

OpenID Connect for SSI

Kristina Yasuda, Microsoft

Dr. Torsten Lodderstedt, yes.com

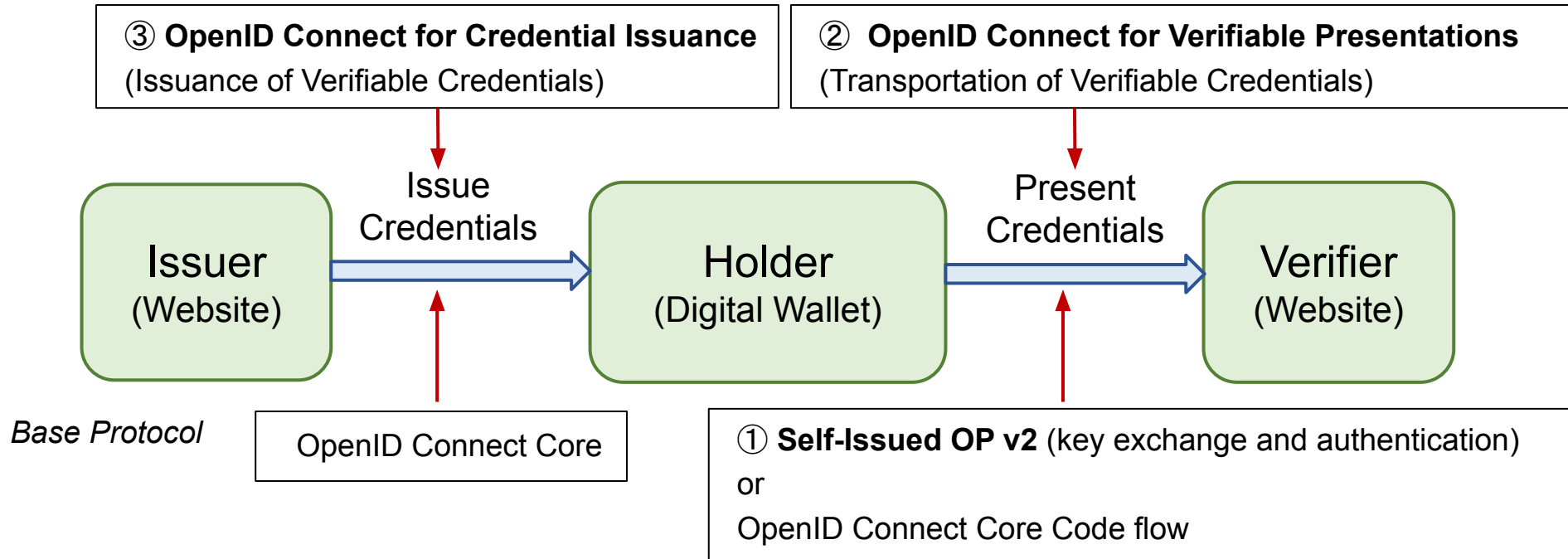
OpenID Connect for SSI

- 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



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 experience 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.

Use Case: University managing alumni

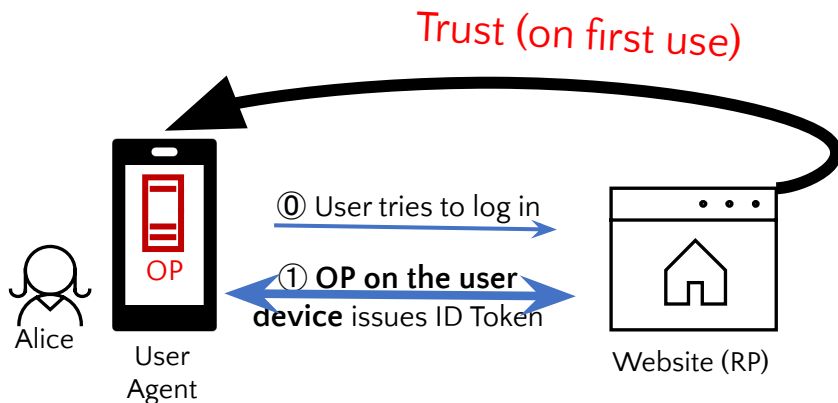
- An employer wants to verify the education history of the employee during the hiring on-boarding process. The employee accepts identification via a digital wallet.
- The employer sends the employee to the wallet. If the wallet does not currently contain a suitable credential, it offers some options to the employee to obtain the required credential.
- The employee selects the university he/she has graduated from, and proceeds to that university's credential issuance experience where he/she is offered to issue an ID credential to the wallet.
- Upon receiving an ID credential to the wallet, when prompted, the employee confirms presentation of the new credential to the employee.
- The employee 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 physically go to the university and collect paper documents or having to exchange emails with the university admin.
- New clinic: ability to move growing number of alumni data to the cold storage once, instead of storing it in an active storage ready to respond to an API call anytime.

Self-Issued OP (SIOP) v2

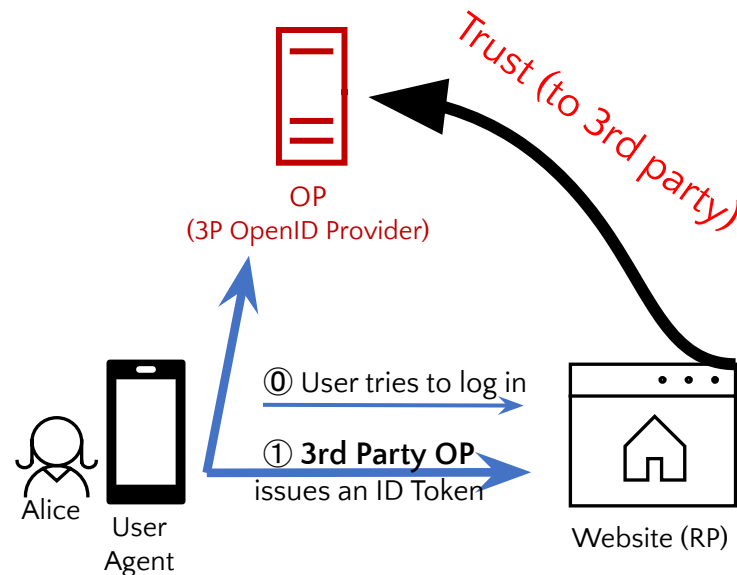
1. SIOP v2

- 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 addition to raw public keys) to “self-issue” an ID Token
- Dynamic SLOP discovery and invocation via HTTPS URLs
 - in addition to “openid://” custom scheme
 - enables use of app/universal links and web wallets
- Dynamic RP registration (DID or Entity statements)
- Cross Device Flow

SLOP request–response example

SLOP 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
```

SLOP Response

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

```
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
```

SLOP request–response example

Decoded ID Token

```
{  
  "iss": "https://self-issued.me/v2",  
  "sub": "did:example:EiC6Y9_aDaCsI",  
  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**
- Supports **different credential/presentation formats**:
 - encoded as JSON or JSON-LD
 - signed as a JWS or Linked Data Proofs
 - ...
- Supports **different transports**:
 - Embedded in ID Token or Userinfo response
 - new VP Token (provided alongside ID Token)

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

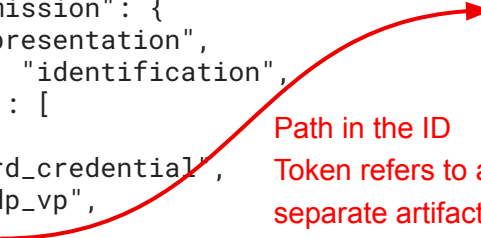
Response

```
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",

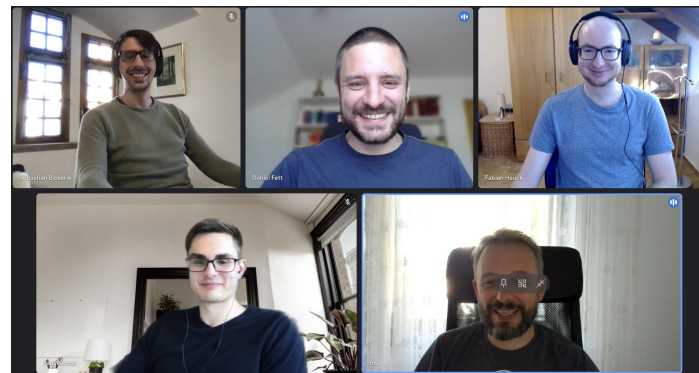
```


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



iDunion
Supported by:



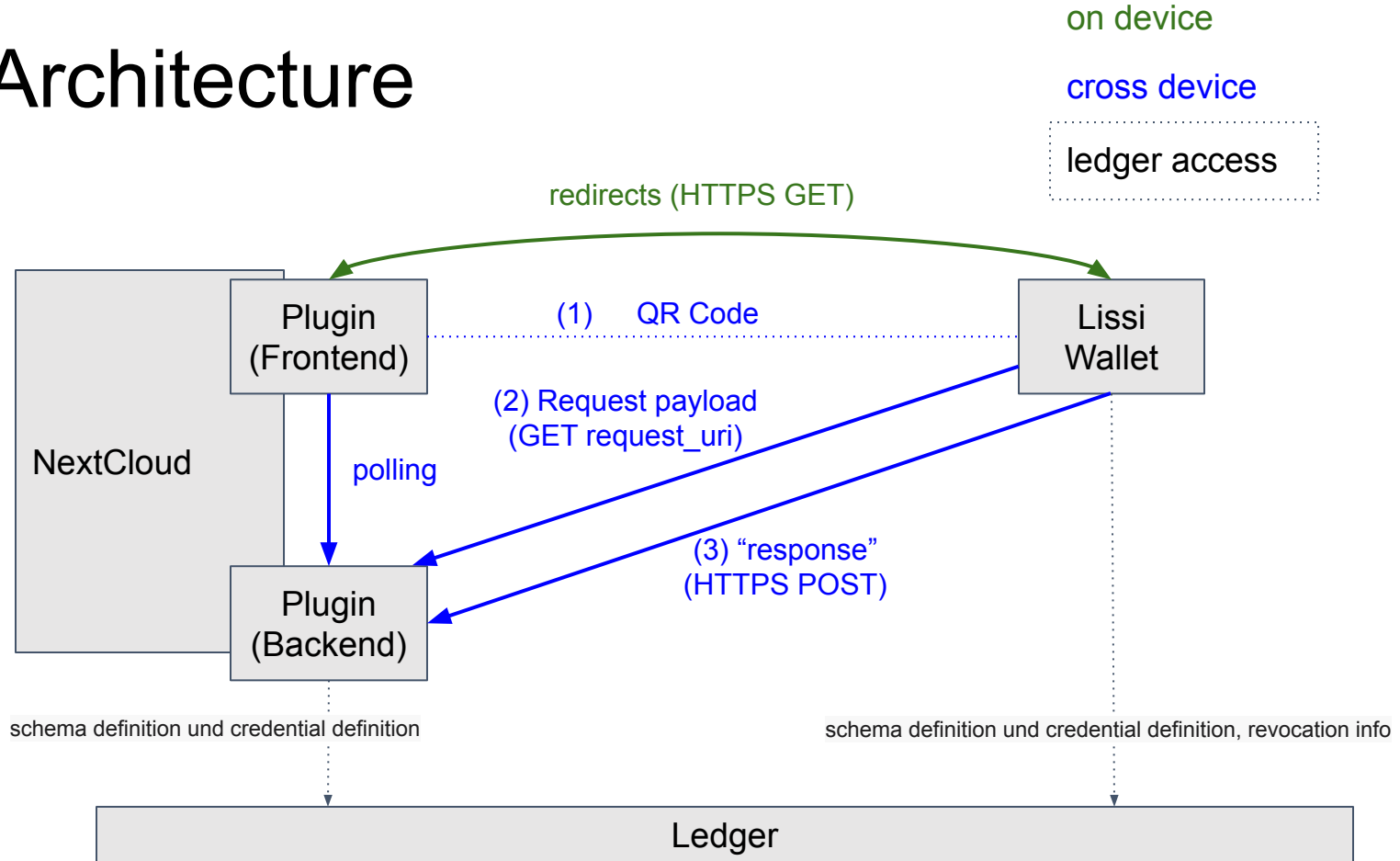
Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag

DEMO

- On device: <https://youtu.be/gDg2ma7TwWU>
- Cross device: <https://youtu.be/hC3VQE-vMnQ>

Architecture



Findings

- QR Codes pretty huge
- Verifiers can be identified
 - Verifier's Web URL shown in wallet (leverages OIDC client data model)
- Simple & Scalable Architecture
- No separate connection establishment step required

Further ongoing implementations

- Microsoft
- Convergence.Tech
- Gematik (within IDunion)
- walt.id (eSSIF-Lab)*
- Sphereon
- Gimly

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

OpenID Connect 4 Credentials Issuance

Status

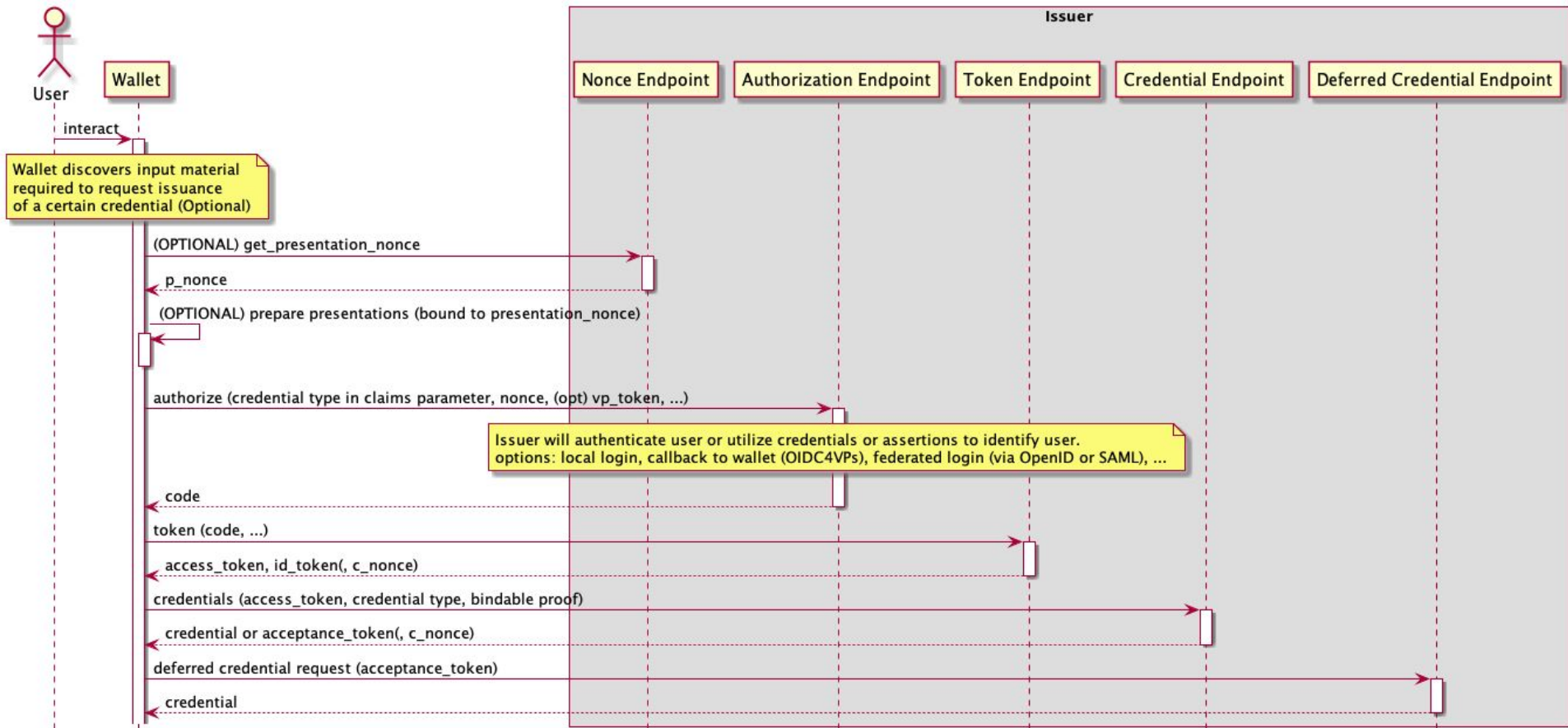
- Based on pre-existing implementation experience (MATTR Ltd & Convergence.tech)
- Work recently started with requirements gathering and brainstorming sessions at IIW #33
- Current WG Draft
https://openid.bitbucket.io/connect/openid-connect-4-verifiable-credential-issuance-1_0.html

Benefits

- Obvious way to turn 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)



New Endpoints

- **Issuance Initiation Endpoint:** An endpoint exposed by the wallet that allows an issuer to initiate the issuance flow
- **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

- **Client Metadata:** new metadata parameter is added to allow a wallet (acting as OpenID Connect RP) to publish its issuance initiation endpoint.
- **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

Planned implementations

- Microsoft
- IDunion
- walt.id & yes.com & BCDiploma (eSSIF-Lab)
- Sphereon
- Talao.io
- Convergence.Tech

Thank you!