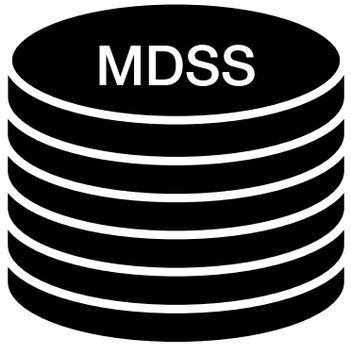
Doing federation the ‘SWAMID’ way

roland @ TNC18

Participants

Federation Operator

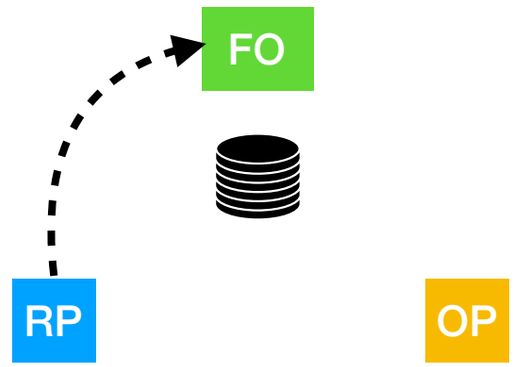


RP

OP

The same enrollment process for both OPs and RPs.

Enrollment

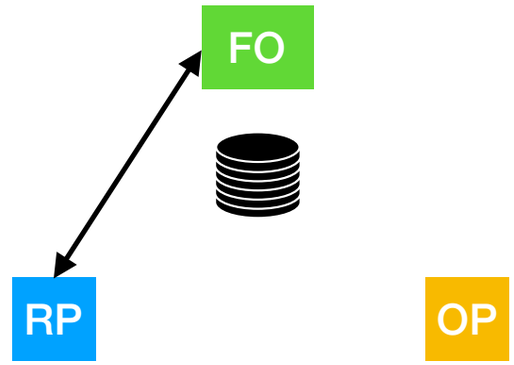


- The enrollment starts with the entity admin transmitting a set of information about the entity to the federation operator.

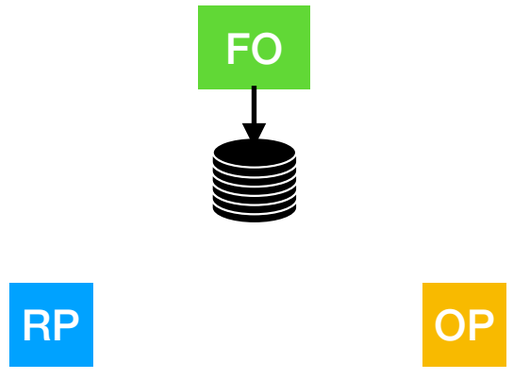

```
{
   “entity_id”: “https://example.com/rp",
   “signing_key”: {
     “keys”:[{
        "kty": "EC",
        "use": "sig",
        "kid": "1iHeY0a7zxaY_GTCci5VZBnYSMWA1k-Y2IAAxQQC_dc",
        "crv": "P-256",
        "x": "JzlE-MlpYdCxbqp5e3EHQcjE1lhL4ugmz-ICPOVYM9I",
        "y": "EnwwVXjphv4ANF1PfF8-6NmgjZ8Jbu91zNy61W_NAVI"
      }]},
   “metadata_endpoint”: “https://example.com/rp"
}
```

Operator processing

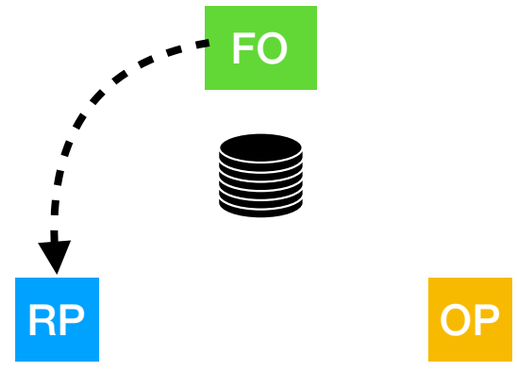
The federation operator (FO):



1. fetches the entity's metadata
2. verifies that the entity metadata doesn't conflict with the Federations policy.
3. may add claims to the metadata
4. sends the metadata to the MDSS
5. sends a enrollment acknowledgement to the entity admin.



Enrollment response




```
{
   “sms_collection_endpoint”: “https://example.com/mdss/sms_col“,
   “signing_keys”: {
      “keys”:[{
         "kty": "EC",
         "use": "sig",
         "kid": "RmRpSU5zWHhQSHdVdzVKVXlkZWJNa1lncERPcnZSekVRNGhla2lwSktPSQ",
         "crv": "P-256",
         "x": "nTzKK5hnhI7760lZUPjySdOldw4Q71VuQUfHUBAb4FQ",
         "y": "qtB-R5H5pzKSsbDBq0DhweD1oS2EWeGaa00p-tTarE4"
      }]},
   “signing_keys_url”: “https://fo.example.com/keys.jwks“
}
```


- The response contains the MDSS endpoint, from where the metadata statement collection can be fetched.
- It also contains the public parts of the Federation Operators signing keys.
- Pointer to where the MDSS’s signing keys can be found.

MDSS processing

- The mdss adds it's signed metadata statements to the entity metadata

- Signs the resulting metadata statement

```
FO(ms_ent + SK[ms_ent) + EDUGAIN(ms_FO + SK[FO]))
```

- And places it in the out-queue

By reference or by value

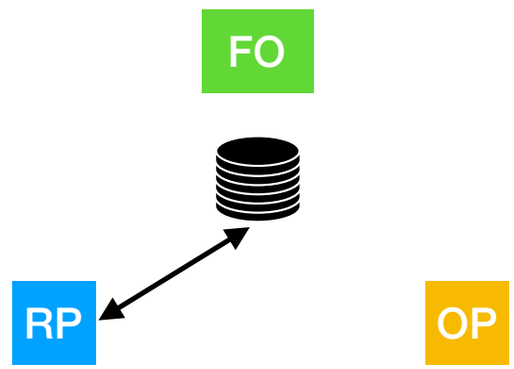
```
By value:
FO(ms_ent + SK[ent] + EDUGAIN(ms_FO + SK[FO]))

By reference:
FO(ms_ent + SK[ent] + url_EDUGAIN)
```

The metadata statement collection

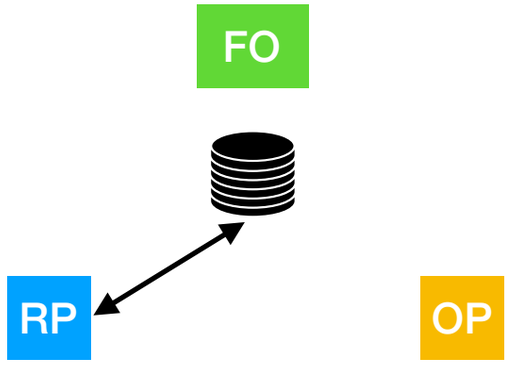
- How to get it
  GET */getsmscol/{context}/{entityID}*

- Format




```
{
   "https://swamid.sunet.se/":
      “https://mdss.sunet.se/getsms/
         registration/
         https%3A%2F%2Frp.example.com%2F/
         https%3A%2F%2Fswamid.sunet.se%2F”,
   “https://edugain.org/":
      “https://mdss.sunet.se/getsms/
         registration/
         https%3A%2F%2Frp.example.com%2F/
         https%3A%2F%2Fedugain.org%2F”
}
```

How to construct a ‘federation’ message



1. Construct a normal OIDC message
2. Make a copy of (1)
3. Fetch the collection of signed metadata statements from the MDSS
4. Per signed metadata statement
    1. Add the signed metadata statement to the copy
    2. Add own signing keys to the copy
    3. Construct a signed JSON Web Token (JWS) with the copy as body
    4. Add the resulting JWS as a metadata_statement to (1).
5. The result is the federation message.

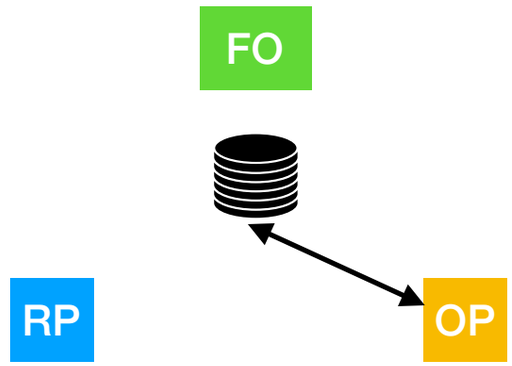
ent(ms$_2$_ent + SK[ent] + url_EduGAIN) => SMS$_1$

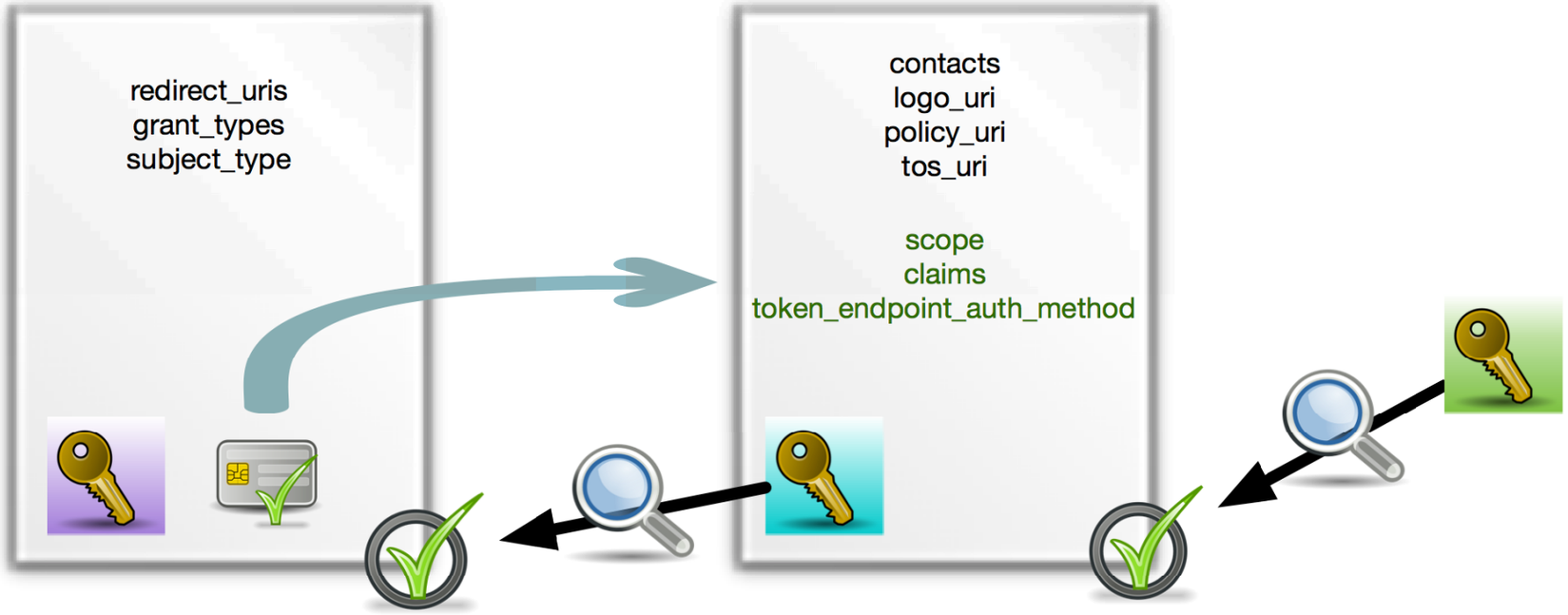
ent(ms$_2$_ent + SK[ent] + url_SWAMID) => SMS$_2$

ms$_2$_ent + SK[ent] + SMS$_1$ + SMS$_2$

```
{
   "redirect_uris": [“https://rp.example.org/86701938519566083 96"],
   "metadata_statements": {
        "https://swamid.sunet.se/":
“eyJhbGciOiJFUzI1NiJ9.eyJyZWRpcmVjdF91cmlzIjogWyJodHRwczovL3JwLmV4YW1wbGUub3JnLzg2NzA
xOTM4NTE5NTY2MDgzOTYiXSwgIm1ldGFkYXRhX3N0YXRlbWVudF91cmkiOiB7Imh0dHBzOi8vZm8uZXhh
bXBsZS5lZHUvIjogImh0dHBzOi8vbWRzcy5mby5leGFtcGxlLmVkdS9nZXRtcy9odHRwcyUzQSUyRiUyRnJw
LmV4YW1wbGUuY29tJTJGbXMuandzL2h0dHBzJTNBJTJGJTJGZm8uZXhhbXBsZS5lZHUlMkYifX0.y73e9d
6Yr6JqaG9iss6GBcudFskHcRCBn6gYD8XW0TqS88b4ELh_G7M5GvTXbeDZ4wU7w-ZViP7srt1htG7HAQ”,
        “https://edugain.org/“:
”eyJhbGciOiJFUzI1NiIsImtpZCI6Ik1uVnRaWEJRTVZoWmRGTkpZMkpVZW5kNVdHOUdNWFpOUkZVelFXc
3RORGMzTWpkNmN6aDFUR2t0Y3cifQ.eyJpc3MiOiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9ycCIsICJpYXQiOi
AxNTI4OTU1NDMzLCAiYXVkIjogWyIiXSwgInJlZGlyZWN0X3VyaXMiOiBbImh0dHBzOi8vcnAuZXhhbXBsZS5
vcmcvODY3MDE5Mzg1MTk1NjYwODM5NiJdLCAic2Inbmluz19rZXlzIjogeyJrZXlzIjogW3sia3R5IjogIkVDIiwgI
nVzZSI6ICJzaWciLCAia2lkIjogIlJtUnBTVTV6V0hoUVNIZFZkelZLVlhsa1pXSk5hMWxuY0VSUGNuWlNla1ZST
kdoSWEySXdTa3RQU1EiLCAiY3J2IjogIlAtMjU2IiwgInngiOiAibIR6S0s1aG5oSTc3NjBsWlVQanlTZE9sZHc0UT
cxVnVRVWZlVUJBYjRGUSIsICJ5IjogInF0Qi1SNUg1cHpLU3NiREJxMERod2VEMW9TMkVXZUdhYTAwcC10
VGFyRTQifV19LCAibWV0YWRhdGFfc3RhdGVtZW50cyI6IHsiaHR0cHM6Ly9lZHVnYWluLm9yZy8iOiAiaHR0
cHM6Ly9tZHNzLnN1bmV0LnNlL2dldHNtcy9yZWdpc3RyYXRpb24vaHR0cHMlM0ElMkYlMkZycC5leGFtcGx
lLmNvbSUyRi9odHRwcyUzQSUyRiUyRmVkdWdhaW4ub3JnJTJGIn0sICJraWQiOiAiTW5WdFpYQlFNVmha
ZEZOSlkySlVlbmQ1V0c5R01YWk5SRlV6UVdzdE5EYzNNamQ2Y3poMVRHa3RjdyJ9.Ex1MM3LDe33JoFDW
Obd46IkP-yLiCPjsaEXaMC-G5NuGppuLKUjkeZRglrwsto4kivwBUhV7UdI1YMutifwLsQ"
    }
}
```

Unpacking a metadata statement





The main points

The FO:

- controls who get to be in the Federation
- fetches the metadata from the entity
- verifies/modifies the metadata before signing/publishing
- publishes the signed metadata