ESCAPE T2.5: Authentication and Authorization

Andrea Ceccanti
ESCAPE WP2 meeting

June 6th 2019

Objectives

Task 2.5 objectives (from the DoW)

"The ESCAPE project **will not build new authentication mechanisms** but **will leverage and build on existing work** to provide the secure composition of data and compute services needed **to enable the data-lake vision**. "

Task 2.5 objectives (from the DoW)

"Through **EGI** and **WLCG** there is a **15-year history of building global AAI**, and with the recent results of the **Indigo-DataCloud** project and the ongoing work in the **AARC** projects to move such AAI structures into the future, **the ESCAPE project will be well placed to integrate such work into the prototypes**."

Task 2.5 objectives (from the DoW)

We will adopt **standards-based** AAI solutions that:

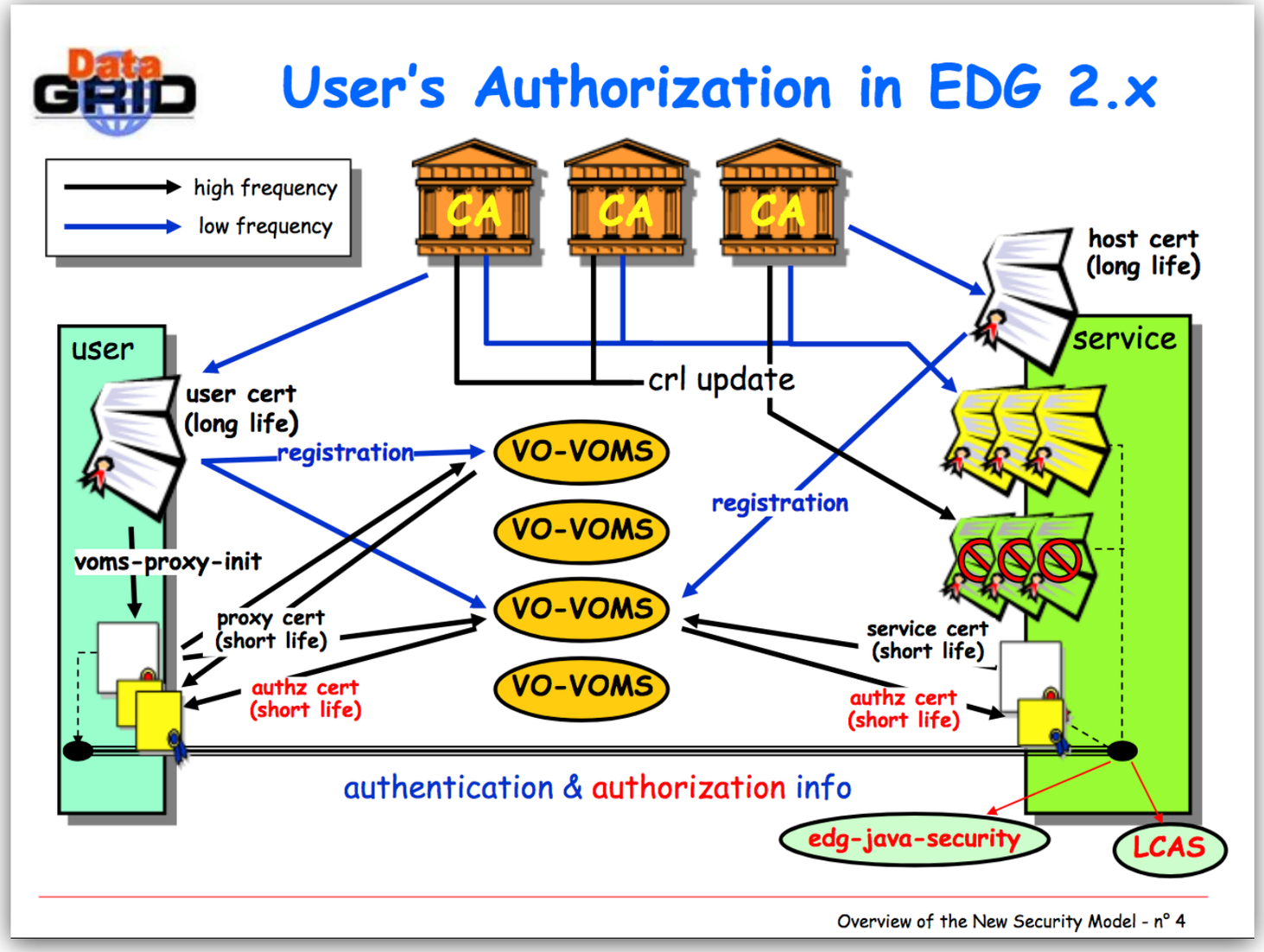
- are flexible enough to support **heterogeneous authentication mechanisms** (federated identities, X.509 certificates, social logins);
- provide the abstraction of **collaboration/virtual organization**, and the tools to manage membership, entitlements and access policies that will regulate access to resources for that organization;
- can support **controlled delegation of privileges** across the distributed chain of services implementing the Data-Lake vision;
- **can be easily integrated** in existing data access and computing software leveraging standard, off-the-shelf libraries and components, in particular to map collaboration-level authentication and authorization attributes and capabilities to local access mechanisms.

The WLCG experience

The current WLCG AAI

In operation since ~2003, and still working nicely:

- **X.509 trust fabric** provided by **IGTF** (tells services which CAs are trusted)
- **X.509 certificates** provided to users for authentication
- **Proxy certificates** for Single Sign-On (SSO) and delegation
- **VOMS attribute certificates** for attribute-based authorization (issued and signed by VO-scoped VOMS servers)

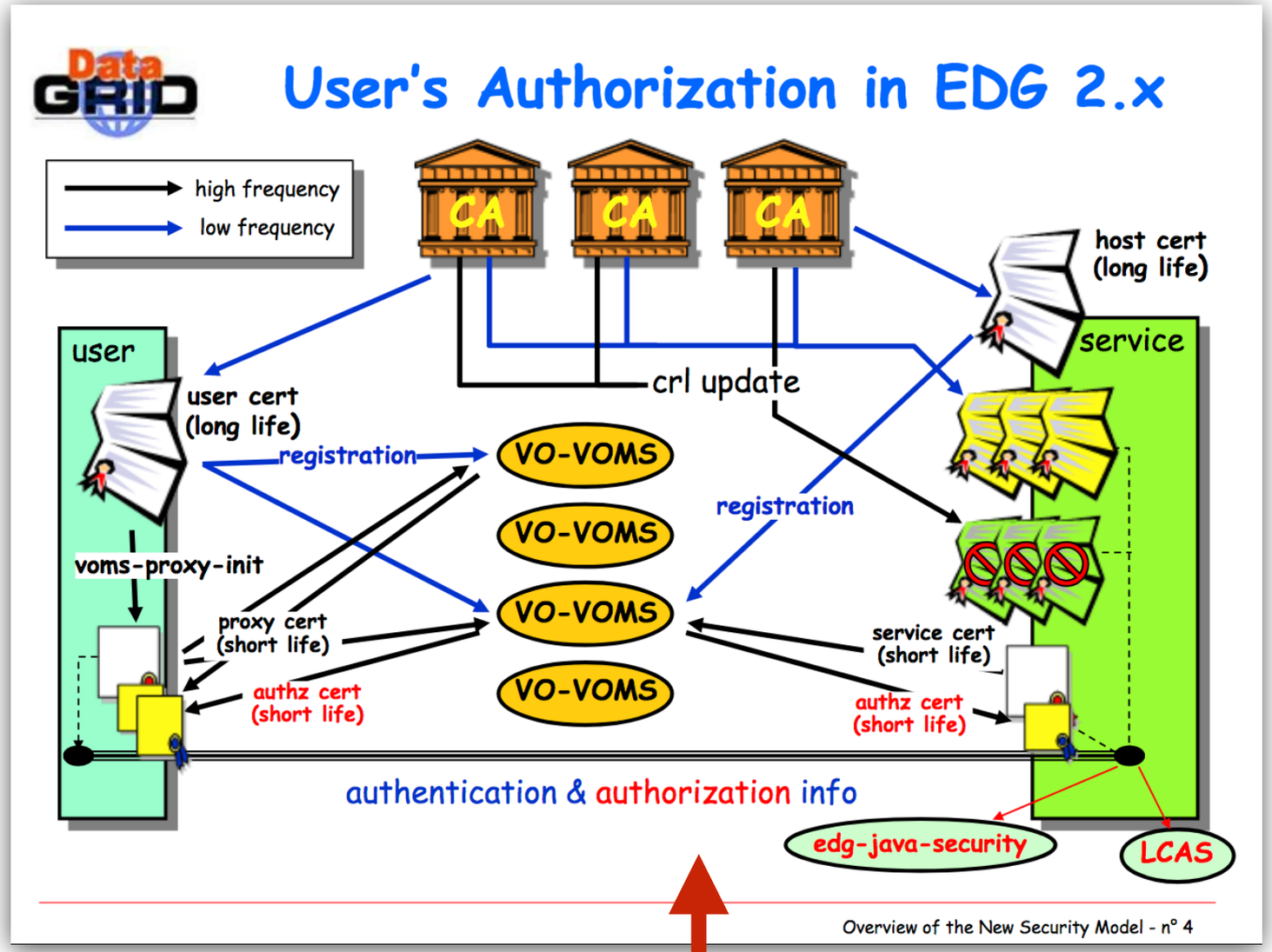


Slide by Ákos Frohner

The current WLCG AAI

In operation since ~2003, and still working nicely:

- **X.509 trust fabric** provided by **IGTF** (tells services which CAs are trusted)
- **X.509 certificates** provided to users for authentication
- **Proxy certificates** for Single Sign-On (SSO) and delegation
- **VOMS attribute certificates** for attribute-based authorization (issued and signed by VO-scoped VOMS servers)



Slide by Ákos Frohner

WARNING:

VO here means Virtual Organization, not Virtual Observatory

Current WLCG AAI: the weak points

Usability

- X.509 certificates are **difficult** to handle for users
- VOMS does not work in browsers

Inflexible authentication

- Only one authentication mechanism supported: X.509 certificates
- Hard to integrate identity federations

Authorization tightly bound to authentication mechanism

- VOMS attributes are inherently linked to an X.509 certificate subject

Ad-hoc solution

- We had to invent our own standard and develop ad-hoc libraries and central services to implement our own AAI

Can we do better today?

A novel AAI for WLCG: main challenges

Authentication

- **Flexible**, able to accomodate various authentication mechanisms
  - X.509, username & password, EduGAIN, social logins (Google, GItHub), ORCID, …

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

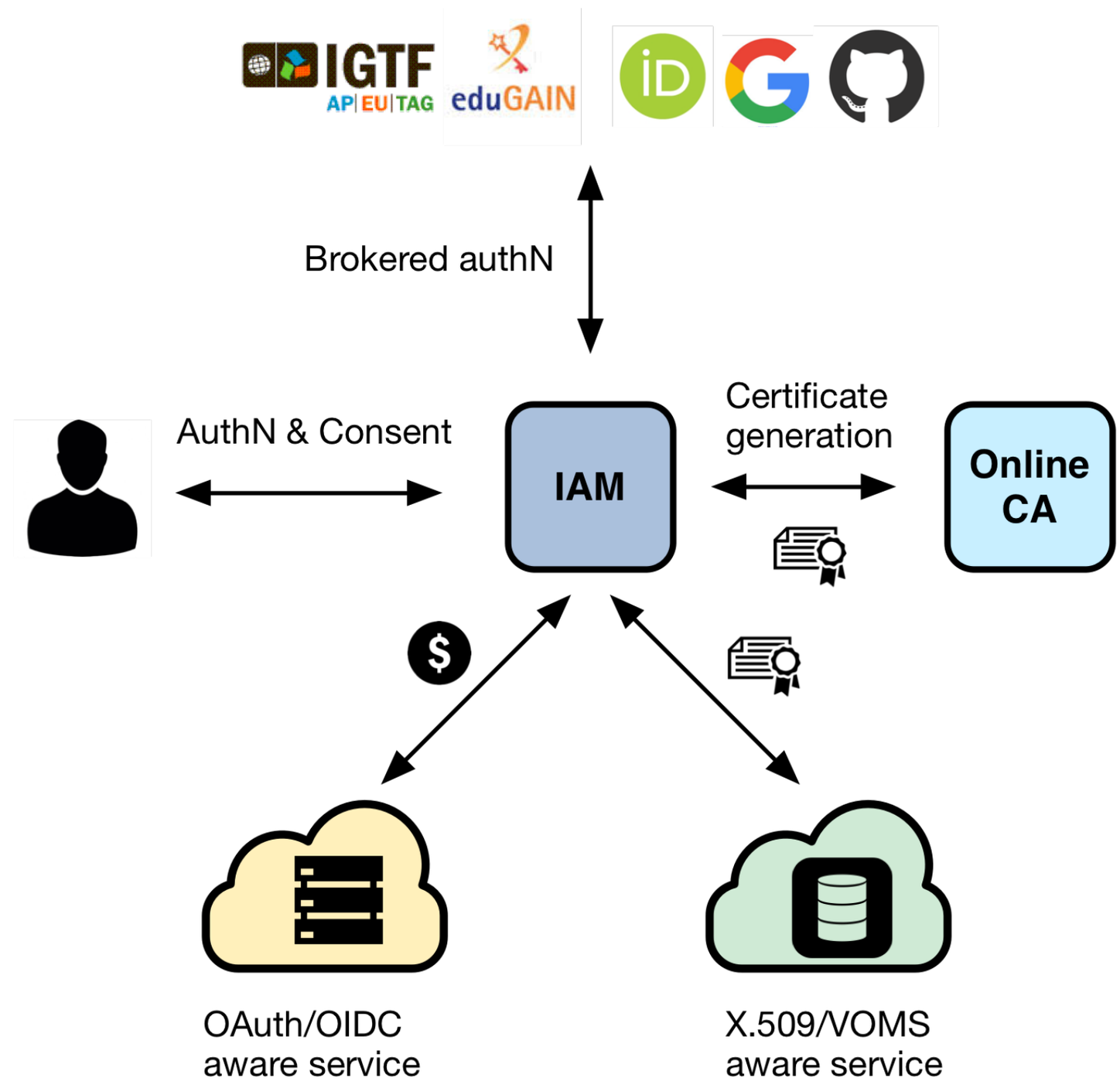
Token translation

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

The future token-based WLCG AAI

Introduce a central VO-scoped authz 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**

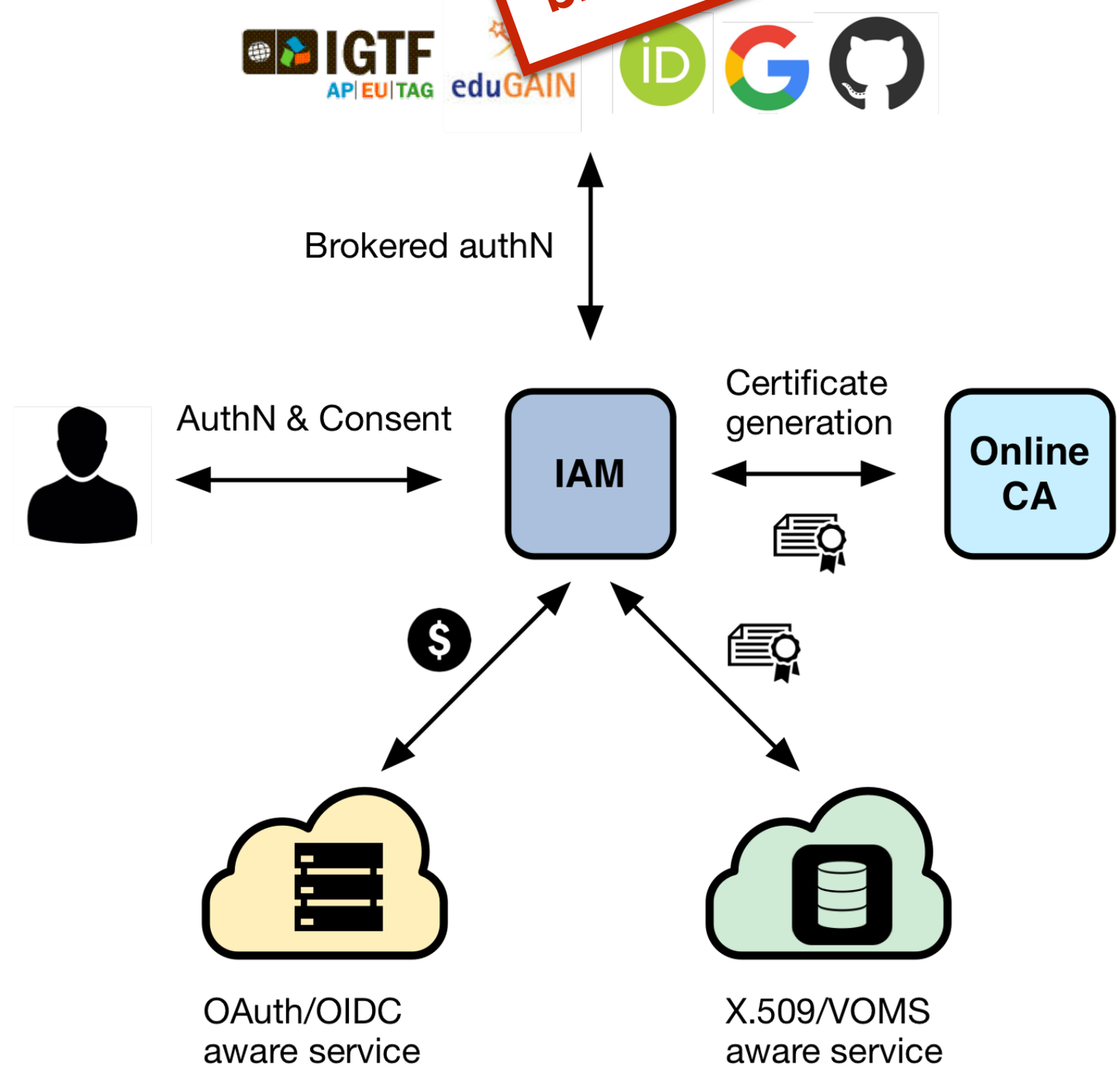


The future token-based WLCG AAI

Compliant with AARC blueprint architecture

Introduce a central VO-scoped authz 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**



Enabling technologies: an overview

Enabling technologies in one slide

OAuth 2.0

- a standard framework for **delegated authorization**
- widely adopted in industry

OpenID Connect

- an **identity layer** built on top of OAuth 2
- "OAuth-based authentication done right"

JSON Web Tokens (JWTs)

- a **compact**, **URL-safe** means of representing **claims** to be transferred between two (or more) parties


```
{
  "sub": "e1eb758b-b73c-4761-bfff-adc793da409c",
  "aud": "iam-client test",
  "iss": "https://iam-test.indigo-datacloud.eu/",
  "exp": 1507726410,
  "iat": 1507722810,
  "jti": "39636fc0-c392-49f9-9781-07c5eda522e3"
}
```

OAuth: a delegated authorization framework

OAuth defines how **controlled delegation of privileges** can happen among collaborating services

Provides answers to questions like:

- How can an application request access to protected resources?
  - How can I obtain **an access token**?
- How is authorization information exchanged across parties?
  - How is the **access token** presented to **protected resources**? (i.e. APIs)

OpenID Connect: an identity layer for OAuth

OAuth is a **delegated authorization** protocol

- an **access token** states the **authorization rights** of the client application presenting the token to access some resources

OpenID Connect extends OAuth to provide a standard **identity layer**

- i.e. information about **who the user is** and **how it was authenticated** via an additional **ID token (JWT)** and a dedicated **user information query endpoint** at the OpenID Connect Identity provider
- provides ability to establish **login sessions** (SSO)

+

JSON Web Tokens (JWT)

JSON Web Token (JWT) is an open standard that defines a compact, self-contained way of securely transmitting information between parties as a JSON object

JWTs are typically **signed** and, if confidentiality is a requirement, can be **encrypted**.

Header

```
{
  "kid": "rsa1",
  "alg": "RS256"
}
```

.

Body

```
{
  "sub": "e1eb758b-b73c-4761-bfff-adc793da409c",
  "iss": "https://iam-test.indigo-datacloud.eu/",
  "exp": 1482163788,
  "iat": 1482160188,
  "jti": "e7bcb54c-8f67-4a77-8415-37adeb4b958c"
}
```

.

Signature

```
QbOfPrha9kp4e7TknXe88
d8v_9e7V2v2xMAKX10xY4
M3P1wragAhQmyoVQwq-uk
```

Why OAuth, OpenID Connect and JWT?

Standard, widely adopted in industry

- Do not reinvent the wheel, reuse existing knowledge and tools, extend when needed

Reduced integration complexity at relying services

- Off-the-shelf libraries and components

Authentication-mechanism agnostic

- The AAI is not bound to a specific authentication mechanism

Distributed verification of access and identity tokens

- It scales

Back to our token-based AAI...

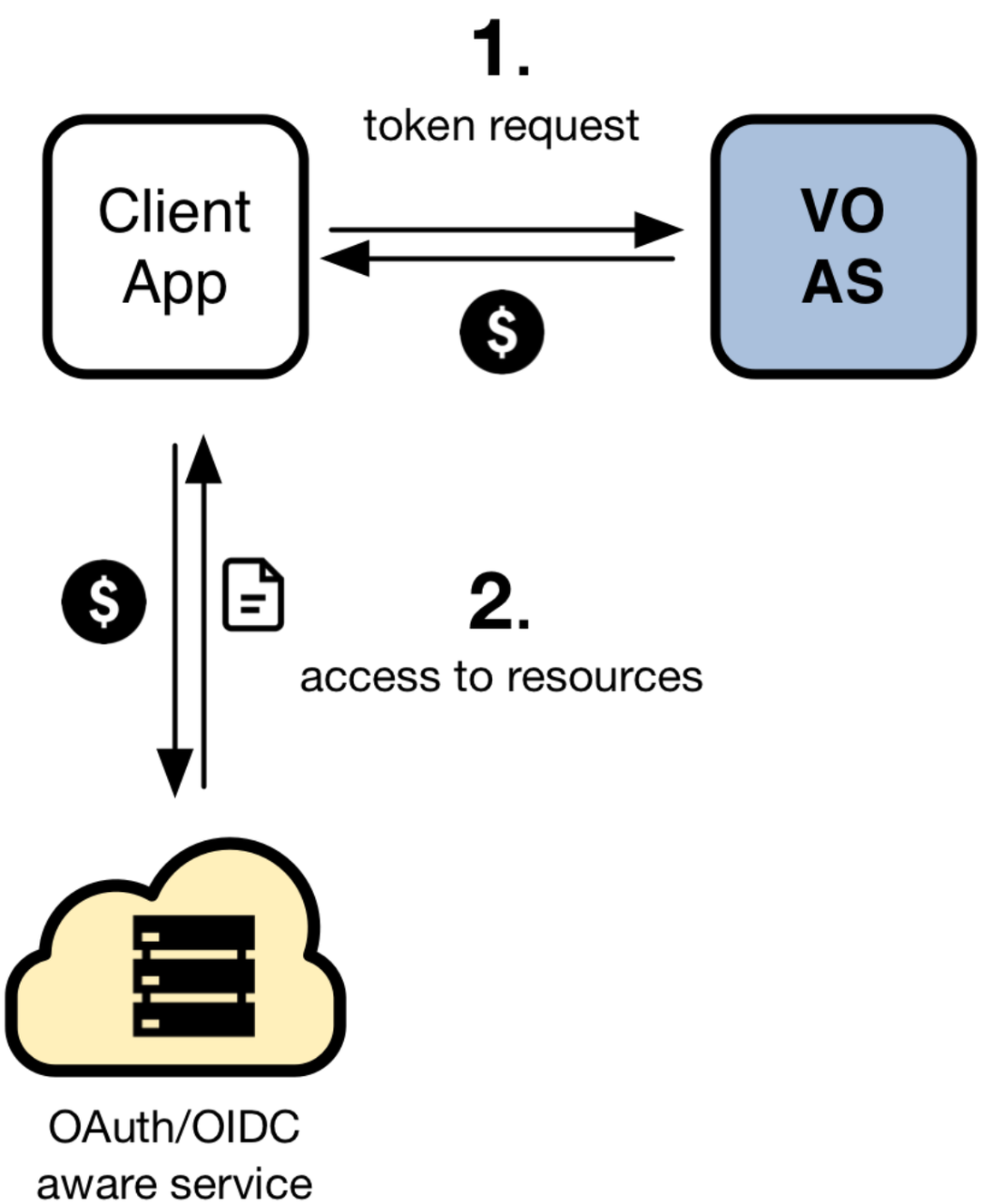
OAuth and OpenID Connect for WLCG

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

The token is obtained from **a VO** (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**, roles, …
- **OAuth scopes**: capabilities linked to access tokens at token creation time

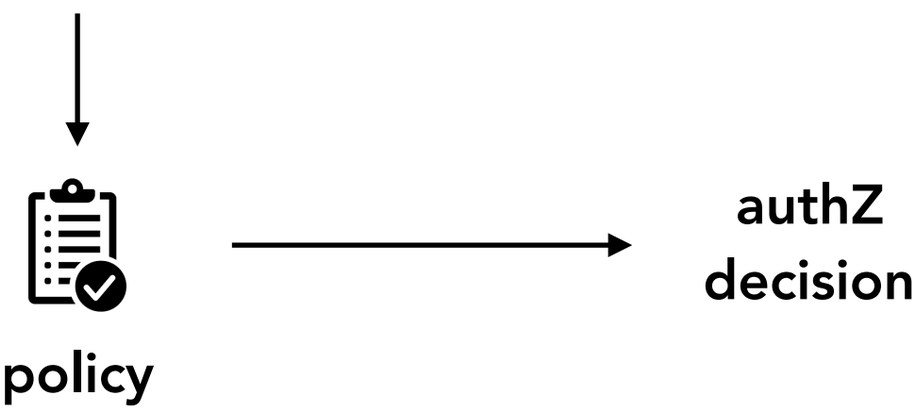


Identity-based vs Scope-based Authorization

Identity-based authorization: the token brings information about attribute ownership (e.g., groups/role membership), the service maps these attributes to a local authorization policy

token claims

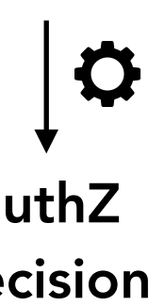
```
{
  "iss": "https://cms.wlcg.example",
  …
  "groups": "/cms"
}
```



Scope-based authorization: the token brings information about which actions should be authorized at a service, the service needs to understand these capabilities and honor them. The authorization policy is managed at the VO level

token claims

```
{
  "iss": "https://cms.wlcg.example",
  …
  "scope": "read:/ upload:/store"
}
```

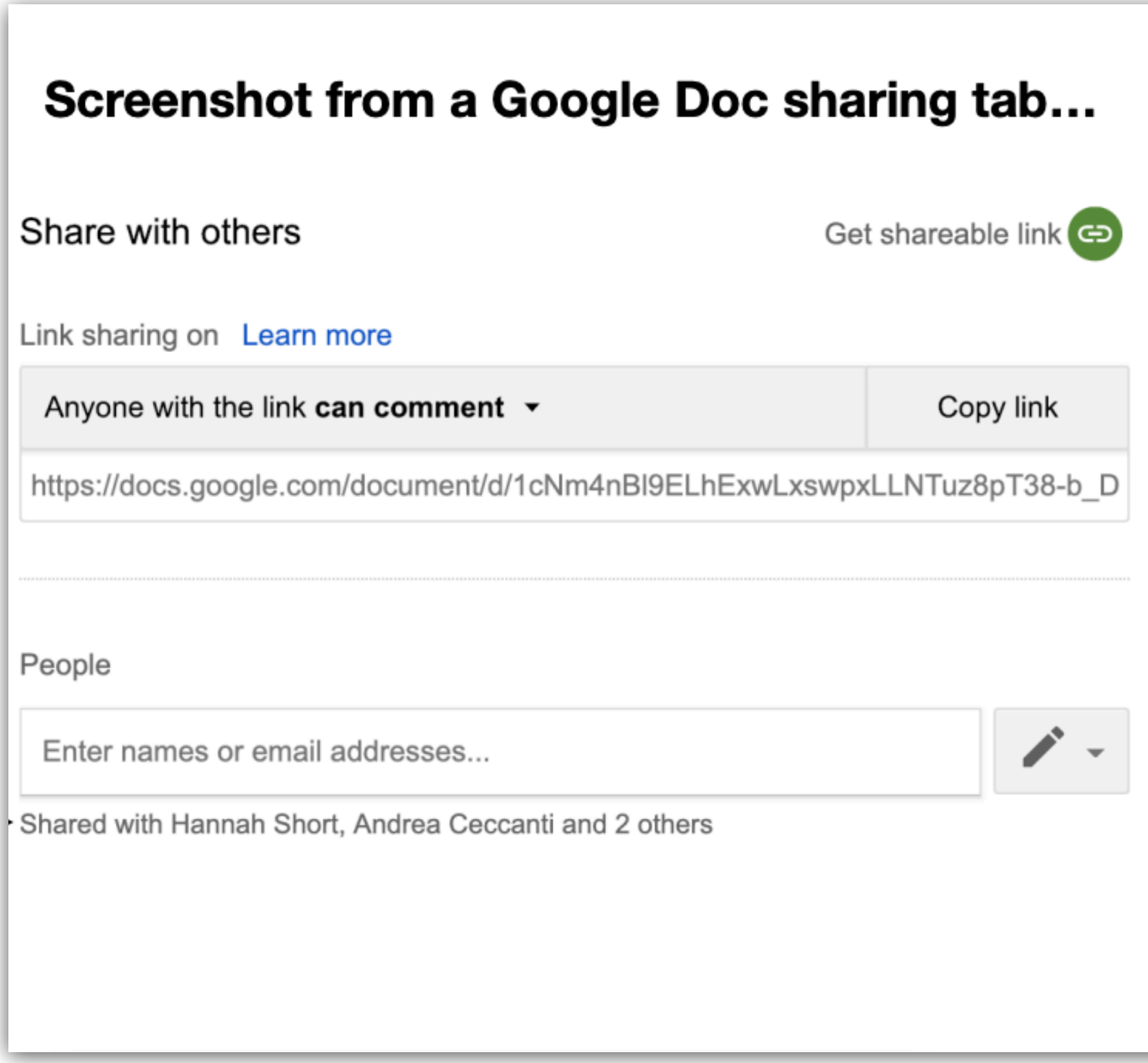


Identity-based vs Scope-based Authorization

The two models can coexist, even in the context of the same application!

scope-based authZ

identity-based authZ



* Slide courtesy of B. Bockelman

Token-based AuthN/AuthZ in practice

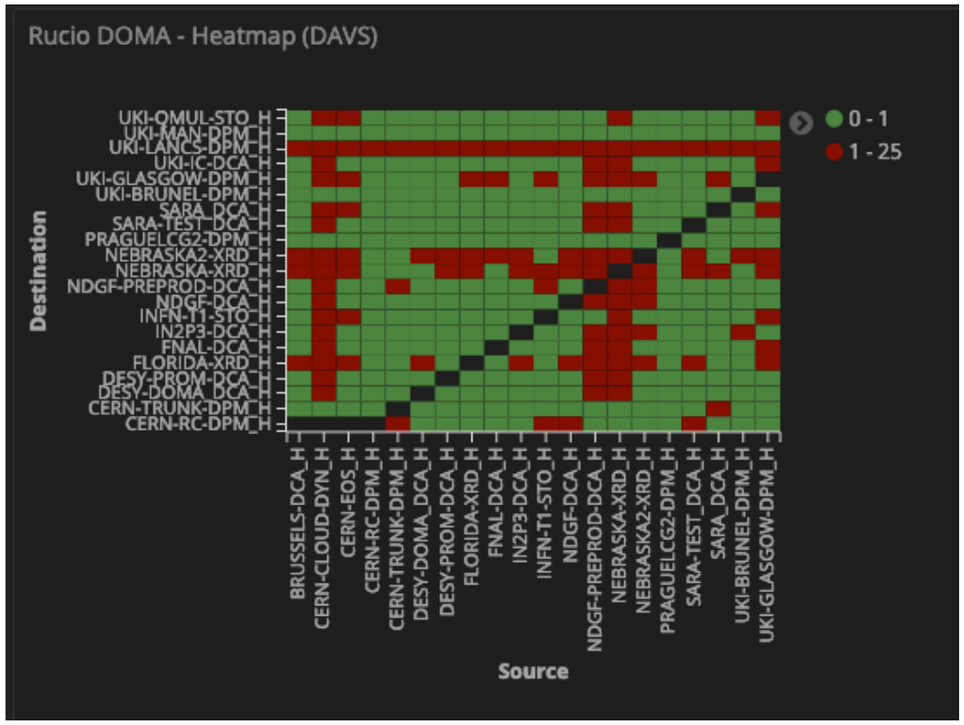
Token-based AuthN/Z in DOMA TPC WG

Bearer tokens used for authorization and delegation in HTTP third-party transfers across WLCG storage elements

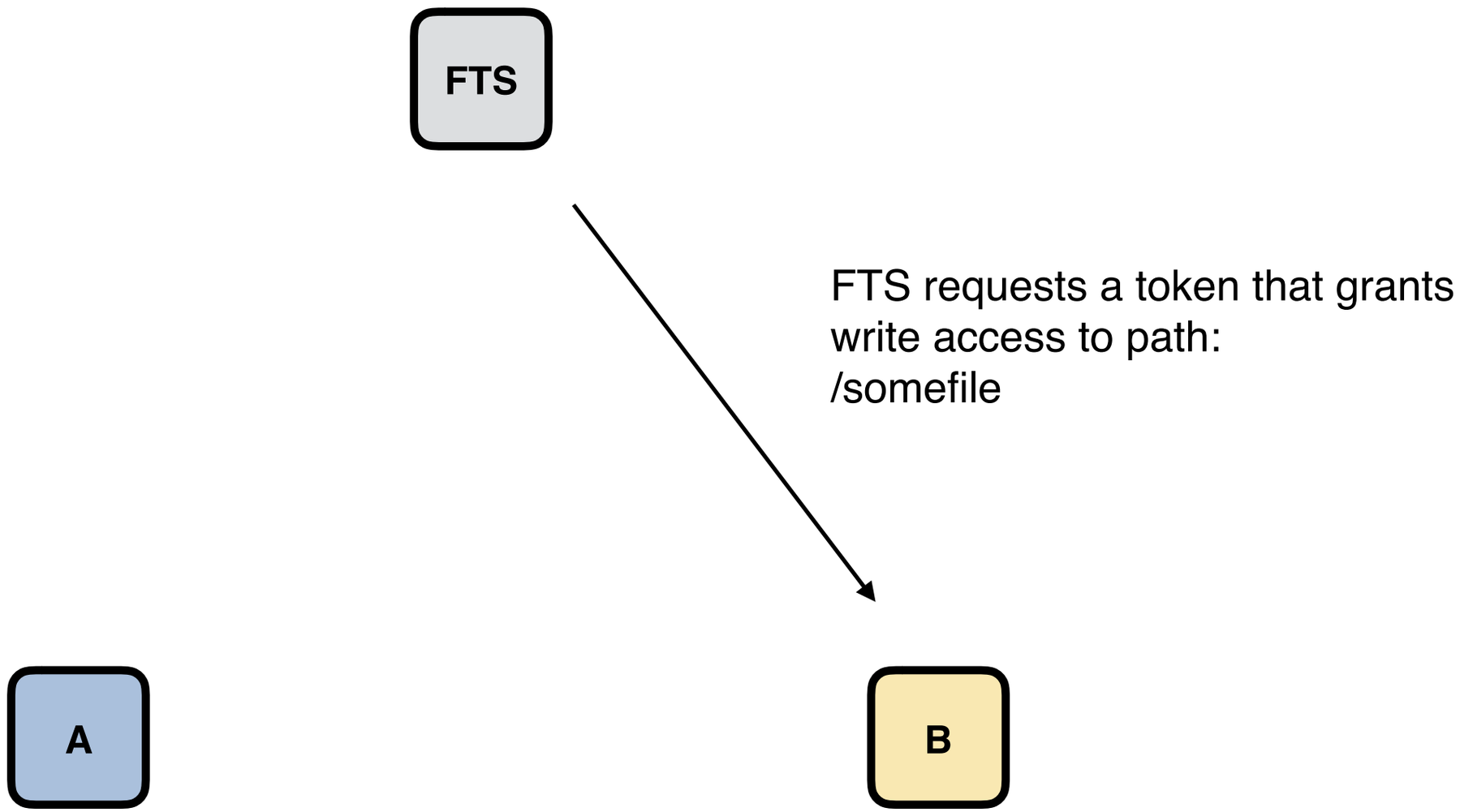
AuthZ already working fine for most SEs: dCache, DPM, StoRM, XRootD

FTS exchanges a VOMS proxy with an **SE-issued** authorization token, which grants (a subset of) the privileges granted by the VOMS proxy on such storage element

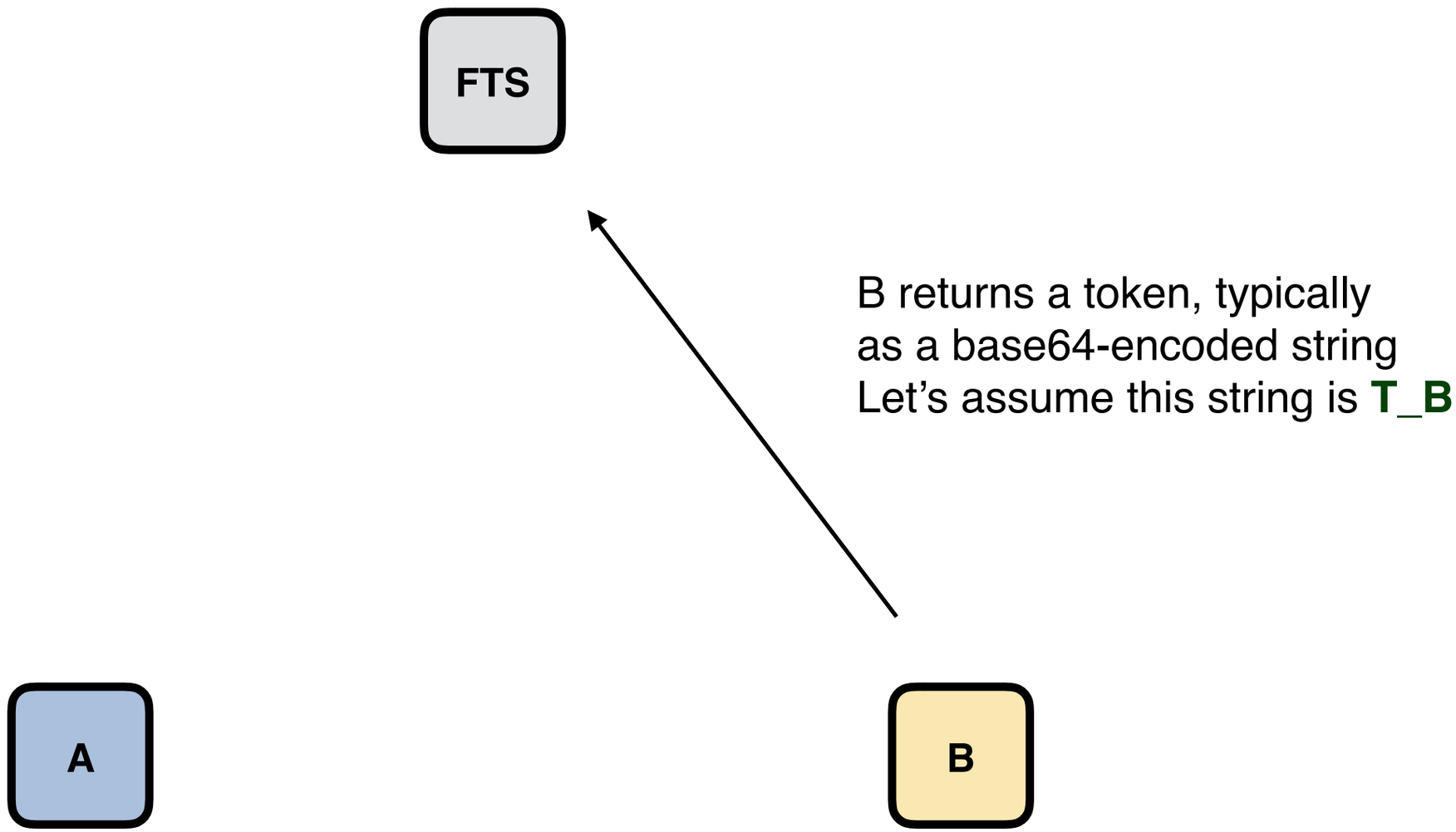
Agreement to converge on a common OAuth-based flow to request tokens



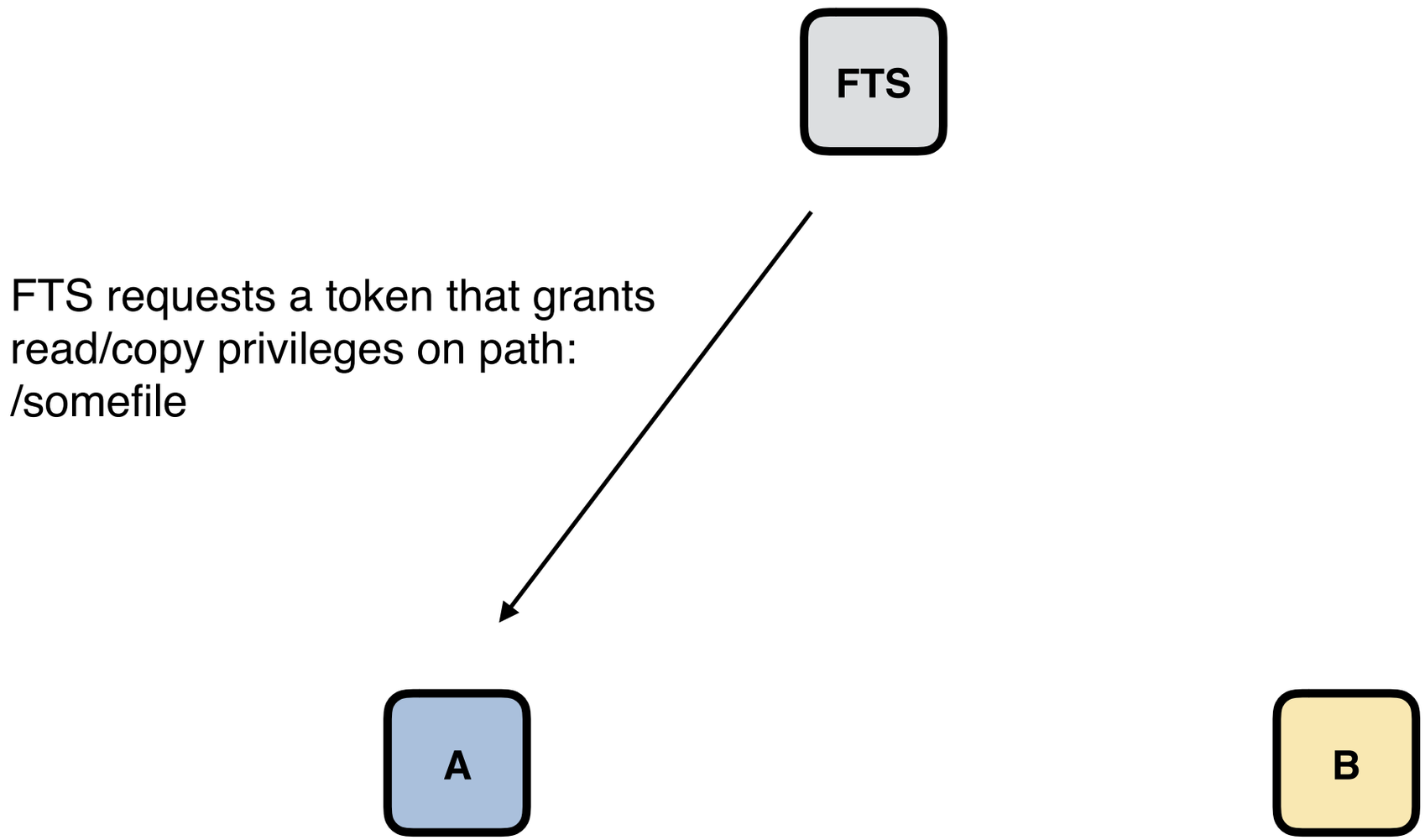
Token-based delegated AuthZ example



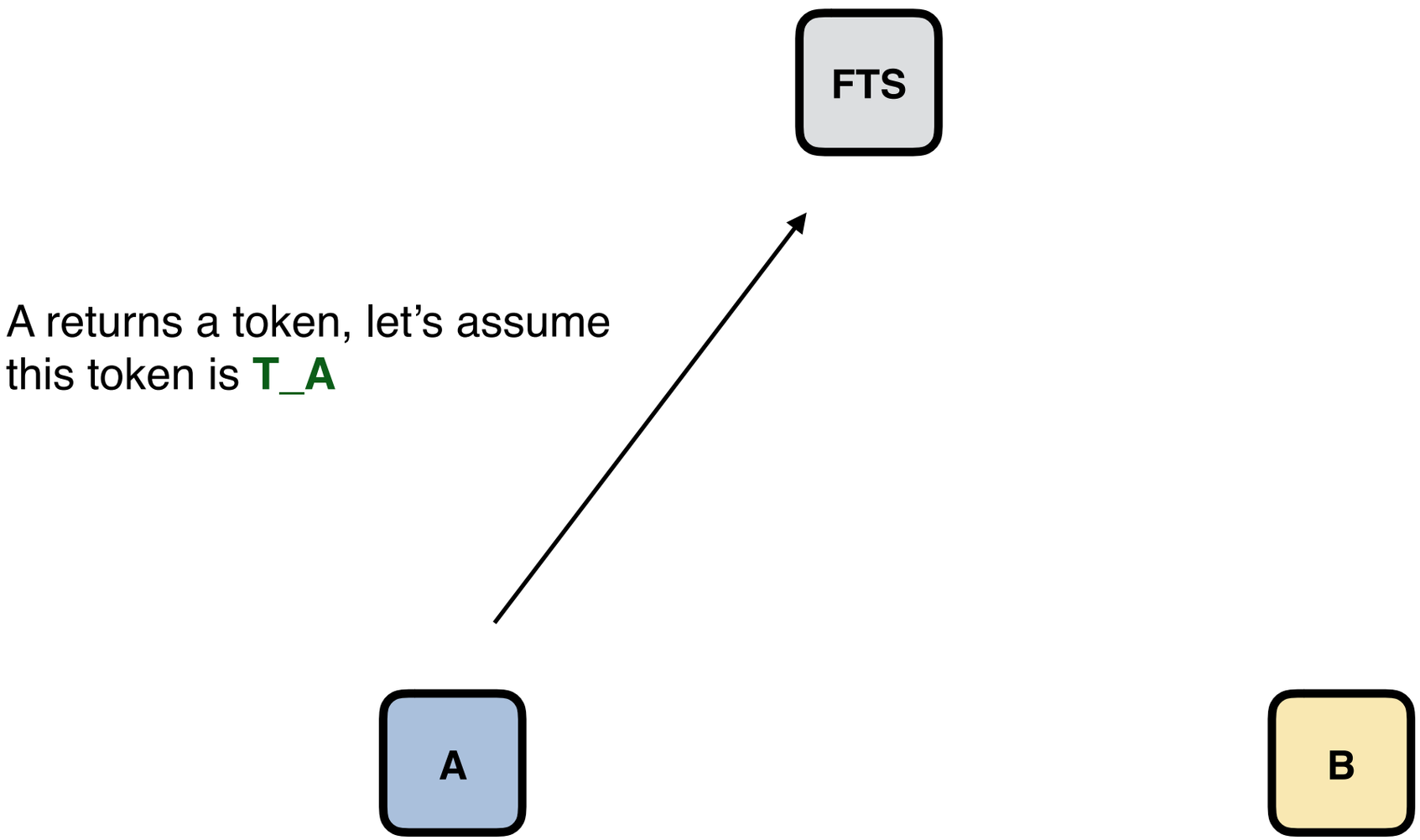
Token-based delegated AuthZ example



Token-based delegated AuthZ example

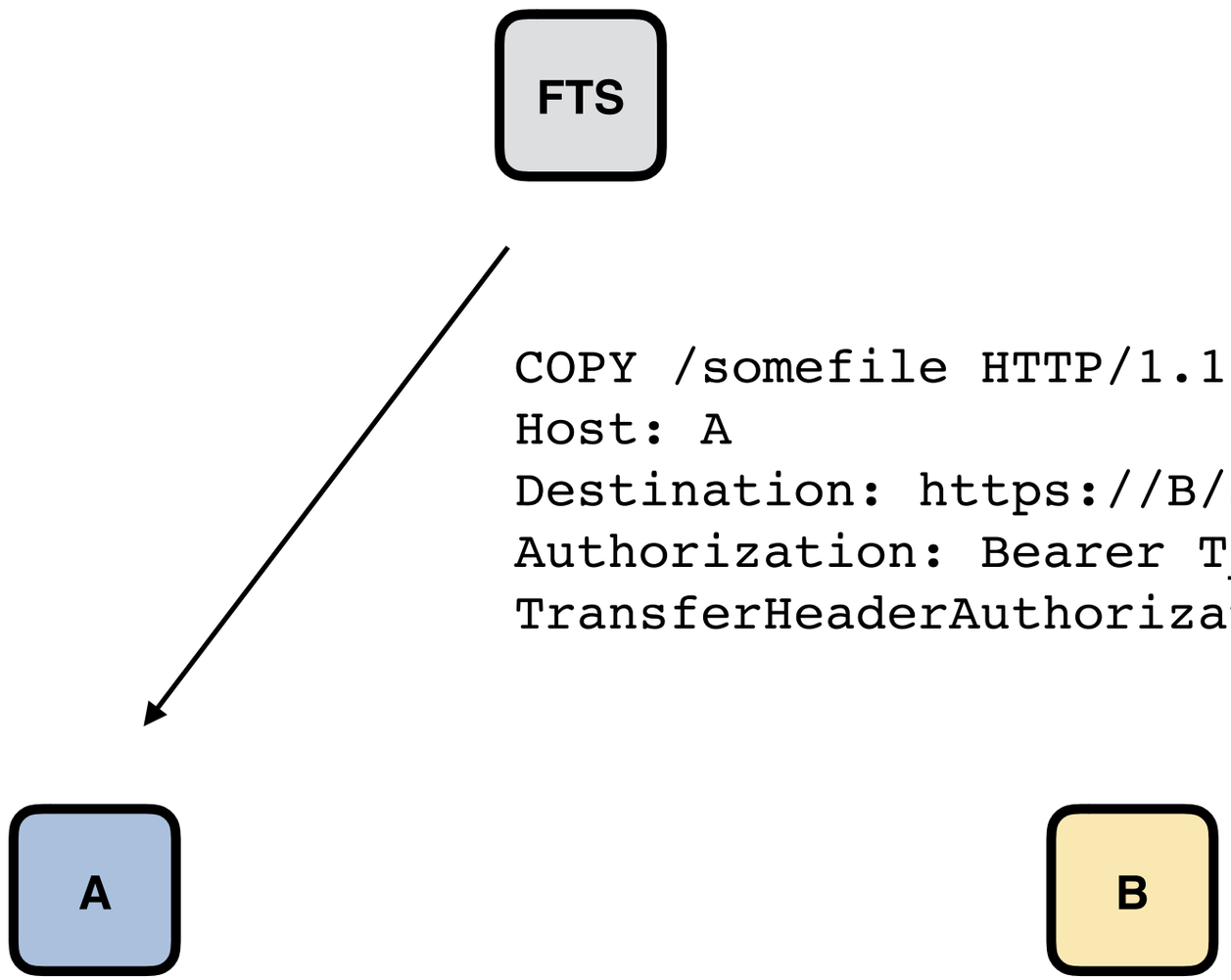


Token-based delegated AuthZ example



Token-based delegated AuthZ example

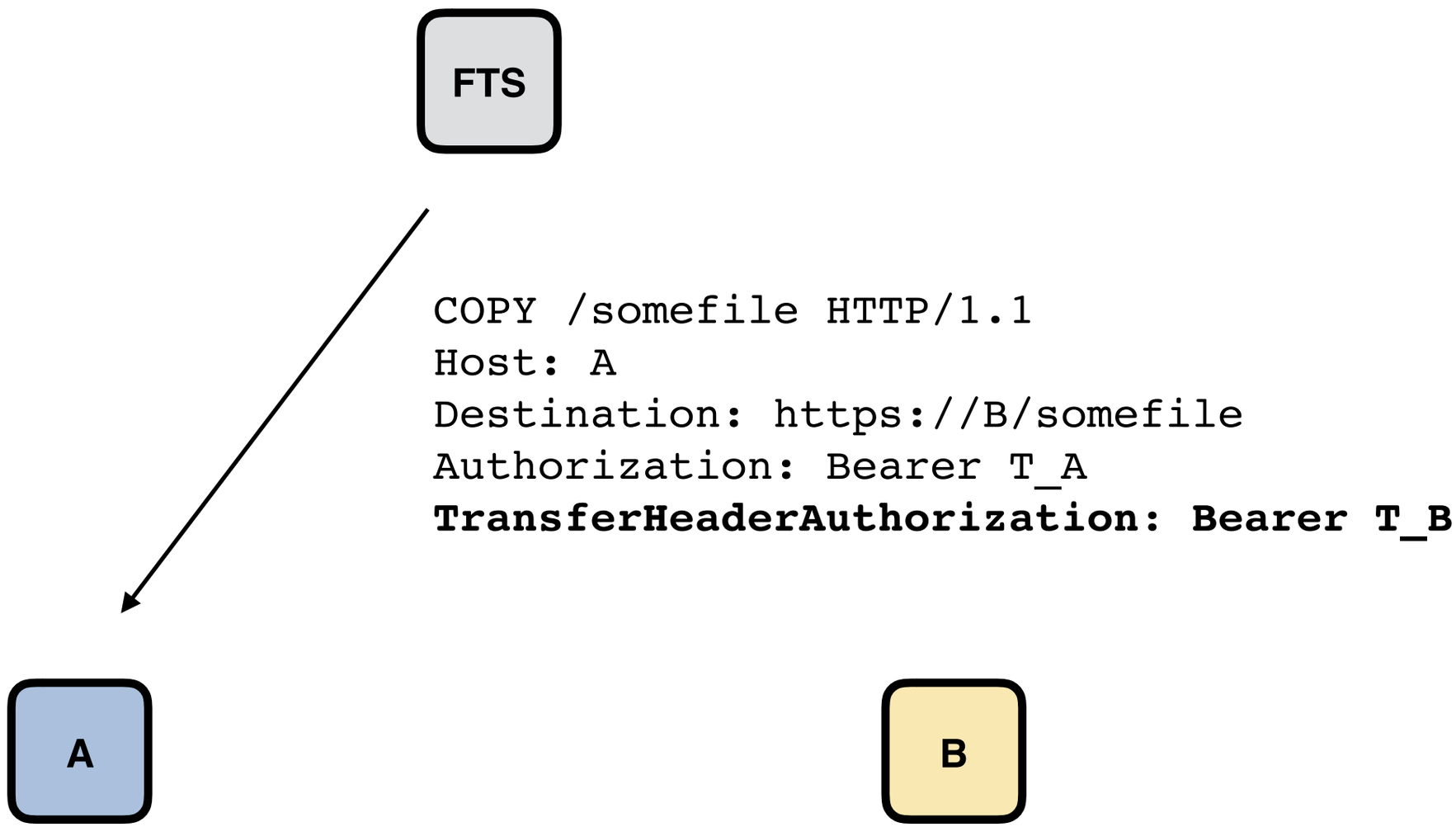
FTS can now request a TPC
from A/somefile to B/somefile



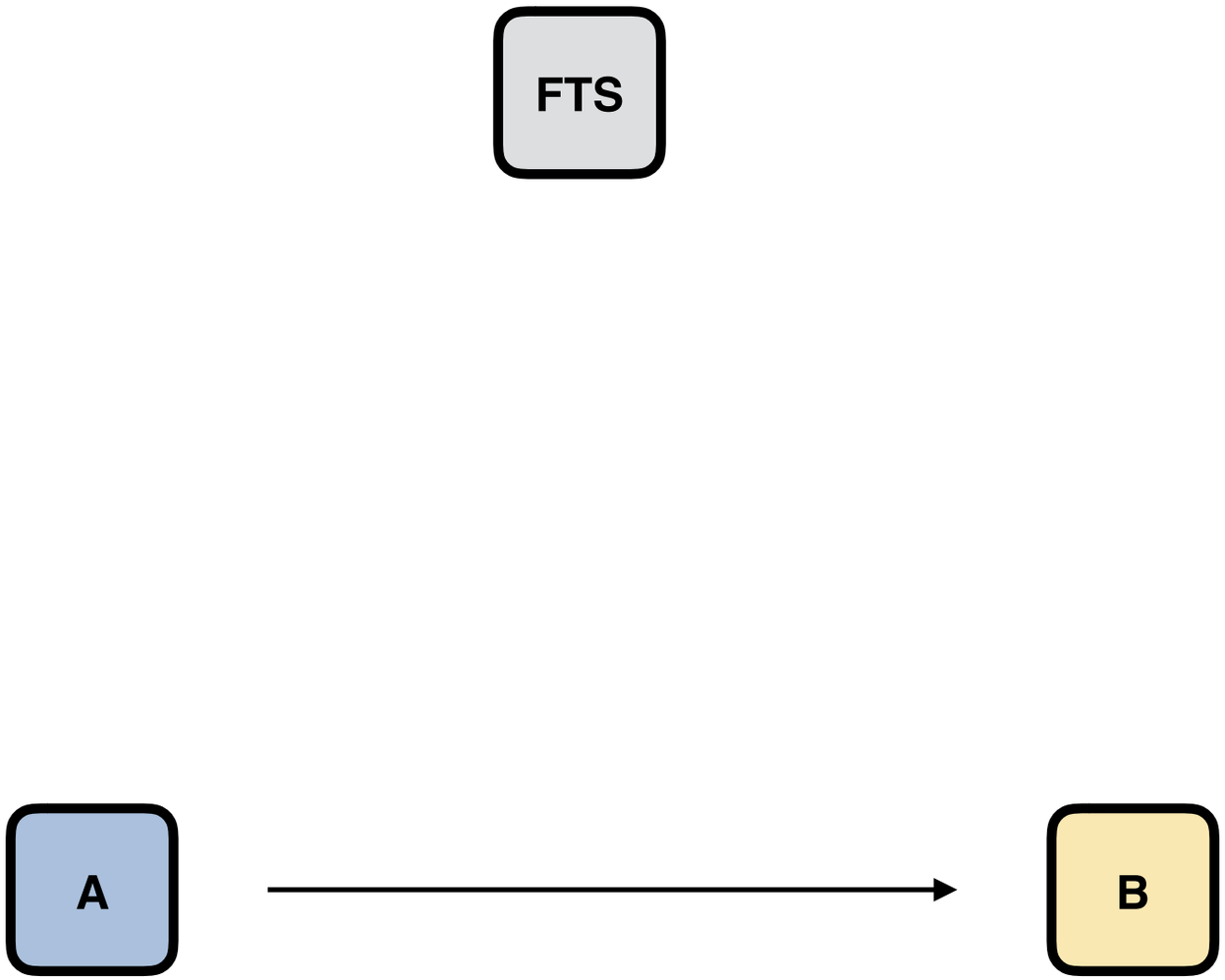

```
COPY /somefile HTTP/1.1
Host: A
Destination: https://B/somefile
Authorization: Bearer T_A
TransferHeaderAuthorization: Bearer T_B
```

Token-based delegated AuthZ example

The protocol provides a way to request that certain headers in the COPY request are included in related transfer requests: all headers in the copy request starting with **TransferHeader** will be copied in the transfer request without such prefix.



Token-based delegated AuthZ example



The INDIGO IAM service

INDIGO Identity and Access Management service

Flexible authentication support

- (SAML, X.509, OpenID Connect, username/password, …)

Account linking

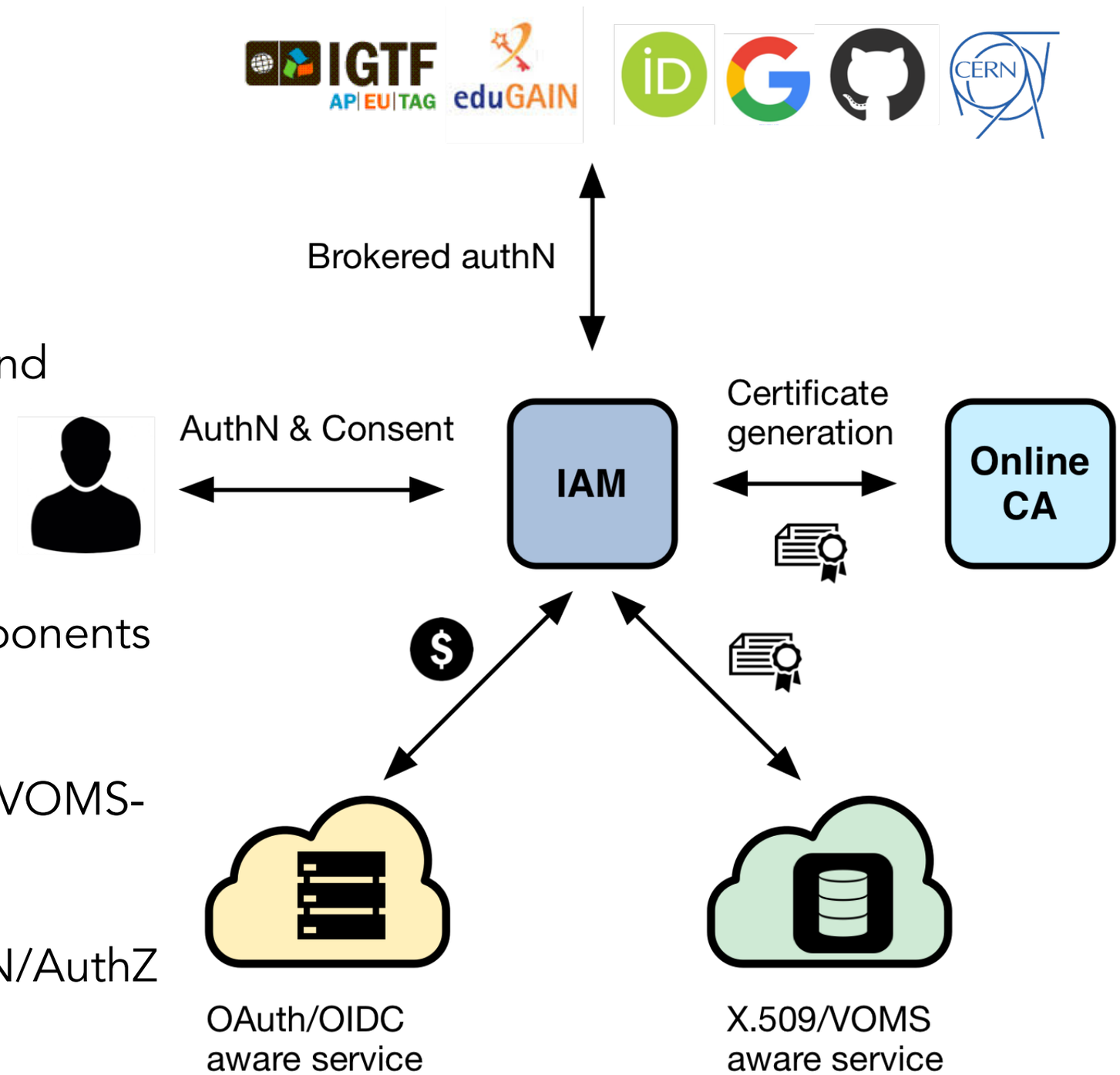
Registration service for moderated and automatic user enrollment

Enforcement of AUP acceptance

Easy integration in off-the-shelf components thanks to **OpenID Connect/OAuth**

VOMS support, to integrate existing VOMS-aware services

Self-contained, comprehensive AuthN/AuthZ solution



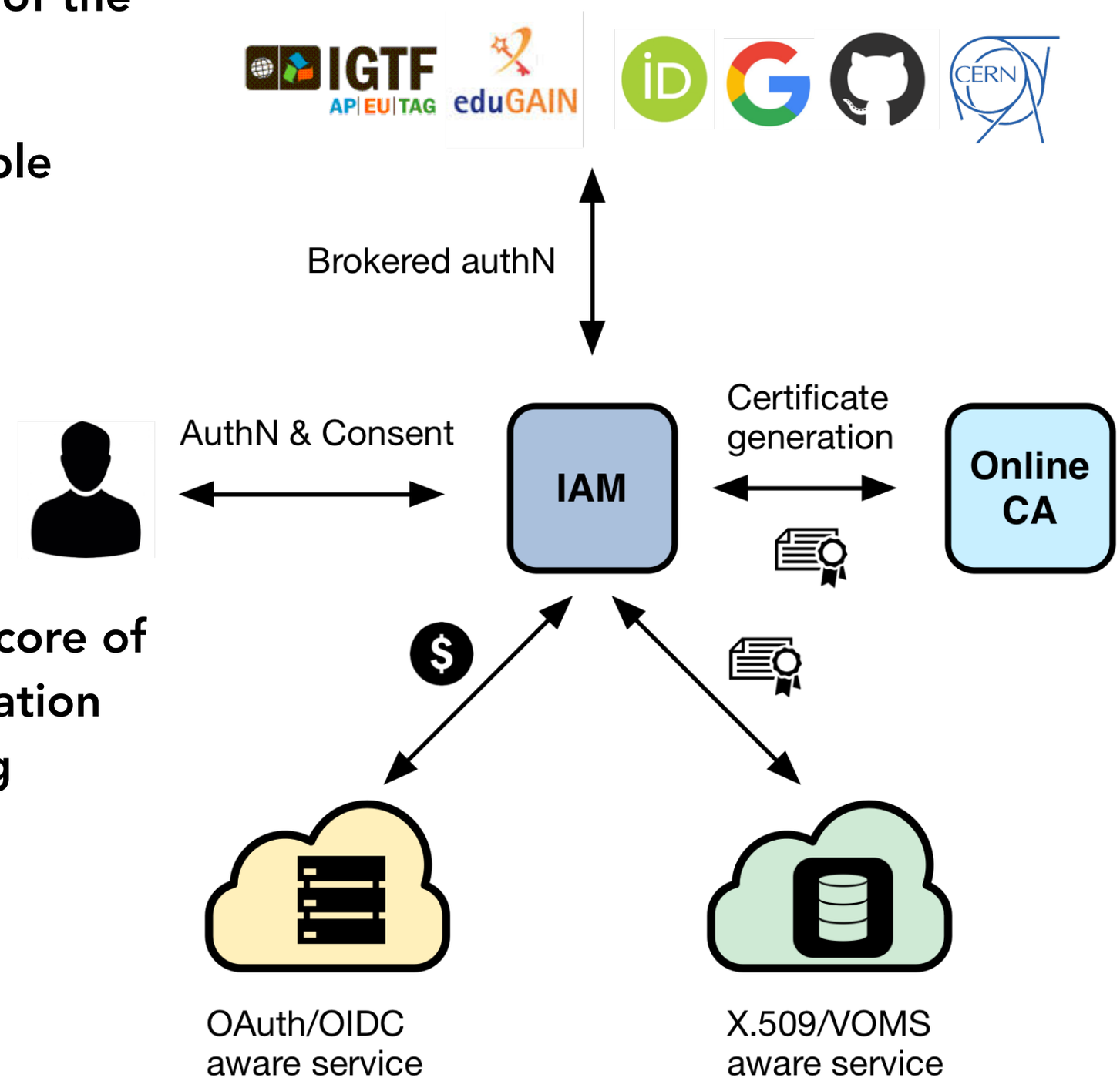
INDIGO Identity and Access Management service

Originally developed in the context of the INDIGO DataCloud project

Sustained by INFN for the foreseeable future with support from:

- EOSC-Hub
- ESCAPE

Selected by WLCG to be the at the core of the next-generation WLCG authorization service in support of LHC computing

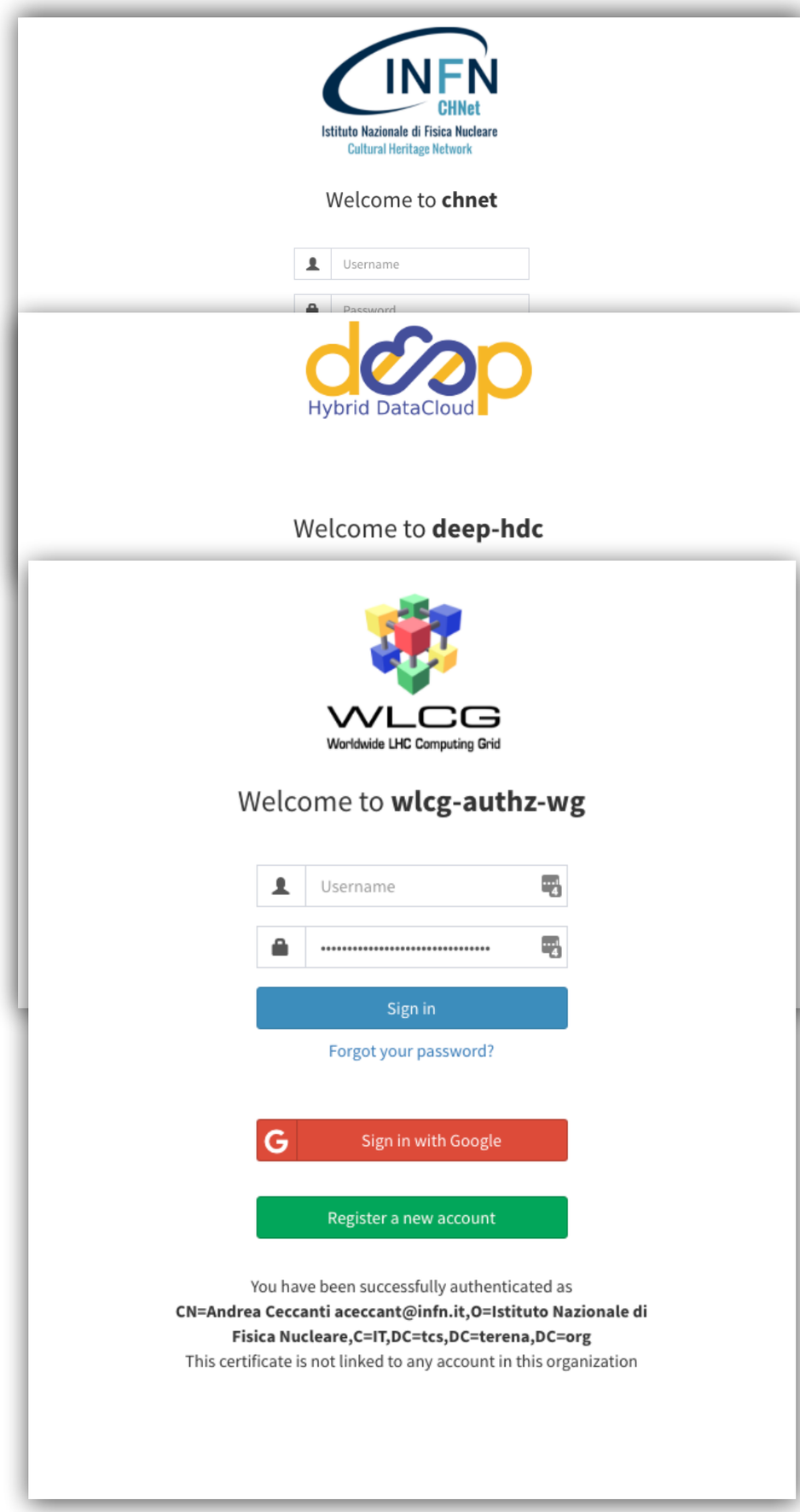


IAM deployment model

An IAM instance is deployed for a **community** of users sharing resources, the good old **Virtual Organization (VO)** concept

Client applications and services are integrated with this instance via **standard OAuth/OpenID Connect**

The IAM Web appearance can be **customized** to include a **community logo**, **AUP** and **privacy policy** document



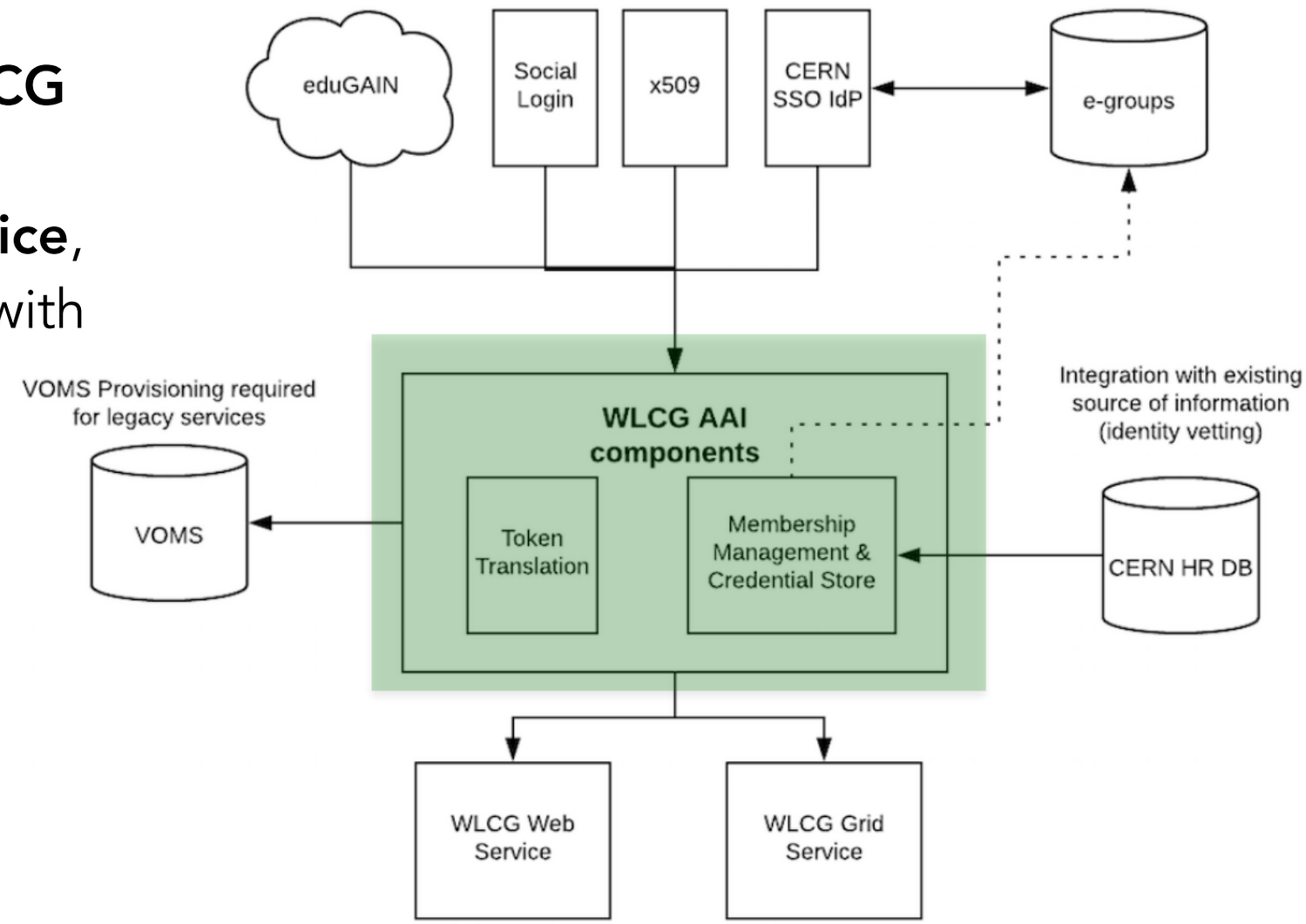
Standardization/Harmonization activities

The WLCG Authorization WG

https://twiki.cern.ch/twiki/bin/view/LCG/WLCGAuthorizationWG

Main objectives:

- Design and testing of a **WLCG Membership Management and Token Translation service**, facilitated by pilot projects with the support of AARC
- Definition of a **token-based authentication and authorization profile for WLCG**

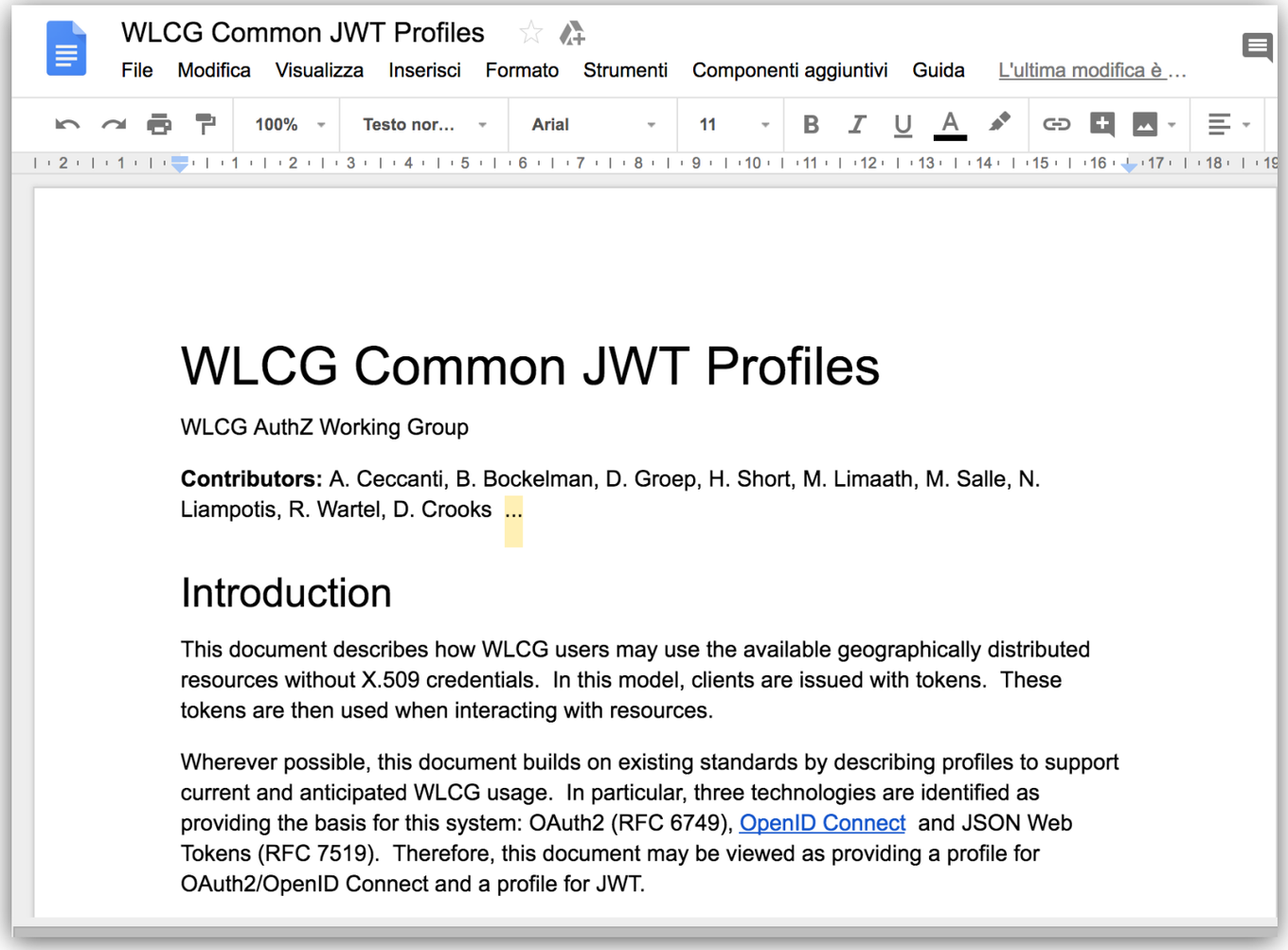


A common profile for Token-based AuthN/AuthZ

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?**



WLCG Common JWT Profiles

File Modifica Visualizza Inserisci Formato Strumenti Componenti aggiuntivi Guida L'ultima modifica è …

WLCG Common JWT Profiles

WLCG AuthZ Working Group

Contributors: A. Ceccanti, B. Bockelman, D. Groep, H. Short, M. Limaath, M. Salle, N. Liampotis, R. Wartel, D. Crooks ...

Introduction

This document describes how WLCG users may use the available geographically distributed resources without X.509 credentials. In this model, clients are issued with tokens. These tokens are then used when interacting with resources.

Wherever possible, this document builds on existing standards by describing profiles to support current and anticipated WLCG usage. In particular, three technologies are identified as providing the basis for this system: OAuth2 (RFC 6749), OpenID Connect and JSON Web Tokens (RFC 7519). Therefore, this document may be viewed as providing a profile for OAuth2/OpenID Connect and a profile for JWT.

Approach:

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

Back to ESCAPE

ESCAPE AAI: possible next steps

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

ESCAPE AAI: possible next steps

Understand how we make and assess progress

- Identify and bring together the "AAI experts" across the communities
  - People that know the experiment/community computing model and can answer nerdy AAI questions
- Do we need AAI-focused, cross-WP communication channels?
  - i.e., a dedicated mailing list or is the e-dios list enough?
- Setup collaborative tools to track requirements collection, integration activities, issues?
  - issue tracker, wiki, …
- Setup a testbed
  - the sooner we find issues, the sooner we start to solve them!

Thanks for your attention.

Questions?

Useful 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-aai.slack.com