Interoperability testing proposal

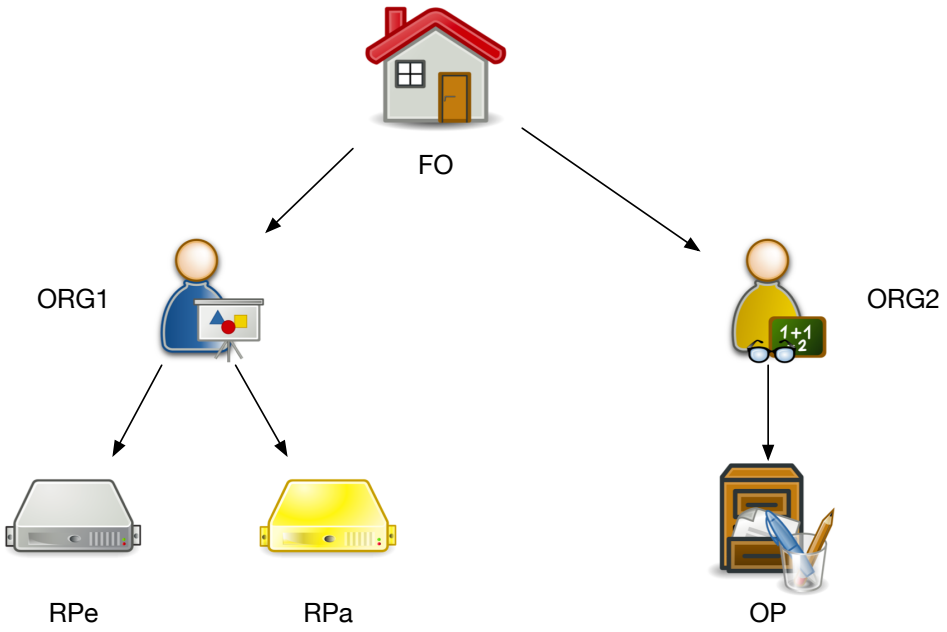
Participants

Henri Mikkonen Shibboleth/GEANT
Vladimir Dzhuvinov, Connect2id
Marko Ivancic, Connect2id
Jouke Roorda, NIKHEF
Roland Hedberg, Catalogix/IdentityPython

Basic setup

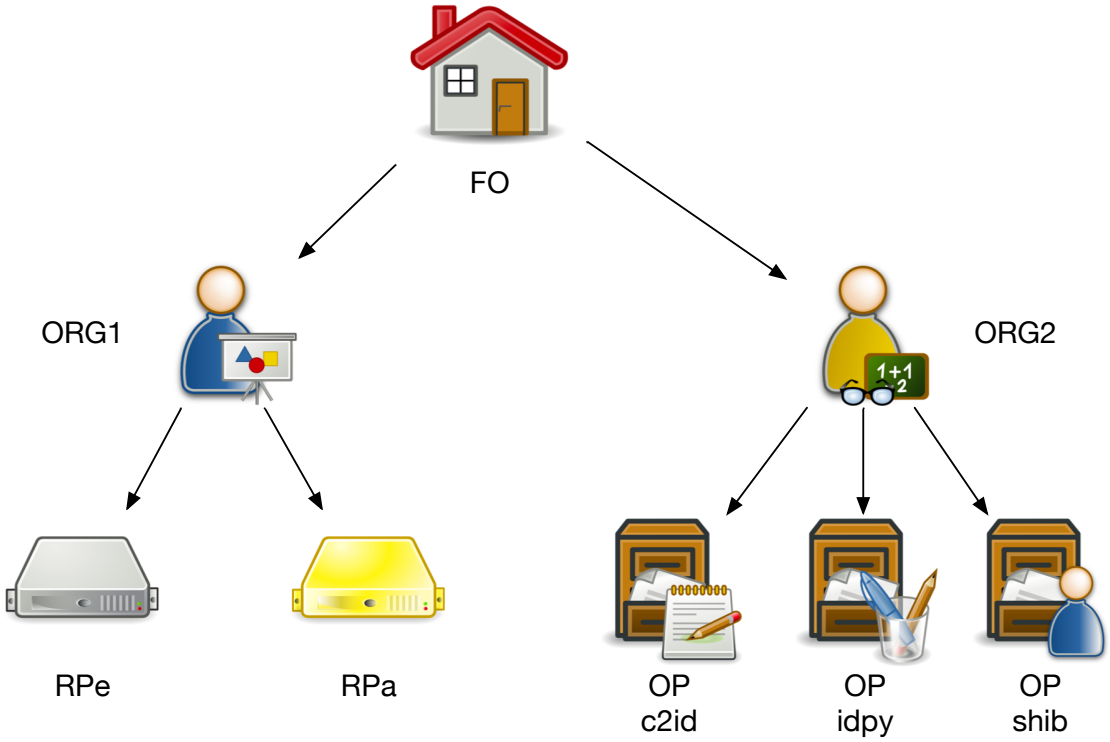
The basic setup (provided by me Roland Hedberg) contained one federation with the following entities:

- An RP that does explicit registration (RPe)
- An RP that does automatic registration (RPa)
- An OP that can handle explicit and automatic registration
- 2 intermediates representing 2 organisations (one owning the RPs and the other the OP)
- A federation operator (FO)



To this framework it was then possible to add the OPs and RPs the test participants provided. In this case 2 more OPs.

Which gave us this setup.



Goal of the exercise

The goal for this test event was to successfully perform a complete OIDC process between an RP and an OP.

Comprising of, in the case of explicit client registration :

- **Provider info discovery**
- **Client registration**
- Authorization request
- Token request
- Userinfo request

or when doing automatic client registration:

- **Provider info discovery**
- **Authorization request**
- Token request
- Userinfo request

The portions of the processes that are touched by the federation specification is marked in bold.

Results

OP-c2id

Using RPe against OP-c2id we where able to do the initial provider info discovery after that we got stuck when OP-c2id was doing a GET on RPe's .well-known/openid-config URL. OP-c2id complained about not being able to verify the SSL certificate (which was correct since it was a self-signed certificate). Since C2id was not able to turn off client certificate verification we couldn't get any further.

Roland took as an action to get a real certificate for the framework.

OP-shib

We used RPa against OP-shib trying to do automatic client registration.

Here the initial provider info discovery worked but then RPa could not figure out that it was supposed to use automatic client registration. The OP should publish that it could do automatic by adding **automatic** to *federation_types_supported* and if it supported client authentication by using the request object setting *automatic_registration_client_authn_methods_supported* to **request_param** in the metadata. This was hard to know since that client auth method was not described in the specification. Also the RP should publish that it supported automatic client registration by setting *federation_type* to **automatic** which it did.

Next step here was for Henri to make sure the OP published the appropriate metadata.

OP-idpy

Does both explicit and automatic client registration together with RPe and RPa. Not surprising since it's the same developer for all of them. The real test comes when we have another RP or other intermediates present.

Next step

I will continue to keep the framework up and running to allow us to reach the goal of the exercise.

I have created a Slack channel to allow for quick question/reply feedback.