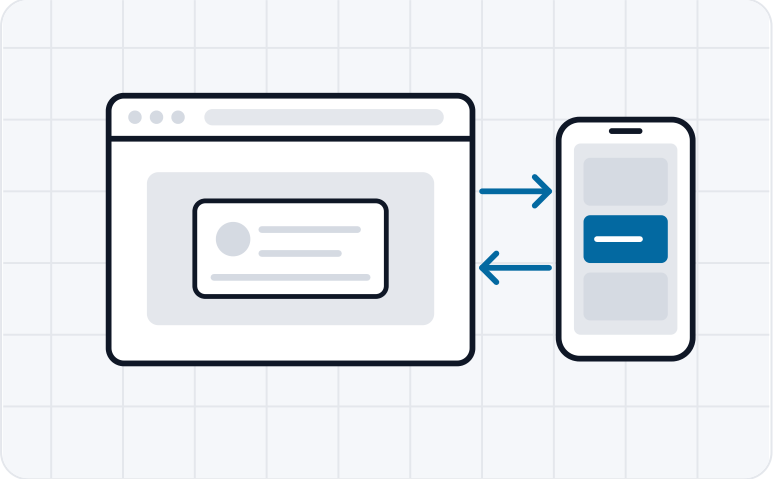
A-SIT Technology

E-Government Innovationszentrum

16. Juni 2025 Analyse · Web & Cloud · Digitale Identität · 10 Min. Lesezeit

Implementierung und Evaluierung der Digital Credentials API

Gerald Palfinger



Autorinnen und Autoren	Gerald Palfinger
Stand	2025-05-26
Quelle	https://technology2.a-sit.at/veroeffentlichungen/implementierung-und-evaluierung-der-digital-credentials-api/

Kernaussagen

- Die Digital Credentials API wird derzeit nur von Chrome unter Android mit Google Mobile Services unterstützt; iOS-Umsetzungen in Safari und Chrome stehen aus.
- Die Integration im Valera Wallet gelang für das Preview-Protokoll, OpenID4VP (draft 28) und ISO 18013-7 Annex C, nur unter Android.
- Die Webseite erhält nur die Attribute, die Nutzende freigeben; einzelne Attribute lassen sich in der Wallet abwählen.
- Mit Chromium 134 ist auch die Freigabe am Desktop per QR-Code und Bluetooth möglich; sie funktioniert derzeit nur zwischen Desktop und Smartphone.

- Die Spezifikation ist noch in einem sehr frühen Stadium; Web Wallets werden nicht unterstützt.

Die Digital Credentials API wird derzeit im Rahmen der Web Incubator Community Group (WICG) entwickelt. Die Schnittstelle gestattet es, digitale Identifikationsnachweise unmittelbar über eine Browser-Schnittstelle abzurufen. Diese ersetzt so die Verwendung von Custom URI Schemes, wodurch die Benutzbarkeit verbessert wird. Aktuell wird die Digital Credentials API in Chrome als sogenannter Ursprungstest (origin trial) evaluiert. Im Rahmen dieses Berichts wird gezeigt, welche Funktionen bereits umgesetzt worden sind. Ebenso wird dargelegt, wie die API im Valera Wallet integriert wurde, um so die darin gespeicherten Ausweise über die neu entwickelte Browserschnittstelle abrufen zu können.

1. Einleitung

Digitale Identitätsnachweise bieten eine Möglichkeit, Identifikationsdokumente direkt auf dem Smartphone in einem sicheren digitalen Wallet zu speichern und sowohl im realen Alltag als auch bei der Nutzung von Online-Diensten eine Identifizierung zu ermöglichen. Allerdings erfolgt die Online-Identifikation aktuell häufig über benutzerdefinierte (Custom) URI Schemes, welche in der Praxis mit gewissen Usability-Problemen behaftet sind. So ist es der aufgerufenen Wallet-Applikation nicht immer möglich, wieder direkt zum Browser zurückzuwechseln. Darüber hinaus besteht für die Nutzerin bzw. den Nutzer die Möglichkeit, den Aufruf von Custom-Schemes zu untersagen, wodurch weitere Aufrufe gegebenenfalls unbemerkt fehlschlagen. Die Einführung der Digital Credentials Browser-Schnittstelle adressiert diese Probleme, indem der Browser über auf dem Gerät gespeicherte Ausweise informiert wird. Dadurch kann der Browser eine aktive Rolle bei der Bereitstellung der Identifikationsdaten einnehmen, anstatt nur passiv Custom-URIs zu öffnen. So kann eine Übersicht über alle am Gerät gespeicherten Ausweise dargestellt werden, bevor die Anfrage an die jeweilige Wallet-Applikation weitergeleitet wird. Zum Abruf der Ausweise muss eine Webseite die standardisierte Schnittstelle implementieren. Auf Seiten der Wallet-Applikation muss dem System mitgeteilt werden, welche Ausweise zur Verfügung stehen. In diesem Bericht wird gezeigt, wie dies erfolgt. Zu diesem Zweck wird die Schnittstelle in die Valera Wallet-Applikation integriert.

2. Hintergrund

Die Digital Credentials API ist eine im Rahmen der Web Incubator Community Group (WICG) derzeit in Entwicklung befindliche Webplattform-Schnittstelle [1], die es ermöglicht, digitale Anmeldedaten wie Führerscheine oder Personalausweise selektiv über das Web zu nutzen. Sie baut auf der Credential Management API auf und erlaubt es Webseiten, überprüfbare Informationen von Nutzerinnen und Nutzern anzufordern, die in digitalen Wallets gespeichert sind. Dabei ersetzt sie Custom URI Schemes wie `openid4vp:` oder `mdoc:` wie diese in den Spezifikationen von OpenID4VP [2] und ISO 18013 vorgeschlagen werden. Die Schnittstelle ist dabei protokollagnostisch und kann somit mit verschiedenen Protokollen wie OpenID4VP oder dem Austroads Request Forwarding Protokoll verwendet werden. Die Spezifikation der Digital Credentials API ist unter [3] verfügbar. Diese ist jedoch derzeit noch in einem sehr frühen Stadium. Dadurch ist ein Großteil der Abschnitte derzeit noch leer und damit nicht spezifiziert. Ebenso entspricht sie

derzeit nicht dem Stand der Umsetzung, so ist das Issuing laut Spezifikation out-of-scope, während die Funktion vor Kurzem in der Implementation der Schnittstelle in Chrome und den Android-Bibliotheken umgesetzt wurde.

Um die Schnittstelle verwenden zu können, muss diese vom verwendeten Browser, dem Betriebssystem (oder einer darauf laufenden Komponente) und der Wallet-Applikation unterstützt werden. Derzeit wird die Digital Credentials-Schnittstelle nur von Chrome unter Android mit laufenden Google Mobile Services (GMS) unterstützt. Eine iOS-Implementierung ist derzeit sowohl unter Safari (Webkit) [4] als auch Chrome für iOS [5] ausstehend.

2.1. Abruf von Identitätsnachweisen

Auf Seiten der aufrufenden Webseite wird der Identitätsnachweis über eine JavaScript-API abgerufen. Um Missbrauch zu vermeiden, muss die Methode durch eine durch den Nutzer beziehungsweise die Nutzerin ausgeführte Aktion wie beispielsweise einen Button-Druck ausgelöst werden. Ein beispielhafter Aufruf der JavaScript API sieht wie folgt aus [6]:

```
const oid4vp = {
  protocol: "oid4vp", // An example of an OpenID4VP request to wallets.
  data: {
    nonce: "n-03W_WT6Ftf",
    presentation_definition: {
      //Presentation Exchange request, omitted for brevity
    },
  },
};

// create an Abort Controller
const controller = new AbortController();
// Call the DC API using the presentation request from the backend
let dcResponse = await navigator.credentials.get({ signal: controller.signal,
mediation: "required", digital: { requests: [ oid4vp ] } });
```

Darüber hinaus kann die Abfrage nach Identitätsnachweisen auch durch native Applikationen erfolgen. Derzeit ist diese Funktion nur unter Android erhältlich. Dazu kann die Jetpack Credentials API in Version 1.5.0 oder neuer verwendet werden [7]. Diese ist in ihrer Funktion ähnlich der JavaScript API aufgebaut.

2.2. Registrierung bestehender Ausweise

Damit die am Gerät vorhandenen und mit dem Request kompatiblen Ausweise dargestellt werden können, müssen diese vorab durch die Wallet registriert werden. Zur Registrierung der Ausweise kann eine Wallet-Applikation die Play Services Identity Credentials-Bibliothek in Version 16 verwenden [8]. Die Kommunikation zwischen Applikation und Browser erfolgt demnach über die Google Play Services und ist noch nicht nativ ins Betriebssystem integriert. Zur Registrierung eines Credentials dient die Methode `registerCredentials(RegistrationRequest)` des `IdentityCredentialClient`. Dazu wird ein `RegistrationRequest` erstellt, der das Credential als Byte Array enthält. Zusätzlich wird diesem auch ein

Matcher übergeben, welcher in Web Assembly geschrieben ist. Dieser Web Assembly-Code wird jedes Mal ausgeführt, wenn eine Webseite einen Ausweis abruft. Der Web Assembly-Code überprüft den Request der Webseite und signalisiert dem Browser, ob das jeweilige Credential dem Request entspricht. Ist dies der Fall, kann das Credential zur Auswahl angeboten werden. Die API die zur Erstellung des Matchers verwendet werden muss kann beispielsweise unter [9] abgerufen werden.

3. Umsetzung

Die Wallet-seitige Unterstützung wurde in das Valera Wallet eingebaut [10]. Die Unterstützung funktioniert aufgrund der derzeitigen Einschränkungen der Schnittstelle nur unter Android. Die Umsetzung implementiert sowohl ein „Preview"-genanntes Protokoll, OpenID4VP nach draft 28 sowie ISO18013-7 Annex C. Die Digital Credentials API wird sowohl mit „alten" als auch mit „neuen" (Stand Mai 2025) JSON-Attributnamen unterstüzt. Um die Umsetzung zu verwenden muss der Chrome-Browser mit aktivierter Unterstützung für die Digital Credentials-Schnittstelle verwendet werden. Die Unterstützung kann unter `chrome://flags` aktiviert werden, wie in Abbildung 1 ersichtlich ist. Nach Aktivierung der Flag ist es Webseiten möglich, digitale Ausweise anzufordern. Ein Demo-Service Provider wird von Google unter [11] bereitgestellt. Dieser funktioniert derzeit nur mit Ausweisen des Typs Mobile Driving License (mDL) und „Preview"-Protokoll sowie OpenID4VP. Andere Service Provider wie beispielsweise [12] unterstützen weitere Ausweistypen. Im Folgenden wird der Same-Device Flow unter Verwendung des Demo-Service Providers von Google beschrieben, also der Abruf eines Ausweises durch eine Webseite am selben Gerät, auf der auch die Valera Wallet-Applikation läuft.

Der Nutzer beziehungsweise die Nutzerin startet den Vorgang durch Klick auf „Request Credentials" beim Demo-Service Provider. Daraufhin werden alle kompatiblen Ausweise angezeigt. Dabei kann die Nutzerin beziehungsweise der Nutzer einen Ausweis auswählen, wie in Error! Reference source not found.Abbildung 2 ersichtlich ist. In der Auswahl wird angezeigt, welche Attribute angefordert werden. Durch den Klick auf Details wird der Inhalt der angeforderten Attribute angezeigt. Diese Details wurden vorab von der jeweiligen Wallet im System registriert, wie in Abschnitt 2.2 beschrieben wurde.

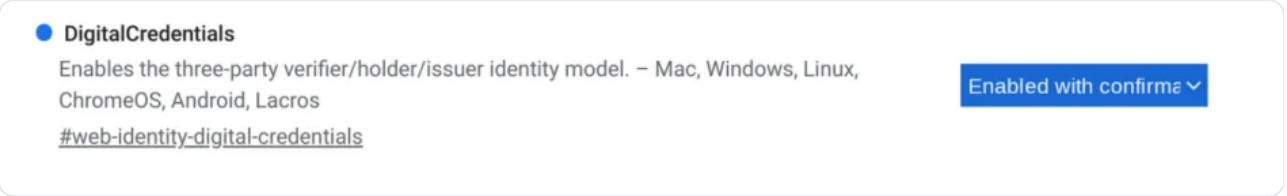
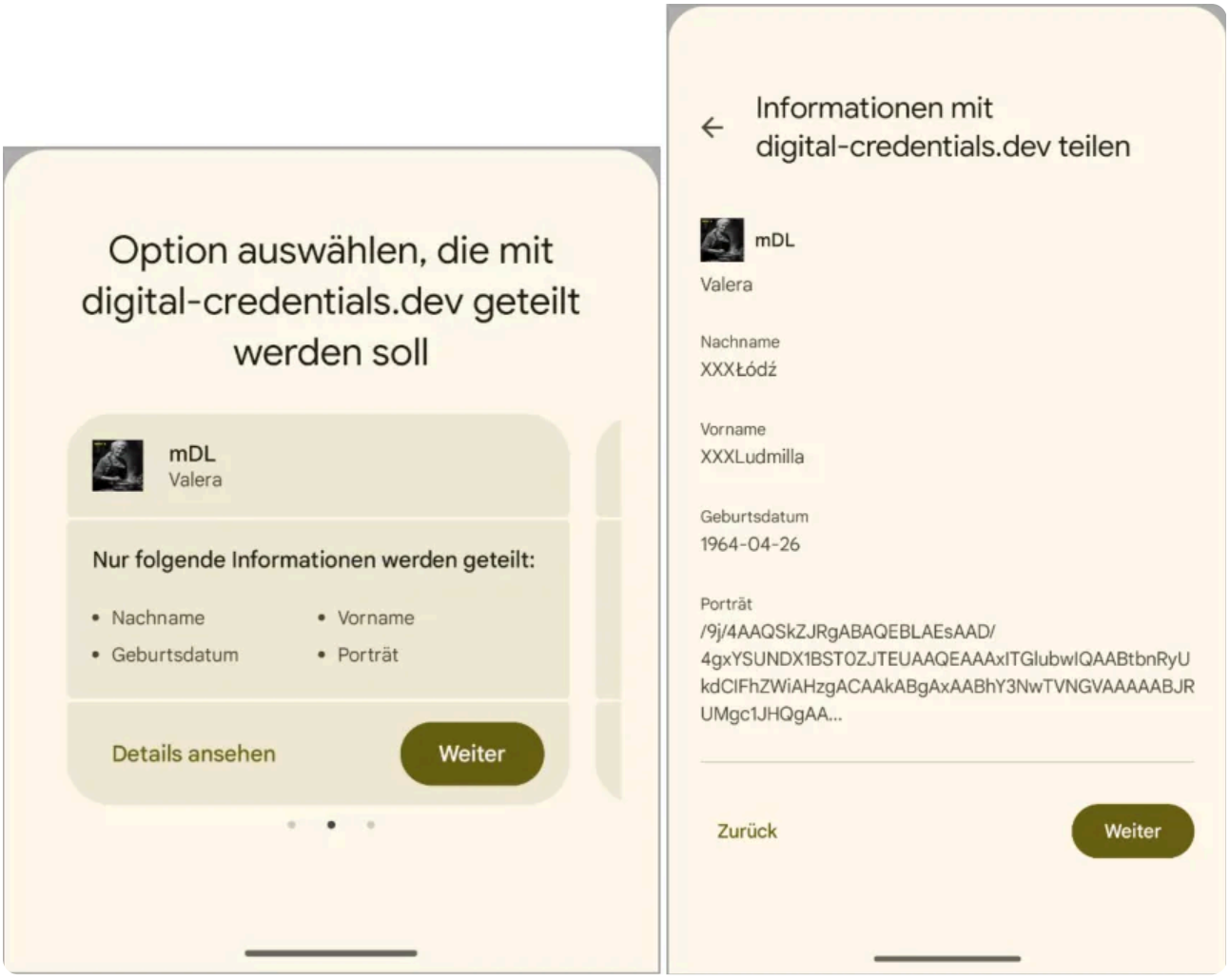


Abbildung 1: Aktivierung der Digital Credentials-API im Chrome-Browser unter chrome://flags.



Abbildung 2: Dialog zur Auswahl des gewünschten Ausweises sowie der Detailansicht des Ausweises.


Informationen mit digital-credentials.dev teilen
mDL
Valera
Nachname
XXXŁódź
Vorname
XXXLudmilla
Geburtsdatum
1964-04-26
Porträt
/9j/4AAQSkZJRgABAQEBLAEsAAD/
4gxYSUNDX1BST0ZJTEUAAQEAAAxITGlubwIQAABtbnRyU
kdCIFhZWiAHzgACAAkABgAxAABhY3NwTVNGVAAAAABJR
UMgc1JHQgAA...
Zurück
Weiter



Nach der Auswahl des Ausweises wird in die jeweilige Wallet Applikation gewechselt. Diese bekommt beim Aufruf ein Intent-Objekt. Dieses enthält Informationen über die aufrufende (Browser-)Applikation, die URL der Relying Party sowie alle angeforderten Attribute. Diese Informationen werden in der Umsetzung in der Valera-Wallet wie in Abbildung 3 angezeigt. Die Nutzerin beziehungsweise der Nutzer hat dabei die Möglichkeit, einzelne Attribute abzuwählen und so deren Freigabe zu verhindern.

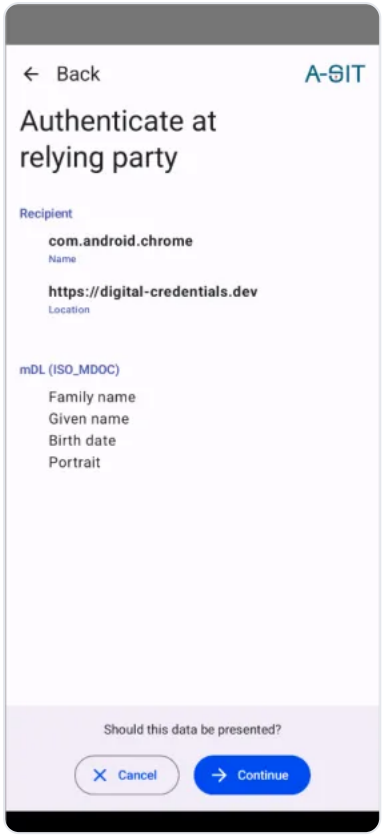


Abbildung 3: Anzeige der aufrufenden Applikation und Webseite sowie der angefragten Attribute in der Valera Wallet-Applikation.



Abbildung 4: Anzeige der freigegebenen Daten auf der Demo-Webseite.

Nach der Freigabe durch die Nutzerin beziehungsweise den Nutzer werden die Attribute über die Schnittstelle an die aufrufende Webseite weitergegeben. Diese werden dann von der Demo-Webseite angezeigt, wie in Abbildung 4 ersichtlich ist. Die Webseite erhält nur die Attribute, die die Nutzerin beziehungsweise der Nutzer freigegeben hat. So wurde im gezeigten Beispiel die Freigabe des Geburtsdatums verweigert, wodurch dieses nicht an die Demo-Webseite weitergegeben wurde und so auch nicht angezeigt werden kann.

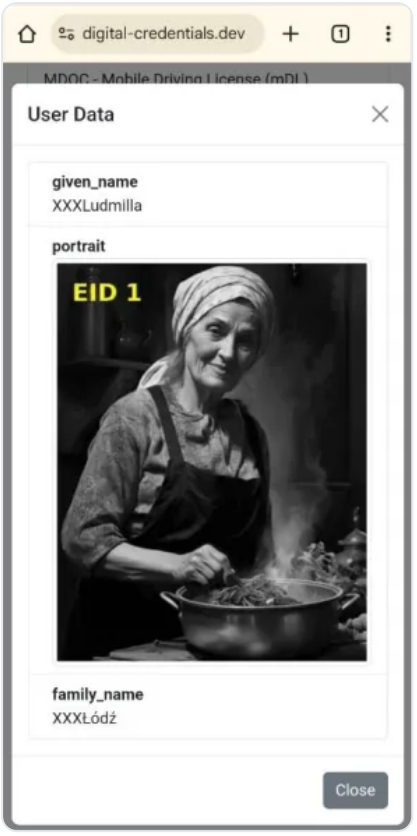


Abbildung 5: QR-Code zum Aufbau der Verbindung zwischen Desktop und Smartphone.

3.1. Geräteübergreifende Freigabe

Mit Version 134 von Chromium und der aktuellen Version der Google Mobile Services (GMS) wird auch die Freigabe von auf Android-Geräten gespeicherten Ausweisen für die Nutzung am Desktop unterstützt. Dazu muss auf dem Smartphone ein QR-Code gescannt werden wie in Abbildung 5 ersichtlich ist. Die Verbindung zwischen Smartphone und Desktop wird dann über Bluetooth aufgebaut. Dazu muss Bluetooth am Desktop vorab aktiviert werden, ansonsten schlägt der Vorgang mit einem unspezifischen Error fehl. Die Implementation basiert dabei auf jener für Passkeys, wie auch in Abbildung 6 ersichtlich ist. Nach Klick durch die Nutzerin beziehungsweise dem Nutzer auf den angezeigten Button wird die Verbindung aufgebaut. Bei erfolgreicher Verbindung wird der Dialog zur Auswahl der kompatiblen Ausweise wie in Abbildung 7 ersichtlich ist angezeigt.

Abbildung 6 Nach dem Scannen bietet die Kamera-App am Smartphone an, die Verbindung über den Button „Use passkey“ herzustellen.

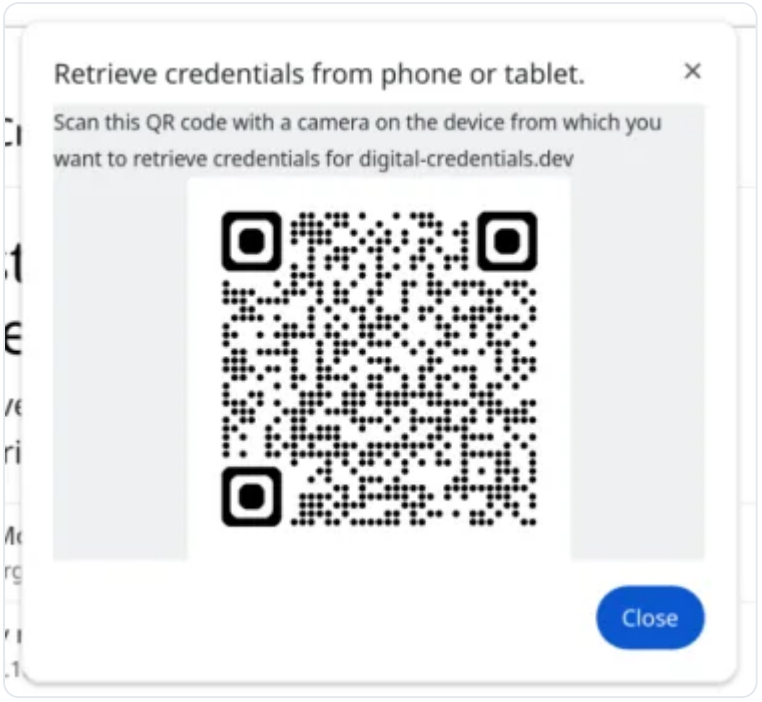


Abbildung 7: Herstellung der Verbindung sowie Auswahldialog mit der Anzeige der kompatiblen Ausweise.

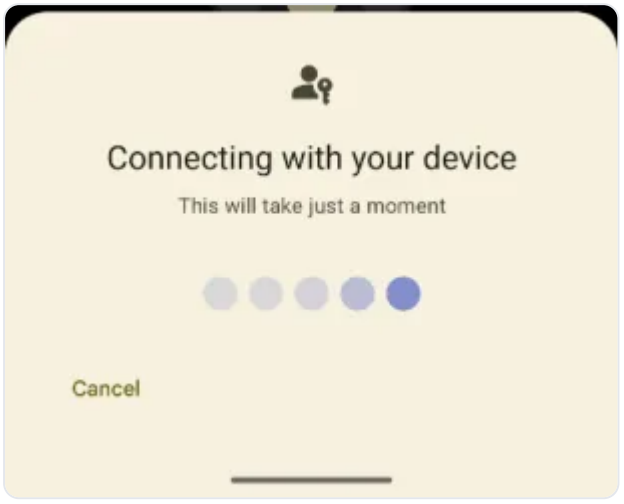


Abbildung 8: Anzeige der angeforderten Attribute in der Valera Wallet-Applikation.

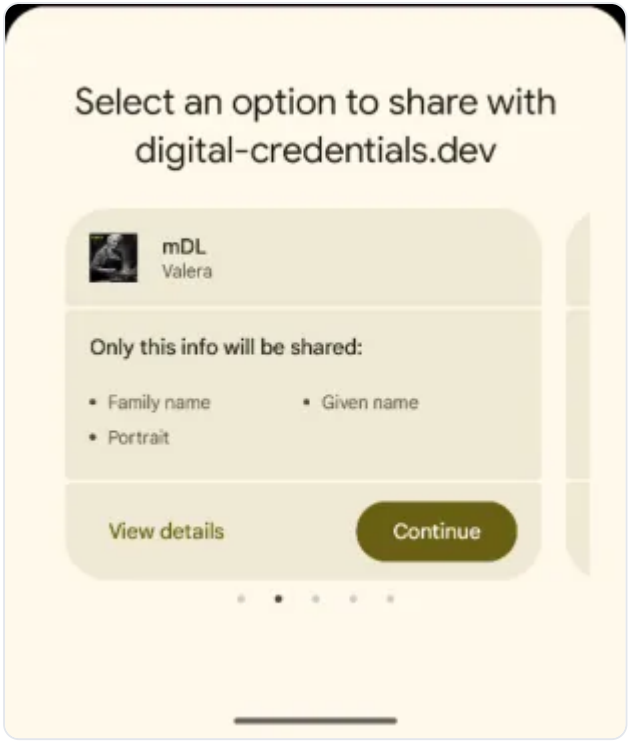


Abbildung 9: Anzeige der freigegebenen Daten auf der Demo-Webseite am Desktop.

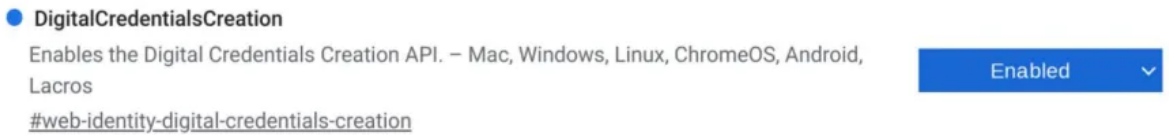
Nach der Auswahl des Ausweises wird auch hier in die Wallet-Applikation gewechselt (s. Abbildung 8). Als Aufrufer wird hier in der aktuellen Implementation nicht Chrome, sondern der Paketname der Google Mobile Services `com.google.android.gms` angezeigt. Nach der Freigabe der Attribute erscheinen diese im Chrome-Browser am Desktop, wie in Abbildung 9 ersichtlich ist.

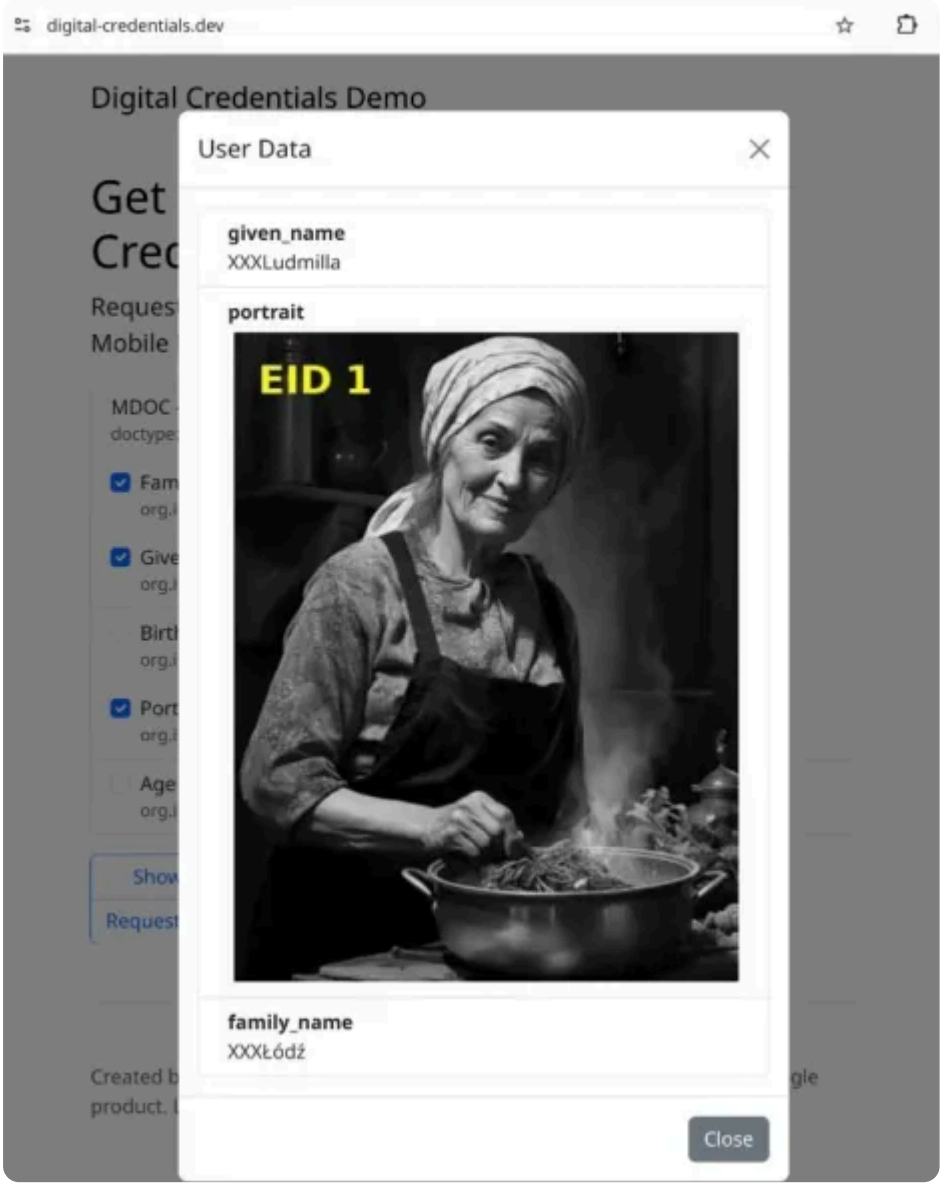
Der geräteübergreifende Modus funktioniert derzeit nur zwischen Desktop und Smartphone. Eine Freigabe zwischen Android-Geräten wird aber zumindest überlegt [13].


21:33
76%
Back
A-SIT
mDL
mDL
mDL
ISO mDoc
XXXLudmilla XXXŁódź
26.4.1964
Family name
Given name
Portrait
Which Credential should be used?
Continue



Abbildung 10: Aktivierung der Issuance-Funktion der Digital Credentials-API im Chrome-Browser unter chrome://flags.





3.2. Weitere Funktionen

Zusätzlich zum Abruf von bestehenden Ausweisen sind auch noch weitere Funktionen geplant oder wurden bereits umgesetzt. So ist es seit Alpha04 der Play Services Identity Credentials-Bibliothek (16.0.0-alpha04 vom 17. Dezember 2024) sowie mit dem Canaray-Release 133.0.6866 von Chrome darüber hinaus auch möglich, den Issuance-Prozess über die Digital Credentials-Schnittstelle durchzuführen [14] [15]. Dazu gibt es eine eigene Flag, die im Chrome-Browser unter `chrome://flags` aktiviert werden muss, wie in Abbildung 10 ersichtlich ist.

Die Unterstützung von Web Wallets ist derzeit nicht umgesetzt. Ebenso gibt es kaum Informationen zum Status beziehungsweise zu einem möglichen Plan zur Umsetzung einer solchen Unterstützung. Im Rahmen eines Meetings welches Teil der W3C TPAC war, gab es jedoch den Plan eine erste Diskussion (initial discussion) zur Unterstützung von Web Wallets zu führen [16]. In den veröffentlichten Meeting-Notes fehlt dieser Teil der Agenda jedoch.

4. Fazit

Dieser Bericht demonstrierte die Implementierung der neuen, sich noch in der Entwicklungsphase befindlichen Digital Credentials Schnittstelle in der Valera Wallet-Applikation für das Betriebssystem Android. Diese ersetzt die derzeit übliche Verwendung von Custom URI Schemes und resultiert dadurch in einer verbesserten Integration zwischen dem Browser und der Wallet-Applikation. Die Integration konnte erfolgreich für das „Preview"-Protokoll und in weiterer Folge auch für OpenID4VP sowie ISO 18013-7 Annex C umgesetzt werden. Eine Unterstützung für das Betriebssystem iOS ist derzeit noch nicht gegeben, da die Integration der Digital Credentials-Schnittstelle in den Safari- oder Chrome für iOS-Browser noch ausstehend ist. Darüber hinaus existiert bislang keine Unterstützung für Online-Wallets in der Schnittstelle, jedoch wurden laut einer online veröffentlichten Agenda zumindest erste Diskussionen zur Umsetzung einer solchen Unterstützung geführt.

Referenzen

[1] „WICG - Digital Credentials," 2024. [Online]. Available: https://github.com/WICG/digitalcredentials/tree/main. [Zugriff am 30 12 2024].

[2] „OpenID for Verifiable Presentations - draft 23 - Wallet Invocation," 02 12 2024. [Online]. Available: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html#name-wallet-invocation. [Zugriff am 30 12 2024].

[3] „Digital Credentials," 2024. [Online]. Available: https://wicg.github.io/digital-credentials/. [Zugriff am 30 12 2024].

[4] M. Caceres, „[Meta] Implement Digital Credentials API," 01 02 2024. [Online]. Available: https://bugs.webkit.org/show_bug.cgi?id=268516. [Zugriff am 30 12 2024].

[5] „Ecosystem Support," 2024. [Online]. Available: https://digitalcredentials.dev/docs/ecosystem-support/. [Zugriff am 30 12 2024].

[6] T. Cappalli, „Digital Credentials API," 07 10 2024. [Online]. Available: https://github.com/WICG/digitalcredentials/blob/main/explainer.md. [Zugriff am 30 12 2024].

[7] A. Developers, „credentials," Google Inc, 30 10 2024. [Online]. Available: https://developer.android.com/jetpack/androidx/releases/credentials. [Zugriff am 30 12 2024].

[8] Google, „Release Notes," 2024. [Online]. Available: https://developers.google.com/android/guides/releases. [Zugriff am 30 12 2024].

[9] „credentialmanager.h," 2025. [Online]. Available: https://github.com/digitalcredentialsdev/CMWallet/blob/main/matcher/credentialmanager.h. [Zugriff am 22 05 2025].

[10] G. Palfinger, „Add support for digital credentials API," 20 12 2024. [Online]. Available: https://github.com/a-sit-plus/valera/pull/151. [Zugriff am 30 12 2024].

[11] Google Android und Chrome Team, „Get Started with the Digital Credential API," 2024. [Online]. Available: https://digital-credentials.dev. [Zugriff am 30 12 2024].

[12] MATTR Labs, „Interop Verifier,“ 2025. [Online]. Available: https://interop-verifier.mattrlabs.dev/. [Zugriff am 22 05 2025].

[13] „Make hybrid flow work between Android devices,“ 30 07 2024. [Online]. Available: https://issues.chromium.org/issues/356399833. [Zugriff am 30 12 2024].

[14] T. Cappalli, „2024 12 02 Meeting Notes - Incubation update,“ 02 12 2024. [Online]. Available: https://github.com/WICG/digital-credentials/wiki/2024-12-02-Meeting-Notes#incubation-update.

[15] samuelgoto, „How should we go about issuance? - digital-credentials Issue Tracker,“ 2024. [Online]. Available: https://github.com/WICG/digital-credentials/issues/167. [Zugriff am 19 12 2024].

[16] T. Cappalli, „2024 09 23 (TPAC) Meeting Notes,“ 23 09 2024. [Online]. Available: https://github.com/WICG/digital-credentials/wiki/2024-09-23-(TPAC)-Meeting-Notes. [Zugriff am 30 12 2024].