FOSDEM 2026

Credentials for Linux
Bringing Passkeys to the Linux desktop

Alfie Fresta (@afresta:noentropy.org)

`github.com/linux-credentials` `#credentials-for-linux:matrix.org`

What are Passkeys *exactly*?

Passkeys

- **Phishing-resistant authentication**
- Public-key credentials
- Built on WebAuthn (W3C) & CTAP (FIDO) standards

Why it matters:

Fewer passwords & better UX

FIDO2 credentials

- **FIDO U2F** evolved to **FIDO2** (WebAuthn)

- "Passkey" = FIDO2 *discoverable* credential
  - Passwordless & username-less

- New FIDO2 flows:
  - Use a Passkey on your phone (QR code)
  - Synced Passkeys

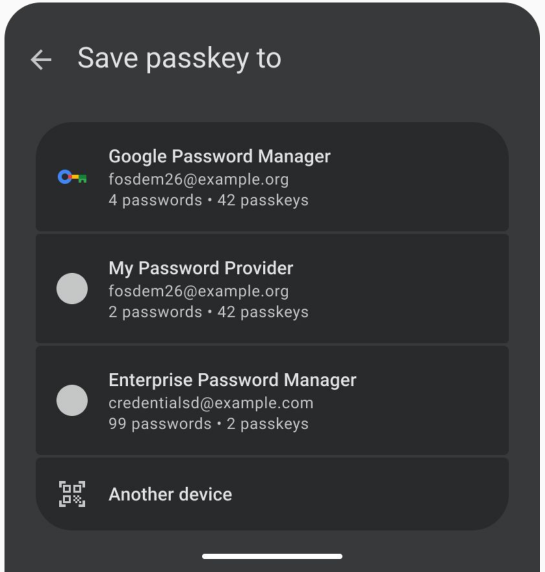
Curious? Watch: "WTF is a passkey with William Brown" by Open Source Security

Modern Passkeys

- Most users will **sync credentials** across devices
- Phones will be the **default *roaming* authenticator** for most users
- Security keys will still be important
    - Especially in enterprise

Third-party credential providers

- Passkeys often live in **third-party credential providers**
  - Automatically synced across devices
- Defaults
  - Google Password Manager
  - iCloud Keychain (Passwords.app)
- Third party
  - Bitwarden, 1Password, and more
  - OSS & self hosted
- Requires **Credential Provider APIs**


Save passkey to

Google Password Manager
fosdem26@example.org
4 passwords • 42 passkeys

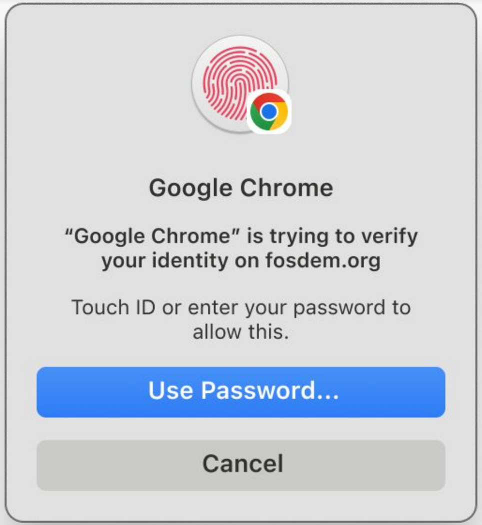
My Password Provider
fosdem26@example.org
2 passwords • 42 passkeys

Enterprise Password Manager
credentialsd@example.com
99 passwords • 2 passkeys

Another device

The problems

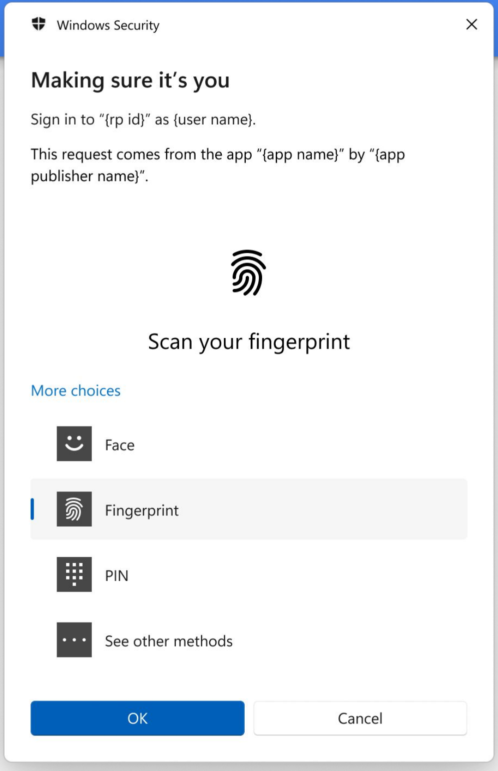
Problem #1: Missing Platform APIs



Google Chrome

"Google Chrome" is trying to verify your identity on fosdem.org

Touch ID or enter your password to allow this.



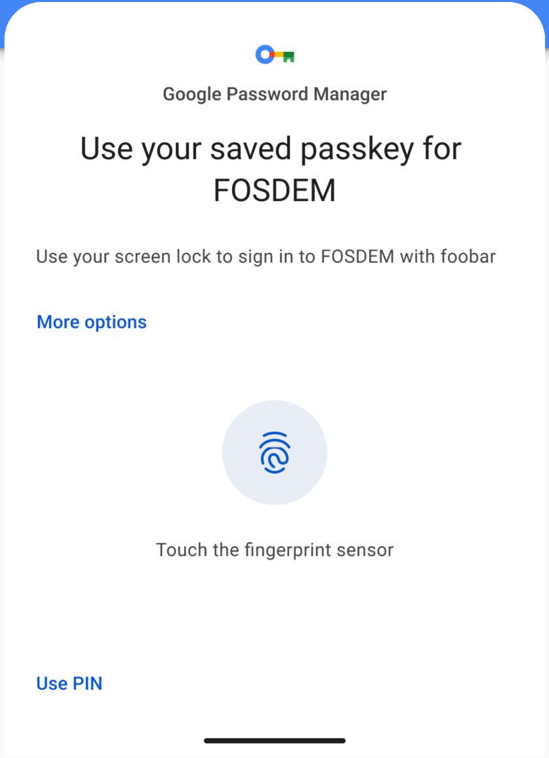
Windows Security

Making sure it's you

Sign in to "{rp id}" as {user name}.

This request comes from the app "{app name}" by "{app publisher name}".

Scan your fingerprint



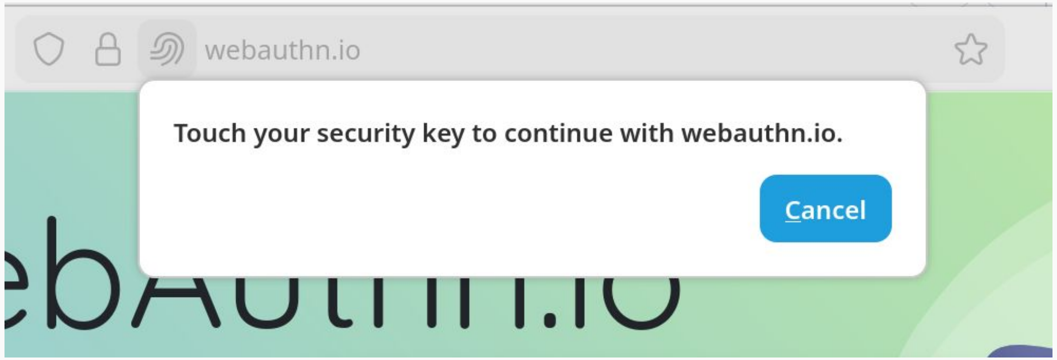
Google Password Manager

Use your saved passkey for FOSDEM

Use your screen lock to sign in to FOSDEM with foobar

Images: passkeycentral.com > Figma Kits

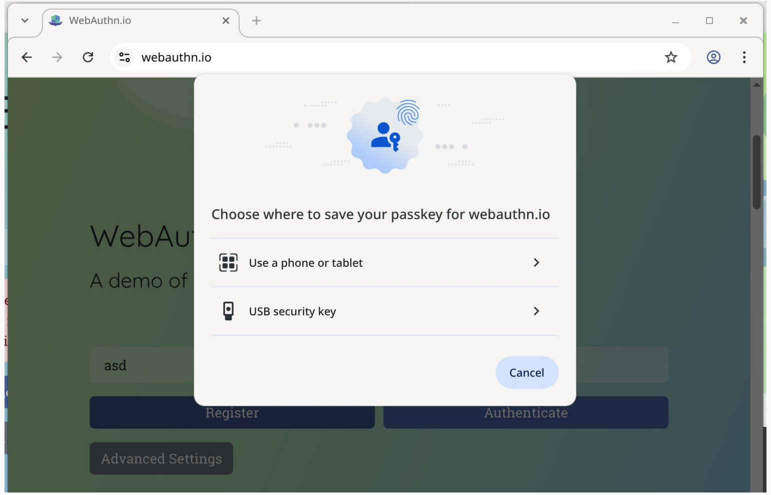
Problem #2: Inconsistent UX & feature gaps


webauthn.io

Touch your security key to continue with webauthn.io.

Cancel

Firefox on Fedora



Choose where to save your passkey for webauthn.io

Use a phone or tablet

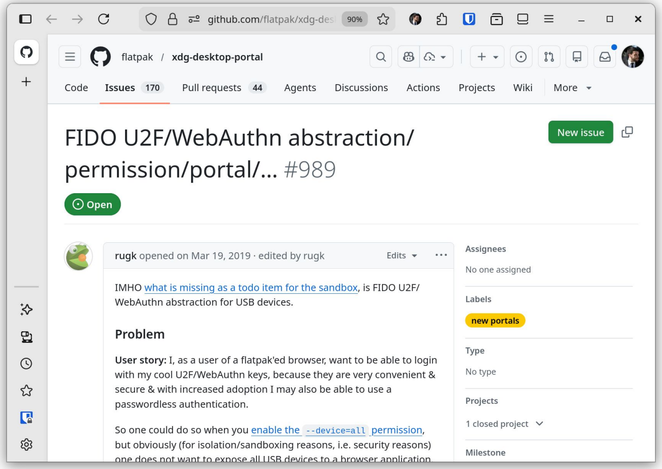
USB security key

Cancel

Chromium on Fedora

Problem #3: Containerised apps support

- Flatpak browsers need WebAuthn, but…
  - No Passkey APIs within sandbox
- Current workaround: `--device=all`
  Broad device access permissions
  - Breaks sandbox goals
  - **Enables origin bypass risk**



FIDO U2F/WebAuthn abstraction/ permission/portal/… #989

Open

rugk opened on Mar 19, 2019 · edited by rugk

IMHO what is missing as a todo item for the sandbox, is FIDO U2F/ WebAuthn abstraction for USB devices.

Problem

User story: I, as a user of a flatpak'ed browser, want to be able to login with my cool U2F/WebAuthn keys, because they are very convenient & secure & with increased adoption I may also be able to use a passwordless authentication.

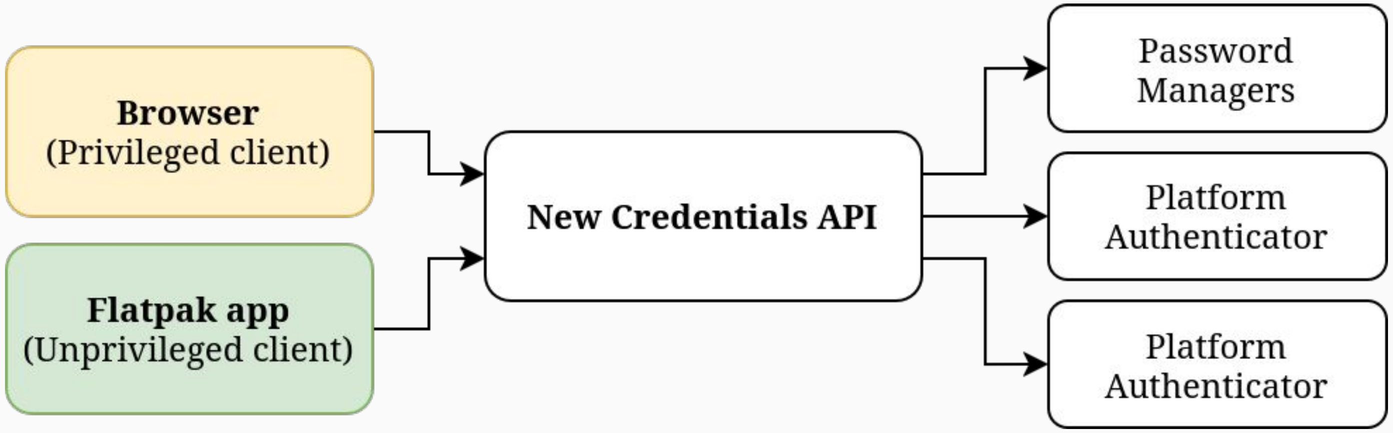
So one could do so when you enable the `--device=all` permission, but obviously (for isolation/sandboxing reasons, i.e. security reasons) one does not want to expose all USB devices to a browser application.

[flatpak/xdg-desktop-portal issue #989]

A new *Credentials API*

The idea

- A **D-Bus API** mediates access to **credentials & authenticators**
- Supports **privileged & unprivileged** clients



New components

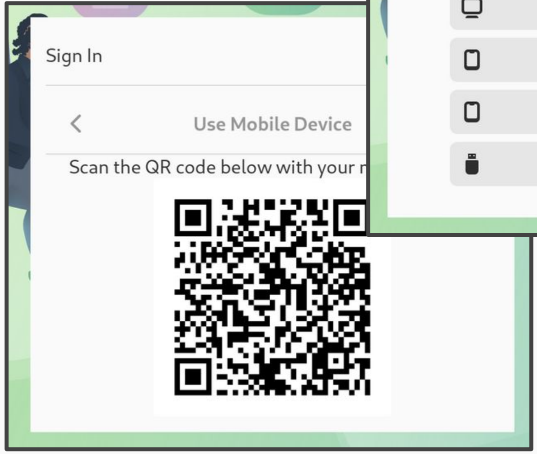
- **libwebauthn**
  Platform library (CTAP/WebAuthn)

- **credentialsd**
  D-Bus Credentials Portal for applications
  & Credential Providers

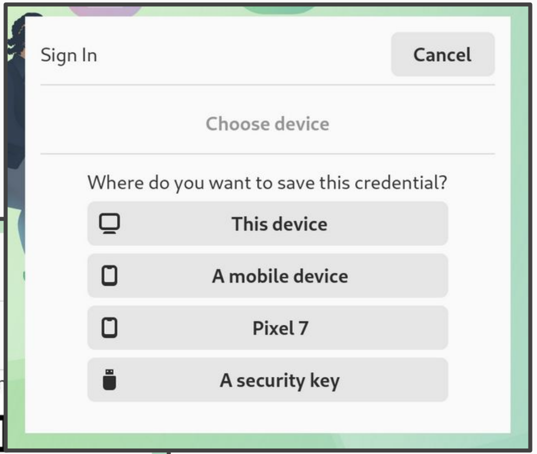
credentialsd

github repo

- A Credentials API for Linux
  - Credentials Portal over D-Bus
  - Credentials Provider API Proposal
- Includes a reference UI
- Packaged for Fedora/openSUSE
- Try it out:
  - Firefox add-on (experimental)
  - Patched Firefox flatpak



Scan the QR code below with your



Where do you want to save this credential?

credentialsd backend & frontend

- Split responsibility:
  - **Frontend:** Protocols + Devices + Providers
  - **Backend**: Trusted UX prompt (owned by DEs)
- Portal-style request verification + UI handoff

XDG Desktop Portals

- Portals are de-facto standard sandbox/platform APIs on Linux

- Consistent mediation across desktop

- We are working with **xdg-desktop-portal** developers to integrate credentialsd, implementing the new Credentials portal API (WIP PR open!)

libwebauthn

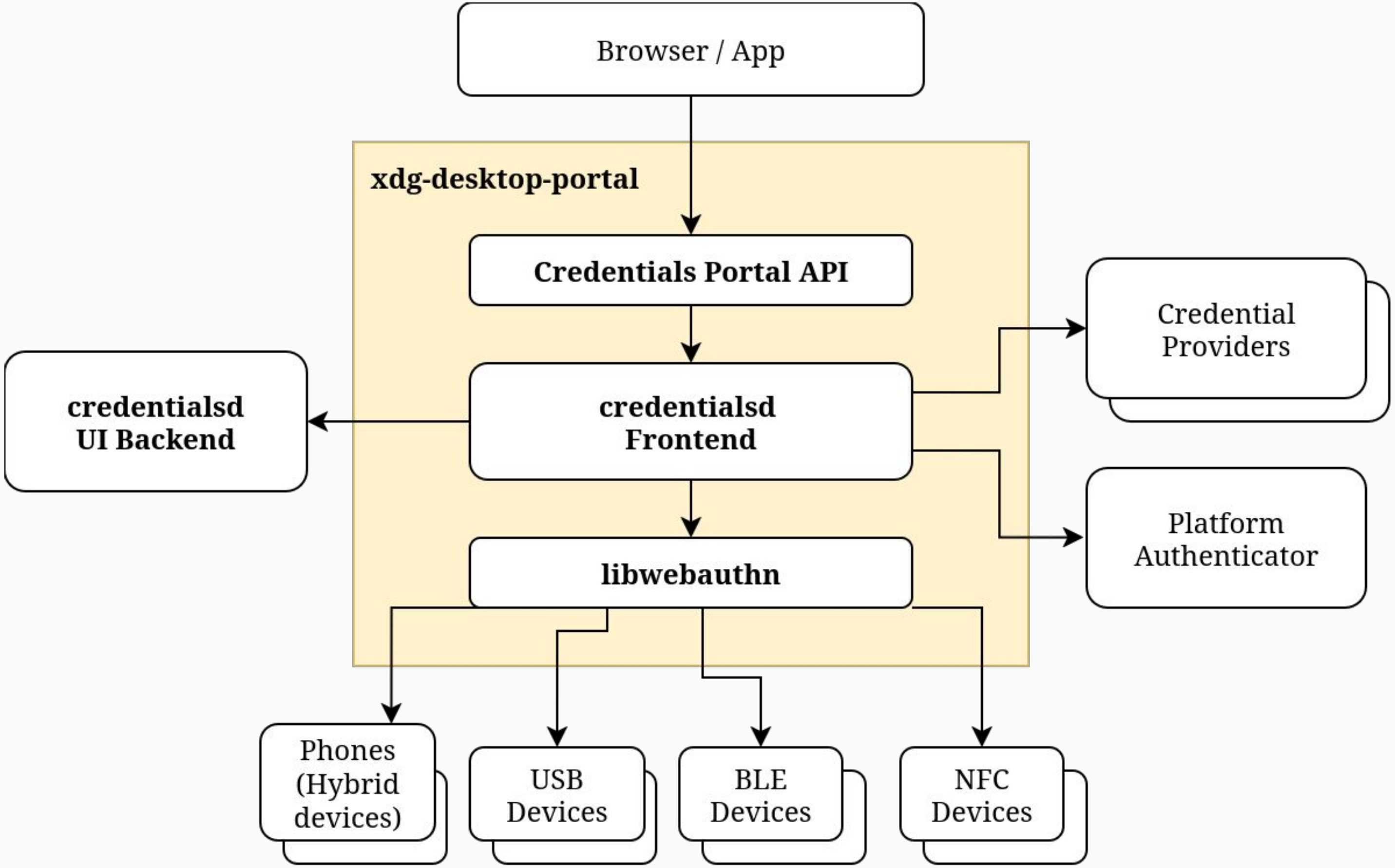
github repo

- Linux native
- Written in Rust
- Pluggable transports
    - Including hybrid (QR code + "remember this phone")
- Supports FIDO U2F & FIDO2
- PIN/UV, discoverable credentials

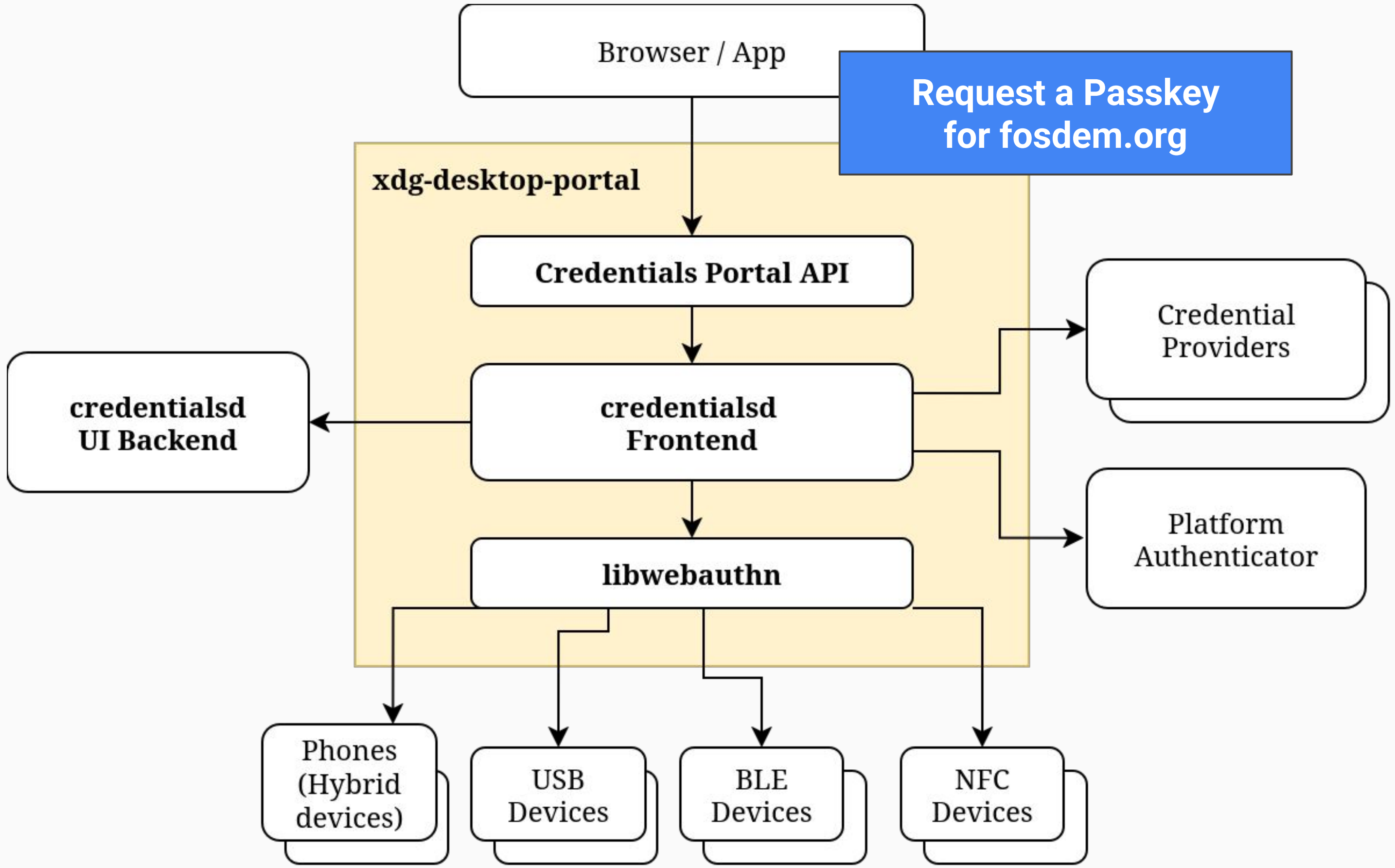
libwebauthn transport matrix

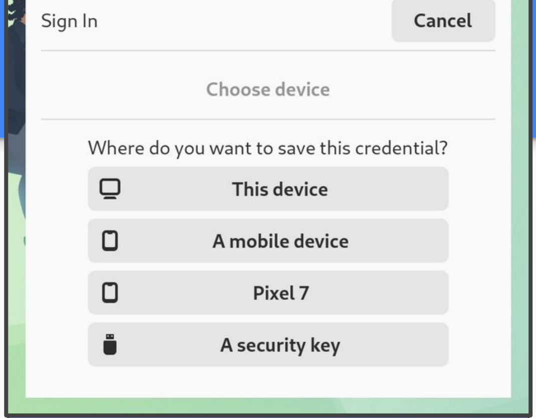
	FIDO U2F	WebAuthn (FIDO2)
USB (HID)	🟢 Supported (hidapi)	🟢 Supported (hidapi)
Bluetooth Low Energy	🟢 Supported (bluez)	🟢 Supported (bluez)
NFC	🟢 Supported (pcsc or libnfc)	🟢 Supported (pcsc or libnfc)
TPM 2.0 (Platform)	🟠 Planned (#4)	🟠 Planned (#4)
Hybrid (QR code scan, aka caBLE v2)	N/A	🟢 Supported

Credentials Portal API



Credentials Portal API





Sign In

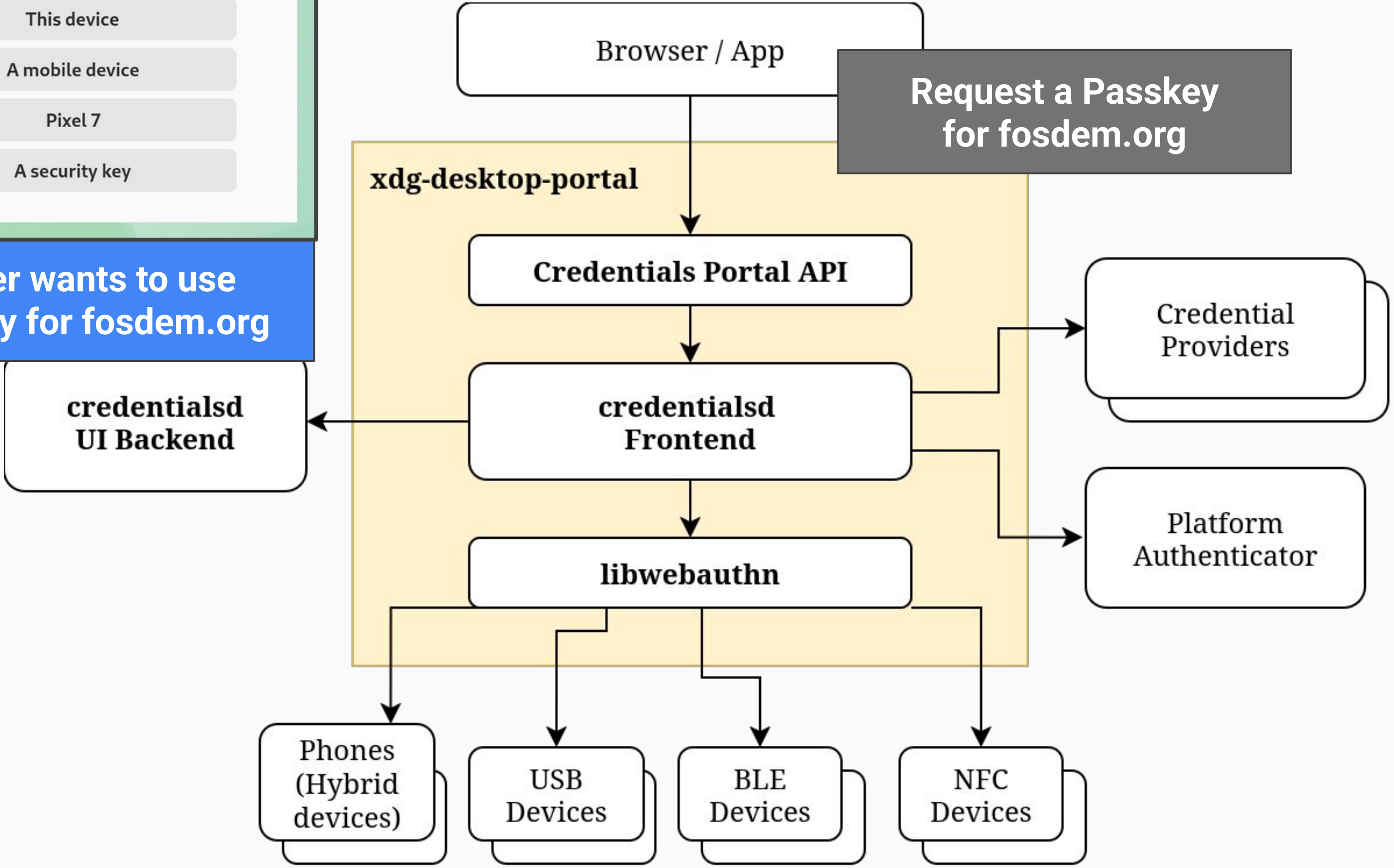
Cancel

Choose device

Where do you want to save this credential?

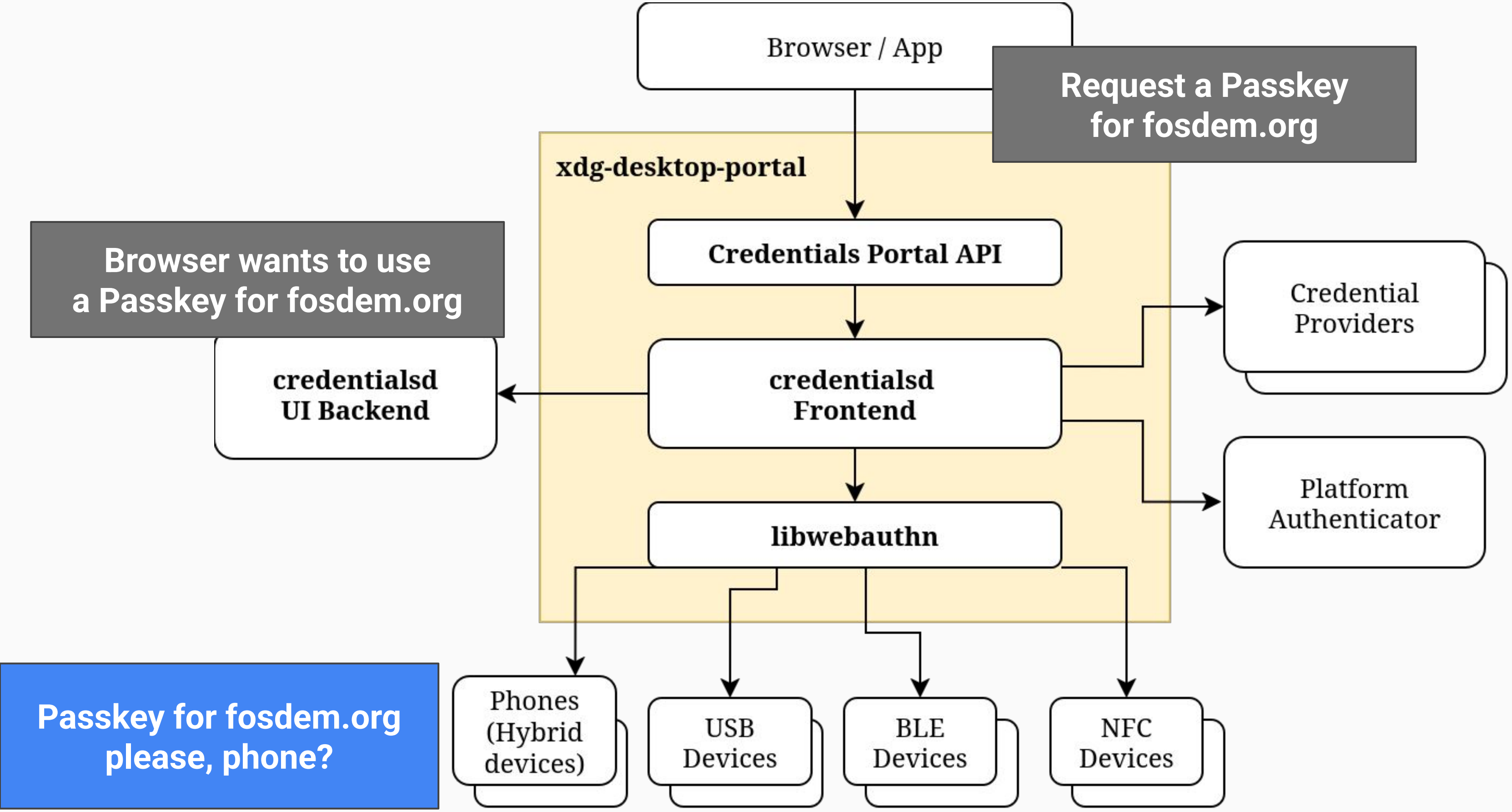
- This device
- A mobile device
- Pixel 7
- A security key

Browser wants to use
a Passkey for fosdem.org

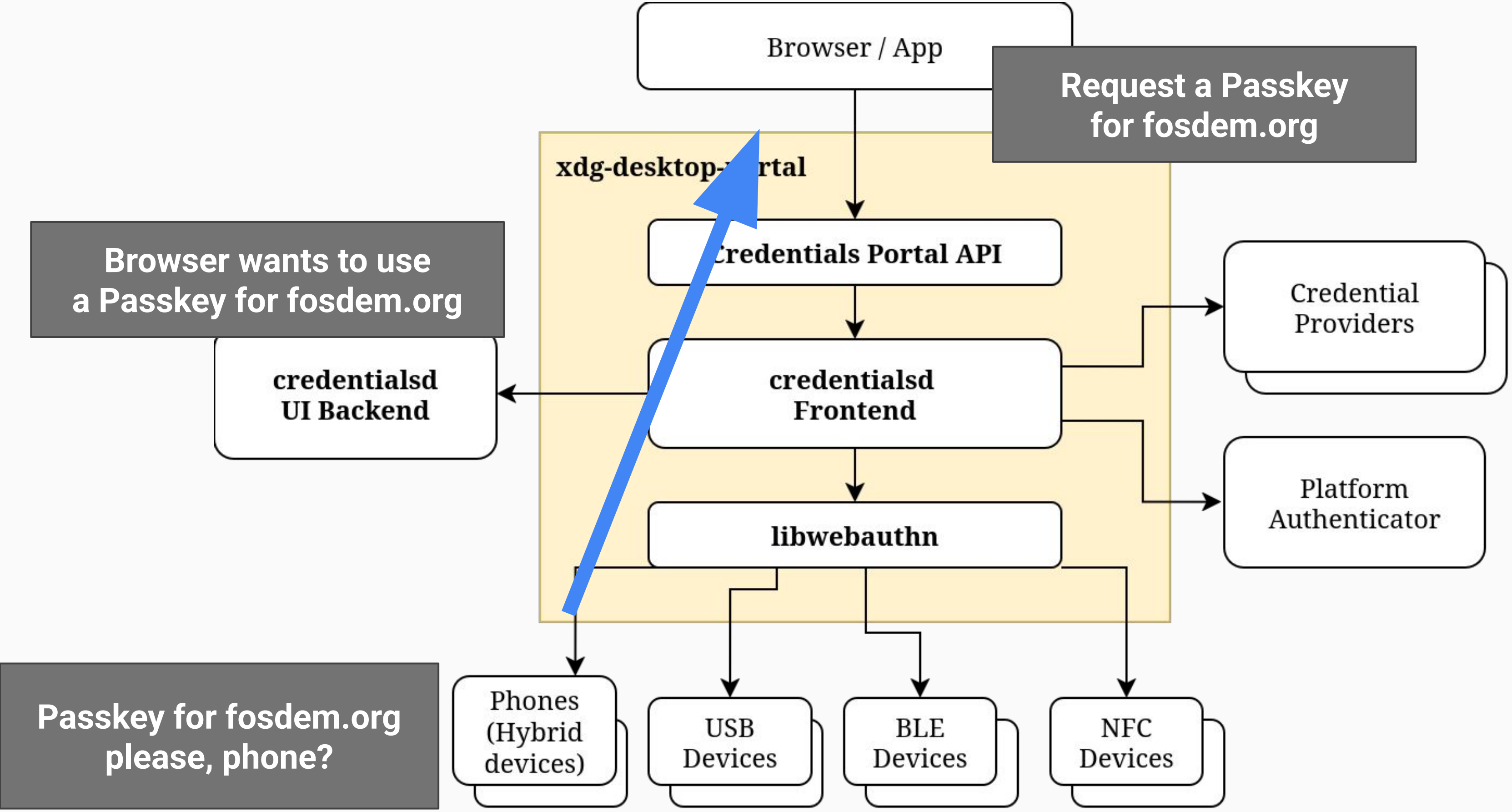


Request a Passkey
for fosdem.org

Credentials Portal API



Credentials Portal API



Open challenges

Challenge #1: Origin scoping

- WebAuthn requires **origin scoping:**
  - Credentials for fosdem.org, can only be accessed by fosdem.org
  - … or related origins
  - … or fosdem.org authorised apps

- Linux: How do we determine the *origin* of the request?

Challenge #2: App identity verification

- To identify *allowed origins* for an app,
  we have to first **verify the app's identity**.
- Android & iOS achieve this via signing keys.
- Linux:
  How do we *authenticate a calling application?*

Privileged vs Unprivileged APIs

- End goal: different APIs & policies
  - **Browsers:** Privileged, can request Passkeys for *any* origin
  - **Apps:** Restricted, can request Passkeys for *verified* origins only
- Current state:
  - Single API, with **explicit** UX confirmation

What's next?

What's next?

- **Upstream collaboration with desktops, browsers, distros**
  - Including Chromium, Firefox, WebKit, Servo
- Support third-party credential providers
- "Remember this phone" storage

What's next?

- TPM-backed platform authenticator
    - PIN or biometrics
- Passwords, OTPs, and other credentials
    - W3C Verifiable Credentials?
- Keeping up with FIDO specs!

Demo time

<A demo showing a custom-built Firefox patch (submitted upstream for review) using credentialsd via D-Bus for Passkeys handling. This shows a Passkey creation, followed by a Passkey use ceremony, on webauthn.io (a demo website). The UI is the prototype frontend, just for demo purposes - we're not UI developers!>

Get involved!

- **Matrix**
  [`#credentials-for-linux:matrix.org`]
- GitHub
  `linux-credentials`

Special thanks to…

- Related projects
  - mozilla/authenticator-rs
  - kanidm
  - solokeys & trussed
  - nitrokey
  - boltgolt/howdy & fprintd

Special thanks to...

- Core contributors
  - **Isaiah Inuwa** (Bitwarden)
  - **Martin Sirringhaus** (SUSE)

Thank you!
Questions?

Credentials for Linux	github.com/linux-credentials #credentials-for-linux:matrix.org
Alfie Fresta	@afresta:noentropy.org afresta@noentropy.org