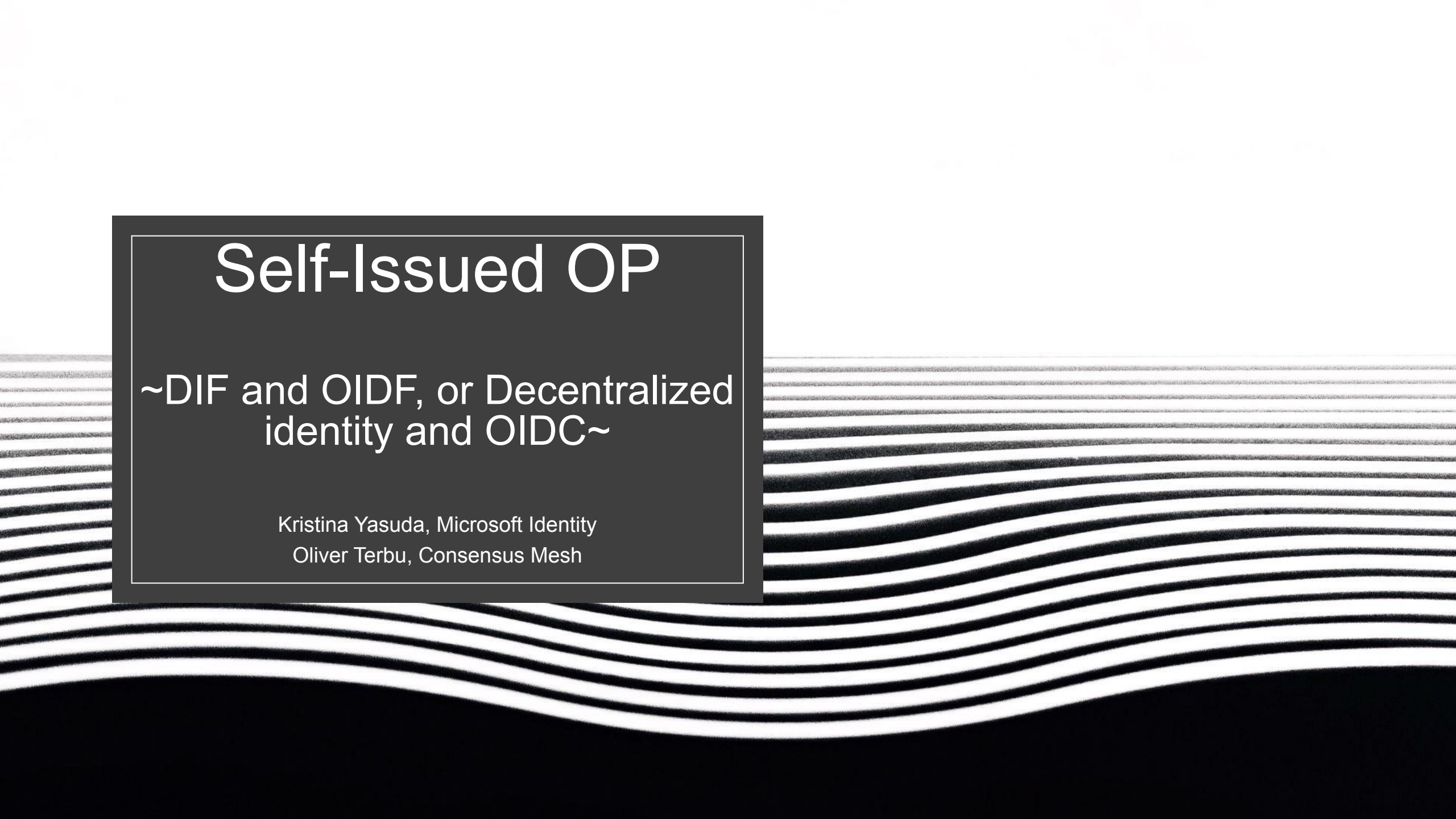
Self-Issued OP

~DIF and OIDF, or Decentralized identity and OIDC~

Kristina Yasuda, Microsoft Identity
Oliver Terbu, Consensus Mesh



1. Three work-streams
    i. Self-Issued OpenID Provider (SIOP V2)
    ii. Presentation of W3C verifiable credentials using OIDC
    iii. Issuance of aggregated/client-bound claims from Claims Providers
2. Use-cases

1-i. Self-Issued OpenID Provider (https://bitbucket.org/openid/connect/src/master/openid-connect-self-issued-v2-1_0.md)

Specific model where users control their own OpenID Providers - extension of Chapter 7

Issues raised

- Different Trust Model between Self-Issued OP and RP from that of the rest of OIDC?
  - Ad Hoc Registration is proposed
- Need to communicate inf about SIOP's provider? iss=self-issued.me
- consent, etc.
- will deep-dive on two

Self-Issued OpenID Provider V2, draft 01

Kristina Yasuda, Microsoft, kristina.yasuda@microsoft.com

Michael B. Jones, Microsoft, mbj@microsoft.com

Tobias Looker, Mattr, tobias.looker@mattr.global

December 8, 2020

1-i. SIOP V2 Issues progress (1/2)

- Which SIOP/wallet under user's possession to invoke? following options:
  - 1. SIOP Chooser (https://bitbucket.org/openid/connect/issues/1212/siop-chooser)
    - a combination of 1/ a list of wallets (maintained by the trust framework); 2/ universal links to open wallet from the browser; and 3/ share sheet to choose between several wallets under the user's control.
    - a current best solution that will work with different kind of wallets - native apps, PWAs, browser wallets.
  - 2. Each wallet pre-registering custom URL schema with RP
    - NASCAR problem remains

Not the ideal solution, but the most viable without OS vendor's collaboration.

1-i. SIOP V2 Issues progress (1/2)

- Need for a user to prove control over the Self-Issued OP
    - in addition to jwk thumbprint, allow DIDs to be used as holder identifier by checking if ID Token is signed by the keys in the DIDDoc controlled by the user - benefit of a key rotation

      3.2. Self-Issued OpenID Provider Response
      sub
      REQUIRED. Subject identifier value, represented by a URI. When sub type is jkt, the value is the base64url encoded representation of the thumbprint of the key in the sub_jwk Claim. When sub type is did, the value is a decentralized identifier.

1-ii. Presentation of W3C verifiable credentials using OIDC

- Support request and presentation of Verifiable Credentials in ID Tokens and Userinfo responses
- Usable with all OpenID Connect Flows (SIOP, code, CIBA, …)
- Leverage OpenID Connect as simple to use protocol for wallet integrations
- Leverage W3C verifiable credentials to existing OpenID Connect deployments

Current Spec work

- Request
  - via “claims” parameter
  - Simply claims or credential type or credential type + claims (selective disclosure)
- Working on a draft that allows for both options to gather implementation feedback with a goal of making a decision on which option to adopt
  - A) Embedding entire VP/VC in any format
    - https://github.com/Sakurann/vp-token-spec
    - ease of adoption in existing implementations
  - B) VP Token as separate artifact returned alongside ID Token from the authorization endpoint
    - https://github.com/awoie/vp-token-spec
    - ‘clean’ technical solution

→ So that VPs are returned using same syntax in both options, will also define generic container to convey VPs - something like an array with objects containing a format identifier and the actual payload (+ potentially some additional metadata).

Will be contributed to the WG & call for adoption in coming week

A. vp_jwt Claim

```
{
   "id_token":{
      "acr":null,
      "vp_jwt":{
         "credential_types":[
            "https://www.w3.org/2018/cre
         ]
      }
   }
}
```

```
{
  "kid": "did:ion:EiC6Y9_aDaCsITlY06HId4seJjJ...b1df31ec42d0",
  "typ": "JWT",
  "alg": "ES256K"
}.{
   "iss":"https://self-issued.me",
   "aud":"https://book.itsourweb.org:3000/client_api/authresp/uhn",
   "iat":1615910538,
   "exp":1615911138,
   "sub":"did:ion:EiC6Y9_aDaCsITlY06HId4seJjJ-9...mS3NBIn19",
   "auth_time":1615910535,
   "nonce":"960848874",
   "vp_jwt":[
            "ewogICAgImlzcyI6Imh0dHBzOi8vYm9vay5pdHNvdXJ3ZWIub...IH0="
   ],
   "sub_jwk":{
      "crv":"P-384",
      "kty":"EC",
      "kid": "c7298a61a6904426a580b1df31ec42d0",
      "x":"jf3a6dquclZ4PJ0JMU8RuucG9T1O3hpU_S_79sHQi7VZBD9e2VKXPts9lUjaytBm",
      "y":"38VlVE3kNiMEjklFe4Wo4DqdTKkFbK6QrmZf77lCMN2x9bENZoGF2EYFiBsOsnq0"
   }
}
```

parameters of ID Token

A. vp_ldp Claim

```
{
    "id_token": {
      "vp_ldp": {
        "credential
         "claims":
        {
          "given_na
          "family_n
          "birthdat
        }
      }
    }
}
```

```
{
   "iss":"https://self-issued.me",
   "aud":"https://book.itsourweb.org:3000/client_api/authresp/uhn",
   "iat":1615910538,
   "exp":1615911138,
   "sub":"did:ion:EiC6Y9_aDaCsITlY06HId4seJjJ...b1df31ec42d0",
   "auth_time":1615910535,
   "vp_ldp":[
    {
     "@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":{
```

parameters of ID Token

B. Separate artifact - ‘VP Token’

`claims` parameter in the request

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

```
{
   "iss":"https://book.itsourweb.org:3000/wallet/wallet.html",
   "aud":"https://book.itsourweb.org:3000/client_api/authresp/uhn",
   "iat":1615910538,
   "exp":1615911138,
   "sub":"urn:uuid:68f874e2-377c-437f-a447-b304967ca351",
   "auth_time":1615910535,
   "vp_hash":"77QmUPtjPfzWtF2AnpK9RQ",
   "nonce":"960848874",
   "sub_jwk":{
```

ID Token contains a `vp_hash`

vp_token content

‘VP Token’ contains an entire VP

```
{
  "@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"
      ],
```

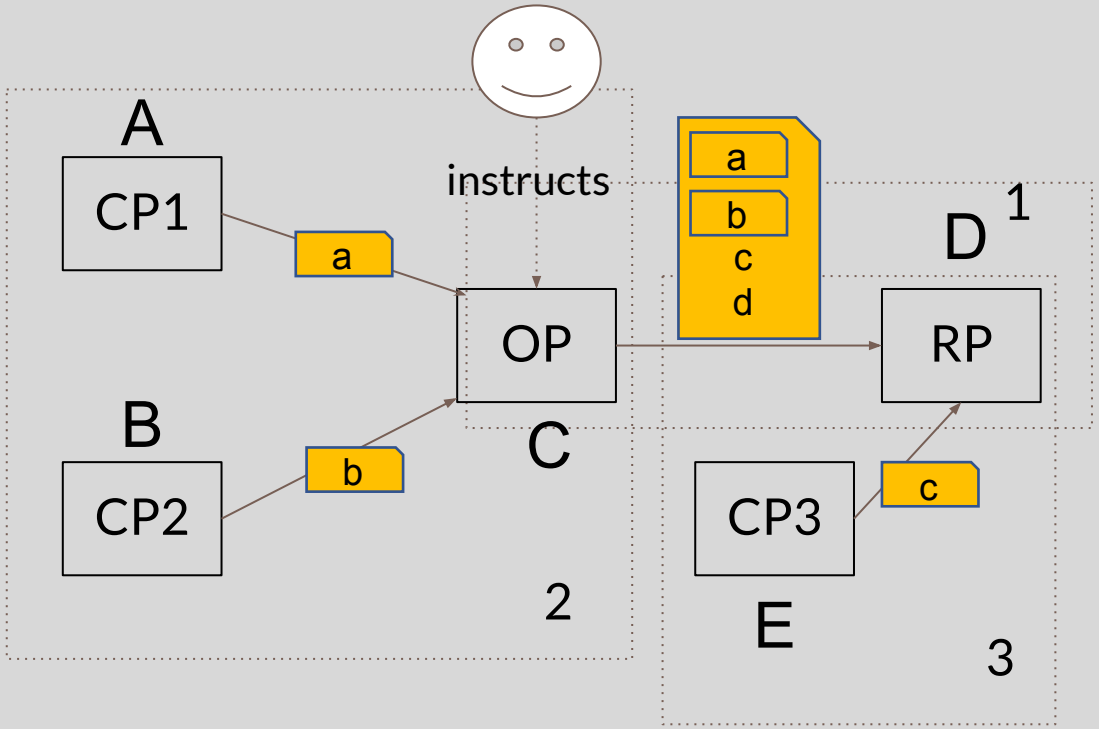
1-iii. Issuance of aggregated/client-bound claims from Claims

Specify the methods for an application to:

- perform discovery for a Claims Provider
- register a client to a Claims Provider
- obtain claims from the Claims Provider
- return aggregated claims from Claims Providers to requesting clients

OpenID Connect has 3 claims models

1. Simple Claims
2. Aggregated Claims
3. Distributed Claims

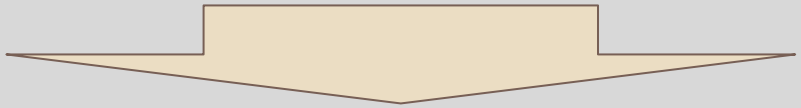


- C acts as an OP to D which is an RP in this context
- A&B acts as an OP to D which is an RP in this context
- E acts as a resource to D

Note: Credit to Nat Sakimura

Weakness of the Connect Core defined aggregated claims

- How to get a token from CP is hand-wavy.
- No specified method to down scope the userinfo of the CP.
- No way to provide a binding information between CP:sub and IdP:sub.

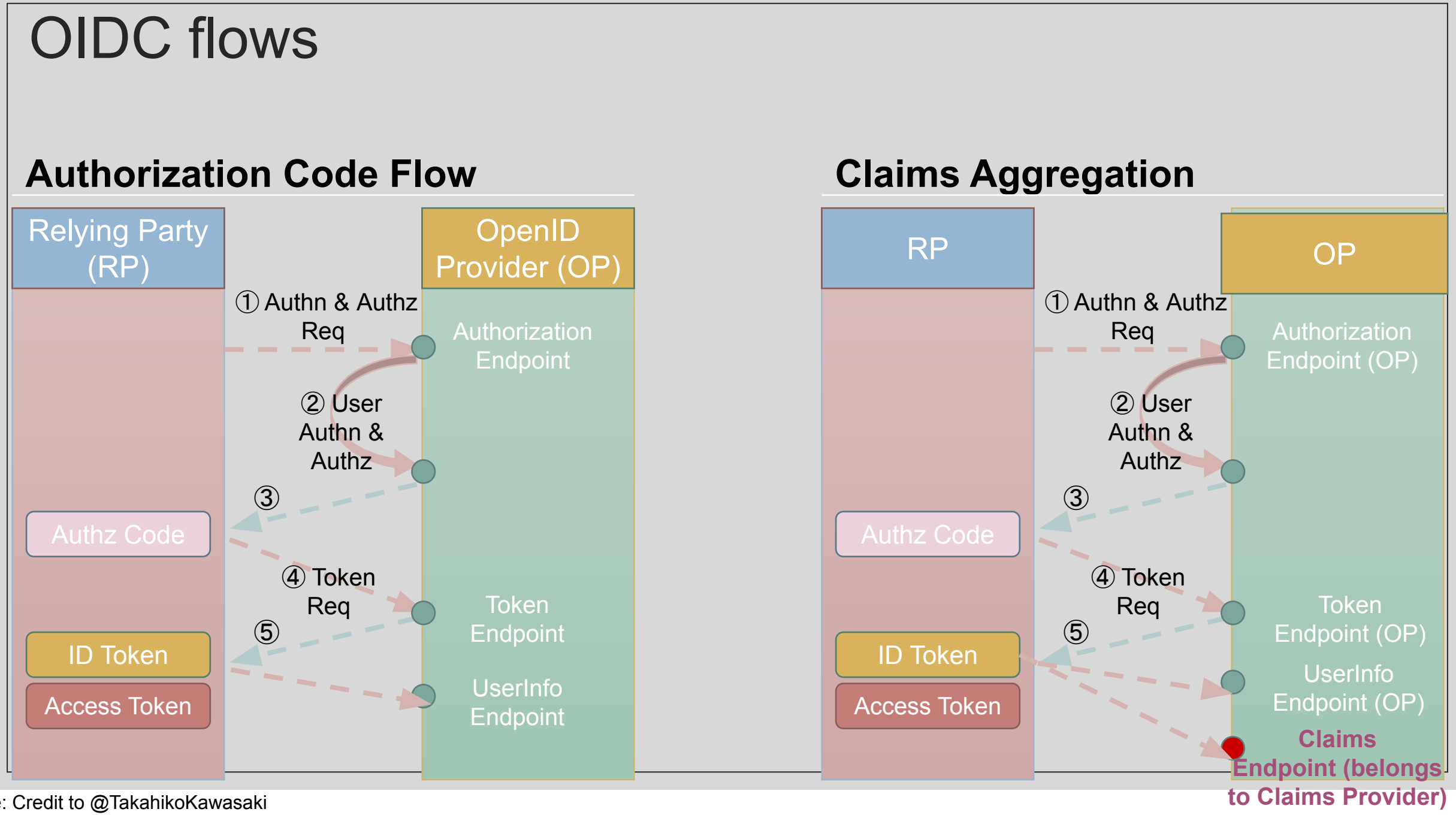


OIDC Claims aggregation draft (WG adopted, issues filled in)
https://bitbucket.org/openid/connect/src/master/openid-connect-claims-aggregation/openid-connect-claims-aggregation-1_0.md

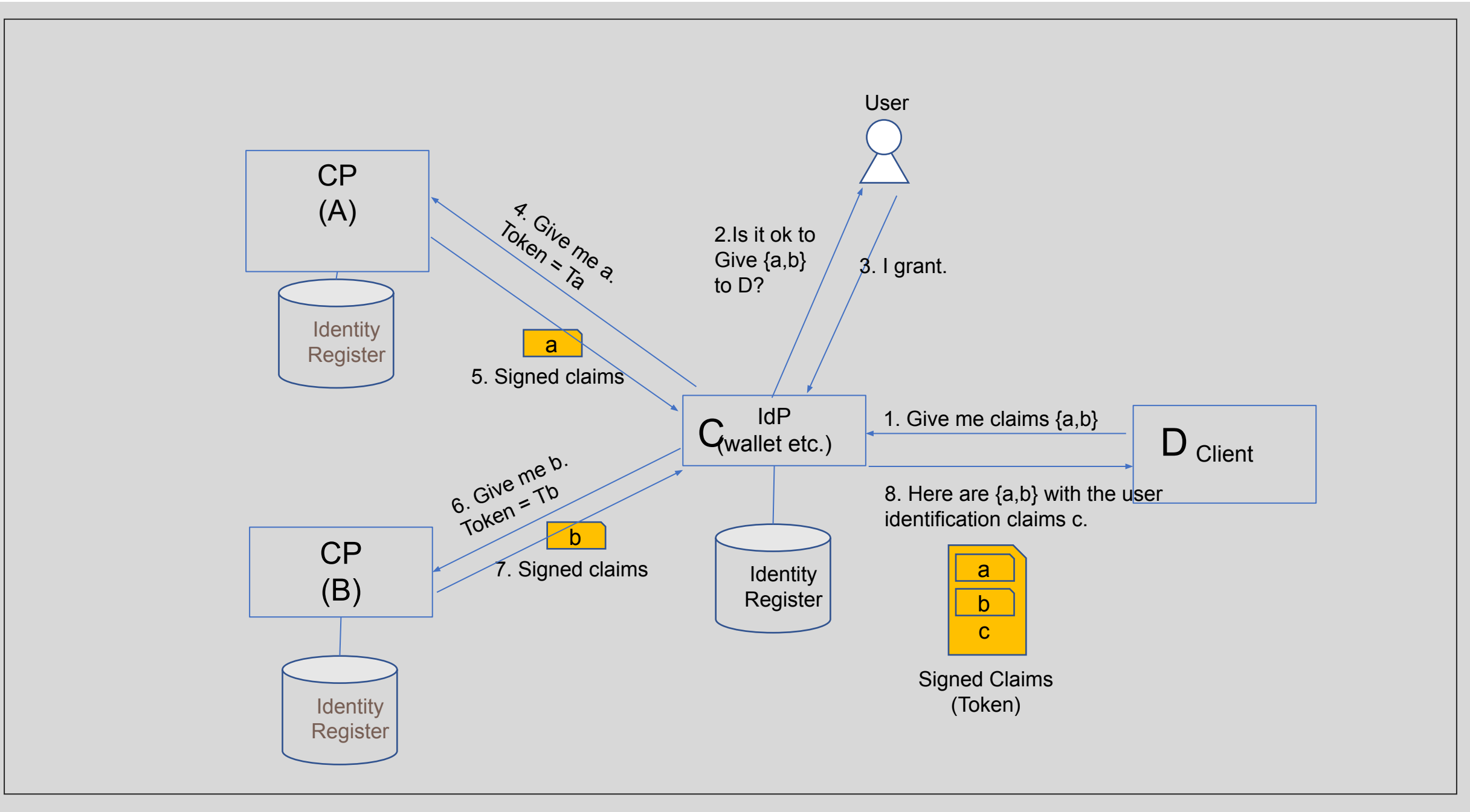
(Discussions to converge with Credential Provider draft - to be contributed)
https://github.com/mattrglobal/oidc-client-bound-assertions-spec

Note: Credit to Nat Sakimura

OIDC flows



Note: Credit to @TakahikoKawasaki



Note: Credit to Nat Sakimura

2. Use-cases

User’s having OPs that they control; users being able to receive and present verifiable credentials

-> “What problem it solves that current technology does not solve”

- Privacy preservation - no issuer call home at presentation.
  - mDL (mobile Driving License defined as ISO/IEC 18013-5)
- Addressing issuers-ceased-to-exist use case.
  - University issues student cards for alumni, which alumni can use regardless of university existence. (also cost saving because university potentially does not have to maintain alumni records in the registry) -> Keio Universty
- Claims Aggregation & User-consent
  - NHS verifying doctors' eligibility using digital claims from several sources and saving patient treating time
- Also remote onboarding, getting app access and self-service recovery
- Other use-cases?

- Weekly SIOP Special Topic Calls
  - Alternating Pacific and Atlantic time-zone calls
- OIDC AB/Connect WG calls
  - Weekly Pacific time-zone calls and
  - Bi-weekly Atlantic time-zone calls

+ Bitbucket issues, PRs ☺