OpenID for SSI

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

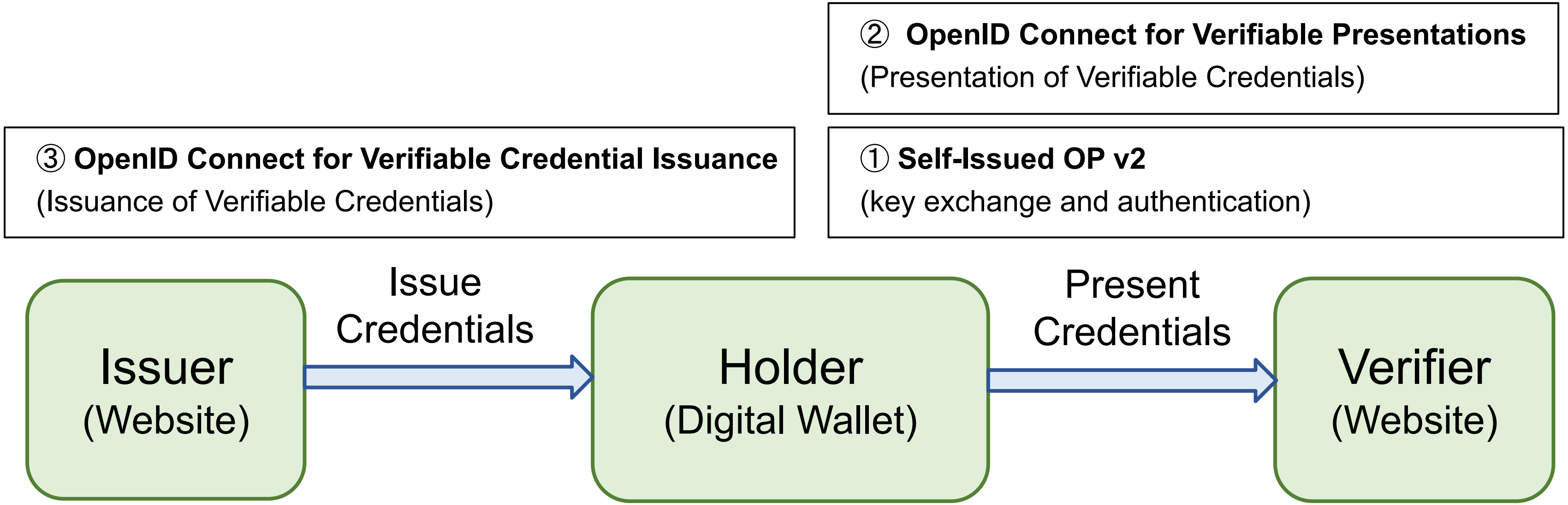
OpenID for SSI

- Aims at specifying a set of protocols based on OpenID Connect and OAuth2.0 to enable SSI applications
- Initiative conducted at OpenID Foundation in liaison with the Decentralized Identity Foundation (DIF)
- One of the specifications is built up on DID-SIOP in DIDAuth WG in DIF and SIOPv1 chapter 7 in OIDC Core

Why use OpenID Connect/OAuth2.0 as basis?

- Self-Issued OP (SIOP) already provides good starting point
- Leveraging the simplicity and security of OpenID Connect and OAuth2.0 for SSI applications
    - Existing libraries, only HTTPS communication, developer familiarity
    - Great for mobile applications, no firewall hassles
    - Security of OpenID Connect has been tested and formally analysed
- Allow existing OpenID Connect RPs to access SSI credentials and existing OpenID Connect OPs to issue credentials

OpenID Connect for SSI Components



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

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

① **Self-Issued OP v2**
(key exchange and authentication)

Issuer
(Website)

Issue
Credentials

Holder
(Digital Wallet)

Present
Credentials

Verifier
(Website)

Can be hosted locally on the user’s device, have cloud components, or be entirely hosted in the cloud

OIDC4SSI allows variety of choices in the SSI tech stack

Using OIDC4SSI as an authentication protocol to present and issue credentials allows implementers to choose a combination of DID methods, credential formats and other components of the SSI tech stack.

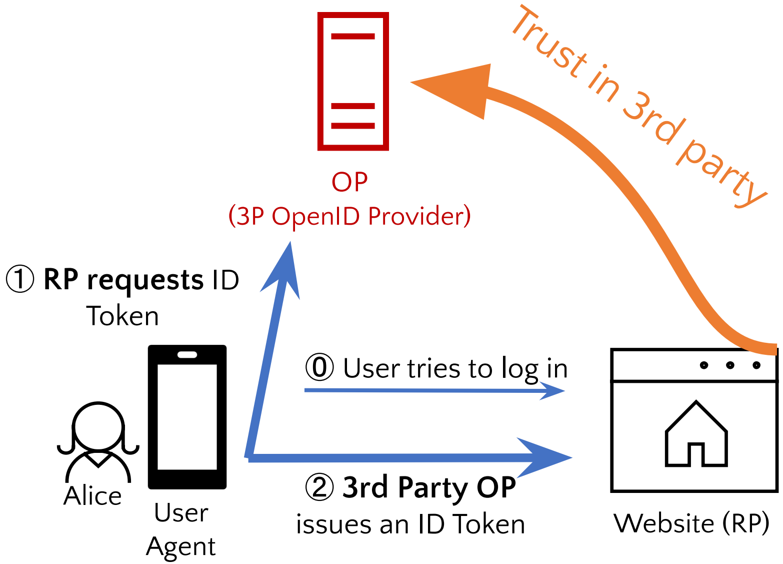
SSI Tech Stack component	**Implementer's choices when using OIDC4SSI as a protocol**
Identifiers	**Any DID method** - user's identifier can also be a JWK Thumbprint (\`sub\` in the ID Token) - verifier's identifier can also be a unique string (\`client_id\` in the request)
Credential Format	**Any credential format** (AnonCreds, LDP-VC, JWT-VC, ISO mDL, etc.)
Revocation	**Any mechanism** (Status List 2021, etc.)
additional trust mechanisms	**Any mechanism** (.well-known DID configuration, etc.)
Cryptography	**Any cryptosuite** (EdDSA, ES256K, etc.)

some use-cases…

SIOPv2

Standard OpenID Connect vs SIOP v2

OpenID Connect standard model



Trust in 3rd party

OP
(3P OpenID Provider)

① **RP requests** ID Token

⓪ User tries to log in

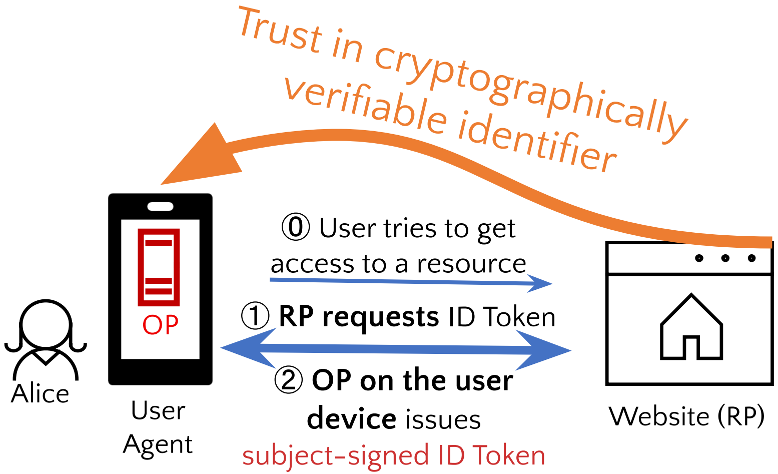
② **3rd Party OP** issues an ID Token

Alice

User Agent

Website (RP)

Self-Issued OP model



Trust in cryptographically verifiable identifier

⓪ User tries to get access to a resource

① **RP requests** ID Token

② **OP on the user device** issues subject-signed ID Token

OP

Alice

User Agent

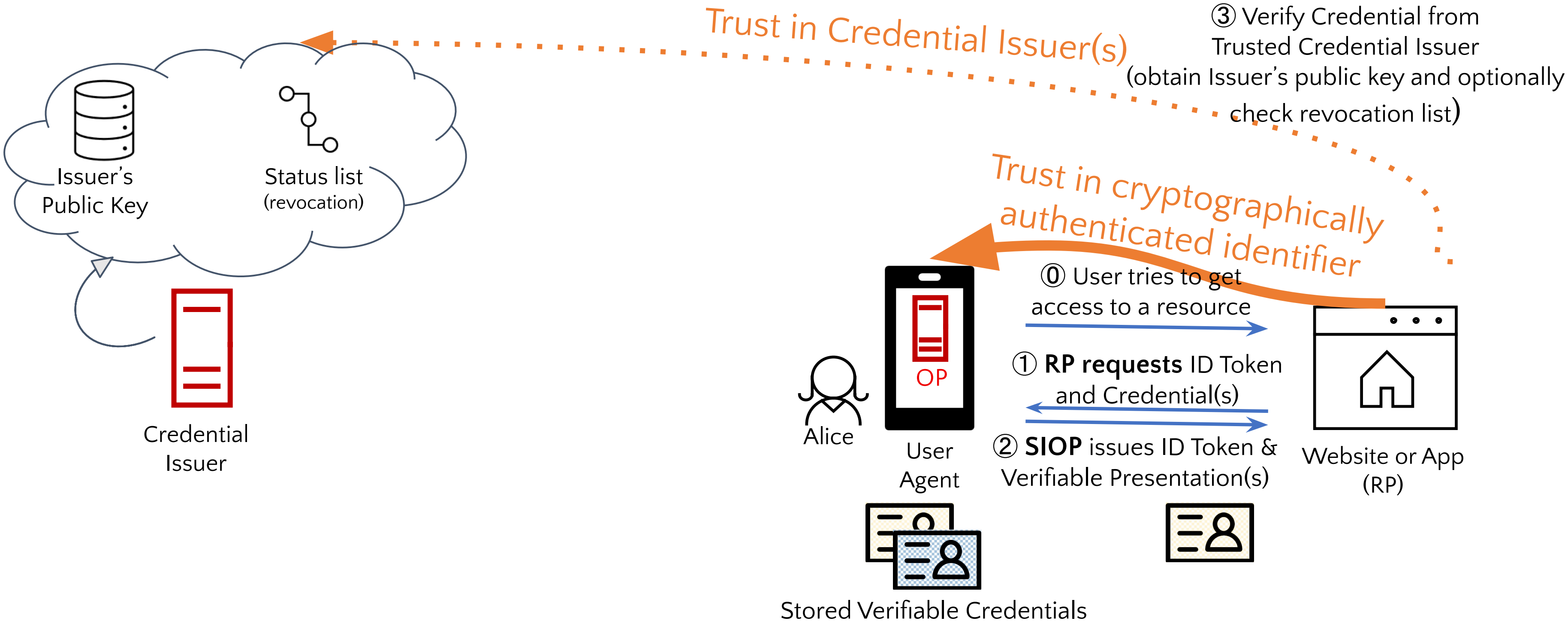
Website (RP)

User-controlled OpenID Connect OP is able to self-sign ID Tokens and authenticate using the user-controlled key material (raw public keys or Decentralized identifiers (DIDs))

OpenID Connect for Verifiable Presentations

SIOP v2 + OpenID Connect 4 Verifiable Presentations

Presenting Credentials



Trust in Credential Issuer(s)

③ Verify Credential from Trusted Credential Issuer (obtain Issuer's public key and optionally check revocation list)

Issuer's Public Key

Status list (revocation)

Credential Issuer

Trust in cryptographically authenticated identifier

⓪ User tries to get access to a resource

① **RP requests** ID Token and Credential(s)

② **SIOP** issues ID Token & Verifiable Presentation(s)

Alice

OP

User Agent

Website or App (RP)

Stored Verifiable Credentials

Credentials Presentation (Key & New Features)

- Protocol is credential/presentation format agnostic NEW
  - Examples for AnonCreds and mDL in OIDC4VP spec
- passing `presentation_definition` PE object by value or by reference NEW
- Support for Trust Schemes NEW
  - for example, request credentials issued by an issuer that is part of a Trust Framework
- Dynamic SIOP discovery and invocation via HTTPS URLs
  - enables use of app/universal links and web wallets
- Leverages all OpenID Connect Flows NEW
  - SIOP can be entirely locally hosted, have cloud components, be entirely cloud-based
- Cross Device Flow enabled
- Leverages OpenID Connect Metadata for verifiers and wallet management
- Clarify that the key feature of SIOP is ability to sign ID Token using a subject-controlled key material (iss==sub in ID Token)
- Ongoing: wallet & key attestation

Credential Presentation (Status)

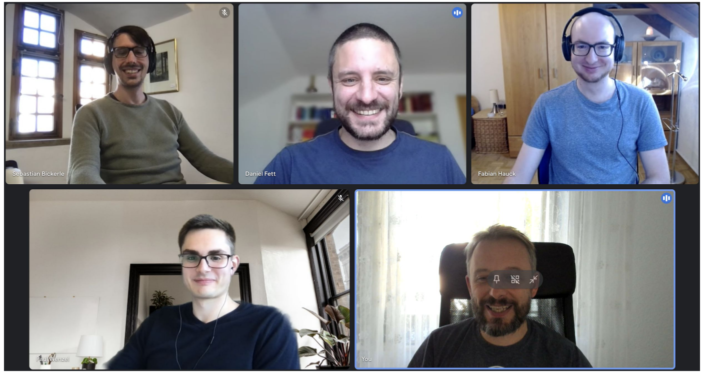
- First Implementer's Drafts of OpenID Connect SIOPV2 and OIDC4VP approved. Targeting Second Implementer's Draft by the end of 2022
- Latest Editor's drafts can be published at:
  - https://openid.net/specs/openid-connect-self-issued-v2-1_0.html
  - https://openid.net/specs/openid-connect-4-verifiable-presentations-1_0.html
- Existing & ongoing Implementations:
  - The European Blockchain Services Infrastructure (EBSI)
  - Microsoft
  - Workday
  - Ping Identity
  - Convergence.Tech
  - IDunion
  - walt.id (eSSIF-Lab)*
  - Sphereon
  - Gimly

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

Demo
Credential Presentation

IDunion 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 & AnonCreds



iDunion

Supported by:

Federal Ministry
for Economic Affairs
and Climate Action

on the basis of a decision
by the German Bundestag

European Banking Identity Prototype

- eSSIF-Lab founded project
- Team: yes.com & walt.id
- Presentation & Issuance via OIDC4SSI
- Based on
  - walt.id Wallet (Web Wallet)
  - JSON LD based credentials
  - did:key (did:ebsi)

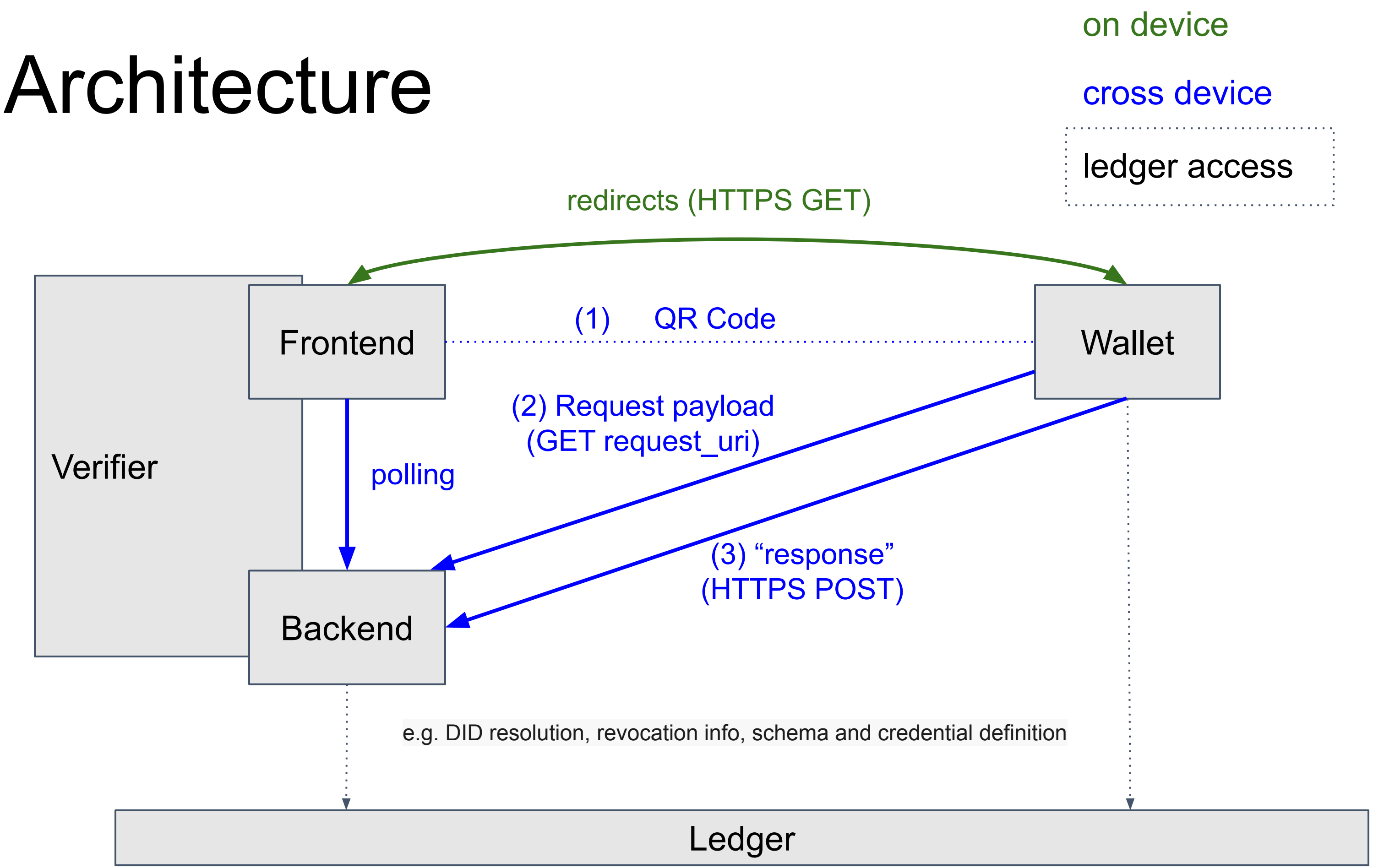
walt.id

yes®

NGI ESSIF-LAB

eSSIF-Lab is funded by the European Commission, as part of the Horizon 2020 Research and Innovation Programme, under Grant Agreement Nº 871932 and it's framed under Next Generation Internet Initiative.

Architecture



Request Example ESSIF Lab (W3C VC)

```
{
   "response_type":"id_token",
   "client_id":"https://example.com/callback",
   "scope":"openid",
   "redirect_uri":"https://example.com/callback",
   "nonce":"67473895393019470130",
   ...
   "claims":{
      "vp_token":{
         "presentation_definition":{
            "id":"1",
            "input_descriptors":[
               {
                  "id":"1",
                  "schema":{
                    "uri":"https://raw.githubusercontent.com/…/EuropeanBankIdentity.json"
                   }
               }
            ]
         }
      }
   }
}
```

Response Example ESSIF Lab (W3C VC)

ID Token

```
{
 "iss": "https://self-issued.me/v2",
 "aud": "https://example.com/callback",
 "sub": "did:key:z6MkqUDiu3MHxAm...mscLT8E9R5CKdbtr7gwR8",
 "exp": 1645469476,
 "iat": 1645465876,
 "nonce": "cdb97870-a3be-49b4-aa55-8c7c7122178a",
 "_vp_token": {
   "presentation_submission": {
     "descriptor_map": [
       {
         "path": "$",
         "format": "ldp_vp",
         "path nested": {
           "path": "$.verifiableCredential[0]",
           "format": "ldp_vc"
       }
     ],
     "definition_id": "1",
     "id": "1"
   }
 }
}
```

VP Token

```
{
  "@context":[
     "https://www.w3.org/2018/credentials/v1"
  ],
  "holder":"did:key:z6MkqUDiu3MHxAmuMQ8jjkLiUu1mscLT8E9R5CKdbtr7gwR8"  ,
  "id":"urn:uuid:04816f2a-85f1-45d7-a66d-51764d39a569"  ,
  "proof":{
     "domain":"https://example.com/callback" ,
     "jws":"...",
     "nonce":"cdb97870-a3be-49b4-aa55-8c7c7122178a" ,
     "proofPurpose":"authentication",
     "type":"Ed25519Signature2018" ,
     "verificationMethod" :"did:key:z6MkqUDiu3 ..."
  },
  "type":[
     "VerifiablePresentation"
  ],
  "verifiableCredential" :[
     {
       …
       "type":[
          "VerifiableCredential" ,
          "EuropeanBankIdentity"
       ],
       "credentialSubject" :{
          "id":"did:key:z6MkqUDiu3MHxAmuMQ8jjkLiUu1mscLT8E9R5CKdbtr7gwR8"  ,
          "familyName":"Family001",
          "givenName":"Given001",
          "birthDate":"1950-01-01",
          "placeOfBirth" :{
             "country":"DE",
             "locality":"Berlin"
          }
       },
```

Request Example IDunion (AnonCred)

```
{
  "response type" :"id token",
  "client id" :"https://example.com/callback" ,
  "scope" :"openid",
  "redirect uri" :"https://example.com/callback ",
  "nonce" :"6747389539301947O130" ,
  ...
  "claims" :{
     "vp token" :{
        "presentation definition" :{
           "id":"NextcloudLogin" ,
           "input_descriptors" :[
              {
                 "id":"ref2",
                 "name":"NextcloudCredential" ,
                 "format": {
                   "ac_vc": {
                       "proof_type": ["CLSignature2019" ]
                   }
                 },
                 "constraints" :{
                    "limit disclosure" :"required",
                    "fields":[{
                               "path": [
                                    "$.schema_id"
                               ],
                               "filter": {
                                   "type":   "string",
                                   "pattern":   "did:indy:idu:test:3QowxFtwciWceMFr7WbwnM:2:BasicScheme:0\\.1"
                               }
                              },
                              {"path":["$.values.email" ]},
                              { "path":["$.values.first name" ]},
                              { "path":["$.values.last_name" ]}]
                 }
              }
```

Response Example IDunion (AnonCred)

ID Token

```
{
  "aud": "https://example.com/callback ",
  "sub": "9wgU5CR6PdgGmvBfgz_CqAtBxJ33ckMEwvij-gC6Bcw" ,
  "auth time": 1638483344,
  "iss": "https://self-issued.me/v2" ,
  "sub jwk": {
    "x": "cQ5fu5VmG…dA_5lTMGcoyQE78RrqQ6" ,
    "kty": "EC",
    "y": "XHpi27YMA…rnF_-f_ASULPTmUmTS" ,
    "crv": "P-384"
  },
  "exp": 1638483944,
  "iat": 1638483344,
  "nonce": "6747389539301947013O ",
  " vp token": {
    "presentation submission" : {
      "descriptor_map" : [
        {
          "id": "ref2",
          "path": "$",
          "format": "ac vp",
          "path nested" : {
            "path":
"$.requested proof.revealed attr_groups.ref2",
            "format": "ac_vc"
          }
        }
      ],
      "definition id" : "NextcloudLogin" ,
      "id": "NexcloudCredentialPresentationSubmission"
    }
  }
}
```

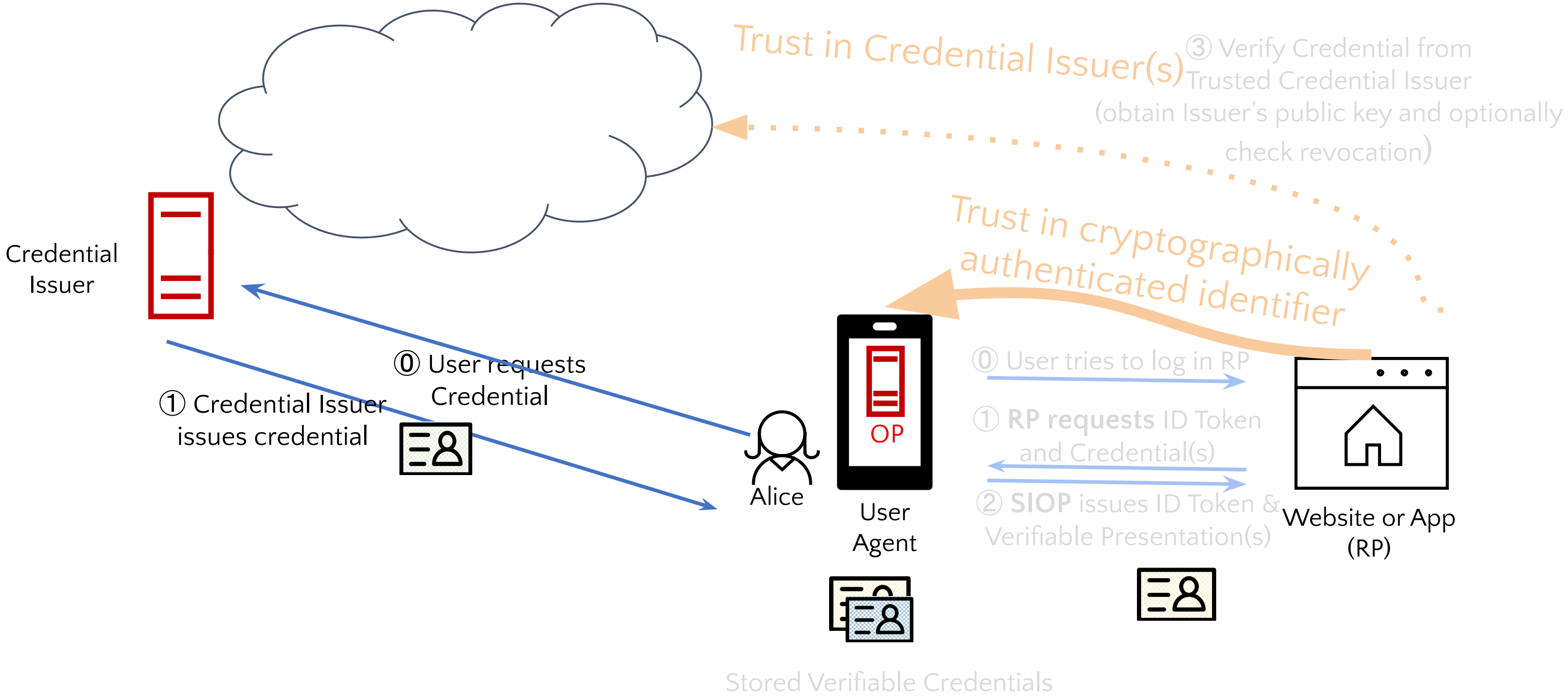
VP Token

```
{
   "proof": {...},
   "requested_proof": {
      "revealed attrs" : {},
      "revealed_attr_groups": {
         "ref2": {
            "sub proof index" : 0,
            "values": {
               "email": {
                  "raw": "alice@example.com" ,
                  "encoded": "115589951…83915671017846"
               },
               "last name" : {
                  "raw": "Wonderland",
                  "encoded": "167908493…94017654562035"
               },
               "first name" : {
                  "raw": "Alice",
                  "encoded": "270346400…99344178781507"
               }
            }
         }
      },
      …
   },
   "identifiers": [
      {
         "schema id" : "3QowxFtwciWceMFr7WbwnM:2:BasicScheme:0.1" ,
         "cred def id" : "CsiDLAiFkQb9N4NDJKUagd:3:CL:4687:awesome_cred" ,
         "rev reg id" : null,
         "timestamp" : null
      }
   ]
}
```

OpenID for Credential Issuance

OpenID Connect 4 Verifiable Credentials Issuance

Issuing Credentials


Trust in Credential Issuer(s)
③ Verify Credential from Trusted Credential Issuer (obtain Issuer's public key and optionally check revocation)
Trust in cryptographically authenticated identifier
Credential Issuer
⓪ User requests Credential
① Credential Issuer issues credential
Alice
OP
User Agent
⓪ User tries to log in RP
① RP requests ID Token and Credential(s)
② SIOP issues ID Token & Verifiable Presentation(s)
Website or App (RP)
Stored Verifiable Credentials

Design Principles

- Issuance via OAuth-protected Credential Endpoint
- Currently two authorization flows:
    - Code flow (others possible)
        - invoked by Wallet requesting authorization for one or more credentials at the Authorization Endpoint (may trigger by presentation request during the issuance)
        - Issuer takes screen control and can authenticate/identify user with means at Issuer’s discretion
    - Pre-authorized code flow (new grant type)
        - Wallet is invoked after completion of process with the Issuer (QR Code or redirect)

Credential Issuance (Key Features)

- Protocol is credential format agnostic
  - W3C Verifiable Credentials, ISO mobile Driving Licence/electronic ID, SMART Health Cards
- Can be customized to use different methods for proofs of possession of key material
  - for example, `jwt` proof type that includes a signature by a key material tied to a DID
- Allows Credential Issuance during Presentation Request (inline issuance)
  - Requested credential not found in the wallet
- Allows just-in-time and batch credential issuance as well as credential refresh
- Allows Presentation Request during Credential Issuance
  - Issuer is requesting to present a VC as a way to identify a user during Issuance
- Can be built on top of existing OAuth/OpenID implementations
- Leverages OpenID Connect Metadata for wallet & issuer management
- Ongoing: wallet & key attestation to build Issuer’s trust in the wallet

Credential Issuance (Status)

- Specification adopted by the working group. Targeting First Implementer’s draft by the end of 2022.
  - https://openid.net/specs/openid-connect-4-verifiable-credential-issuance-1_0.html
- Planned and ongoing implementations:
  - The European Blockchain Services Infrastructure (EBSI)
  - Microsoft
  - Mattr
  - IDunion
  - walt.id & yes.com & BCDiploma (eSSIF-Lab)
  - Sphereon
  - Talao.io
  - Convergence.Tech

Demo
Credential Issuance

European Banking Identity Prototype

- eSSIF-Lab founded project
- Team: yes.com & walt.id
- Presentation & Issuance via OIDC4SSI
- Based on
  - walt.id Wallet (Web Wallet)
  - JSON LD based credentials
  - did:key (did:ebsi)

walt.id

yes®

NGI ESSIF-LAB

eSSIF-Lab is funded by the European Commission, as part of the Horizon 2020 Research and Innovation Programme, under Grant Agreement Nº 871932 and it's framed under Next Generation Internet Initiative.

Authorization Request

```
HTTP/1.1 302 Found
Location: https://server.example.com/authorize?
  response_type=code
  &client_id=s6BhdRkqt3
  &code_challenge=E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM
  &code_challenge_method=S256
  &scope=openid credential:ttps://…/EuropeanBankIdentity.json
  &redirect_uri=https://client.example.org/cb
```

Credential Issuance (W3C VC)

Request

```
POST /credential HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: BEARER czZCaGRSa3F0MzpnWDFmQmF0M2JW

type=https://…/EuropeanBankIdentity.json
format=ldp_vc
did=did:key:z6MkqUDiu3MHxAmuMQ8jjkLiUu1mscLT8E9R5CKdbtr7gwR8
proof=%7B%22type%22:%22jwt%22…0aW9EkL1nOzM%22%7D
```

Response

```
HTTP/1.1 200 OK
  Content-Type: application/json
  Cache-Control: no-store
  Pragma: no-cache

{
   "format": "ldp_vc",
   "credential" : "eyJjcmVkZW50a...d0MifQ=="
}
```

Issued Credential

```
{
   ...
   "issuer": "did:key:z6MkgF2pvVNEFXCksupWKrdPhL6ubecis3AWbWVsr9bNAbwC",
   "type": [
       "VerifiableCredential",
       "EuropeanBankIdentity"
   ],
   "credentialSchema": {
       "id": "https://raw.githubusercontent.com/…/EuropeanBankIdentity.json",
   },
   "credentialSubject": {
       "placeOfBirth": {
           "country": "DE",
           "locality": "Berlin"
       },
       "familyName": "Family001",
       "givenName": "Given001",
       "id": "did:key:z6MkmY9NFeyqNTS6nYN1tSeuxg6Sbxi7ntt2wR4Upy9HHSDS",
       "birthDate": "1950-01-01"
   }
   ...
}
```

Demo 2

- Interoperability profile relying on SIOP v2 and OIDC4VP
  - Microsoft
  - Workday
  - Ping Identity
  - (Mattr)
  - (IBM)

OIDC4SSI allows variety of choices in the SSI tech stack

Using OIDC4SSI as an authentication protocol to present and issue credentials allows implementers to choose a combination of DID methods, credential formats and other components of the SSI tech stack.

SSI Tech Stack component	Implementer’s choices when using OIDC4SSI as a protocol
Identifiers	**Any DID method** - user’s identifier can also be a JWK Thumbprint (`sub` in the ID Token) - verifier’s identifier can also be a unique string (`client_id` in the request)
Credential Format	**Any credential format** (AnonCreds, LDP-VC, JWT-VC, ISO mDL, etc.)
Revocation	**Any mechanism** (Status List 2021, etc.)
additional trust mechanisms	**Any mechanism** (.well-known DID configuration, etc.)
Cryptography	**Any cryptosuite** (EdDSA, ES256K, etc.)

Announcements

- OIDF Slack channel **#wg-connect**

Q&A