



Google's approach to Digital Credentials

Global Platform Digital Wallet Seminar
Oct 16, 2025 – Brussels, Belgium



David Zeuthen (zeuthen@google.com)

Agenda

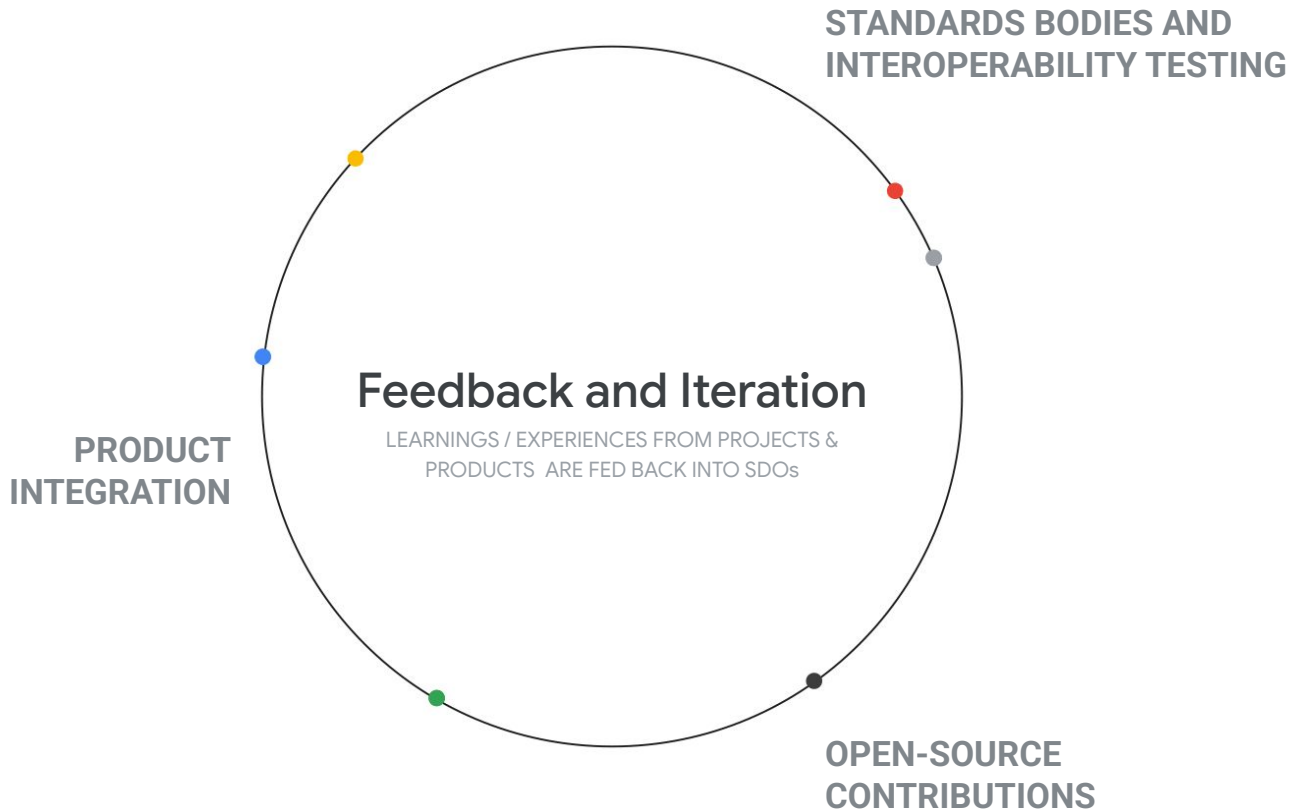
- **Our involvement in Digital Credentials**
- **SDKs and Platforms**
- **Google Wallet**
- **Takeaways**

Why Google is involved in Digital Credentials

- **We have a digital wallet**
 - Google Wallet has been available on Android for a long time
 - Digital Identity is one of the top requested features in Google Wallet
- **We're a relying party**
 - For KYC, among other things
 - Digital Credentials are often safer and easier to accept than paper and plastic cards
- **To protect our users**
 - If specified and/or implemented poorly, Digital Credentials can actively harm the security, privacy, and safety of users
 - On the other hand, if specified carefully, Digital Credentials can significantly enhance the security, privacy, and safety of users
- **We're an identity provider and credential issuer**
 - For example Sign in with Google and Google Wallet ID Pass

How Google is involved in Digital Credentials

- **Participation in SDOs and other open groups**
 - Contributing member of ISO SC17 WG10 since 2018, working on mDLs
 - Active in Global Platform, OpenID Foundation, W3C, IETF, FIDO, and many other places
 - EWC, WE BUILD large scale pilots, OpenWallet Foundation
- **Open-source Software contributions**
 - Multipaz SDK for Digital Credentials
 - W3C Digital Credentials API, AOSP, Chromium, Longfellow ZK, Ready SE Alliance
- **Product integration**
 - Android
 - Chrome
 - Google Wallet
- **Public / Private partnerships**



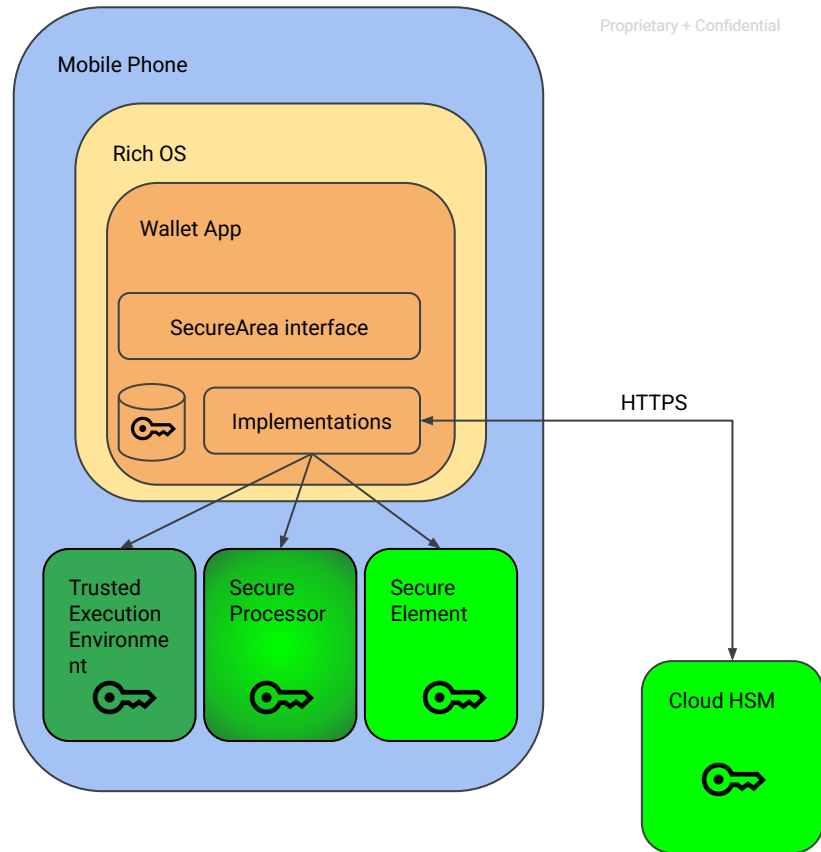


Multipaz



Multipaz

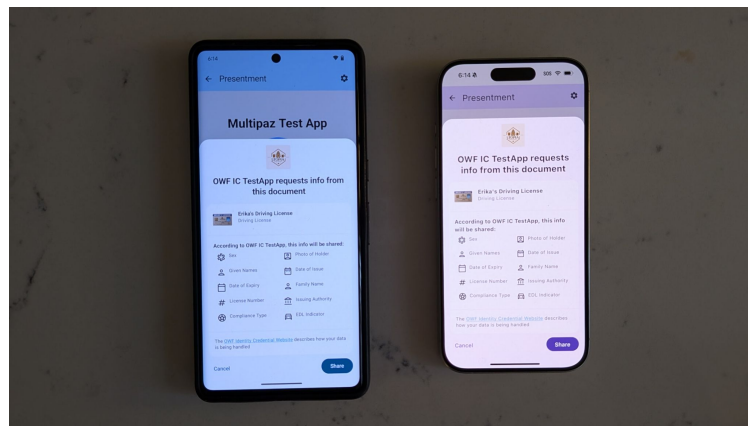
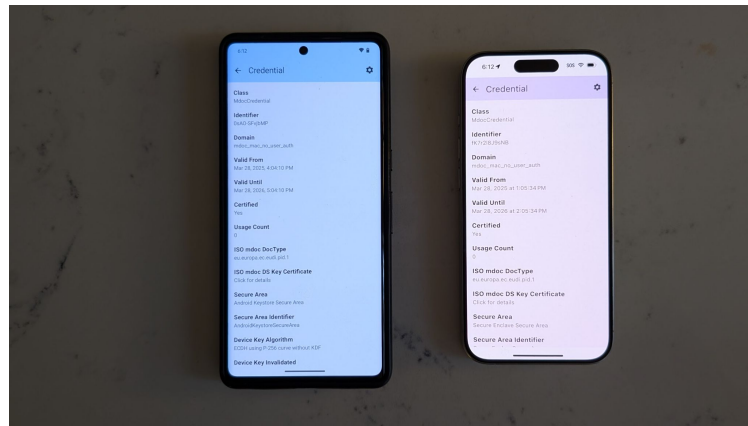
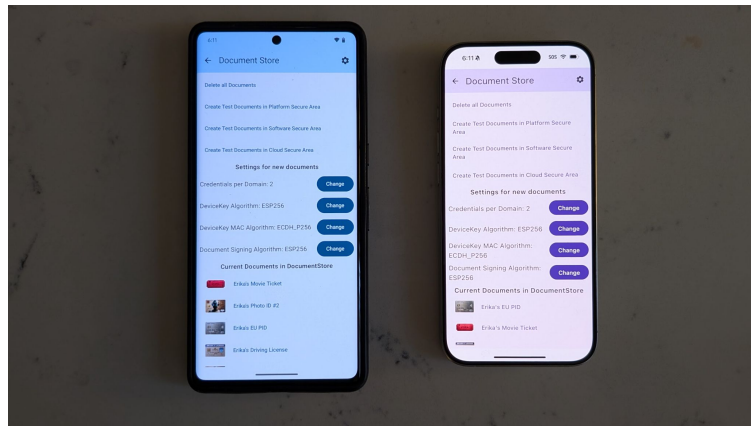
- **Started by Google in 2019, now hosted at the OpenWallet Foundation**
 - Originally focus was ISO/IEC 18013-5:2021 on Android
 - <https://github.com/openwallet-foundation/multipaz>
- **Libraries providing building blocks for all three parties in the ecosystem**
 - In scope for wallet apps, reader apps/systems, issuer systems
 - Works on Android, iOS, and server side, via Kotlin Multiplatform
 - Includes reference apps showing how to use the libraries
- **Used in production apps, including Google Wallet and EUDIW Reference**
- **Provides SecureArea interface with multiple implementations**
 - TEE and SE on Android (via Android Keystore and KeyMint/StrongBox)
 - iOS Secure Enclave (via CryptoKit)
 - Software-backed (for low-value portable credentials)
 - Cloud Secure Area (when device has no suitable security chip)
- **Provides primitives for user authentication for use of key material**
 - Integration with Rich OS (BiometricPrompt on Android, LAContext on iOS)
 - Prompts for asking for passphrase/PINs
- **Many other APIs**
 - Provisioning, Presentment, Consent Dialogs, etc
- **Time-based release schedule (every 1-2 months)**



Proprietary + Confidential

Multipaz - Multiplatform Support

Proprietary + Confidential




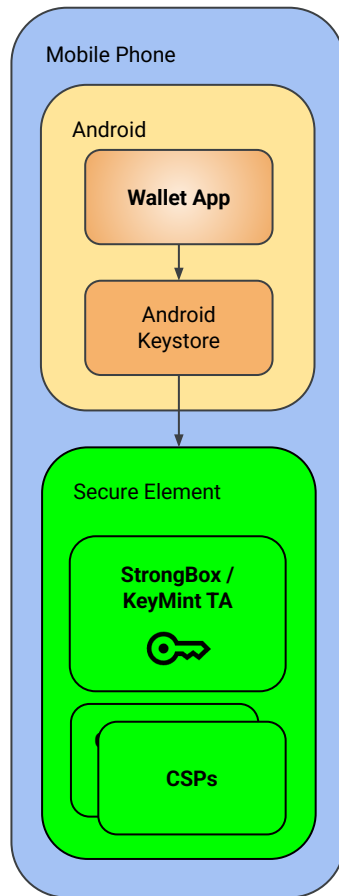
Multipaz demo - W3C DC API and Cloud Secure Area



Hardware-backed Keystore

Android's approach to LoA High

- **Android has had a fully-featured Rich OS API for HW-backed keystore services for a long time:**
 - It's called Android Keystore and been available in devices shipping with Android 8.1 (Dec 2017) or later
 - Supports ECC w/ curve P-256 and ECDSA and ECDH (on newer devices)
 - Rich integration with the OS (phone-wide password/pattern/PIN or biometrics to unlock keys)
 - **No separate wallet or credential PIN necessary**
 - Key attestation with secure provisioning of attestation keys
 - Designed so integrity is maintained even when Rich OS is compromised
 - Implemented in TEE with optional "StrongBox" variant for SE (as per [Android CDD 9.11.2](#))
 - StrongBox Requirements
 - Hardware and Firmware must be resistant to High Attack Potential
 - Don't currently have any certification requirement about the StrongBox TAs
 - Moving to having StrongBox/KeyMint TAs sit on top of CSP which will give e.g. AVA_VAN.5 
- **Our goal is to make it as frictionless as possible for app developers to write LoA High applications**
 - If the app is using basic cryptography (like e.g. P-256 keys and ECDSA) like used for e.g. ISO mdoc and IETF SD-JWT, we believe Android Keystore w/ StrongBox is suitable.
 - If the device lacks StrongBox (e.g. no Secure Element available), Cloud HSM can be used.
 - **This way the app developer will not have to worry about developing a TA or its distribution**
- **Apps with exotic cryptography needs (e.g. digital currency) may not be able to use current Strongbox**
 - In this case the application will need to write and distribute its own TA



Credential Manager and Chrome

Our goals

Create **simple, secure, and privacy preserving** ways for users to store and share credentials using the **wallet of their choice**



- 1 Support all wallet developers with open, private, and secure foundations to store IDs



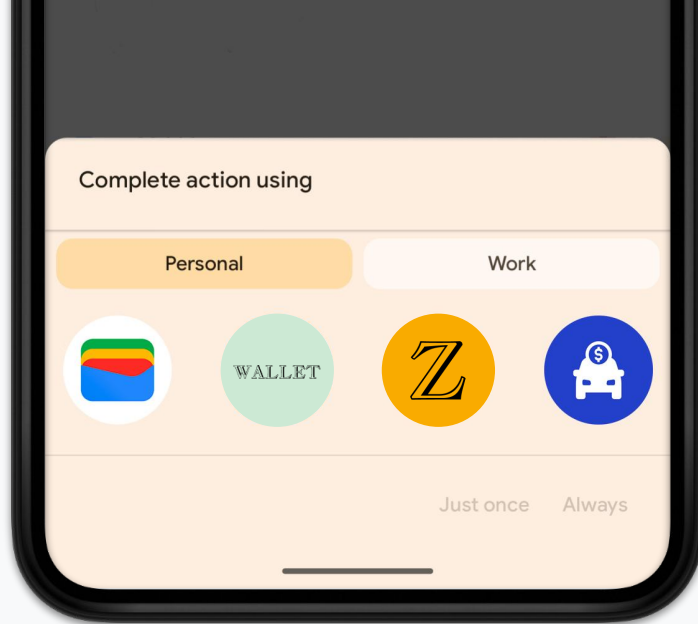
- 2 Enable open and standards-based ecosystem for **online presentment**

Focus today



The problem

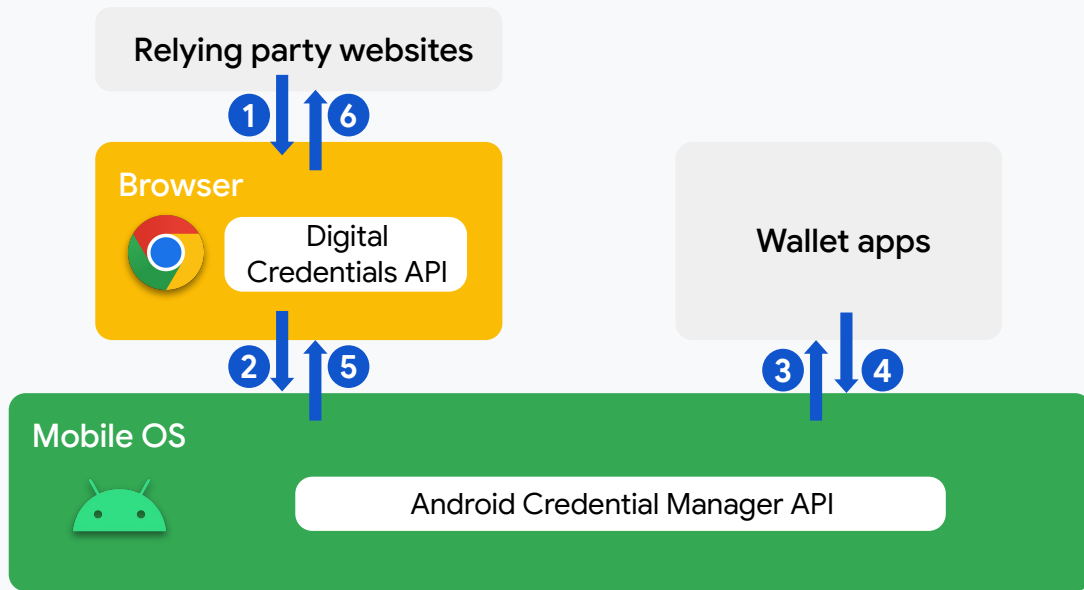
Digital credential presentation on the web currently relies on primitives such as **custom schemes** which have **poor security properties** and **user experience**



- Confusing app selection
- No requester info displayed
- App switching after selection
- No graceful fallback for errors

The solution

Purpose-built browser and OS APIs for online presentation



The benefits

The browser and OS are uniquely positioned to enable online presentment



Scale



**Privacy and
security**

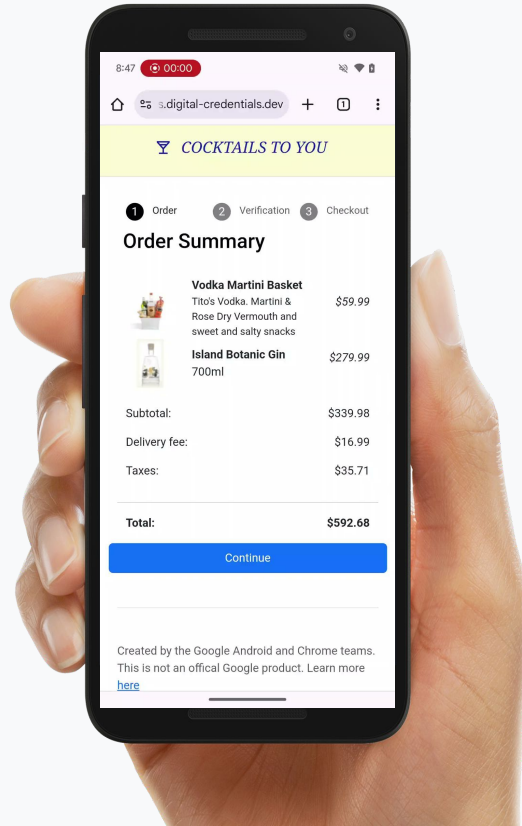


**User
experience**

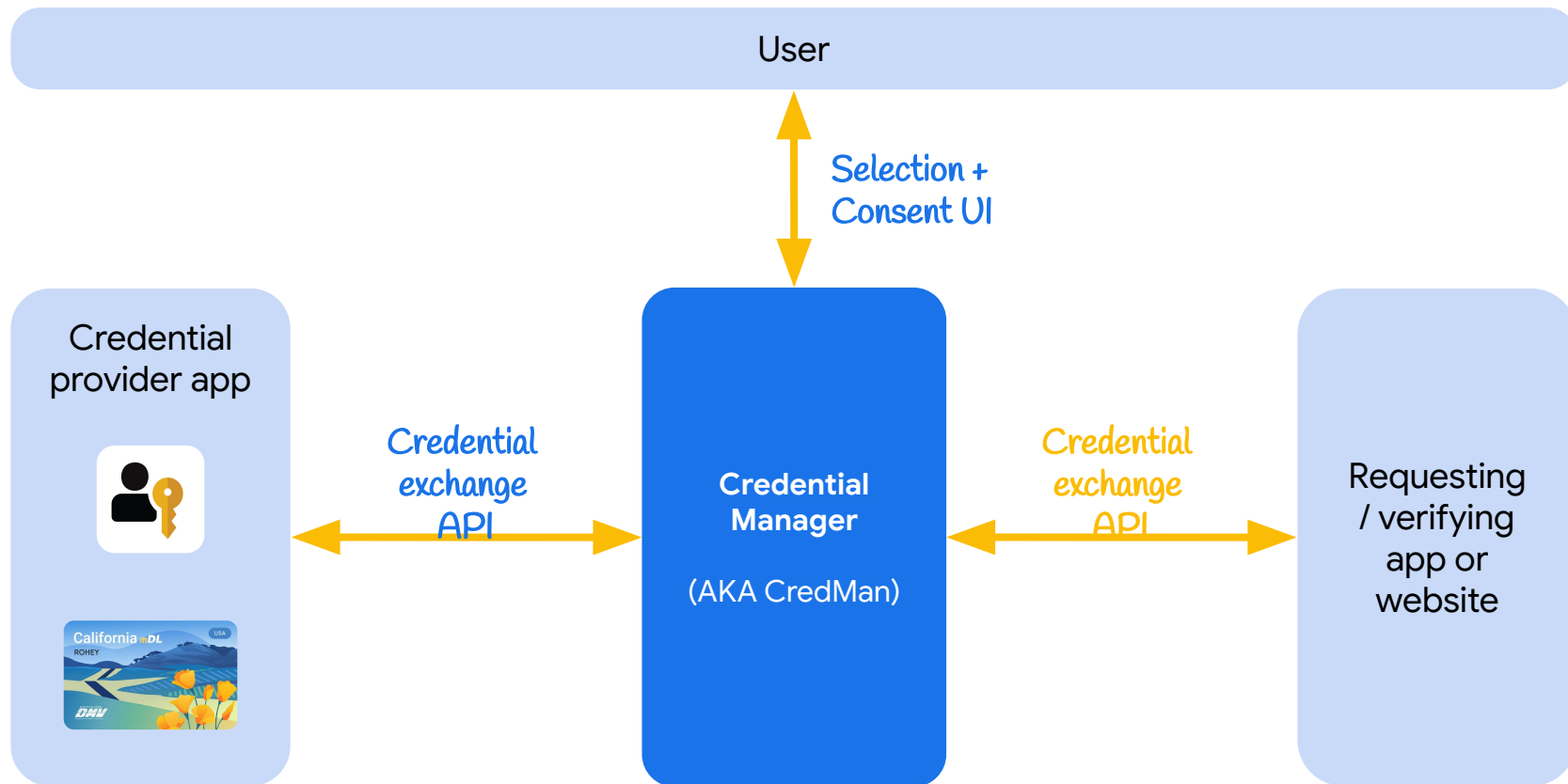


**Cross-device
presentment**

Demo - Alcohol purchase

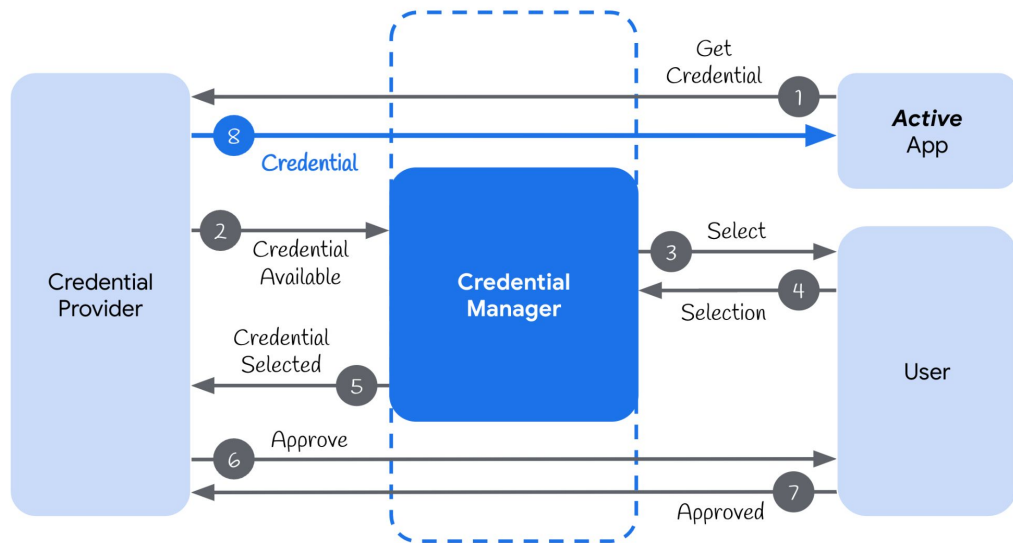


Credential Manager is this credential exchange API + UI layer in Android



How CredMan works

Proprietary + Confidential



1. The App / Website explicitly calls the CredMan API with a get credential request, which is routed to the user's credential provider
2. The credential provider replies whether it has a matching credential
3. If a credential is found, CredMan shows the credential UI
4. The user selects the credential
5. CredMan routes the selection to the credential provider to release the credential
6. The credential provider confirms with the user (e.g. biometric prompt)
7. The user approves the release of the credential
8. CredMan releases (send) the credential to the app

Google Wallet

Digital identity is essential for Google users

Users list **ID cards** as the #1 item they wish to digitize



Provide a comprehensive digital Wallet with **IDs in Google Wallet**



Ensure all users (regardless of what wallet they use) have trusted secure digital IDs on **Android** and **Chrome**.

ID Card

Vaccine card

Loyalty card

Passport

Medical insurance card

Train or bus ticket

Gift card

Metro card

National insurance card

Event ticket

Flight and boarding pass

Gym membership

Biometric info

House key

Google is building an **open, secure, private, standards-based** ecosystem for all wallets on Android



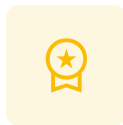
Open

Support the coexistence of multiple wallets, encouraging user choice and driving interoperability between them.



Secure and private

Require user consent for data sharing. Data stored in a secure area on the device.



Standards-based

Support for multiple data formats, such as ISO mdoc and W3C verifiable credentials to ensure interoperability across browsers and wallets.

To achieve that, we made the following available on Android and Chrome

Identity storage

Open-source implementations

Libraries for developers to build secure digital identity wallets for Android.

<https://github.com/openwallet-foundation/multipaz>

Secure key storage

StrongBox allows to securely store and handle cryptographic keys in the secure element on Android devices.

Identity presentation

NFC access

Any wallet can use NFC to allow **in-person presentation** of digital IDs to ISO-compatible readers.

Online access

Digital Credentials API so users can share their digital identities online while using the wallet of their choice.

Google Wallet is a comprehensive digital wallet built on these foundations

Broad range of use cases

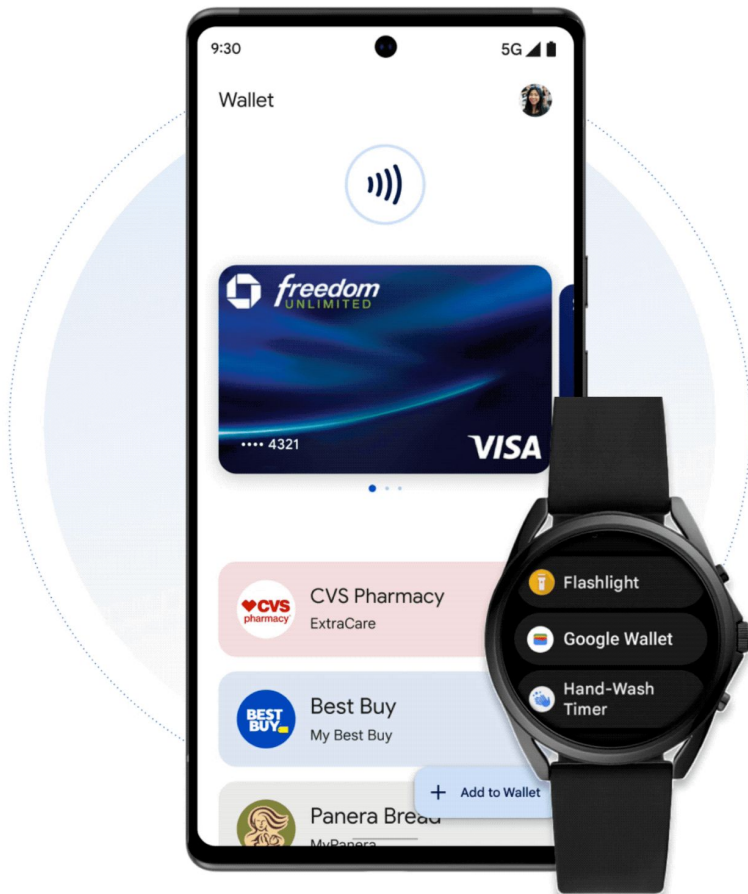
Payments, loyalty, transit, tickets, boarding passes, car keys, hotel keys, and more.

Wide geo availability

Available in +100 countries.

Android footprint

Google Wallet is available in over 3B Android devices.



Two type of Digital IDs are available today in Google Wallet

Government-issued



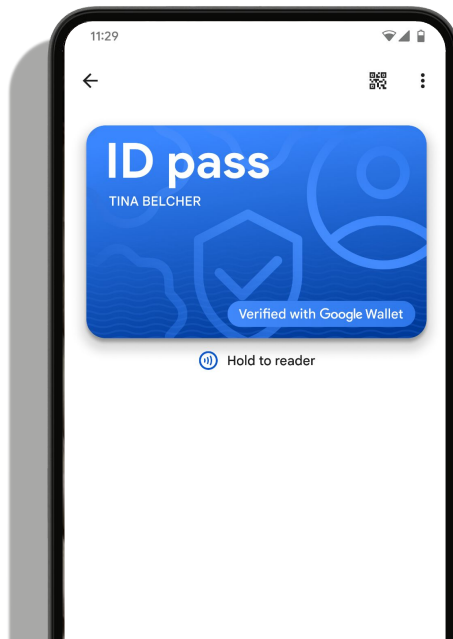
mDL and State ID

Data is verified and issued by state issuing authority.

Credential contains all data elements found in driver's license or state ID.

mdoc format

Google-issued



ID Pass

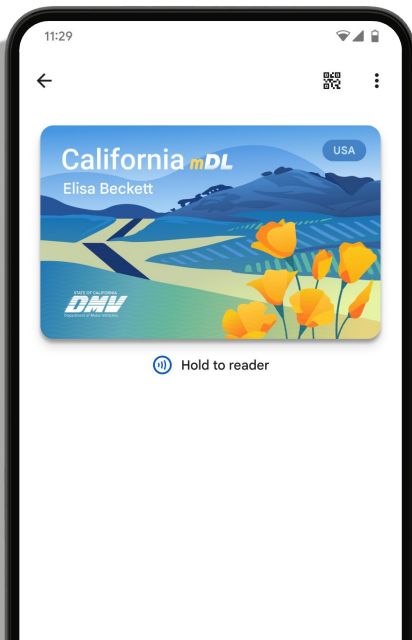
Data is validated against authoritative source and verified by Google.

Credential contains all data elements found in e-passports.

mdoc format

Two type of Digital IDs are available today in Google Wallet, with progressively expanding coverage

Government-issued



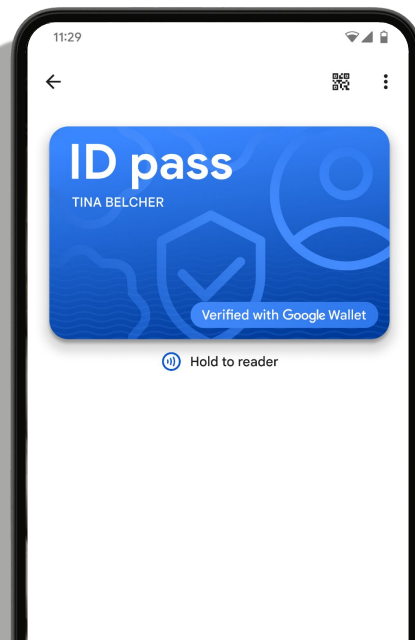
Live



Coming



Google-issued



Our core principles



Trusted secure data

Data is verified against an authoritative source, digitally signed to mitigate tampering, and stored encrypted only accessible by the user.



Privacy-forward

Data is only shared with user consent, with ability to only share the data needed for the use-case.



Easy to use

Sharing ID is as simple as reviewing requested data and authenticating to share.



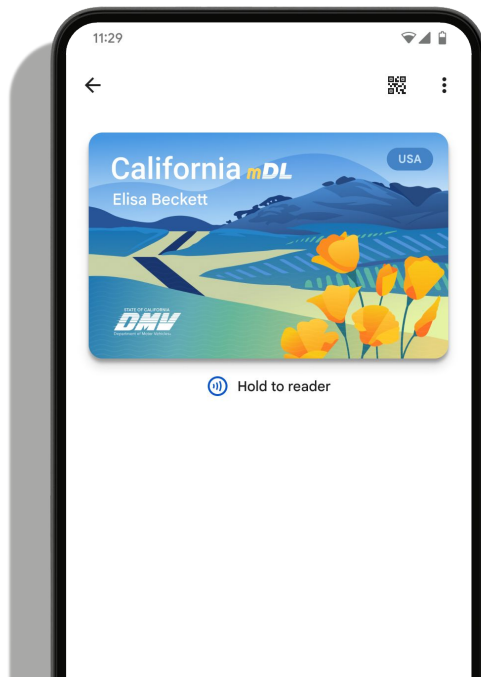
Interoperable

Collaborated with industry stakeholders to define ISO standards for digital IDs to ensure ease of developer adoption and an open ecosystem.

Google Wallet can complement **government wallets** and help you reach users where they are today



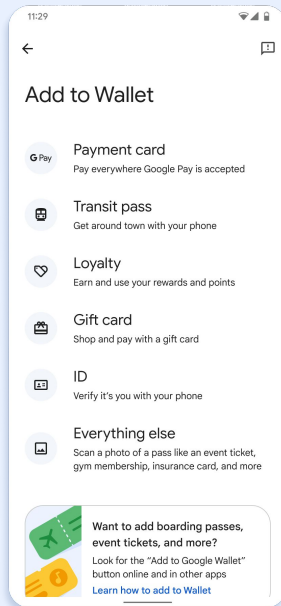
Government wallet



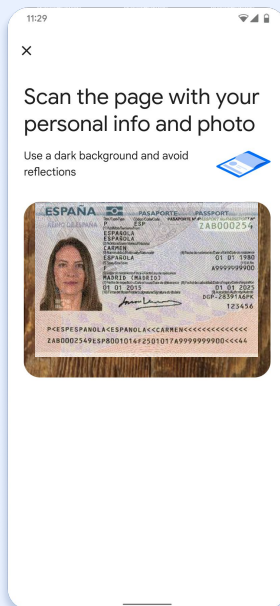
Google Wallet



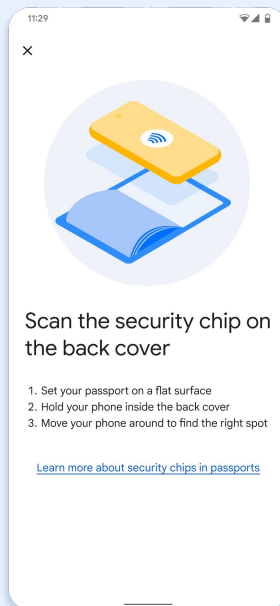
Provisioning an ID Pass in Google Wallet



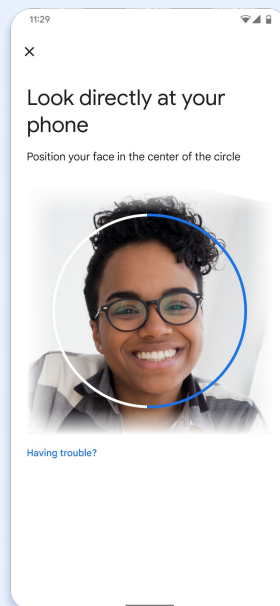
Google Wallet > Add ID



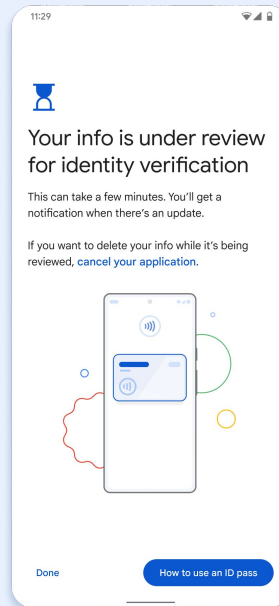
Camera reads MRZ code on the passport



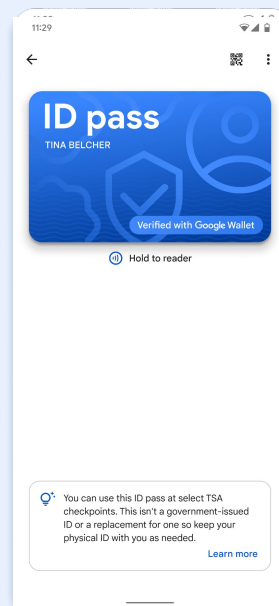
NFC Scan



Scans face

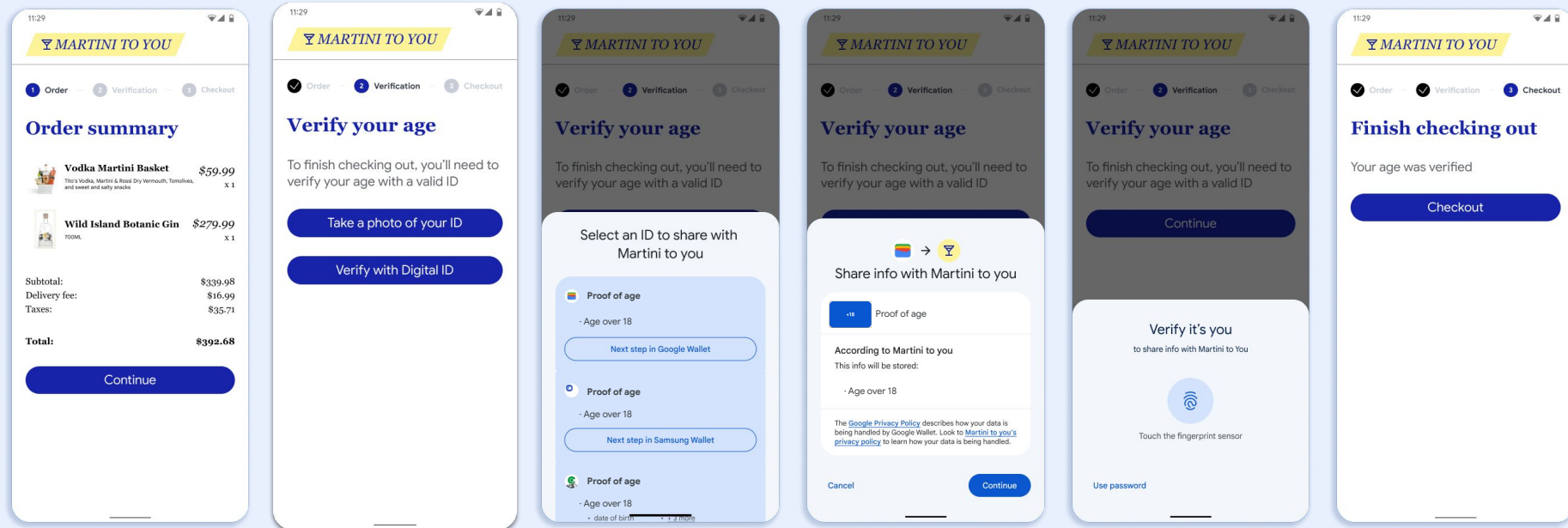


ID application under review



ID Pass available in Google Wallet

Presenting an ID Pass from Google Wallet



Takeaways

Google's approach to Digital Credentials

- **Standardization + Open-source + Productization**
 - Our approach for interoperability, security, privacy, and safety
 - We love working with partners
- **W3C Digital Credentials API**
 - Available to all users since Chrome 141
 - Same-device and cross-device
- **Android is ready for Digital Credentials**
 - StrongBox for meeting the highest levels of assurance
 - Credential Manager for providing a delightful experience



Thank you