Network Working Group
Internet-Draft
Intended status: Informational
Expires: 4 September 2025

L. Johansson
Sunet

3 March 2025

A reference architecture for direct presentation credential flows
draft-johansson-direct-presentation-arch-00

Abstract

This document defines a reference architecture for direct presentation flows of digital credentials. The architecture introduces the concept of a presentation mediator as the active component responsible for managing, presenting, and selectively disclosing credentials while preserving a set of security and privacy promises that will also be defined.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/leifj/wallet-refarch.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 4 September 2025.

Copyright Notice

Table of Contents

1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Introduction

Digital credentials, which assert claims about individuals, organizations, or devices, have become essential tools in modern identity systems. Whether verifying an individual’s qualifications, attesting to an enterprise’s compliance, or authorizing an IoT device, these credentials rely on secure, efficient, and privacy-preserving mechanisms for their use.

Traditional federated identity systems often rely on intermediaries or delegation, which can compromise user privacy or introduce inefficiencies. This document presents an architecture for direct presentation flows, where credentials are presented directly to verifiers without unnecessary intermediaries, empowering the data subject or their authorized representative to maintain control over the credential’s use.

At the heart of this architecture is the presentation mediator, an active software component responsible for facilitating secure and privacy-aware interactions. This mediator works in tandem with passive credential stores, verifiers, and issuers, creating a scalable and interoperable system that can adapt to diverse regulatory and operational environments.

3. Terminology

3.1. Naming the elephant in the room

The term "digital wallet" or "digital identity wallet" is often used to denote a container for digital objects representing information about a subject. Such objects are often called "digital credentials". The use of the word "wallet" is both historic, stemming from the origin of some types of wallet in the "crypto" or digital asset community, aswell as meant to make the user think of a physical wallet where digital credentials correspond to things like credit cards, currency, loyalty cards, identity cards etc.

Arguably the use of the term wallet is often confusing since it may lead to assumptions about the fungibility of identity or that credentials are exchanged as part of a monetary transaction. In this specification we will use the term "persentation mediator" when traditionally the term "identity wallet" or "wallet" has been used.

3.2. Terminology used in this specification

To anchor this architecture, we define key terms:

* A presentation mediator is an active software component that manages the presentation of credentials to the verifier on behalf of the data subject.

* A credential store is a passive repository for securely storing credentials. It supports the presentation mediator by providing access to stored credentials without performing active operations.

* The data subject is the entity the credential pertains to, such as an individual or organization.

* A presenter is the actor that delivers a credential to a verifier. While often the data subject, the presenter could also be an authorized agent or software acting on their behalf.

* A credential is a signed, structured document containing claims about a subject, issued by a trusted entity.

* An attestation is a statement about a credential, often used to validate or certify its properties, such as its integrity or scope.

* A presentation proof is a derived artifact that proves claims from a credential in a specific interaction with a verifier.

4. A Note on History

The origins of the notion of digital identity goes back to the mid 1990s. Historically, Internet protocols were designed to deal with authentication and (sometimes) authorization, i.e. the question of what entity is accessing the protocol and what they are allowed to do. Digital identity can be thought of as a generalization of the concept of a user identifier in a protocol. Today we typically use the term data subject (abbreviated as ’subject’ when there is no risk of confusion) to denote the actor whoese data is beeing acted upon by the protocol. Most internet protocols represent the data subject as a "user" identified by a single unique identifier. Identifier in use by Internet protocols were typically never designed to be unified -

each security protocol typically designed a separate structure of identifiers.

Identifier schemes such as kerberos principal names or X.509 distinguished names are often assumed to be unique across multiple protocol endpoints. This introduces linkability across multiple protocol endpoints. Historically this was never seen as an issue.

When web applications were build that required some form of user authentication the notion of externalized or _federated_ user authentication was established as a way to offload the work involved in user management from each web application to some form of centralized service. This is sometimes called "single sign on" - a term used to describe the (sometimes, but not alwasy desirable) property of authentication flows that a user can login in (sign on) once and have the "logged in" state recognized across multiple applications. State replication across multiple web application carries with it a separate set of concerns which is not discussed here.

In the late 1990s multiple protocols for "web single sign-on" were developed. Soon the need to connect multiple "SSO-systems" across different administrative and technical realms was recognized. Bridging administrative realms is often called "federating" those realms and the term "federated identity" owes its origin to this practice. The development of standard protocols for federating identity such as the Security Assertion Markup Language [SAML] and Open ID Connect [OPENIDC] were initially created in the early to mid 2000s. These protocols are widely deployed today.

The notion of digital identity evolved as a generalization of the "single sign-on" concept because modern federation protocols (OIDC, SAML etc) are able to transport not only shared state about the sign-in status of a user (eg in the form of a login-cookie) but can also be used to share information about the subject (user) accessing the service. In some cases identity federation protocols made it possible to fully externalize identity management from the application into an "identity provider"; a centralized service responsible for maintaining information about users and _releasing_ such information in the form of _attributes_ to trusted services (aka relying parties).

Federated identity can be thought of as an architecture for digital identity where information about data subjects is maintained by identity providers and shared with relying parties (sometimes called service providers) as needed to allow subjects to be authenticated and associated with appropriate authorization at the relying party.

Here is an illustration of how most federation protocols work. In this example the Subject is requesting some resource at the RP that requires authentication. The RP issues an authentication requests which is sent to the IdP. The IdP prompts the user to present login credentials (username/password or some other authentication token) and after successfully verifying that the Subject matches the login credentials in the IdPs database the IdP returns an authentication response to the RP.

A brief illustration of the typical federation flow is useful. For the purpose of this illustration we are not considering the precise way in which protocol messages are transported between IdP and RP, nor do we consider how the Subject is represented in the interaction between the IdP and RP (eg if a user-agent is involved).

Subject -> RP: Initiate authentication flow
RP -> IdP: Authentication request
IdP --> Subject: Prompt for login credentials
Subject --> IdP: Presents login credentials
RP <-- IdP: Authentication response
Subject <-- RP: Success!

Note that

* The Subject only presents login credentials to the RP

* The IdP learns which RP the subject is requesting access to

* The RP trusts the IdP to accurately represent information about the Subject

The limitation of this type of architecture and the need to evolve the architecture into direct presentation flow is primarily the second point: the IdP has information about every RP the Subject has ever used. Together with the use of linkable attributes at the RP this becomes a major privacy leak and a signifficant drawback of this type of architecture.

The notion of "Self Sovreign Identity" (SSI) was first introduced in the blogpost [PathToSSI] by Christopher Allen. The concept initially relied heavily on the assumed dependency on blockchain technology. Recently there has been work to abstract the concepts of SSI away from a dependency on specificy technical solutions and desribe the key concepts of SSI independently of the use of blockchain.

The purpose of this document is to create a reference architecture for some of the concepts involved in SSI in such a way that different implementations can be contrasted and compared. This document

attempts to define a set of core normative requirement and also introduce the notion of direct presentation flow to denote the practice of using a mediator to allow the data subject control over the digital credential sharing flow.

Direct presentation flow should be seen as a generalization of the Self-Sovereign Identity concept in the sense that unlike SSI, direct presentation make no assumptions or value judgements about the relative merits of third party data ownership and control. The basic architecture of direct presentation does empower the user with more control than the federated model does but in the SSI architecture the user always has full control over every aspect of data sharing with the RP. This is not necessarily true (eg for legal reasons) in all cases which is why there is a need to desribe the technical underpinnings of direct presentation flows in such a way that the full SSI model can be a special case of a direct presentation architecture.

5. Actors and Entities

5.1. Subject and Presenter

The data subject is the entity that the credential describes, such as an individual, an organization, or even an IoT device. However, the presenter—the actor delivering the credential to the verifier—may not always be the data subject. For example, an administrator might present credentials on behalf of an organization, or a software agent might act as a presenter in automated workflows.

This distinction between the data subject and the presenter allows the architecture to support complex use cases, such as power-of-attorney scenarios or enterprise credentialing systems.

5.2. Presentation Mediator

The presentation mediator (mediator for short) is the core active component of this architecture. It initiates and mediates credential presentations, ensuring compliance with data subject preferences and system policies. For example, it might enforce selective disclosure, revealing only the subject’s date of birth to a verifier while withholding other personal details. The subject controls a presentation mediator.

Unlike a credential store, the presentation mediator is responsible for orchestrating interactions with verifiers, performing cryptographic operations, and generating presentation proofs.

The mediator is used by the subject to communicate with issuers and verifiers. The nature of the control the user has over the mediator varies but minimally the user must be able to initiate the receipt of credentials from an issuer and the generation and transmission of presentation proofs to a verifier.

5.3. Credential Store

The credential store is a passive repository where credentials are securely stored. Its primary function is to provide the presentation mediator with access to the credentials it needs to generate presentation proofs. By separating storage from active mediation, the architecture enhances modularity and allows credential stores to be managed independently from presentation logic.

5.4. Credentials and Presentation Proofs

A digital identity credential (abbreviated as ’credential’ in this document) is an object representing a set of data associated with a subject. The credential MAY contain data that uniquely identify a single subject. A digital identity credential is typically cryptographically bound both to the issuer and to the mediator where it is stored. A presentation proof (abbreviated as ’presentation’ in this document) is a proof that a particular issuer has provided a particular set of credentials to the mediator. A presentation can be verified by at least one verifier. A presentation proof can be based on data present in a single credential or in multiple or even on the result of computations based on a set of credentials. A common example is a presentation proof that a subject is legally permitted to take driving lessons. This is a binary attribute the result of a computation involving knowledge of both the biological age of the subject aswell as legal restrictions that apply to the juristiction where the verifier is operating.

5.5. Issuer and Verifier

An issuer is a set of protcol endpoints that allow a mediator to receive a credential. Credentials issued by the issuer are cryptographically bound to that issuer and to the receiving mediator.

A verfier is a set of protocol endpoints that allow a mediator to send a presentation to a verifier. A verifier is typically a component used to provide an application with data about the subject - for instance in the context of an authentication process.

6. Presentation Flows

Credential presentation flows describe how information from credentials are transmitted from the mediator to the verifier. This architecture focuses on direct presentation flows, but it also accommodates variations such as delegated and assisted presentations.

6.1. Direct Presentation Flow

The basic direct presentation flows looks like this:

```
group issuance
   Subject --> Mediator: <<initiate credential request>>
   activate Mediator
   Issuer <-- Mediator: request credential
   activate Issuer
   Issuer --> Issuer: <<generate credential>>
   return credential
   deactivate Issuer
   deactivate Mediator
   deactivate Subject
end

group verification
   Verifier --> Mediator: request presentation
   activate Mediator
   Mediator --> Presenter: <<prompt to select credential(s)>>
   activate Presenter
   Mediator <-- Presenter: <<select claims from credential(s)>>
   deactivate Presenter
   Mediator --> Mediator: <<generate presentation proof selection>>
   return presentation proof
   deactivate Mediator
end
```

The mediator (acting on behalf of the subject) requests a credential from the issuer. The way this flow is initiated is implementation dependent and in some cases (notably in [OIDC4VCI]) the flow often starts with the subject visiting a web page at the issuer where the subject is first authenticated and then presented with means to launch a credential issuance request using their mediator. These details are left out from the diagram above.

The credential is generated by the issuer presumably based on information the issuer has about the data subject but exactly how the credential is generated is implementation dependent and out of scope for this specification. The claims in the credential typically comes from some source with which the issuer has a trust relationship. The

term "authentic source" is sometimes used when there is a need to distinguish the source of the claims in a credential from the source of the credential which by definition is the issuer.

The mediator receives a credential from the issuer. The credential is bound both to the mediator and to the issuer in such a way that presentation proofs generated from the credential can be used to verify said bindings.

At some later point, the subject wants to use the credentials in their mediator to provide identity data to an application. The application has a verifier (a specific software component responsible for verifying presentation proofs) associated with it. The mediator - often after involving the user in some form of interaction to choose which credential(s) to use and what parts of the credential(s) to include - generates a presentation proof and sends it to the verifier. The precise way this flow is initiated is again implementation dependent and in some cases (notably [OIDC4VP]) the flow starts with the subject visiting the application and hitting a "login" button which directs the users device to launch the mediator to complete the flow. These details are left out of the diagram above.

Upon receipt of the presentation the verifier verifies the issuer and mediator binding (aka holder binding) of the proof and - if the implementation supports revocation - the current validity of the underlying credential(s). If successful the data in the proof is made available to the application.

6.2. Delegated or Assisted Presentation Flow

Delegated flows occur when a third party, such as an enterprise or legal representative, is authorized to present credentials on behalf of the data subject. The presentation mediator ensures that delegation is properly scoped and authorized, preventing misuse.

Assisted flows involve granting limited rights to a third party to act on behalf of the data subject. This may take the form of a secondary credential that grants access to the mediator for the purpose of generating and transmitting presentation proofs on behalf of the data subject.

7. Normative Requirements

7.1. Subject control

The mediator SHOULD provide the subject with the means to control which data from a credential is used in a presentation proof.

The mediator MUST NOT be able to generate a presentation proof without the participation and approval of the data subject.

7.2. Selective Disclosure

A conformant implementation SHOULD identify a format for representing digital credentials that make it possible for the subject to select a subset of the data present in the credential for inclusion in a presentation proof.

Note that there are situations where selective disclosure isn’t applicable, for instance when the data subject is legally complelled to present a credential. Exactly when selective disclosure is available as an option and what aspects of the credential is meaningful to select is an implementation issue and out of scope.

7.3. Issuer Binding

A verifier MUST be able to verifiy the identity of the issuer of the credential from a presentation proof.

7.4. Mediator Binding

The verifier MUST be able to verify that the mediator sending the presentation proof is the same mediator that received the credential from which the presentation proof was derived.

Note that this is often termed ’holder’ binding because the mediator is sometimes called the holder.

7.5. Non-linkability and data minimization

The verifier MUST NOT be able to infer information about data or subjects not present in the presentation. This includes any association between the mediator or subject and other issuers and verifiers not associated with the presentation. In particular, colluding verifiers MUST NOT be able to infer data not present in presentation proofs.

7.6. Revocation

A conformant implementation SHOULD provide a way for an issuer to revoke an issued digital credential in such a way that subsequent attempts by a verifier to verify the authenticity of proofs based on that credential fail.

8. Profiles

Several profiles of this reference architecture exist. We present two below.

8.1. OpenID and SD-JWT

A minimal profile of the direct presentation credential architecture is as follows:

1. Digital credentials are represented as SD-JWT objects [SDJWT]

2. An issuer implements the OP side of [OIDC4VCI]

3. A verifier implements RP side of [OIDC4VP]

4. A mediator implements the RP side of [OIDC4VCI] and the OP side of [OIDC4VP]

A mediator conforming to this profile is essentially an openid connect store-and-proove proxy with a user interface allowing the subject control over selective disclosure.

This minimal profile fulfills several of the requirements in the previous section:

* Selective disclosure is provided by the use of SD-JWT objects to represent credential and presentation objects.

* Issuer binding is provided by a combination of digital signatures on SD-JWTs and OpenID connect authentication between the mediator and issuer.

* Non-linkability is provided by not reusing SD-JWTs from the issuer for multiple presentations. The mediator MAY obtain multiple copies of the same SD-JWT credentials from the mediator at the same time. These can then be used to generate separate presentation objects, never reusing the same SD-JWT credential for separate verifiers.

This profile does not provide any solution for revocation and it leaves the question of how OpenID connect entities (issuers, verifiers and mediator) trust each other. There are also real scalability issues involved in how the digital signature keys are managed but as a minimal profile it illustrates the components necessary to make a direct presentation architecture work.

8.2. Anoncreds

TODO: write about hyperledger & anoncreds

8.3. The EU Digital Identity Wallet

The EU digital identity wallet (EUID wallet) as defined by the architecture reference framework [ARF] is an evolving profile for a direct presentation architecture that includes several aspects of the minimal profile above. Note that the EUID wallet specification is in flux and subect to signifficant change.

9. Security Considerations

One of the main security considerations of a direct presentation credential architecture is how to establish the transactional trust between both the entities (mediators, issuers and verifiers) aswell as the technical trust necessary for the cryptographic binding between the digital credentials and their associated presentation. Digital credentials are sometimes long-lived which also raises the issue of revocation with its associated security requirements.

10. IANA Considerations

None so far

11. References

11.1. Normative References

[OIDC4VCI] Lodderstedt, T., Yatsuda, K., and T. Looker, "OpenID for Verifiable Credential Issuance", n.d., <https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html>.

[OIDC4VP] Terbu, O., Lodderstedt, T., Yatsuda, K., Lemmon, A., and T. Looker, "OpenID for Verifiable Presentations", n.d., <https://openid.net/specs/openid-connect-4-verifiable-presentations-1_0-07.html#name-authors-addresses>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[SDJWT] Fett, D., Yasuda, K., and B. Campbell, "Selective Disclosure for JWTs (SD-JWT)", Work in Progress, Internet-Draft, draft-ietf-oauth-selective-disclosure-jwt-17, 1 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-selective-disclosure-jwt-17>.

11.2. Informative References

[ARF] COM, "The European Digital identity Wallet architecture and Reference framework", n.d., <https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework>.

[OPENIDC] Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, "OpenID Connect Core 1.0", 2014.

[PathToSSI] Allen, C., "The Path to Self-Sovereign Identiy", n.d., <http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html>.

[SAML] Hallam-Baker, P. and E. Maler, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)", OASIS Committee Specification sstc-core, 31 May 2002, <http://www.oasis-open.org/committees/security/docs/cs-sstc-core-01.pdf>.

Acknowledgments

Several people have contributed to this text through discussion. The author especially wishes to acknowledge the following individuals who have helped shape the thinking around trust and identity in general and this topic in particular.

* Pamela Dingle
* Heather Flanagan
* Peter Altman
* Giuseppe DeMarco
* Lucy Lynch
* R.L. ’Bob’ Morgan
* Jeff Hodges

Author’s Address

Leif Johansson
Sunet
Email: leifj@sunet.se