Digital Credentials API

W3C CCG • March 2025

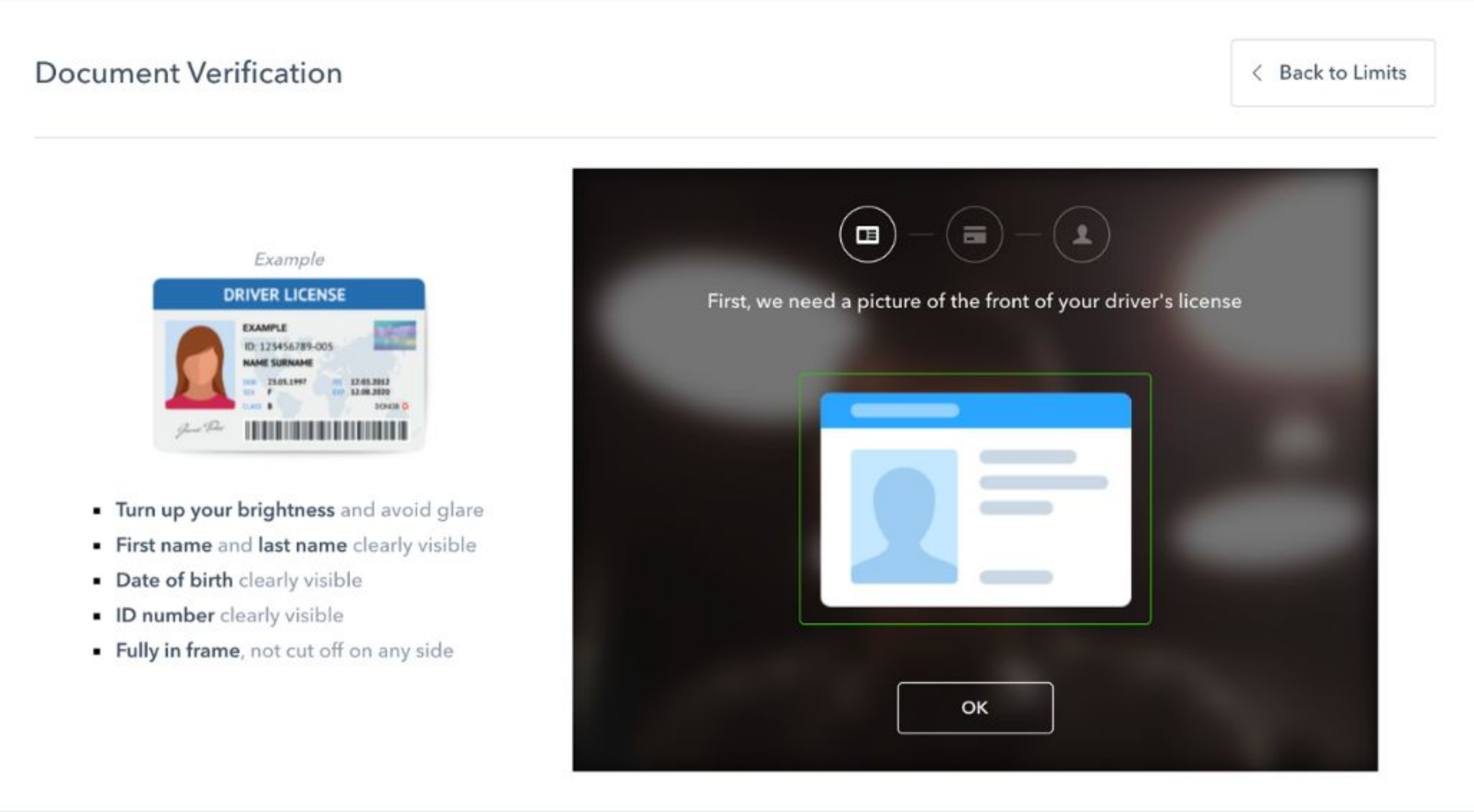
Tim Cappalli • Sam Goto

Agenda

- Background & Design Goals
- Components & Layering
- The API
- Demo
- Work Status
- DC API vs FedCM

Background & Design Goals

The Problem, Gen 1



Document Verification

- **Turn up your brightness** and avoid glare
- **First name** and **last name** clearly visible
- **Date of birth** clearly visible
- **ID number** clearly visible
- **Fully in frame**, not cut off on any side

First, we need a picture of the front of your driver's license

The Problem, Gen 2

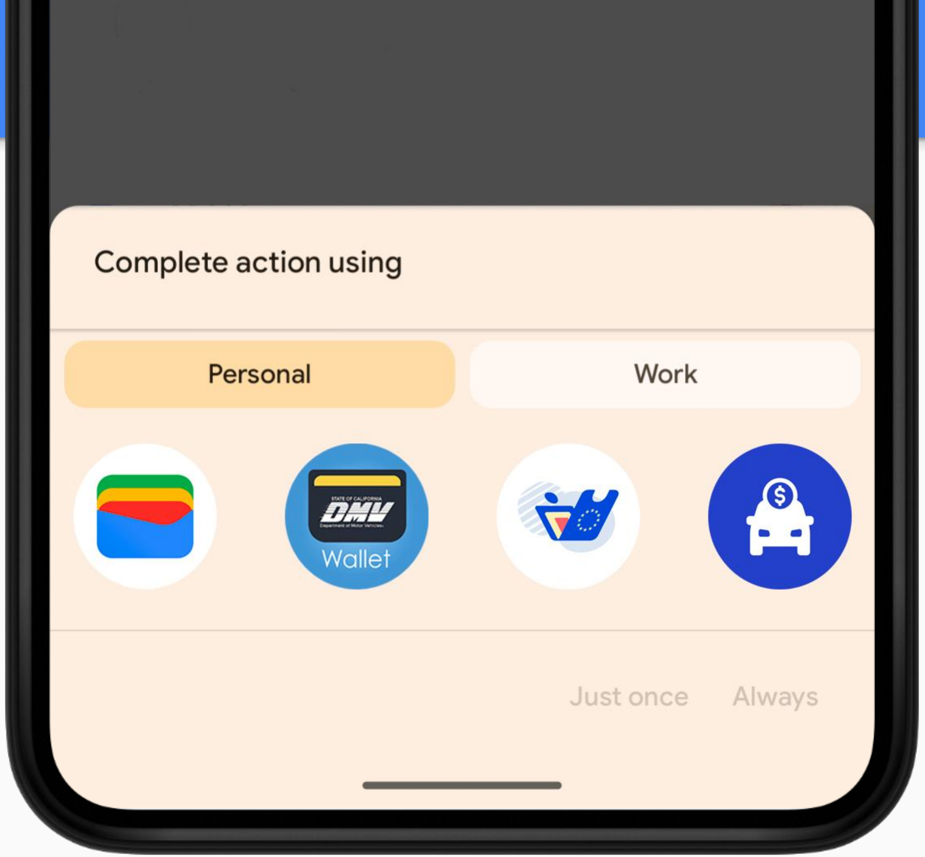
digital credential presentation on the web currently relies on primitives such as ***custom schemes*** *and* ***QR codes*** *which have* ***poor security properties*** *and an even* ***worse user experience***

Custom schemes in the wild

```
mdoc://
openid4vp://
eudi-wallet://
eudi-openid4vp://
mdoc-openid4vp://
openid-credential-offer://
```

Challenges with custom schemes

- invocation from insecure contexts
- on-device phishing via app selection
- no requestor origin / identity
- not standardized & not guaranteed
- context switch during app launch
- no graceful fallback from errors



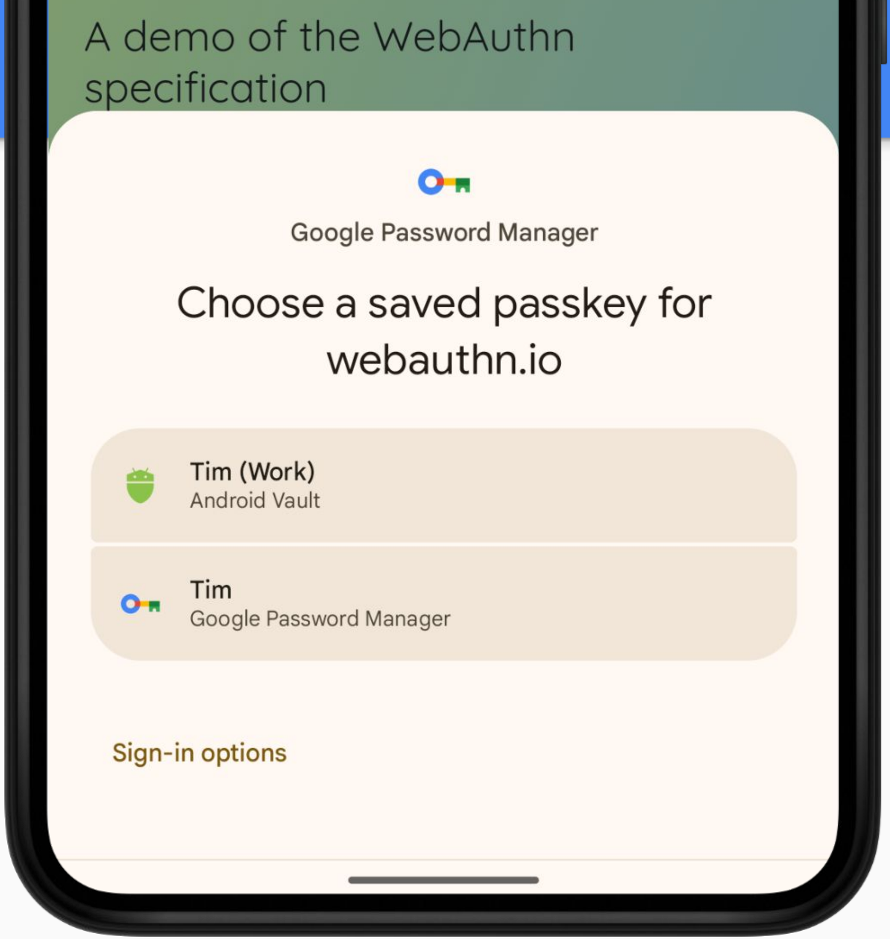
poor UX for credential selection
(users don’t understand wallet selection)

Learnings from passkeys

users think about **accounts** and **credentials**, not **authenticators**

caller context is key

cross-device authentication needs to be **secure, easy, and resistant to phishing**



A demo of the WebAuthn specification

Google Password Manager

Choose a saved passkey for webauthn.io

Tim (Work)
Android Vault

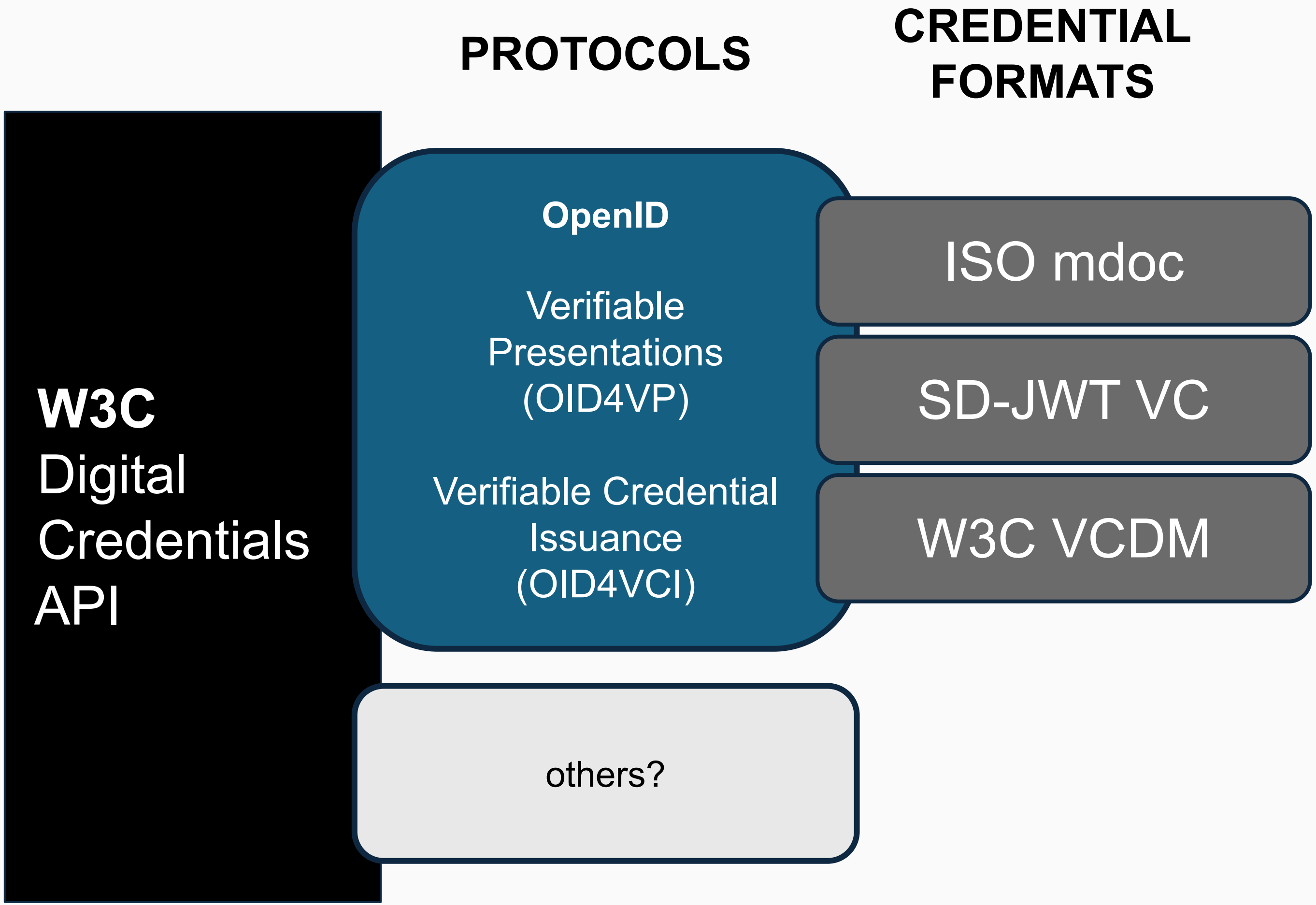
Tim
Google Password Manager

Sign-in options

Design Principles

- Separate the act of requesting from the specific protocol, allowing flexibility in both the protocol and credential formats. This way, the pace of changes in browsers won't hinder progress or block new developments.
- Require request transparency, enabling user-agent inspection for risk analysis
- Assume response opacity (encrypted responses), enabling verifiers and holders to control where potentially sensitive PII is exposed
- Prevent website from silently querying for the availability of digital credentials and communicating with credential providers without explicit user consent

Components & Layering



PROTOCOLS

CREDENTIAL FORMATS

W3C Digital Credentials API

OpenID

Verifiable Presentations (OID4VP)

Verifiable Credential Issuance (OID4VCI)

others?

ISO mdoc

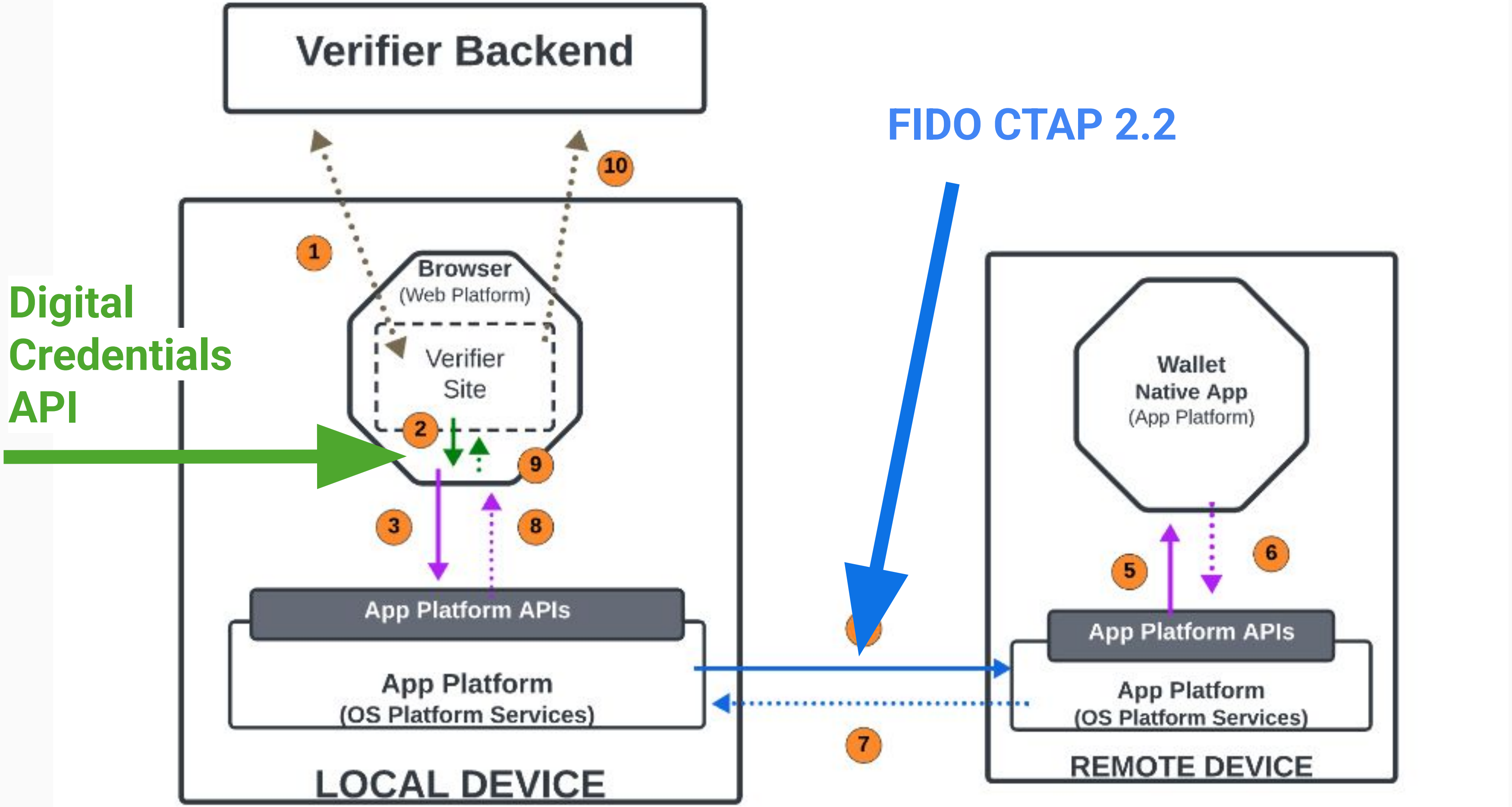
SD-JWT VC

W3C VCDM

Relationship to other protocols and specifications

Roles and Responsibilities

Browser (web platform)	**OS Platform** (app platform)	**Credential Provider** (app/wallet)
<<<<<< Permission >>>>>>		Holder consent
API surface	Credential selector (presentation)	Holder verification
Basic request validation	Provider selector (issuance)	Presentation & Issuance Protocols (verifier / RP authentication, selective disclosure, signing, encryption)
Secure context validation	Cross-device transport	
Interaction with OS platform	Native app requests	Key management

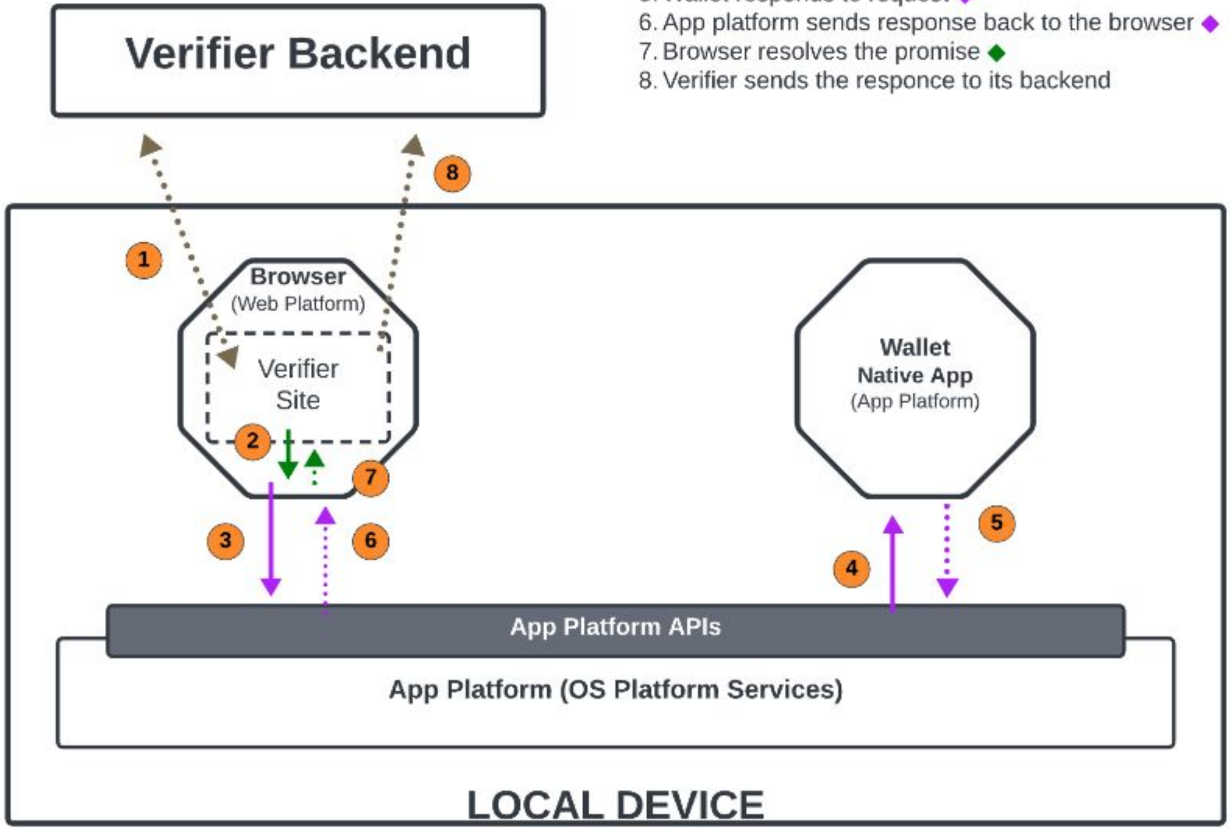


Digital Credentials API

FIDO CTAP 2.2

SCENARIO
same-device
web-based verifier
native app wallet

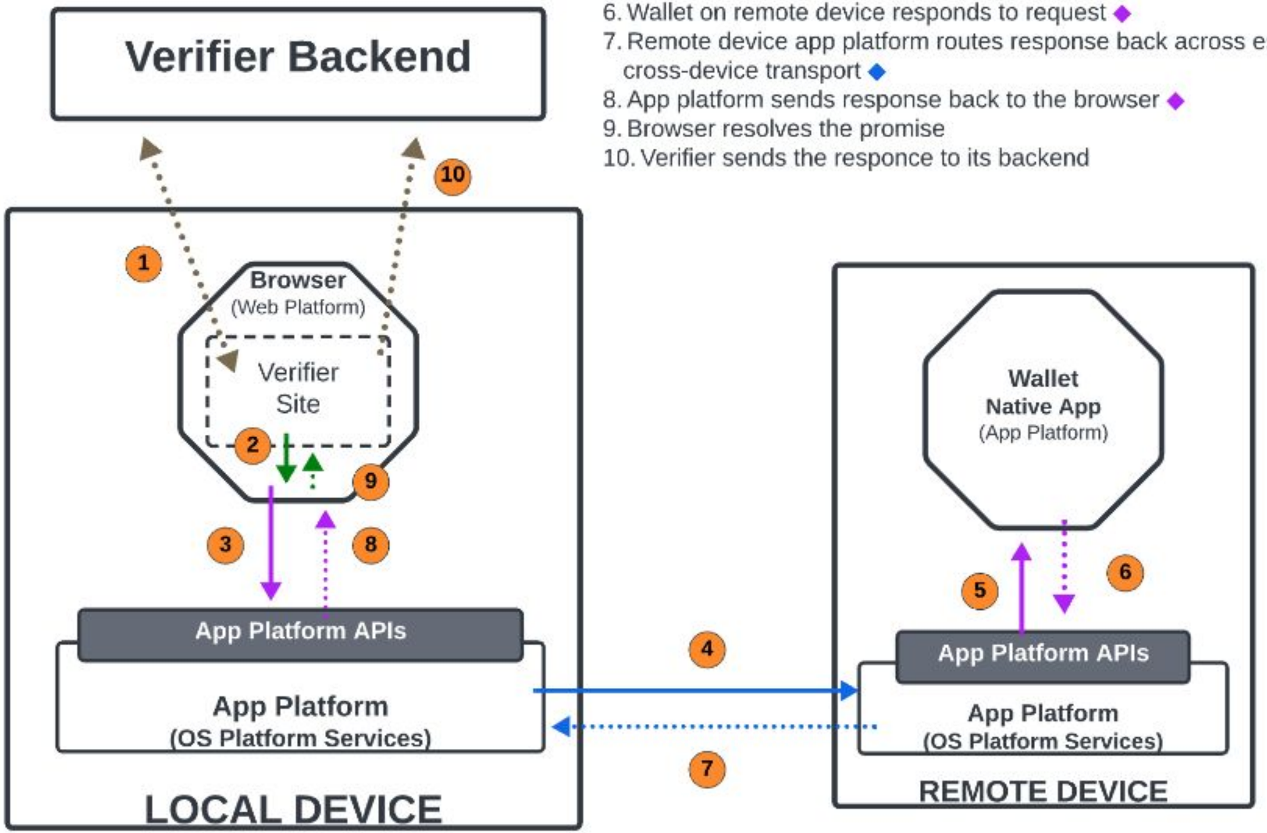
1. Verifier site loaded in browser, request initiated
2. Web platform API request initiated ◆
3. Browser processes request and routes to the app platform ◆
4. App platform processes request and routes to wallet ◆
5. Wallet responds to request ◆
6. App platform sends response back to the browser ◆
7. Browser resolves the promise ◆
8. Verifier sends the responce to its backend



SCENARIO

cross-device
web-based verifier
native app wallet

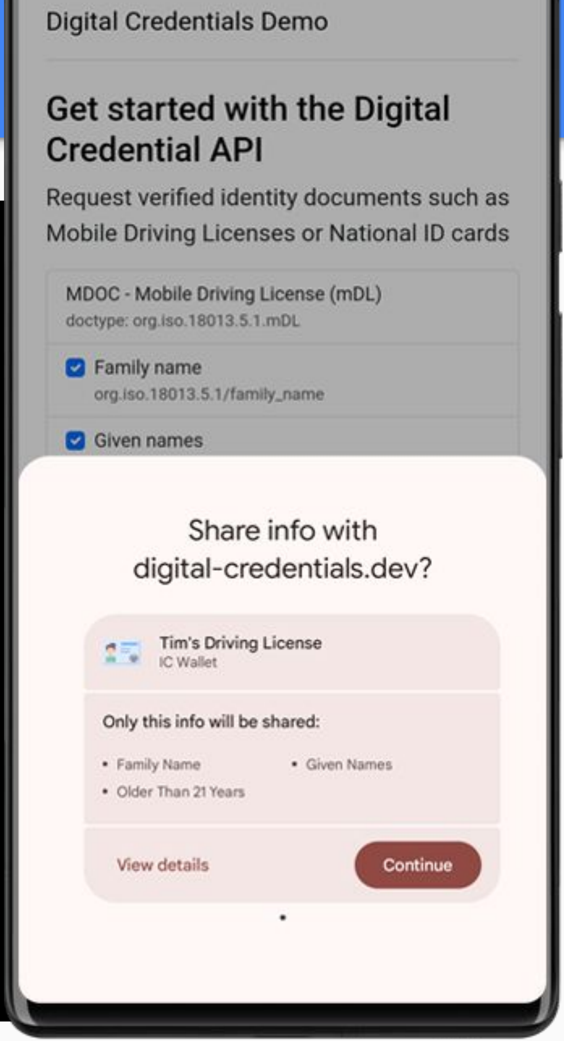
1. Verifier site loaded in browser, request initiated
2. Web platform API request initiated ◆
3. Browser processes request and routes to the app platform ◆
4. App platform processes request and initiates cross-device transport ◆
5. Remote device app platform processes request and routes to wallet ◆
6. Wallet on remote device responds to request ◆
7. Remote device app platform routes response back across established cross-device transport ◆
8. App platform sends response back to the browser ◆
9. Browser resolves the promise
10. Verifier sends the responce to its backend



The API

Presentation

```
let cred = await
  navigator.credentials.get({
     signal: controller.signal,
     digital: {
       requests: [{
         protocol: "openid4vp-v1-unsigned",
         data: { ...request }
       }]
     }
  });
```



Digital Credentials Demo

Get started with the Digital Credential API

Request verified identity documents such as Mobile Driving Licenses or National ID cards

MDOC - Mobile Driving License (mDL)
doctype: org.iso.18013.5.1.mDL

☑ Family name
org.iso.18013.5.1/family_name

☑ Given names

Share info with digital-credentials.dev?

Tim's Driving License
IC Wallet

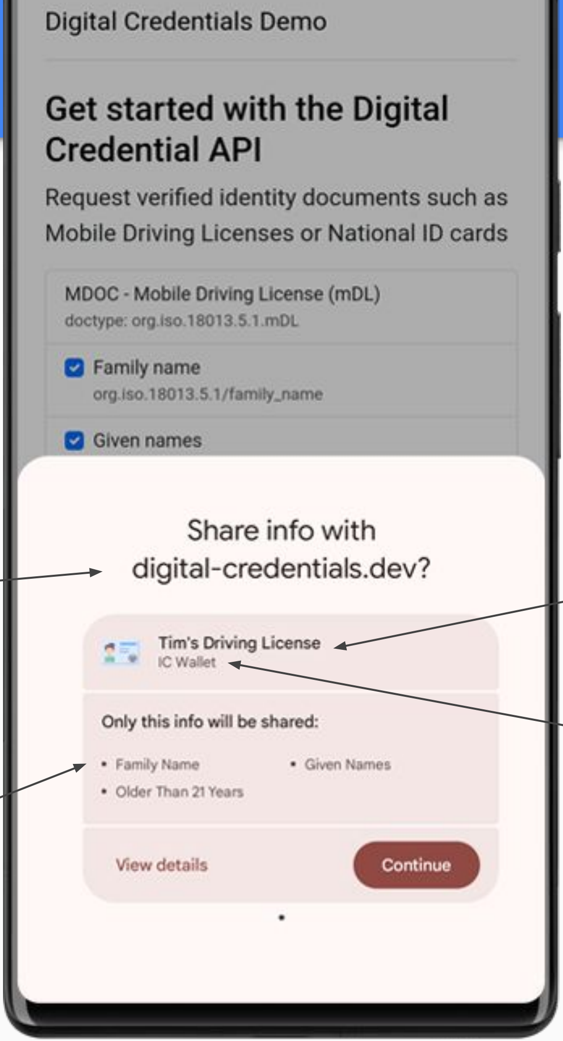
Only this info will be shared:

- Family Name
- Given Names
- Older Than 21 Years

View details

Continue

Presentation



Digital Credentials Demo

Get started with the Digital Credential API

Request verified identity documents such as Mobile Driving Licenses or National ID cards

MDOC - Mobile Driving License (mDL)
doctype: org.iso.18013.5.1.mDL

☑ Family name
org.iso.18013.5.1/family_name

☑ Given names

Share info with digital-credentials.dev?

Tim's Driving License
IC Wallet

Only this info will be shared:

- Family Name
- Given Names
- Older Than 21 Years

Who is asking?

credential-centric

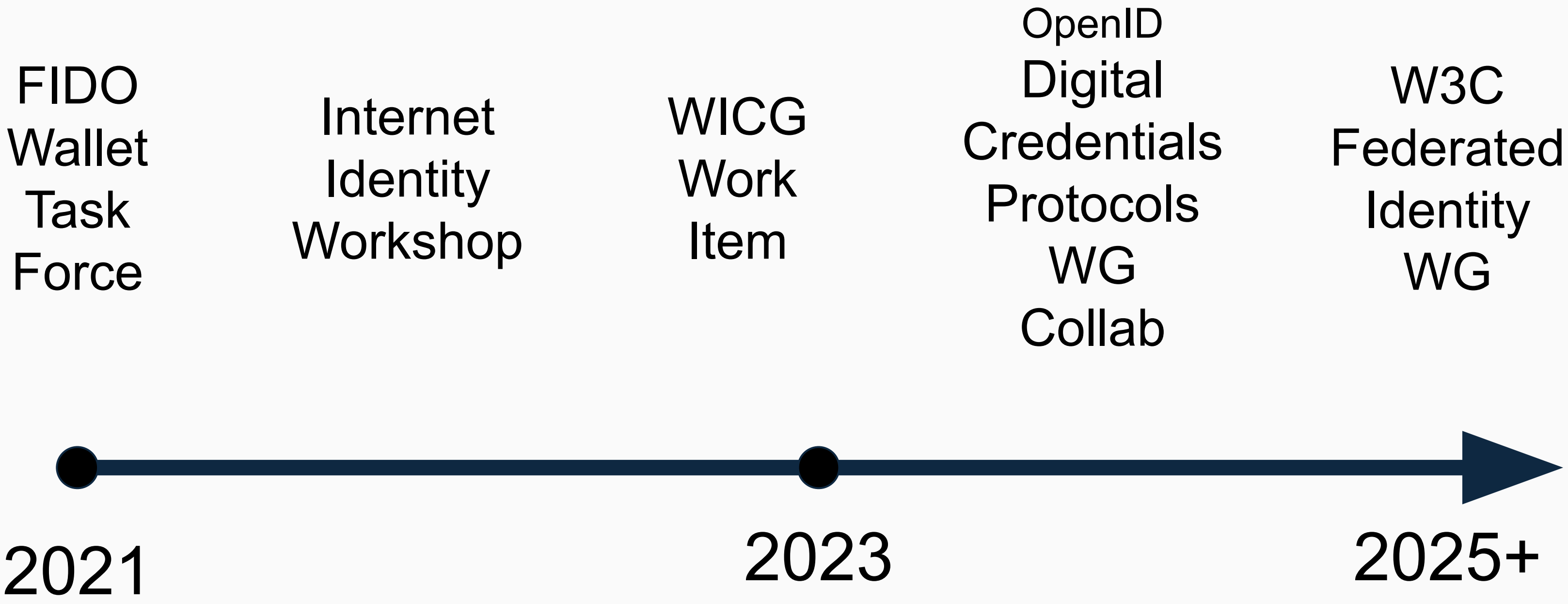
Who can provide it?

What are they asking for?

Demo

Work Status

Evolution



FIDO Wallet Task Force

Internet Identity Workshop

WICG Work Item

OpenID Digital Credentials Protocols WG Collab

W3C Federated Identity WG

2021

2023

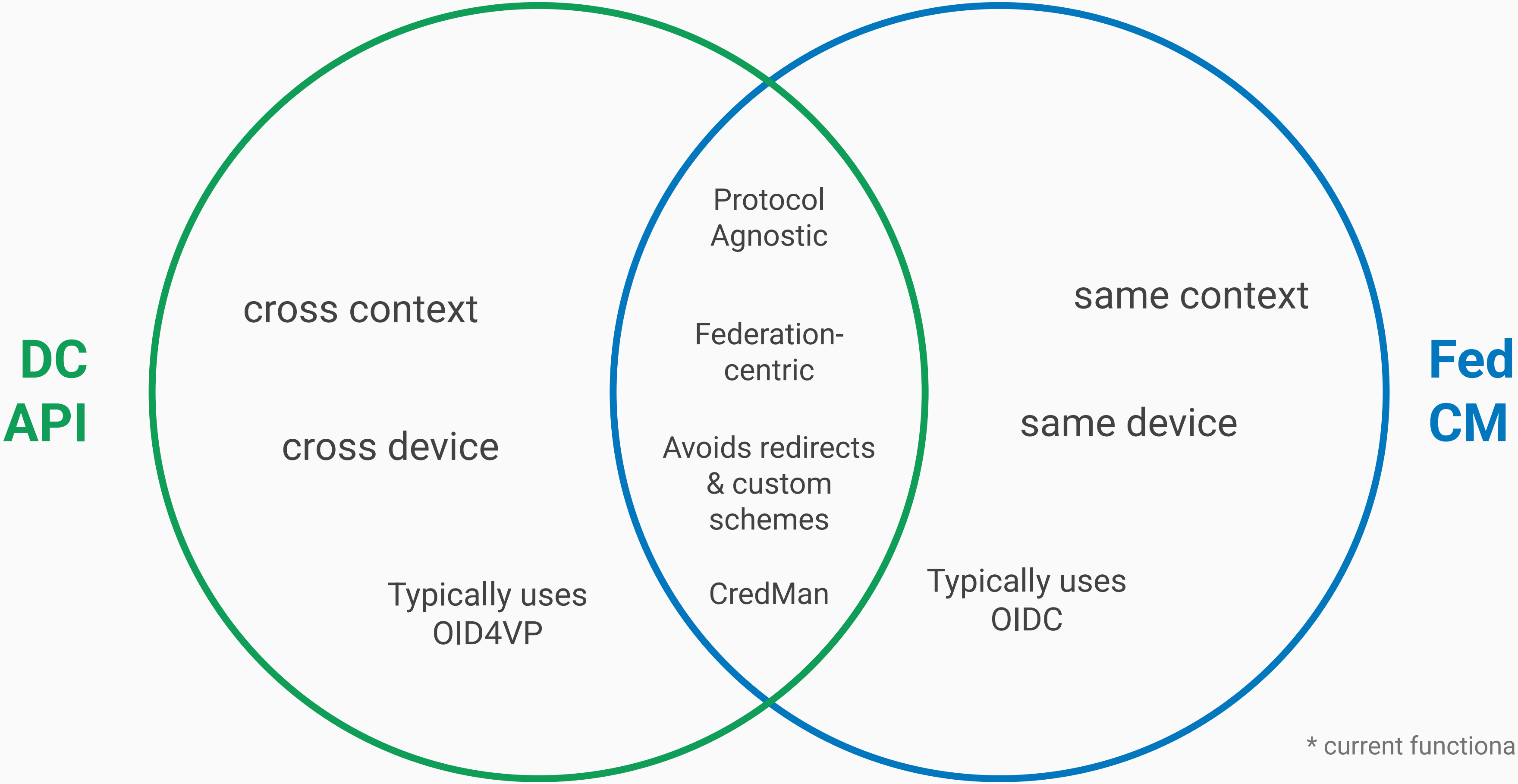
2025+

Current Major Items

- Define issuance (`credentials.create`)
- Protocol registry criteria
- Migration to Federated Identity WG and FPWD

DC API vs FedCM API

DC API vs FedCM API* (Presentation Only)



DC API

cross context

cross device

Typically uses OID4VP

Protocol Agnostic

Federation-centric

Avoids redirects & custom schemes

CredMan

Fed CM

same context

same device

Typically uses OIDC

* current functionality

Discussion