogoLight;
  DOMString     providerLogoDark;
};

dictionary FriendlyNames {
    DOMString *IETFLanguageCodes-members...;
}
```

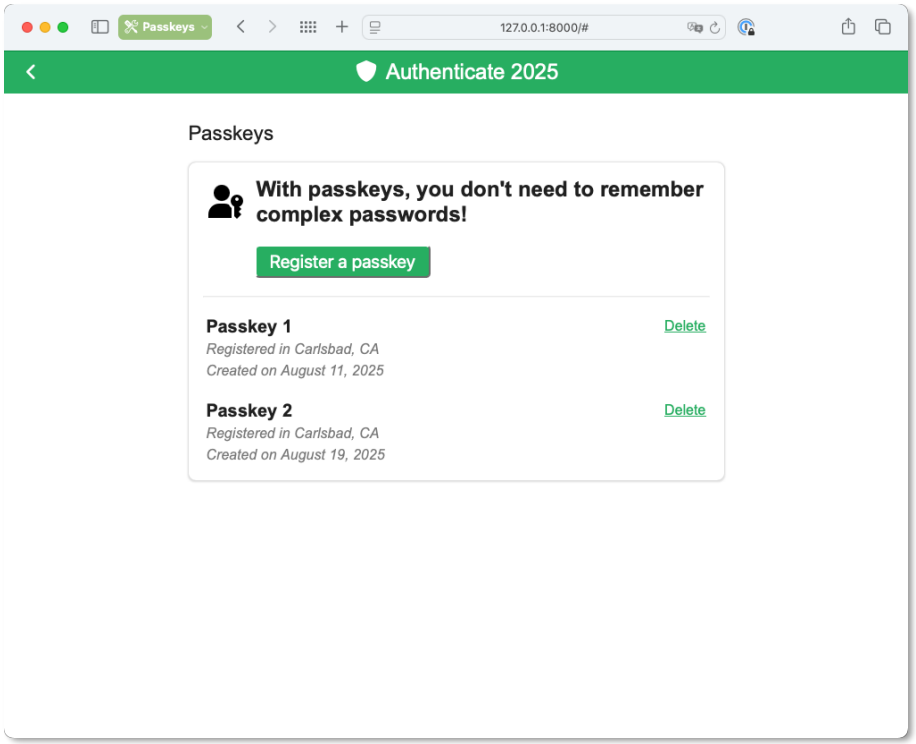
```
{
  "no": 85,
  "ea9b8d66-4d01-1d21-3ce4-b6b48cb575d4": {
    "friendlyNames": { "en-US": "Google Password Manager" },
    "providerLogoDark": "...",
    "providerLogoLight": "..."
  },
  "08987058-cadc-4b81-b6e1-30de50dcbe96": {
    "friendlyNames": { "en-US": "Windows Hello" },
    "providerLogoDark": "...",
    "providerLogoLight": "..."
  },
  "a25342c0-3cdc-4414-8e46-f4807fca511c": {
    "friendlyNames": { "ja-Hani-JP": "YubiKey 5 シリーズ (NFC 搭載)" },
    "icon": "..."
  }
}
```

Our Starting Point



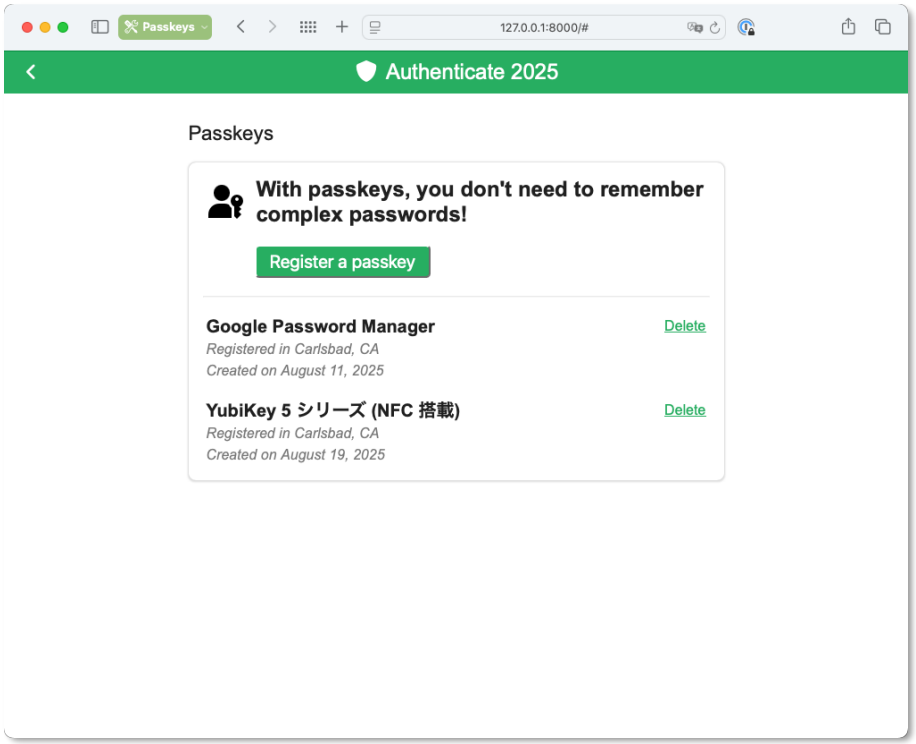
List their passkeys

- Let the user see the individual passkeys they can use to log in
- Showing a single passkey can still help users (or those helping them) remember what provider is managing it



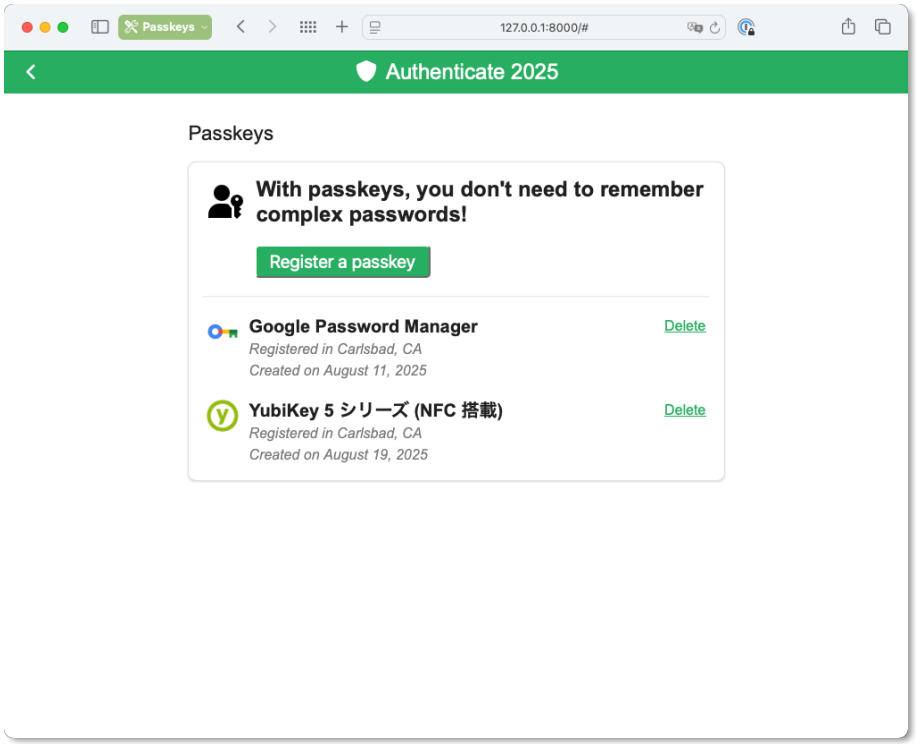
Name their passkeys

- Look up provider's friendly name by AAGUID
- Consider offering the ability to manually nickname passkeys
- Nothing is guaranteed about unattested AAGUID, but it's Good Enough for most deployments

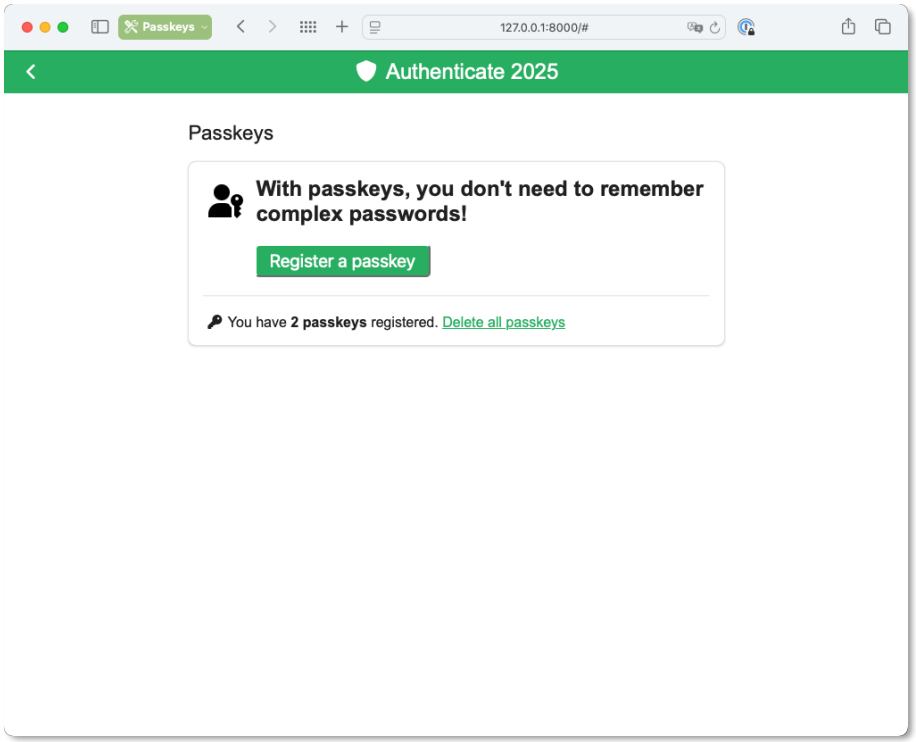


Show familiar provider iconography

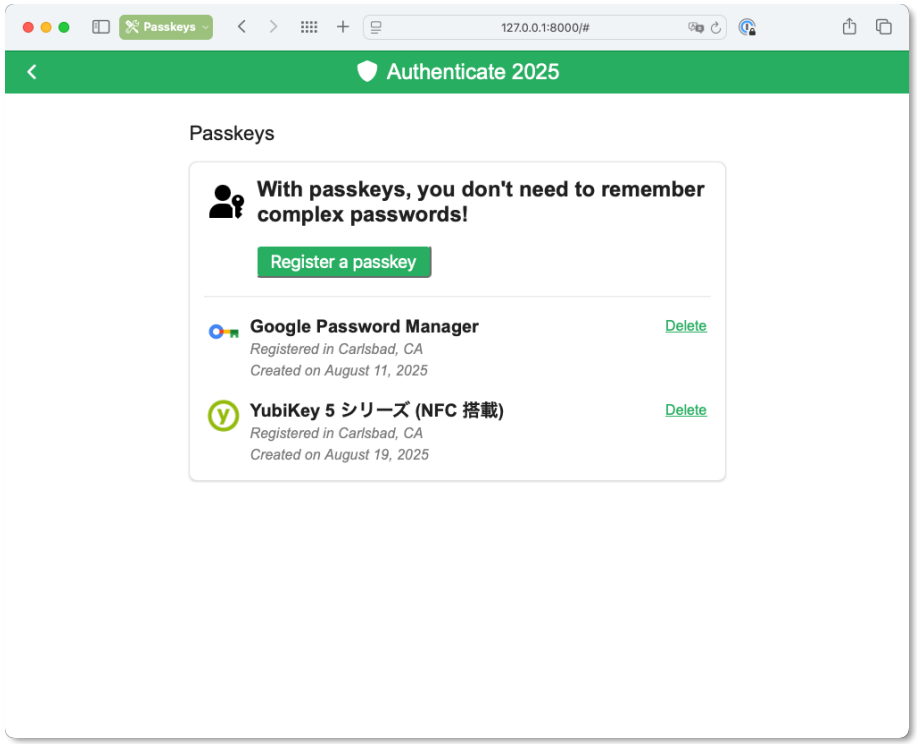
- Look up provider's icon by AAGUID
- Offer users something to visually distinguish entries
- As with nickname, using AAGUID to determine iconography to show does not impact the security of passkeys



Before



After



Taking things one step further

WebAuthn Signal APIs

www.w3.org/TR/webauthn-3/#sctn-signal-methods

Three **PublicKeyCredential** methods for Relying Parties to help keep passkey providers in sync when passkey state changes:

signalUnknownCredential()

A newly registered passkey is **rejected by the RP**.

The provider **deletes the new passkey** so the user stops being prompted to use it to log in.

signalAllAcceptedCredentials()

The user **deletes a passkey** from the RP's user settings.

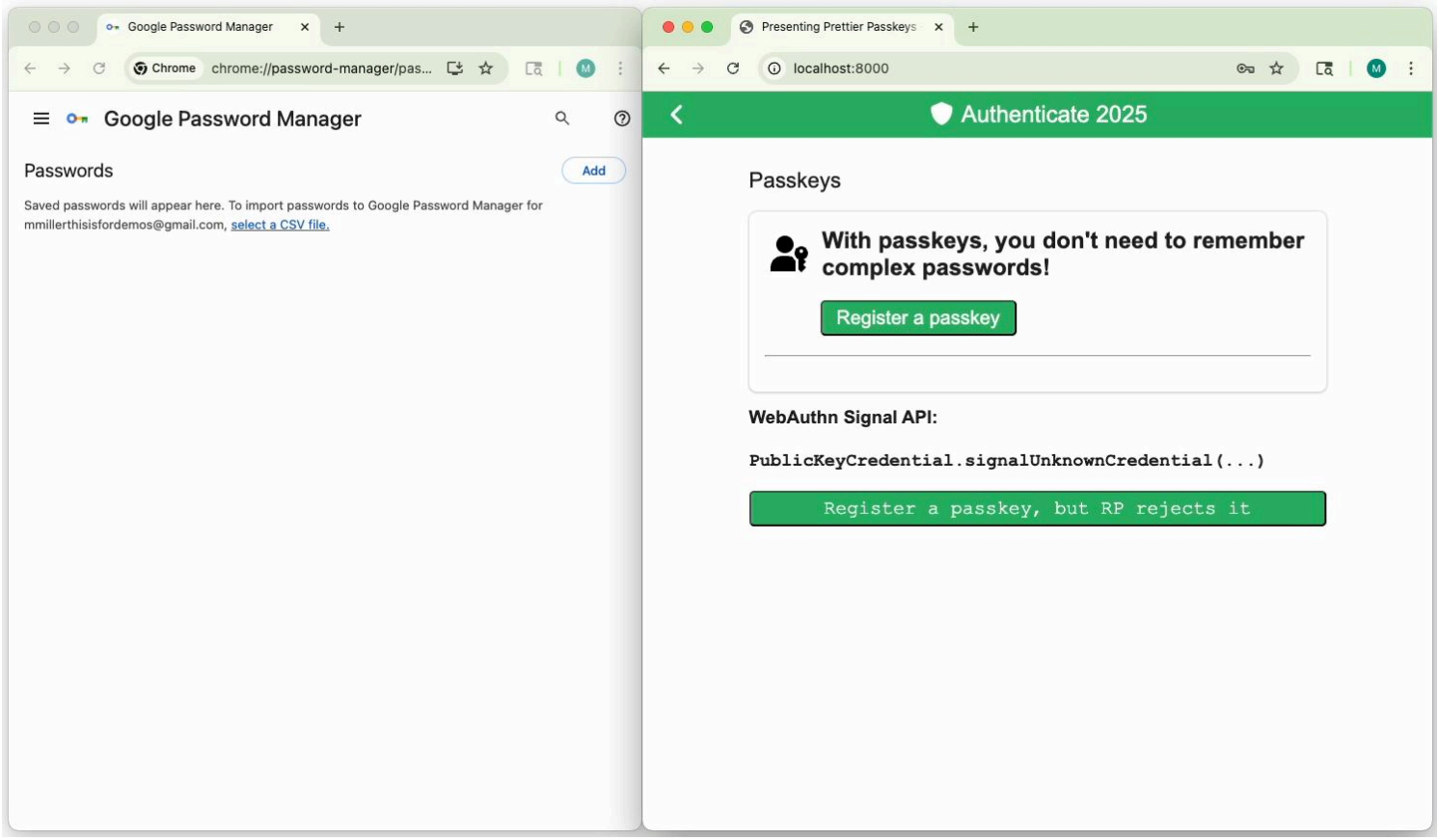
The provider **deletes its existing passkey** so the user stops being prompted to use it to log in.

signalCurrentUserDetails()

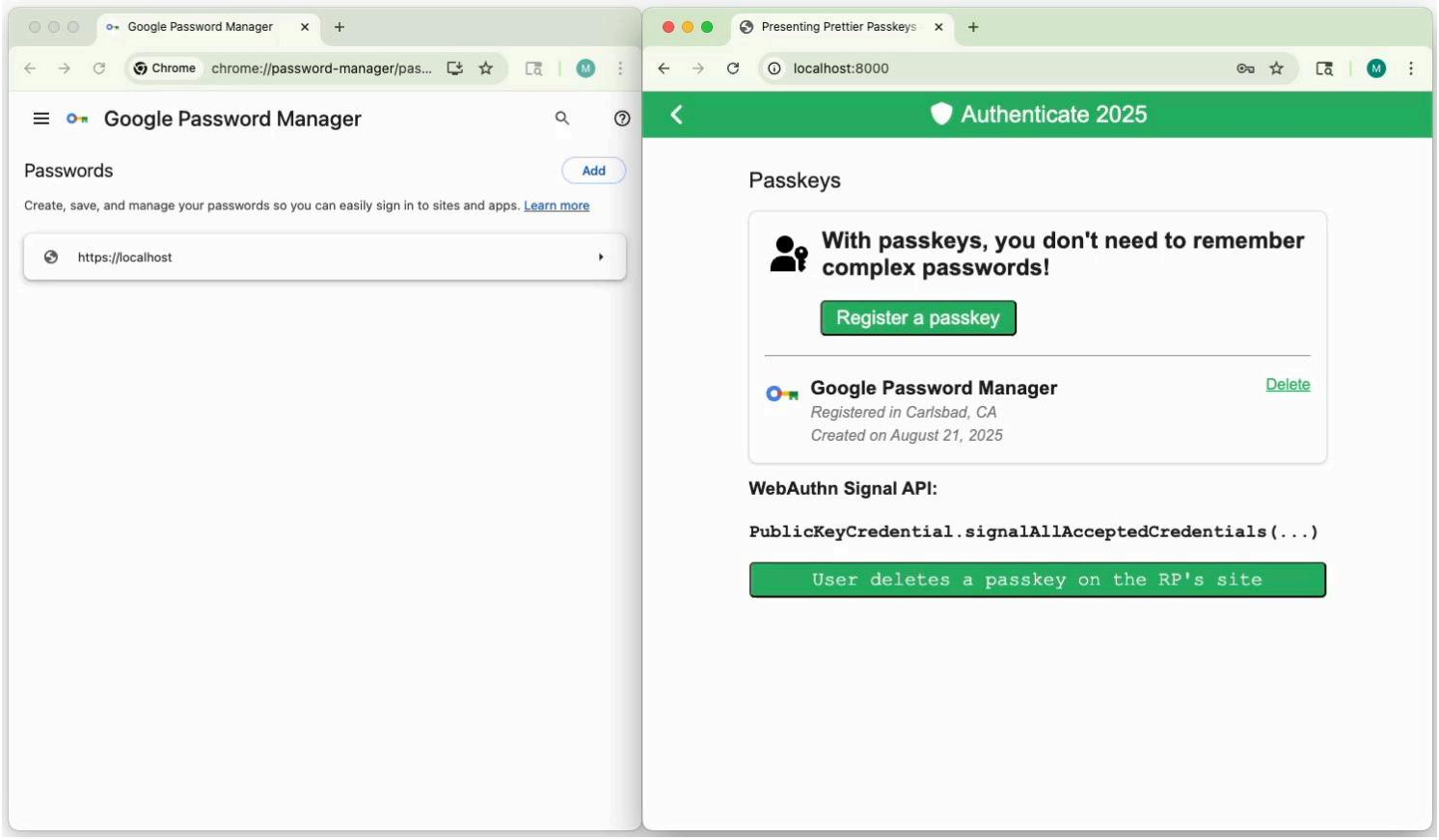
The user **updates their RP email, username, etc...**

The provider **updates its metadata** for the existing passkey to stop showing old email, username, etc...

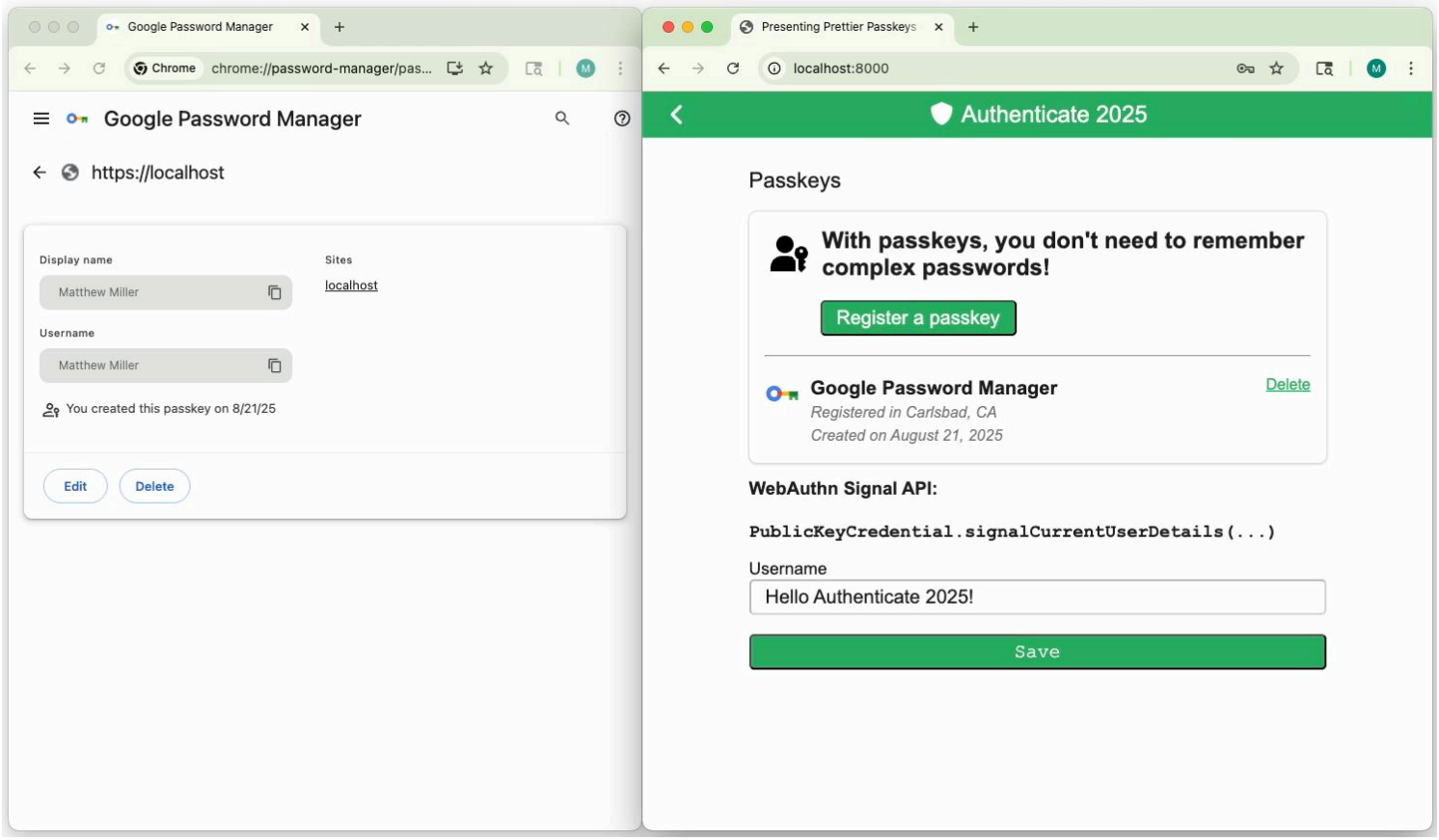
PublicKeyCredential.signalUnknownCredential()



PublicKeyCredential.signalAllAcceptedCredentials()



PublicKeyCredential.signalCurrentUserDetails()



Q & A

Thank you!