Login System for OpenID Connect with Verifiable Credentials

The 22nd International Symposium on Network Computing and Applications (NCA 2024)

Dario Castellano[12], Roberto De Prisco[1], Pompeo Faruolo[2]

[1]University of Salerno, Computer Science Department, Salerno, Italy.

[2]eTuitus, Fisciano (SA), Italy - www.etuitus.it

{dcastellano, robdep}@unisa.it, pompeo.faruolo@etuitus.it

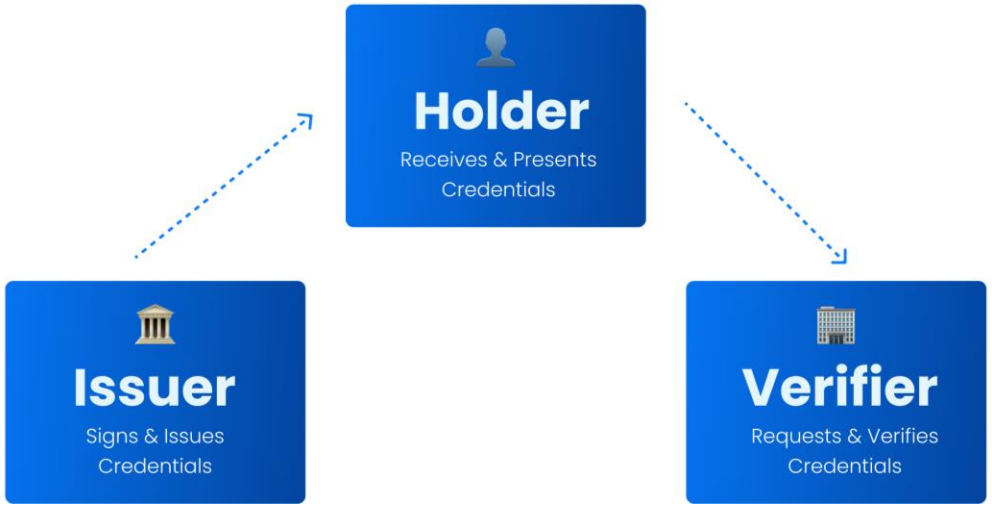
Overview

- SSI and OpenID Connect
- Implementation and Architecture
- Key benefits and challenges
- Results and Future Directions

SSI and OpenID Connect

Fundamentals of Identity Management and Standards

SSI and Verifiable Credentials



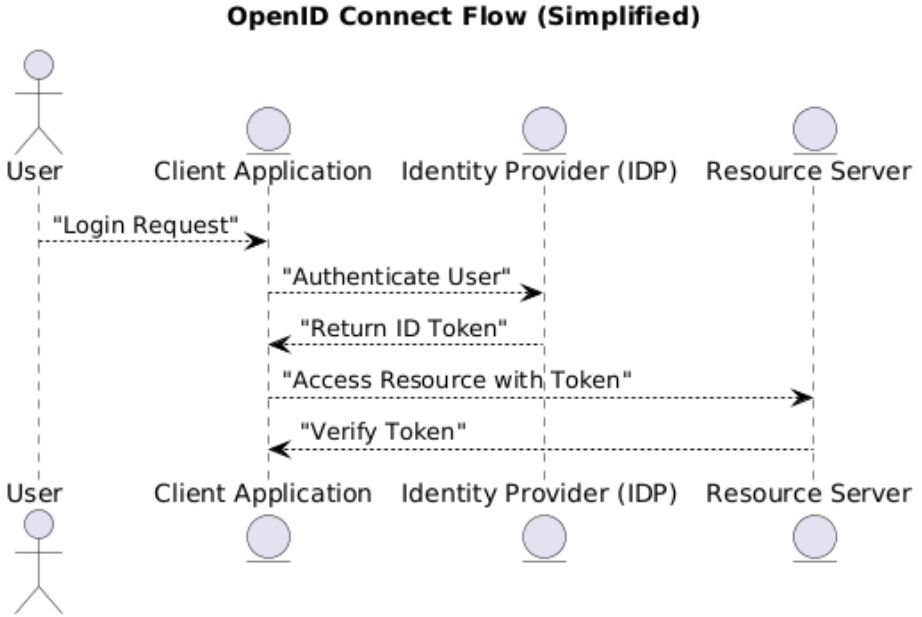
The Issuer-Holder-Verifier Model

https://docs.walt.id/concepts/digital-credentials/sd-jwt-vc#the-issuer-holder-verifier-model

- Issuer-signed claims
- Owned by holder
- Multiple formats
  - W3C data model
  - Sd-JWT
  - mdoc
  - AnonCred

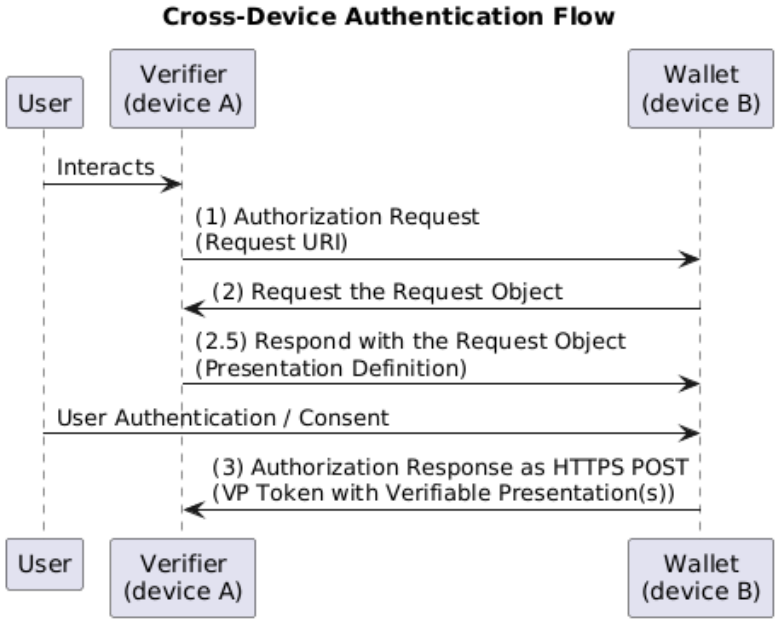
OpenId Connect

- Built on OAuth 2.0
- Usage of Tokens
- Numerous RFC published
- Extensions
  - OID4VCI
  - OID4VP
  - OID4VP over BLE
  - SIOPv2



OIDC4VP

- Extends OIDC allowing a `vp_token`
- Different flow
  - *Same device*
  - *Cross platform*

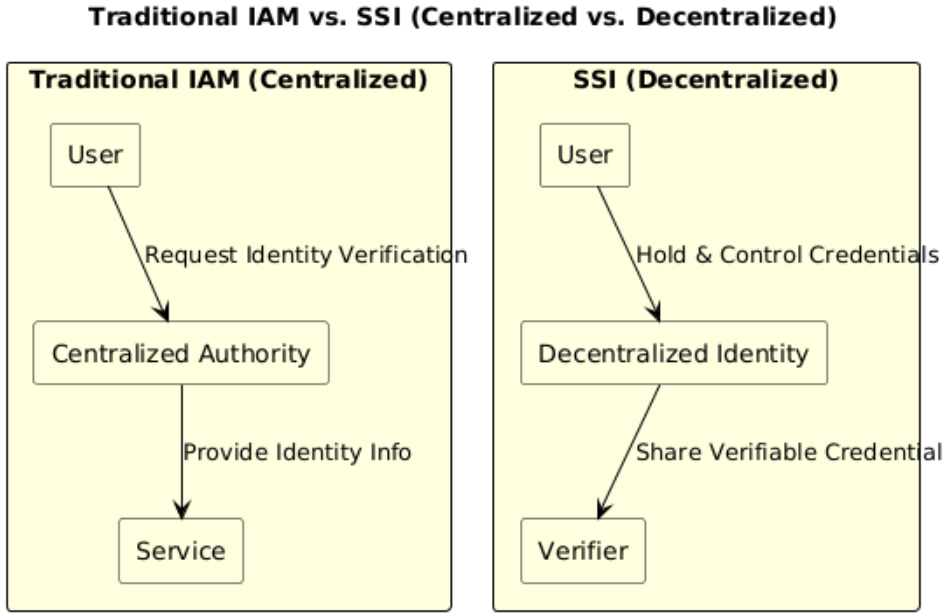


Implementation and Architecture

Designing the Solution and Technical Flow

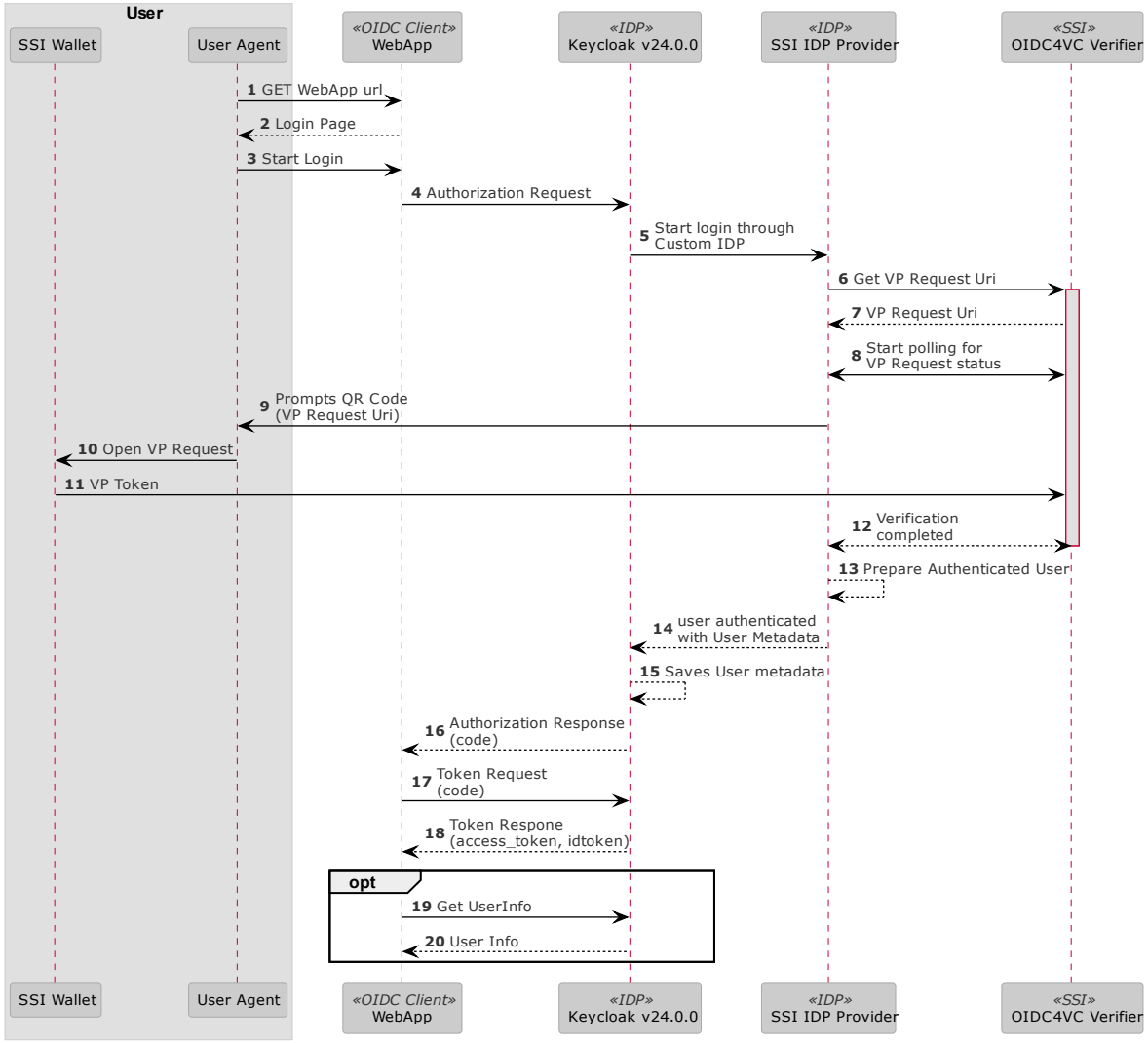
Problem and Proposed Solution

- Traditional IAM systems' limitations
- SSI Verifier component in Authorization Servers
- Keycloak IAM
  - Open-source availability
  - Extendible ecosystem



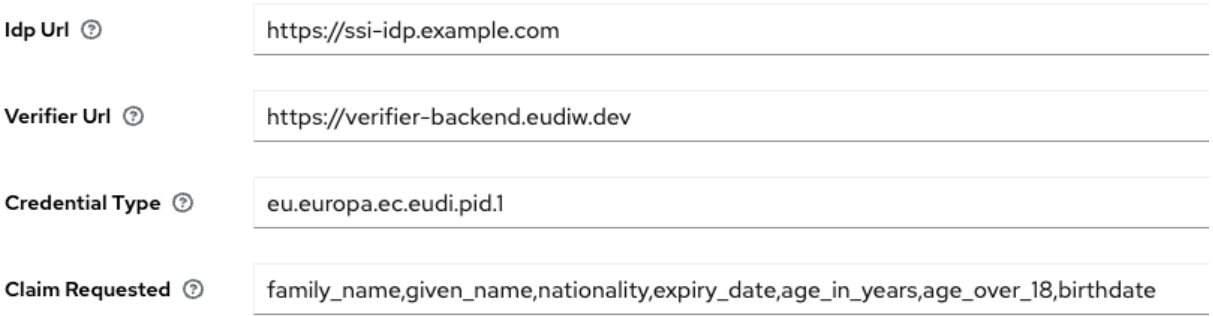
1. Navigate to application URL
2. Login Page
3. Start Login
4. Authorization Request
5. Start login through Custom IDP
6. Get VP Request URI
7. VP Request URI
8. Start polling for VP Request status
9. Prompts QR Code (VP Request URI)
10. Open VP Request
11. VP Token
12. Verification completed
13. Prepare Authenticated User
14. User authenticated with User Metadata
15. Saves User metadata
16. Authorization Response (code)
17. Token Request (code)
18. Token Response (access_token, id_token)

SSI Verifier and Authentication Flow



Keycloak and SSI Integration

- `AbstractIdentityProvider`
  - Anonymous identity creation
- Console Configuration
- Supporting service for UI



Implementation choices

- Cross-device authentication
  - Common choice
  - Similarity with other authentication systems
- European directions
  - Reference implementation
  - Pilot projects
- QR code-based verifiable credential sharing

Key Benefits and Challenges

Advantages, Risks, and Overcoming Obstacles

Challenges

Fragmented presentation protocols

Lack of standardized methods for VPs.
Work on OID4VP and SIOPv2 still in progress.

Ecosystem dependence and Lock-in

Implementations lock-in due to specific DID methods or proprietary solutions (e.g., Walt.ID SSI IDP).

Complexity of bridging solutions

Current bridging solutions are complex, requiring extensive configuration across multiple SSI ecosystems, hindering fast deployment.

Lack of standardization and practical reference

Absence of standardized, practical guidance.
Recurring draft revisions of specifications create inconsistency.

Emerging Efforts Towards Standardization

The European Digital Identity (EUDI) Wallet aims to address these issues by offering a modular and interoperable platform, though it's still in the early stages of adoption.

Valuable outcomes

Protocols Interoperability

Full adoption of OIDC and OID4VP standards.
Integration with existing IAM systems.

Mitigating SSI Ecosystem Dependence

Avoidance of proprietary protocols, leveraging on standards and reference implementations

Simplified configuration

Single configuration point with default settings for fast deployment, no barrier if using OIDC-compatible identity managers.

Contributions to Technical References

Detailed reference code.
Resources for developers and integrators.

Deployment Flexibility

Enables each service provider to deploy their own instance of the SSI provider.
Integration with other OID4VP party

Results and Future Directions

Testing outcomes and pathways for further development

Testing and Results

- Testing with EUDI Wallet
  - https://github.com/eu-digital-identity-wallet
- Results of the implementation
  - Fully compatibility with reference implementation
  - OID4VP draft 19
  - Keycloak v23 v24 compatible
  - Github release milestone
    - https://github.com/dizme/keycloak-ssi-provider

Future Work

Different research and evolution paths:

- Further standardization efforts
- Exploring interoperability between OIDC4VCI and OIDC4VP
- Evolution toward IDPKit
- Other IAM integration

Thank you!

Any questions or feedback?