Digital Credentials API

NCCoE Mobile Driver's License (mDL) Webinar

June 2024

Lee Campbell

Identity and Authentication Lead, Android
Google

Tim Cappalli

Sr. Architect, Identity Standards
Okta

https://timcappalli.me

Digital Credentials API

- Background
- Demo
- Components
- The Digital Credentials API
- Cross-Device Presentation
- Issuance
- Q&A

Background

The problem

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***

What is a custom URI scheme?

A custom identifier that an app can register with an operating system with the goal of being invoked from other contexts, such as other apps or from the web.

In many cases, these identifiers are not globally unique, and may be shared.

CUSTOM SCHEMES IN THE WILD

`mdoc://`

`openid4vp://`

`eudi-wallet://`

`eudi-openid4vp://`

`mdoc-openid4vp://`

`openid-credential-offer://`

Issues w/ custom schemes

invocation from insecure contexts

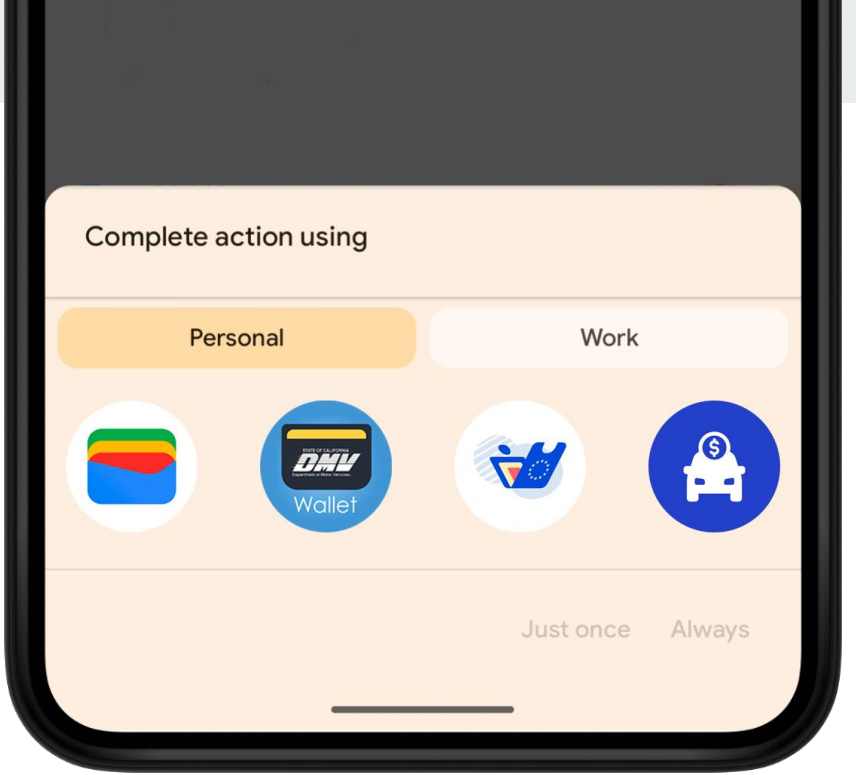
on-device phishing via app selection

no requestor origin / identity

not standardized & not guaranteed

context switch during app launch

no graceful fallback for errors



poor UX for credential selection

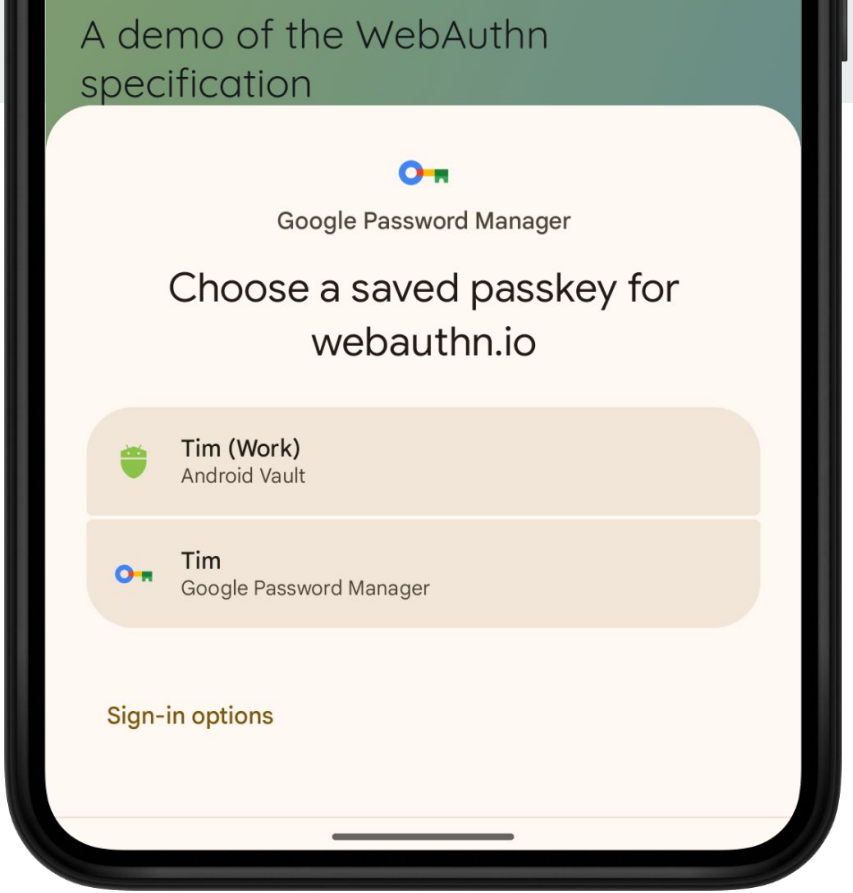
(*users don't understand wallet selection*)

Learnings from passkeys

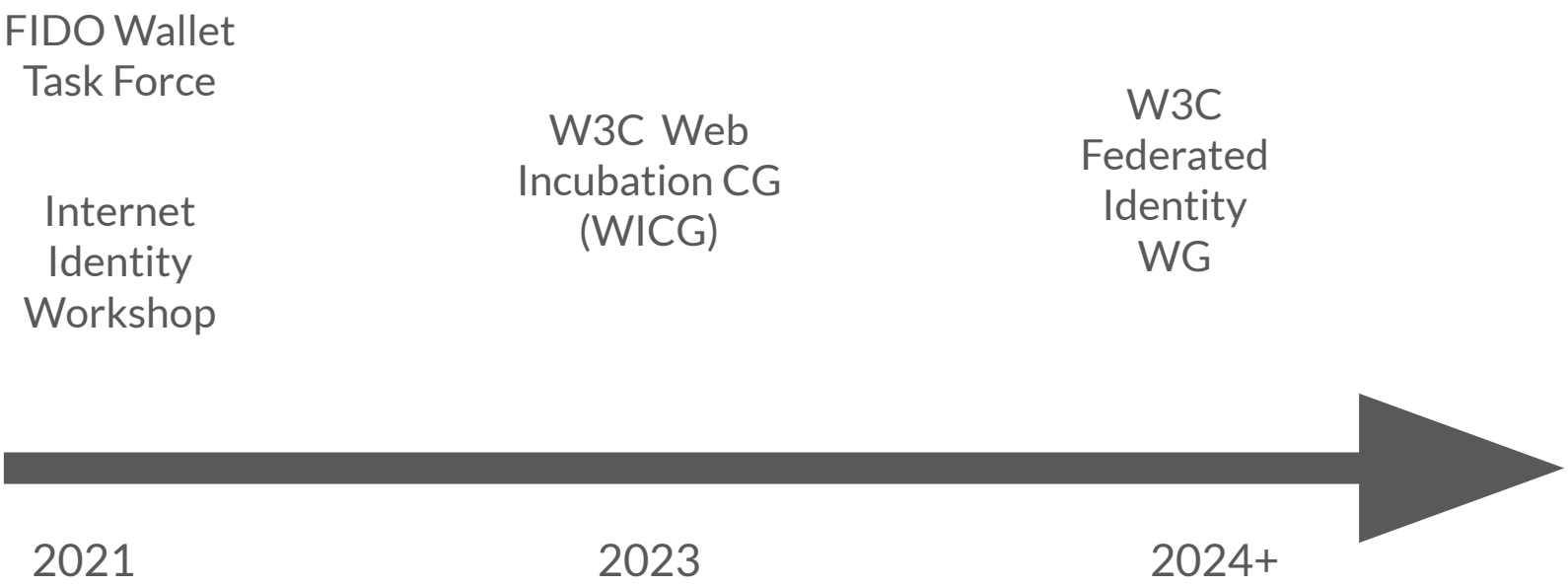
users think about **accounts** and **credentials**, not **authenticators**

caller context is key

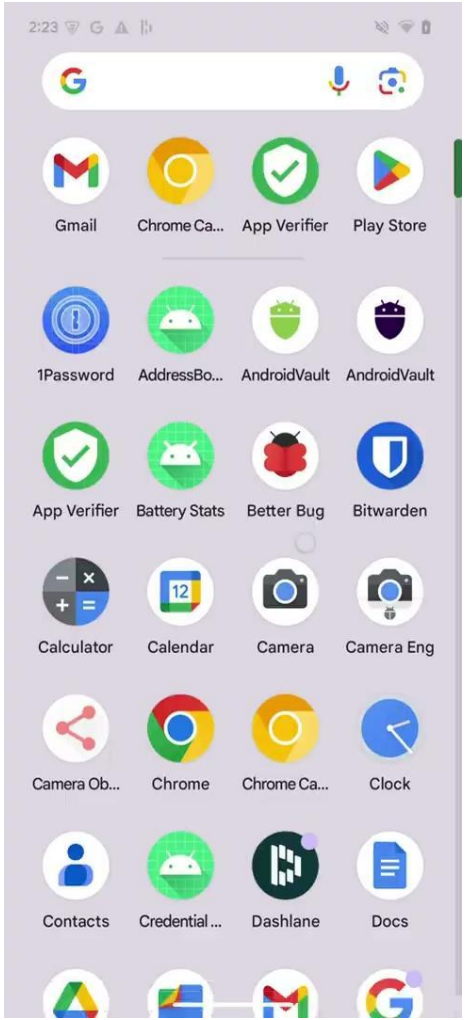
cross-device authentication needs to be **secure**, **easy**, and **resistant to phishing**

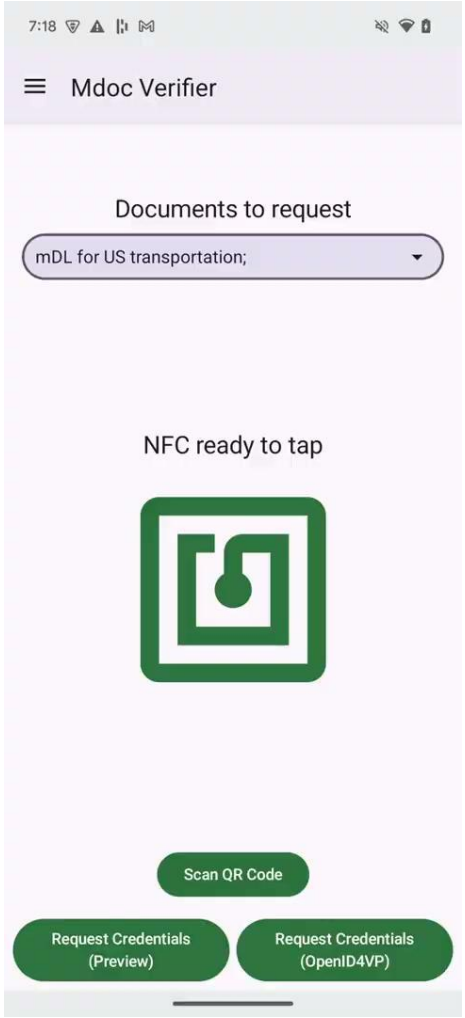


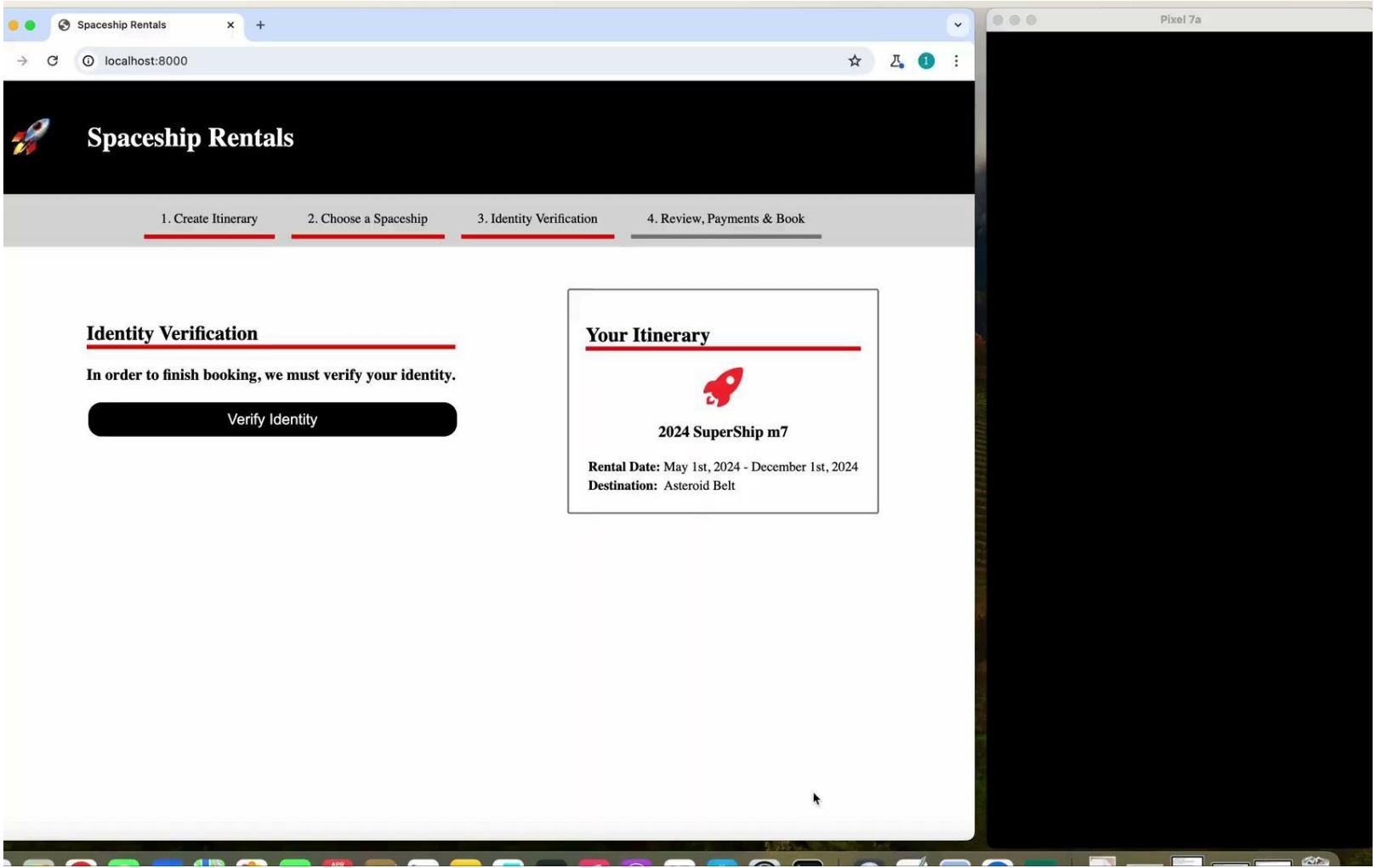
Past, Present, and Future



Demo


2:23
Gmail
Chrome Ca...
App Verifier
Play Store
1Password
AddressBo...
AndroidVault
AndroidVault
App Verifier
Battery Stats
Better Bug
Bitwarden
Calculator
Calendar
Camera
Camera Eng
Camera Ob...
Chrome
Chrome Ca...
Clock
Contacts
Credential ...
Dashlane
Docs


7:18
Mdoc Verifier
Documents to request
mDL for US transportation;
NFC ready to tap
Scan QR Code
Request Credentials (Preview)
Request Credentials (OpenID4VP)


Spaceship Rentals
localhost:8000
Spaceship Rentals
1. Create Itinerary
2. Choose a Spaceship
3. Identity Verification
4. Review, Payments & Book
Identity Verification
In order to finish booking, we must verify your identity.
Verify Identity
Your Itinerary
2024 SuperShip m7
Rental Date: May 1st, 2024 - December 1st, 2024
Destination: Asteroid Belt
Pixel 7a

Components

Components: Same Device

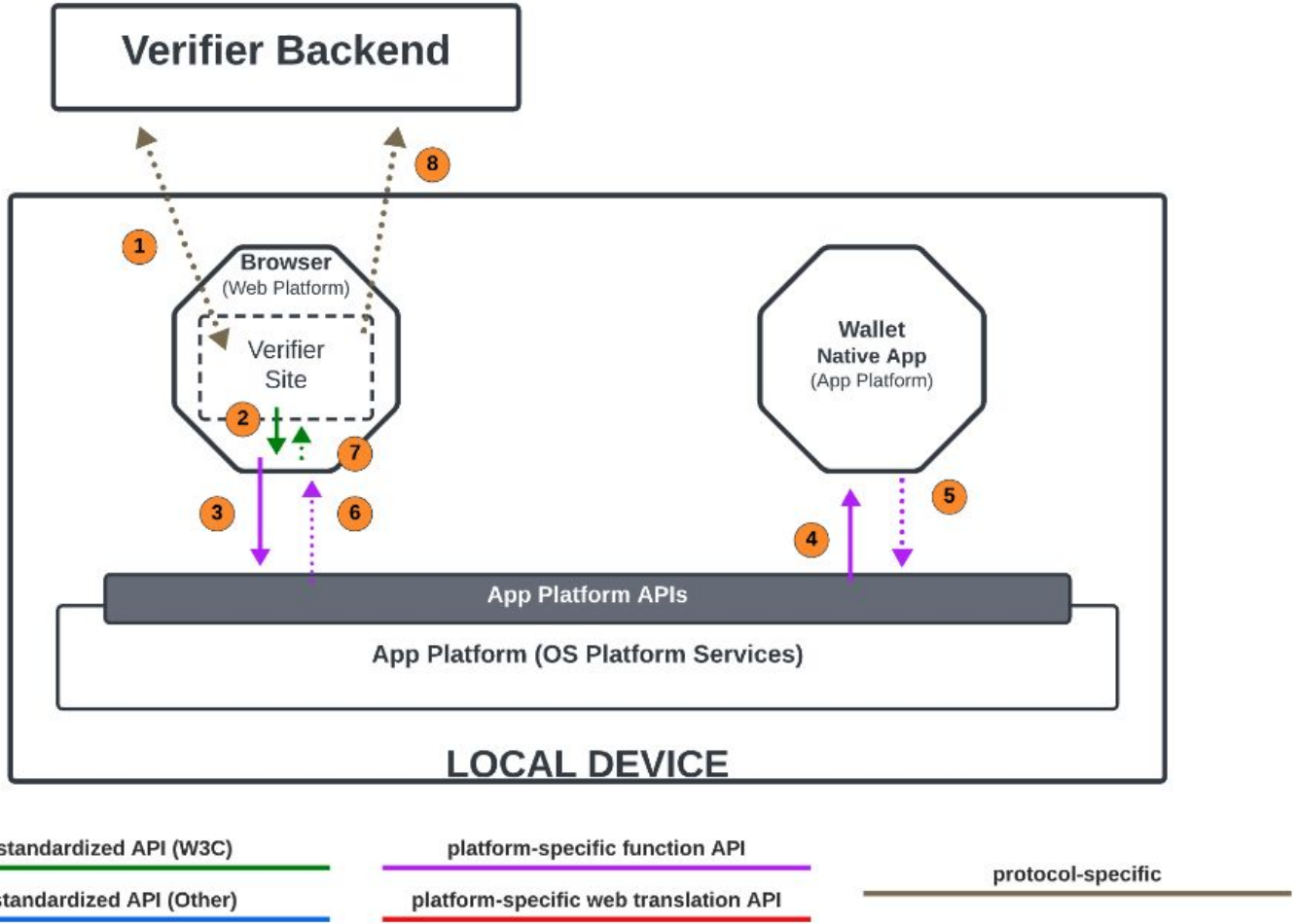
Verifier: website or native app

Client: web browser or app instance

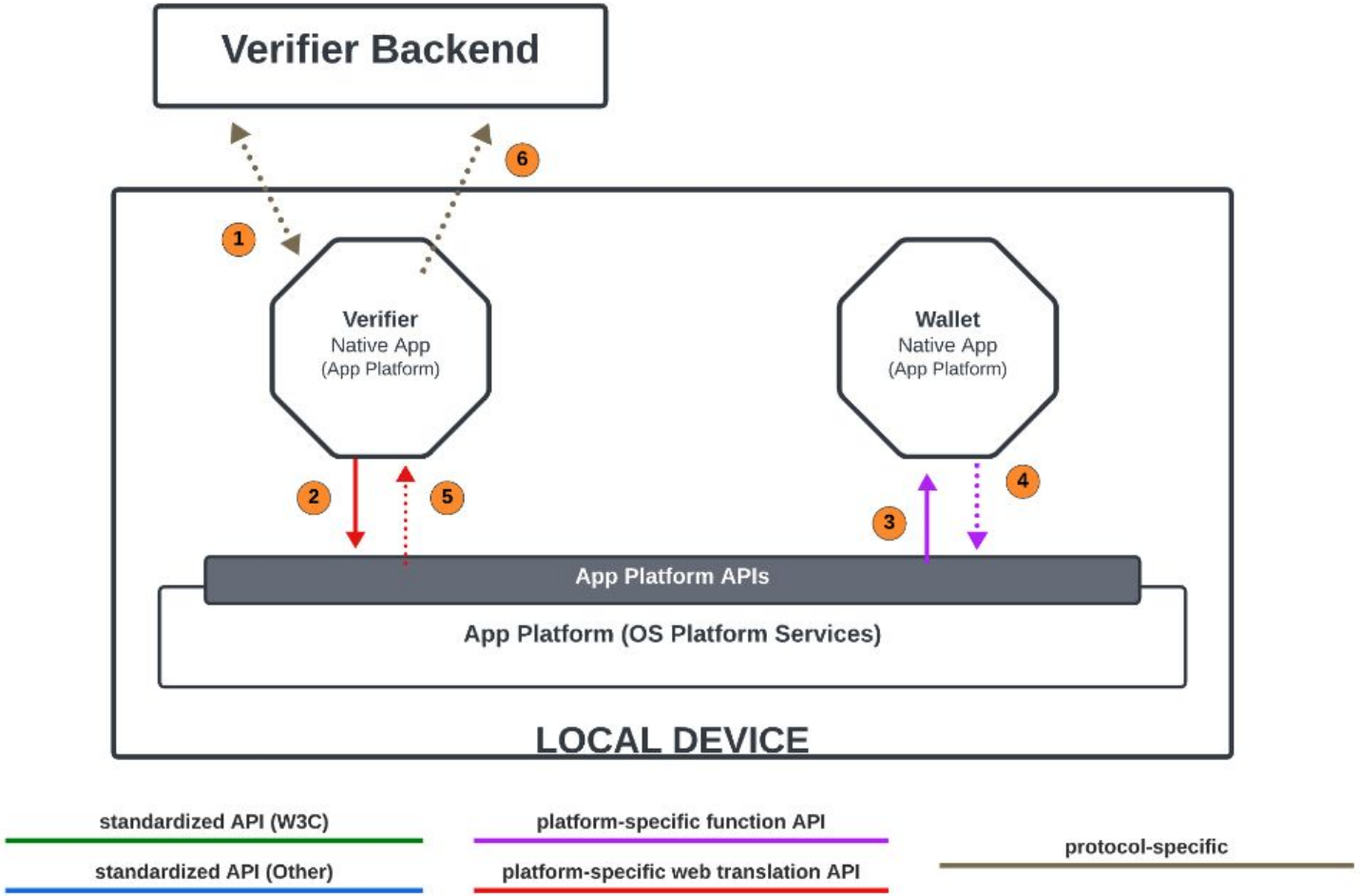
App Platform: underlying OS

Identity Wallet: native app

Layers: Same Device (Web Verifier)



Layers: Same Device (App Verifier)



Components: Cross-Device

Verifier: website or native app

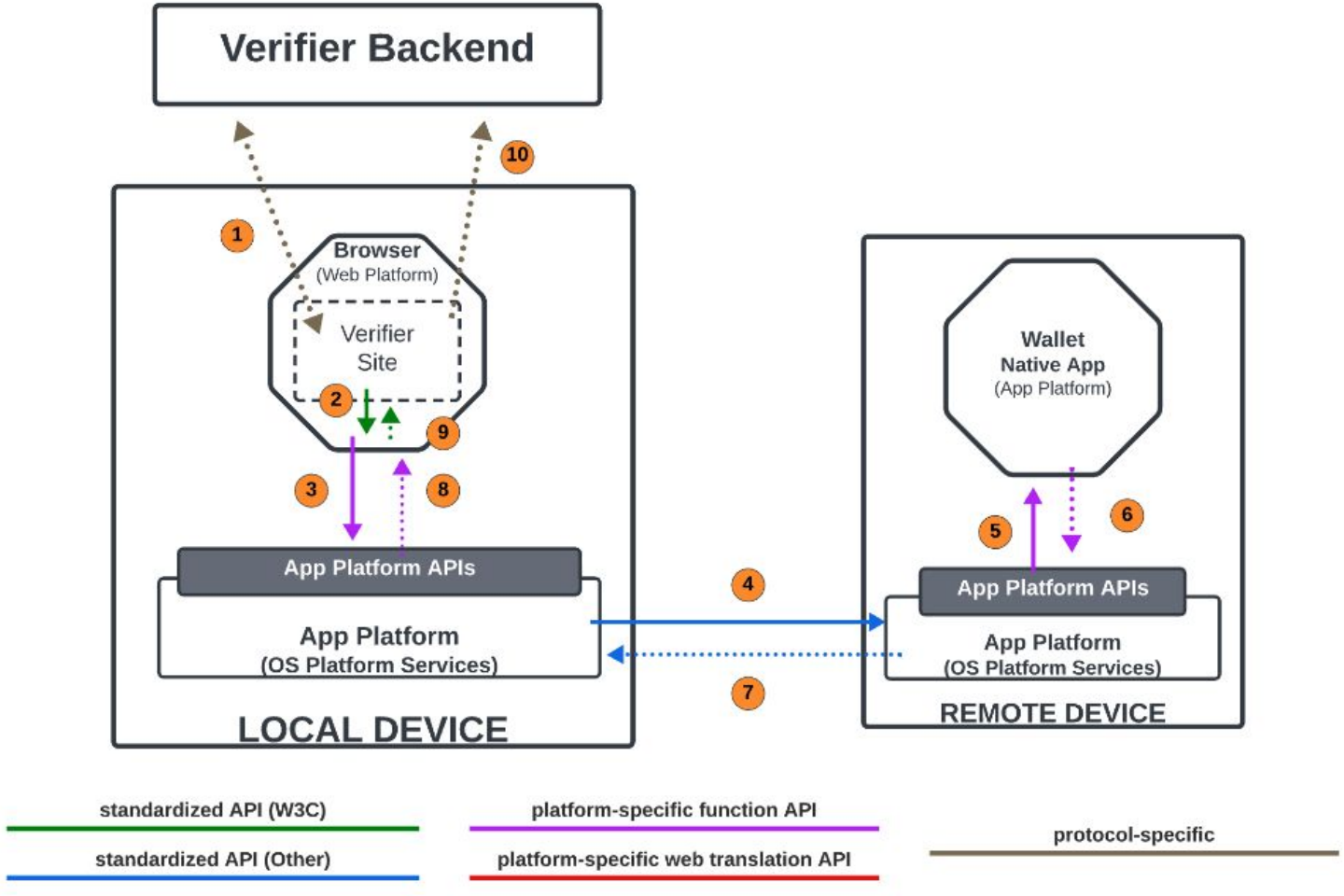
Local Client: web browser or app instance

Local App Platform: underlying OS on calling device

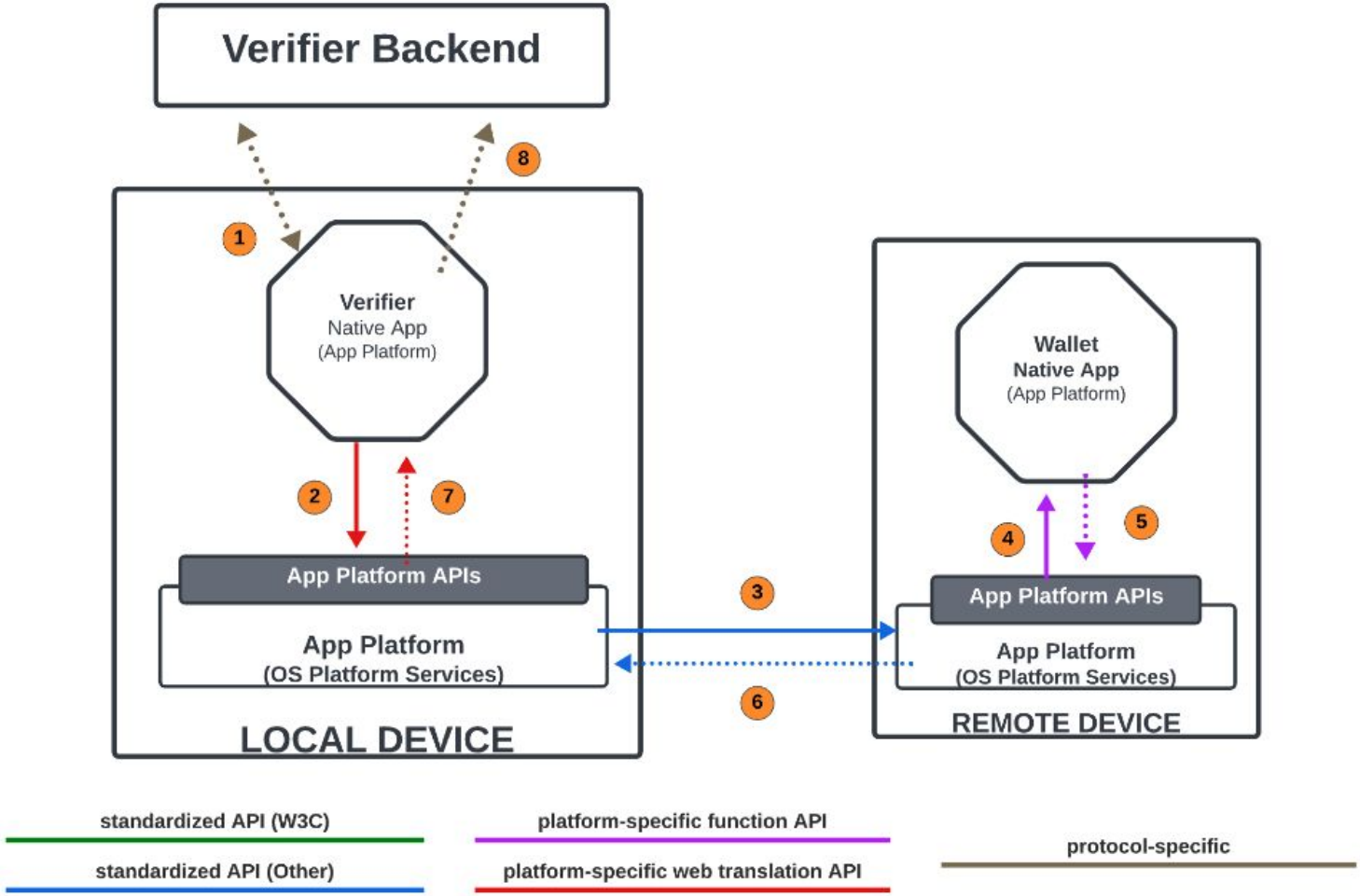
Remote App Platform: underlying OS on remote device

Remote Identity Wallet: native app on remote device

Layers: Cross-Device (Web Verifier)



Layers: Cross-Device (App Verifier)



The API

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 wallet providers without explicit user consent

The API


```
const presentation = await navigator.identity.get({
  digital: {
    providers: [{
      protocol: "urn:openid.net:oid4vp",
      request: {
        client_id: "client.example.org",
        client_id_scheme: "entity_id",
        expected_origins: ["https://verify1.example.com"],
        response_type: "vp_token",
        nonce: "n-0S6_WzA2Mj",
        client_metadata: { jwks: {} },
        presentation_definition: {
          id: "mDL-Request", input_descriptors: [ ... ], ... }
      }
    }]
  }; });

const credential = presentation.data;
```

Cross-Device Presentation

Cross-Device Presentation

- FIDO CTAP 2.2 with hybrid transports
- Implemented by the OS platform (transparent to the wallet and verifier)
- QR code not required after linking
- Potential for metadata-like sync in the future

Issuance

Issuance

- Currently out of scope for initial version
- API being designed with it in mind

Get Involved

Get Involved

Discussion is currently via the W3C Web Platform Incubation CG (WICG)

You **do not need to be a W3C member**, but you do need to create a W3C account and accept the terms:

`w3.org/groups/cg/wicg`

`github.com/wicg/digital-credentials`

meeting details here ^^

Get Involved

Prototype with Android and Chrome!

Instructions:

Short link: **`tcslides.link/dc-androidprotoype`**

Full link

Q&A