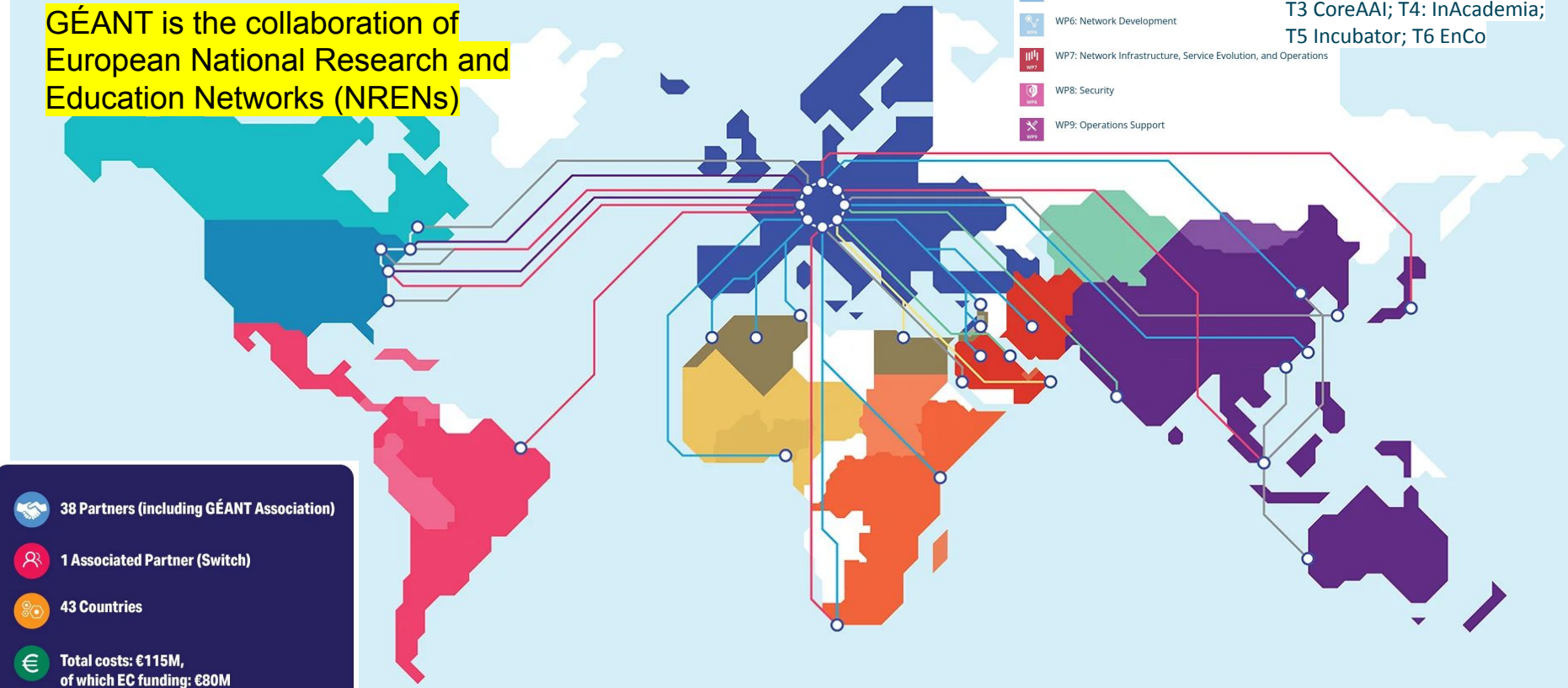
TI Incubator cycle 11 half-time demo

05 February 2026

TRUST & IDENTITY INCUBATOR

Introduction to the GÉANT project

GÉANT is the collaboration of European National Research and Education Networks (NRENs)



GN5-2 is arranged around nine Work Packages covering 42 Tasks:

- WP1: Project Management
- WP2: Marketing Communications, Events, and Policy Engagement
- WP3: User and Stakeholder Engagement
- WP4: Above-the-Net Services
- WP5: Trust & Identity
- WP6: Network Development
- WP7: Network Infrastructure, Service Evolution, and Operations
- WP8: Security
- WP9: Operations Support

GÉANT
Networks • Services • People

WP5
T1: EDUROAM; T2 eduGAIN;
T3 CoreAAI; T4: InAcademia;
T5 Incubator; T6 EnCo

- 38 Partners (including GÉANT Association)
- 1 Associated Partner (Switch)
- 43 Countries
- Total costs: €115M, of which EC funding: €80M

Introductions

TI Team

- Alexandr Petrunin
- Alexandru Cacean
- Andrijana Todosijevic
- Amin Mahnamfar
- Amineh Akhavan Saraf
- András Bodor
- Boro Jakimovski
- Febri Kazazi
- Gabriel Zachmann (on parental leave)
- Halil Adem
- Harm Roukema
- Janne Lauros
- Jovan Simonoski
- Kushal Das
- Marko Ivančić
- Mihály Héder
- *Niels van Dijk*
- Peter Bolha

WP5 Leads

- *Irina Sidorova*
- *Maarten Kremers*
- *Marina Adomeit (on parental leave)*

Some topics in cycles (2019-present)

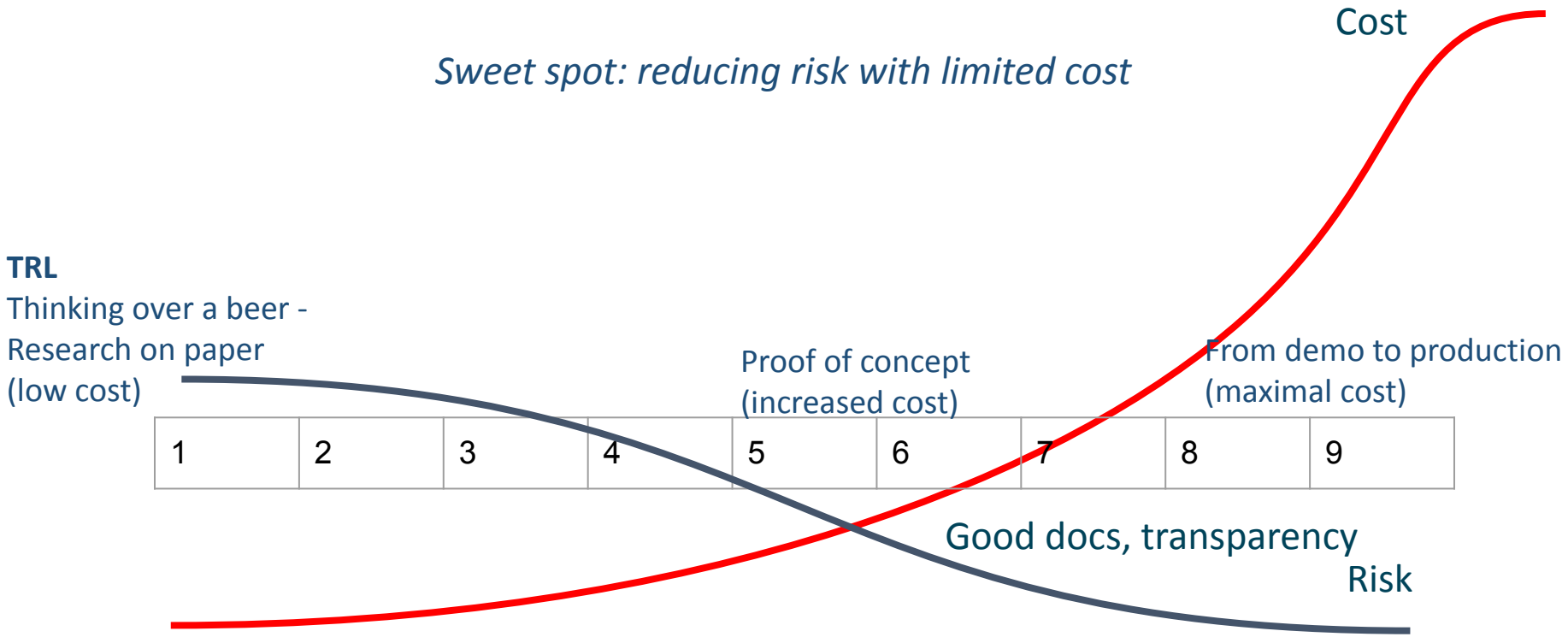
1. Trust Marks -> SIRTIFI+JAGGER
   SeamlessAccess trials at GEANT
   Cryptech HSM
2. eduGAIN status reporting
   Push MDQ
3. eduLNK
4. SSP OIDC OP
   OIDC agent for windows
   DID for researchers
5. DID+EBSI+IRMA wallet
6. SSH certificates on federated login
   IdP user profile
7. geteduroam linux client
   SSP OIDFed
8. Scalable testing for Insecure XMLsig
   Trust fabric for wallets ARF
9. eduGAIN OIDFed PoC
   OIDFED in Shib and SSP (part2)
10. GO OP, TA, RP implementation
    SeamlessAccess OIDFed PoC
    VC issuer IDPs
    Bona Fide researcher vetting

Past cycles on the dashboard:

Now running: cycle 11:
October 25-May 26

Innovation theory in a nutshell

Sweet spot: reducing risk with limited cost



Shibboleth and SSP Verifiable Credential Issuers' DIIP compliance

Janne Lauros
Marko Ivancic
Mihály Héder

Introduction

Previous cycle was about having Shibboleth IdP and SimpleSAMLPHP issue credential to wallet - any credential to any wallet and by any means.

This cycle we aim for interoperability conformance.

Our named goal is DIIP, https://fidescommunity.github.io/DIIP

There is also HAIP, https://openid.net/specs/openid4vc-high-assurance-interoperability-profile-1_0.html

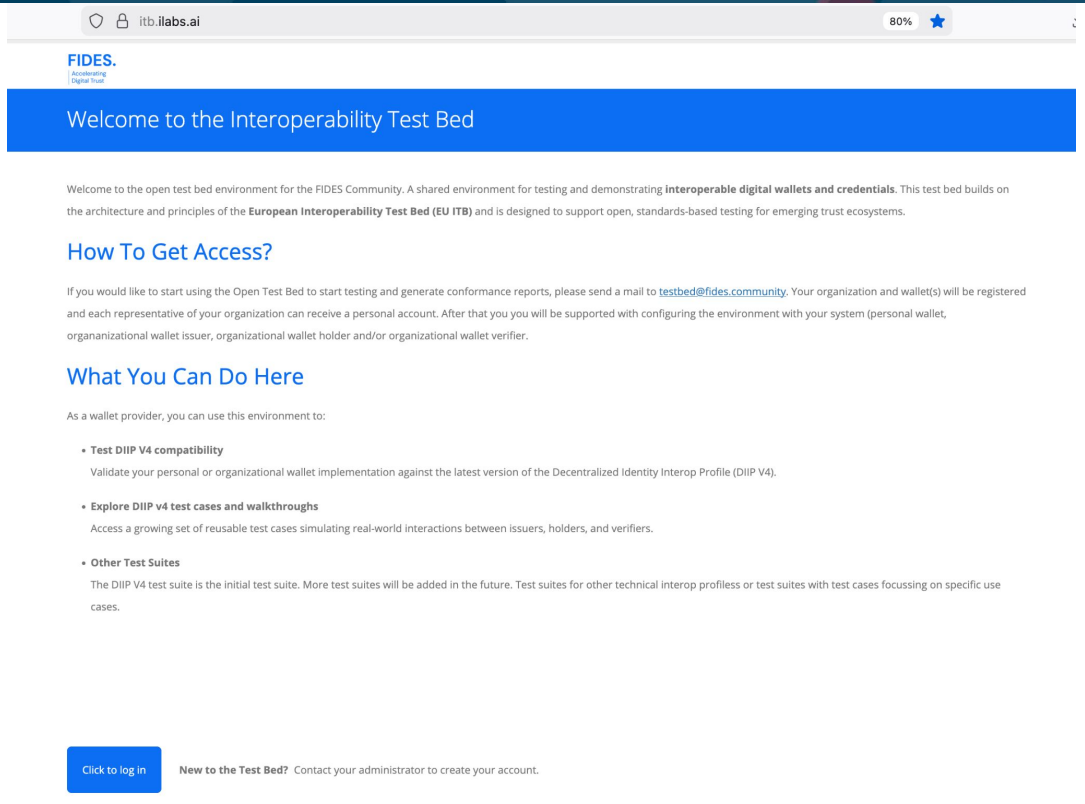
Full conformance will not be reached in this cycle but we are closing

Conformance Tools - DIIP https://itb.ilabs.ai/

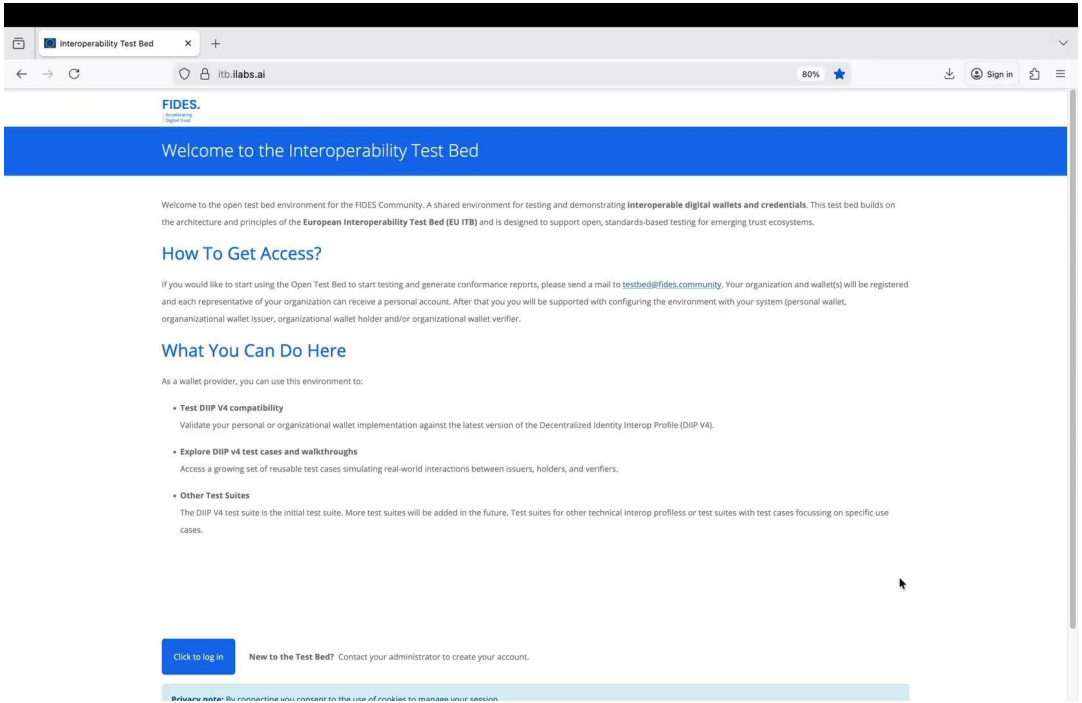
Should be our main tool for running interoperability tests

Has only one test for Issuer (that I am able to locate)

We pass that test



https://itb.ilabs.ai/ - pre-authorized flow without tx code



Shib Admin Task - Credential Definition

Define a credential as per

https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#name-credential-issuer-metadata #`credential_configurations_supported`

The definition is used by:

Shibboleth itself as a instruction what credentials it supports and the specifics of them like format.

Wallets as they read Issuer configuration for which credentials that are being offered have suitable specifics like format.

`/opt/shibbolet-idp/metadata/verifiable-credentials.json`

Shib Admin Task - pre-authorized flow relying parties for DIIP test

The party that is allowed to initiate pre-authorized flow i.e. generate the url/qr code is considered in this flow the actual client

```
openid-credential-offer://?credential_offer={"grants":{"urn:ietf:params:oauth:grant-type:pre-authorized_code":{"pre-authorized_code":"AAd..6dw"}},"credential_issuer":"https://geant-vci.2.rahtiapp.fi","credential_configuration_ids":["GeantIncubatorDiploma"]}
```

Wallet not presenting any client identifier is referred with identity of the party that initiated the flow

`/opt/shibboleth-idp/conf/relying-party.xml`

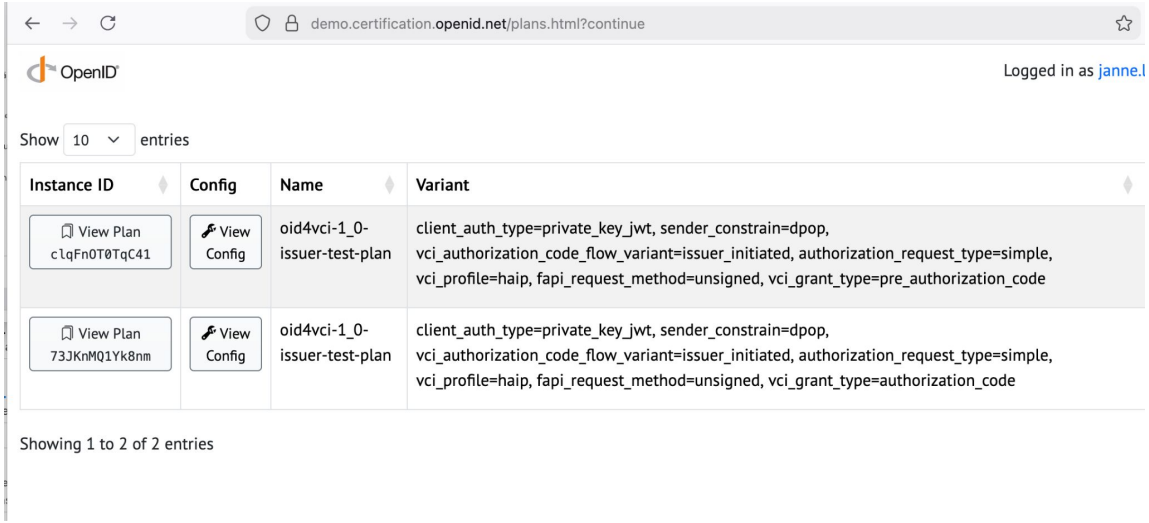
Conformance Tools - OpenID
https://demo.certification.openid.net/login.html

Also still work in progress

Only “happy flow” cases exist for both pre-authorized and code flows

We pass both tests!

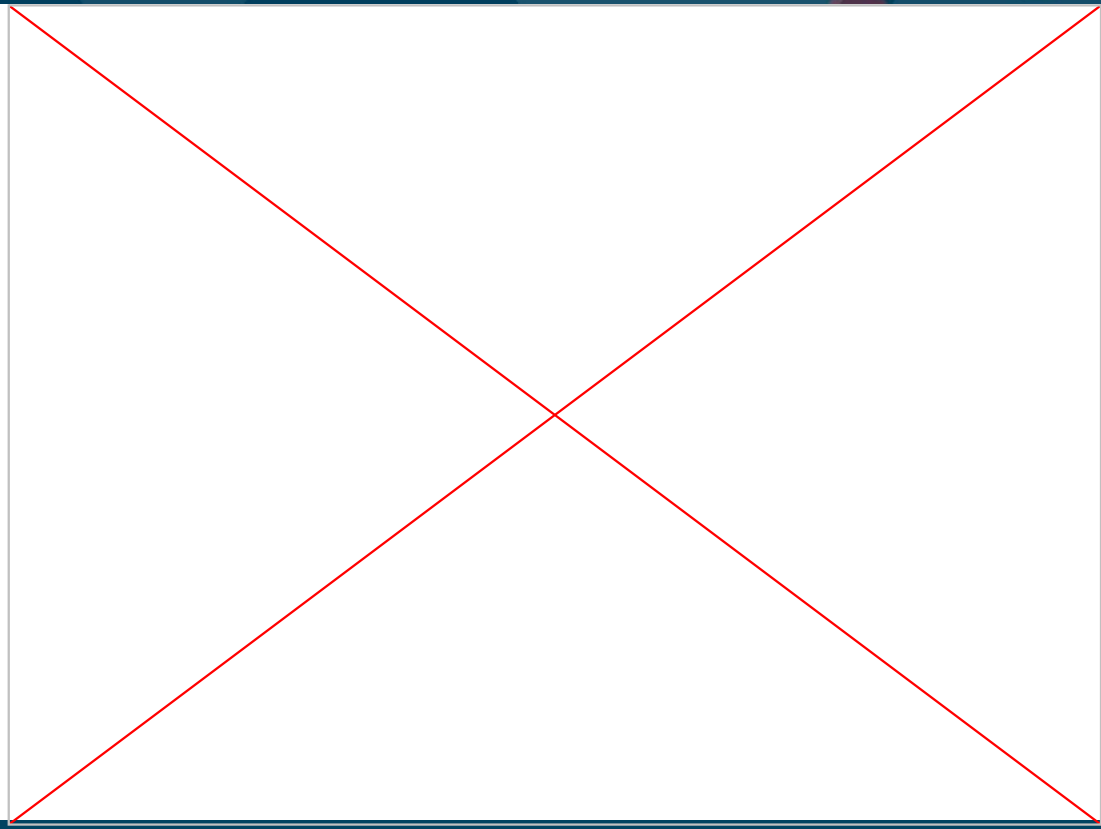
Excellent tooling even in this draft stage!



https://demo.certification.openid.net/login.html - pre-authorized flow

We run here all available pre-authorization flow tests.

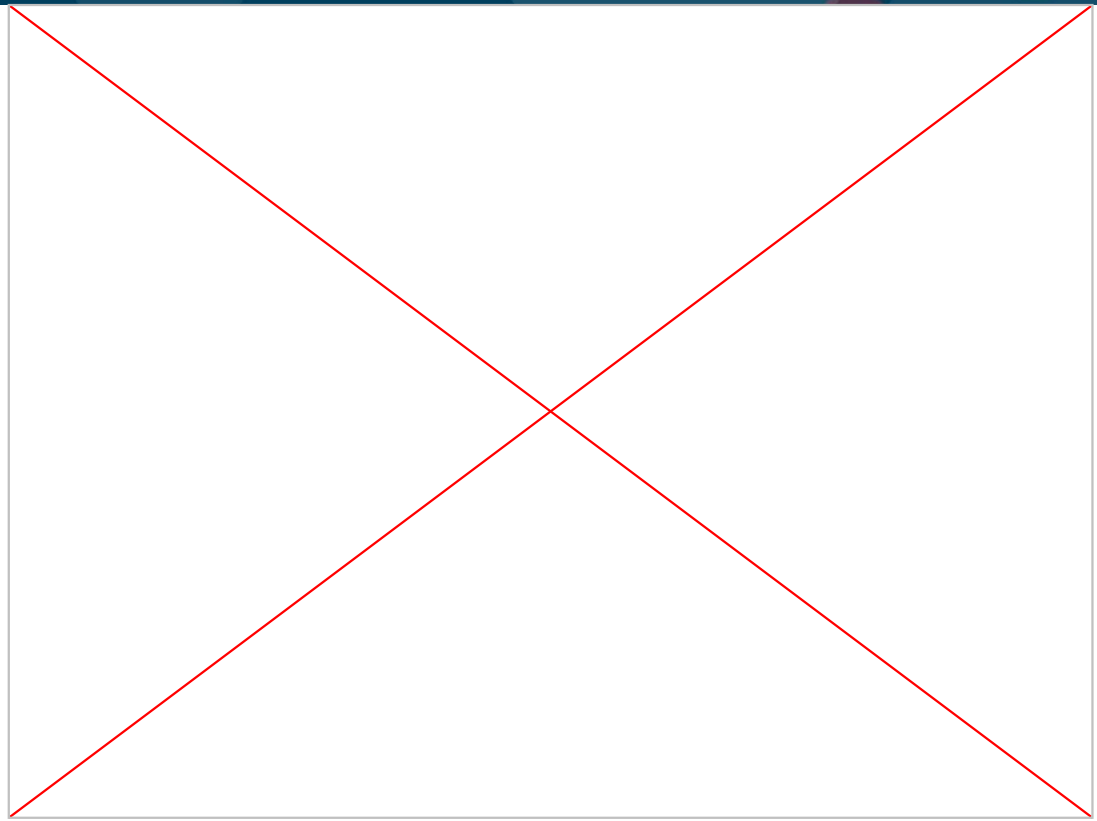
Excellent tooling! Notice the number of specific checks and explanations.



https://demo.certification.openid.net/login.html - code flow

We run here all available code flow tests.

Still Excellent tooling! I wish it had already the tricky tests that try to fool my implementation.



Shib Admin Task - Relying parties for OpenID test

“Normal” known relying parties vs. DIIP case. No inherited client identifiers. DPOP configured, PAR configured for Code flow.

`/opt/shibboleth-idp/conf/relying-party.xml`

Shib Admin Task - Code flow requires attribute, for now like this:

```
<AttributeDefinition id="GeantIncubatorDiploma" xsi:type="ScriptedAttribute">
  <InputAttributeDefinition ref="mail" />
  <InputAttributeDefinition ref="eduPersonPrincipalName" />
  <InputAttributeDefinition ref="givenName" />
  <InputAttributeDefinition ref="sn" />
  <InputAttributeDefinition ref="eduPersonAffiliation" />
  <Script><![CDATA[
      var credential = '{"requestedCredential":[\
                        {"path":["diploma"],"value":"Scrum Master"},\
                        {"path":["mail"],"value":"'+mail.getValues().get(0)+'"},\
                        {"path":["eppn"],"value":"'+eduPersonPrincipalName.getValues().get(0)+'"},\
                        {"path":["givenName"],"value":"'+givenName.getValues().get(0)+'"},\
                        {"path":["familyName"],"value":"'+sn.getValues().get(0)+'"},\
                        {"path":["affiliation"],"value":"'+eduPersonAffiliation.getValues().get(0)+'"}]\
                      }';
      GeantIncubatorDiploma.addValue(credential.replaceAll("\\s+",""));
  ]]></Script>
  <AttributeEncoder xsi:type="oidc:OIDCString" name="GeantIncubatorDiploma"/>
</AttributeDefinition>
```

Compare to `/opt/shibbolet-idp/metadata/verifiable-credentials.json`

Shib Features

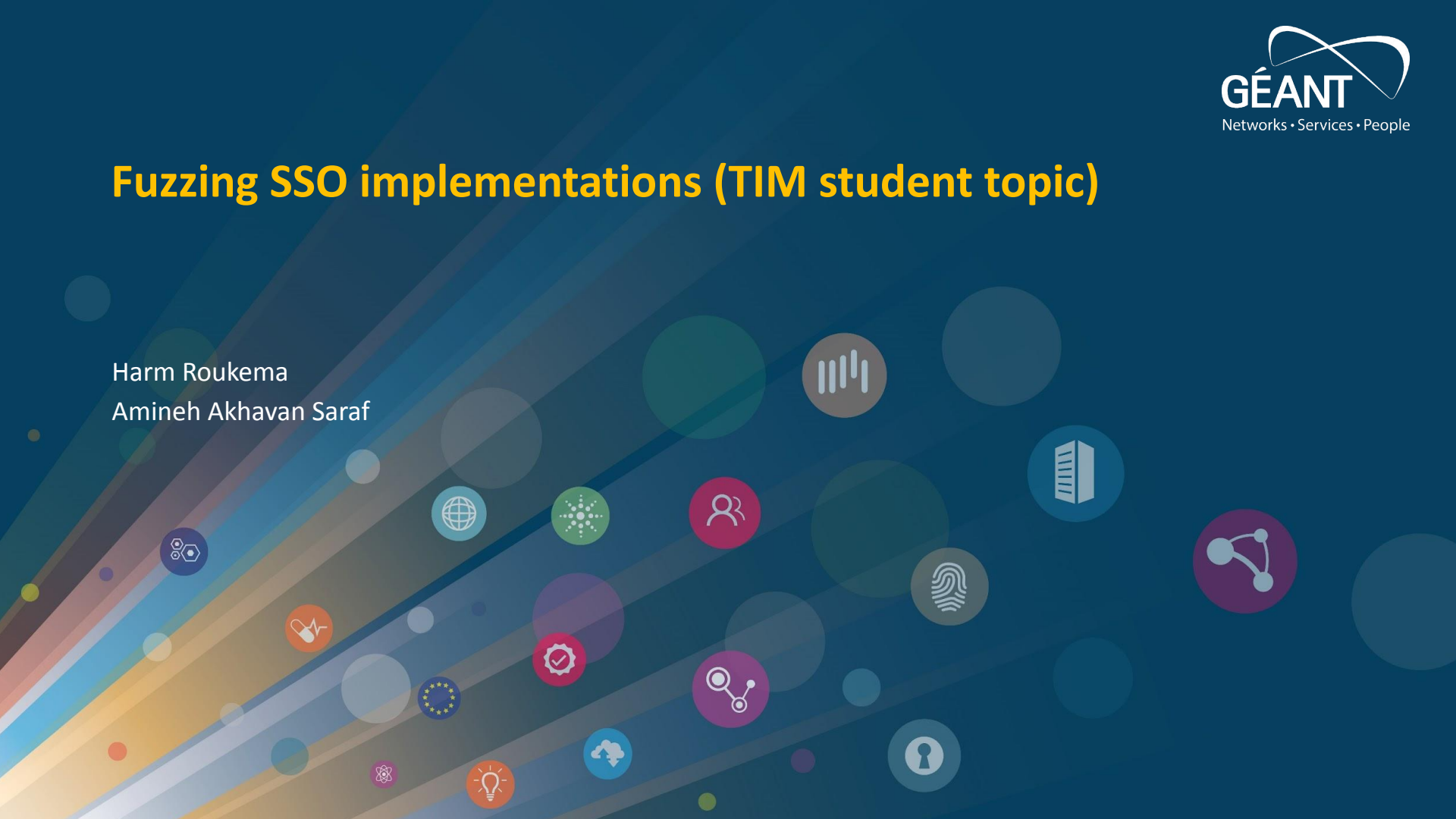
Features still missing to meet DIIP (v4) on paper.

- “did:jwk” and “did:web” identifiers, analysis what responsibilities comes to Issuer from them and possible implementations. Undersigned has no clue yet.
- IETF Token Status List as token status list mechanism i.e. revoking credentials. Undersigned has no clue yet.
- Some other more minor details.

To reach maturity have first public version would require still a lot more than merely implement the features above. There are missing validations, not every corner case has been implemented and configurations need lot of work still. What is good we seem to be able to write credentials to several wallets without much pain. ***Luckily Incubator is about incubating and not birthing****.*

Fuzzing SSO implementations (TIM student topic)

Harm Roukema
Amineh Akhavan Saraf



Introduction

- Why look into security of SSO implementations?
  - Protocols are secure, but can be complex to implement
  - Bugs can lead to severe vulnerabilities like account takeover
- Fuzzing: automatically randomize inputs to uncover bugs
- Focus on OIDC
  - SimpleSAMLphp OIDC module
- Later: Shibboleth, possibly SAML

Example: redirect_uri vulnerability

- Example OIDC authentication request:

https://server.example.org/authorize?response_type=code&scope=openid%20profile%20email&client_id=s6BhdRkqt3&state=af0ifjsldkj&**redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb**

- Example OIDC authentication response:

```
HTTP/1.1 302 Found
 Location: https://client.example.org/cb?
    code=SplxlOBeZQQYbYS6WxSbIA
    &state=af0ifjsldkj
```

Example: redirect_uri open redirect

- Example implementation error:

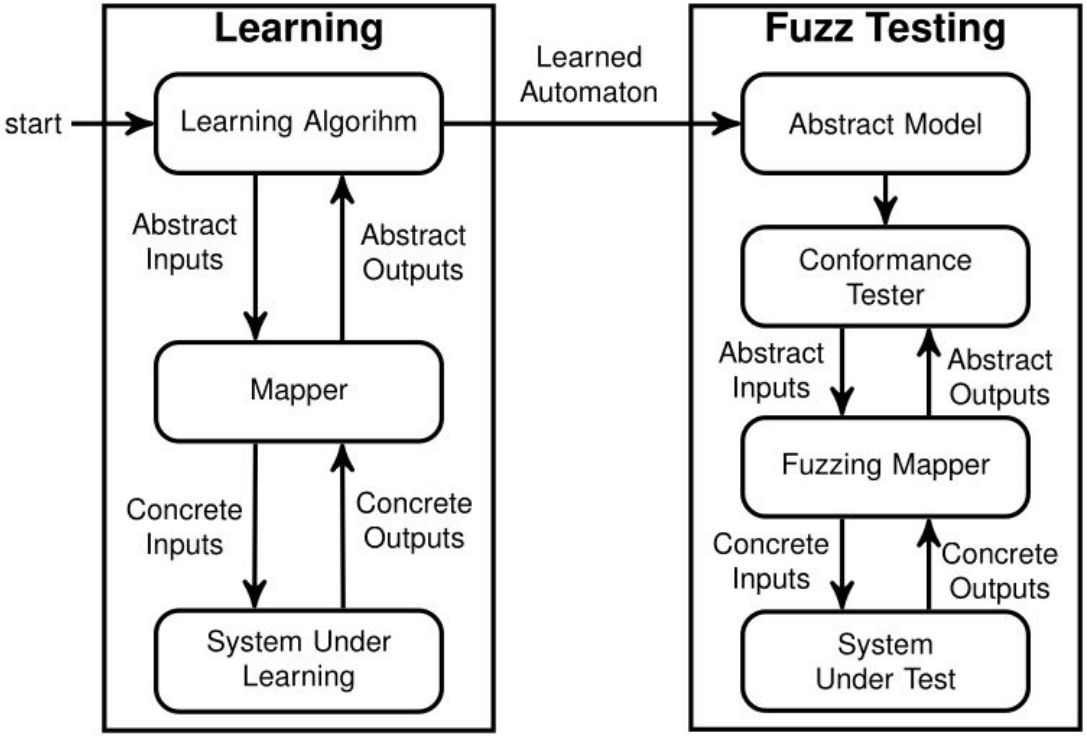
```
if redirect_uri.startswith("https://client.example.org"):
    ...
```

- Bypass:

https://server.example.org/authorize?response_type=code&scope=openid%20profile%20email&client_id=s6BhdRkqt3&state=af0ifjsldkj&r**edirect_uri=https%3A%2F%2Fclient.example.org.evil.com%2Fcb**

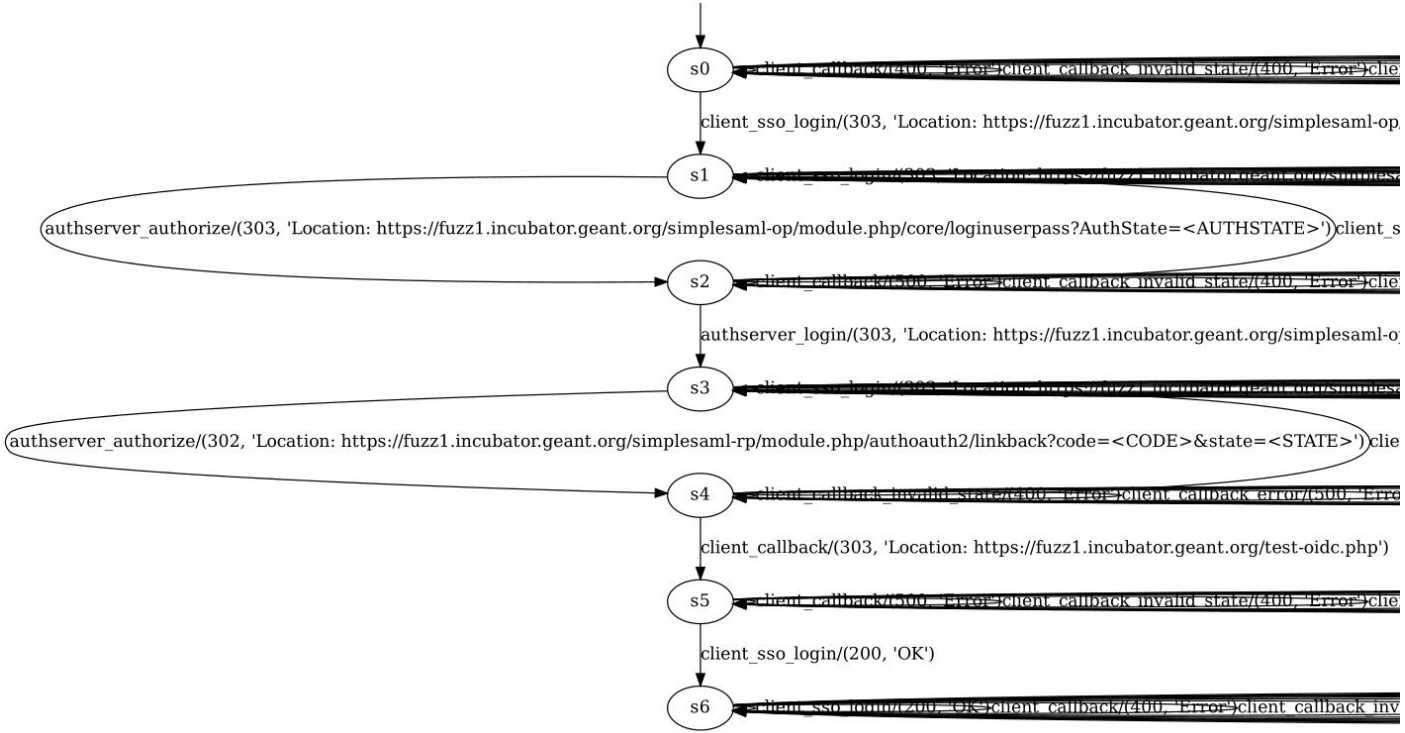
- Impact: one-click account takeover (authorization code is sent to evil.com!)

Learning-based Fuzzing



[1]

SimpleSAMLphp OIDC authorization code flow



Fuzzing (WIP)

- Examine learnt state machines (manually)
  - Differences between SimpleSAMLphp and Shibboleth?
- Fuzz using state machines
  - Random walks through states with fuzzed inputs
- Fuzzing parameters: define properties of parameters [2]
  - Constant/variable, mandatory, used once or multiple, user-specific, session-specific
  - E.g. `state`: session-specific -> get new session and substitute
  - URLs (`redirect_uri`): use known specific tricks, e.g. redirect-fuzzer [3]

Demo of vulnerable redirect_uri

- Uses previously learned state machine
- Walks through state machine using fuzzed inputs
- Finds **difference in outputs** between fuzzed inputs and expected behavior:
  - Redirect to `https://client1-oauth-ctf.harmroukema.nl.evil.com?code=<CODE>&state=<STATE>`
- Shows concrete traces for reproducibility

Goals for the cycle

- Implement more fuzzing techniques and fuzz SSP OIDC
  - JWT fuzzing
  - Learn more flows
- Fuzz Shibboleth
  - Compare state machine with SSP OIDC

Stretch goals:

- Fuzz SAML implementations

References

[1] Bernhard K. Aichernig, Edi Muškardin, and Andrea Pferscher. Learning-based Fuzzing of IoT Message Brokers.
[2] Ronghai Yang, Guanchen Li, Wing Cheong Lau, Kehuan Zhang, and Pili Hu. Model-based security testing: An empirical study on OAuth 2.0 implementations.
[3] Wang, X., Lau, W. C., Yang, R., & Shi, S. (2019). Make redirection evil again: URL parser issues in OAuth.

Accessibility testing of wallet ecosystem

Febri Kazazi
Esther Ruiz Ben

Introduction

- Departing from **DUxDI-Project's results** (GÈANT Innovation Programme 2024/25 - University of Málaga and DFN) related to Accessibility of Digital Wallets in education and research contexts.
- Target in T&I Incubator cycle 11:
  - **Improve accessibility of digital wallets prototypes for education and research**.
- Methods currently used:
  - **Tests of digital wallets with available tools, including those AI-based** (i. e. WAVE, Lighthouse, AInspector). Considering selected scenarios (Wallet Registration, VC Issuance, VC Presentation, Selective Disclosure, Interoperability with assistive tools such as NVDA).
- Methods to be further applied: **digital wallets UX-accessibility tests by persons with disabilities** active in education and research contexts.

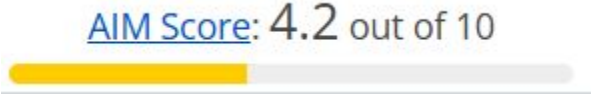
Shibboleth Wallet Accessibility Report

Lighthouse report

Desktop score

Mobile score

WAVE tool report



Shibboleth Wallet Accessibility Report

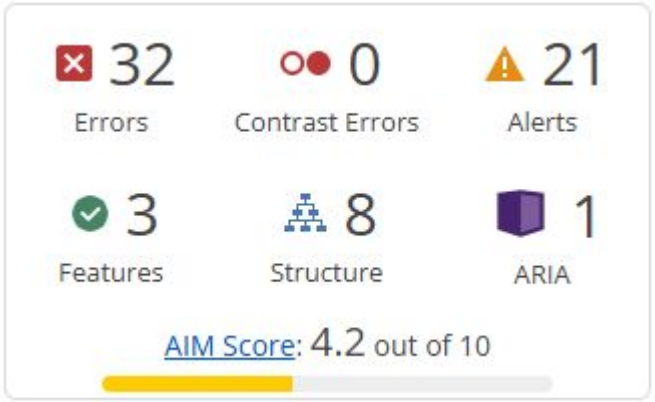
Lighthouse report

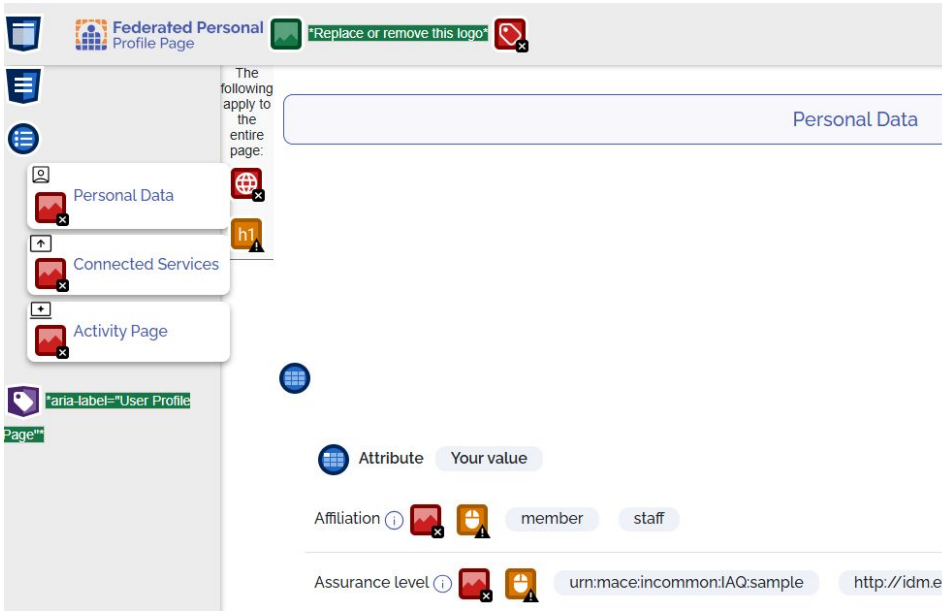
Desktop score

Mobile score

Shibboleth Wallet Accessibility Report

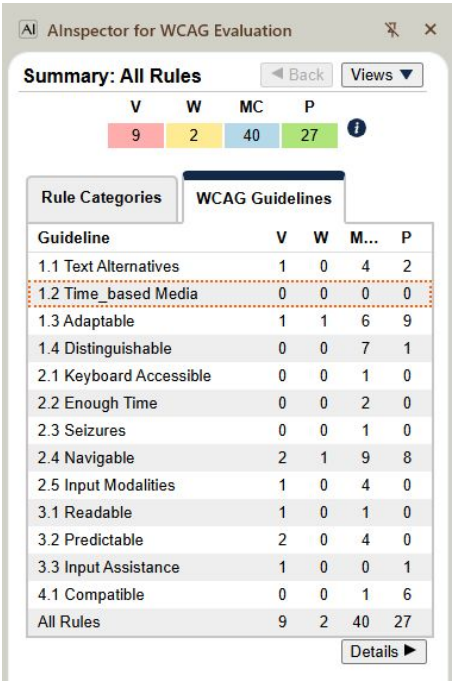
WAVE tool report

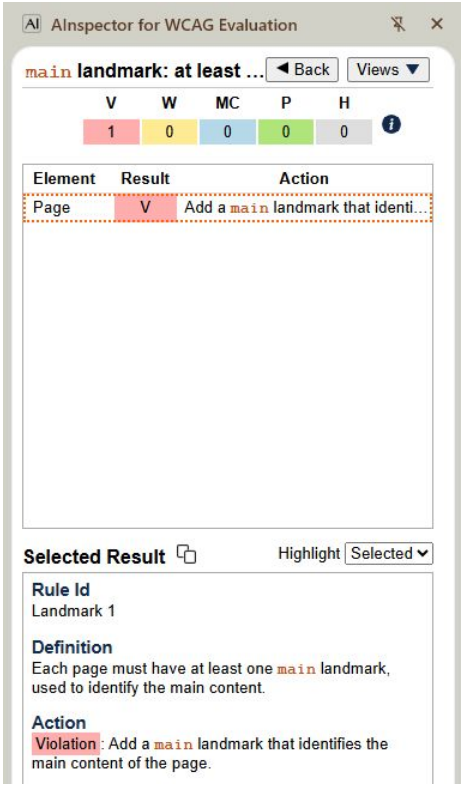




Shibboleth Wallet Accessibility Report

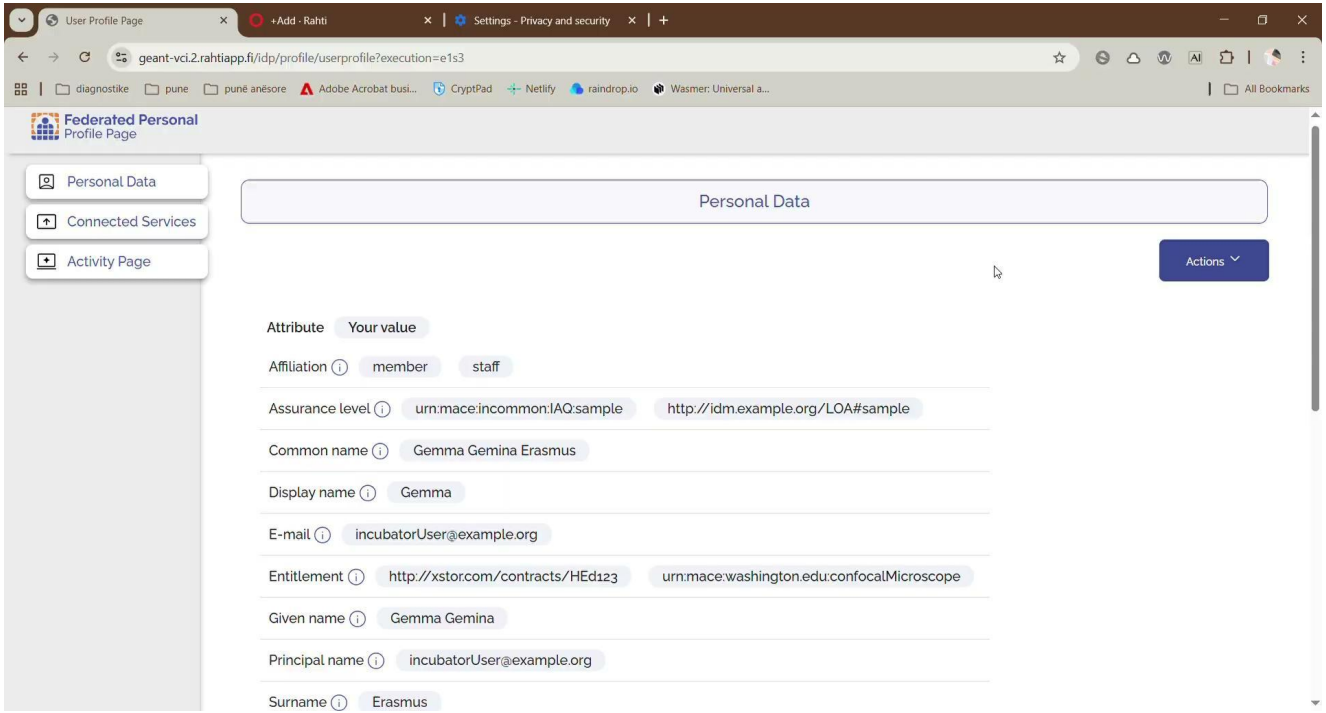
University of Illinois AInspector report





Shibboleth Wallet Accessibility Report

NVDA screen reader testing



Shibboleth Wallet Accessibility Report

Issues

Severe

- Image elements do not have [alt] attributes.
- <html> element does not have a [lang] attribute.
- Document does not have a main landmark.
- Missing [aria*] attributes.
- Missing and/or empty form labels.
- Relying on device dependent event handlers only.

Light

- Incorrect navigation order.
- Improper use of HTML semantics.

Next Steps

- Apply the fixes to Shibboleth and SimpleSAMLphp wallets.
- Perform another quality test to identify any other.
- Run a “user flow” simulation with people with disabilities.
- Smartphone APP testing - how?

OIDFed topics

Introduction

OpenID Federation topics:

- PHP OIDFed library and RP
- Local discovery for OIDFed
- Apache mod_oidfed (TIM)
- OpenID Federation Registry

PHP OIDFed library and RP

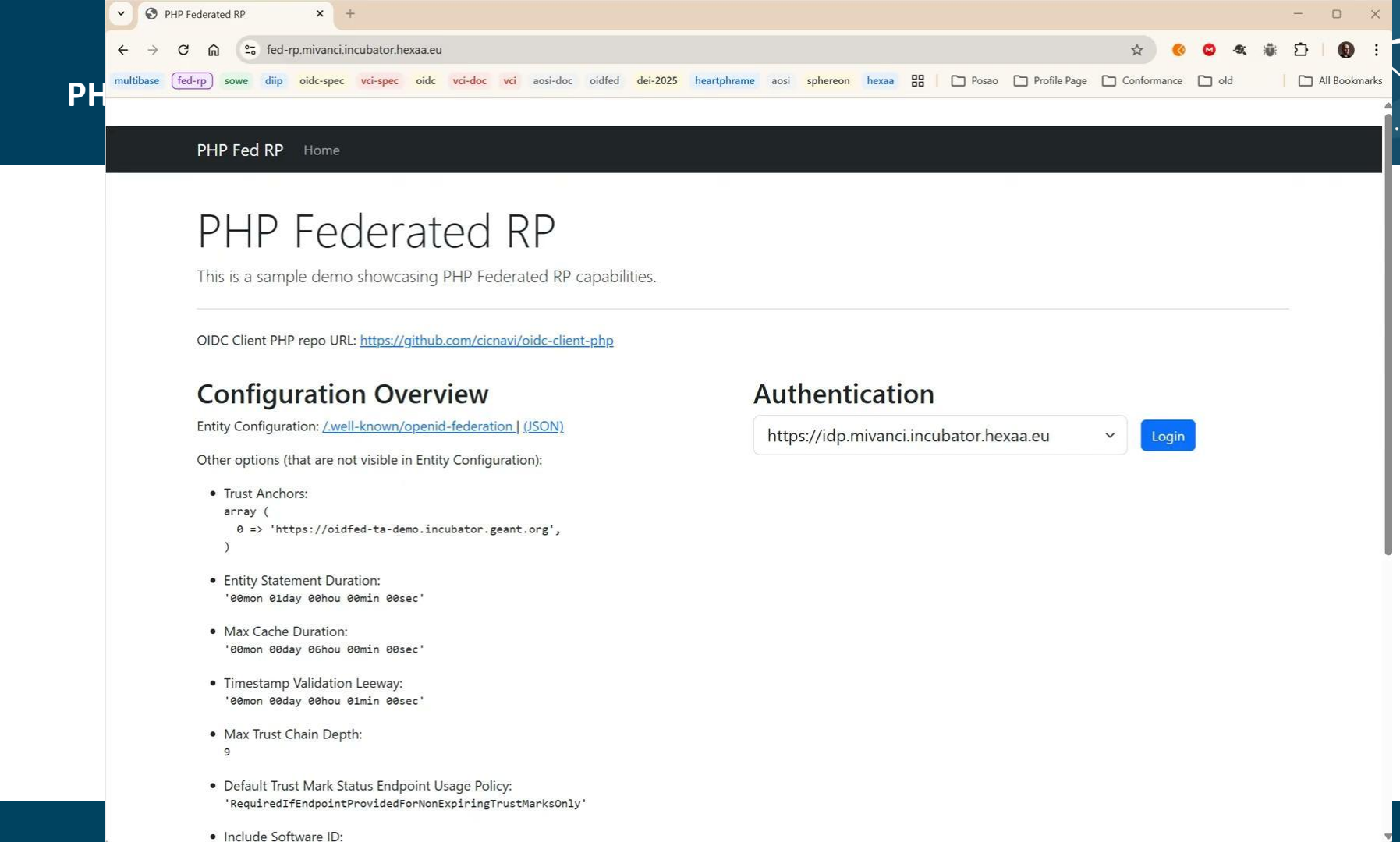
Marko Ivančić
Peter Bolha
Jovan Simonoski

PHP OIDFed library and RP

- PHP OpenID Library
  - https://github.com/simplesamlphp/openid
  - Contains various abstractions / tools related to OpenID specs
  - Related to Federation spec: has tools to build Entity Statements, resolve Trust Chains, fetch and Validate Trust Marks… with auto-cache-until-expiration feature
- Generic Relying Party in PHP with federated capabilities
  - https://github.com/cicnavi/oidc-client-php/tree/v3.x
  - Proof-of-concept generic PHP RP with federated capabilities
  - Automatic client registration and authorization code flow
  - Uses simplesamlphp/openid tools under the hood
  - Demo app: https://fed-rp.mivanci.incubator.hexaa.eu/

PHP OIDFed library and RP

- SimpleSAMLphp OP module (`oidc`)
  - OAuth 2.0 Token Introspection (Jovan, Peter)
  - Support for defining and using multiple signature algorithms and keys (prepared for v7 which is WIP)
- Next steps
  - Federation capabilities for SimpleSAMLphp RP module https://github.com/cirrusidentity/simplesamlphp-module-authoauth2
  - SimpleSAMLphp as a Trust Anchor / Intermediate : https://github.com/simplesamlphp/simplesamlphp-module-oidanchor
  - If we have time:
    - Pushed Authorization with automatic client registration
    - Dynamic / Explicit client registration


PH
People
PHP Federated RP
fed-rp.mivanci.incubator.hexaa.eu
multibase
fed-rp
sowe
diip
oidc-spec
vci-spec
oidc
vci-doc
vci
aosi-doc
oidfed
dei-2025
heartphrame
aosi
sphereon
hexaa
Posao
Profile Page
Conformance
old
All Bookmarks
PHP Fed RP
Home
PHP Federated RP
This is a sample demo showcasing PHP Federated RP capabilities.
OIDC Client PHP repo URL: https://github.com/cicnavi/oidc-client-php
Configuration Overview
Entity Configuration: /.well-known/openid-federation | (JSON)
Other options (that are not visible in Entity Configuration):
Trust Anchors:
array (
0 => 'https://oidfed-ta-demo.incubator.geant.org',
)
Entity Statement Duration:
'00mon 01day 00hou 00min 00sec'
Max Cache Duration:
'00mon 00day 06hou 00min 00sec'
Timestamp Validation Leeway:
'00mon 00day 00hou 01min 00sec'
Max Trust Chain Depth:
9
Default Trust Mark Status Endpoint Usage Policy:
'RequiredIfEndpointProvidedForNonExpiringTrustMarksOnly'
Include Software ID:
Authentication
https://idp.mivanci.incubator.hexaa.eu
Login

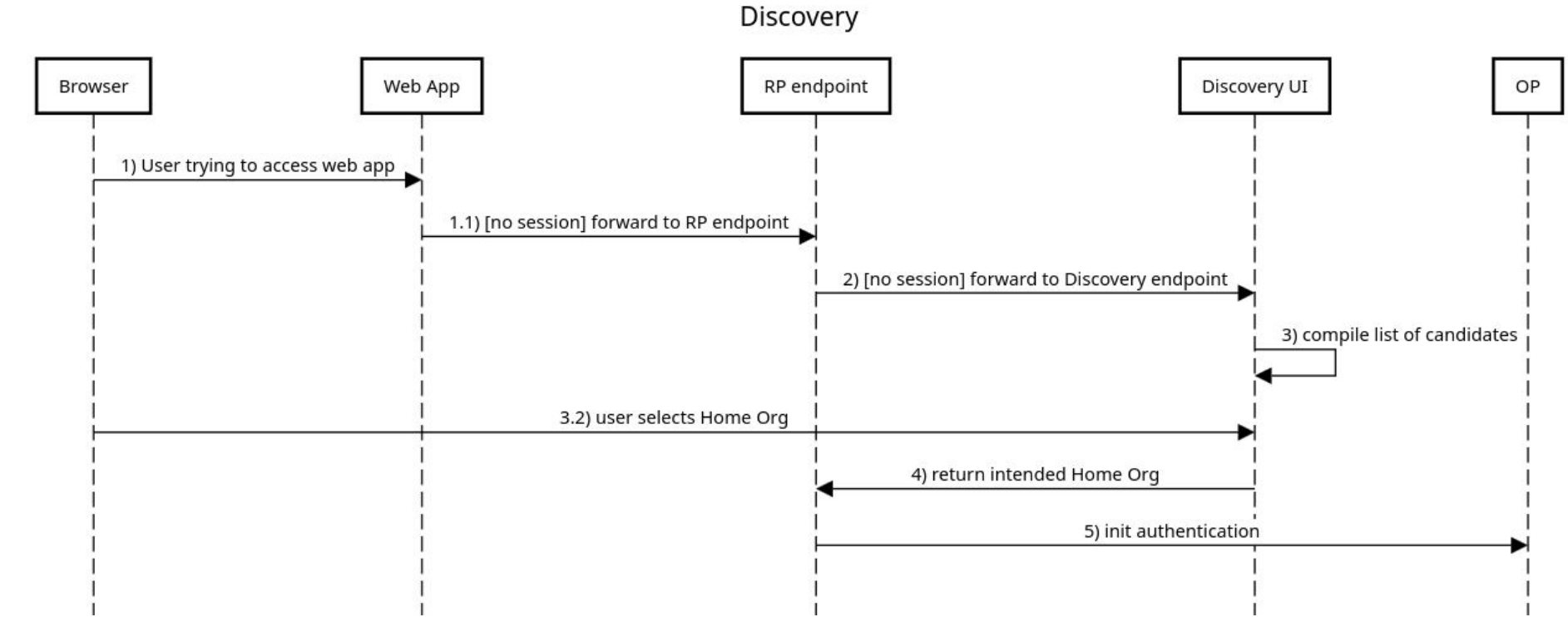
Local discovery for OIDFed

Mihály Héder

Local discovery for OIDFed

- Motivation
  - In a typical RP deployment, OP selection is the first GUI element the user experiences
  - Often this is the only visible part of the RP stack itself
  - Deployers will want to have this branded & nice UX
  - Deployers will want their own caching and even business logic
  - Separation of concern, modularisation
    - Let's not make this the burden of the RP implementer necessarily
  - Let's use the **same flow for centralized discovery** (like SeamlessAccess, as seen in cycle 10), and **local**
- **More on that at TIIME 13 Friday at Discovery section**

Suggested flow



https://docs.google.com/document/d/1heW6U7INvlj-FEOOKjCQ33-ifOyhlDXlnlQo-RvfcdU/edit?usp=sharing

Apache mod_oidfed (TIM student topic)

András Bodor

Apache mod_oidfed - Motivation

- Different approach from auth_memcookie (was demoed: Cycle 10)
- Apache httpd is widely used
  - Reverse proxy, HTTP server, etc.

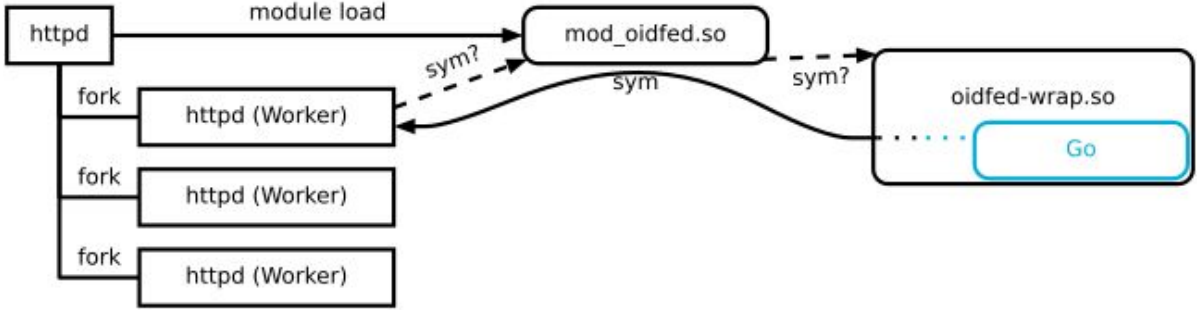
- Built-in module to act as RP in a federation
  - Anything behind httpd can implicitly enter the federation

=> mod_oidfed

Apache mod_oidfed - Architecture

- Wrapping the existing Go library
  - Allows reusing of existing federation logic
  - Considerably faster development
  - Compatibility for free
  - Slightly larger resource usage than pure-native (~15 MiB/Worker)
  - Have to be especially careful with runtime clashes (Go vs Apache)

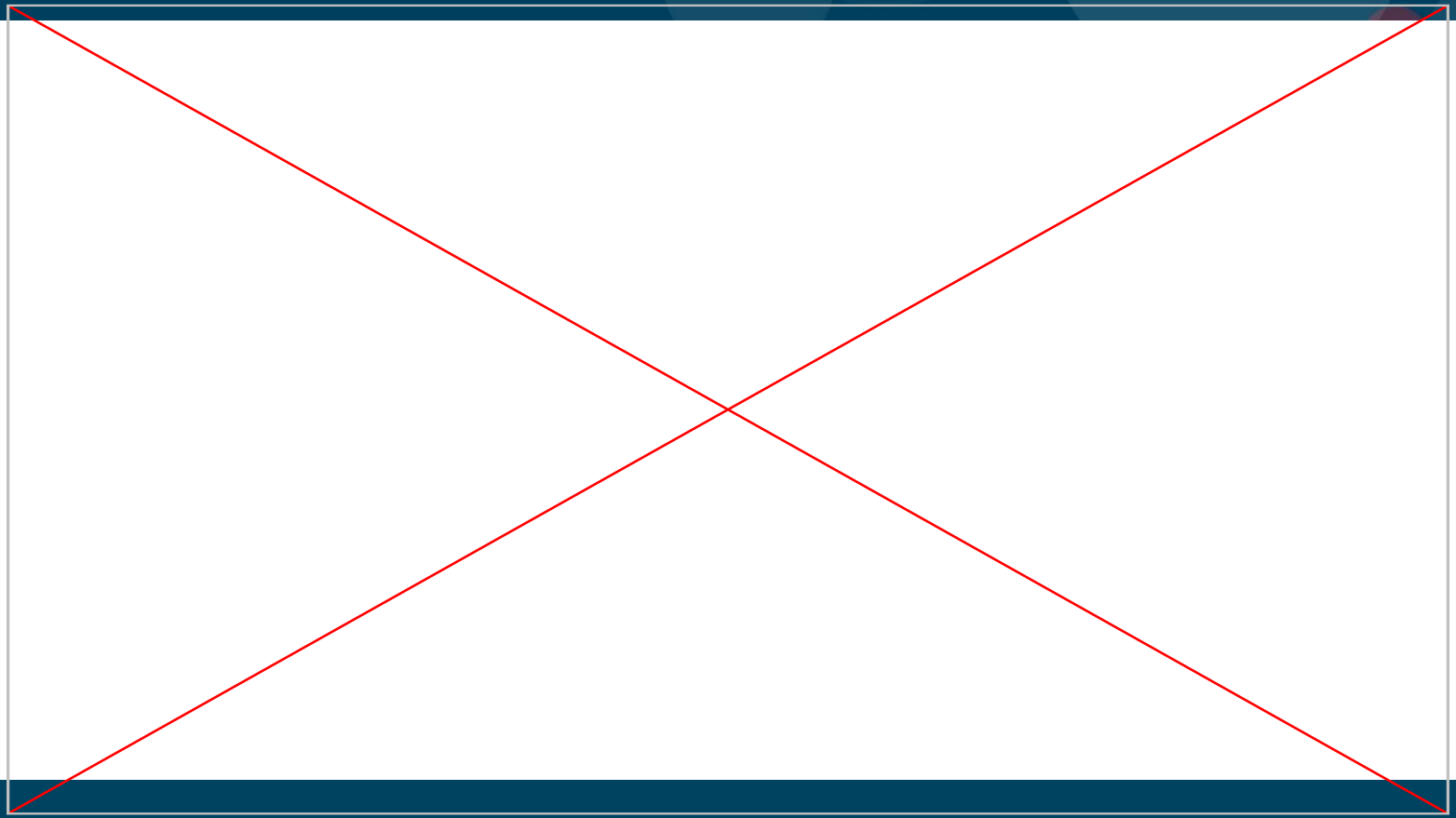
=> Each worker manually handles symbol resolution



Apache mod_oidfed - Status quo

- Federation entity configuration from Apache httpd.conf
  - e.g. `OidfedEntityId` https://rp.example.com
- `/.well-known/openid-federation` endpoint
  - Module automatically registers and handles this path
- Apache authn module integration
- *Work in progress:*
  - *Login page (OP selection)*
- *Demo!*

Apache mod_oidfed - Demo



OpenID Federation Registry

Gabriel Zachmann
Amin Mahnamfar
Niels van Dijk
Alexandr Petrunin
Halil Adem
Kushal Das

OIDFed Federation Admin API

- OpenAPI spec: https://gitlab.software.geant.org/TI_Incubator/federation-admin-api/-/blob/main/federation_admin_openapi.yaml?ref_type=heads
    - Fairly complete, but not considered final
- The API covers relevant actions a federation administrator wants to do.
- Usage:
    - [nerds] from the command line (e.g. curl)
    - Swagger UI
    - Proper frontend

OIDFed Federation Admin API - Content

- Management of the Federation Authority (itself)
  - Authority Hints
  - Management of Signing Keys
  - Metadata
  - Published Trust Marks
  - Allowed Trust Marks, Trust Mark Issuers & Trust Mark Owners
  - Additional Entity Configuration Claims
  - Entity Configuration Lifetime
- Subordinates
  - General and individual Metadata Policies
  - General and individual Constraints
  - General and individual additional claims for subordinate statements
  - Management of Subordinate’s JWKS
  - Metadata
  - Lifecycle Management of Subordinates
- Trust Mark Issuance

LightHouse v0.20.0

- New “major” version of LightHouse
  - Major changes
  - Version jump to 0.20.0
- SQL Databases instead of document storage
- HTTP API to manage LightHouse
- New capabilities, like individual Metadata Policies per entity
- There will be breaking changes, but a smooth upgrade path will be provided
  - Config file migration
  - DB migration

OIDFed Federation Admin API - LightHouse Implementation State (v0.20.0)

- Management of the Federation Authority (itself)
  - Authority Hints
  - Management of Signing Keys
  - Metadata
  - Published Trust Marks
  - Allowed Trust Marks, Trust Mark Issuers & Trust Mark Owners
  - Additional Entity Configuration Claims
  - Entity Configuration Lifetime
- Subordinates
  - General and individual Metadata Policies
  - General and individual Constraints
  - General and individual additional claims for subordinate statements
  - Management of Subordinate's JWKS
  - Metadata
  - Lifecycle Management of Subordinates
- Trust Mark Issuance

Done

TODO

Ongoing

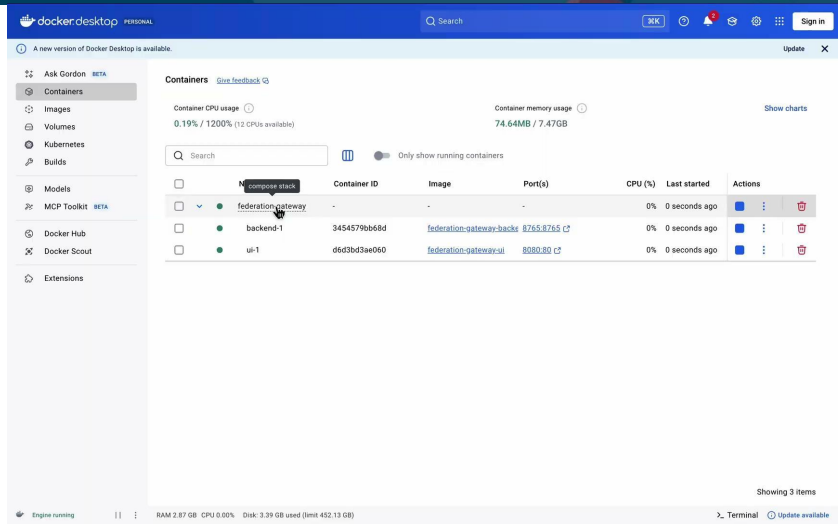
OIDFed Registry GUI

Usability prototype:

- Trust Anchors & Intermediates - CRUD
  - National level
  - Superior level
  - Subordinate level
- Leaf Entities - registration flow

Future work:

- Wire to Admin API backend implementation
- Add fine-grained RBAC
- Support Multi-tenant contexts



OIDFed Federation Admin API - Inmor implementation

OIDFed Federation Admin API - Inmor implementation

- Inmor Admin API Reference
  https://inmor.readthedocs.io/en/latest/api/admin.html

Future work

TI Incubator Call for Ideas:

https://wiki.geant.org/spaces/G52W5/pages/927596665/TII+call+for+Ideas

Final Demo: in May

TIM, TIIME

Thank you

Any questions?