OpenID Connect Federation

Hur man bygger multilaterala federationer med OIDC

Roland Hedberg, 16 februari 2022

Kort om OpenID Connect

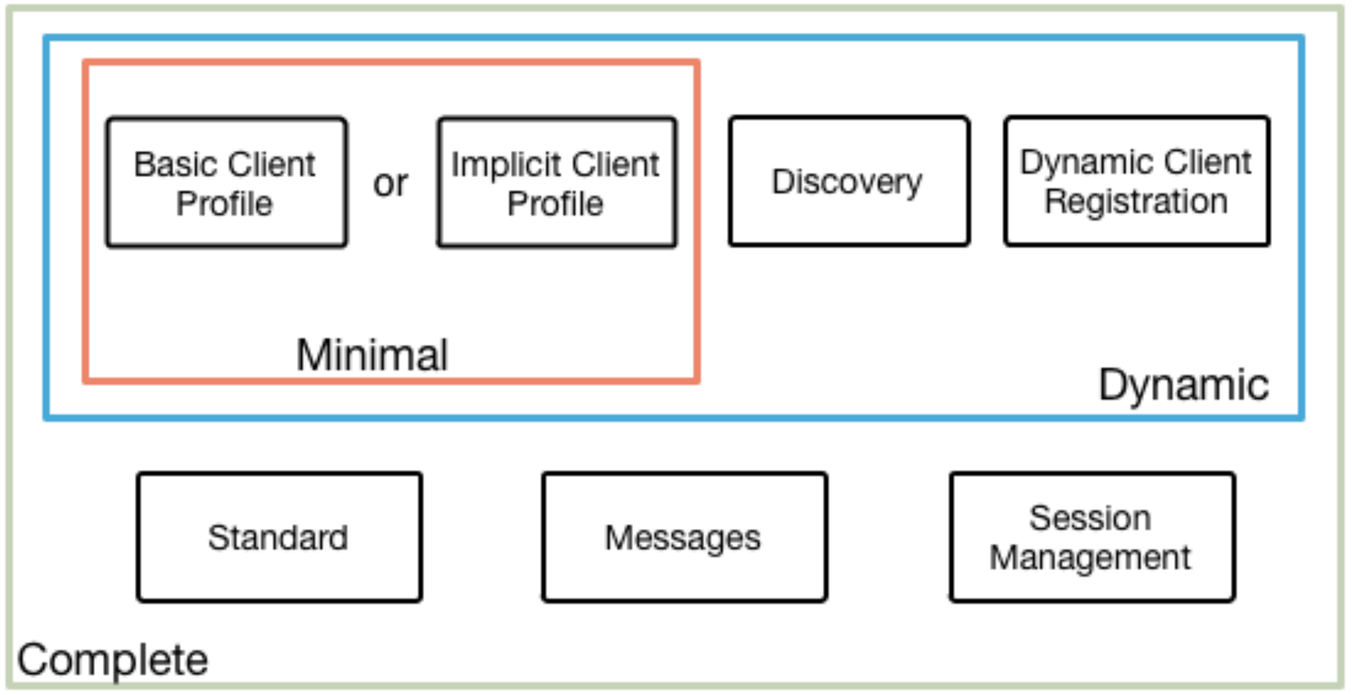
Standard texter

- OIDC Specifikationerna ägs av OpenID Foundation (November 2014).
  - OpenID Connect Core
  - OpenID Connect Discovery
  - OpenID Connect Dynamic Client Registration

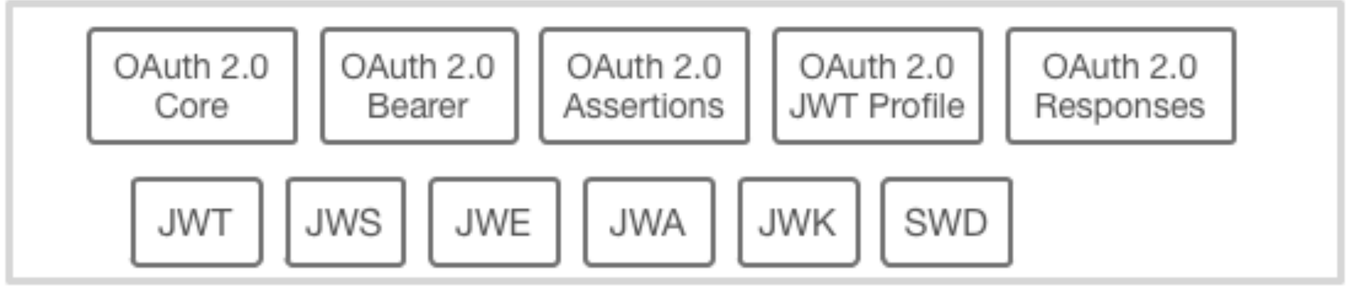
25 May 2012

OpenID Connect Protocol Suite

http://openid.net/connect



Underpinnings



OIDC grundbultar

- HTTPS POST & GET
- JSON/URL-kodat
- JSON Web Token (JWT) RFC 7519 (May 2015)
- JSON Web Signature (JWS) RFC 7515
- ID Token

OIDC tjänster

1. Provider discovery
2. Dynamic client registration
3. Authorization/Authentication
4. Access Token/Refresh Token
5. Userinfo

Fördelar med OIDC

- Enklare att implementera
- Bättre signerings-/krypteringssystem
- Bättre implementationsvalidering (OIDF test suite sedan Jan 2015)

OIDC Federation

Delar som påverkas

1. Provider discovery
2. Dynamic client registration
3. Authorization/Authentication
4. Access Token/Refresh Token
5. Userinfo

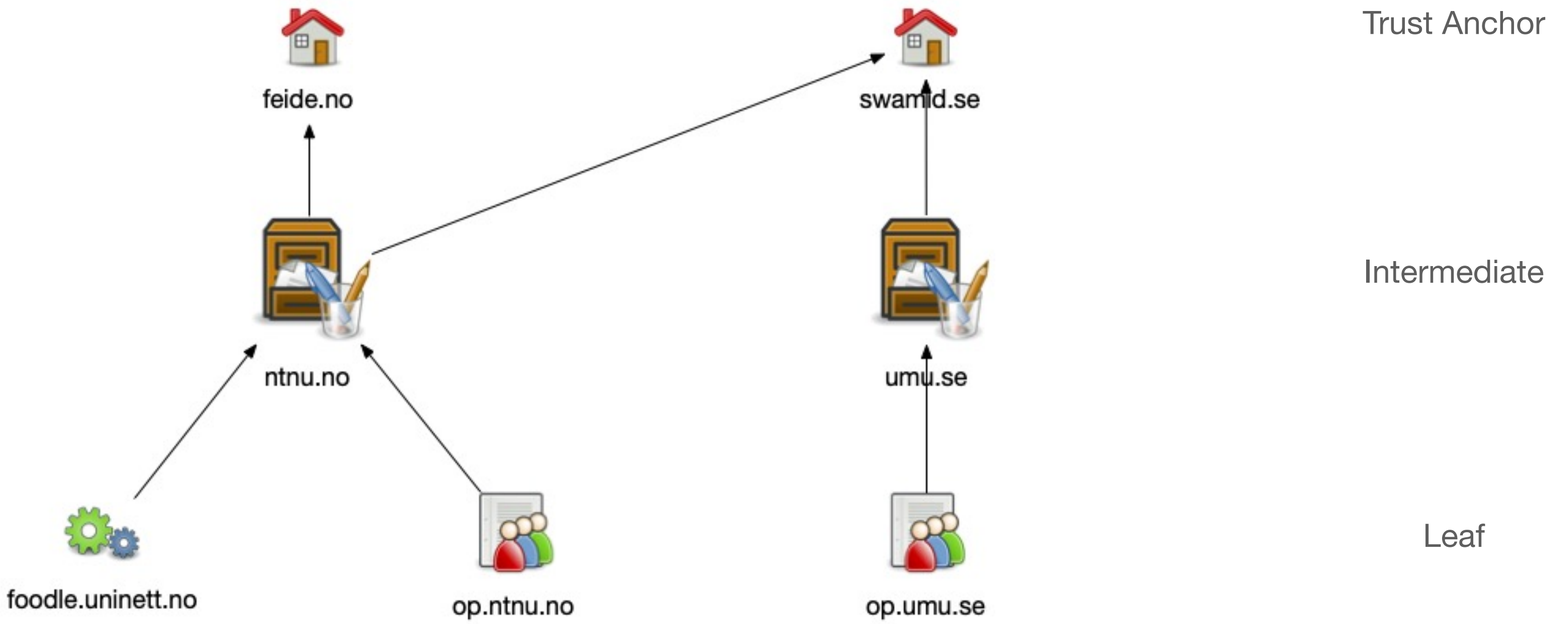
Verktygslåda

Betrodd information

- Korrekt
  - Motstå modifiering
  - Regler för metadata
- Baserad på en betrodd tredje part (trust anchor)
- Uttryck som en kopplad kedja
- Hämta när man behöver

Förtroendekedja (trust chain)

En sekvens av ’entity statements’ som börjar med ett ‘löv’ och slutar i ett ’trust anchor’.



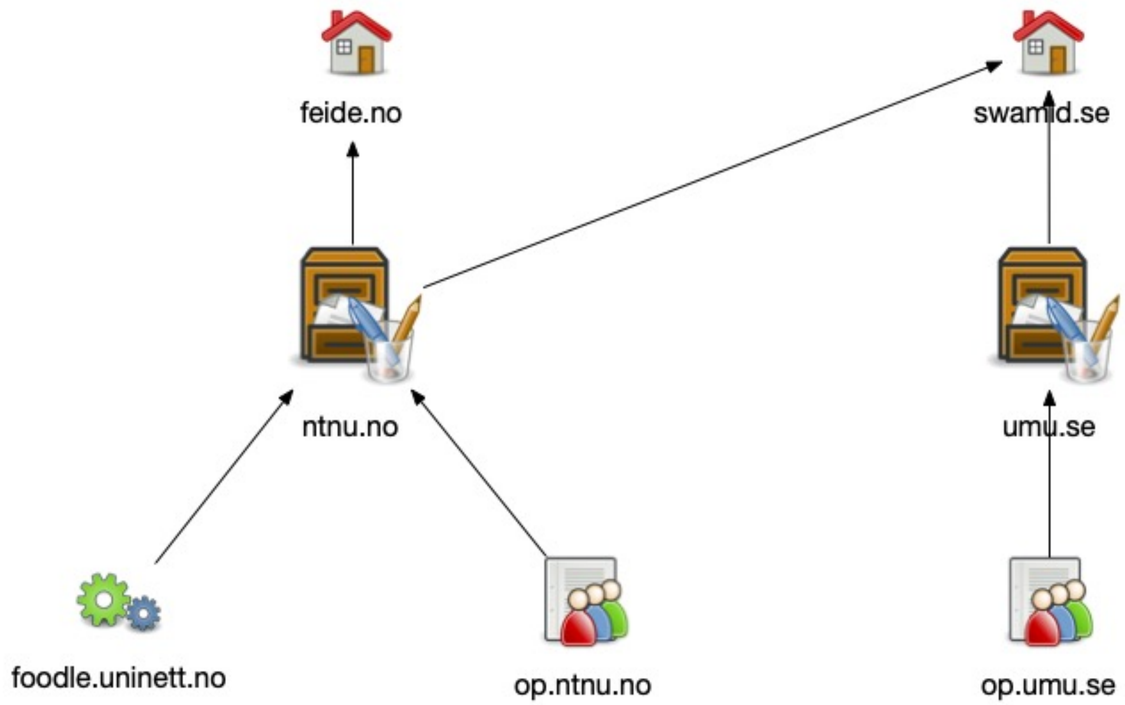
Entity Statement

- Information om en enhet
- Kan vara en enhets syn på sig själv (self-signed entity statement/entity configuration) eller en överordnads syn på en underordnad.
- Signerad JWT

```
{
  "iss": ”https://rp.umu.se”,
  "sub": ”https://rp.umu.se”,
  "iat": 1516239022,
  "exp": 1516298022,
  "metadata": {
    "openid_relying_party": {
      "application_type": "web",
      "redirect_uris": [
        ”https://rp.umu.se/rp/callback”
      ],
      "grant_types": [
        "authorization_code",
        "implicit"
      ],
      "jwks_uri": ”https://rp.umu.se/static/jwks.json”
    }
  },
  "jwks": {
    "keys": [
      {
        "kid": "key1",
        "kty": "RSA",
        "use": "sig"
        "e": "AQAB",
        "n": "pnXBOusEANuug6ewezb9J_...",
      }
    ]
  },
  "authority_hints": [
    ”https://federation.umu.se”
  ]
}
```

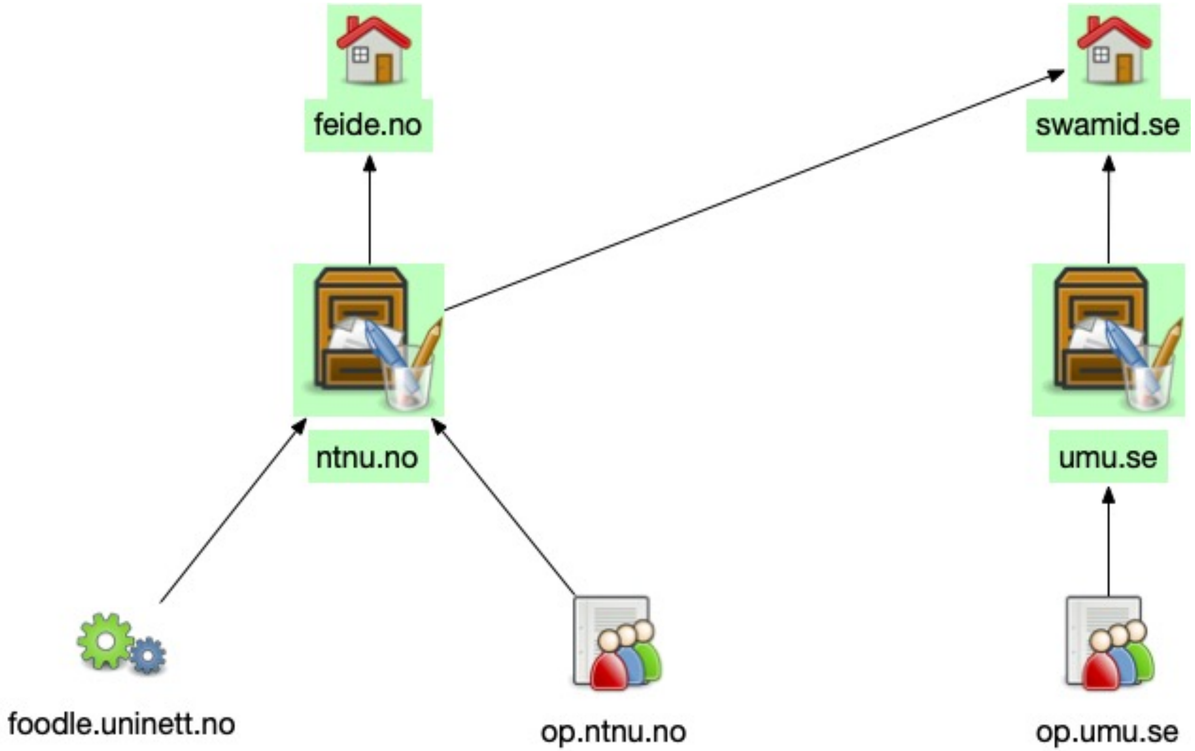
Entitetskonfiguration

- Alla enheter i en federation skall publicera information om sig själv på en välkänd ändpunkt.
- Alla enheter har en identifierare (entity_id, URL)
- Well known URI, RFC 5785/8615 (.well-known/openid-federation)

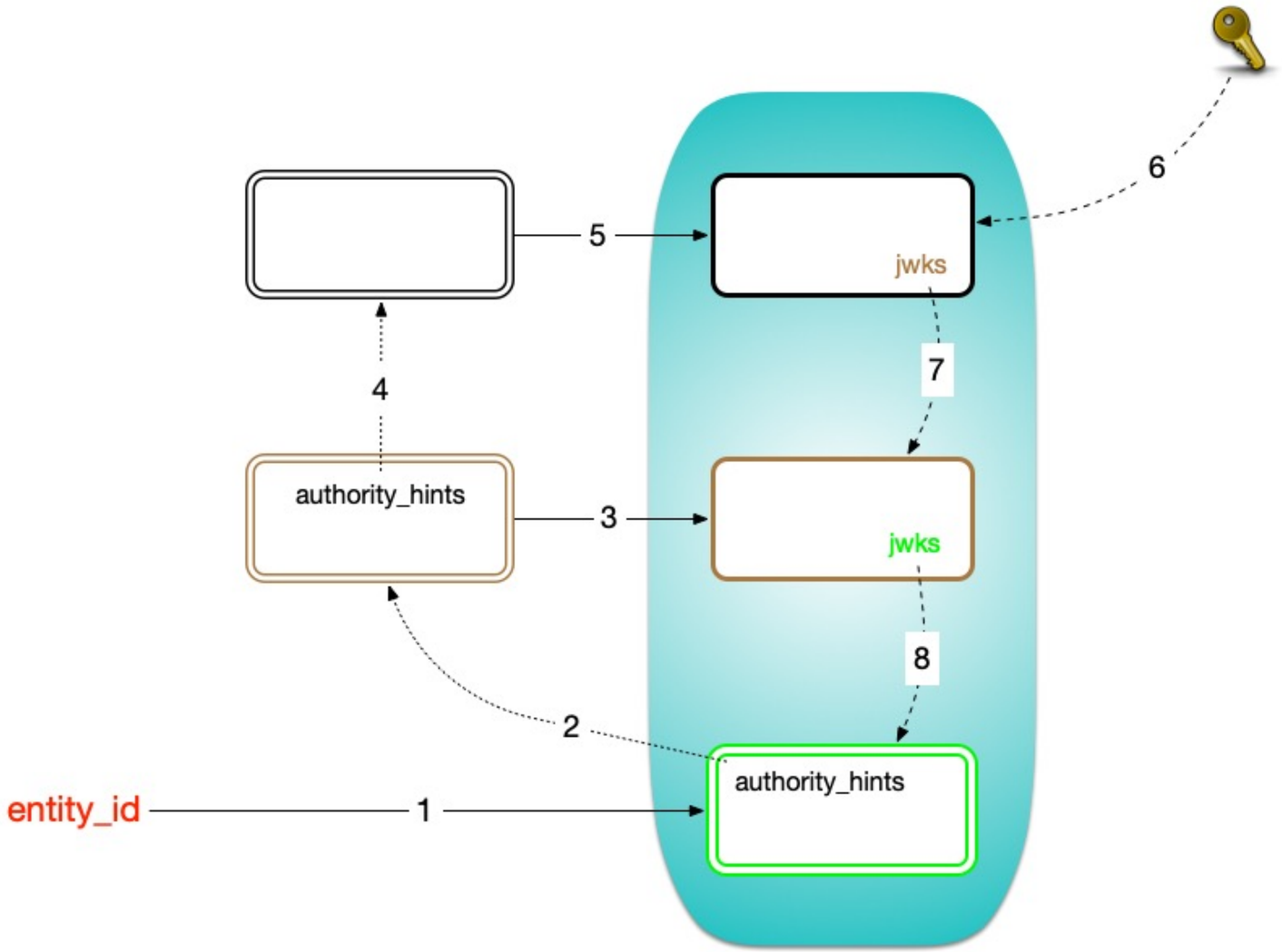


Fetch

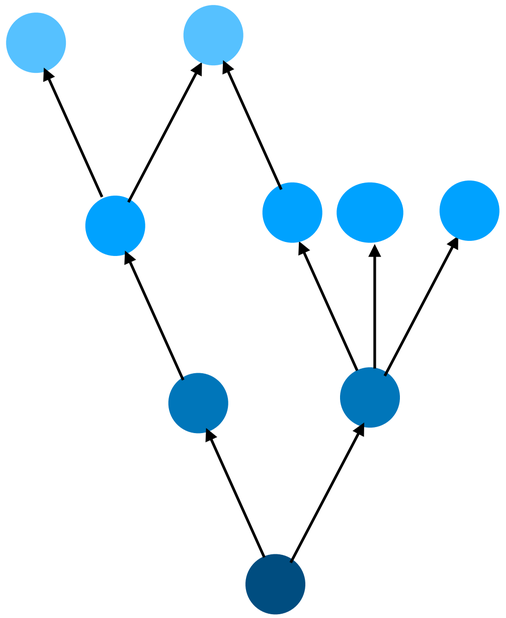
- Alla enheter som förväntas publicera information om andra enheter måste implementera en ‘fetch’-ändpunkt.



Insamling av ‘entity statements’



Entity statement träd(?)



Två enheter som har gemensam “trust anchor” tillhör samma federation.

Metadata policy

- add
- default
- one_of
- subset_of
- superset_of
- essential

```
{
  “metadata_policy”: {
    “openid_relying_party”: {
      “scopes”: {
        "subset_of": ["openid","eduperson"],
        "default": [“openid", "eduperson"]
      },
      "id_token_signed_response_alg": {
        "one_of": [“ES256", "ES384"],
        "default": "ES256"
      },
      "contacts": {
        "add": [
          "helpdesk@federation.example.org",
          "helpdesk@org.example.org"
        ]
      }
    }
  }
}
```

Trust Marks

Technically, trust marks as used by this specification are signed JWTs that represent a statement of conformance to a well-scoped set of trust and/or interoperability requirements.

Vem kan utfärda ’trust marks’

Alla enheter i en federation !

- Trust Anchor/Federations operatör’en
- Standardiseringsenhet
- Enhet om sig själv - OIDF test suite
- Enhet om annan enhet. AA om RP.

Trust mark introspektion

- En trust mark utfärdare kan/bör tillhandahålla en verifieringstjänst. Dit man kan skicka en trust mark och få veta om den fortfarande är aktiv.

Federationstjänster

- Fetch
  - En auktoritet om en underordnad -> entity statement
- Status
  - En ‘trust_mark’ utfärdare’s syn på status för ett ‘trust_mark’. -> True/False
- Resolve
  - En enhets (resolver) syn på en annan enhet -> {metadata, [trust_marks]}
- List
  - Lista över alla underordnade till en enhet -> [entity_id]

Klientregistreringsmetoder

- Automatisk
  - Klienten utför ingen registrering! Den skickar bara en authorization förfrågan med *client_id == entity_id* och client authentication method *private_key_jwt.*
  - OP’n måste hämta och verifiera RP’ns self-signed entity statement. För att sedan verifiera authenticerings JWT med nycklar från metadatat.
- Explicit
  - Klienten utför en dynamisk registrering. Registrerings frågan innehåller en self-signed entity statement.
  - OP’n hämtar förtroendekedjor
  - OP svarar med ett entity statement om RP’n.

Italian OIDC Federation 1.0

- 2 federationsoperatörer, AgID (SPID) och Ministero dell’Interno (CIE id)
- Kommer att använda automatisk registrering
- Planerar använda attributauktoriteter (baserat på trust marks)
- Ett fåtal OPer, >15.000 RPer, ~50% av Italiens befolkning till att börja med.
- Platt struktur.
- Multipla trust anchors.
- Implementerar SDKer i 5 språk: java, ruby, aspnetcore, python, nodejs, php

Questions!?