OpenID Connect for SSI

Kristina Yasuda, Microsoft
Dr. Torsten Lodderstedt, yes.com

v.02
The power of Verifiable Credentials and SSI

End-users directly receiving credentials from the issuers, and directly presenting credentials to the verifiers.
The Problem

Verifiable Credentials is only a data model...  

... How to transport Verifiable Credentials when implementing?
The Simple and Secure Solution: OpenID Connect for SSI

“OpenID Connect for SSI” spec family

Issuer (Website) → Issue Credentials → Holder (Digital Wallet) → Present Credentials → Verifier (Website)

OIDC4SSI work is conducted in liaison between OpenID Foundation and DIF (Decentralized Identity Foundation)
Why extend OpenID Connect to support SSI?

- Provide the community with a solution for SSI applications leveraging the simplicity and security of OpenID Connect
  - Security of OpenID Connect has been tested and formally analysed
- Allow existing OpenID Connect RPs to access SSI credential
Direct interaction between End-users and the verifiers (Self-Issued OP v2) (former DIF DID-SIOP)

Transportation of Verifiable Credentials using OpenID Connect (OpenID Connect for Verifiable Presentations - OIDC4VP)

Issuance of Verifiable Credentials using OpenID Connect (Claims Aggregation)
<table>
<thead>
<tr>
<th>What Each Specification Provides</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SIOP V2</th>
<th>OIDC4VP</th>
<th>Claims Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Proof of possession of signing keys</td>
<td>• Presentation of verifiable credentials</td>
<td>• Unified approach for intermediaries</td>
</tr>
<tr>
<td>• Self-Signed Claims</td>
<td>issued by trusted third parties</td>
<td>(Identity Agents) to obtain claims</td>
</tr>
<tr>
<td>• Supports on same-device and cross-</td>
<td>• Can be used with SIOP v2 and</td>
<td>and credentials from trusted third</td>
</tr>
<tr>
<td>device flows</td>
<td>&quot;traditional&quot; OpenID Connect</td>
<td>parties</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Will support issuance of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>verifiable credentials</td>
</tr>
</tbody>
</table>
SIOP v2
1. SIOP v2

Self-Issued OP is an OP within the End-user’s local control. SIOP enables End-users to interact with verifiers directly, without relying on a third-party provider or having to operate their own hosted infrastructure.

OpenID Connect 3P Provider model (simplified)

① User tries to log in
3rd Party OP (3P OpenID Provider) issues an ID Token

Trust (to 3rd party)

Self-Issued OP model

① User tries to log in
OP (OP on the user device) issues ID Token

Trust (on first use)
Same-device and Cross-device SIOP

• **Same-device**
  
  User opens up a RP Website on the **same device** than where Self-Issued OP is also located

• **Cross-device**
  
  User opens up a RP Website on a **different device** than where Self-Issued OP is also located
SIOP request–response example

**SIOP Request**

```
{
  "response_type": "id_token",
  "response_mode": "post",
  "client_id": "did:example:AI6Yl81d6k...SnAxniJYu",
  "redirect_uri": "https://client.example.org/cb",
  "scope": "openid",
  "nonce": "acIlfiR6AKqGHg",
  "registration": {
    "subject_identifier_types_supported": ["did", "jkt"],
    "did_methods_supported": ["did:key", "did:example:"],
    "client_name": "Decentralized Identity Team",
    "client_purpose": "DID Authentication",
    "tos_uri": "https://client.example.org/tos.html",
    "logo_uri": "https://client.example.org/images/did_logo.png"
  },
  "exp": 1311281970,
  "iat": 1311280970
}
```

**SIOP Response – ID Token**

```
{
  "iss": "https://self-issued.me/v2",
  "sub": "did:example:EiC6Y9_aDaCsI",
  "aud": "https://client.example.org/cb",
  "nonce": "n-0S6_wzA2Mj",
  "exp": 1311281970,
  "iat": 1311280970
}
```
OpenID Connect 4 Verifiable Presentations
OpenID Connect for Verifiable Presentations enables presentation of W3C Verifiable Credentials using OpenID Connect.

- Works with all OpenID Connect Flows (SIOP v2, code, CIBA, ⋅⋅⋅)
- Request syntax uses "claims" parameter & DIF Presentation Exchange
- Supports different credential/presentation formats:
  - encoded as JSON or JSON-LD
  - signed as a JWS or Linked Data Proofs
- Supports different transports:
  - Embed in ID Token or Userinfo response
  - Return in (newly defined) VP Token alongside ID Token from authorization or token endpoint
OIDC4VP request–response example (SIOP, LD Proofs, VP Token)

Request with `claims` parameter and DIF Presentation Exchange

```
{
  "response_type": "id_token",
  "response_mode": "post",
  "client_id": "did:example:A6YL8ld6k...HaXnkJVu",
  "redirect_uri": "https://client.example.org/cb",
  "scope": "openid",
  "nonce": "acl1fiR6AKqGHg",
  "claims": {
    "id_token": "null",
    "vp_token": {
      "presentation_definition": {
        "id": "BasicProfile",
        "input_descriptors": [
          "id": "IDCardCred",
          "schema": "https://www.w3.org/2018/credentials/examples/v1/IDCardCredential",
          "constraints": {
            "limit_disclosure": "required",
            "fields": [
              {
                "path": "/.vc.credentialSubject.given_name"},
              {
                "path": "/.vc.credentialSubject.family_name"},
              {
                "path": "/.vc.credentialSubject.birthdate"
              }
            }
          }
        }
      }
    },
    "registration": {...},
    "exp": 1311281970,
    "iat": 1311280970
  }
```

Response – decoded ID Token

```
{
  "iss": "https://self-issued.me/v2",
  "sub": "did:example:E1COY9_aDaCSI",
  "aud": "https://client.example.org/cb",
  "nonce": "n=O5G_nzA2Mj",
  "exp": 1311281970,
  "iat": 1311280970
}
```

Response – VP Token containing Verifiable Presentation

```
"vp_token": {
  "format": "idp_vp",
  "presentation": {
    "@context": "https://www.w3.org/2018/credentials/v1",
    "type": "VerifiablePresentation",
    "id": "bc6f1c2",
    "proof": {
      "type": "Ed25519Signature2018",
      "created": "2021-09-19T15:30:15Z",
      "challenge": "n=O5G_nzA2Mj",
      "domain": "https://client.example.org/cb",
      "jws": "eyJhbGciOiJFZERTQ0SIaW1zIjoiLiBOelBleGNzcGFuZ3I5NSB4eXZLdW5t
        proofPurpose": "authentication",
        "verificationMethod": "did:example:holder#key-1"},
      "verifiableCredential": {
        "@context": "https://www.w3.org/2018/credentials/v1",
        "id": "https://example.com/2018/credentials/verifiableCredential"}
  }
}
```
DEMO

Bringing it all together ...
SIOP v2 / OIDC4VPs Prototype

• Implemented within IDUnion project
• Team: Sebastian Bickerle, Paul Wenzel, Fabian Hauck, & Dr. Daniel Fett
• Use Case: Login to NextCloud using Verifiable Credentials
• Based on
  • Existing NextCloud OpenID Connect Plugin
  • lissi Wallet
  • Hyperledger Indy & Indy SDK
DEMO

• On device: https://youtu.be/gDg2ma7TwWU
• Cross device: https://youtu.be/hC3VQE-vMnQ
Details & Findings

• SIOP instead of DIDComm
• No separate connection establishment step required
• On device:
  • Direct communication between verifier and wallet w/o cloud agent/network communication
• Cross device:
  • Additional backend call from wallet to verifier (HTTPS POST)
  • QR Code pretty huge
Next Steps

- **SIOP v2**
  - Resolvable client ids (DIDs, Entity Statements)
  - OP Discovery
  - Security Analysis
- **OIDC4VP**
  - Integration of presentation submissions
  - Additional Security Considerations
  - Gather Implementors Feedback
- **Claims Aggregation**
  - Request by credential type
  - Proof of possession of key material (vs client authentication)
  - Use with other grant type than "code"
If you want to learn more

Wednesday, September 15, 2021 17:20–17:40
Location: AMMERSEE II

Self-Issued OP, or Decentralized Identity with OpenID Connect
Backup
3. Claims Aggregation

Enables Holder to obtain Verifiable Credentials from the Issuer(s).

- Under Development (merged with Credential Provider draft)
Status and Topics being worked on

- Adoption
- Prototypes
- Open Topics
3 components of “SIOP” work

Presentation
1. Self-Issued OpenID Provider model
2. SIOP can present claims to the RP as W3C Verifiable Presentations

Issuance
3. SIOP get claims issued from the Claims Providers

* 2 and 3 are applicable to the entire OpenID Connect
Use Cases

1. Resilience against Sudden or Planned OP Unavailability (natural disasters, a planned business decision, etc.)
2. Authentication at the edge, in environments which may have reduced connectivity.
3. Sharing credentials from several issuers in one transaction
4. Aggregation of multiple personas under one Self-Issued OP, as an alternative to using multiple OPs for different RPs
2. OIDC4VP - A) Embedding an entire VP inside the ID Token (SIOP)

Request

```json
{
  "id_token": {
    "acr": "null",
    "verifiable_presentations": {
      "credential_types": [
        {
          "type": "https://did.itso.org/documents/verifiable-credential-type"
        }
      ],
      "verifiable_presentations": [
        {
          "format": "vp-jwt",
          "presentation": "ewogICAgImlzc3Jvcll0b3Itd2FtcGxhY2UueXN0cmluZw=="
        }
      ],
      "sub": "did:ion:E1c6Y9_0DaCsITtY06HId4seJjJ...b1df31ec42d0",
      "alg": "ES256",
      "kid": "did:ion:E1c6Y9_0DaCsITtY06HId4seJjJ...b1df31ec42d0"
    }
  }
}
```

Response

```json
{
  "id_token": {
    "acr": "null",
    "verifiable_presentations": {
      "credential_types": [
        {
          "type": "https://did.itso.org/documents/verifiable-credential-type"
        }
      ],
      "verifiable_presentations": [
        {
          "format": "vp-jwt",
          "presentation": "ewogICAgImlzc3Jvcll0b3Itd2FtcGxhY2UueXN0cmluZw=="
        }
      ],
      "sub": "did:ion:E1c6Y9_0DaCsITtY06HId4seJjJ...b1df31ec42d0",
      "alg": "ES256",
      "kid": "did:ion:E1c6Y9_0DaCsITtY06HId4seJjJ...b1df31ec42d0"
    }
  }
}
```

Base64URL encoded VP in a JWT format
2. OIDC4VP – B) Returning VP as a VP Token (code flow)

Request

```
{
    "id_token": {
        "acr": "null",
        "vp_token": {
            "format": "json-ld",
            "credential_types": [
                {
                    "type": "https://www.w3.org/2018/context/v1",
                    "claims": {
                        "given_name": "null",
                        "family_name": "null",
                        "birthdate": "null"
                    }
                }
            ]
        }
    }
}
```

Response – ID Token

```
{
    "iss": "http://server.example.com",
    "sub": "248289761001",
    "aud": "s6HdRkgt3",
    "nonce": "n-0S6_WzA2Mj",
    "exp": 1311281970,
    "iat": 1311280970,
    "auth_time": 1615910535
}
```

ID Token and VP Token are bound via `nonce`

Response – VP Token

```
{
    "access_token": "SIAV32hkKG",
    "token_type": "Bearer",
    "refresh_token": "8xLOxBtZp8",
    "expires_in": 3600,
    "id_token": "eyJ0 ... NiJ9.eyJ1c ... I6IjJifX0.DeWt4Qu ... ZXso",
    "vp_token": {
        "format": "vp_ldp",
        "presentation": {
            "@context": [
                "https://www.w3.org/2018/credentials/v1"
            ],
            "type": "VerifiablePresentation",
            "verifiableCredential": [
                {
                    "@context": [
                        "https://www.w3.org/2018/credentials/v1",
                        "https://www.w3.org/2018/credentials/examples/v1"
                    ],
                    "type": "VerifiableCredential"
                }
            ]
        }
    }
}
```

‘VP Token’ contains an entire VP
## All variations of OIDC4VP

<table>
<thead>
<tr>
<th>Format (JWT/JSON-LD)</th>
<th>Way to present (ID Token/VP Token)</th>
<th>Protocol (SIOP / usual OIDC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JWT</td>
<td>Inside ID Token</td>
<td>SIOP</td>
</tr>
<tr>
<td>JWT</td>
<td>VP Token</td>
<td>SIOP</td>
</tr>
<tr>
<td>JWT</td>
<td>Inside ID Token</td>
<td>Usual OIDC</td>
</tr>
<tr>
<td>JWT</td>
<td>VP Token</td>
<td>Usual OIDC</td>
</tr>
<tr>
<td>JSON-LD</td>
<td>Inside ID Token</td>
<td>SIOP</td>
</tr>
<tr>
<td>JSON-LD</td>
<td>VP Token</td>
<td>SIOP</td>
</tr>
<tr>
<td>JSON-LD</td>
<td>Inside ID Token</td>
<td>Usual OIDC</td>
</tr>
<tr>
<td>JSON-LD</td>
<td>VP Token</td>
<td>Usual OIDC</td>
</tr>
</tbody>
</table>