ESCAPE T2.5:
Authentication and Authorization

Andrea Ceccanti
INFN CNAF

ESCAPE WP2 meeting

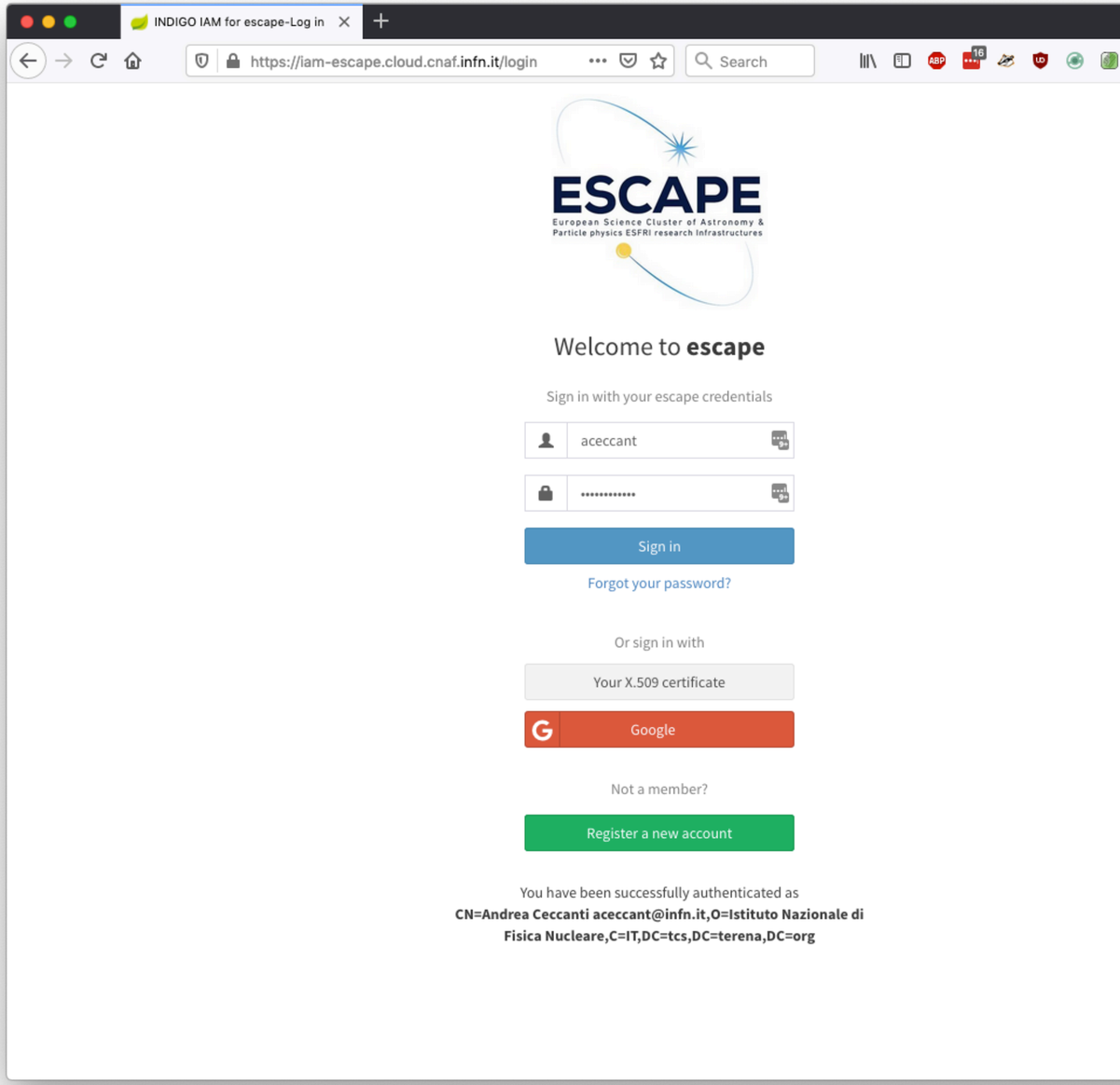
December, 4th 2019

Status of the testbed

ESCAPE IAM instance

Escape IAM instance available

- 35 registered users
- AuthN with X.509 certificates, Google and username/password
  - EduGAIN coming too…
- VOMS endpoint available
- Registration open
- Documentation available here



WLCG JWT profile implementation

WLCG JWT profile has reached v1.0

IAM **implements it today** in the latest development branch

- Already deployed for the WLCG IAM instance

The Escape IAM instance will soon be upgraded to the latest version

Example WLCG JWT access token:


```
{
  "wlcg.ver": "1.0",
  "sub": "a1b98335-9649-4fb0-961d-5a49ce108d49",
  "scope": "openid wlcg.groups profile",
  "iss": "https://wlcg.cloud.cnaf.infn.it/",
  "exp": 1571989553,
  "iat": 1571985953,
  "jti": "a535baea-0f7b-461e-a5de-cd7bceb8be3c",
  "wlcg.groups": [
    "/wlcg"
  ]
}
```

AuthN/Z on the Datalake testbed

Currently mostly using GSI for authentication and VOMS for authorization

Token-based AuthN/Z already working for some storage elements:

- StoRM WebDAV @ INFN CNAF
- dCache Prometheus instance

DEMO

Token-based AuthN/Z for RUCIO managed data xfers

Lots of work and discussions in the past months on how to enable X509-free data transfers managed by RUCIO

A document is being prepared in the DOMA TPC WLCG context describing the flows used to request and exchange tokens with IAM and across services

Token-based AuthN/Z for RUCIO managed data xfers

In this scenario,
RUCIO delegates its identity to FTS to manage a third-party data transfer between SE 1 and SE 2

Token-based AuthN/Z for RUCIO managed data xfers

RUCIO gets a token from IAM using the OAuth **client_credentials** grant type. The token needs to provide the minimum privileges need to interact with FTS

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

se1.example

SE 1

Token request

```
POST /token HTTP/2
Host: iam.example
Authorization: Basic ZG...B
Accept: */*
Content-Length: ...
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&scope=fts:submit-transfer
&audience=fts.example
```

IAM

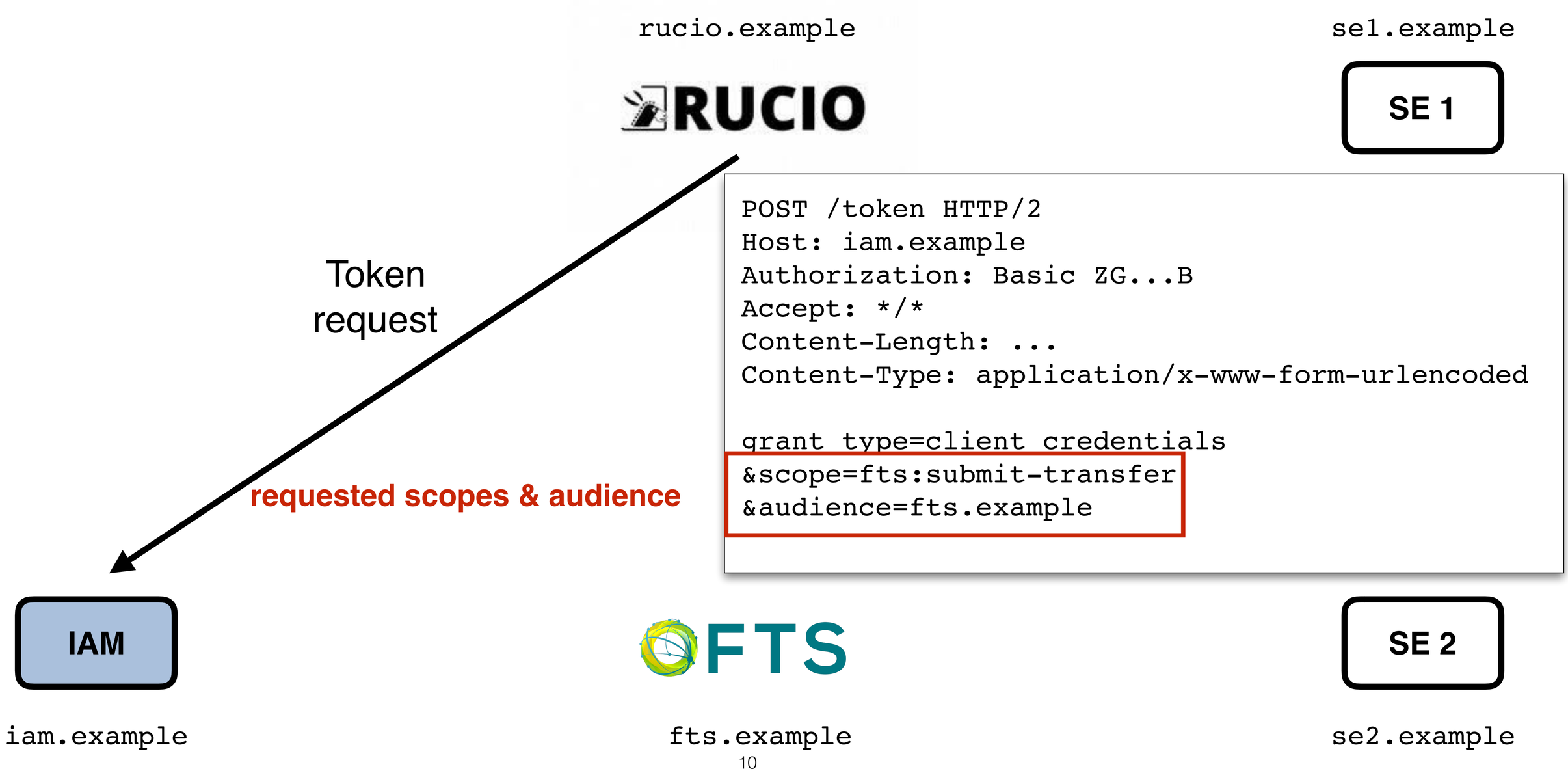
SE 2

iam.example

fts.example

se2.example

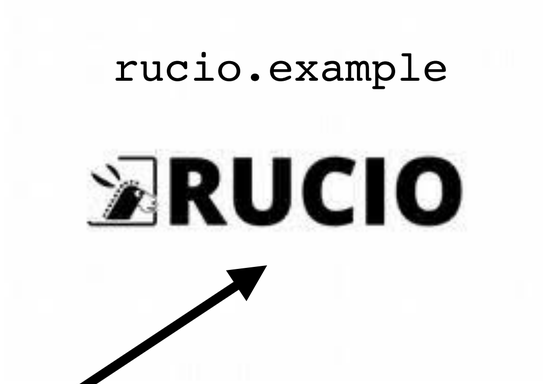
Token-based AuthN/Z for RUCIO managed data xfers



Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

se1.example



SE 1

Token response

```
{
  "access_token": “eyJra…HvBfTpM”,
  "token_type": "Bearer",
  "expires_in": 3599,
  "scope": “fts:submit-job”
}
```

IAM

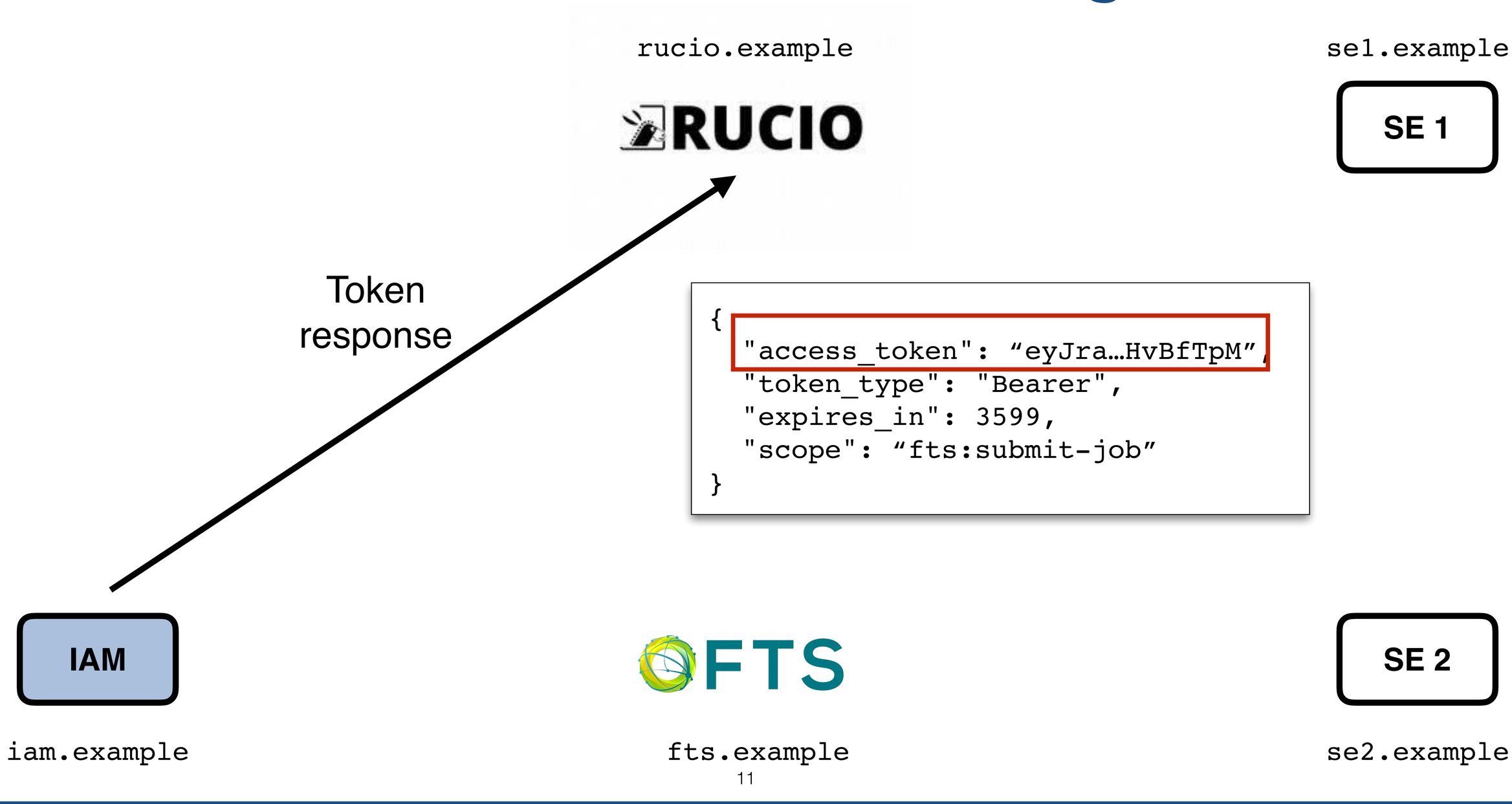
SE 2

iam.example

fts.example

se2.example

Token-based AuthN/Z for RUCIO managed data xfers



Token-based AuthN/Z for RUCIO managed data xfers

Rucio extracts the access token from the response, and stores it in local memory.

rucio.example

se1.example

SE 1

```
{
  "access_token": “eyJra…HvBfTpM”,
  "token_type": "Bearer",
  "expires_in": 3599,
  "scope": “fts:submit-job”
}
```

parse & validate JWT

access token body:

```
{
  "sub": "rucio.example",
  "aud": "fts.example",
  "nbf": 1572840340,
  "scope": "fts:submit-transfer",
  "iss": "https://iam.example/",
  "exp": 1572843940,
  "iat": 1572840340,
  "jti": "be48f2ab-8dd9-4df2-ae0b-bcb1fdfafaa6"
}
```

IAM

iam.example

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

The token audience is limited to FTS, and the requested scope has been granted.

rucio.example

se1.example

SE 1

access token body:

```
{
  "sub": "rucio.example",
  "aud": "fts.example",
  "nbf": 1572840340,
  "scope": "fts:submit-transfer",
  "iss": "https://iam.example/",
  "exp": 1572843940,
  "iat": 1572840340,
  "jti": "be48f2ab-8dd9-4df2-ae0b-bcb1fdfafaa6"
}
```

IAM

iam.example

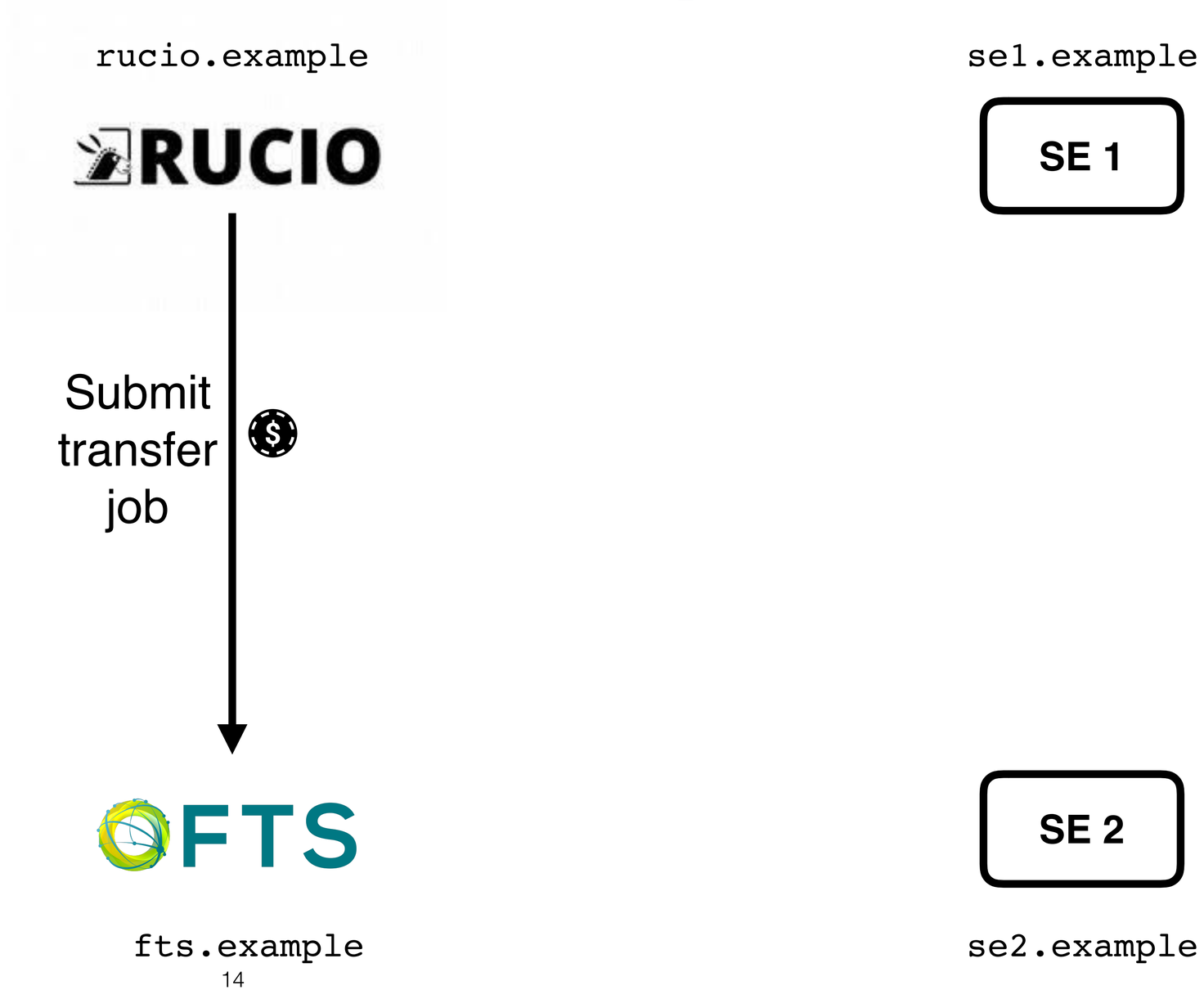
fts.example

SE 2

se2.example

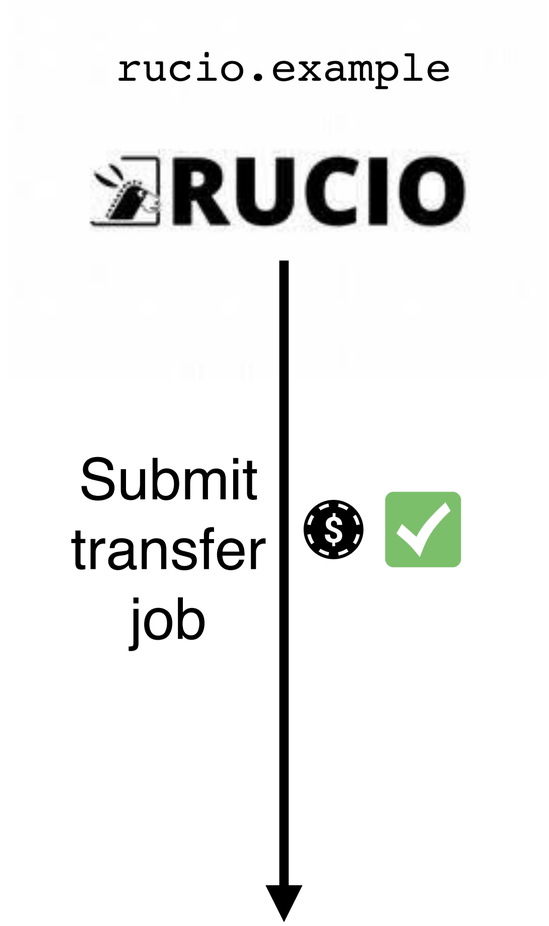
Token-based AuthN/Z for RUCIO managed data xfers

RUCIO submits a transfer job to FTS, including the token obtained from IAM in the request



Token-based AuthN/Z for RUCIO managed data xfers

FTS validates the token extracted from the request and accepts the transfer, assuming the token is valid and provides the necessary rights



SE 1

se1.example

IAM

iam.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

se1.example

SE 1

FTS now needs a token that will be used for AuthN/Z at the storage elements. In this scenario, FTS impersonates RUCIO.

IAM

iam.example

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

se1.example

SE 1

The token it already has cannot be used for the transfer:
it's scoped to fts.example and does not provide the necessary rights to read and store files
at storage elements

IAM

iam.example

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

se1.example

SE 1

FTS then **exchanges the obtained token** with a couple of tokens, an access token and refresh token, that will be used to manage the transfer

IAM

iam.example

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

FTS requests the following scopes:
storage.read:/
storage.write:/
offline_access

rucio.example

RUCIO

se1.example

SE 1

```
POST /token HTTP/2
Host: iam.example
Authorization: Basic u89…
Accept: */*
Content-Length: ...
Content-Type: application/x-www-form-urlencoded

grant_type=urn:ietf:params:oauth:grant-type:token-exchange
&subject_token=eyJra…HvBfTpM
&audience=se1.example%20se2.example
&scope=storage.read%3A%2F%20storage.write%3A%2F%20offline_access
```

Token exchange request

IAM

FTS

SE 2

iam.example

fts.example

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

The audience of the token is limited to only apply to the storage elements involved in the transfer

rucio.example

RUCIO

se1.example

SE 1

```
POST /token HTTP/2
Host: iam.example
Authorization: Basic u89…
Accept: */*
Content-Length: ...
Content-Type: application/x-www-form-urlencoded

grant_type=urn:ietf:params:oauth:grant-type:token-exchange
&subject_token=eyJra…HvBfTpM
&audience=se1.example%20se2.example
&scope=storage.read%3A%2F%20storage.write%3A%2F%20offline_access
```

Token exchange request

IAM

FTS

SE 2

iam.example

fts.example

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

IAM validates the token exchange request, and assuming there's a policy that authorizes the exchange, issues the requested tokens

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

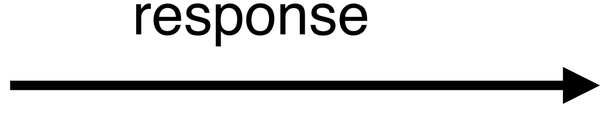
se1.example

SE 1

```
{
  "access_token": “e7nd…HvBfTpM”,
  “refresh_token": “9njuk…”,
  "token_type": "Bearer",
  "expires_in": 3599,
  "scope": “storage.read:/ storage.create:/ offline_access”
}
```

Token exchange response

IAM



FTS

SE 2

iam.example

fts.example

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

FTS extracts the tokens from the response and saves them locally

rucio.example

RUCIO

se1.example

SE 1

```
{
  "access_token": “e7nd…HvBfTpM”,
  “refresh_token": “9njuk…”,
  "token_type": "Bearer",
  "expires_in": 3599,
  "scope": “storage.read:/
      storage.create:/
      offline_access”
}
```

IAM

iam.example

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

The new access token can be refreshed from IAM with the **refresh_token** flow.

Refresh tokens are typically much longer lived than access tokens and

rucio.example

RUCIO

se1.example

SE 1

access token body:

```
{
  "sub": "rucio.example",
  "aud": “se1.example s2.example”,
  "nbf": 1572840345,
  "scope": “storage.read:/ storage.create:/
            offline_access”,
  "iss": "https://iam.example/",
  "exp": 1572843945,
  "iat": 1572840345,
  "jti": “be48…”
}
```

IAM

iam.example

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

FTS will enqueue the transfer job, and when the transfer is about to start can use the refresh token to get a fresh access token that will be used for the transfer.

Token-based AuthN/Z for RUCIO managed data xfers

FTS then submits the third-party transfer against SE 2, including the token in the request

Token-based AuthN/Z for RUCIO managed data xfers

The same token will be used for authn/z at se1 and se2.

It's also possible to have two separate tokens for each SE

se1.example

SE 1

```
COPY /example/file HTTP/2
Host: se2.example
Source: https://se1.example/example/file
Authorization: Bearer e7nd…
TransferHeaderAuthorization: Bearer e7nd…
```

Submit
third-party transfer

IAM

iam.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

SE2 will then use the obtained token for authn/z against SE1

Next steps

LOFAR EGI CheckIn integration

Integrate IAM with the EGI CheckIn managed LOFAR organization:

- Allow users to login in the ESCAPE VO using their LOFAR credentials
  - Integration already working on an IAM EOSC instance, soon will be active on the ESCAPE IAM instance
- Allow users to automatically onboard the ESCAPE VO (without administrator approval) when authenticated using their LOFAR credentials, and placed in an ESCAPE IAM lofar group
  - Requires some development on the IAM side, ETA: March 2020

Use this as a pilot experience to showcase how we can integrate and interoperate with other, standards-based AAI

Other next steps (taken from July slides)

Collect and **understand key AAI requirements** across the ESCAPE cluster

- How are users and agents authenticated?
- What's the authorization model? What's the delegation model? How are authorization privileges and policies managed?
  - **Focus on data access**
- What are the legacy auhtn/authz mechanisms that must be supported?

Agree on **a common way to express Authn/Auhtz information** and expose this information to services

- Start from the WLCG experience and expand/adapt it as needed

Understand what are the **key software components** that needs to be integrated

- and whether the integration requires changes in the software

Thanks for your attention.
Questions?

References

IAM @ GitHub: https://github.com/indigo-iam/iam

IAM documentation: https://indigo-iam.github.io/docs

WLCG Authorization WG: https://twiki.cern.ch/twiki/bin/view/LCG/WLCGAuthorizationWG

WLCG AuthZ WG Demos: https://indico.cern.ch/event/791175/attachments/1806605/2948665/demos.mp4 (IAM starts at minute 46)

IAM in action video: https://www.youtube.com/watch?v=1rZlvJADOnY

Contacts:

- andrea.ceccanti@cnaf.infn.it
- indigo-iam.slack.com