WORKSHOP GARR 2026

NET MAKERS

INDIGO IAM: Trust Relationships and Interoperability from OpenID Federation to EOSC

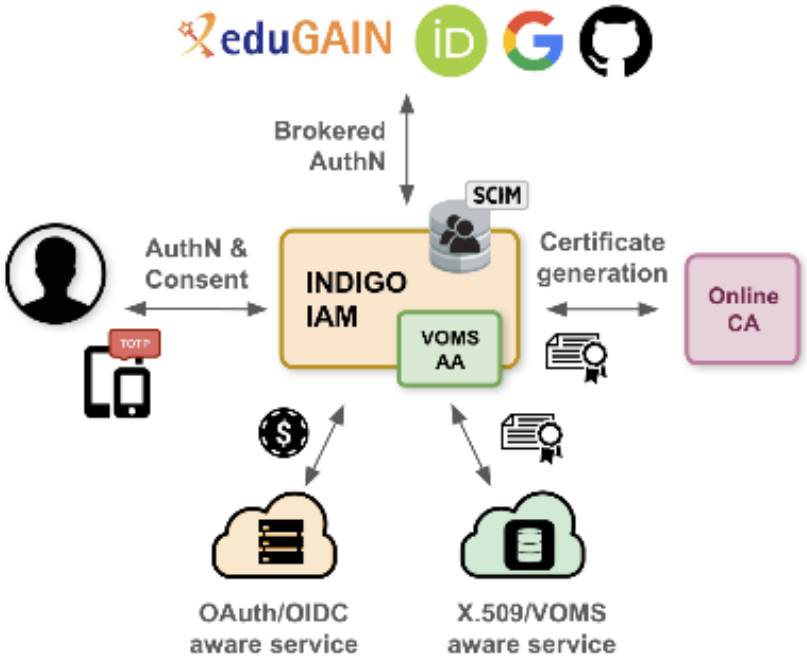
Roberta Miccoli

INFN CNAF

INDIGO IAM

Identity and Access Management (IAM) service

- Open-source software
- Spring Boot application written in Java
- Centralized solution for identity, authentication and authorization management
- Designed to enable secure access to distributed services and resources
- Supports widely adopted standards and protocols, including OAuth 2.0, OpenID Connect, SAML and SCIM
- Used as an IAM component in distributed scientific infrastructures, e.g., WLCG



Latest developments

Latest release:
IAM v1.15.0

Main features from latest IAM releases:

- option to not store access tokens in the database
- option to make 2FA mandatory
- client secrets stored in the database as SHA-256 digests
- support for Mutual TLS client-bound access tokens
- support for AARC guidelines
  - e.g., AARC-G052 OAuth 2.0 Proxied Token Introspection
- support for OpenID Federation

Future work

IAM core evolution

A major development priority is the evolution towards IAM v2, adopting modern Spring Security.

Dashboard modernization

A new dashboard has been developed as a TypeScript application based on Next.js and React, replacing the deprecated AngularJS-based frontend.

Next step: deploy the new dashboard to production instances.

OpenID Federation

WORKSHOP GARR 2026

NET MAKERS

Support for OpenID Federation

The OpenID Federation specification describes how two entities wishing to interact can dynamically discover and resolve trust relationships and retrieve metadata for a given protocol through a third-party Trust Anchor.

Features supported by INDIGO IAM:

- Federation Entity Configuration
- Trust Chain Resolution and Validation
- Explicit Registration
- Automatic Registration

Enable the OpenID Federation profile!

Entity Configuration

Self-signed Entity Statement (JWT) available at */.well-known/openid-federation*

JWT header

```
{
  "kid": "rsa1",
  "typ": "entity-statement+jwt",
  "alg": "RS256"
}
```

OpenID Provider (OP) metadata

```
{
  "sub": "https://iam-dev.cloud.cnaf.infn.it/",
  "metadata": {
    "openid_provider": {
      "response_types_supported": [
        "code",
        "token"
      ],
      "client_registration_types_supported": [
        "explicit",
        "automatic"
      ],
      "federation_registration_endpoint": "https://iam-dev.cloud.cnaf.i
nfn.it/iam/api/oid-fed/client-registration",
      "request_uri_parameter_supported": false,
```

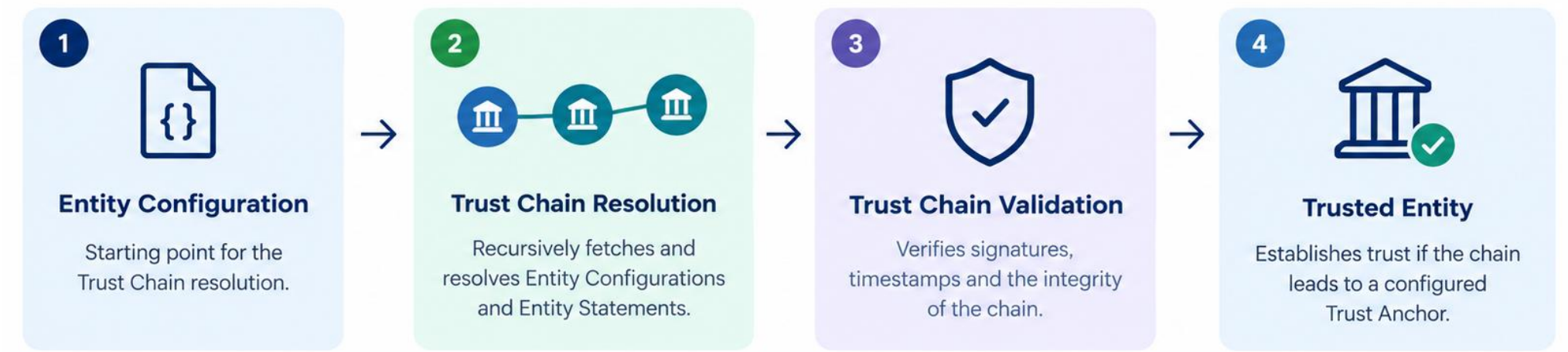
Relying Party (RP) metadata

```
    "openid_relying_party": {
      "client_registration_types": [
        "explicit"
      ],
      "application_type": "web",
      "jwks_uri": "https://iam-dev.cloud.cnaf.infn.it/jwk",
      "redirect_uris": [
        "https://iam-dev.cloud.cnaf.infn.it/openid_connect_login"
      ]
    }
```

```
"authority_hints": [
  "https://trust-anchor.sandbox.eosc.grnet.gr"
],
```

Trust Chain Resolution and Validation

Starting from an Entity Configuration, INDIGO IAM recursively resolves and validates the Trust Chain up to a configured Trust Anchor.



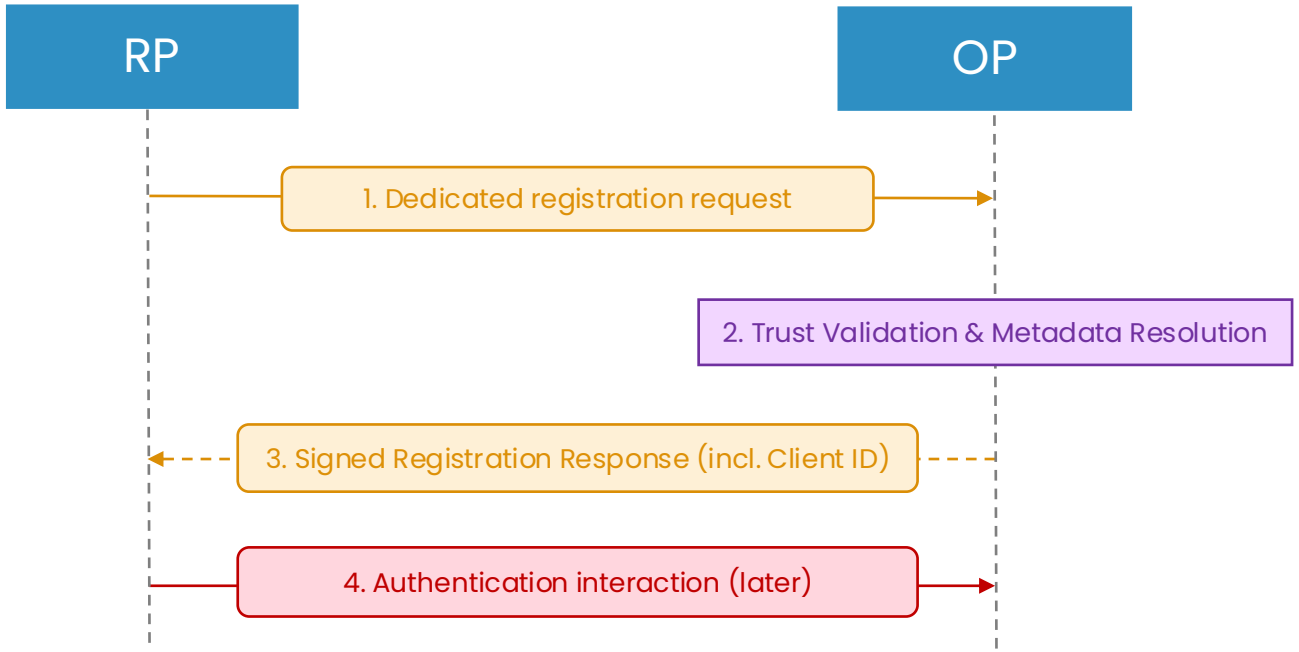
Explicit Registration

✓ RP role
✓ OP role

Timing: client relationship established before authentication

Process: dedicated federation registration request sent ahead of time

Validation: OP validates federation trust and resolves RP metadata

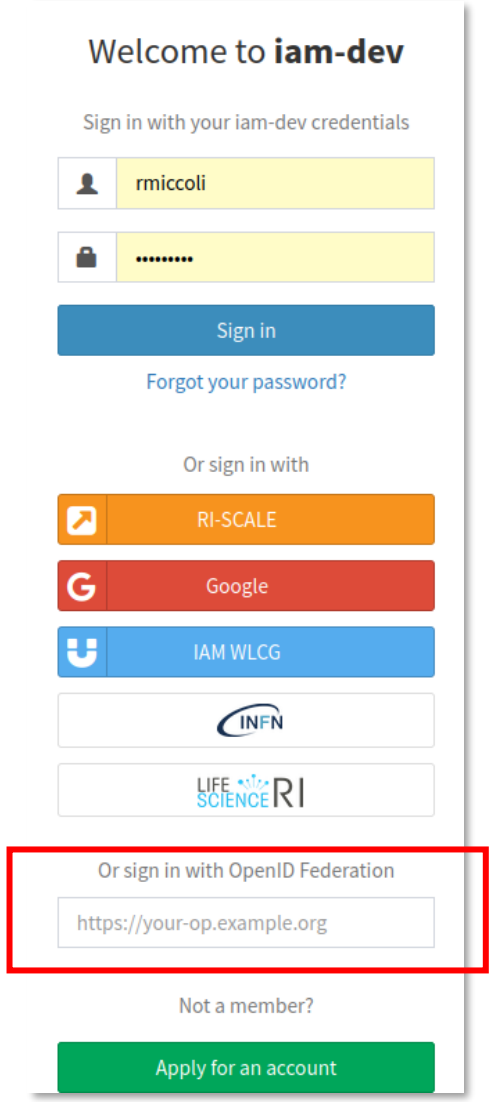


Explicit Registration

Adding *?oidfed=y* to the IAM login URL enables the OpenID Federation login option.

This allows IAM, **acting as an RP**, to initiate the Explicit Registration flow with a federated OP.

Next step: explore the [OpenID Federation Entity Collection Endpoint draft] to enable dynamic discovery and selection of federated OPs.



https://iam-dev.cloud.cnaf.infn.it/login?oidfed=y

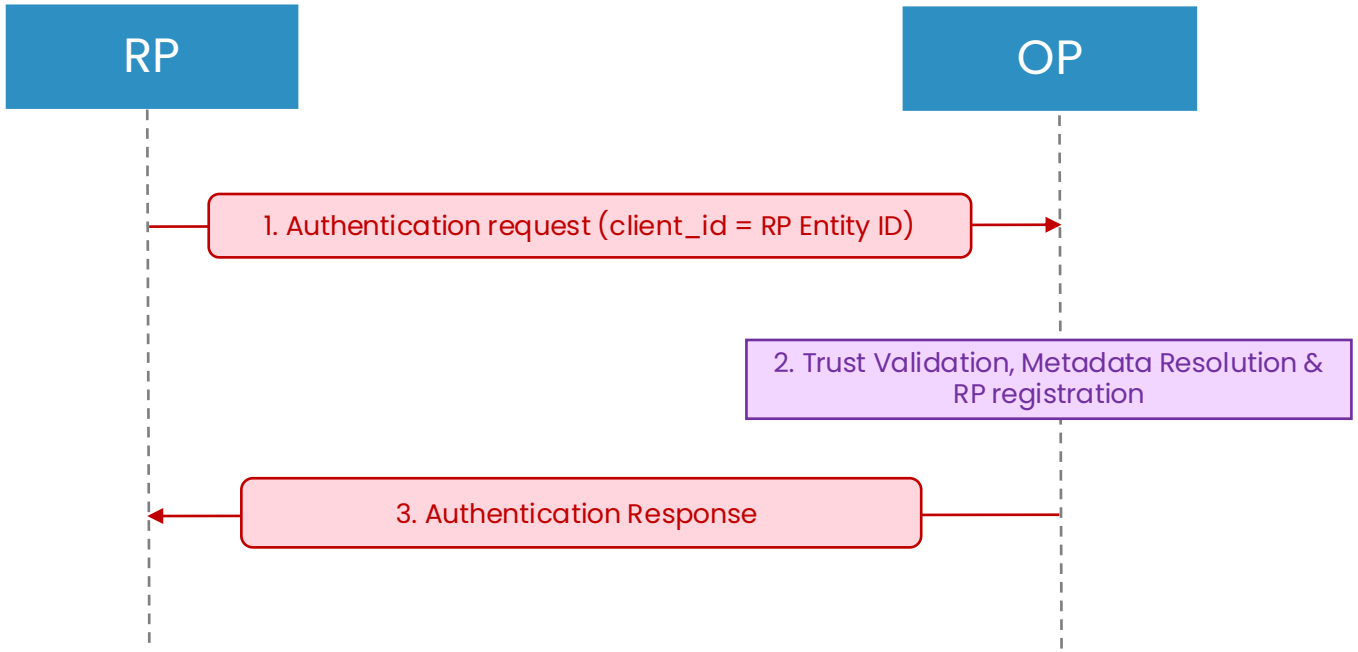
Automatic Registration

✓ OP role

Timing: registration happens during authentication interaction

Process: no prior client registration required

Validation: OP validates federation trust and resolves RP metadata



Validation and testing

Validation tool

- OpenID Federation Browser used to inspect and explore federation entities and trust relationships

Testing environment

- Testing performed using dedicated playground instances in the context of the *EOSC Beyond* European project
- OpenID Federation Conformance Testing ongoing



AARC compliance

The AARC-G100 guideline defines how to establish trust between AARC-compliant AAI services using OpenID Federation.

Two trust profiles:

- AARC-G100.1 Basic Trust Model
  - Trust Authority Onboarding
  - Trust Chain Validation
  - Policy Requirements

✓ Fully supported

- AARC-G100.2 Fine-Grained Trust Model

  [built on all G100.1 requirements]
  - Metadata Policies
  - Trust Marks

Planned work

- Federation-based trust evaluation when acting as a Resource Server (RS)
- Metadata Policies integration
- Trust Marks integration

AARC-G100.2

EOSC AAI Federation

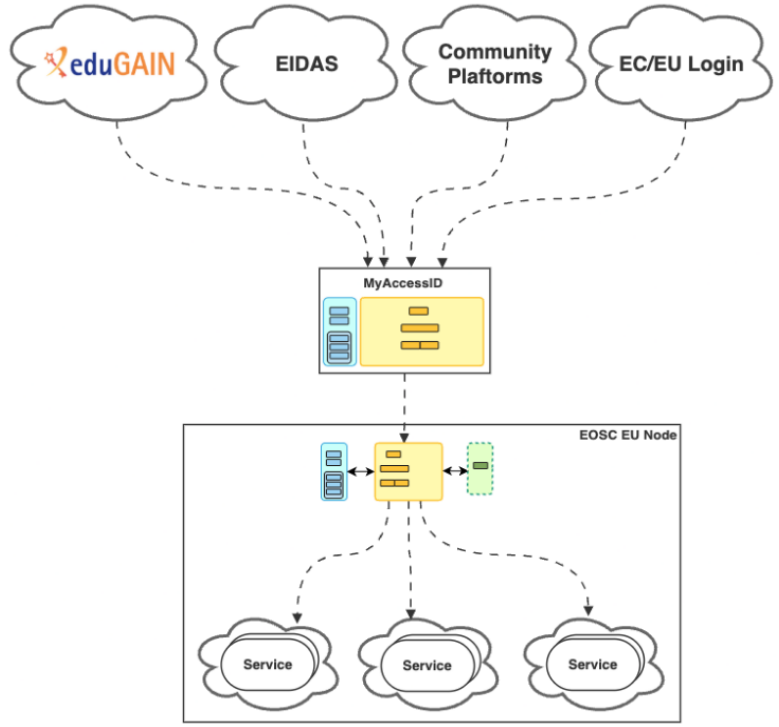
WORKSHOP GARR 2026

NET MAKERS

EOSC AAI Federation

- Each Node integrates with the **Central Hub** (MyAccessID), which mediates interactions between participating AAIs
- Infrastructure Proxies act as OIDC RPs
- Community AAIs act as OPs
- Nodes and their AAI entities are recorded in the EOSC AAI Federation Registry
- Cross-Node Token Introspection is routed through the Central Hub

Target Architecture: **Dynamic trust establishment with OpenID Federation**



EOSC AAI Architecture 2025

OpenID Federation Pilot

Evaluate OpenID Federation as the trust layer of the EOSC AAI Federation, moving towards a full-mesh model based on a shared Root Trust Anchor and Node Infrastructure Proxies as federation members.

Requirements for participation:

- An Infrastructure Proxy supporting OpenID Federation
- Dynamic Client Registration to register a proxy as a client of another proxy
- Cryptographic verification of signed Entity Statements
- Support for Trust Anchor-supplied Metadata Policies and enforcement of constraints

✓ AARC-G100.1

✗ AARC-G100.2

**Scope: Testing instances of EOSC Nodes only — production instances are excluded*

INDIGO IAM at CERN EOSC Node

INDIGO IAM has been selected as the core AAI component of the **CERN EOSC Node**.

Current setup

- Just-in-Time (JIT) account provisioning
- Proxied Token Introspection
- Identity Provider (IdP) Hinting
- MFA enabled and enforced for all users
- Connected to the MyAccessID test environment
- Deployed in the development instance of the CERN EOSC Node

Thank you!

Questions?

WORKSHOP GARR 2026

NET MAKERS

References & contacts

- IAM on GitHub
- New IAM dashboard on GitHub
- IAM documentation

Mailing list:
iam-support@lists.infn.it

Backup slides

INDIGO IAM Community

The INFN-CNAF Software Development team benefits from fruitful collaboration with developers and IT experts, particularly from STFC and CERN.

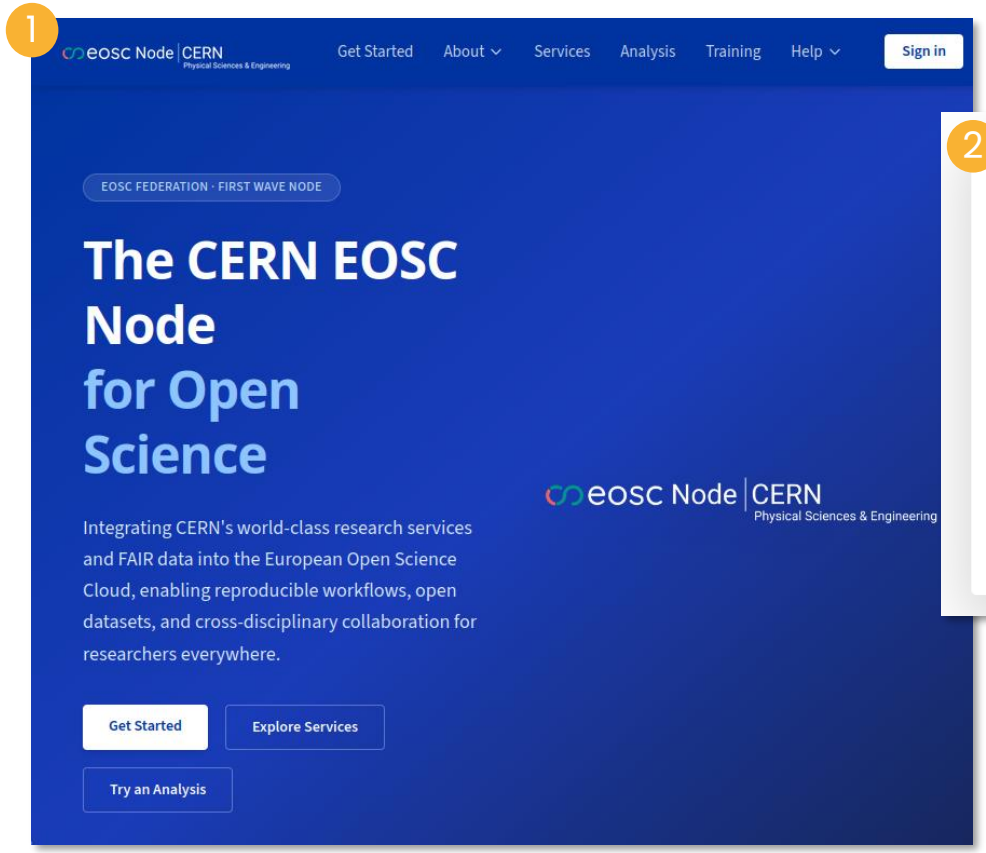
Community Activities:

- Bi-weekly community meetings since October 2023
- Workshops and hackathons - approximately two per year since February 2023

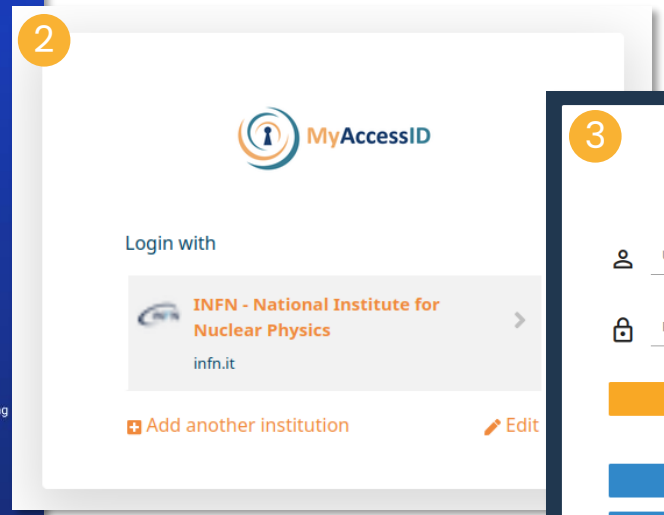
Communication:

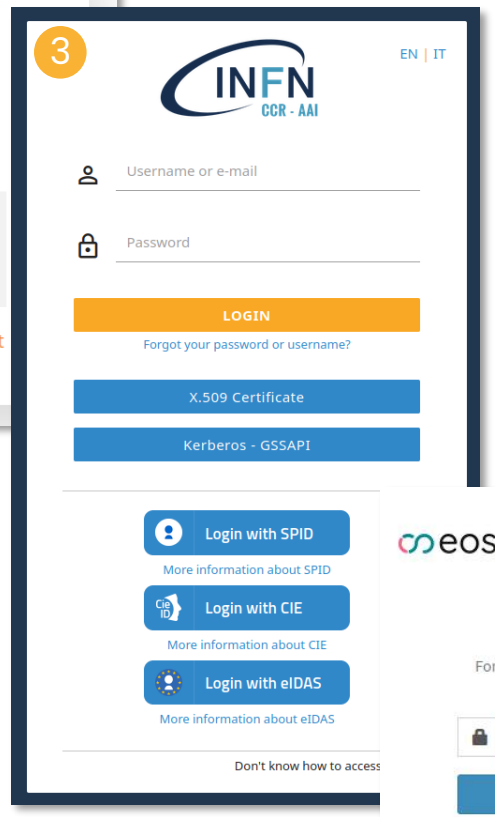
Dedicated Mattermost channels hosted by CERN.

CERN EOSC Node Authentication with INDIGO IAM

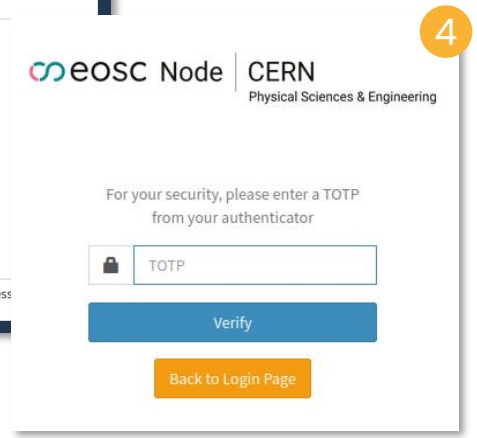


https://eosc.cern/

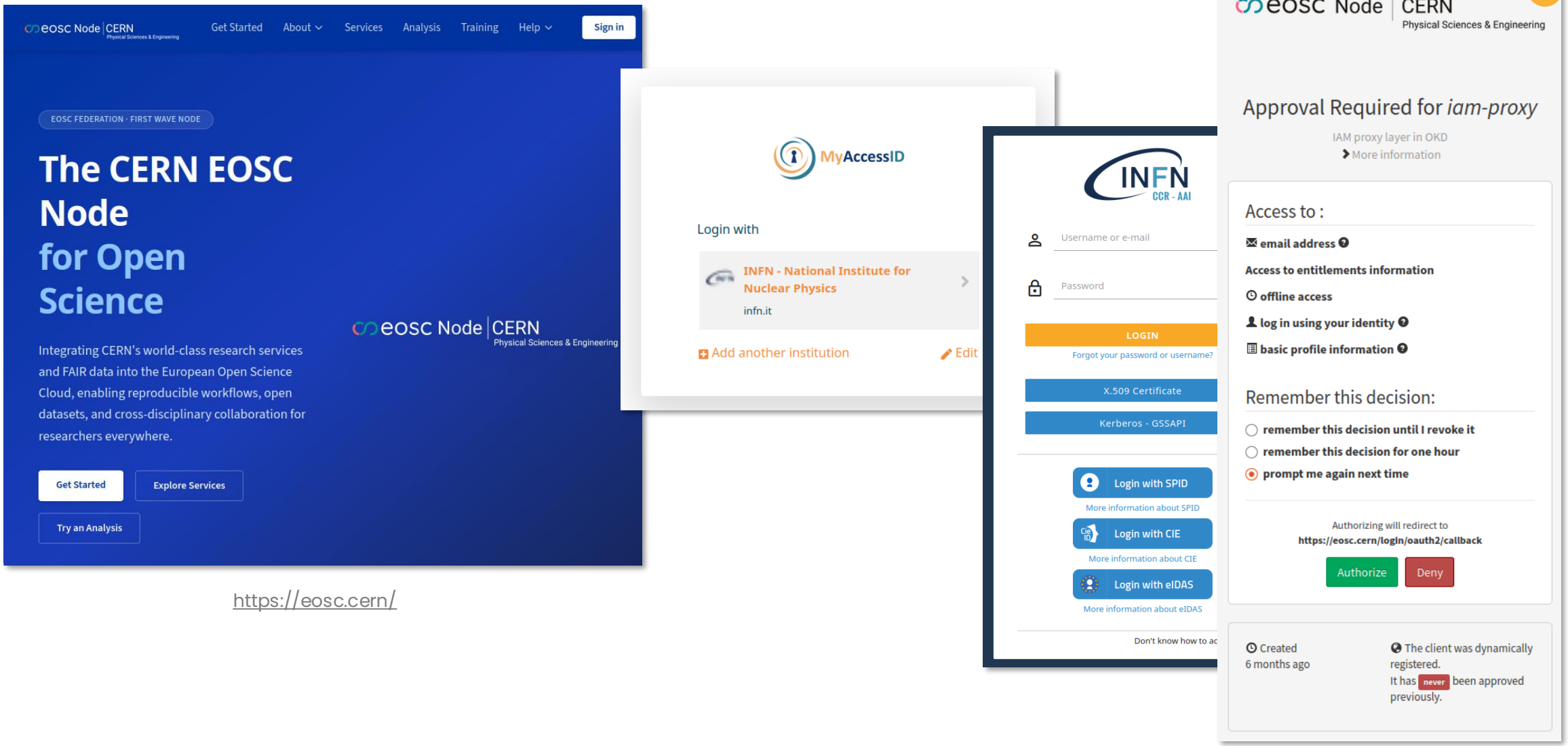




IAM instance:
https://eosc-auth.cern.ch/

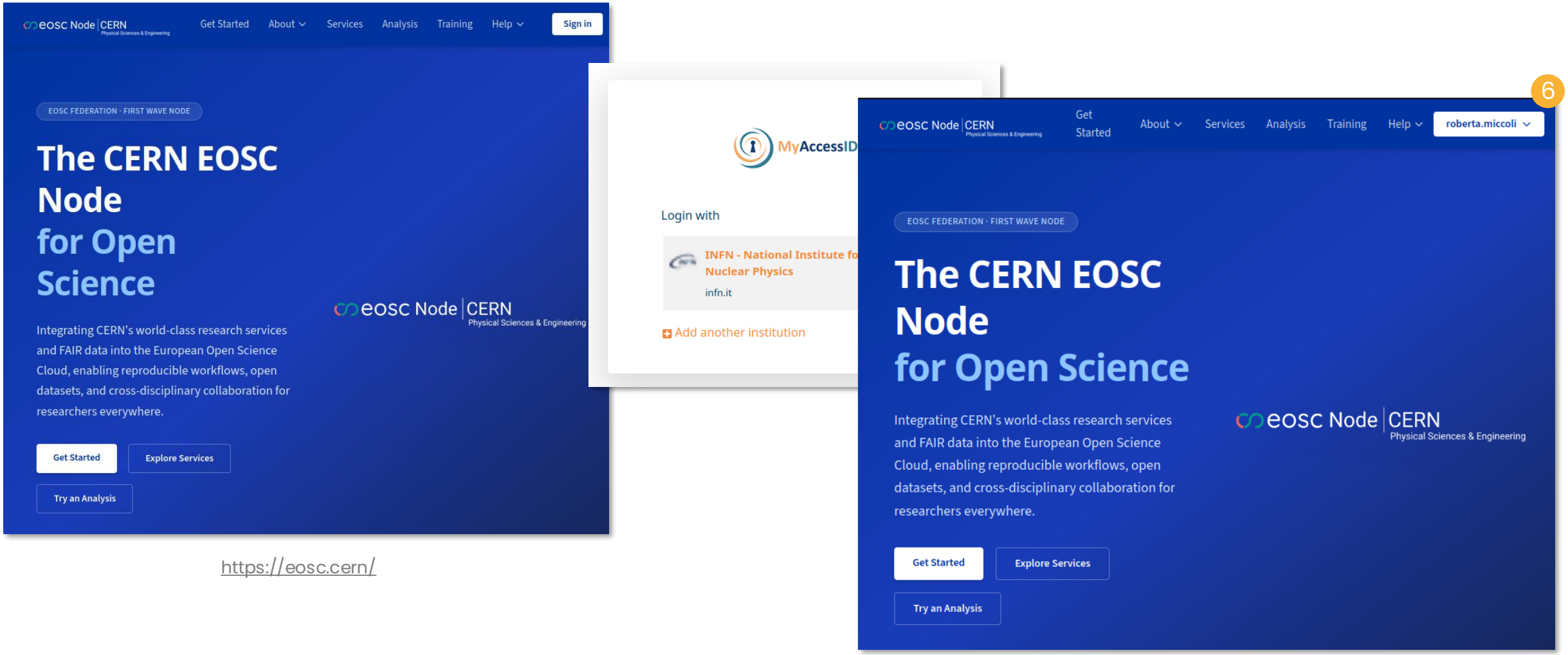


CERN EOSC Node Authentication with INDIGO IAM



https://eosc.cern/

CERN EOSC Node Authentication with INDIGO IAM



https://eosc.cern/