Towards Practical OIDCfed
But first: SAML Trust Relations

\[ N_{rel} = N_{sp} \times N_{idp} \]

\[ N_{rel} = N_{sp} + N_{idp} \]
SAML Federations

<table>
<thead>
<tr>
<th>Entities in eduGAIN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>All entities</td>
<td>5816</td>
</tr>
<tr>
<td>IdPs</td>
<td>3186</td>
</tr>
<tr>
<td>SPs</td>
<td>2628</td>
</tr>
<tr>
<td>Standalone AAs</td>
<td>4</td>
</tr>
</tbody>
</table>
Let’s zoom in

SURFnet provides services
 Saunders of SURFconext

Nikhef wants to collaborate
 Saunders eduGAIN
Size matters

“edugain-v1.xml” downloaden?

Bestandsnaam: edugain-v1.xml
Bestandsgrootte: 47,64 MB
Host: mds.edugain.org

☐ Altijd de standaard downloadmap gebruiken

[Opslaan] [Opslaan als...] [Openen] [Annuleren]
Metadata is stored in endpoints and federations

Nobody likes duplication
Nobody likes duplication

Multiple truths in T&I is not ideal
Federation redesigned: OIDCfed

Distributed

Hierarchical

Flexible

Policy enabled

Not actually OIDC only

https://openid.net/specs/openid-connect-federation-1_0.html
OIDCfed is built on trust

No more aggregation
  ↪ Trust relations

No more golden gateways
  ↪ Direct connections
OIDC Federations vs SAML Federations

Relying Party

OpenID Provider

Trust Anchor/Signing Service

Service Provider

Identity Provider

(Proxy/MDQ/New)
Introducing Trust Anchors/MDSSs

Hierarchically similar to proxies
- AARC BPA

Technically very different
- MDSS only expresses trust
In short

OIDC Federations only have to express trust

New component: MDSS

No more full trust in one proxy

*And now for the practical part*
How to use the MDSS

Roland’s fedservice
- Python
- File/DB based
- Keyfile signing
- Partial implementation
- Proof of Concept

Nikhef MDSS
- Java 8
- MongoDB based
- Keyfile & PKCS#11 signing
- Full implementation
- Production ready
Why Nikhef MDSS?

<table>
<thead>
<tr>
<th>Name</th>
<th>Last commit</th>
<th>Last update</th>
</tr>
</thead>
<tbody>
<tr>
<td>gradle/wrapper</td>
<td>Add gradle wrapper</td>
<td>4 months ago</td>
</tr>
<tr>
<td>src/main</td>
<td>Fix race condition (?) in MongoDB initialization</td>
<td>2 months ago</td>
</tr>
<tr>
<td>.gitignore</td>
<td>Add strict mode, where the MDSS will only sign for the specified issuer</td>
<td>4 months ago</td>
</tr>
<tr>
<td>.gitlab-ci.yml</td>
<td>Enable GitLab CI</td>
<td>4 months ago</td>
</tr>
<tr>
<td>LICENSE</td>
<td>Update LICENSE</td>
<td>4 months ago</td>
</tr>
<tr>
<td>build.gradle</td>
<td>Fix small issue with group naming in the buildscript</td>
<td>1 week ago</td>
</tr>
<tr>
<td>gradlew</td>
<td>Initial MDSS, will sign anything under any issuer, metadata not included. Do...</td>
<td>2 months ago</td>
</tr>
<tr>
<td>gradlew.bat</td>
<td>Initial MDSS, will sign anything under any issuer, metadata not included. Do...</td>
<td>5 months ago</td>
</tr>
<tr>
<td>settings.gradle</td>
<td>Improvements based on a SonarQube inspection</td>
<td>4 months ago</td>
</tr>
</tbody>
</table>
Nikhef MDSS

Running at IGTF
Easy to deploy
Strict mode
Built for HSM usage

To do: import without clearing state

Built on Spring
- Understands reverse proxies
- Easy configuration
- Modular design

https://gitlab.com/oidcfed/mdss
OIDCfed explicitly has no discovery, but...

```bash
$ curl -S "https://oidcfed.igtf.net/signing-service?operation=listing&iss=https://oidcfed.igtf.net/root" 
["https://oidcfed.igtf.net/root"]
```
Validation

Construct trust chains

Verify a trusted element in chain

Verify policy and constraints

Select a preferred chain if N>1
~ jwk "https://oidcfed.igtf.net/.well-known/openid-federation"
{
  "iss": "https://oidcfed.igtf.net/root",
  "sub": "https://oidcfed.igtf.net/root",
  "iat": 1580901972,
  "exp": 1580905572,
  "jwks": [
    {
      "keys": ["kty": "EC",
                "kid": "Nikhef/IGTF Test",
                "use": "sig",
                "alg": "ES512",
                "x": "AZd5qWvwgWmzi3r8fH3duSRBJfYNS7IzKWXACKJEeDvUyhAqf5Dqi1oVVbnutcUyeE3vWuP3WAXyKN05csCjZw-",
                "y": "AaaETWyc_9Aihl0D29d7WT-1DoxQfyyuBA3j14k980v6Rjx-VEw2wmXHNwcc90LflpaL9L-P9zABghmEhkb4BrCm",
                "crv": "P-521"
        }
    ],
  ],
  "metadata": {
    "federation_entity": {
      "federation_api_endpoint": "https://oidcfed.igtf.net:443/signing-service"
    }
  }
}
A set of these statements forms a chain

Authority hints

Policy and constraints on the chain

OP or RP as base

Many things are optional
Status of the validation engine

Library is done-ish

Being integrated into GEANT Shibboleth OIDC

Interop tests with Roland’s work coming soon

Prevent against endless chain DoS

Review software correctness

Optimizations?

Can we support SAML?!?
DIY w/ Docker Compose - Step 0

Create a new folder

Have docker and docker compose

Download the latest MDSS build: https://gitlab.com/oidcfed/mdss

Create a JWK
https://mkjwk.org/
version: "3.7"
services:
  mongodb:
    image: 'mongo'
  mdss:
    image: 'openjdk:8-alpine'
    volumes:
      - './mdss.jar:/opt/mdss/mdss.jar'
      - './application.properties:/opt/mdss/application.properties'
      - './key.jwk:/opt/mdss/key.jwk'
      - './import:/opt/mdss/import'
    ports:
      - '8080:8080'
    working_dir: '/opt/mdss'
    command: 'java -jar mdss.jar'
application.properties:
oidcfed.keyprovider=file
oidcfed.keyconfig=/opt/mdss/key.jwk
oidcfed.import_folder=/opt/mdss/import
oidcfed.strict=false
spring.data.mongodb.host=mongodb
Have some metadata

```json
{
    "authority_hints": [],
    "exp": 0,
    "iat": 0,
    "iss": "https://www.igtf.net/oidcfed",
    "sub": "https://www.igtf.net/oidcfed",
    "metadata": {"federation_entity": {}},
    "jwks": {"keys": [{
        "kty": "EC",
        "use": "sig",
        "crv": "P-521",
        "kid": "IGTF",
        "x": "AE9BlV3IR1XVuc01rewAQLel6hsfbZiRzRTAC6argWRbFg2B1SBZn-Uof-ZOlm_aKAU-IWFmfh8xdikRNV1lb6EF",
        "y": "ANSai2rNNMoCVuq5iZzPjBJEAWr8RtWvBjy8vBxQD2ozaSvJpA0Ye7VVqbaOdSXTgv3YJITnVylmMe6epj9Sarm",
        "alg": "ES512"
    }]
}
}```
Start the docker compose file

Import metadata with: curl -S "http://localhost:8080/mdss-management/import-all"
You can now request statements about https://www.igtf.net/oidcfed at your MDSS

curl -S

It should be that easy!