

Verifiable Credentials and Digital Identity Wallets in Research Collaborations (AARC-I101)

Publication Date [2025-12-31]
Authors: Valeria Ardizzone (EGI), Marcus Hardt (ed.) (KIT), Christos Kanellopoulos (GÉANT), Nicolas Liampotis (ed.) (GRNET), Esther Ruiz Ben (DFN), Mischa Sallé (Nikhef)
Document Code: AARC-I101
DOI: 10.5281/zenodo.23163770
Community: Architecture Area

Abstract

This informational document outlines how Verifiable Credentials (VCs) and Digital Identity Wallets—as defined by the W3C Verifiable Credentials model and the European Digital Identity framework (eIDAS 2.0)—relate to AARC-compliant Authentication and Authorisation Infrastructures (AAs). It summarises key standards, examines how AARC services could act as verifiers or issuers of VCs, and highlights trust and interoperability aspects relevant to research collaborations. The document does not define normative behaviour but provides background for future guidelines and pilot work.

© members of the AARC community.

This work is licensed under a Creative Commons Attribution CC-BY 4.0 Licence



This work received funding from the AARC-TREE project (HORIZON-INFRA-2023-DEV-01 call) and other sources.

1 Introduction	3
1.1 Terminology	4
2 Overview of Concepts and Specifications	5
2.1 Core Concepts	5
2.2 Relevant Specifications	6
3 Architectural Considerations for Wallet-based Credential Flows in the AARC Blueprint Architecture	6
3.1 Architectural Patterns	7
3.1.1 Credential Issuers	8
3.1.2 Credential Verifiers	9
3.1.3 Wallets	9
3.2 Architectural Flows	9
3.3 Component-level Requirements for the BPA	11
3.3.1 Issuer Requirements	11
3.3.2 Verifier Requirements	12
3.3.3 Wallet Requirements	13
3.4 Example Use Cases	14
3.4.1 Example Use Case: Institutional Affiliation Credential	14
3.4.2 Example Use Case: Research Collaboration membership and role Credential	14
3.4.3 Example Use Case: Cross Infrastructure Credentials	15
4 Technical Trust Establishment and Metadata Handling	16
4.1 Federated Trust Mechanisms	16
4.2 Verifiable Credential-Based Trust Mechanisms	16
4.3 Bridging the Two	17
4.4 Trusting the Verifier	18
5 Open Technical Challenges	19
5.1 Schema Harmonisation	19
5.2 Granular Disclosure and User Experience	20
5.3 Global Trust Interoperability	20
5.4 Integration with Non-Web Workflows	21
5.5 Token Translation	21
5.6 Attribute Release	21
References	22

1 Introduction

The deployment of federated identity and access management has enabled global research communities to share digital resources securely. The AARC Blueprint Architecture (BPA) defines the reference architecture model adopted by many research infrastructures and e-infrastructures to design and operate interoperable Authentication and Authorisation Infrastructures (AAs).

Recent regulatory and technological developments — most notably eIDAS 2.0 and the European Digital Identity Wallet (EUDI Wallet) — introduce a new paradigm for identity management based on Verifiable Credentials (VCs) and holder-controlled wallets. These mechanisms, based on cryptographically verifiable assertions, allow users to receive, store, and selectively disclose credentials issued by trusted entities. Within Research and Education, these mechanisms are **not intended to replace existing federated Authentication and Authorisation Infrastructures**, but rather to **complement and extend them**. Federated authentication remains the primary means for establishing a high-assurance binding between a user and an identity, while VC-based mechanisms enable the controlled reuse and portability of selected attributes beyond the scope of a single authentication session.

This document explores how these decentralised identity mechanisms could be used within AARC-compliant Authentication and Authorisation Infrastructures (AAs). It analyses how the different Proxy components of the BPA — the intermediary responsible for harmonising trust and attributes — can be interconnected with the new paradigms of wallet-based ecosystems. The wallet itself being a conceptually new and distinct entity, not present in the current BPA, while conceptual similarities can be found in Issuers and Verifiers. The discussion covers:

- Verification paths, where a Proxy or Community AA validates VCs;
- Issuance paths, where Community AAs or Attribute Authorities issue VCs representing research-specific information such as membership, roles, or assurance; and
- Trust alignment, examining how federation frameworks (e.g. OpenID Federation 1.0) could underpin interoperability between these approaches.

The document is informational and does not define implementation requirements. Its goal is to describe emerging models, highlight opportunities for integration, and identify open questions that may guide future AARC guidelines or pilot activities. It builds on related analyses and on work by the REFEDS Verifiable Credentials Working Group, aligning them with the objectives to update the AARC Blueprint Architecture for new technologies.

1.1 Terminology

This informational document uses the terms “verifiable credential” (“VC”), “holder”, “verifier”, “issuer” defined by [\[W3C-VC-DM2\]](#), “Digital Credentials Query Language” (“DCQL”) defined by [\[OID4VP\]](#), “access token”, “authorization endpoint”, “authorization grant”, “authorization

server" ("AS"), "client", "protected resource", "resource owner", "resource server" ("RS"), and "token endpoint" defined by OAuth 2.0 [\[RFC6749\]](#), the terms "claim names" and "claim values" defined by JSON Web Token (JWT) [\[RFC7519\]](#), the terms "token introspection" and "introspection endpoint" defined by OAuth 2.0 Token Introspection [\[RFC7662\]](#), the terms "entity", "entity identifier", "trust anchor", "federation entity", "entity statement", "entity configuration", "subordinate statement", "entity type", "entity type identifier", "leaf entity", "subordinate entity", "federation entity discovery", "trust chain", "trust mark", "federation entity keys" defined by OpenID Federation [\[OID-Fed\]](#), and the terms "Community AAI", "Infrastructure Proxy", "SP-IdP-Proxy" or "Proxy" defined by the AARC Blueprint Architecture 2019 [\[AARC-G045\]](#).

2 Overview of Concepts and Specifications

This section summarises the core concepts of Verifiable Credentials and Verifiable Presentations. Detailed definitions are provided in the W3C Verifiable Credentials Data Model [\[W3C-VC-DM2\]](#) and relevant OpenID specifications [\[OID4VCI\]](#), [\[OID4VP\]](#).

2.1 Core Concepts

A **Verifiable Credential (VC)** is a tamper-evident assertion containing a set of one or more claims about a subject, cryptographically signed by an **Issuer**.

A **Verifiable Presentation (VP)** is a cryptographically verifiable artefact used by a Holder to present one or more Verifiable Credentials, or selected claims from them, to a Verifier, typically for a specific purpose or transaction.

The **Holder** is the entity that receives and controls Verifiable Credentials and initiates Verifiable Presentations. A **Digital Identity Wallet** allows the Holder to manage such credentials and selectively disclose information to a Verifier.

The trust relationship among **Issuer**, **Holder** (via their **Wallet**), and **Verifier** requires trust not only between the issuing and verifying services but also on the Holder's side. The Holder must trust the Issuer to provide accurate attestations and the Verifier to handle the presented data correctly. Equally important, the Holder must trust their Digital Identity Wallet and its underlying device for secure, user-controlled storage and key management.

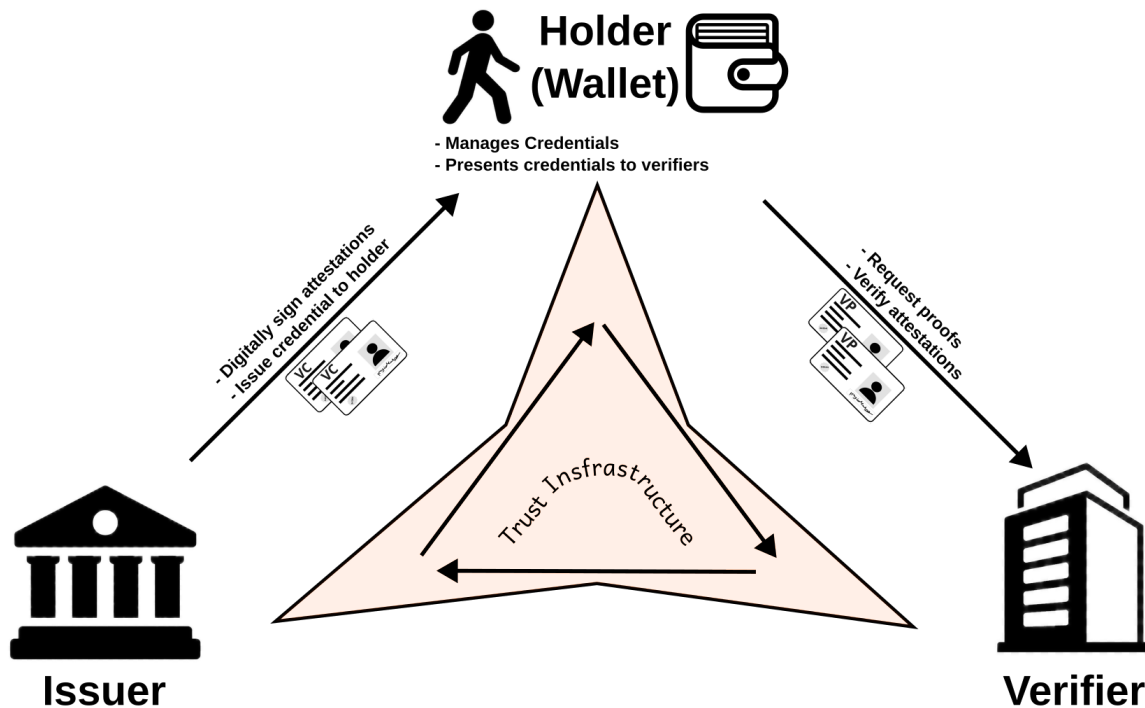


Figure 1: The VC trust triangle: Issuers provide signed VCs to the holder's wallet. The holder can present a selectable subset (VPs) to Verifiers. Trust is established by: (a) Verifiers needing to trust Issuers. (b) Issuers needing to trust Wallets. (c) Holders/Wallets needing to trust Verifiers to handle attributes correctly. The Holders must also trust the Issuer (to provide accurate data) and their own Wallet (device) to securely manage their keys.

To understand how these components operate, it is necessary to distinguish between the static assertions and the dynamic protocols used to share them. Verifiable Credentials and Verifiable Presentations serve distinct architectural purposes:

- Verifiable Credentials (VCs) are cryptographically verifiable artefacts issued by authoritative entities (Issuers) under a defined governance and trust framework. They encode assertions about a subject and can be stored and reused beyond a single interaction.
- Verifiable Presentations (VPs) are runtime interactions through which a Holder selectively discloses one or more credentials to a Verifier, possibly applying privacy-enhancing techniques such as selective disclosure or zero-knowledge proofs.

In other words, VC represents governed data, while VPs represent governed interactions. This distinction is essential when later mapping VC/VP usage onto the AARC Blueprint Architecture.

2.2 Relevant Specifications

Area	Standard / Initiative	Scope
Core Model	W3C VC Data Model 2.0 [W3C-VC-DM2]	Defines credential structure and verification semantics

Protocols	[OID4VCI] / [OID4VP] / SD-JWT [RFC 9901] / OID-Fed-Wallet	Enable issuance and presentation flows using OIDC paradigms
Ecosystem	EUDI Wallet Architecture and Reference Framework [EUDI-ARF]	Specifies architecture and governance for EU wallets

3 Architectural Considerations for Wallet-based Credential Flows in the AARC Blueprint Architecture

This section describes how Verifiable Credentials (VCs), Verifiable Presentations (VPs), and Digital Identity Wallets could be used in AARC-compliant AAls, examining architectural patterns, message flows, and component implications.

VCs introduce new participants (Issuer, Holder, Verifier) and new interaction models based on cryptographic credential issuance and presentation, which can complement the federated Single Sign-On (SSO) and attribute release models commonly used in research collaborations, extending them beyond session-bound assertions.

Trust in VC-based interactions continues to rely on governed trust frameworks, organisational accountability and established federation principles (for details see [Chapter 4](#)).

Several options exist for mapping these new participants and interactions onto existing components (Home IdPs, Community AAls, Infrastructure Proxies, Attribute Authorities, Services). These are discussed in this section, with particular attention to where credential verification and policy enforcement take place.

3.1 Architectural Patterns

This section explores different patterns for the use of Verifiable Credentials and Verifiable Presentations within existing research Authentication and Authorisation Infrastructures. These patterns describe how the roles of Issuer, Holder and Verifier may be mapped onto existing BPA components and Home Identity Providers and Services.

The following principles are common to all patterns described below:

- Verifiable Credentials are issued by entities operating within established Trust Frameworks.
- Verifiable Presentations are created by user-controlled Wallets and represent runtime interactions.
- Infrastructure Proxies or Community AAls act as the Verifiers of Verifiable Presentations in BPA-aligned architectures.
- Community AAls act as Issuers of Verifiable Credentials

- Services consume authorisation context, either directly as Verifiable Presentations or in a form derived from them.

In the baseline scenario, the user authenticates locally (via their wallet) and presents a VP to a component of the research infrastructure — in the AARC context this is typically the Proxy. The Proxy validates the credential, extracts the necessary attributes, and follows established mechanisms to issue a trusted assertion for downstream services. This pattern mirrors flows already familiar from token translation or attribute aggregation.

With Verifiable Credentials, the range of Issuers is broader. Users are expected to collect credentials from multiple Issuers in their Wallets, allowing the Proxy to obtain attestations about a user's identity (e.g. government-issued identity), in addition to attestations about their institutional affiliation and associated roles (e.g. as in the eduPerson schema), as well as identity assurance (e.g. as in the REFEDS Assurance Profile [\[RAE\]](#)).

Figure 2 provides an overview of the architectural patterns discussed in this section by illustrating how the Issuer and Verifier roles can be realised within the AARC Blueprint Architecture. In particular, it shows that AARC-BPA Proxies may assume the role of an Issuer and/or a Verifier. The following subsections examine these patterns in more detail, describing how Verifiable Credential issuance and presentation are mapped onto existing BPA components and interaction models.

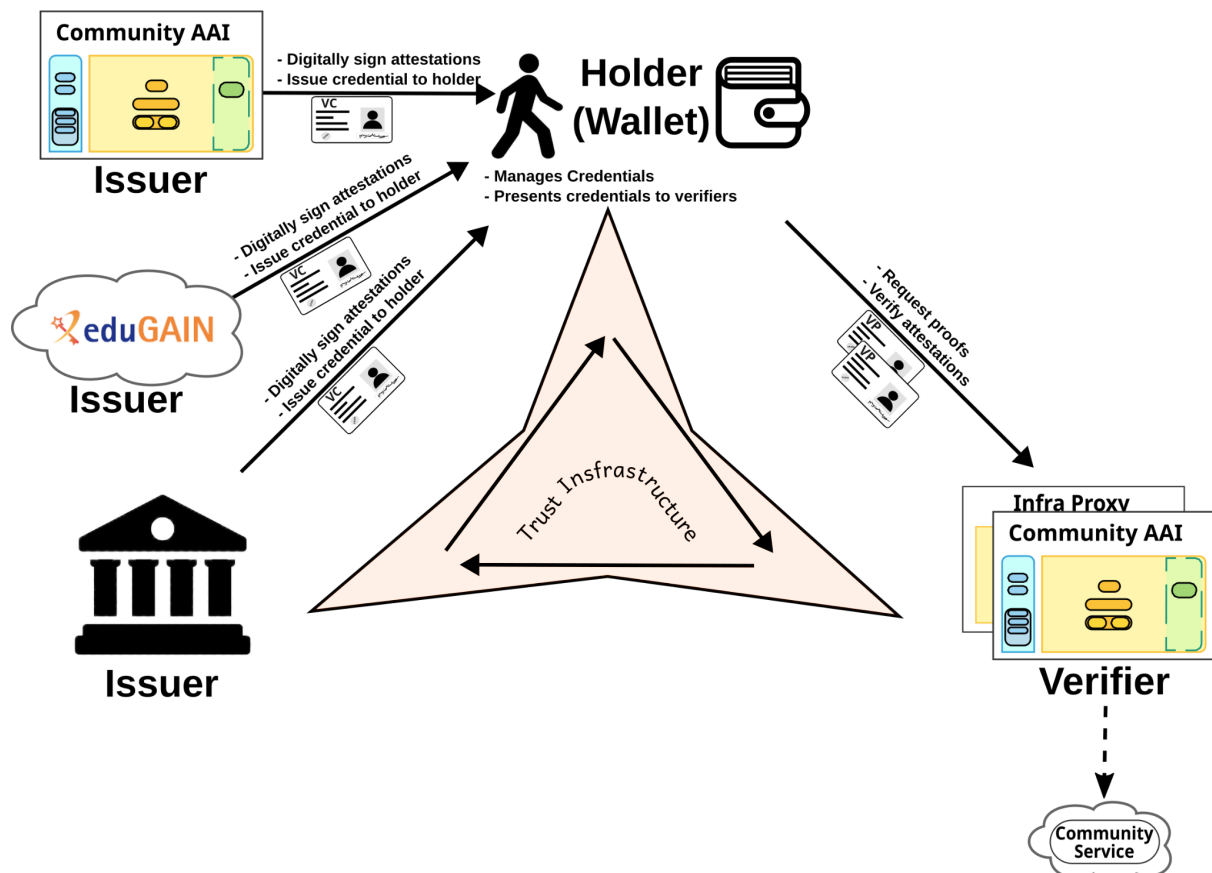


Figure 2: Possible roles of AARC-BPA Proxies in Verifiable Credential-based flows.

3.1.1 Credential Issuers

Following the EUDI concept, the user's Home Organisation (e.g. a university or research institute), as well as the Community AAI can act as Credential Issuers. Multiple issuers may issue VCs to a Holder's Wallet. Issuers we consider in addition to governmental ones, that provide well-assured identity information (i.e. RAF-IAP high), are the institutional home-IdPs, as well as AARC Community AAIs. The Home IdPs are authoritative sources for information (such as [\[SCHAC\]](#) and [\[EDUPERSON\]](#)) and potentially "VC information" (such as diploma, and courses). The AARC Community AAIs provide additional VCs such as collaboration-specific attributes (entitlements, memberships, roles).

This approach preserves the BPA principle that services remain isolated from identity and trust complexity, while enabling Home-issued credentials to be reused across collaborations. The combination with government-issued VCs promises a straightforward support for the Identity-Assurance of the REFEDS Assurance Framework [\[RAF\]](#).

A Community AAI that issues verifiable credentials to the user's wallet may encode:

- project affiliation, collaboration group membership and role information,
- resource Capabilities,
- training completion, or
- other structured collaboration attributes.

The issuance path allows users to carry research-specific identity information into other contexts or reuse them across infrastructures.

3.1.2 Credential Verifiers

As in other BPA-aligned patterns, the preferred integration model positions the Infrastructure Proxy as the Verifier of Verifiable Presentations, ensuring consistent policy evaluation across services. In this scenario, Verifiable Presentations are created by the user's Wallet at runtime, based on credentials issued by the Home Organisation. These presentations are not consumed directly by services, but are instead presented to a Community AAI or an Infrastructure Proxy, which assumes the role of Verifier. The Proxy validates the presentation, evaluates it against federation and community policy, and translates the outcome into standard authorisation assertions consumable by downstream services.

Direct verification of Verifiable Presentations by services remains possible but introduces additional complexity and governance responsibilities. Such patterns should therefore be considered advanced and context-specific. When services assume the Verifier role, they must also take responsibility for trust framework alignment, policy enforcement and auditability.

3.1.3 Wallets

As shown in Fig. 2, no AARC Blueprint Architecture component directly maps to the Digital Identity Wallet role. The wallet is a conceptually distinct entity, designed to operate under

direct user control — typically on personal devices such as smartphones or browsers — and decoupled from issuers and verifiers.

Although some wallet functions overlap with existing Community AAs (e.g. attribute aggregation) and others could be added (e.g. selective disclosure), full wallet implementations require secure key management, cross-platform compatibility, sophisticated user interfaces, and robust interaction with multiple issuers and verifiers.

Community AAs and Infrastructure Proxies remain well-suited to act as Issuers or Verifiers, issuing research-specific credentials to user wallets or validating presentations from them. Both issuance and verification paths can coexist within the same architecture.

3.2 Architectural Flows

The SP-IdP-Proxy (or Proxy) defined in AARC-BPA-2019 ([AARC-G045](#)) can operate as an Issuer or Verifier for Wallet-presented credentials. This aligns well with its existing architectural responsibilities: aggregating attributes, harmonising trust, and providing a consistent abstraction to downstream services.

A typical sequence for the patterns discussed in [Section 3.1](#) involves the following steps:

1. The Issuer issues a VC to a Holder, who stores it in the wallet. (3.1.2)
2. The Proxy (acting as a Verifier) requests the Holder to identify / provide a set of claims.
3. The Holder retrieves a VP from their wallet.
4. The Proxy verifies the credential's signature, issuer metadata, and validity.
5. The Proxy normalises the attributes (e.g., mapping JSON-LD or SD-JWT claims to internal attribute schemas).
6. The Proxy (acting as an OIDC OP or OAuth2.0 AS) issues a new JWT/OIDC token trusted by services.
7. Downstream services consume the Proxy-issued token without needing to understand the credential format or the issuer ecosystem.

This Proxy-mediated trust delegation allows relying parties to trust one entity (the Proxy) rather than every external VC issuer.

Credential Issuance

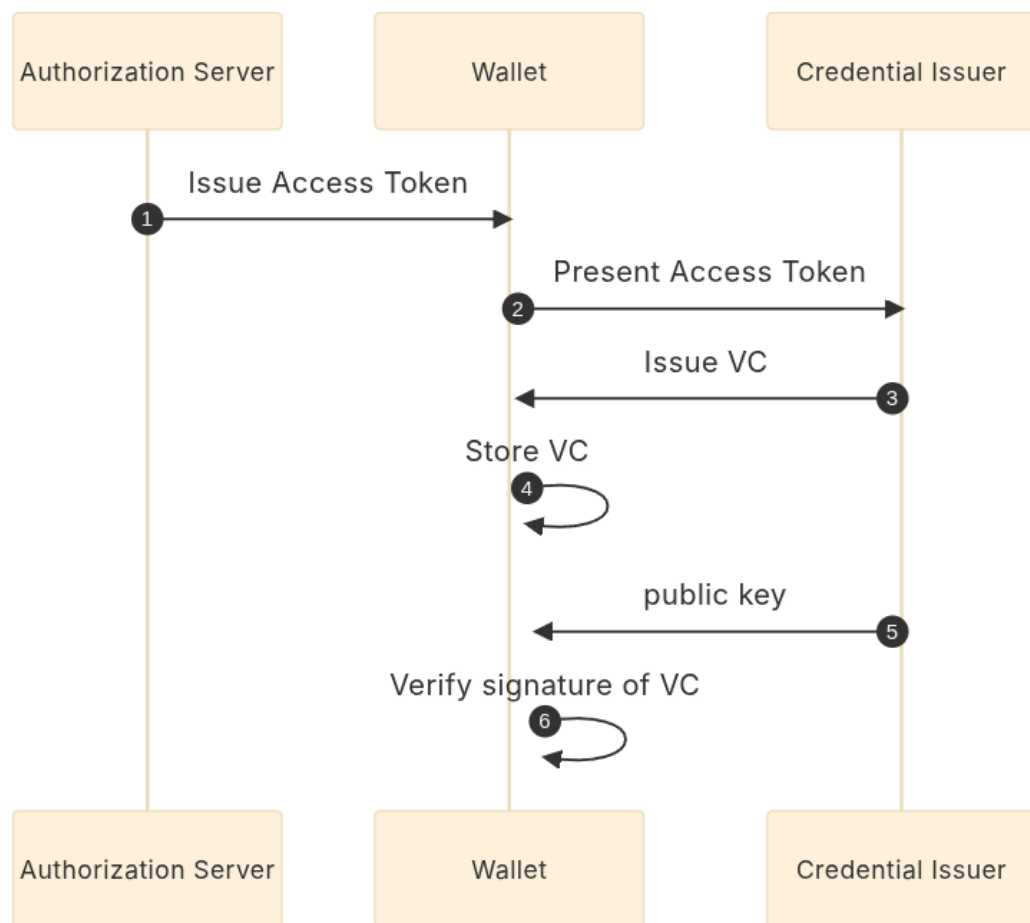


Figure 3: Sequence for obtaining a Verifiable Credential

Credential Verification

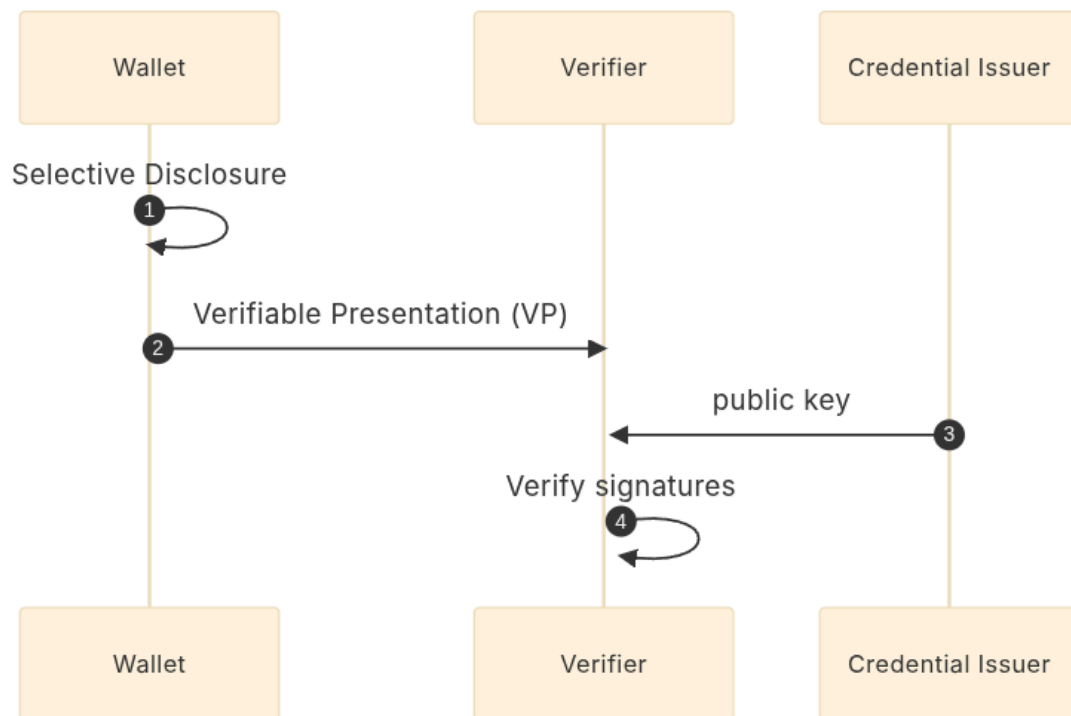


Figure 4: Sequence for Selective Disclosure and Credential Verification

3.3 Component-level Requirements for the BPA

The following subsections outline the specific requirements and differences introduced when AARC Blueprint Architecture components assume the roles of Credential Issuer, Verifier, or interact with Wallets.

3.3.1 Issuer Requirements

Community AAls (and Identity Proxies) can extend their existing OpenID Provider (OP) functionality to support Verifiable Credential issuance in accordance with [[OpenID4VCI](#)]. A fundamental distinction in this model is the provision of persistent, portable credentials, as opposed to the session-based tokens of standard OIDC.

To implement this capability, the following requirements apply:

- **Multiple Format & Schema Support** - Must handle JWT-VC, JSON-LD, SD-JWT formats with structured credential templates vs only
- **JWT**
 - Support for W3C VC Data Model 2.0 and JSON-LD contexts
 - Predefined credential schemas for specific use cases (education, employment, etc.)
 - Template-based credential generation with standardized claim structures

- **Deferred Issuance** - Async credential generation vs immediate token issuance
 - Support for approval workflows and manual verification processes
 - Polling mechanisms for credential readiness status
 - Notification systems for credential availability
- **Revocation Management** - Maintain revocation lists vs simple token expiration
 - Real-time revocation list publication and updates
 - Integration with revocation registries and status checking
 - Credential lifecycle management beyond simple expiration
- **Batch Operations & Holder Binding** - Issue multiple credentials simultaneously and cryptographically bind to holders vs single bearer token per flow
 - Bulk credential issuance for organizational use cases
 - Proof-of-possession verification to bind credentials to holder keys
 - Support for credential families and related credential sets
- **New/Modified Endpoints:**
 - `/well-known/credential-issuer` - Credential issuer metadata discovery
 - `/credential` - The Credential Endpoint issues one or more Credentials of the same Credential Configuration and Credential Dataset upon presentation of a valid Access Token
 - `/deferred_credential` - This endpoint is used to issue one or more Credentials previously requested at the Credential Endpoint in cases where the Credential Issuer was not able to immediately issue this Credential
 - `/nonce` - This endpoint allows a Client to acquire a fresh `c_nonce` value
 - `/notification` - This endpoint is used by the Wallet to notify the Credential Issuer of certain events for issued Credentials
 - Enhanced `/authorization` and `/token` endpoints for credential-specific flows

3.3.2 Verifier Requirements

A Verifier is comparable to an OpenID Connect Relying Party (OIDC RP), but differs in several aspects:

- **Cryptographic Verification** - Must verify signatures vs trusting OP assertions
 - Validate VP signature instead of ID token signature
 - Direct signature validation using the issuer's published keys
 - Certificate chain validation and trust path verification
 - Complex credential combinations
- **Selective Disclosure** - Handle partial credential reveals vs full claim sets
 - Replace scope parameters with a Digital Credentials Query Language (DCQL) query
 - Processing of redacted or hidden claims in SD-JWT format
 - Zero-knowledge proof verification for privacy-preserving presentations
- **Presentation Requests** - Specify exactly what credentials are needed vs generic scopes
 - Support for alternative credential types and flexible acceptance criteria
 - Conditional logic for credential combinations and requirements

- **Offline Capability & Multiple Issuers** - Verify credentials without real-time OP contact
 - Cached issuer metadata and key material for offline verification
 - Trust registry integration for dynamic issuer acceptance
 - Deferred Responses
- **Nonce & Replay Protection** - Prevent presentation replay attacks vs session-based OIDC security
 - Generate and validate unique nonces for each presentation request
 - Timestamp validation to prevent stale credential presentations
 - Challenge-response mechanisms for fresh proof generation
- **New/Modified Endpoints:**
 - /.well-known/verifier - Verifier metadata discovery
 - /request_uri endpoint - Information about Verifier capabilities
 - /response_uri endpoint - Allow the Wallet to send the Authorization Response via an HTTP POST request
 - Enhanced /authorization endpoint supporting response_type=vp_token

3.3.3 Wallet Requirements

The Digital Identity Wallet acts as a user-controlled client, in contrast to server-side identity components. While the AARC Blueprint Architecture does not define wallet internals, the architectural patterns described in this document assume that the wallet provides the following capabilities:

- **User Sovereignty** - User controls their credentials vs OP-managed identity
 - User decides which credentials to accept and store
 - Complete control over credential sharing and presentation
 - No central authority can revoke, access, or modify user's credential collection
- **Credential Storage** - Persistent, encrypted credential vault
 - Long-term encrypted storage with backup and recovery mechanisms
 - Support for multiple credential formats and versions
 - Cross-device synchronisation while maintaining security
- **Selective Sharing** - User chooses what to disclose vs OP-determined claim release
 - Granular control over which claims to reveal in each presentation
 - Support for conditional disclosure based on verifier requirements
 - Privacy-preserving presentation with minimal data exposure
- **Key Management & Direct Interaction** - User owns cryptographic keys and can interact directly with other wallets vs OP-managed keys and mediated flows
 - Hardware-backed key generation and storage where available
 - Direct peer-to-peer credential exchange without intermediaries
 - Support for multiple key types and cryptographic protocols
- **New Endpoints:**
 - /.well-known/wallet-config - Wallet metadata discovery
 - /wallet/credentials - Credential management API
 - /wallet/presentations - Presentation handling endpoint
 - /wallet/attestations - Wallet attestation services endpoint

3.4 Example Use Cases

This subsection provides example use cases for the architectural patterns described in [Section 3.1](#).

3.4.1 Example Use Case: Institutional Affiliation Credential

A researcher affiliated with a university participates in multiple research collaborations. While federated authentication via the Home Identity Provider establishes the user's identity at login time, services frequently require proof of current institutional affiliation beyond a single session.

The actors of this use case are:

- **Home Organisation** (e.g. university) acts as the Credential Issuer, attesting institutional affiliation.
- **User** (e.g. a researcher) holds credentials in a personal Digital Identity Wallet.
- **Community AAI/ Infrastructure Proxy** acts as the Verifier of the Verifiable Presentations.
- **Protected Services** rely on Proxy-issued authorisation context.

Credential Issuance:

Following a successful authentication at the Home IdP, the university issues a verifiable credential stating that the user is an active member of the institution, including attributes such as affiliation type and validity period. The credential is cryptographically signed and stored in the user's Wallet.

Runtime Access Using Verifiable Presentation:

When the researcher later accesses a service, the Infrastructure Proxy requests proof of institutional affiliation. The user's Wallet creates a verifiable Presentation derived from the affiliation verifiable credential, disclosing **only the attributes required by policy**.

The Infrastructure Proxy verifies the presentation and enforces access rules, issuing an authorisation token to the service. **The service itself does not process the Verifiable Presentation.**

3.4.2 Example Use Case: Research Collaboration membership and role Credential

A scientific collaboration manages access to shared datasets and computing resources based on membership and assigned roles. These attributes are dynamic and governed by community-specific policies.

The actors of this use case are:

- **Community AAI** acts as a Credential Issuer, attesting collaboration membership and roles.
- **User** (e.g. a researcher) holds credentials in a Digital Identity Wallet.
- **Infrastructure Proxy** acts as the Verifier of Verifiable Presentations.
- **Community Services** consume the authorisation context.

Credential Issuance:

After onboarding and approval by the collaboration governance body, the Community AAI issues a **Verifiable Credential** to the user's Wallet, stating the user's membership and role within the collaboration, with a defined validity period.

Runtime Access Using Verifiable Presentation:

When accessing a protected dataset or service, the Infrastructure Proxy requests a Verifiable Presentation proving the required role. The user's Wallet constructs the presentation, disclosing only the relevant role information. The Proxy verifies the presentation, evaluates community policies and grants access accordingly.

3.4.3 Example Use Case: Cross Infrastructure Credentials

Research Infrastructure A operates a set of services that require users to complete mandatory security training and formally acknowledge acceptable use and data protection policies before access is granted. Infrastructure B operates security trainings and issues such credentials. Infrastructure A happens to find that B's credentials are exactly what they need.

The actors of this use case are:

- **Infrastructure B Provider** (e.g. a Community AAI) acts as a credential Issuer, with well defined training requirements and policy compliance criteria.
- **User** (e.g. a researcher) holds several credentials, including security course credentials of B, in a personal Digital Identity Wallet.
- **Infrastructure A Proxy** made the business decision to trust B's security course credentials. It acts as the Verifier of Verifiable Presentations and uses B's security course credentials to enforce access policies. Provider B is not involved in this step.
- **Protected Services** consume authorisation context derived from Proxy-enforced decisions.

Credential Issuance:

After authenticating via federated Single Sign-On, the user completes the required online training and explicitly accepts the infrastructure B's acceptable use and data protection policies. Upon successful completion, the infrastructure B issues a **Verifiable Credential** attesting that the user:

- has completed the mandatory training,
- has accepted the applicable policies,
- is eligible to access the infrastructure services for a defined validity period.

The credential is cryptographically signed by the infrastructure B provider and delivered to the user's Digital Identity Wallet.

Runtime Access Using Verifiable Presentation:

At a later time, when the user attempts to access a protected service, the Infrastructure A Proxy requests proof of training and policy compliance. The user's Wallet constructs a **Verifiable Presentation** derived from the previously issued credential, disclosing only the

attributes required for the access decision. The Infrastructure A Proxy verifies the presentation, checks its validity and expiration, and evaluates it against current infrastructure A's policies. If the requirements are satisfied, the Proxy grants access by issuing a standard authorisation assertion or token to the service.

Importantly, the service itself does not process the Verifiable Presentation and remains unaware of the underlying credential mechanics.

4 Technical Trust Establishment and Metadata Handling

This section outlines how trust can be established when integrating wallet-based credential flows with AARC-compliant federated architectures.

4.1 Federated Trust Mechanisms

Trust in AARC-compliant federated architectures has grown to a complex, hierarchical, distributed, multi-national, fully meshed network of trust relationships. The SAML based identity federation eduGAIN is migrating towards the upcoming OpenID Federation [\[OID-Fed\]](#), because of scalability and complexity.

OID-Fed offers multiple mechanisms for establishing trust between two entities (e.g. an Issuer and a Verifier, an IdP and an SP, or an OP and an RP).

Trust Anchor

Trust Anchors are entities that are trusted, based on an external business decision. To evaluate whether a Proxy can be trusted, the trust chain is evaluated until it reaches a trusted root of the chain, the Trust Anchor. This structure allows modelling the hierarchical structure of today's eduGAIN.

Trust Marks

In addition, Trust Marks provide a separate structure for Trust Mark Issuers to make statements about entities. This separate structure allows modelling the different policies supported by federation members.

The [\[AARC-G100\]](#) guidelines for establishing trust between AARC-compliant AAI services using OpenID Federation, for example, considers two profiles. In a basic profile, two entities are supposed to trust each other, if they can establish a path to a Trust Anchor. In a more advanced profile, the OID-Fed feature of Trust Marks is used to assert specific policy support (e.g. RAF, Sirtfi, DPCoCo) to entities. This allows establishing a well informed trust decision in the complex domain of federated identity management.

4.2 Verifiable Credential-Based Trust Mechanisms

Digital Identity Wallets operate on a holder-centric trust model that differs from the operator-centric, server-to-server model of traditional federated AAls. In the wallet ecosystem, the Verifier establishes trust directly with the Issuer of a Verifiable Credential rather than with an intermediate authentication provider.

Issuer Trust

Trust in an Issuer is typically anchored in pre-published trust sources rather than dynamic, bilateral metadata exchange. The main mechanisms are:

- the EU/EEA Trusted Lists [\[EU-Trust-List\]](#), which list accredited Qualified Trust Service Providers;
- decentralised issuer registries or DID resolution methods used in non-regulated deployments.

Cryptographic Verification

The Verifier validates the digital signature (or other cryptographic proof) on the Verifiable Credential using the Issuer's public key, retrieved from a trusted source (e.g. a trust list entry, a DID document, or an issuer metadata service such as an OpenID Federation entity configuration). This establishes authenticity and integrity independently of the transport protocol.

Credential Status

Credential validity beyond the cryptographic proof is determined via status mechanisms such as Bitstring Status List v1.0 [\[W3C-BSL\]](#) or issuer-specific revocation registries. These mechanisms are deliberately decoupled from the Issuer's immediate operational availability and, in many cases, support fully offline verification by the Verifier.

A key challenge of the current European digital identity (EUDI) for research collaborations is that the EU/EEA Trusted Lists [\[EU-Trust-List\]](#) are limited to nationally accredited providers and do not currently include the numerous Community AAls/Collaboration Platforms or Infrastructure Proxies that legitimately issue research-specific credentials (membership, roles, entitlements, etc). This motivates the hybrid trust models described in [Section 4.3](#).

4.3 Bridging the Two

To enable interoperability between AARC-compliant AAls and the Wallet ecosystem, OpenID Federation 1.0 [\[OID-Fed\]](#) can serve as the common trust layer, as proposed in the *OpenID Federation for Identity Wallet Architectures* specification [\[OID-Fed-Wallet\]](#). The development of the OpenID Federation Wallet Architectures 1.0 specification (Draft 03) was specifically guided by community feedback to prevent conflicts with existing OpenID wallet standards. The feedback led to the exclusion of any metadata parameter extensions in this draft that would affect other specifications, such as [\[OpenID4VP\]](#) and [\[OpenID4VCI\]](#), ensuring that [\[OID-Fed-Wallet\]](#) functions purely as a structural trust layer and not as a competing extension.

Approaches for bridging EU/EEA Trusted Lists with wallet-based architectures include direct consumption of Trusted List entries or using OpenID Federation as an intermediate trust layer. In practice, the latter approach does not require entities listed on EU/EEA Trusted Lists to directly implement OpenID Federation. A Trust Authority or federation operator may instead consume EU/EEA Trusted Lists and expose corresponding OpenID Federation metadata. In this model, EU/EEA Trusted Lists remain the regulatory source of trust, while

OpenID Federation provides the technical mechanism for trust discovery and metadata distribution.

In [\[OID-Fed-Wallet\]](#), the "Issuer" and "Verifier" roles of the Wallet ecosystem are treated as Federation Entities:

- **Discovering Trusted Issuers:** A Verifier (e.g. a downstream Proxy or service) can determine if a VC Issuer is legitimate by resolving the Issuer's **Entity Configuration**. If a valid **Trust Chain** exists from the Issuer to a recognised Trust Authority, the Issuer is trusted.
- **Metadata Distribution:** The OID-Fed Entity Configuration is used to distribute the specific metadata required for Wallet interactions, such as OID4VCI (Issuance) and OID4VP (Presentation) configurations, alongside standard OIDC metadata.
- **Policy Alignment:** The Fine-grained Trust Profile defined in [\[AARC-G100\]](#) leverages OID-Fed Trust Marks to express compliance and capabilities for AARC-compliant entities in a federation. Similar Trust Marks could be applied to VC Issuers. For example, a "Research Collaboration Issuer" Trust Mark could signal to a wallet or verifier that an entity is authorised to issue credentials for a specific scientific collaboration.

This approach allows Trust Authorities, as defined in [\[AARC-G100\]](#), to extend their scope to include Wallet Issuers and Verifiers without deploying separate infrastructure.

4.4 Trusting the Verifier

Digital Identity Wallets must ensure that attributes are only released to authorised Verifiers. In wallet ecosystems this is known as Verifier Attestation, and it is a critical user-protection requirement: the Holder must be able to understand who is requesting data and whether that entity is trustworthy.

AARC-G100 and OpenID Federation for Identity Wallets [\[OID-Fed-Wallet\]](#) provide mechanisms that allow Wallet implementations to satisfy these requirements while maintaining compatibility with federated AAls.

Verifier Identity

When an AARC Proxy acts as a Verifier, it identifies itself using its Federation Entity Identifier, normally its HTTPS URL. This Entity ID must correspond to a published Entity Configuration, signed by a Trust Authority as defined in AARC-G100. Wallets or wallet-intermediary services can use this metadata to determine whether the Verifier is legitimate.

Discovering Verifier Metadata

Wallet ecosystems can obtain Verifier metadata in several ways:

- Client Registration using [\[OID-Fed\]](#);
- Pre-registration in regulated ecosystems (e.g. [\[EUDI-ARF\]](#) where Verifiers must be approved by a national authority);

- DID-based resolution in decentralised architectures;
- Direct metadata resolution via OpenID Federation trust chains ([OID-Fed-Wallet](#)).

Automatic Registration as defined in [\[OID-Fed\]](#) is one mechanism for Verifier metadata discovery. Wallet ecosystems may rely on alternative approaches depending on their trust framework. In practice, regulated ecosystems (such as the EUDI Wallet) tend to rely on pre-approved Verifier lists, whereas OpenID Federation enables dynamic trust discovery through cryptographically verifiable metadata.

Trust Marks

Using AARC-G100 and OpenID Federation, the Verifier can publish Trust Marks, for example:

- “Data Protection Code of Conduct”,
- “SIRTFI Security Framework”,
- or a community-defined mark indicating that the Verifier is authorised to request specific collaboration attributes.

Wallets may display these trust marks to the Holder during the consent step, providing meaningful assurance that:

- the Verifier is accredited by a recognised authority, and
- the requested attributes will be handled according to community policies.

Implications for Research Infrastructures

This approach enables a Proxy to act as a trusted Verifier without requiring bespoke wallet-side configuration. The Proxy participates in a federation trust fabric defined in [\[AARC-G100\]](#), and the wallet retrieves its metadata and trust marks through the standard OpenID Federation mechanisms defined in [\[OID-Fed-Wallet\]](#). This provides a unified trust model where the wallet can authorise attribute release based on the same cryptographically verifiable metadata that enables interoperability between AARC-compliant services.

5 Open Technical Challenges

5.1 Schema Harmonisation

AARC deployments today rely on the AARC Attribute Profile [\[AARC-G056\]](#) and associated schemas such as eduPerson, SCHAC and voPerson. VCs, however, permit more flexible claim structures, and without alignment the same attribute (e.g. affiliation or entitlements) may be encoded inconsistently across issuers. Ongoing work in the REFEDS VC Subcommittee is to explore mappings to represent R&E attributes in VC format [\[REFEDS-VC-WP\]](#). Key challenges include:

- **Credential Representation:** Standardising how identifiers (persistent/pairwise, global), identity/authentication assurance, affiliation, and entitlements are represented in credential form.
- **Type Registration:** Establishing collision-resistant naming for credential types.

- **Identity Binding and Recovery:** A specific architectural challenge lies in the mapping between upstream "Core Identities" (such as the EUDI Wallet PID or an Institutional ID such as subject-id or eduPersonPrincipalName) and downstream collaboration identifiers. It is unclear whether a Collaboration VC must explicitly encode the upstream persistent identifier (e.g. voPersonID derived from EUDI) to ensure traceability.
 - The Linking Problem: If the link between the eGov/Institutional ID and the Collaboration ID is maintained only via the user's wallet (i.e. "wallet binding"), the loss of the wallet (private key loss) breaks the chain of trust.
 - The Recovery Problem: Guidelines are needed on whether collaboration VCs should embed the upstream personID to enable automated re-issuance and account recovery, or if privacy requirements dictate a strict separation that forces manual re-onboarding upon wallet loss.

5.2 Granular Disclosure and User Experience

Researchers often hold large sets of entitlements (collaboration membership, project roles, dataset access), creating both technical and User Experience (UX) challenges in Wallet-based flows [[W3C-VC-UC](#)].

- **Technical aspect:** Standard VC encoding may treat lists as indivisible. Mechanisms such as Selective Disclosure JWT (SD-JWT) allow claim-level disclosure, but applying selective disclosure to dozens or hundreds of entitlement values requires careful indexing, credential structuring, and caching strategies.
- **Usability aspect:** Presenting a large flat list of entitlements to the user leads to cognitive overload. Wallet UX should avoid situations where researchers must manually select from numerous VO entitlements.
- **Accessibility aspects:** Accessibility aspects include challenges related to the equal perception, understanding and management of verifiable credentials, their integration in the digital wallets, their functionality connected to selective disclosure and their privacy and security. For example, to avoid exclusion of persons with visual impairments, auditive alternatives for navigation/selection VCs should be included. Compatibility of assistive technologies already provided by the devices with the VCs might help to avoid accessibility inequalities. [[W3C-WCAG3.0](#)].

Mechanisms such as Digital Credential Query Language (DCQL) in [[OID4VP](#)] could enable the Verifier to request only relevant entitlements, reducing disclosure and avoiding user decision fatigue.

5.3 Global Trust Interoperability

Research is inherently international, yet Digital Identity Wallet ecosystems are evolving along jurisdiction-specific trust frameworks. The EUDI Wallet under eIDAS 2.0 is a European model, but many countries operate or are developing their own government eID systems and digital wallets. A key challenge is enabling global participation in R&E collaborations, regardless of whether a user holds:

- an EUDI Wallet backed by the EU Trust List,
- a national government eID wallet outside Europe, or
- a research-issued (R&E) wallet using community trust roots.

To avoid fragmentation, research infrastructures need mechanisms that allow Verifiers to trust both government trust anchors and R&E trust frameworks.

5.4 Integration with Non-Web Workflows

Research workloads frequently operate outside a browser — HPC batch submission, CLI tools, SSH, Jupyter notebooks, delegation, API access. Current Wallet interactions assume a web-mediated UX, which is not always available in such environments. It is therefore essential to ensure that Wallet support is not limited to web-login use cases.

Pattern candidates include:

- OAuth 2.0 Device Flow adapted to Wallet-based attestation
- Token brokerage/robot credentials for long-running jobs
- VC-backed SSH certificates or OIDC/Oauth tokens minted post-verification

5.5 Token Translation

While Section 3 describes the mechanism where a Proxy verifies a VC and issues an Access Token, the open challenge lies in defining a standardised translation profile that ensures consistent behaviour across collaborations and infrastructures. Even if attribute schemas are harmonised (see [Section 5.1](#)), most services today consume OIDC tokens rather than VCs directly — and many are expected to continue doing so during the transition to wallet-native architectures. This raises a number of design questions that remain open for community alignment, including:

- How should token lifetime relate to VC expiry or revocation status?
- How should identity and authentication assurance from a VC propagate into AAI session context?
- How is replay prevented, and should Access Tokens be cryptographically bound to a Verifiable Presentation event?
- How should refresh or renewal occur without repeatedly invoking the Wallet?

5.6 Attribute Release

Privacy is a major concern for the acceptance of a system. The wallet architecture allows wallet holders to selectively disclose attributes to verifiers. Without appropriate measures, verifiers can potentially ask users for more attributes than necessary to provide the service. In this regard, it is currently not clear

- how verifiers can be restricted to a set of allowed attributes, e.g. using the situation specific DCQL Queries that a verifier can use [OID-Fed-Wallet, Sec. 6.3.1].
 - Potentially, when registering to the Trust-Fabric of OID Fed, a list of allowed DCQL Statements may be registered.

- Potentially, allowed DCQL Statement-registration scales better with OID-Fed Trust Marks.
- how issuers can categorise different attributes to be allowed for different types of verifiers or different situations in which a verifier acts.

References

- [AARC-G045]** AARC Community members; Applnt members; Nicolas Liampotis (ed.), “AARC Blueprint Architecture 2019”, DOI: 10.5281/zenodo.3672784, November 2019, <<https://aarc-community.org/guidelines/aarc-g045/>>.
- [AARC-G056]** AARC Community members; Nicolas Liampotis (ed.), “AARC Profile for Expressing Identity Attributes”, Draft, <<https://aarc-community.org/guidelines/aarc-g056/>>.
- [AARC-G100]** AARC Community members; Diana Gudu (ed.), “Guidelines for Establishing Trust between AARC-compliant AAI services using OpenID Federation”, Final Draft, <<https://aarc-community.org/guidelines/aarc-g100/>>.
- [EU-Trust-Lists]** EU/EEA Trusted List Browser: Trusted lists of qualified trust service providers in accordance with the EUDI Framework. <<https://eidas.ec.europa.eu/efda/trust-services/browse/eidas/tls>>.
- [EUDI-ARF]** European Commission, “European Digital Identity Wallet Architecture and Reference Framework”, Version 2.7.3, November 2025, <<https://eudi.dev/2.7.3/>>.
- [OID4VCI]** T. Lodderstedt, K. Yasuda, T. Looker, P. Bastian, “OpenID for Verifiable Credential Issuance 1.0”, OpenID Foundation Final Specification, 16 September 2025, <https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html>.
- [OID4VP]** O. Terbu, T. Lodderstedt, K. Yasuda, D. Fett, and J. Heenan (eds.), “OpenID for Verifiable Presentations 1.0”, OpenID Foundation Final Specification, 9 July 2025, <https://openid.net/specs/openid-4-verifiable-presentations-1_0.html>.
- [OID-Fed]** Hedberg, R., Ed., Jones, M.B., Solberg, A.Å., Bradley, J., De Marco, G., and Dzhuvinov, V., “OpenID Federation 1.0”, Draft 46, December 2025, <https://openid.net/specs/openid-federation-1_0.html>.
- [OID-Fed-Wallet]** De Marco, G., Hedberg, R., Jones, M.B., and J. Bradley, “OpenID Federation Wallet Architectures 1.0”, OpenID Foundation Implementer’s Draft 03, October 2024, <https://openid.net/specs/openid-federation-wallet-1_0.html>.
- [RAF]** “REFEDS Assurance Framework version 2.0”, December 2023, <<https://refeds.org/assurance>>.
- [REFEDS-VC-WP]** “White Paper on implementation of identifiers, attributes and claims for user identity in Research and Education”, REFEDS VC Subcommittee

Draft, December 2025,

<https://docs.google.com/document/d/1b-Mlet3Lq7qKLEf1BnHJ4nL1fq-vMe7fzpXyrg2wp08/edit>.

- [RFC6749]** Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <http://www.rfc-editor.org/info/rfc6749>.
- [RFC7515]** Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.
- [RFC7519]** Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/rfc/rfc7519>.
- [RFC7662]** Richer, J., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015, <https://www.rfc-editor.org/rfc/rfc7662>.
- [RFC8414]** Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Authorization Server Metadata", RFC 8414, DOI 10.17487/RFC8414, June 2018, <https://www.rfc-editor.org/info/rfc8414>.
- [RFC8693]** Jones, M., Nadalin, A., Campbell, B., Ed., Bradley, J., and C. Mortimore, "OAuth 2.0 Token Exchange", RFC 8693, DOI 10.17487/RFC8693, January 2020, <https://www.rfc-editor.org/info/rfc8693>.
- [RFC8707]** Campbell, B., Bradley, J., and H. Tschofenig, "Resource Indicators for OAuth 2.0", RFC 8707, DOI 10.17487/RFC8707, February 2020, <https://www.rfc-editor.org/info/rfc8707>.
- [RFC8725]** Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", BCP 225, RFC 8725, DOI 10.17487/RFC8725, February 2020, <https://www.rfc-editor.org/info/rfc8725>.
- [RFC9068]** Bertocci, V., "JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens", RFC 9068, DOI 10.17487/RFC9068, October 2021, <https://www.rfc-editor.org/info/rfc9068>.
- [RFC9901]** Fett, D., Yasuda, K., and B. Campbell, "Selective Disclosure for JSON Web Tokens", RFC 9901, DOI 10.17487/RFC9901, November 2025, <https://www.rfc-editor.org/info/rfc9901>.
- [W3C-BSL]** M. Sporny and I. Herman (eds.), "Bitstring Status List v1.0", W3C Recommendation, 15 May 2025, <https://www.w3.org/TR/vc-bitstring-status-list/>.
- [W3C-VC-UC]** W3C Working Group, Note, 24 September 2019 "Verifiable Credentials Use Cases", <https://www.w3.org/TR/vc-use-cases/#user-tasks>
- [W3C-VC-DM2]** M. Sporny, D. Longley, M. Sabadello, et al. (eds.), "Verifiable Credentials Data Model v2.0", W3C Recommendation, 15 May 2025, <https://www.w3.org/TR/vc-data-model-2.0/>.
- [W3C-WCAG3.0]** R. B. Montgomery, C. Adams, A. Campbell, K. White, J. Spellman, F. Storr (eds.), "W3C Accessibility Guidelines (WCAG) 3.0", <https://www.w3.org/TR/wcag-3.0/>.