Introduction the INDIGO IAM service

Andrea Ceccanti
INFN
andrea.ceccanti@cnaf.infn.it

JUNO Distributed Computing Meeting
July 3rd 2020

Brief intro to the WLCG AAI

WLCG

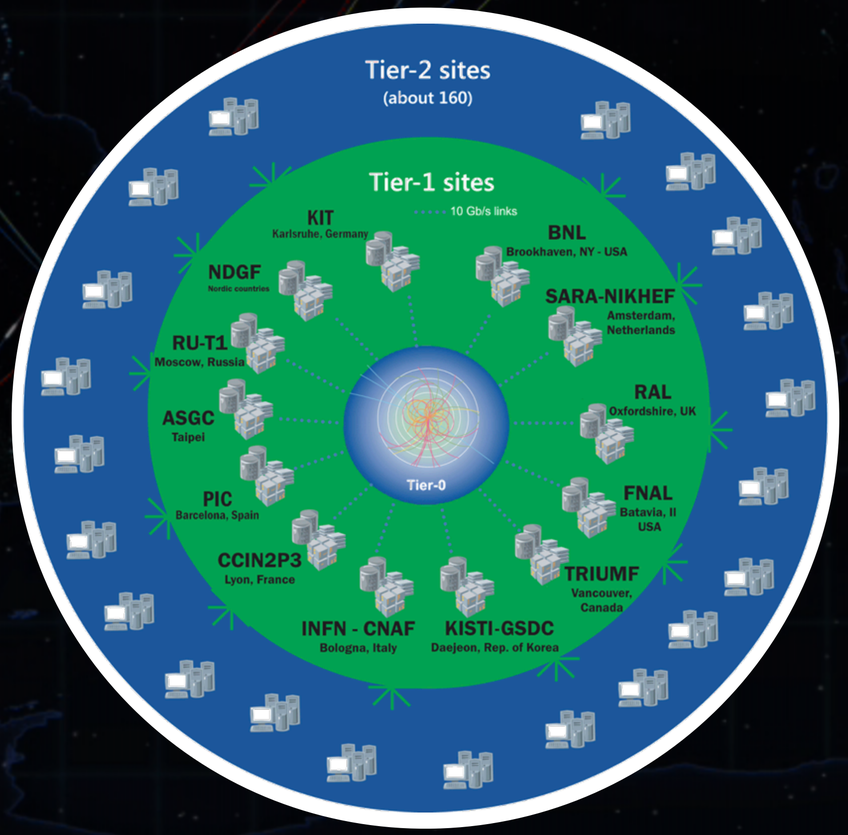
The Worldwide LHC Computing Grid (WLCG), is a distributed computing infrastructure arranged in tiers – giving a community of over 12,000 physicists near real-time access to LHC data.

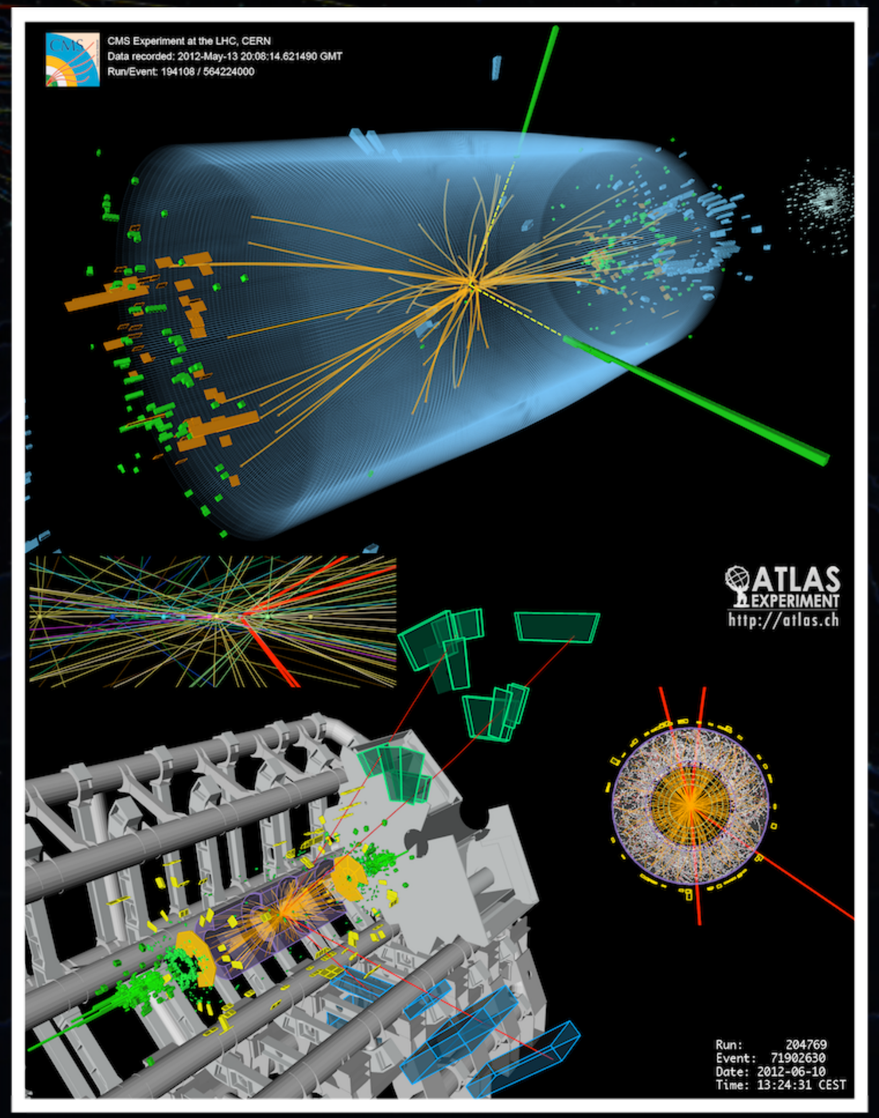
167 sites, 42 countries

~1M CPU cores

~1EB of storage

> 2 million jobs/day





Running jobs: 365118
Active CPU cores: 795836
Transfer rate: 18.35 GiB/sec

Virtual Organizations

WLCG brings together researchers from several institutions distributed all over the world.

Researchers typically collaborate in the context of scientific collaborations which represents **Virtual Organisations** (VOs). **VOs**:

- group together users with **a common purpose**
- represents **a single integration point for infrastructure** providers (Grid sites, public clouds, etc…)
- provide **centralised management** of users enrolment and user lifecycle
- define their **authorization space** by organizing users in groups, assign them roles & other attributes

The WLCG AAI objectives (from 10K mt)

Controlled, secure sharing of computing and storage resources, provided across heterogeneous infrastructures (Grid, Cloud, HPC) in support of the WLCG experiments requirements

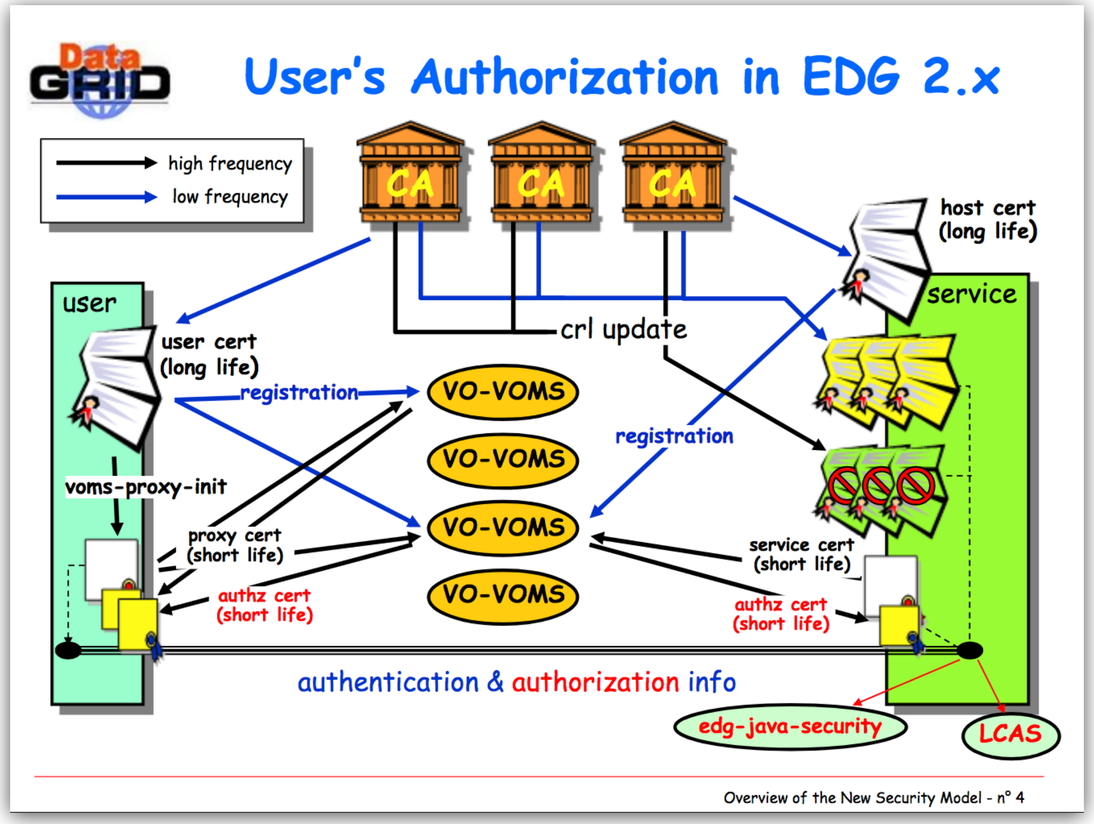
Consistent authentication and authorization across the infrastructure

VO-centric authorization model

Auditing, Traceability, Accounting

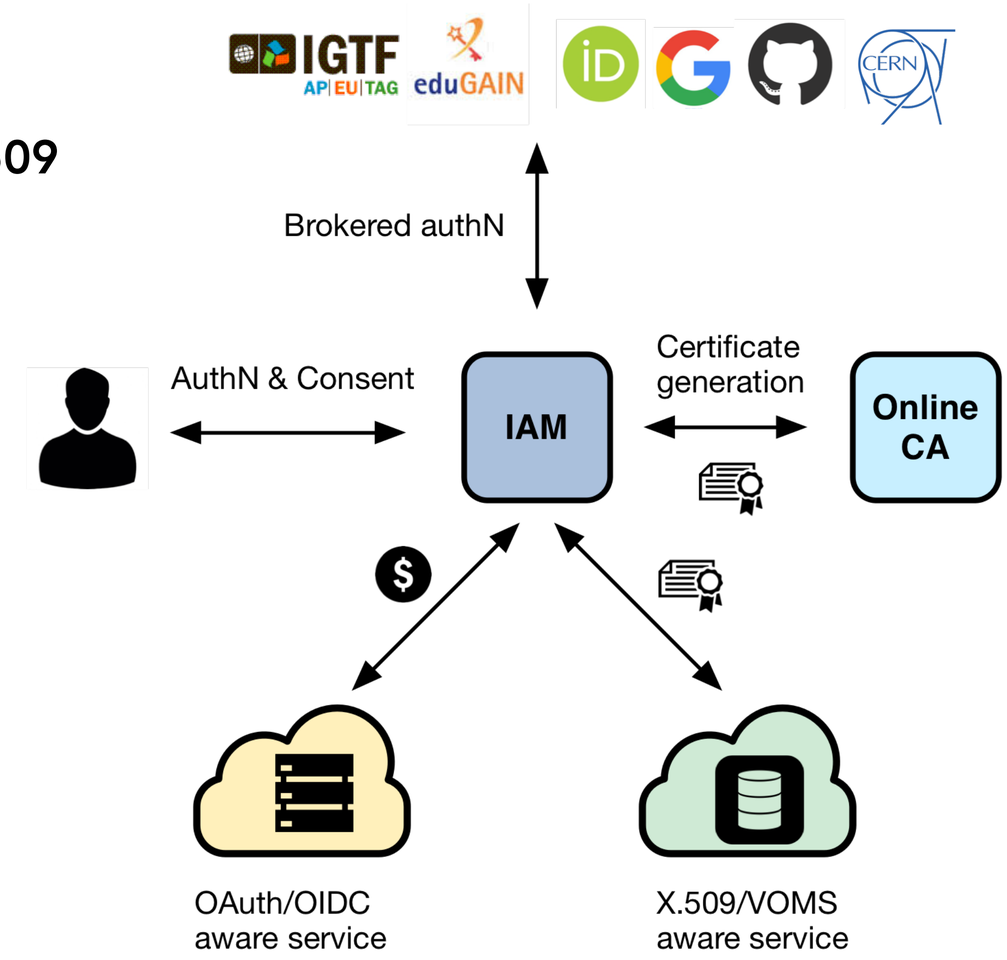
The WLCG AAI evolution

Current, X.509 based AAI



Future, token-based AAI

Move beyond X.509



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:

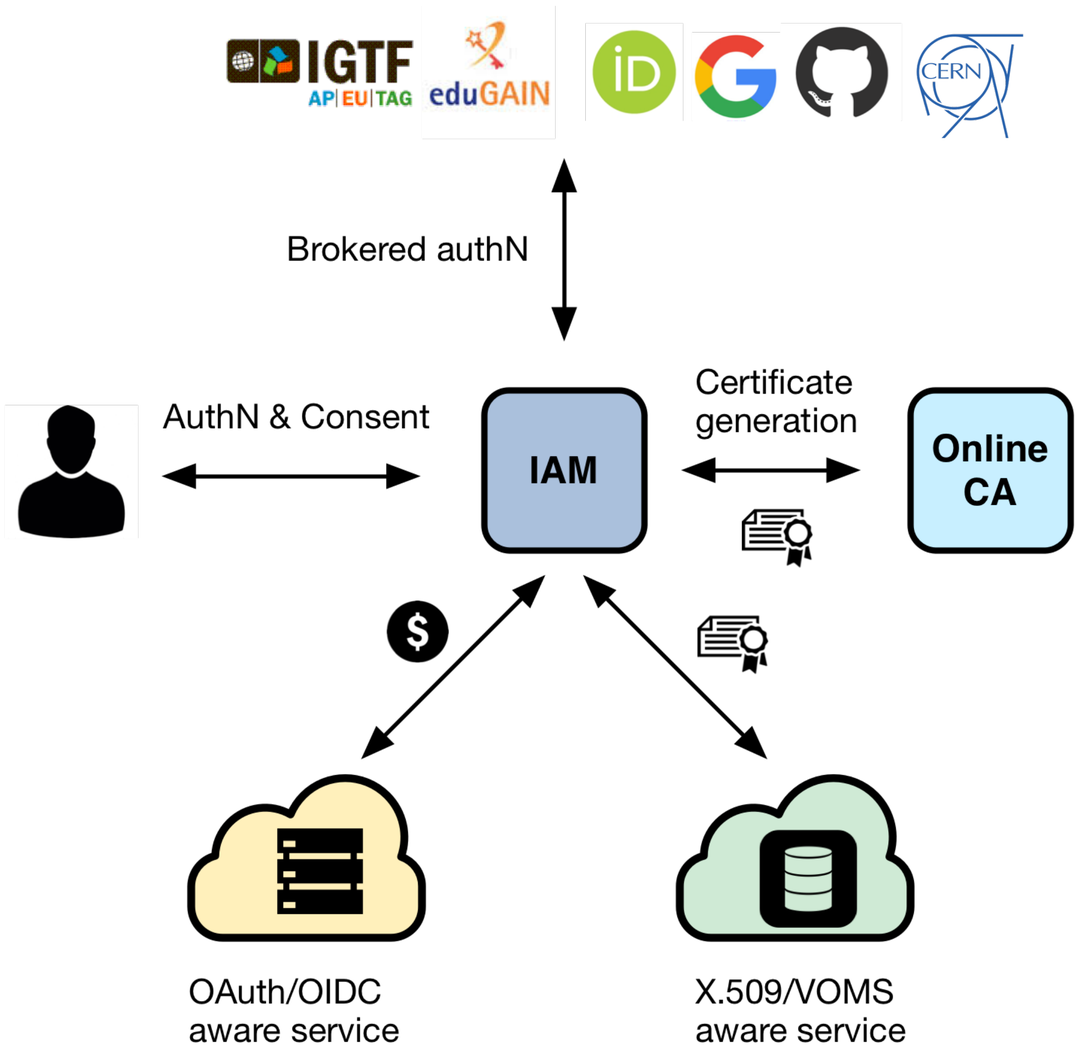
allow a gradual transition to the new AAI!

INDIGO IAM overview

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**
- supports **SCIM-compliant** provisioning APIs

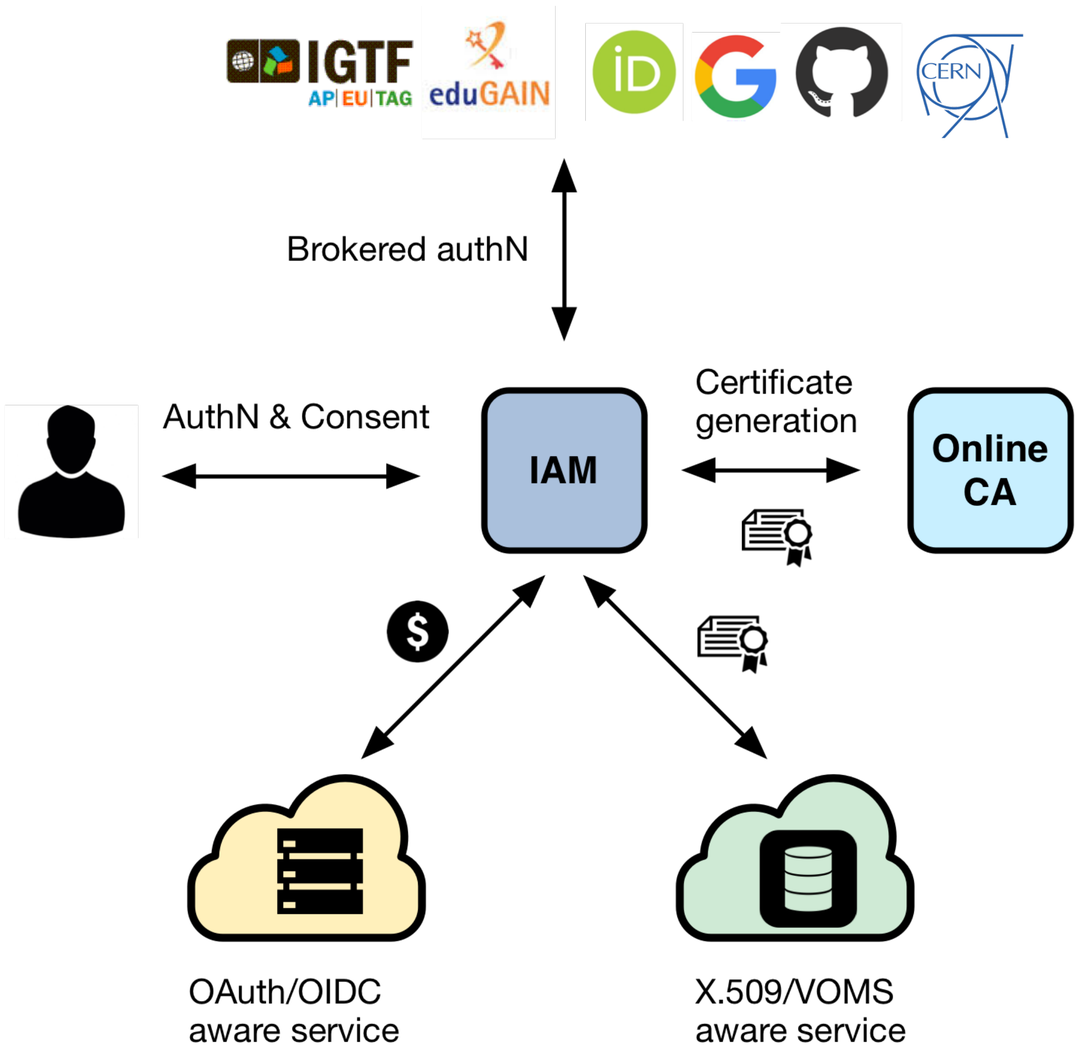


INDIGO Identity and Access Management Service

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

- while ensuring backward compatibility with the existing infrastructure

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



IAM key features

User enrolment & registration service

IAM currently supports two **enrolment flows:**

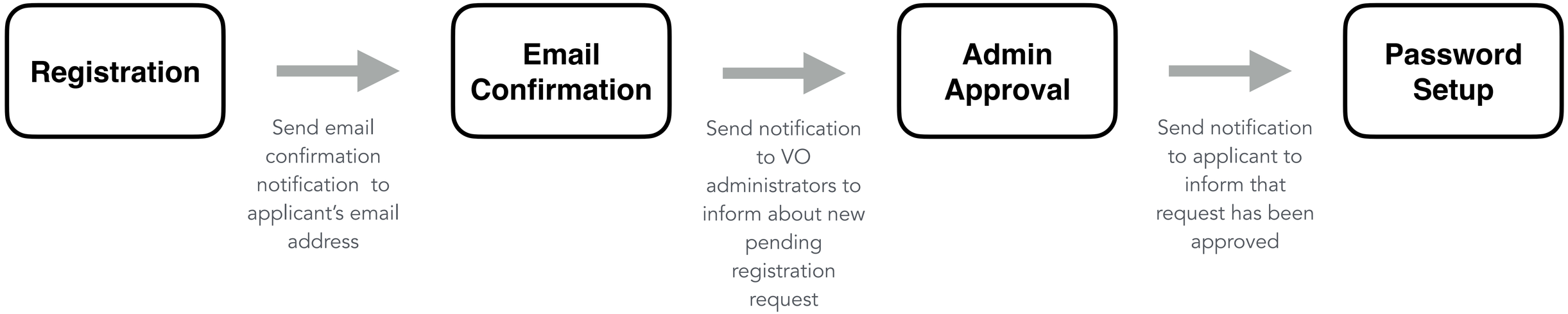
Admin-moderated flow

- The applicant fills basic registration information, accepts AUP, proves email ownership
- VO administrators are informed by email and can approve or reject incoming membership requests
- The applicant is informed via email of the administrator decision

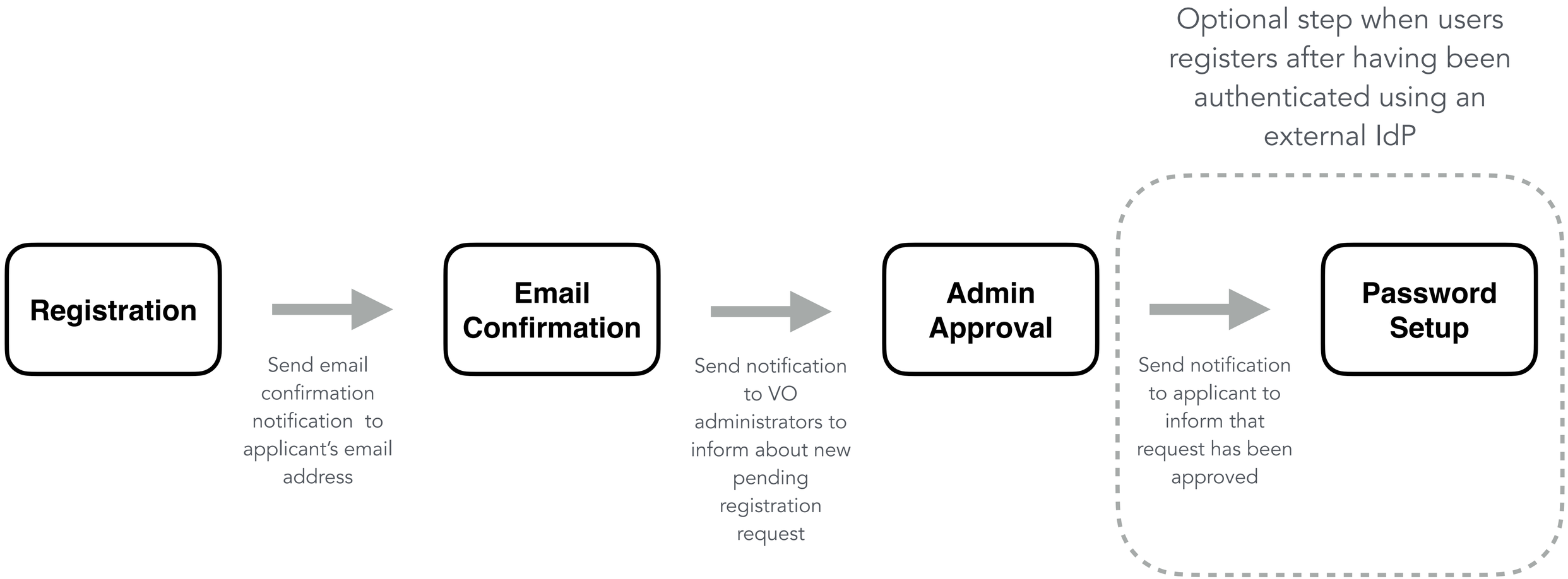
Automatic-enrolment flow

- Users authenticated at **trusted**, **configurable** SAML IdPs are automatically on-boarded, without requiring administrator approval

IAM moderated enrolment flow



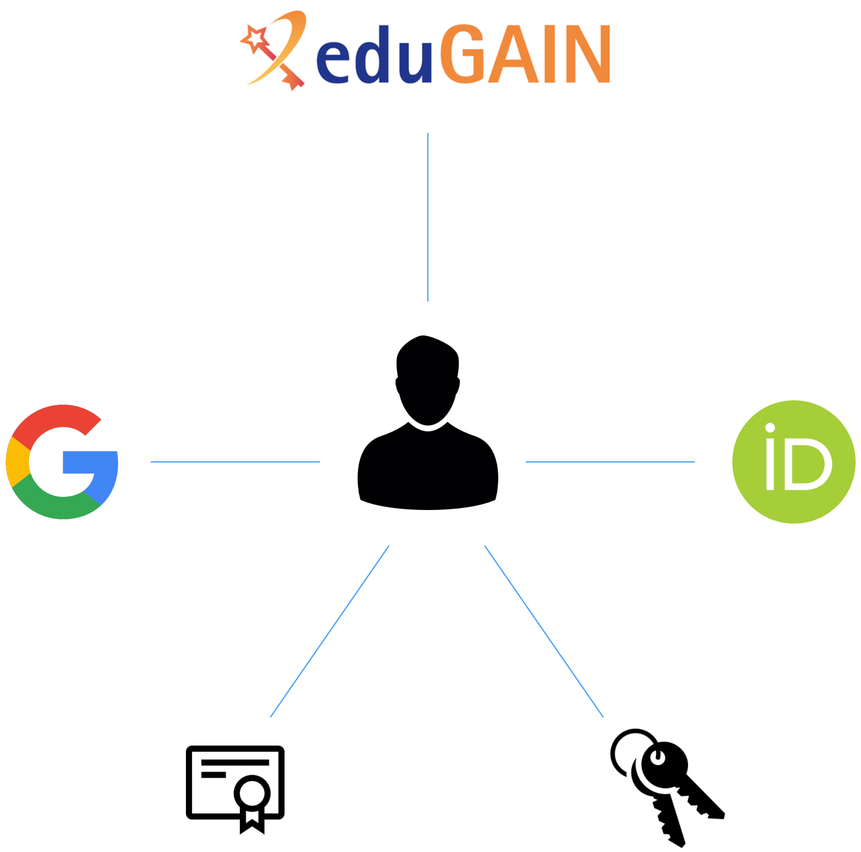
IAM moderated enrolment flow



Flexible authentication & account linking

Authentication supported via

- **local username/password** credentials (created at registration time)
- **SAML** Home institution IdP (e.g., EduGAIN)
- **OpenID Connect** (Google, Microsoft, Paypal, ORCID)
- **X.509** certificates



Users can link any of the supported authentication credentials to their IAM account at registration time or later

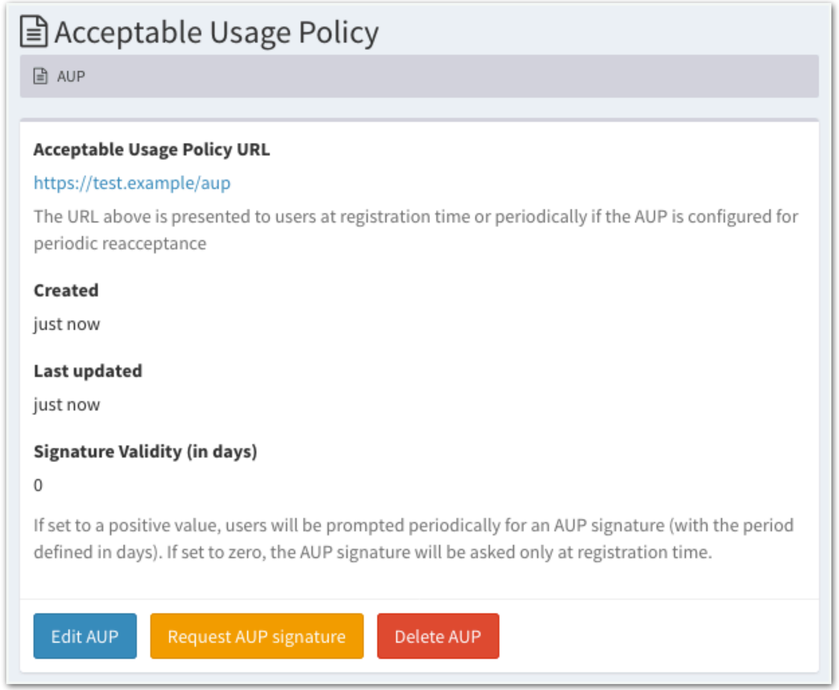
To link an external credential/account, the user has to **prove** that he/she owns such account

AUP enforcement support

AUP acceptance, if enabled, can be configured to be:

- requested once at user registration time
- periodically, with configurable period

User cannot login to the system (and as such be authenticated at authorized at services) unless the **AUP** has been accepted



SCIM provisioning APIs

IAM provides a RESTful API, based on the System for Cross-domain Identity Management (SCIM) standard, that can be used to access information in the IAM database

- users, groups, group memberships, etc…

The API can be used as an integration point towards external systems

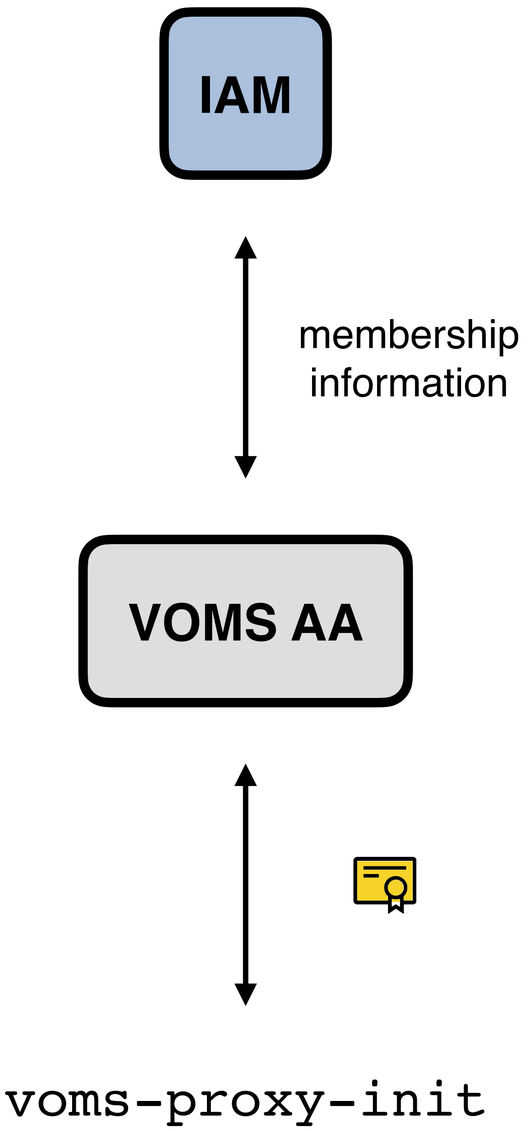
- Example:
  - The SCIM API is used in the integration with the HTCondor batch system to do account pre-provisioning based on IAM account information

VOMS provisioning

IAM includes a VOMS attribute authority micro-service that can encode IAM membership information in a **standard VOMS Attribute Certificate**

Proven compatibility with existing latest supported clients and Grid services

- e.g., data transfers in the ESCAPE data lake testbed rely on this



Easy integration with relying services

Standard OAuth/OpenID Connect enables **easy integration** with off-the-shelf services and libraries.

IAM has been successfully integrated with

- Openstack, Atlassian JIRA & Confluence, Moodle, Rocketchat, Grafana, Kubernetes, JupyterHub, **dCache, StoRM, XRootD (HTTP), FTS, RUCIO, HTCondor**

Software Quality in IAM

Aim to have **~90% unit test coverage on all code**:

- now 33K LoC, 86,4% branch coverage, >1.2K tests

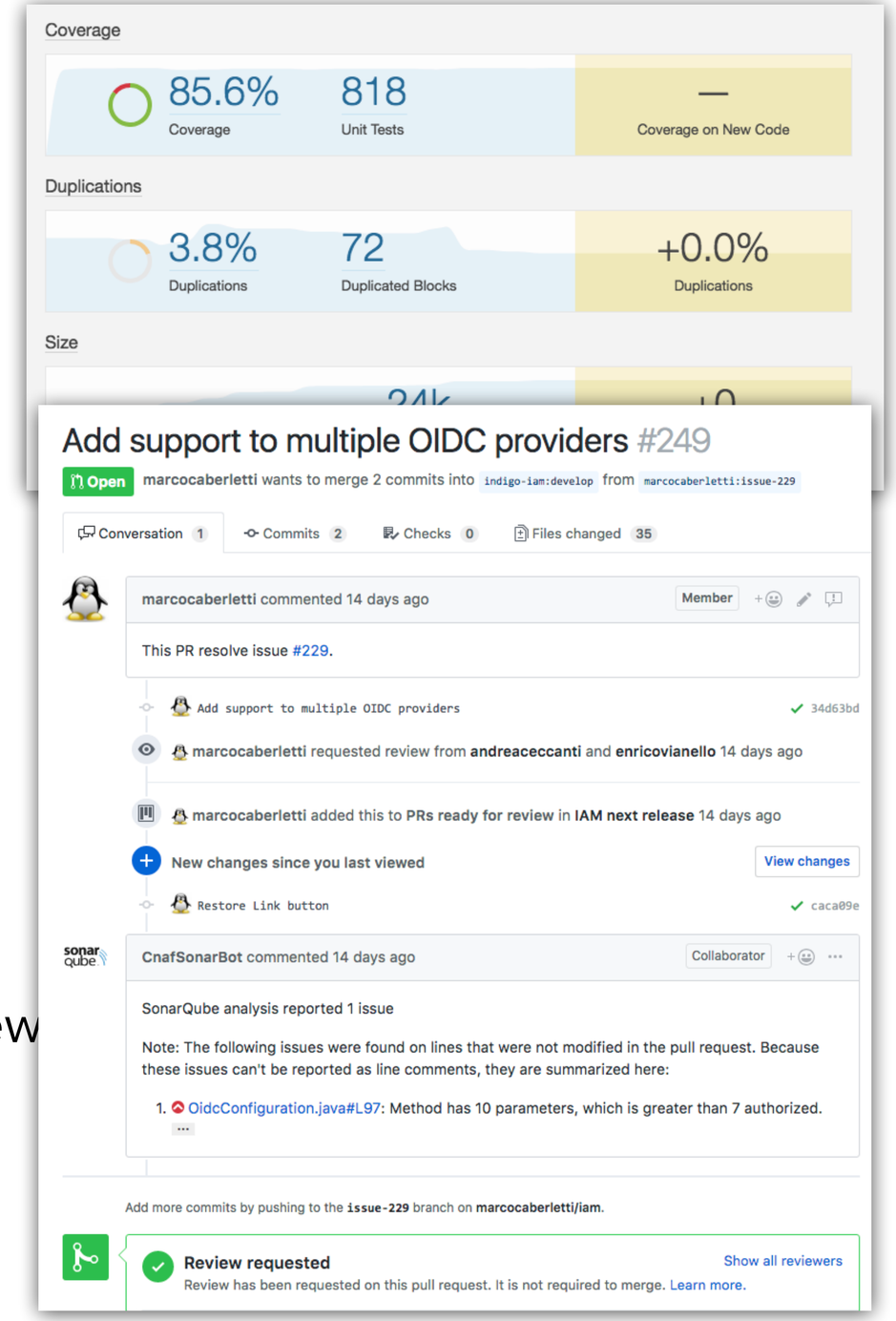
Open, **test-driven** development process

Static analysis tools

- SonarCloud IAM page

Multiple test suites

- **Unit tests**
- **Frontend test suite** (based on Selenium and Robot framew
- **Deployment tests** (in CI)



Deployment options

IAM as a service

- INFN provides IAM as a service to partner research communities. In this scenario, a dedicated IAM instance is deployed on the INFN infrastructure and configured according to the community needs. INFN takes care of keeping the service operational and up-to-date, while administrative control on the IAM instance is granted to the community.
- See https://indigo-iam.github.io/docs/v/current/iam-aas/

IAM on premise deployment

- IAM is an Apache-licensed identity solution, for which we provide **Docker images** on **Dockerhub** and RPMs and Deb packages
- See https://indigo-iam.github.io/docs/v/current/admin-guide/

Token-based AuthN/Z

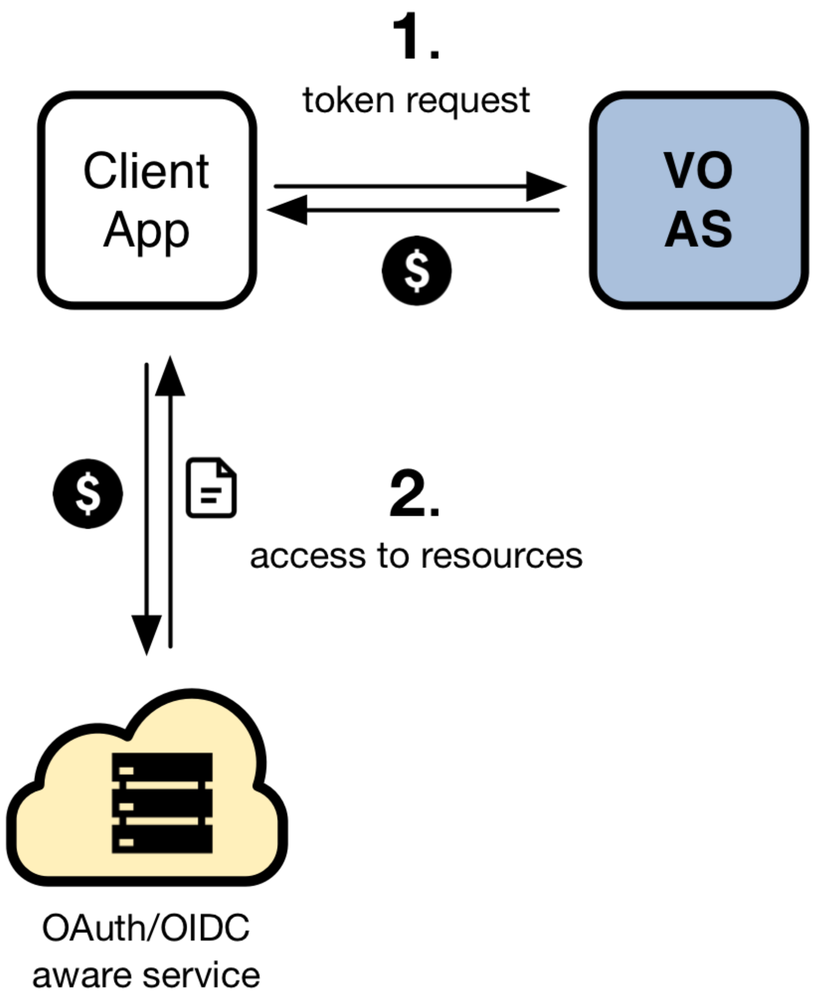
Token-based AuthN/Z from 10000 mt

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



In practice…

The central authorization server (i.e., IAM) provides **attributes** that can be used for authorization at services, e.g.:

- groups/roles, e.g.: **analysis**, **production-manager**
- capabilities, e.g.: **storage.read:/, submit-job**

This information is exposed to services via **signed JWT tokens** and **via OAuth/ OpenID Connect protocol message exchanges** (aka flows)

Services can then **grant or deny access** to resources based on this information. Examples:

- allow read access on the **/analysis** folder to all members of the **analysis** group
- allow read access on the namespace to anyone with the capability **storage.read:/**

The WLCG Common JWT profile

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

Demo

Thanks!

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

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

Contacts:

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