

Electronic Attestation of Attributes Extended Validation Services

Luigi Castaldo¹, Gianmario Cortese¹, Simone Izzo¹ and Fabrizio Balsamo¹

¹Namirial S.p.A, Italy

Abstract

We propose an advanced credential validation schema managed by the European Digital Identity (EUDI) Wallet. This new protocol preserves Users' privacy and unlinkability while allowing direct privacy-preserving communication between the Credential Issuer and Credential Verifier. This protocol utilizes Hierarchical Deterministic Key Derivation, a mechanism in which each derived private key is generated in a manner that allows the computation of the corresponding public key without revealing the private key itself. As a result, only the Credential Issuer is capable of generating the corresponding decryption key. The Credential Verifier cannot access the credential without a direct communication with the Credential Issuer. The communication happens without sharing any info related to the Credential Holder and allows the Credential Issuer to count the number of verifications performed by every Credential Verifier. This mechanism enables the development of a pricing policy for credentials and advanced business models that facilitate the effective large-scale adoption of the EUDI Wallet, which we analyze in this study.

Keywords

European Digital Identity Wallet, eIDAS, Digital credential, Architecture and reference framework, OpenID for Verifiable Credential Issuance, OpenID for Verifiable Credential Presentation, Business Model

1. Introduction

eIDAS 2.0 [1] provides the legal framework for the European Digital Identity (EUDI) Wallet, designed to allow citizens and businesses (public and private sectors) to store, manage and present their credentials, such as identities (name, date of birth, tax number, nationality, etc.), certificates, diplomas, professional credentials and more, securely in online and offline, national and cross-border use cases, and electronically sign or seal documents as an individual, as a legal person or as a representative.

To build an interoperable system, the European Commission, through its Architecture and Reference Framework (ARF) [2] and the EUDI Wallet Toolbox, provides a structured approach to the design, implementation and operation of the EUDI Wallet. The Framework itself represents the core architecture that defines the technical specifications, guidelines and architectural principles to ensure consistency and interoperability across different EUDI Wallet implementations.

To operationalize the vision of eIDAS 2.0 and the ARF, the digital identity ecosystem revolves around three pivotal roles, namely the *Credential Issuer*, the *Wallet Holder* (User), and the *Credential Verifier*.

At the core of this framework, there are two foundational technical properties focused on preserving security and privacy of the EUDI Wallet User [1, Article 5a §4 (a) and §16 (a) and (b)], namely selective disclosure and unlinkability. The first means a User shall be able to share only specific attributes from their credential, instead of revealing it in its entirety, ensuring data minimization and enabling them to prove necessary information without exposing unrelated personal data. The latter means that transactions of the same User cannot be linked or traced; there are three forms of unlinkability: it can be with respect to Credential Verifiers, namely they cannot correlate a Wallet Holder's interactions across different services, preventing them from linking activities to build a profile of the Holder; with respect to Credential Issuer, namely they cannot track or identify where and how a Holder presents their credential to Credential Verifiers, preventing them from monitoring the Holder's activities or

TDI 2025: 3rd International Workshop on Trends in Digital Identity, February 3, 2025, Bologna, Italy

✉ l.castaldo@namirial.com (L. Castaldo); g.cortese@namirial.com (G. Cortese); s.izzo@namirial.com (S. Izzo); f.balsamo@namirial.com (F. Balsamo)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

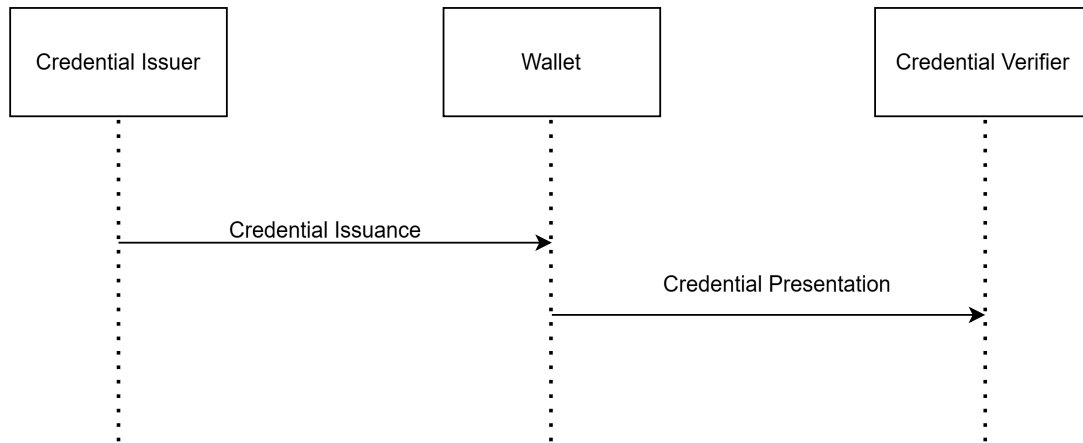


Figure 1: High level main components interaction.

interactions; finally, with respect to both Credential Verifiers and Credential Issuer, namely this present case guarantees unlinkability even if Credential Verifiers and Credential Issuer collude.

In this document, we introduce an advanced model designed to address key challenges in credential presentation, with a dual focus on preserving User privacy and establishing a foundation for sustainable business models and pricing policies. Our approach ensures User unlinkability while enabling direct, privacy-preserving communication with the Credential Issuer and the Credential Verifier. Furthermore, the lack of a well-defined business model motivates the introduction of a new policy, the pricing policy, associated with the VC issued by the Credential Issuer. When combined with the proposed approach, this will enable the introduction of a sustainable business model. This mechanism allows for credential verification without disclosing any information about the Wallet Holder, counting the number of verifications performed by each Credential Verifier. By incorporating this capability, the model not only strengthens privacy protections but also provides a structured basis for defining diverse credential monetization strategies, facilitating scalable and sustainable adoption of the EUDI Wallet.

The remainder of this paper is structured as follows: Section 2 presents research related to our work. Section 3 presents credential verification background and our extension to the credential validation framework. Section 4 is dedicated to its benefits and in Section 5 and 6 there are limitations and privacy consideration, respectively. The paper concludes with Section 7.

2. Technical Background

The EUDI Wallet ecosystem is built upon the interaction between the Wallet and various Credential Verifiers. Notably, it places significant emphasis on the process of credential presentation and subsequent verification. The successful execution of this process is contingent upon the orchestration of numerous components, each of which plays a pivotal role in safeguarding the integrity and dependability of the entire ecosystem.

In this context, various specifications for credential formats, credential issuance and presentation have emerged.

Main actors in this ecosystem are:

- **Credential Issuer:** A role an entity can perform by asserting attributes about one or more subjects, creating credentials and transmitting the digital credential to the Wallet Holder. This process is performed using OID4VCI protocol.
- **(EUDI) Wallet Holder:** The user who has control over their EUDI Wallet. The EUDI Wallet grants user full control over their digital credentials, ensuring privacy, security, and ease of use. It provides seamless experience for managing, acquiring and presenting digital credentials.

- **Credential Verifier:** A role an entity performs by receiving one or more digital credentials. This process is performed using OID4VP protocol.

2.1. Protocols for Credential Issuance and Presentation

Protocols for issuing and presenting credentials are crucial to ensure secure and interoperable interactions between Credential Issuers, Wallet Holders, and Credential Verifiers. Key protocols include OpenID for Verifiable Credential Issuance (**OID4VCI**) and OpenID for Verifiable Presentations (**OID4VP**).

OID4VCI [3] defines a standardized process for Verifiable Credential (VC) issuance using the OAuth 2.0 [4] and OpenID Connect frameworks. It provides a mechanism for Wallet Holders to obtain VC from Credential Issuers. In particular, Wallet Holder initiates the process by requesting the issuance of a specific VC; then authenticates with the Credential Issuer using an OAuth-based flow to receive the VC.

OID4VP [5] extends OAuth 2.0 to facilitate secure VC presentation. The credentials are digitally signed, providing authenticity can be cryptographically verified. This specification defines a mechanism on top of OAuth 2.0 that enables presentation of VC as Verifiable Presentations, meaning a Holder-signed credential whose authenticity can be cryptographically verified to provide Cryptographic Holder Binding.

3. Digital Credential Verification

To verify that a Digital Credential is valid and has not been suspended or revoked, the ARF [2] proposes the use of status lists or revocation lists. In this approach, the Digital Credential must include revocation information, which contains a URL pointing to the chosen list and an identifier or index that allows the Credential Verifier to locate the specific credential within that list.

However, this method fails to meet the privacy and unlinkability requirements mandated by the eIDAS 2.0 Regulation [1]. The core issue lies on one hand, in the Credential Verifier's reliance on the Credential Issuer for credential status verification; this interaction can reveal the identity of the Wallet Holder presenting the credential, allowing the Credential Issuer to track and correlate their interactions with different Credential Verifiers, ultimately compromising the Wallet Holder's privacy and unlinkability. On the other hand, once the Credential Verifier receives the parameters for a status list, it may persistently store the URI and index, enabling it to check the status list again at a later time. For instance, consider a scenario where a Credential Verifier's repeated access to a status list, such as the one defined in [6], to check the revocation status of a credential could be deemed as excessive monitoring of the Wallet Holder's activities.

At present, the ARF endorses the OID4VP protocol for credential presentation. However, even if it defines how a Wallet Holder can present credentials to a Credential Verifier in a standardized way, ensuring interoperability across different systems, it only addresses the technical aspects of credential presentation.

As documented in [7], without unlinkability, adversaries (including Credential Verifiers, Credential Issuers or third parties) may track Users, creating risks such as unwanted profiling, surveillance, and data leaks. Unlinkability is achievable as long as the disclosed claims do not contain information that directly identifies the User. For example, if a User presents their birthdate to one Credential Verifier and their postal code to another, the Credential Verifiers should not be able to determine that they are interacting with the same User.

To achieve unlinkability, [7] proposes batch issuance as an effective approach. However, as documented in [8], issuing multiple credentials in batch can only partially mitigate unlinkability with respect to Credential Verifiers.

This work presents an extension to the digital credential validation framework that not only addresses the concerns outlined but also introduces a flexible and comprehensive solution. Furthermore, this advancement enables the development of diverse business models associated with Verifiable Credentials, encompassing issuance, validation, and unrestricted free usage. At present, only issuance and free usage are achievable without the proposed approach.

3.1. Extended Validation Service(s)

Our proposal does not require changes to the existing credential issuance process. It is **format-agnostic**, maintaining compatibility with all available Digital Credential formats.

The concept involves two steps:

1. **Refreshing** the credential before every presentation, to guarantee the validity without the need for the **Credential Verifier** to undertake additional controls, except integrity and authenticity of the credential.
2. **Cyphered VC**: the credentials are encrypted by the Holder's Wallet using a mechanism described later in this document, before being shared with a **Credential Verifier**. The Credential Verifier can only access the credential through direct communication with the Credential Issuer. This communication occurs without revealing any information about the Credential Holder and allows the Credential Issuer to count the number of verifications performed by each Credential Verifier.

3.1.1. Credential Refreshing

The concept involves periodically refreshing VCs by communicating directly with the Credential Issuer. This could be achieved with 2 different approaches:

- **Linked credentials**, as documented in [9], the Credential Issuer provides the Wallet Holder with a Status Assertion, which is linked to a Digital Credential. This enables the Wallet Holder to present both the Digital Credential and its Status Assertion to a Credential Verifier as a proof of the Digital Credential's validity status.
- **Credential reissuance**, based on multiple access to the credential endpoint [3]. It is possible to refresh an issued credential, the Wallet can retrieve an updated VC using a valid access token or refresh it with a valid refresh token, without the interaction with the User. If the Wallet lacks both a valid Access and Refresh Token, the Credential Issuer must reissue the Verifiable Credential by initiating the issuance process from the beginning, which requires interaction with the User.

This Credential Refreshing could be done when the Wallet starts up, on demand, or before presenting the VC.

Those approaches would apply only to credentials that require refreshing. For example, it would not be useful for static credentials. The Credential Verifiers can implement their own policies based on the type of service provided. For instance, they could accept a VC refreshed within the last N hours, or for more critical services, only accept credentials refreshed at the last minute.

The main benefits are:

- **No need for complex validation mechanisms**: since the VC is refreshed directly from the Credential Issuer, third parties are not required to validate it.
- **Flexibility for both Credential Issuers and Credential Verifiers**: different policies can be implemented based on service type and level of criticality.

3.1.2. VC Cyphered Presentation

Verifiable Credential is encrypted by the User's Wallet. The Credential Verifier must then contact the Credential Issuer to retrieve the decryption key in order to verify the credential. This adds an additional layer of security by ensuring that the Credential Verifier cannot access the VC directly without authorization from the Credential Issuer.

As previously mentioned, the protocol does not imply any change to the current credential issuance process. The Credential Issuer creates and signs the VC, embedding relevant identity or attribute information for the User (e.g., identity, access rights, etc.). Then the VC is transmitted to the User's Wallet over a secure channel using TLS.

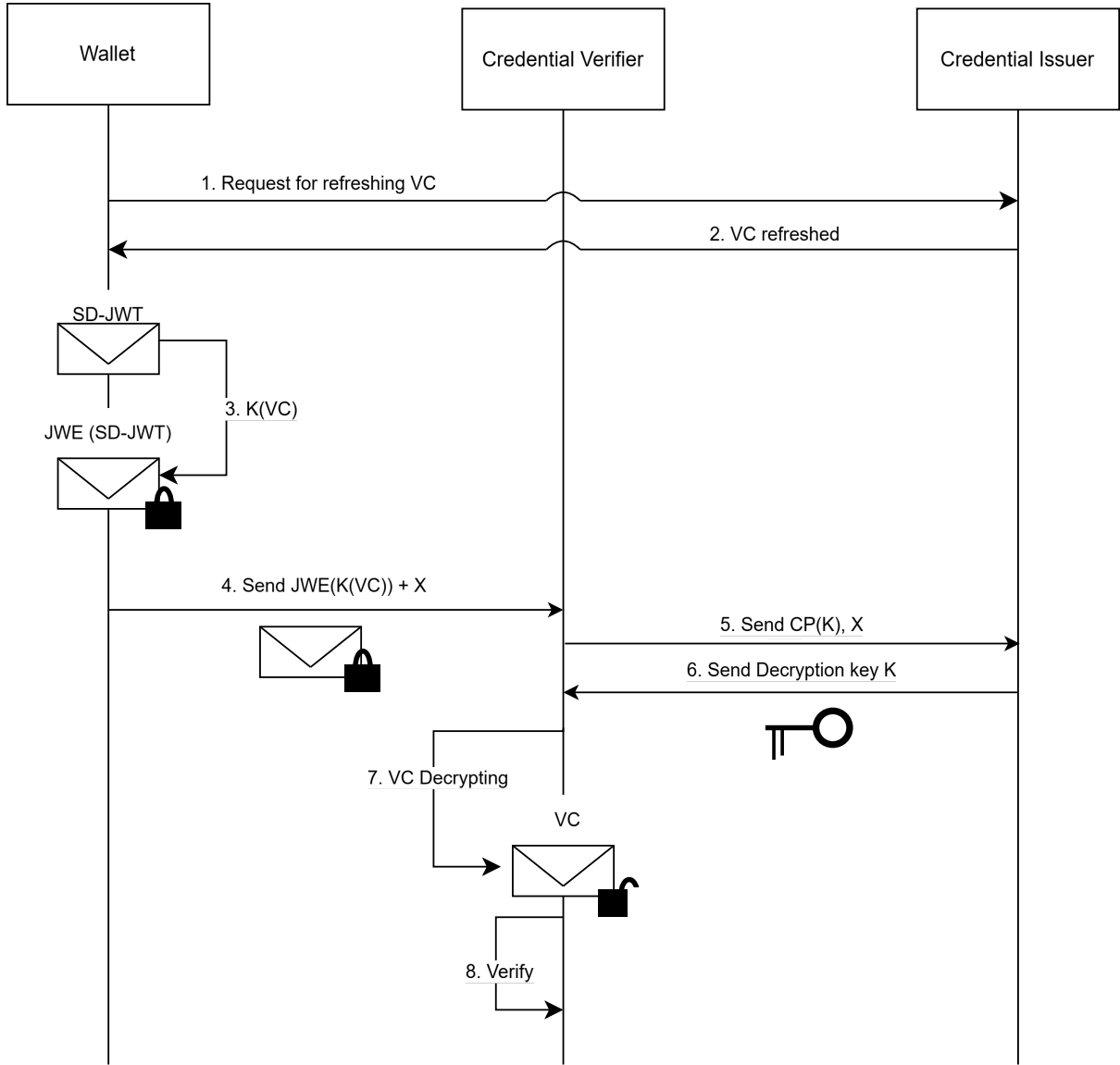


Figure 2: VC Cyphered Presentation flow.

The proposed method involves encrypting the **Verifiable Credential** at the time of presentation, leveraging the properties of elliptic curve cryptography. Specifically, the **Hierarchical Deterministic (HD) Key Derivation**.

In particular, the method takes advantage of the properties of Hierarchical Deterministic structures, where each derived private key is generated in such a way that the corresponding public key can be computed without knowing the private key itself.

When a Wallet initiates the presentation of a Verifiable Credential (which could be in various formats, such as SD-JWT VC or mdoc), the credential will be encrypted according to the **JWE** standard [10]. The encryption key for the VC will be generated at the time of the new presentation, it is derived from the Credential Issuer's public key.

In order for the **Credential Verifier** to decrypt the JWE, the Wallet must also send them additional information along with the JWE. This will be forwarded to the Credential Issuer. Upon receiving the necessary information, the Credential Issuer retrieves the cryptographic material using which the Credential Verifier can obtain the VC.

More in detail, the expected flow of messages is listed below, subdivided into eight steps:

- 1. Wallet sends a request to refresh the VC**, as described in the previous section.

2. **Credential Issuer sends an updated VC**, notifying the Wallet in case the VC is not valid anymore.
3. **The Wallet generates a transaction-specific symmetric encryption key K using which it encrypts the verifiable credential VC**. The result of the encryption process is denoted here as $K(VC)$.
4. **The Wallet generates the JWE of the Verifiable Credential and send it to the Credential Verifier** as follows:
The header of the JWE contains, as a plaintext, all of the necessary information for decrypting the body of the JWE itself; this includes:
 - X: a concatenation between a timestamp and a random nonce generated by the Wallet;
 - KID: a Key Identifier, needed to identify the correct Credential Issuer and then ask this Credential Issuer to decrypt $CP(K)$ (see below);
 - $CP(K)$, i.e. the key K encrypted using an asymmetric public key CP belonging to the Credential Issuer and derived from a master public key of the same Credential Issuer.
 The body of the JWE contains $K(VC)$, that is the Verifiable Credential VC encrypted using the symmetric key K (see step 1). At the end, the Wallet sends the JWE and X to the Credential Verifier.
5. **The Credential Verifier receives the JWE and sends $CP(K)$ and X to the Credential Issuer**, asking the Credential Issuer to use the derived private key linked to X in order to decrypt $CP(K)$;
6. **The Credential Issuer retrieves from X which private key to use in order to decrypt $CP(K)$** ; after doing so, it obtains K and sends K to the Credential Verifier.
7. **The Credential Verifier uses K to decrypt $K(VC)$** , thus retrieving VC.
8. **VC Validation**: once decrypted, the Credential Verifier checks the Credential Issuer's signature on the VC to ensure that it is valid and has not been tampered. The Credential Verifier can now process the VC based on the User's attribute attestation (e.g., identity verification, access rights, etc.).

3.2. Attestation Rulebooks catalogue

To achieve a high level of interoperability, it is essential to establish a common framework for Verifiable Credential Providers. This framework must go beyond technical specifications, standards, and protocols to include a shared semantic scheme for Verifiable Credentials. That is why, since version 1.2.0 of the ARF document, the concept of Attestation Rulebook¹ [2] has been introduced (each attestation has an *attestation type*² and *namespace(s)*³ it uses), as well as the concept of using catalogues.

To address these challenges, Credential Issuers could utilize the **Attestation Rulebooks catalogue** to publish their attribute when necessary. This catalogue mitigates the risk of uncontrolled implementation practices that could degrade system quality and increase complexity and maintenance costs.

This catalogue would serve as a comprehensive repository of credential-related information, including **Credential Metadata**, which provides essential details for identifying, verifying, and contextualizing credentials, as well as information on any associated policies.

Analogous to **Credential Issuer Metadata**, where the parameter "*credential_configuration_supported*" lists the IDs of credentials available for issuance along with their corresponding metadata, Credential Metadata consists of various parameters as defined in [3]. To enhance this structure, a new parameter, "**pricing_policy**", can be introduced. This parameter would specify all relevant details regarding the applied policy, including pricing model, price, currency, and a URI linking to the Credential Issuer's detailed policy page.

¹Some Rulebooks have already been defined by the European Commission, in consultation with the eIDAS Expert Group, namely the PID Rulebook, the mDL Rulebook and the Pseudonym Rulebook.

²For each type of attestation (PID, QEAs, PuB-EAs, EAs), in essence, an Attestation Rulebook specifies the attribute schema, data format and proof mechanism of that attestation.

³The namespace(s) used by an attestation define the identifier, syntax, and semantics of all attributes that can be part of that attestation.

```

"SD_JWT_VC_example_in_OpenID4VCI": {
  "format": "dc+sd-jwt",
  "scope": "SD_JWT_VC_example_in_OpenID4VCI",
  "cryptographic_binding_methods_supported": ["jwk"],
  "credential_signing_alg_values_supported": ["ES256"],
  "pricing_policy": {
    "pricing_type": "verification_based",
    "price": "0.01",
    "currency": "USD",
    "business_model": "https://generic_issuer.com/credential_price_info"
  },
  "display": [
    {
      "name": "IdentityCredential",
      "logo": {"uri": "https://university.example.edu/public/logo.png"},
      "locale": "en-US"
    }
  ],
  "proof_types_supported": {
    "jwt": {
      "proof_signing_alg_values_supported": ["ES256"]
    }
  },
  "vct": "SD_JWT_VC_example_in_OpenID4VCI",

```

Figure 3: Credential Metadata.

While multiple pricing models can be applied, we outline three example approaches:

- **Issuance based:** the User must pay to obtain the new credential.
- **Verification based:** the Credential Verifier must pay the Credential Issuer for a credential verification.
- **Free:** no payment is required.

These models serve as illustrative examples, highlighting the flexibility in defining pricing strategies within EUDI Wallet ecosystem.

Figure 3 is an example snippet of Credential Metadata, where the metadata for a specific credential includes a newly proposed parameter “**pricing_policy**”.

The proposed solution could also allow to apply different price options:

1. **Static price**, which refers to a pricing strategy where the cost of the service remains unchanged; in practice, a static price is fixed and does not undergo frequent updates. This can be embedded directly into the attribute metadata, clearly exposing the unit price per verification. This transparency allows the Credential Verifier to quickly assess whether they are willing to accept the price or opt not to use the attribute, simplifying decision-making.
2. **Dynamic price**, which refers to a pricing strategy where the cost of the service fluctuates based on various factors. This approach allows businesses to maximize revenue by adjusting prices in real-time or at regular intervals. The affected stakeholders must check a specific URL, periodically updated, where prices are listed. This also allows Credential Issuers to better manage the attributes offering on a daily-basis and/or based on different agreements with several Credential Verifiers.

The **OID4VP** protocol can be extended to enable interaction with the central registry in the following manner:

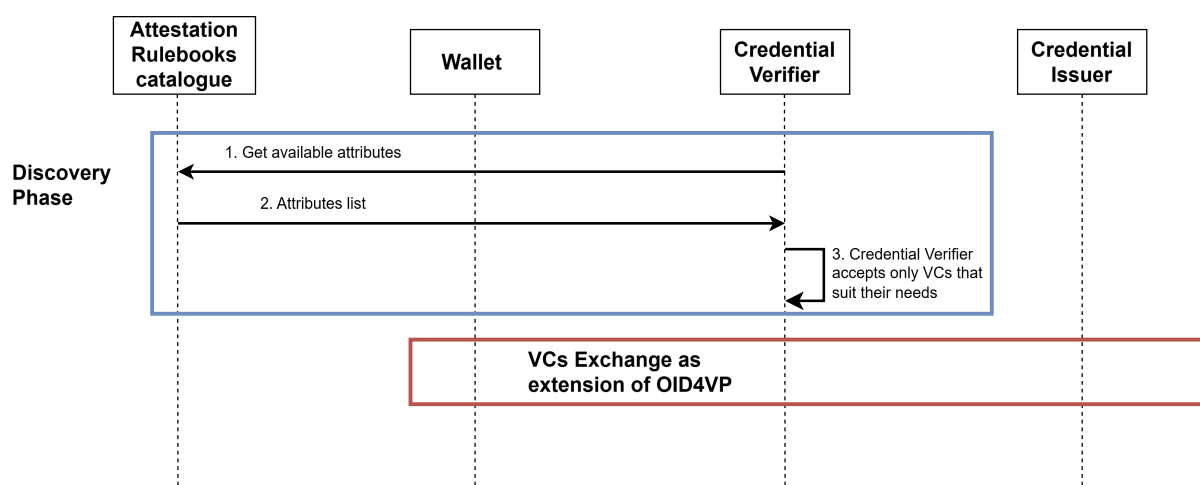


Figure 4: Discovery Phase flow.

1. **Discovery Phase:** before a Credential Verifier requests a Verifiable Presentation (VP), it queries the Attestation Rulebooks catalogue to understand the available attributes, Credential Issuers, and associated costs. The Credential Verifier includes the selected attributes in the request, specifying the willingness to pay associated verification fees.
2. **Encrypted VCs exchange:** described in Section 3.1.2.

4. Benefits

This proposal entails minimal modifications to existing models within the current ecosystem, yet these changes facilitate the introduction of a new business model. The approach is highly adaptable, allowing for seamless integration with emerging needs.

The following key benefits are worthy of note:

- **User Privacy and Unlinkability:** this approach ensures that Credential Issuers are unable to identify which User is presenting their credential to a specific Credential Verifier requesting its verification. This enhances User privacy significantly and minimizes the risk of tracking.
- **Creation of a Credential-Based Market:** the implementation of a rulebook that defines the value associated with each credential establishes a marketplace where Credential Verifiers must compensate Credential Issuers for verifying the authenticity of a credential. This model fosters a market driven by the value and trustworthiness of credentials, where the cost of verification reflects the intrinsic value of the credential itself.
- **A Credential Verifier** cannot continuously request a specific credential because the verification process involves requesting the key necessary to decrypt the received credential. Without this key, the Credential Verifier cannot access the credential's content, ensuring privacy and preventing unauthorized repeated checks.

5. Limitations

A potential limitation of this approach is that it does not work if the Credential Verifier is offline. This means that the approach is not suitable for NFC scenarios, where the Credential Verifier and Wallet could be offline. In such cases, the Credential Verifier would need to interrupt the presentation as it would not be able to reach the Credential Issuer to obtain the decryption key.

It is important to note that in any scenario where the Credential Verifier is offline and cannot contact the Credential Issuer, verifying authenticity through signature check becomes impossible, and therefore, the presentation cannot be completed.

However, if only the Wallet is offline, the presentation can still proceed, thanks to the refreshed credential that was obtained in a previous moment, provided that is recent enough to be valid for the presentation.

6. Privacy Considerations

- The Credential Issuer can retrieve user information or build a user profile after a refresh and a consecutive presentation operation. Although the presentation process is user-anonymous, the refresh process is not. Therefore, after a refreshing VC and a subsequent decryption request for a presentation (initiated by the Credential Verifier), the Credential Issuer could potentially build a user profile. However, there are three conditions to consider:
 1. **Low refresh request flow:** The ideal scenario is to have a low number of refresh requests to the Credential Issuer, so that all refreshed credentials could be effectively tracked by the Credential Issuer.
 2. **Low decryption request flow:** The ideal scenario is to have a low number of decryption requests to the Credential Issuer, so that the refreshed credentials could be properly associated with the decryption requests.
 3. **Knowledge of the Credential Verifier's behavior:** The Credential Issuer must be aware that the Credential Verifier only accepts credentials that have been recently refreshed. If the Credential Verifier does not have this restriction, the entire process would lose its effectiveness. For example, if the Credential Verifier only accepts credentials updated within the last 5 minutes, the Credential Issuer would be aware that, most likely, one of the credentials refreshed in the past 5 minutes is associated with the Credential Verifier's decryption request. However, if the Credential Issuer is unaware of this Credential Verifier restriction, false positives could occur. For instance, if a person updates their credential and, within a minute, the Credential Issuer receives a verification request, the Credential Issuer might mistakenly associate the refreshed credential with the new request, assuming it comes from the same person.
- With regard to Section 3.1.2 about VC Cyphered Presentation, it shall be highlighted that the Credential Issuer and the Credential Verifier could collude in order to decrypt the JWE: this is due to the possibility to share the entire JWE with the Credential Issuer.
- In principle, the Credential Issuer could generate a new private-public key couple for each request of a new credential. This fact would allow a tracking of the Wallet Holder's activity by the Credential Issuer, resulting in a concern for the privacy.

Therefore, it would be advisable to control the number of public keys available for the Credential Issuer: if the number of public keys increases over time, this could be an indicator of potentially malicious Credential Issuer's behavior.
- The entire mechanism allows for the Holders' privacy because:
 - The cryptographic material (e.g. the public key CP in 3.1.2) is not directly generated by the Credential Issuer. It is, instead, derived by the Wallet starting from one of the Credential Issuer's public keys in such a way that the Credential Issuer is not aware of this generation; moreover, several "child" public keys can be generated starting from the same "parent" public key belonging to the Credential Issuer.
 - The Credential Issuer never gets the VC in plain, unless the Verifier explicitly shares it with the Credential Issuer, without the Holder's consent.
- In the worst-case scenario where the Credential Issuer has issued only one credential, it would be possible to link the credential to the Wallet Holder.

7. Conclusions

In conclusion, our efforts are directed towards creating conditions for new business models that ensure compliance with unlinkability requirements, while also fostering a virtuous flywheel effect that can increase the number of use cases. This can be achieved through the involvement of more Credential Issuers of electronic attributes, who will be incentivized by a sustainable business model. Consequently, this will benefit Credential Verifiers and Users who will be more incentivized to utilize the Wallet.

The proposal aims to lay the foundations for this growth mechanism, with what we believe to be the right balance between privacy, sustainability, implementation effort, and incentives for adoption by the various stakeholders.

References

- [1] European Parliament and Council of the European Union, Regulation (EU) 2024/1183, in: Official Journal of the European Union, 2024. URL: <https://eur-lex.europa.eu/eli/reg/2024/1183/oj>.
- [2] European Commission, The European Digital Identity Wallet Architecture and Reference Framework, 2025. URL: <https://eu-digital-identity-wallet.github.io/eudi-doc-architecture-and-reference-framework/1.4.0/arf/>.
- [3] T. Lodderstedt, K. Yasuda, T. Looker, OpenID for Verifiable Credential Issuance - draft 15, 2024. URL: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html.
- [4] D. Hardt, The OAuth 2.0 Authorization Framework, RFC 6749, 2012. doi:10.17487/RFC6749.
- [5] O. Terbu, T. Lodderstedt, K. Yasuda, T. Looker, OpenID for Verifiable Presentations - draft 23, 2024. URL: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html.
- [6] T. Looker, P. Bastian, C. Bormann, Token Status List, draft-ietf-oauth-status-list-06, 2024. URL: <https://datatracker.ietf.org/doc/draft-ietf-oauth-status-list/>.
- [7] D. Fett, K. Yasuda, B. Campbell, Selective Disclosure for JWTs (SD-JWT), draft-ietf-oauth-selective-disclosure-jwt-14, 2024. URL: <https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/14/>.
- [8] C. Baum, O. Blazy, J. Dfinity, J.-H. Hoepman, E. Lee, A. Lehmann, A. Lysyanskaya, R. Mayrhofer, H. Montgomery, N. Nguyen, B. Preneel, A. Shelat, D. Slamanig, S. Tessaro, S. Eller, T. Partisia, C. Troncoso, Cryptographers' Feedback on the EU Digital Identity's ARF, 2024. URL: <https://files.dyne.org/eudi/cryptographers-feedback-june2024.pdf>.
- [9] G. De Marco, O. Steele, F. Marino, M. Adomeit, OAuth Status Assertions, draft-demarco-oauth-status-assertions-03, 2024. URL: <https://datatracker.ietf.org/doc/draft-demarco-oauth-status-assertions/>.
- [10] M. B. Jones, J. Hildebrand, JSON Web Encryption (JWE), RFC 7516, 2015. doi:10.17487/RFC7516.