Self-issued OpenID Provider (SIOP) Update

Kristina Yasuda
Microsoft

December 2021
OpenID Foundation Workshop

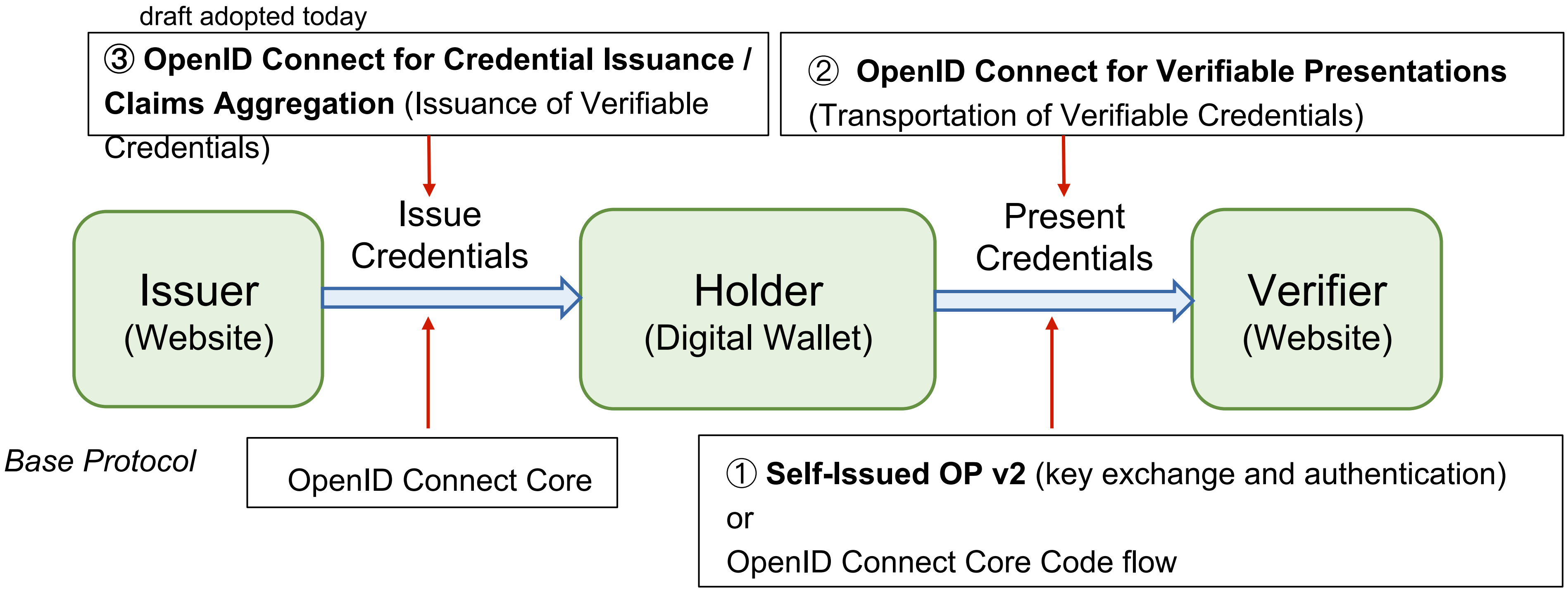
OpenID Connect for SSI

- Initiative conducted at OpenID Foundation in liaison with the Decentralized Identity Foundation
- Aims at specifying a set of protocols based on OpenID Connect to enable SSI applications

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 and existing OpenID Connect OPs to also issue credentials

OpenID Connect for SSI Components



draft adopted today

③ **OpenID Connect for Credential Issuance / Claims Aggregation** (Issuance of Verifiable Credentials)

② **OpenID Connect for Verifiable Presentations** (Transportation of Verifiable Credentials)

Issuer (Website)

Issue Credentials

Holder (Digital Wallet)

Present Credentials

Verifier (Website)

Base Protocol

OpenID Connect Core

① **Self-Issued OP v2** (key exchange and authentication)
or
OpenID Connect Core Code flow

Use Case: eKYC using Bank ID Credential

- A Bank customer wants to sign a car leasing contract. The leasing provider is obliged to identify the customer (Anti-Money Laundering Law). The leasing provider accepts identification via a digital wallet.
- The leasing provider sends the customer to the wallet, which currently does not contain a suitable credential. The wallet offers some options to the customer to obtain the required credential.
- The customer selects the bank-based approach, selects her bank, and proceeds to the bank's digital banking experience where she is offered to issue an ID credential to the wallet.
- Upon receiving an ID credential to the wallet, when prompted, the customer confirms presentation of the new credential to the leasing company.
- The leasing company receives the credential and proceeds to the next step in the contract signing process…
- Customer benefit: ease of use of bank-based identification in combination with additional privacy preservation through the wallet since there is no call-home to the Bank directly by the leasing provider.
- Leasing provider benefit: ability to accept ID credentials from the Issuers/Banks without establishing a federation with them

Use Case: NHS doctors moving around 1200 clinics

- A clinic wants to verify identity a doctor who has been assigned to work there from another clinic. The doctors are moved between clinics quite often and each time a clinic onboards a new doctor, it is obliged to verify the doctor’s identity and privileges. The clinic accepts identification via a digital wallet.
- The clinic sends the doctor to the wallet. If the wallet does not currently contain a suitable credential, it offers some options to the doctor to obtain the required credential.
- The doctor selects the previous clinic where he/she worked, and proceeds to that clinic’s credential issuance perience where he/she is offered to issue an ID credential to the wallet.
- Upon receiving an ID credential to the wallet, when prompted, the doctor confirms presentation of the new credential to the new clinic.
- The new clinic company receives the credential and proceeds to the next step in the contract signing process…
- Customer benefit: ease of use of digital credentials instead of having to collect paper documents from all the previous clinics.
- New clinic: ability to accept credentials in a digital format significantly shortens onboarding time and gives more time to the doctors to actually treat the customers instead of dealing with administrative processes.

Specifications Status

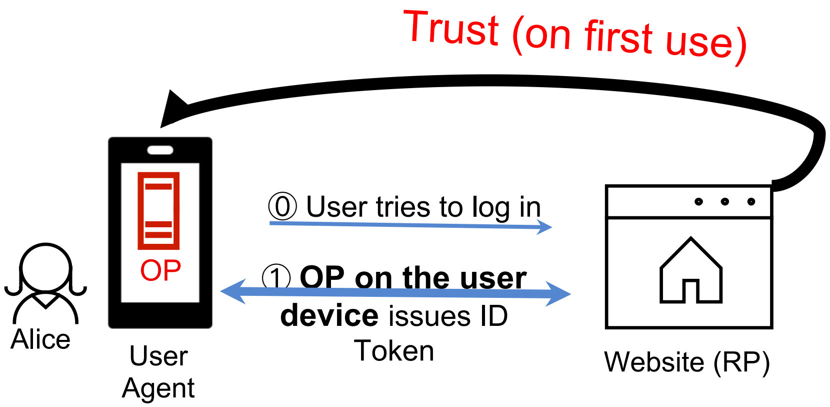
- OIDC for Credential Issuance: draft adopted by the Connect WG on 2021-12-09
- SIOP v2: call for the Implementor’s draft planned to start next week
- OIDC for Verifiable Presentations: call for the Implementor’s draft planned to start next week
- SIOP v2 (potentially OIDC4VP) in ISO/IEC 23220-4 mID

Self-Issued OP (SIOP) v2

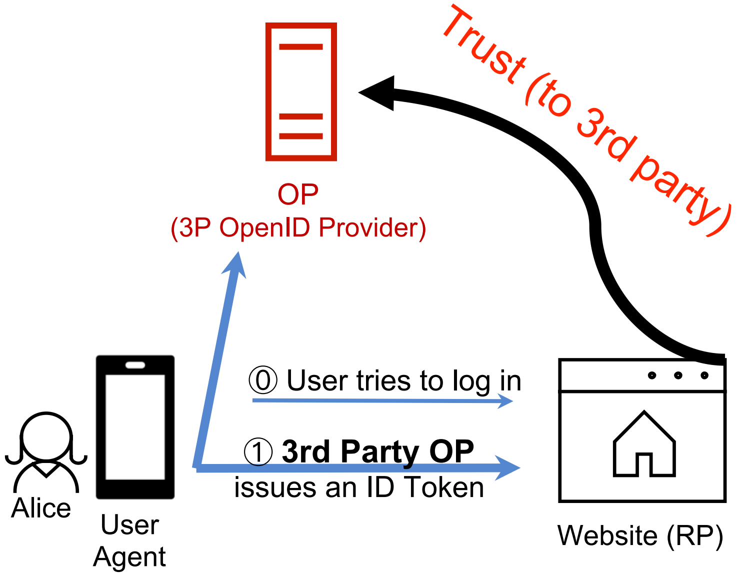
1. SIOP v2

- Self-Issued OP is an OP within the End-user’s local control.
- Enables End-users to interact with verifiers directly, without relying on a third-party providers.
- Performs key exchange and authentication.

Self-Issued OP model



OpenID Connect standard model



What’s new in v2?

- DIDs as “sub” values (in additional to raw public keys) to “self-issue” an ID Token
- Dynamic SIOP discovery when SIOP identifier is pre-known
- SIOP invocation via HTTPS URLs
  - in addition to “openid://” custom scheme
  - enables use of app/universal links and web wallets
- Dynamic RP registration
  - Pre-registration
  - `registration` parameter in the request (DID or Entity statements)
- Cross Device Flow
  - response_mode=post
  - Security considerations

SIOP request–response example

SIOP Request

on device

```
HTTP/1.1 302 Found
  Location: openid://?
 response_type=id_token
 &client_id=https://client.example.org/cb
 &redirect_uri=https://client.example.org/cb
 &scope=openid%20profile
 &nonce=n-0S6_WzA2Mj
```

cross device

```
openid://?
 response_type=id_token
 &response_mode=post
 &client_id=https://client.example.org/cb
 &redirect_uri=https://client.example.org/post_cb
 &scope=openid%20profile
 &nonce=n-0S6_WzA2Mj
```

SIOP Response

on device

```
HTTP/1.1 302 Found
  Location: https://client.example.org/cb#
    &id_token=eyJ0 ... NiJ9.eyJ1c ... I6IjIifX0.D ... ZXso
```

cross device

```
POST /post_cb HTTP/1.1
  Host: client.example.com
  Content-Type: application/x-www-form-urlencoded

    &id_token=eyJ0 ... NiJ9.eyJ1c ... I6IjIifX0.D ... ZXso
```

SIOP request–response example

Decoded ID Token

```
{
   "iss": "https://self-issued.me/v2",
   "sub": "did:example:EiC6Y9_aDaCsl",
https url -> jwks_uri hosted under .well-known
   "aud": "https://client.example.org/cb",
   "nonce": "n-0S6_WzA2Mj",
   "exp": 1311281970,
   "iat": 1311280970
}
```

“sub” can be raw public key or DID

OpenID Connect 4 Verifiable Presentations

2. OIDC4VP

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 v2**
- Supports **different credential/presentation formats**:
  - encoded as JSON or JSON-LD
  - signed as a JWS or Linked Data Proofs
  - Anon creds
  - ...
- VP returned as a newly defined VP Token (provided alongside ID Token) :
  - Embedding VP in ID Token or Userinfo response is not supported anymore

OIDC4VP Request

SIOP+OIDC4VP Request

```
HTTP/1.1 302 Found
  Location: openid://?
 response_type=id_token
 &client_id=https://client.example.org/cb
 &redirect_uri=https://client.example.org/cb
 &scope=openid%20profile
 &nonce=n-0S6_WzA2Mj
 &claims=...
```

`claims` parameter with DIF Presentation Exchange Syntax

```
{
   "vp_token": {
      "presentation_definition": {
         "id": "identification",
         "input_descriptors": [
            {
               "id": "id_card_credential",
               "schema": [
                  {
                     "uri":"https://www.w3.org/2018/credentials/examples/v1/IDCardCredential"
                  }
               ]
            }
   ...
}
```

OIDC4VP Response

Response

```
HTTP/1.1 302 Found
  Location: https://client.example.org/cb#
    &id_token=eyJ0 ... NiJ9.eyJ1c ...
    &vp_token=...
```

Decoded ID Token

```
{
   ...
   "_vp_token": {
    "presentation_submission": {
      "id": "example presentation",
      "definition_id": "identification",
      "descriptor_map": [
       {
         "id": "id_card_credential",
         "format": "ldp_vp",
         "path": "$"
         "path_nested": {
           "format": "ldp_vc",
           "path": "$.verifiableCredential[0]"
         }
    ...
}
```

Path in the ID Token refers to a separate artifact - VP Token

VP Token containing Verifiable 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"
        ],
        "id": "https://example.com/credentials/1872",
        "type": [
           "VerifiableCredential",
           "IDCardCredential"
        ],
        "issuer": {
           "id": "did:example:issuer"
        },
        "issuanceDate": "2010-01-01T19:23:24Z",
        "credentialSubject": {
           "given_name": "Fredrik",
```

OpenID Connect 4 Credentials Issuance

Status

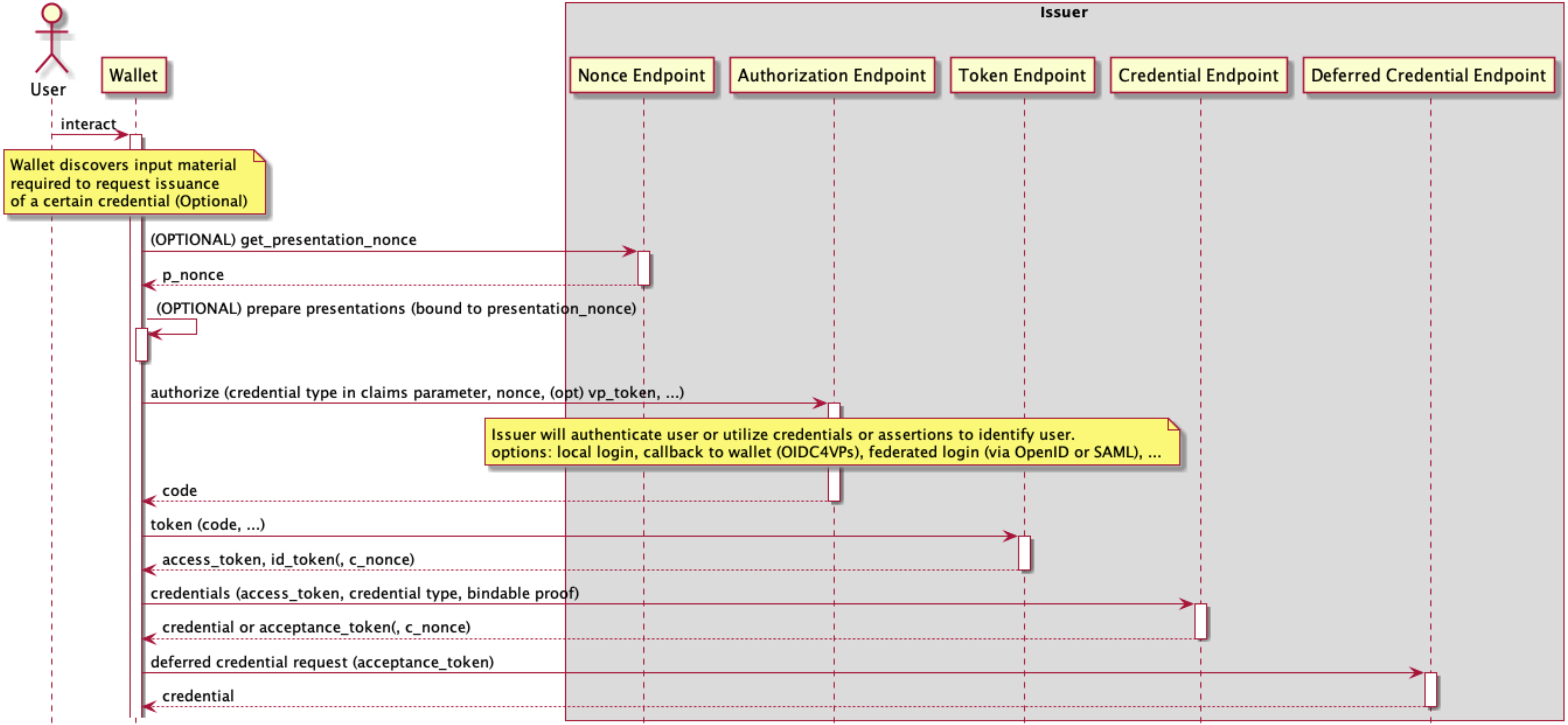
- Based on pre-existing implementation experience (MATTR Ltd & Convergence.tech)
- Work recently started with requirements gathering and brainstorming sessions at IIW #33
- Draft being checked into Bitbucket
  - openid-connect-4-credential-issuance-1_0.

Benefits

- Turning existing OpenID Connect OPs into Issuers
- Allows for inline credential issuance
- Issuer-controlled UX gives flexibility
- Wallet and Issuer identification utilizing OpenID Connect metadata

Key Ideas

- Issuance via OAuth protected API: Credential endpoint
- Multiple credentials, Same credential in different formats/keys
- Support all kinds of proof methods (also non JWS) for key material the new credential shall be bound to
- Separate client authentication & message integrity protection from proof of possession of this key material
- Allow presentation of credentials (in authz request and dynamically obtained) as input for the issuance
- Allow deferred issuance
- Support Credential metadata (Credential Manifest)



Issuer

User

Wallet

Nonce Endpoint

Authorization Endpoint

Token Endpoint

Credential Endpoint

Deferred Credential Endpoint

interact

Wallet discovers input material
required to request issuance
of a certain credential (Optional)

(OPTIONAL) get_presentation_nonce

p_nonce

(OPTIONAL) prepare presentations (bound to presentation_nonce)

authorize (credential type in claims parameter, nonce, (opt) vp_token, ...)

Issuer will authenticate user or utilize credentials or assertions to identify user.
options: local login, callback to wallet (OIDC4VPs), federated login (via OpenID or SAML), ...

code

token (code, ...)

access_token, id_token(, c_nonce)

credentials (access_token, credential type, bindable proof)

credential or acceptance_token(, c_nonce)

deferred credential request (acceptance_token)

credential

New Endpoints & New Parameters

- New endpoints:
  - **Nonce Endpoint**: provides the RP with a nonce it will include into verifiable presentations sent to the authorization endpoint
  - **Credential Endpoint**: OAuth-protected API to issue verifiable credentials
  - **Deferred Credential Endpoint**: used for deferred issuance of verifiable credentials
- Extended endpoints:
  - **Server Metadata**: new metadata parameters (credential types, …)
  - **Authorization Endpoint**:
    - `claims` parameter allows to request authorization for issuance of one or more credentials.
    - new parameters to convey verifiable presentations and further data to alternatively callback to the RP (acting as wallet) to request further verifiable credentials.
    - PAR is recommended
  - **Token Endpoint**: optional parameters to provide RP with a nonce to used for proof of possession of key material

Ongoing/Planned implementations

- Microsoft
- walt.id & yes.com & BCDiploma (ESSIF)
- Ping Identity
- Convergence.Tech
- Gematik (within IDunion)
- Sphereon
- Gimly
- Talao.io
- Workday
- Spruce
- …

*Some ESSIF projects already utilizes SIOP (based on DID-SIOP & OpenID Connect 4 Identity Assurance)

Future Work Items

- OIDC for Credential Issuance
- SIOP v2
  - Pushed Authorization Response Mode
  - Cross-device Security
  - Making each piece stable (JWK Thumbprint URI, etc.)
- OIDC4VP
  - PE syntax version
- Potential synergy with FIDO/WebAuthn

*Some ESSIF projects already utilizes SIOP (based on DID-SIOP & OpenID Connect 4 Identity Assurance)