Authentication and Authorization in ESCAPE WP2

Andrea Ceccanti
INFN CNAF

WP5 Technical meeting

February, 3rd 2020

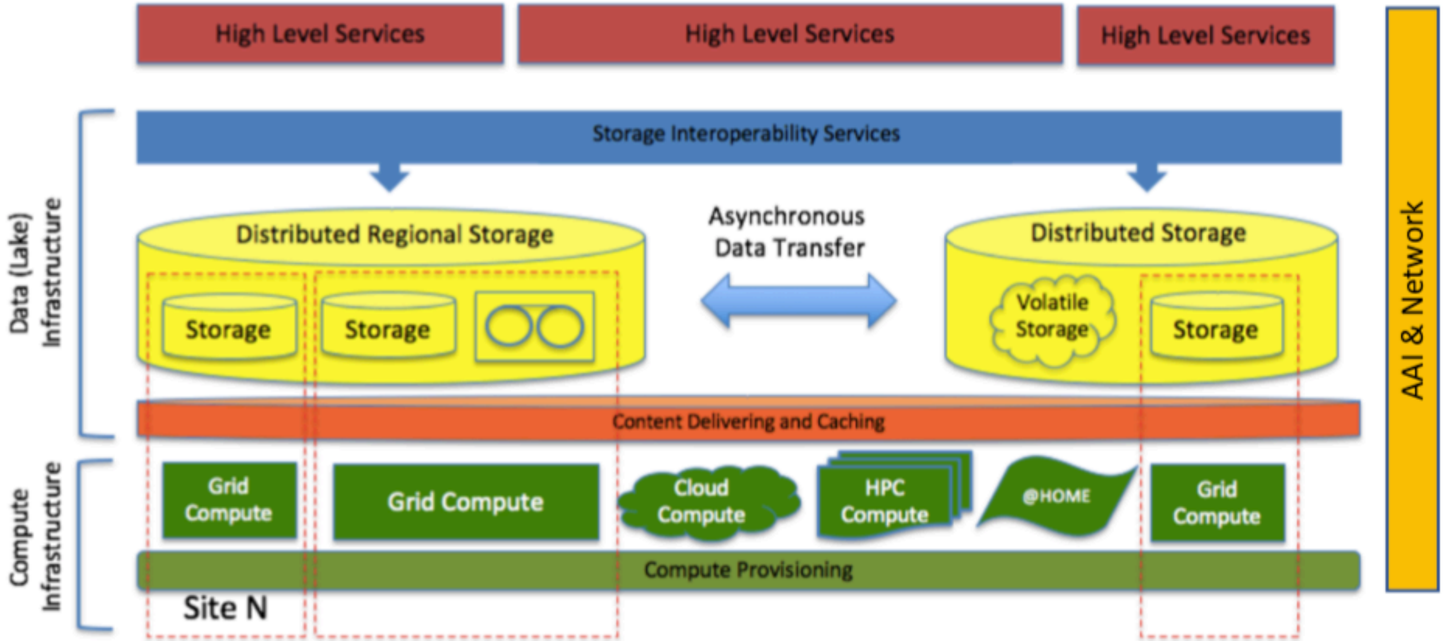
ESCAPE
European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures

INFN CNAF

Introduction

The ESCAPE Data lake

Data Lake building blocks

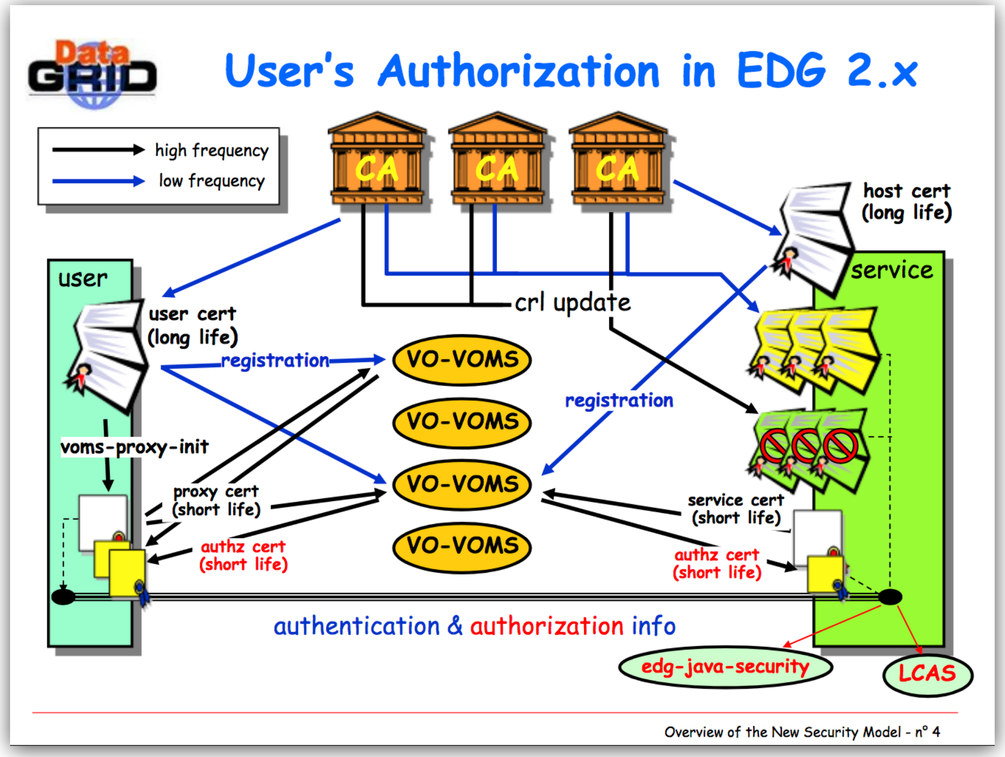


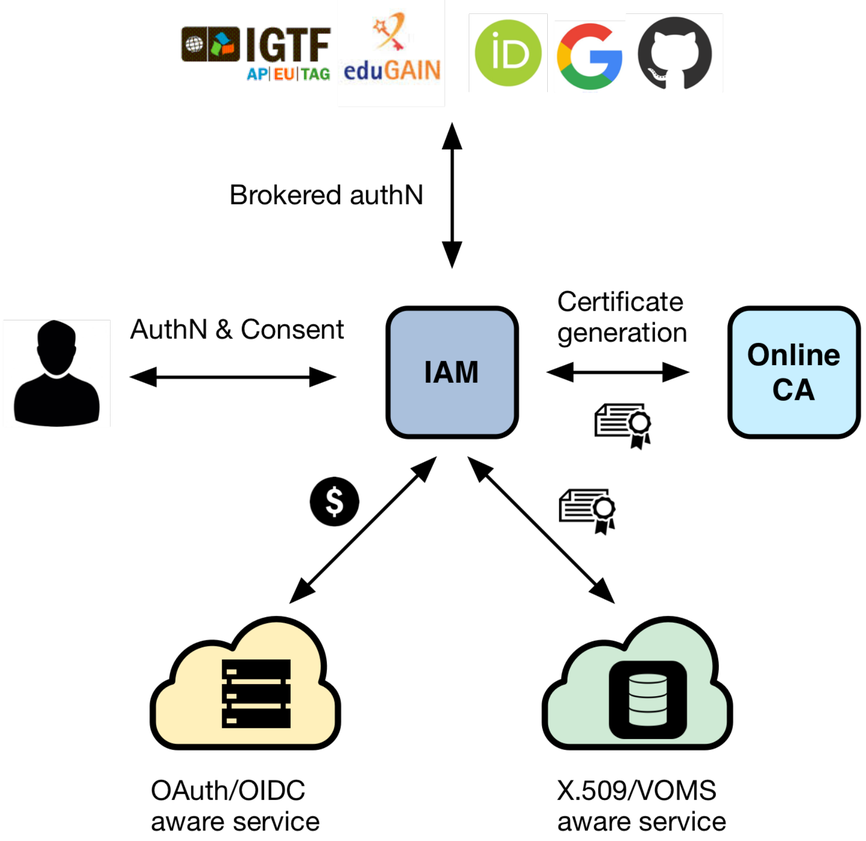
In synergy and complementing the work of WLCG DOMA: to define, integrate and commission an ecosystem of tools and services to build a data lake

Leaves to the science projects the flexibility to choose the services and layout most suitable to their needs. Provides a reference implementation

Contributes to deliver Open Access and FAIR data services: relies on trustable data repositories; enables data management policies; hides the complexities of the underlying infrastructure providing a transparent data access layer

ESCAPE Data Lake AAI and WLCG





Approach: leverage and build upon the WLCG experience

Moving beyond X.509: main challenges

Authentication

- **Flexible**, able to accomodate various authentication mechanisms
  - X.509, username & password, EduGAIN, …

Identity harmonization & account linking

- Harmonize multiple identities & credentials in a single account, providing a **persistent identifier**

Authorization

- **Orthogonal** to authentication, **attribute** or **capability-based**

Delegation

- Provide the ability for **services to act on behalf of users**
- Support for **long-running applications**

Provisioning

- Support provisioning/de-provisioning of identities to services/relying resources

Token translation

- Enable **integration with legacy services through controlled credential translation**

Moving beyond X.509: main challenges

Authentication

- **Flexible**, able to accomodate various authentication n
  - X.509, username

Identity harmon
linking

- Harmonize mult
  in a single accou
  identifier

Authorization

- **Orthogonal** to authentication, **attribute** or **capability-based**

Delegation

- Provide the ability for **services to act on**

g applications

provisioning of
ing resources

- Enable **integration with legacy services through controlled credential translation**

Key challenge:

allow a gradual transition to the new AAI!

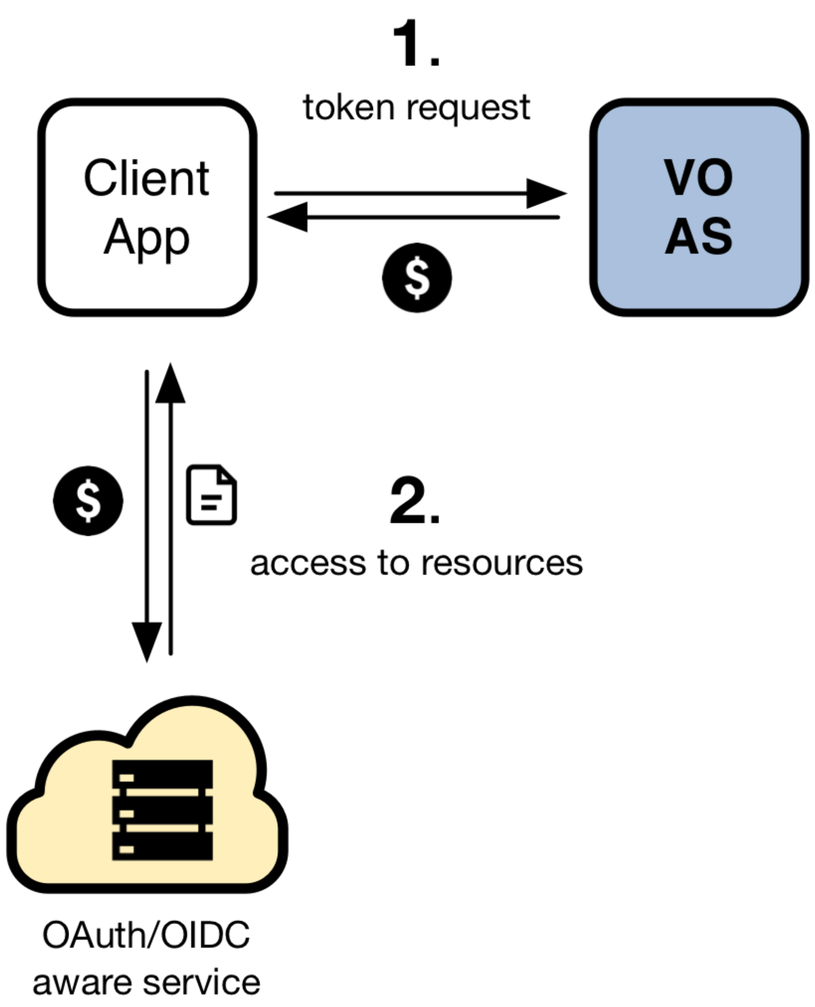
Token-based AuthN/Z from 10000 mt

In order to access resources/services, a **client application** needs an **access token**

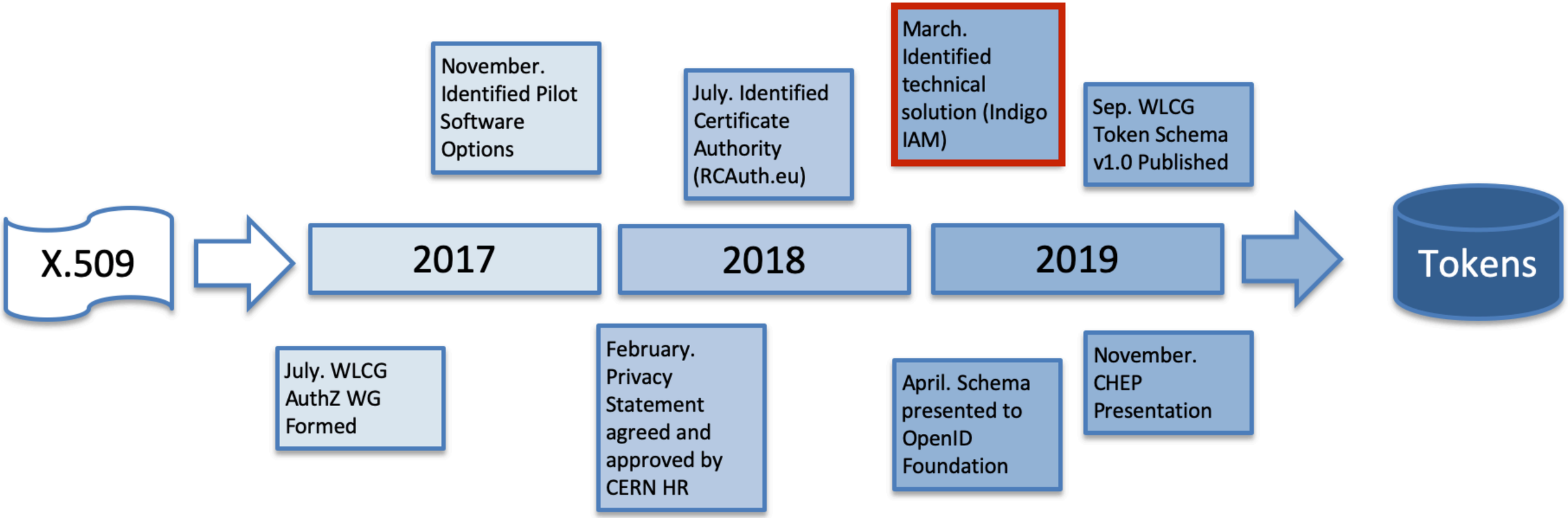
The token is obtained from **a Virtual Organization** (which acts as an OAuth Authorization Server) using standard **OAuth/ OpenID Connect** flows

Authorization is then **performed at the services** leveraging info extracted from the token:

- **Identity attributes**: e.g., **groups**
- **OAuth scopes**: capabilities linked to access tokens at token creation time



Towards Tokens

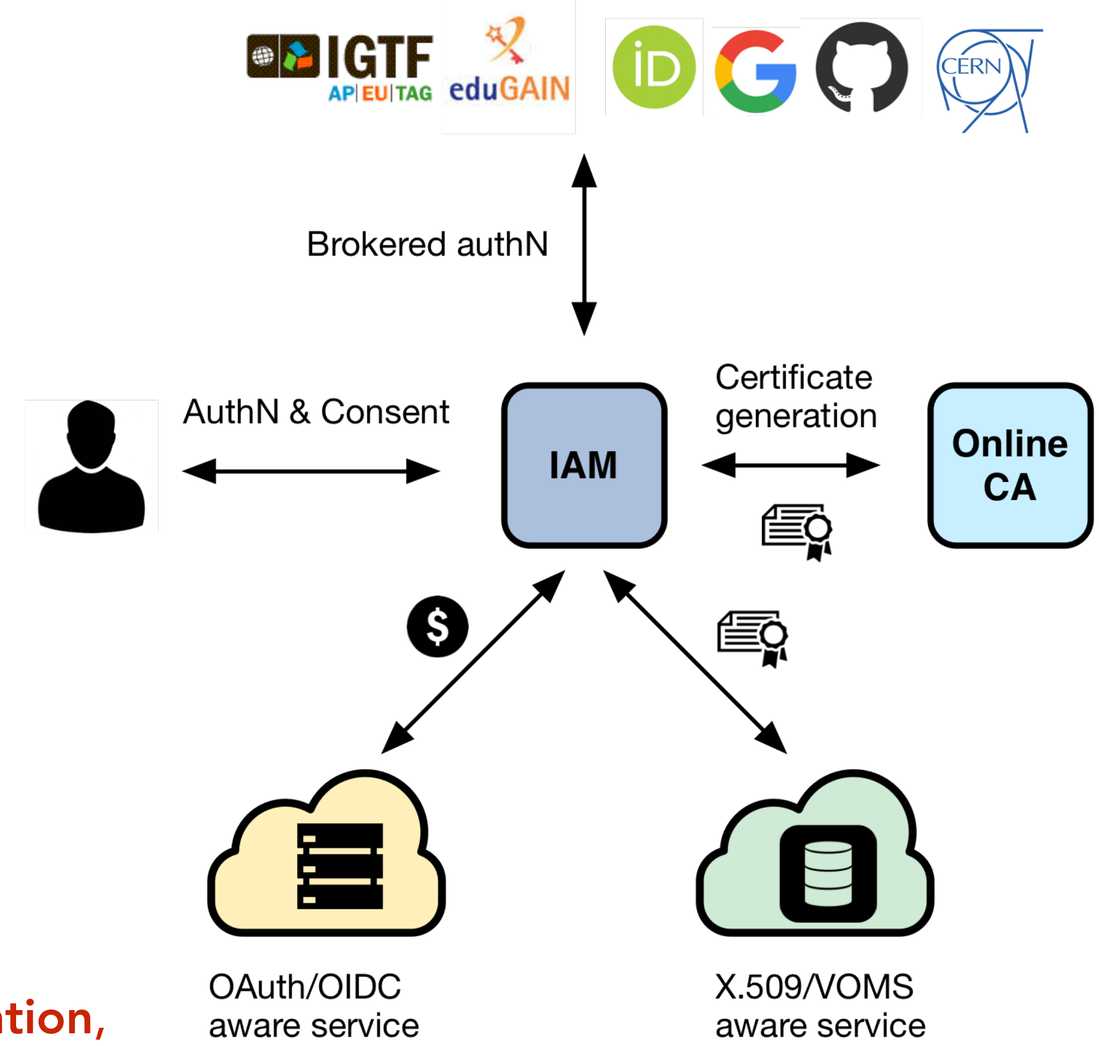


* Slide courtesy of H. Short

INDIGO Identity and Access Management Service

A **VO-scoped** authentication and authorization service that

- supports **multiple authentication mechanisms**
- provides users with a **persistent, VO-scoped** identifier
- exposes **identity information**, **attributes** and **capabilities** to services via **JWT** tokens and standard **OAuth & OpenID Connect** protocols
- can integrate existing **VOMS**-aware services
- supports **Web** and **non-Web access**, **delegation** and **token renewal**

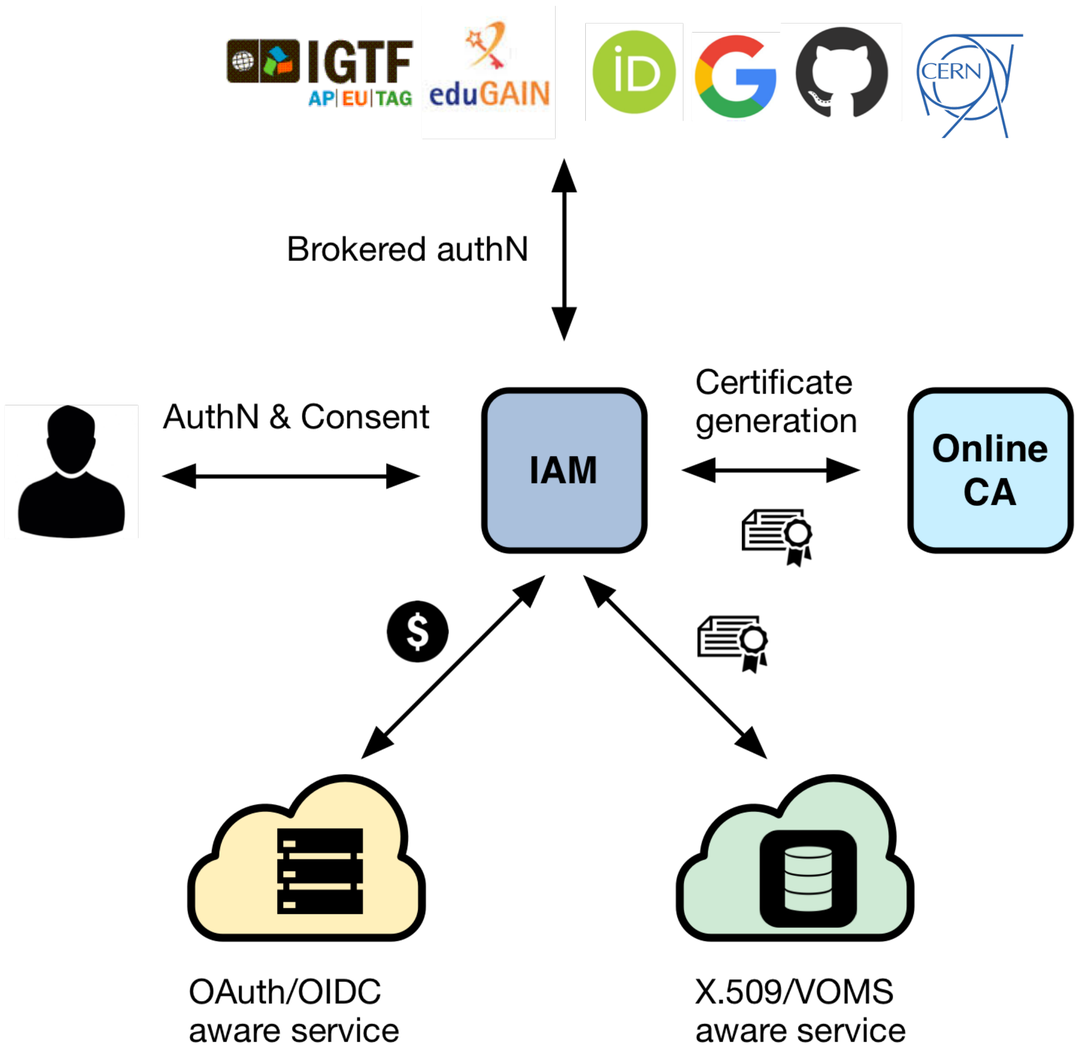


WARNING:
VO here means **Virtual Organization**, not Virtual Observatory

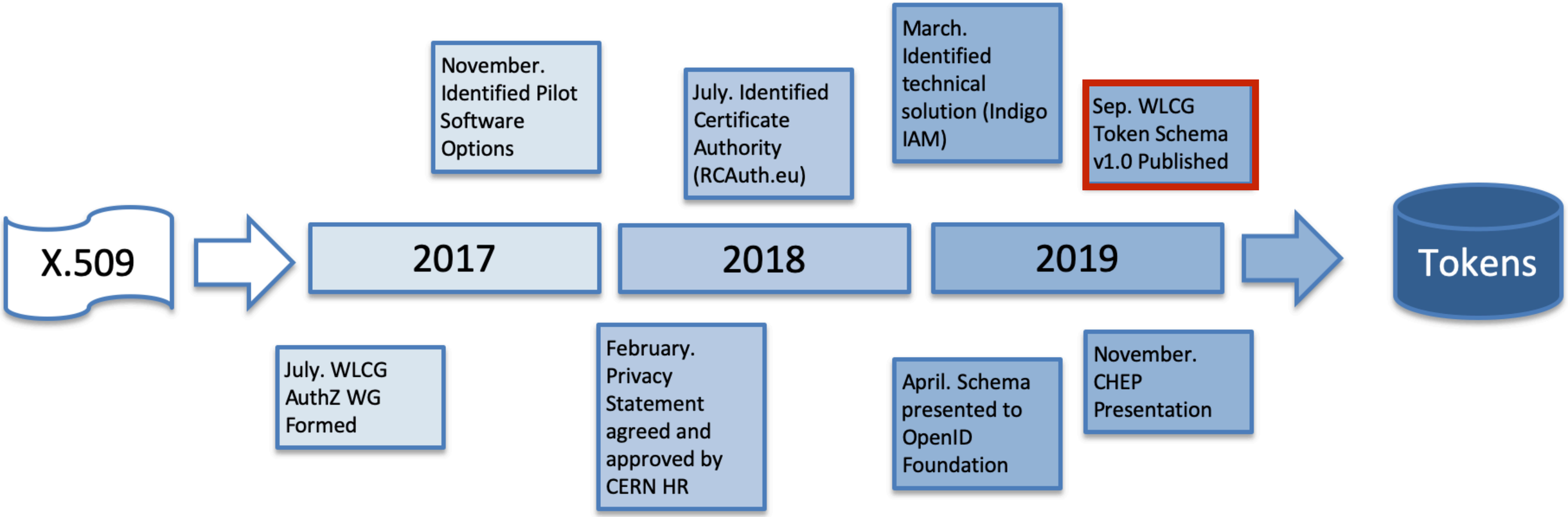
INDIGO Identity and Access Management Service

Selected by the WLCG MB to be the core of the future, token-based WLCG AAI

Sustained by INFN for the foreseeable future, with current support from:



Towards Tokens



* Slide courtesy of H. Short

The JWT common profile has reached v1.0

How is **authentication** and **authorization** information encoded in **identity** and **access tokens**?

How is **trust** established between parties exchanging tokens?

What's the recommended **token lifetime?**

Approach:

rely on existing standards as much as possible, extend only when needed

WLCG Token-based AuthN/Z Hackathon

Objectives: sort out as many problems as possible while discussing things and coding together in a room with the objective of demonstrating **a full stack HTTP X509-free data transfer management chain**

- RUCIO->FTS->SEs
- SEs: EOS, dCache, DPM, StoRM, XRootD, Echo

following the rules imposed by the WLCG Common JWT profile

When: January 16th, CERN

WLCG Token-based AuthN/Z Hackathon

Hackathon notes shared document

Scope-based WLCG JWT Interoperability Matrix

Below is the "interoperability table" between various implementations when tested with FTS and a WLCG JWT (using fine-grained scopes; streaming mode not counted as a success).

From; To->	Vanilla XRD	dCache	DPM	StoRM	EOS
Vanilla XRD			N/T		N/T
dCache			N/T		N/T
DPM	N/T	N/T			N/T
StoRM					N/T
EOS	N/T	N/T	N/T	N/T	N/T

Token-based auth/z flows

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

RUCIO

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

iam.example

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

se1.example

SE 1

Token request

requested scopes & audience

```
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

iam.example

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

se1.example

SE 1

Token response

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

IAM

iam.example

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

se1.example

SE 1

Token response

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

IAM

iam.example

FTS

fts.example

SE 2

se2.example

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

RUCIO

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"
}
```

parse & validate JWT

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

IAM

iam.example

FTS

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

RUCIO

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

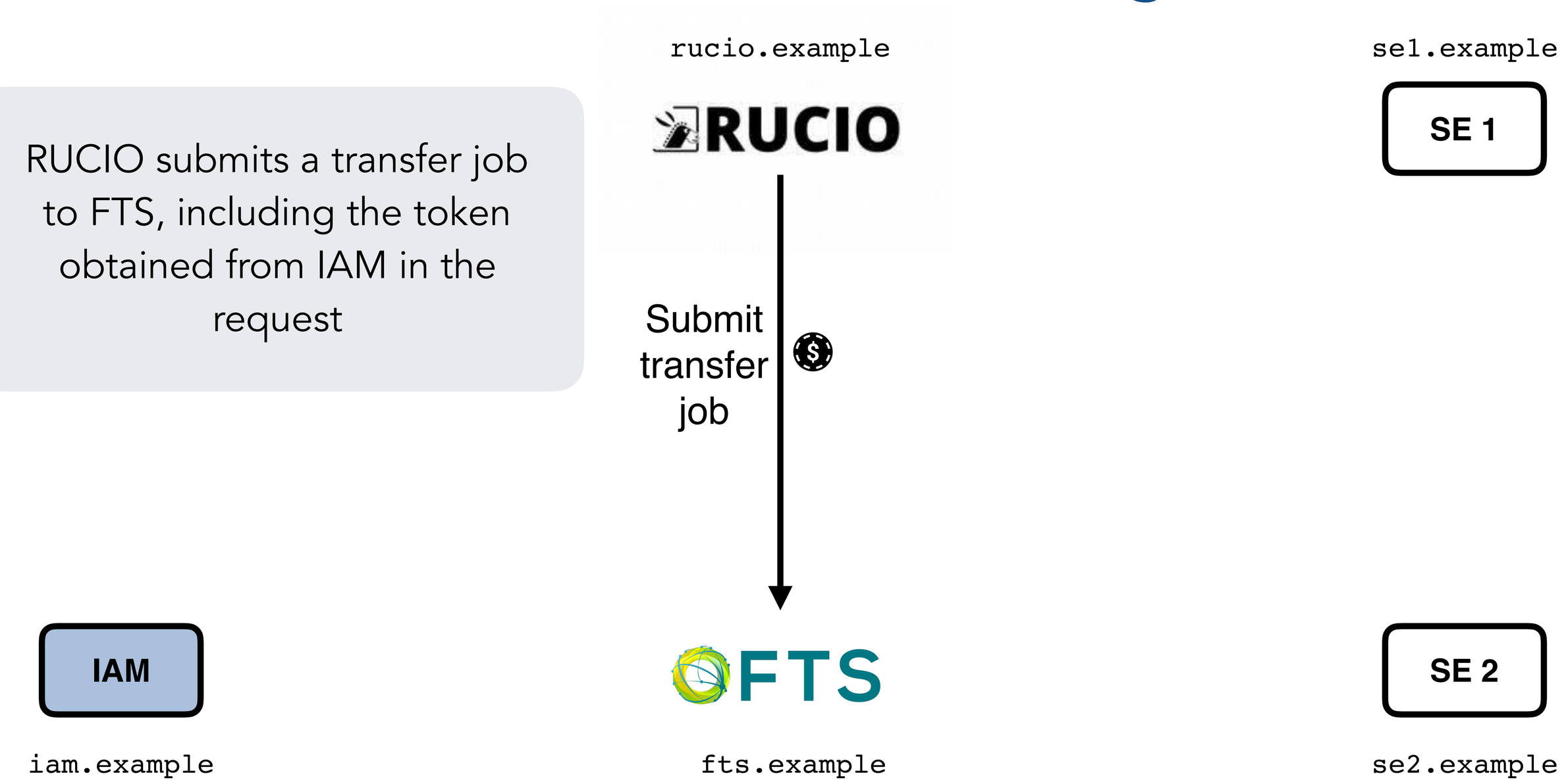
fts.example

SE 2

se2.example

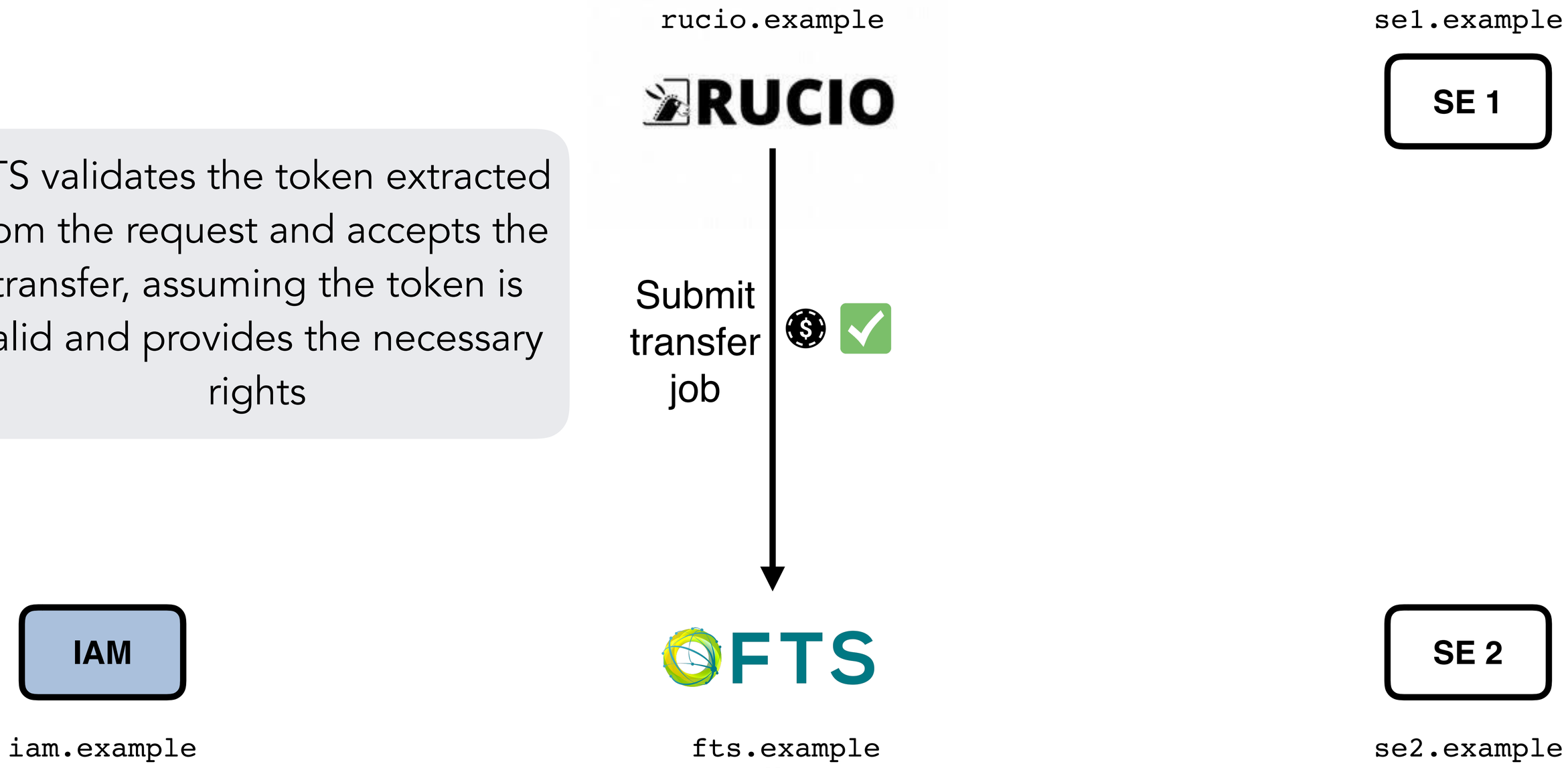
Token-based AuthN/Z for RUCIO managed data xfers

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



Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

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

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

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

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

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

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

iam.example

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

rucio.example

RUCIO

se1.example

SE 1

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

```
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

iam.example

FTS

fts.example

SE 2

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

iam.example

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

FTS extracts the tokens from the response and saves them locally

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

rucio.example

RUCIO

se1.example

SE 1

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, “se2.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.

rucio.example

RUCIO

se1.example

SE 1

IAM

iam.example

FTS

fts.example

SE 2

se2.example

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

rucio.example

RUCIO

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

FTS

fts.example

SE 2

se2.example

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

rucio.example

RUCIO

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

FTS

fts.example

SE 2

se2.example

Token-based AuthN/Z for RUCIO managed data xfers

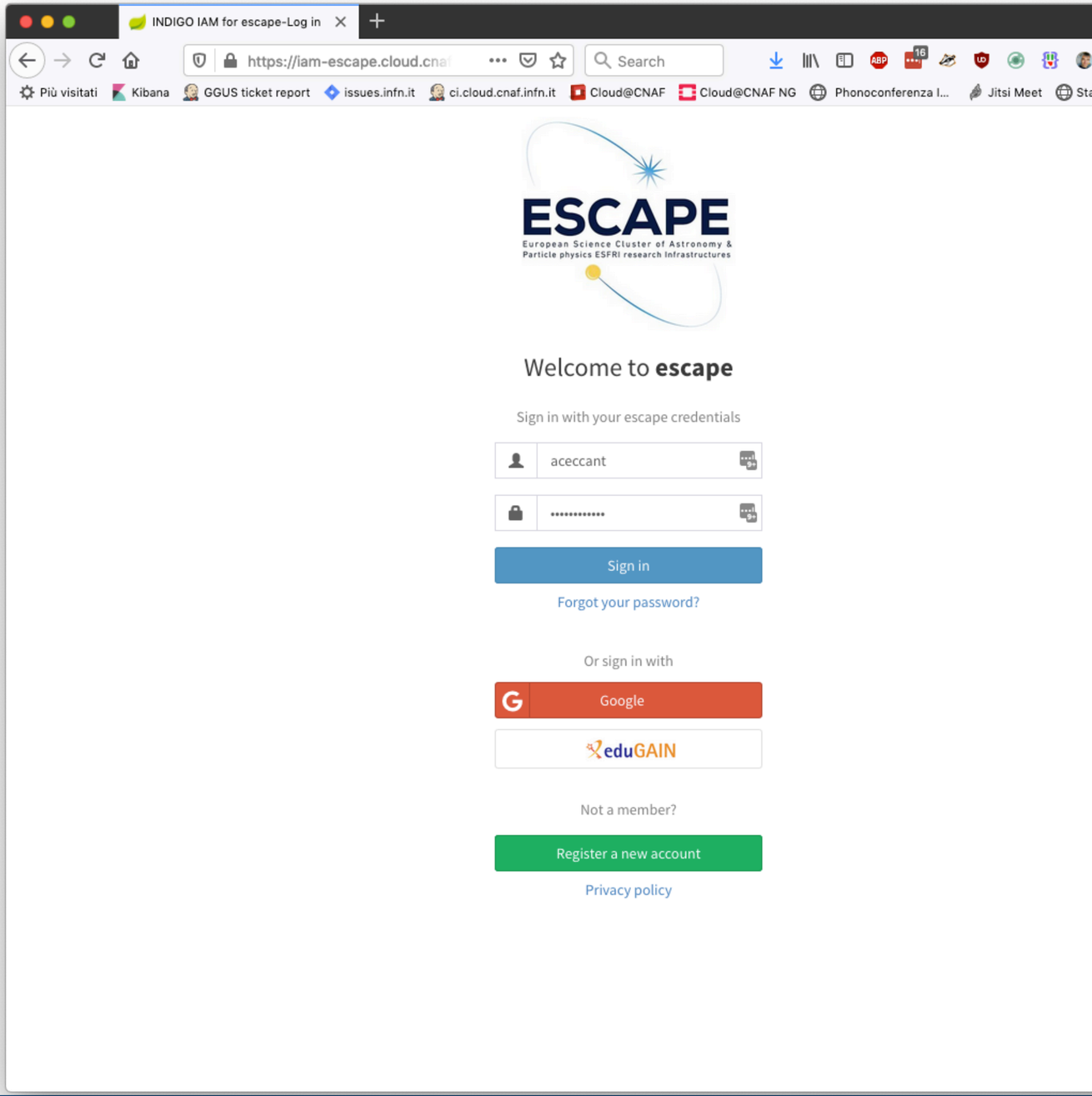
SE2 will then use the obtained token for authn/z against SE1

Status of the ESCAPE testbed

ESCAPE IAM instance

Escape IAM instance available

- 44 registered users
- AuthN with X.509 certificates, Google, username/password, **EuGAIN**
- 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 version

The Escape IAM instance is at the latest version and provides support for it

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"
  ]
}
```

Integration activities

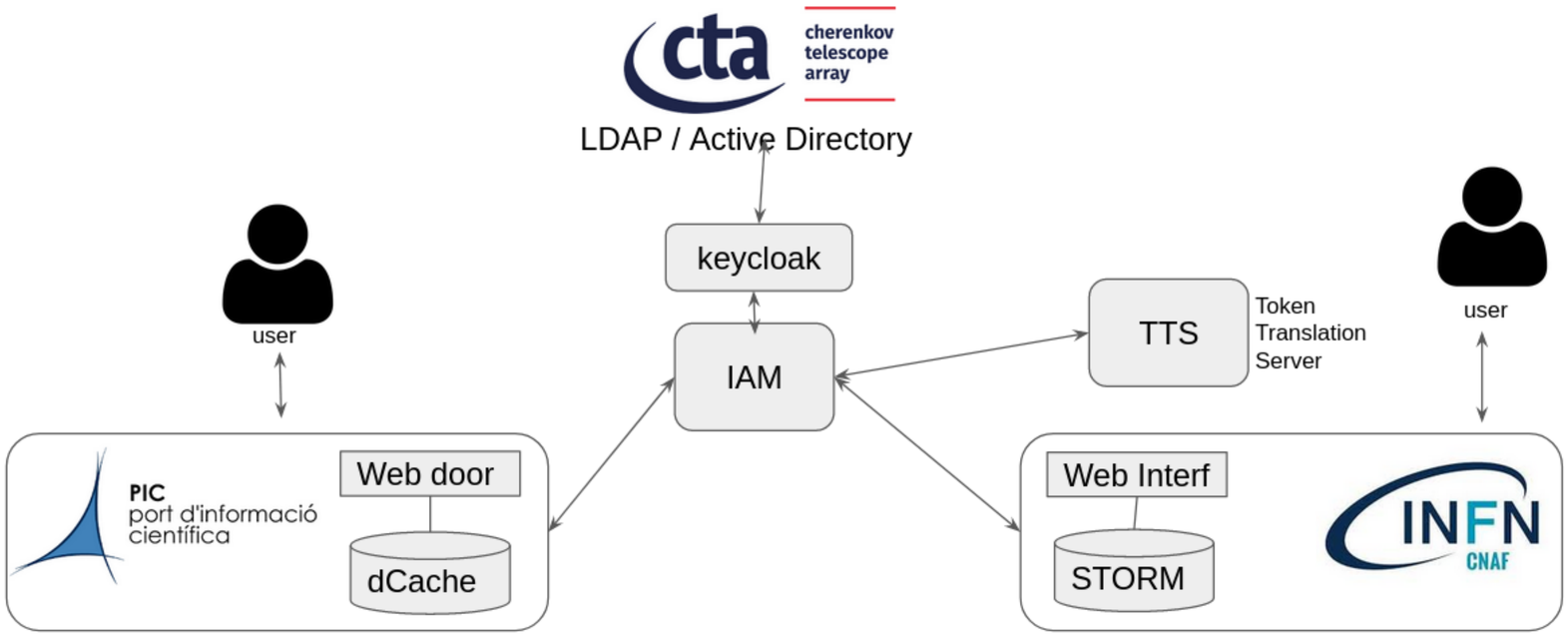
LOFAR EGI CheckIn integration

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 test 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

CNAF-PIC data transfers in support of LST1



Deploy an IAM instance @ PIC, integrated with the CTA LDAP, to provide support for VOMS and token-based AuthN/Z for LST1

ESCAPE progress meeting preparation

AAI at the ESCAPE progress meeting

https://indico.in2p3.fr/event/20203/timetable

90 minutes session planned in the first day afternoon

- Input from WPs (15 minutes per contribution)
- Discussion on joint actions across WPs on AAI (30 minutes)

Strict time constraints impose preparation

- AAI discussions can be **endless**

Proposed plan

- Start discussion today
- have another meeting next week to fine-tune the topics for Brussells

Possible AAI session topics

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?

Identify an **initial set of cross-WP integration use cases**, where higher level services access and manage data in a secured manner

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?