Spanish Data Protection Authority
AEPD, Agencia Española de Protección de Datos
Calle Jorge Juan, 6. 28001 Madrid, Spain
https://www.aepd.es/
División de Innovación Tecnológica
dit@aepd.es

Response to ARF Discussion topics:

Topic F "Digital Credentials API (former known as browser API)"

We want to make the following observations concerning the discussion paper's global approach:

- It is not clear to us if the goal of this discussion topic is to define high-level requirements for the interface between the wallet and browsers and/or the operating system or to decide that EUDI wallets will rely only on the Digital Credentials API.
    - *Requirement 3: "Wallet Units and Relying Party Instances SHALL support the Digital Credentials API for remote presentation flows"* is not the same as "Wallet Units and Relying Party Instances SHALL support at least one API for remote presentation flows that meets all the specified requirements".
- The decision to enforce the use of the Digital Credentials API is taken practically from the beginning of the discussion paper without adequately justifying it.
- Furthermore, it is not clear if other communication channels between the wallet units and the remote Relying Party instances are going to be supported. Arguments such as *"variability in user experiences across different browsers and operating systems"*, *"operational inefficiencies"*, or *"potential security risks"* are mentioned in the Problem Statement section. However, they are not sufficiently elaborated or discussed, and evidence is not presented. This type of evidence-based discussion should be introduced in this discussion paper if the goal is to enforce a specific communication channel or to decide that the ARF will only support a particular API.
- Many wallet products and pilot projects already employ different communication channels without the need to introduce browsers as intermediaries or to use this specific API. What specific use cases/challenges justify the need for this type of API that involves so many risks? Is it necessary to implement OpenID4VP over a browser API, and specifically, over the Digital Credentials API?
- The discussion paper establishes that *"Digital Credentials API is a possible solution to the identified challenges. Digital Credentials API has the potential to enhance usability, scalability, and security while providing a consistent and reliable user experience"*. But again, the challenges have not been discussed enough, nor how

this specific API addresses them (and other possible solutions don't) or to what extent it does so.

- It seems that this specific API is being chosen when 1) it is not clearly stated why/when it is necessary for communication between wallets and the RP to go through a browser (wen layer), 2) the requirements that the interface between the wallet and browsers and/or the operating system should meet have not yet been defined 3) the Digital Credentials API has not yet been fully specified.
- Furthermore*, "As of January 2025, Digital Credentials API support is provided only by the Chrome browser and the Android mobile operating system ".* Have aspects such as technological neutrality, resilience, digital sovereignty or the need for European citizens to be able to decide which browser or operating system they want to work with been considered?
    - Later in the document, we can read, "*Furthermore, the use of the Digital Credentials API SHALL provide cross-platform interoperability, ensuring users are not locked into a specific vendor's browser or operating system".* How can this be guaranteed if it has already been decided that a specific API is going to be used (at least, no other alternatives have been mentioned in the discussion paper), and this API has not yet been fully defined? For example, what is going to happen if the Digital Credentials API is finally not neutral and open with respect to the format of attestations to be used?

In addition, we would like to comment on some specific points discussed in the paper:

- Consent in the discussion paper is not GDPR consent, and this kind of confusion should be avoided. Please use permission, acceptance or acknowledgement instead.
- It is unclear if the API can work with Private Browsing or Incognito mode. It is also unclear how the API is going to support pseudonyms.
- When the browser acts as an intermediary or proxy between the wallet and the RP, a shared responsibility (wallet, browser) must be explicitly clarified.
    - For example, concerning the transparency principle. The discussion paper says, "*The browser presents to the User a selector that includes a list of potentially suitable attestations"* We can see in the screenshots the Request origin, info that will be shared and Details (what details?). The user needs to know the types of data requested by the RP, the purpose, etc. Even the level of data protection risk involved in the request. Who is responsible for what? All the information provided by the wallet or the browser should be consistently presented in plain language and in a machine-readable format.
    - Reading the discussion paper, it is not clear if the browser is only "passing" the information through the API or if it is expected to have an active role in

inspecting requests and providing additional support to users. Again, it is essential to identify the browser's responsibilities clearly:

- *"Browsers may present wallets that meet selective disclosure requirements but do not directly enforce it".*

But, at the same time:

- *"Browsers SHALL evaluate, block, or warn users about potentially untrusted verifiers requesting wallet information."*
- *"Browsers SHALL not decide which verifiers are authorized to request attributes; this responsibility lies with national issuers and regulators".* But are they required to implement the provided methods (allow/block lists) without discussion or flexibility? In the previous point, it is said that browsers *SHALL block*.

- If we do not want the wallet to be reduced to a simple “attestation storage”, the browser should have almost no responsibility in the entire attestation presentation process. The proposal made in this paper could be interpreted as a way to centralize digital identity management again, with a very limited number of browsers (one at this moment) responsible for essential functionality.
- Furthermore, using the browser as an intermediary during attestation presentation implies different threats through malicious browser extensions or data combination, for example. It has to be considered that this approach makes personal data more accessible through browsers. Therefore, browsers must maintain strict privacy controls to prevent unauthorized access. These controls should be listed and discussed explicitly, expressed as requirements for the API, for example mechanisms to minimize OS/browser access, secure browser environment, or end-to-end encryption (wallet-RP).
- To mitigate Detection threats and fingerprinting, the API must return the same messages and errors and behave the same (including timing) whether one or more wallets are installed, whether they have the required credentials (valid or not), whether the user provides permission to present them, etc. Nothing should be inferred from observing the API's behaviour.
- We miss details about the trust management model: device -> OS -> wallet -> browser -> RP. Can we guarantee privacy and data protection even when one or more of these parties cannot be trusted? How can they be trusted, and what are the minimum requirements (expressed in protocols metadata, API parameters, trusted registers, etc.)? Aspects concerning governance and accountability should be carefully addressed.

All of the above makes section 3.4, Privacy Preservation, although welcome, incomplete and not concrete or specific enough. Section 1.2 (Related risks in the Risk Register) is of no use if the identified risks are not subsequently discussed in detail and the conclusions of this discussion are not incorporated as requirements in the new version of the ARF.