

WICG - Digital Credentials - TPAC 2024 Session

M 2024-09-23 14:00-16:00 PT

<https://www.w3.org/events/meetings/456057ed-2184-4870-86a4-f01c8158d3c0/>

Session Chair: Tim Cappalli

Session Scribe(s): Pam Dingle and Rick Byers

Payments-focused session: [Web Payments WG: Digital Credentials and Payments](#)

Directly Related Breakout Sessions


- [Harmonizing Identity-Related Web Platform APIs](#)
- [Real World Identity and the Web... Continued](#)
- [Mitigate Threats for Digital Credentials API](#)

Other Breakout Sessions

- [Threat Modeling @ W3C](#)
- [Registries for W3C Specifications](#)
- [DID Method Standardization](#)
- [Electronic Transferable Records: Implemented Using Transferable Verifiable Credentials](#)

Agenda

14:00 - 14:10	Welcomes & Administrivia	all
14:10 - 14:15	Working Group charter update	Simone, Heather, Wendy
14:15 - 14:20	Chrome + Android demo	Lee & team
14:20 - 14:35	Browser Engine Roundtable	Browser engines
14:35 - 15:05	Last Call for Breaking Changes for ARF/eiDAS/OpenID4VP	<i>discussion</i>
15:05 - 15:20	Graduating from the WICG: <i>What would it take to get to a FPWD in the FedID WG?</i>	<i>discussion</i>
15:20 - 15:30	Issuer identity in selective disclosure cases (issue 139)	<i>discussion</i>
15:30 - 15:45	Support for issuance (issue 167)	<i>discussion</i>
15:45 - 16:00	Web wallet integration (initial discussion)	<i>discussion</i>

 Please Join the [Zoom Meeting](#) and Use Hand Raise to Join the Queue

Attendees

- Heather Flanagan (Spherical Cow Consulting)
- Matthew Miller (Self)
- Helen Qin (Google Android)
- Lee Campbell (Google Android)
- Wendy Seltzer (Tucows, Invited Expert)
- Sam Goto (Google Chrome)
- Pamela Dingle (Microsoft)
- Rick Byers (Google Chrome)
- Kyle Den Hartog (Brave)
- Joseph Heenan (Authlete / OIDF)
- Alan Buxey (MyUNiDAYS Ltd.)
- Zachary Tan (Google Chrome)
- Ashima Arora (Google Chrome)
- Mohamed Amir Yosef (Google Chrome)
- Martin Alvarez (Huawei)
- Bumblefudge (Protocol Labs)
- Benjamin VanderSloot (Mozilla)
- Martin Thomson (Mozilla/TAG)
- Nicolas Pena Moreno (Google Chrome)
- Simone Onofri (W3C)
- Wonsuk Lee(ETRI)
- George Fletcher (Capital One)
- Aaron Selya (Google Chrome)
- Susan Stroud (self)
- Yi Gu (Google Chrome)
- Theresa O'Connor (Apple/TAG)
- Daniel Appelquist (TAG)
- Kristina Yasuda (SPRIND)
- Nina Satragno (Google Chrome)
- Brian Campbell (Ping Identity)
- David Waite (Ping Identity)
- Hadley Beeman (TAG)

Regrets

- Ted Thibodeau (OpenLink Software) — conflict with DID WG

Notes

- (review of agenda by Tim Cappali) — <https://tclslides.link/tpac24-dcapi>
- Note: We are using the Zoom hand raise to queue comments, please everyone join the meeting
- Heather: There is a formal objection against the recharter of Federated Identity WG
 - Working with the person/org, looking for compromise. A council is being formed and a note went out to the advisory council
 - In the meantime — the issues will be discussed in the Federated Identity WG meeting Tuesday afternoon, all welcome
 - Simone is here if anyone wants process questions answered (point from Tim)
 - Links to the formal objection will be added to these notes
 - Quick summary is that the objection centers on certain identity operations being made easier. The question is, is that a good thing or a bad thing?
- Lee: What is the timeline?
- Heather: Council not formed until early October, but preparations and conversations are already in process. I am hoping this is a 2–3 month resolution rather than a year
- Tim: In the repo, there is a document called “custom schemes”. We are fixing some of the market interactions as we go.
 - <https://github.com/WICG/digital-credentials/blob/main/custom-schemes.md>
- Mark: How much of the council deliberations will be public?
- Tess: The council releases a report; the deliberations will not be released.
- Wendy: The team report that goes to the Council is also public.
- Tim: This item will be an evergreen item until it is resolved.
- Tim: Questions on charter, procedural? (no)
- Lee/Helen demo
 - Multiple players in API. This is using Chrome’s origin API, calls into Android, which then calls to the wallet, driving consent.
 - First demo — same machine
 - Phone with wallet app and fake drivers license
 - A request to verify results in a choice screen
 - One credential matches, user selects continue
 - Wallet App UI shown with final consent
 - Verifying website shows what was presented.
 - When it comes to cross-domain, we can use the CTAP protocol (from passkeys) to create a “tunnel”
 - The same demo occurs as before
 - QR code on another platform links to the phone
 - Same flow proceeds
- Questions:
 - Tony: When you say issuer with a wallet, is it true the mDL comes from an issuer? If so, taking that information from the mDL and doing selective disclosure is happening, correct? If so, wouldn’t it contain the original signature?

- Lee: This is a complex answer. What comes back depends on what you request. You will see statements signed by the issuer.
- Tony: So you will return the same response as 18013-5?
- Lee: Yes.
- John B: Different wallets may do different things.
- Tony: But it is up to the wallet what comes back.
- Lee: It could even be encrypted stuff coming back.
- Tony: So would the API cover all the different formats (mdoc, VC)?
- Tim: This is just plumbing; whatever comes back, comes back. Right now, the API only supports presentations from XX (somebody help here if you can) and OpenID4VP. More could follow.
- Martin: Concerned about the privacy angle with selected disclosure. We saw on the screen that you were sharing some information, but in practice that isn't what you are sharing, because this is a salted hash SD-JWT that has very specific privacy challenges.
- Lee: Things like MDOC convey more than what the API covers; you will also reveal other salted hashes that are tracked across sites. We need to manage these correctly.
 - This is an important discussion on how what is disclosed is communicated to the end user. This is the question of whether we should encourage selective disclosure.
- Martin: The implications of this "buying alcohol" use case is not obvious.
- Lee: An identifying presentation is definitely less obvious. If there are presentations that are anonymizing, then it is problematic if the implementation reveals extra hashes and other information.
- Tony: Looking at the API, I didn't see ... people want to know how the identity piece happens, not just how the authentication piece happened.
- Tim: This API is deliberately transparent. This selector may not have shown it.
- Lee: The API itself takes 2 parameters: the protocol and (something else). Our API tries to figure out the credential intersection. On Android, we need to query each wallet to see if a credential matches. Say there are 3 wallets; we don't want the other wallets to know what the other apps have. Any app can be a wallet. We take the query, fire it to the wallet, and allow the app to run code to decide if they are affected; this happens in a sandbox. Once you decide you are affected, you (the app) will authenticate the user, show consent, etc.
- Tim: If you are familiar with deep links, this API just takes over that process. In the diagram, we are talking about the green line. (need link to diagram).
- Tony: How does this apply to custom URL strings?
- Tim: That's what this API is replacing.
- Next section: Browser Vendor update
- SamG: We have an origin trial running; they run 3-6 months. It happens in Chrome Stable, so real users are using it — a great way to get feedback. From there, we assess user satisfaction and applicability to the ecosystem. It is a

structure where we can change the API as we learn. origin trial is running from browser version 128 to 134. This may also be available in Edge

- Cross-device flow is also getting ready to be behind a flag in Chrome (it is almost ready) — will merge into the origin trial
- Rick: We announced that Google Accounts are starting to use this soon, in cases where somebody changes a birth date (e.g., becoming a major after being a minor)
 - <https://developer.chrome.com/blog/digital-credentials-api-origin-trial>
 - Driver's license and e-passports will be part of that
- Tim: There is a change to a CTAP2.2 protocol that has gone through workgroup approval but needs board approval. Hopeful-ETA is EOY.
- Martin: The dystopia is here, it's just unevenly distributed. Wrt selective disclosure, we (Mozilla) can't make it work so far. More than just the things you choose to disclose are being revealed. We think it would be irresponsible to ship that without understanding it.
- SamG: In this model, part of the responsibility is shared with the wallet.
- Martin: If you're OK with that, then great, but we don't think that will work; we need people to properly understand what we are shipping vs. what we are delivering. The age example demoed worries me quite a bit. I don't think folks would want to be surprised by the data going back to the issuer. It seems to be an inherent property of this API to handwave over whose responsibility this is.
- Rick: Age verification is already widespread and already relying on a contractual relationship that says privacy will be maintained. We don't think this is any worse than that.
- Martin: We don't need to aid and abet that process.
- Rick: Today, Mozilla will allow you to take a pic of your license.
- Martin: Yes, but today the consequence of that action are obvious.
- Lee: But if you want to just share "age over 18"...
- Martin: But you aren't only sharing that.
- Kyle: This is the same issue we see — how do you break down agency? We are deferring to another entity, and how do you know if that entity is behaving? Is consent done? This appears to us as a risk. The way in which we approach it by overscaling, it doesn't allow us to represent the model.
- MattM: I can hear the concerns on implementing this functionality. There are organizational liabilities, but the idea that we can have the same sort of identifiable information over a custom scheme rather than over an API, but if we go through the W3C process there is value in having greater visibility/oversight — I don't think that should be discounted.
- Ben (Mozilla): Issuer/verifier linkability is as much of a concern re: the browser's user agency as issuer identity, and is harder to explain to the user.
- JohnB: I agree, and EU folks also agree that issuer/verifier collusion is a concern — the EU has a timeline for delivering this functionality; they need to mandate the use of custom URLs or something else, but they will mandate *something*. Question is, can we use this pipe to influence some of the other decisions —

there are some people who will do some fairly bad things; it would be good if we can use this to encourage other parts of the ecosystem to come up with better zero knowledge options. The short term options may not be great, but they allow for oversight

- Tim - Simone is leading a threat model discussion, later today is the Federated Identity WG where this will be updated.
- Section:
- Tim: There is a lot going on with the ARF and the EU. Did a last call for major breaking credentials, big ones were navigator.credentials move, and changing provider to request and request to data (somebody confirm); those very core changes are done. The API is currently simple, most of the complexity is in OpenID4VP — is there anything else that needs to be changed that is big? What can we do to create to make the first CG report — using the tag `cgr1-blocker`
 - First question, what protocols are supported? — now we have 1 but want to be ready for #
 - JSON serialization methods — want to get the serialized request/response from the server.
 - Want to add a statement that all responses to verifier must be encrypted.
- Tony: Is encryption expected to be point-to-point or something else? There could be backend components that prevent encryption. Is this message level?
- (lots of people chiming in)
- Tim: What passes back through the API is expected to be encrypted. Getting into the registry may also need some unencrypted information.
- Martin: Malicious router extensions...
- Lee: We accept that we need to protect the request from other parties, but the request is more point-to-point, so it is less available.
- Martin: If your solution requires that you protect the information — most of those requirements can be met by an application that receives unprotected information and does what it needs to do. Having a wallet do the encryption causes the browser to be blind.
- Lee: I don't know about Sam but helping the app to maintain its requirements...
- Sam: Can we remove as many issues as we can to get to the CG report as quickly as possible? There are time considerations it is important to be sensitive to.
- Tim: We tentatively set a deadline of Oct 31.
- Kristina: This document that legally mandates technical mechanisms is being written. If you believe this technology should be used to protect 300M Europeans, then you need to report out in time for that process. We are grateful for the WICG cooperation so far. It would be great to have a stable version by the end of year. We need a stable reference — Oct 31 needs to be checked but sounds like a (reasonable? good? fair?) deadline.
- Sam: That's just over a month.
- Tim: There are 3 open issues on screen; does anyone think any of the 3 can/should not go forward? (if someone with github open could link to those open issues that would be great)
- Sam: Wrt encryption, there is no browser code that gets written.

- Kristina: We could mandate this in OpenID4VP, too.
- Sam: Maybe that's easier to do.
- Tony: There is the same issue in mDL, where VP was doing the encryption, but it still had to be stated in the spec that this is end-to-end encryption.
- Sam: A non-normative statement is reasonable to me, but I want to be clear that this isn't something that the browser can enforce.
- JohnB: I'm not sure everyone wants end-to-end encryption.
- Tony: That's why it would be non-normative.
- Tim: If we are saying it is required to be in the registry, I know some of the registry work is in the air.
- JohnB: If there are 2 versions where one is E2E, and there is an alternative, that would be fine.
- SamG: At the minimum, it would have to be a SHOULD not a MUST. Isn't it possible in OpenID4VP that it doesn't use encryption?
- JohnB: Yes, there is alg:none.
- BrianC: There is something in OpenID that allows a non-encrypted response.
- Heather: The questions of normative, etc., aren't relevant while we are doing a CG report anyway.
 - Comment from Zoom chat: bumblefudge to Everyone (Sep 23, 2024, 3:07 PM):
To come back to Tessa's point about a CG Report misrepresenting its opinions as normative statements, it might be safer for those "two sentences" to describe the prototype that inspired the report rather than all future conforming implementations, i.e. "one way of handling the message level security that we prototyped with OIDC4VP...but how future registries works is TBD by WG process
- SamG: Can we just omit?
- Kyle: May be missing context. What are we trying to address with encryption if we can guarantee we are passing the data on?
- JohnB: There could be a man in the middle. The wallet may have the ability to validate the encryption key so that only the truster verifier can encrypt.
- Martin: Is the key negotiation expected to be magic?
- JohnB: It is encoded in OpenID4VP; original proposal had the key in the top level of the API.
- Kyle: So this is the case where there are intermediary OPs acting as verifiers in OpenID4VP?
- JohnB: Yes, either intentionally or unintentionally.
- Kyle: So this is where you are breaking the original model, and you are compensating for that — this answers my question.
- JohnB: If this was just a matter of encrypting to a random key, that wouldn't make sense, I agree.
- Tim: Can we remove this as a blocker for the first CG report?
- (now have 2 open items)
- Looking at "Add JSON deserialization methods"
- Lee: We need to be able to serialize.

- Tony So would the serialization be a new API.
- Tim: They would be methods.
- SamG: Is it fair to say this is entirely polyfillable?
- Lee: This is just an object that you can fill any way and it gets serialized. We just need to state somewhere that all data needs to be serializable.
- SamG: Can we clarify — if this is just spec text to say “this information must be JSON serializable”, that is one thing, but also the proposal was to create new methods.
- Lee: This serialization is defined in CTAP2.2 hybrid too, so what is needed here is similar.
- MAttM: Is there a way to describe this such that we can *not* require serialization.
- Lee: You have to define it somewhere.
- MAttM: In WebAuthn there is a 2JSON method that must be called explicitly, but if the expectation was just that the data would be serialized somehow, we wouldn't need the method
 - If the API was designed such that you could just specify the strings, there would be no need for the consumer to understand.
- Lee: We had that at the beginning, but we changed it back. SamG, do you remember why?
- SamG: Had to do with CTAP/hybrid.
 - In the web class, there is `json.stringify` and `json.parse`. Those are well-defined and accomplish this goal. The results of those two functions give you the data that can be serialized.
- Lee: You're still going to have to comply around objects that aren't serializable.
- Matt: Is there going to be a challenge that some data may introduce non-serializable values?
- Tim: That would be part of the registry discussion. That might work to be enough for now.
- Lee: I still prefer the methods. I think it makes it clearer to have something to point to, but if it is just for now, it could be ok.
 - It could just be one sentence. We did that for WebAuthn because it has some non-serializable things in it. But we can make one line work.
- Tim: SamG, will you make a new issue?
- SamG: Let's just revise this one.
- Tim: But don't you want to keep this one for later?
- SamG: Let's keep it as an issue and discuss in the CG calls.
- Tim: That leaves us with the “check protocols” issue (#168), which is easy.
- Questions on the Oct 31 CG report?
 - Assuming we have the doc on time, it would be published Nov 3 or 4.
- Tim: Any updates from Webkit?
- Marcos: Nothing additional — it's looking good, excited. To Sam's point, let's not add stuff until we need stuff.
- Lee: We can choose between a query (is this protocol supported?) or an enumeration (show all registered protocols).

- Marcos: Question is, what do we filter on, and what gets passed through the OS? It isn't about [real-time] wallet support, it's about what's in the spec, in the abstract.
- Tim: We also added patterns to WebAuthn for disclosure.
- Kyle: Isn't there an issue with returning different thumbprints?
- Martin: It would just be, "this product supports these 3 things"; it wouldn't have any OS details, etc.
- Ben: Confirming that world view.
- Marcos: As a developer, you don't want to show a thing that is guaranteed to fail.
- [Issue #139 - Martin](#)
- Martin: It turns out that this technology is not mature yet — I want to make sure it is properly documented; I don't think it is right now. I think we have it covered.
- Lee: You need to talk about a set of issuers, not a specific issuer.
- Martin: Yes, you need to talk about a set (some disagreeing about disagreeing) you have a situation where people are excluded from participating because the only issuer that could issue for them is something like North Korea, that would result in rejections at verifiers anyway. It is critically important to understand this — the cop-out is that you let the verifiers choose, but that is a bad outcome for equitable access. Let verifiers discriminate based on whether you are from Louisiana, Texas, or Mexico, for example.
- Kyle: To add, one use case is social media — being able to assert that social media is only used by people in your country. This opens doors to certain speech potentially — from a technological perspective, not supporting that use case may make it clear that this kind of use case should not be done — saying no is sometimes just as important as saying yes.
- Lee: Any issuers can issue; it doesn't imply these are government IDs. If you make an argument only based on the "gov ID" use case, you may need to rely on RP regulation, but we have to support all the variants.
- Ben: It is useful to think of this beyond gov creds, but the decisions we are making are specifically for the gov use case. Yes, there are other federation use cases, but they probably wouldn't use an mDL.
- Lee: Yes, this is an accelerator, I agree, but we (Google) see a decent number of non-government immediate use cases; gov is the driver but it...
- Hadley: If the question is, "what's my address?" I might be picking up somebody else's sofa, I may need that mDL data as the best authority.
- Lee: If you are getting a delivery address, then maybe this doesn't make sense, but if you need a verified gov't address, then you're right.
- Kristina: I may be missing context but can't get away from feeling that the problem being raised is different from the selective disclosure issue. The wallet should be certified, which should mean it has been made very clear what is expected. During issuance, the issuer has to determine whether the wallet is capable of keeping the data safe, protecting keys. There are many things that the browser won't be able to guarantee.
- Issue #167 — What is returned from API when issuance process ends?

- Tim: Gotten market feedback that issuance is important, too, but we've punted it for now. Only main difference from presentation is that there's no structured response back other than an ACK. One of the questions is, what does the ACK look like? What does an issuing site need from a response? For structuring, we'll need a separate registry. Big question is, do we want to follow the same pattern of largely opaque except for the response, and what does the response look like?
- Sam: I personally haven't looked at issuance enough myself. It's not clear to me why there are different protocols — why do we have OpenID4VCI and OpenID4VP. From an OS affordance perspective, it's more of a wallet selector than a credential selector.
- Tim: Right, the user is selecting where to put it.
- Sam: We moved from `identity.get` to `credentials.get`, which means we got `credentials.create` for free. You're right; we'd draw a wallet selector on the platform side. One of the reasons we need this is that the EU is asking us to support these issuance APIs, and we need it for some of the large scale pilots. We've also been talking about payments, which need issuance.
- Brian: I'm not sure I see the similarities between issuance and presentation at our layer. Just cautious, as I suspect it won't be similar. Significant differences at the layer of OpenID.
- Tim: Yes, I was just thinking solely at the web platform layer — there's a protocol and request and need to get out of the browser into the platform.
- Brian: Yes, at some level
- Tim: Complexity at OS/platform layer is different. But at the web API layer it's very similar.
- Lee: We expect what's in the request JSON to be wildly different, but the web API when we're just plumbing into the wallet, looks very similar.
- Kristina: Technical discussion or requirement discussion? I want us to stay at the higher-level requirements level. Request is going directly into the wallet, and the response is going through all these layers. Give me claims from this credential from this wallet going through. Wallet may have a back end. In issuance, the flow is a wallet asking the issuer to issue a credential. If the wallet knows the issuer, we don't need this. But if the wallet doesn't know the issuer, the wallet asks to reach the user to say whether to start an issuance with some issuer. Structure has a very different meaning; what's passed over the API is "I'm an issuer, you can find me here, this is the credential I'm trying to issue to you". Whether that means it has to be a different structure in the API, I don't know, but it impacts the requirements right now.
- Tim: Equivalent is the "save to wallet" button that shows up on the web today. Or if I'm in an app and I want to save my credit card to google wallet, same thing.
- Lee: Yes, standard way — gym membership, library card, etc.
- John: Who's actually triggering this API when we consider what's going to be handed back? 3p shouldn't be able to say whether the wallet got the credential. Should return nothing to the RP unless there's some guarantee that it's only the issuer.
- Martin: Given the nature of interactions when you pull something out, you're not concerned

- John: I'm assuming who presenting is different from who issued it. I'm concerned that saying whether or not it was issued might leak information to someone not the issuer. Need to make sure it's the issuer. There may be cases where some RP asks to go to an issuer. May be cases — should a 3p be able to say "go get a credential from here". But we shouldn't say whether or not it was provisioned. You can ask for it separately.
- Martin: Useful to know if it was stored or not.
- John: Issuer will know via a back channel — whether it was successfully provisioned. Question is whether or not the site knows.
- Martin: They can talk to the issuer.
- John: Probably want the web API response to be just "yes, I heard you — sent to a wallet".
- Helen: We might want to say it's finished so 3P can move on.
- Lee: Need to be able to handle case where there's no wallet.
- Jon: Yes, saying it was processed by some wallet.
- Helen: I'm sensing we kind of want to support issuance; if so, do we get hybrid for free?
- Lee: Yes.
- Tony: Trying to force an issuance protocol. There's many today for mobile driver's licenses — at least 2 if not 3. If you're forcing a particular issuance protocol, that may cause some issuances. We have the VC that CA uses today; we have mDL that CA uses today; and they're different.
- Tim: Presentation work is to improve UX, security, phishing. Issuance
- Tony: You're forcing one in particular?
- Tim: Yes, we will have a registry that can list multiple. Probably have two registries; one for presentation, one for issuance.
- Tony: Ok. How do you know that the data issued by the issuing authority hasn't changed since the issuance? At protocol layer?
- Tim: Yes, from web platform layer, it's one time fire and get response back
- Kyle: Jon said that by returning a response, might leak data with privacy concerns. But I think you could issue and then verify, and if verification didn't pass, then revoke the credential.
- John: Not about success or failure from issuer's perspective. If I'm a 3p website and I can offer "issue CA drivers license", I shouldn't be able to learn whether it was issued without getting the user's consent to disclose it.
- Nina: The alternative is worse — if you get back an error saying CA mDL can't be issued, now I know.
- John: Yes, we have to be careful with what's exposed to 3ps.
- Kyle: Can you combine this with a presentation in order to perform a bypass? It's a little different if you control the credential and control revocation later. Thinking at a higher level, let's not look at issuance in isolation, but how is the combination able to produce the leak?
- John: Not normally combined, but these are the sorts of things we need to think through. Issuance is complicated; different beast than presentation,
- Tim: Is this simply a dynamic deep link for the platform to help you find the right app? Or is it more than that?

- Lee: I think you'd only get two things back: user canceled (user declined or there is no wallet), or get "this made it to a wallet" but you should check back end to see if it got issued.
- Tim: We have our senior director of error codes on the web platform here, Matt Miller.
- David Waite: Big issue with issuance, compared with WebAuthn you're kind of self-issuing. But for abstract credential type, we don't have a surefire way for a protocol to get some assurance from the wallet that the site interaction is appropriate. Second issue is that you can have different protocols. If you're talking about something any wallet could hold, it could be as simple as a direct push of the data. If you're talking about something more complex, then you start getting into cryptographic exchange and knowing the identity of the wallet, whether it has the properties necessary to hold it and that's multi-step. And unless the abstract API here is multi-step then it's going to be an interaction with a back-end.
- Kyle: Probably a better example is probably credential re-issue, perform verification of the old credential that may not be expired yet, then issue new one, verify new one, then expire old. That seems a very useful use case why someone would do this with a driver's license, for example. In that exactly you have to think through the combination of the two APIs.
- Tim: Going to wrap here. Issue #167 that Sam started, please add comments there. Now in 30 minutes we have a joint meeting with Web Payments on DC API and payments.