Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

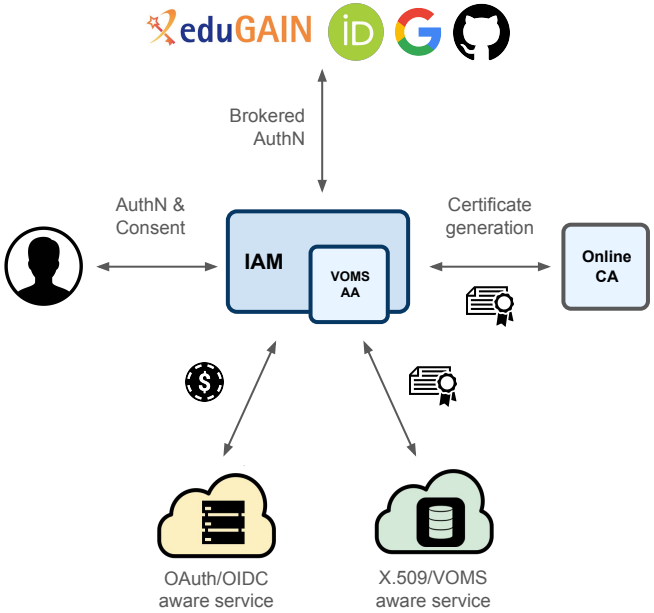
Evoluzione di INDIGO IAM: sicurezza, interoperabilità e nuove funzionalità

Roberta Miccoli - INFN CNAF

Workshop sul Calcolo nell'INFN - La Biodola | 26 - 30 maggio 2025

INDIGO IAM in one slide

- Standard **OAuth2 Authorization Service** and **OpenID Connect Provider**
  - Easy integration with (web) applications
- **Java** application based on the **Spring Boot** framework
- **Multiple authentication mechanisms**
  - SAML, OpenID Connect, X.509, username/password
  - 2FA available
- **Account linking**
- Moderated and automatic user enrollment
- Enforcement of **AUP acceptance**
- VO membership management
- Issuance of **JWT** tokens and VOMS attribute certificates with **identity** and **membership information**, **attributes** and **capabilities**
- Typically deployed as a **Docker container**



Synergies with other projects

- First developed in the context of the H2020 INDIGO DataCloud project

- **Selected by the WLCG management board** to be the core of the future, token-based WLCG AAI

- INFN commitment for the foreseeable future, with additional support of several **Italian and European projects**

RI SCALE

eosc | BEYOND

AARC

terabit

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Latest release IAM v1.12.0

- **Full support of MFA (*experimental*)**
  - **integration with SAML/OIDC external providers and X.509 certificates**
- Support for RFC 8707 (*resource indicators*) to request AT audience
- Support for AARC-G026 guideline (*guidelines for expressing community user identifiers*)
- Assigning ownership of oidc-agent clients to the user who approved them
- Addition of a `READER` role that gives read access to users and groups info
- Improvement of VOMS logs
- . . .

Project board v1.12.0
CHANGELOG v1.12.0

Development roadmap

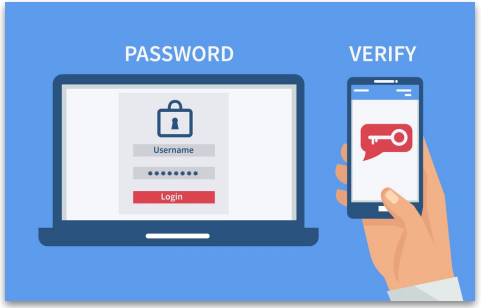
- **Security**
  - Enforce mandatory use of MFA
  - Further consolidate the security requirements related to MFA with the INFN CNAF security team
- **Superseded obsolete dependencies**
  - MITREid → Spring Authorization Server
  - AngularJS → React JS
- **Interoperability** focus
  - Support OpenID Federation
  - Improve conformance with AARC BluePrint Architecture and its guidelines
- **Scalability and performance** improvements
  - Access tokens not stored in the database
  - Dedicated garbage collector service
  - Fine grained AuthZ with Open Policy Agent (OPA)

The core of the development team is mainly at CNAF, with significant contributions from other people at INFN, CERN and STFC

Two-Factor Authentication (2FA)

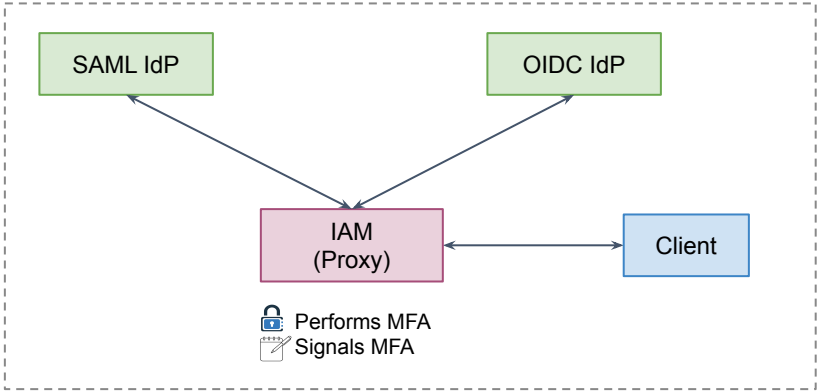
Two-Factor Authentication (2FA)

- Introduced as an *experimental* feature
  - applicable to **login with username and password** since IAM v1.11.0
  - integrated with **SAML**, **OIDC remote providers** and **X.509 certificates** since IAM v1.12.0
- 2FA enabled by configuration
  - using `mfa` spring profile (*deployment config*)
  - by default disabled for all users
  - each user can enable/disable it from dashboard
  - IAM administrator can disable it for each user



Integration with SAML/OIDC external providers

- Work also carried out for the *EOSC Beyond* project
- In compliance with the REFEDS MFA profile and RFC 9470 (*OAuth 2.0 Step Up Authentication Challenge Protocol*), ensuring standardized and interoperable MFA signaling across federated environments



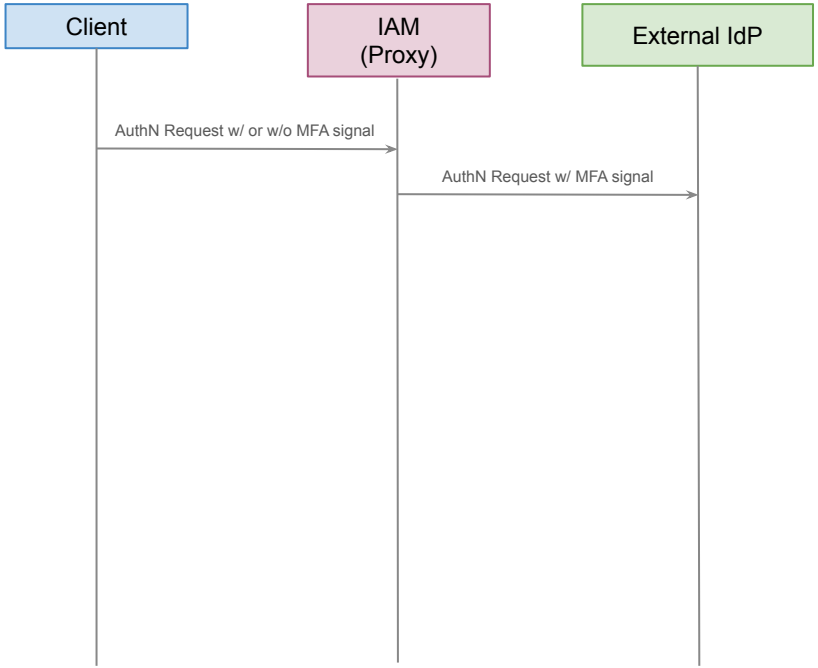
Integration with SAML/OIDC external providers

- When the user authenticates via an external OIDC/SAML provider
  - if MFA was used, IAM skips second-factor authentication, based on the information received from the remote IdP
    - `acr` claim (for OIDC) in the ID token
    - `AuthnContextClassRef` (for SAML) in the SAML assertion
  - if MFA was not used, IAM performs second-factor authentication (if MFA is enabled)
- To prevent IdP failures, IAM optionally requests MFA by
  - including the `acr_values` parameter in the OIDC authN request
  - specifying multiple `AuthnContextClassRef` values in the SAML authN request
- IAM signals 2FA in the access token, ID token and introspection response as `acr` claim (e.g., `acr=https://refeds.org/profile/mfa`)

Integration with SAML/OIDC external providers

Description of the implemented flow

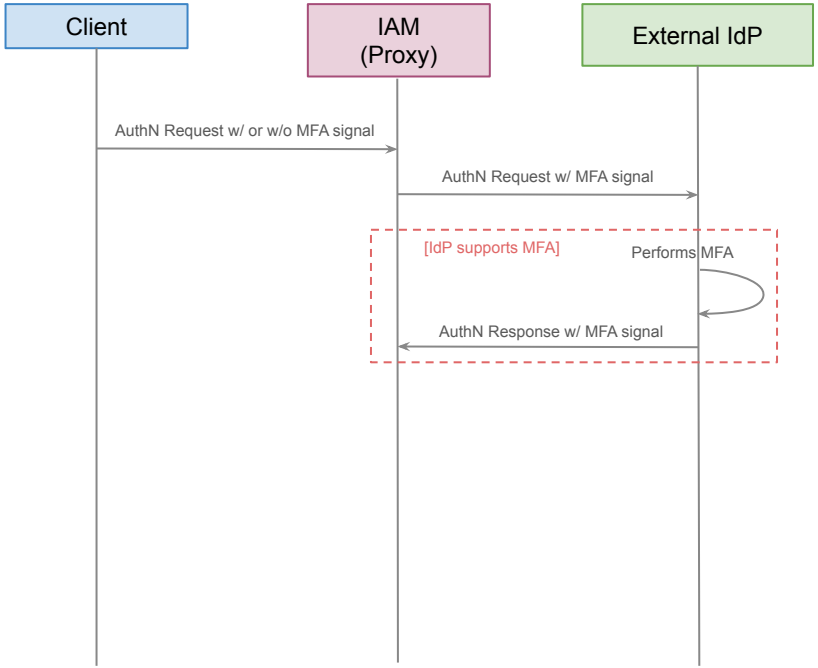
1. **Client initiates the authentication**
   - If the Client supports MFA signaling, the Client sends an authN request to IAM (Proxy), signaling the requirement for MFA
   - If the Client does not support MFA signalling, the Client sends an authN request to IAM
2. **IAM passes authN request to external IdP** by signaling MFA as an *optional* requirement to prevent IdP failures



Integration with SAML/OIDC external providers

Description of the implemented flow

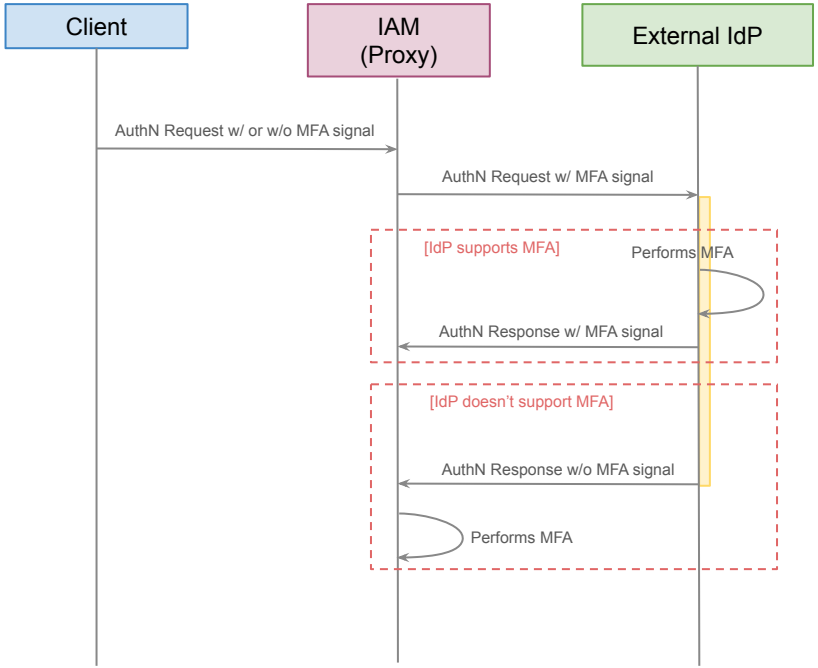
3. **IdP MFA alternatives**
   - If the IdP supports MFA and can signal it, the IdP performs the MFA process (*user is prompted to provide a second authentication factor*) and responds to IAM with an authN response that includes an MFA signal, confirming that MFA has been performed successfully



Integration with SAML/OIDC external providers

Description of the implemented flow

3. **IdP MFA alternatives**
   - If the IdP does not support MFA or cannot signal it, the IdP responds to IAM with an authN response that does not include the MFA signal, indicating that MFA was not performed (or signaled)
     - IAM then performs the MFA process itself, if enabled

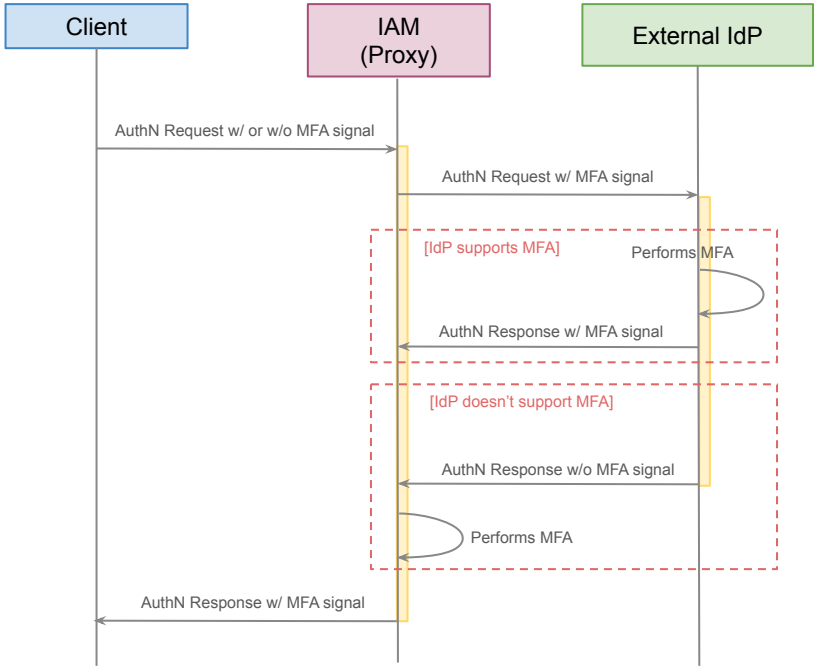


Integration with SAML/OIDC external providers

Description of the implemented flow

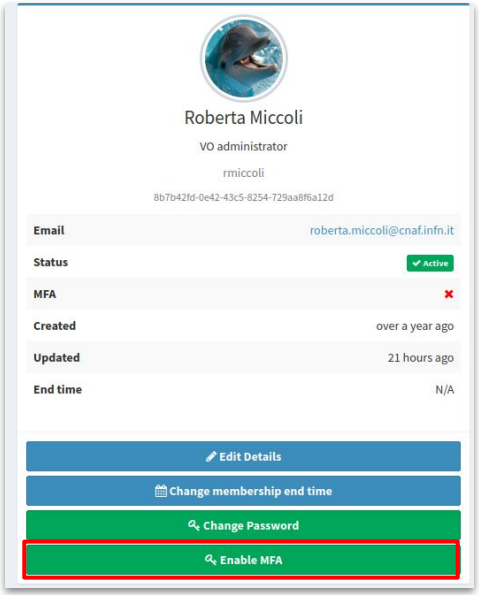
4. **IAM responds to Client**
   - IAM sends the final authN response back to the Client, which contains the MFA signal (`acr` claim), indicating that MFA has been successfully handled during the authentication process

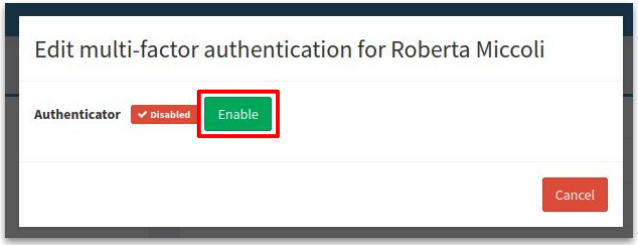
     `acr: https://refeds.org/profile/mfa`

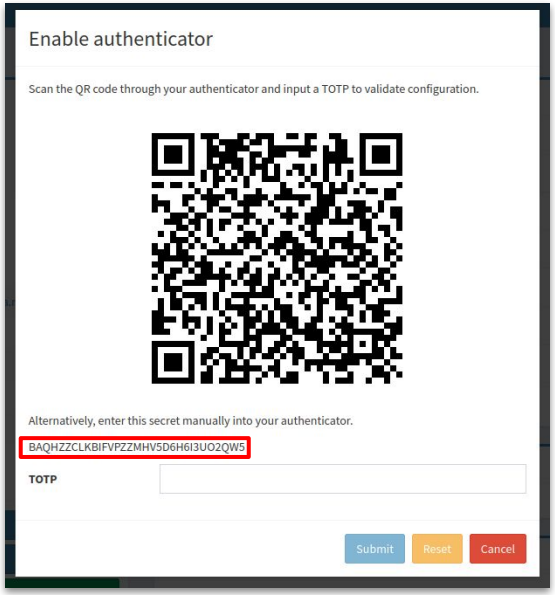


MFA demo

Enabling MFA on the [IAM dev instance]



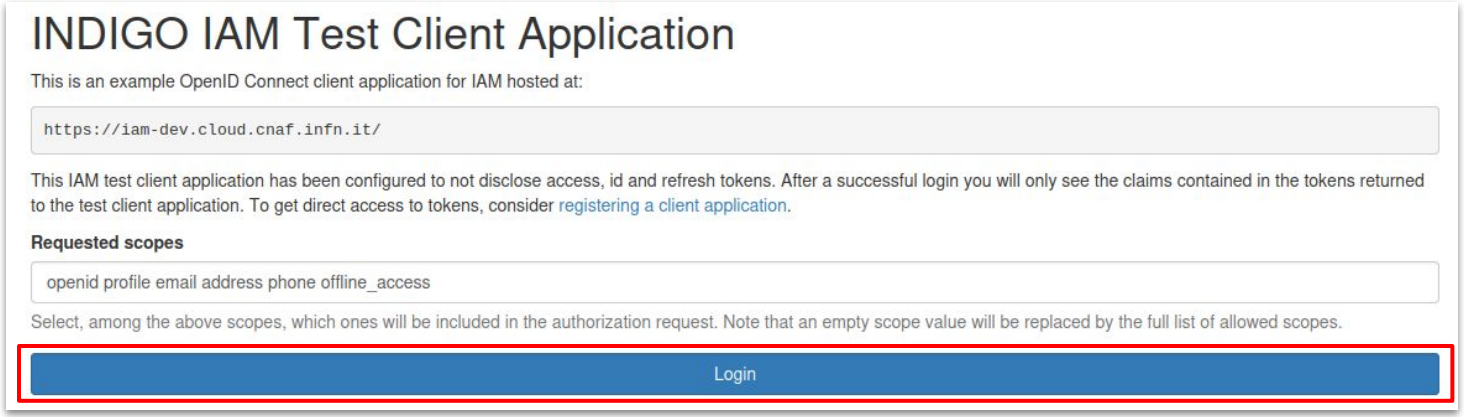




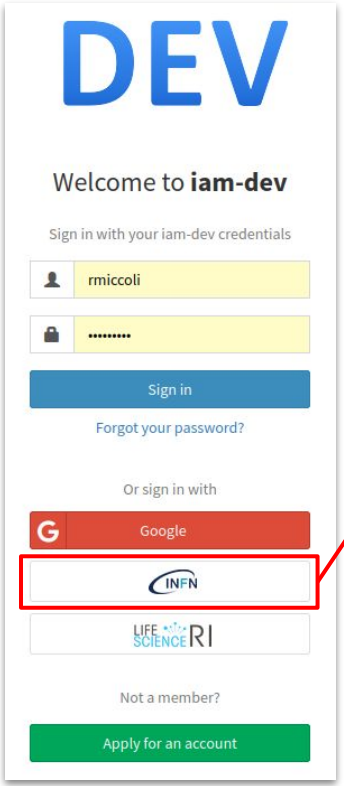
MFA demo

MFA at user's Home IdP

Using the Test Client application, which performs OAuth Authorization Code flow to get access tokens

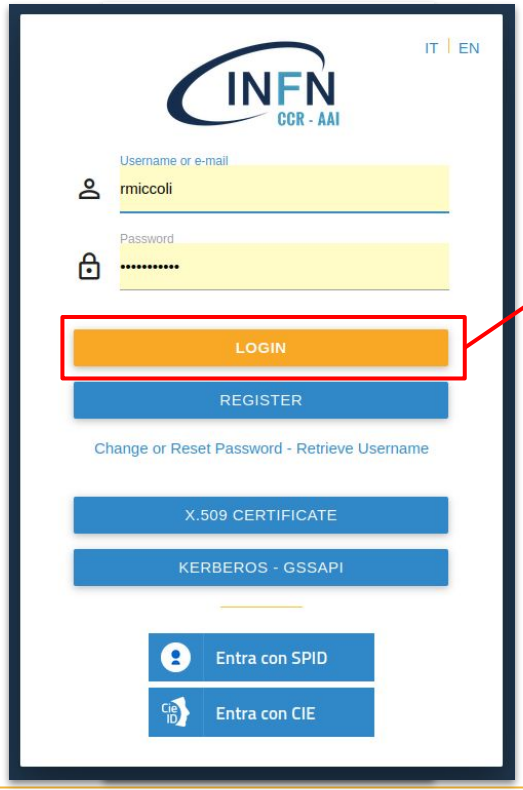


MFA demo



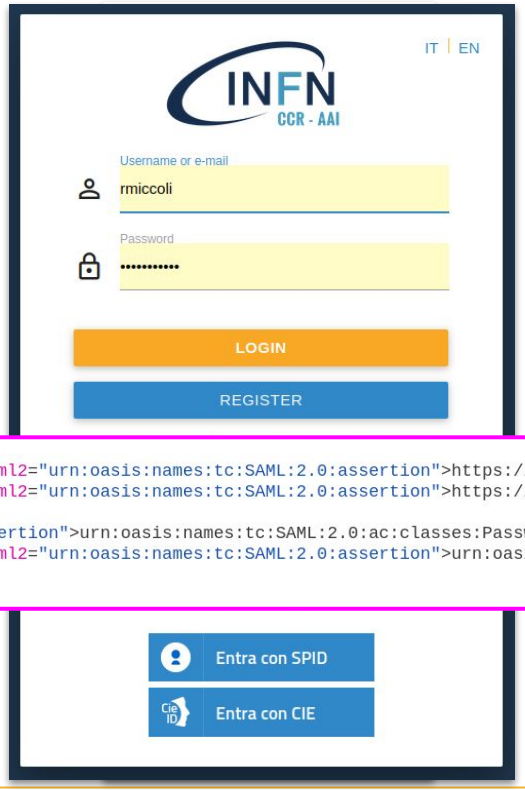
Login with INFN AAI (SAML IdP) - **2FA enabled**

MFA demo



Login with username & password (first authentication factor)

MFA demo

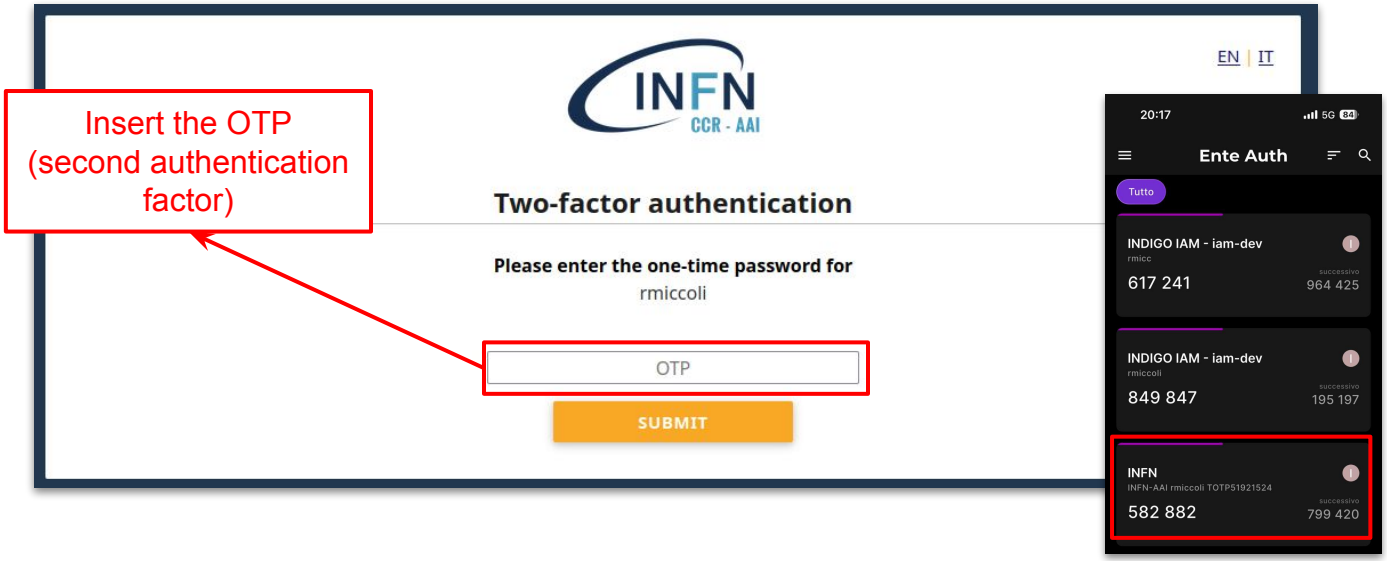


IAM *optionally* requests MFA by specifying multiple `AuthnContextClassRef` values in the SAML request

```
    <saml2p:RequestedAuthnContext>
        <saml2:AuthnContextClassRef xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">https://refeds.org/profile/mfa</saml2:AuthnContextClassRef>
        <saml2:AuthnContextClassRef xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">https://refeds.org/profile/sfa</saml2:AuthnContextClassRef>
        <saml2:AuthnContextClassRef
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml2:AuthnContextClassRef>
        <saml2:AuthnContextClassRef xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</
saml2:AuthnContextClassRef>
    </saml2p:RequestedAuthnContext>
```

SAML-tracer output!

MFA demo


Insert the OTP (second authentication factor)
EN | IT
INFN
CCR - AAI
Two-factor authentication
Please enter the one-time password for
rmiccoli
OTP
SUBMIT
20:17
Ente Auth
Tutto
INDIGO IAM - iam-dev
617 241
964 425
INDIGO IAM - iam-dev
849 847
195 197
INFN
582 882
799 420

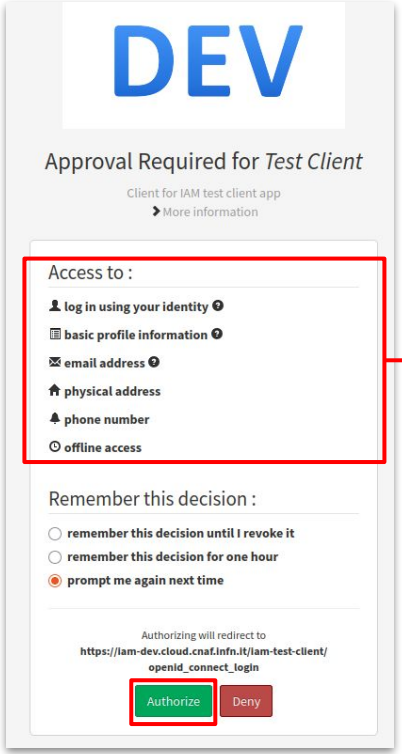
MFA demo

The INFN IdP **signals MFA** in the SAML response

```
<saml:AuthnContext>
    <saml:AuthnContextClassRef>https://refeds.org/profile/mfa</saml:AuthnContextClassRef>
</saml:AuthnContext>
```

SAML-tracer output!

MFA demo



Consent page:
the user is asked to authorize the Client to access this information!

MFA demo

INDIGO IAM Test Client Application

You're now logged in as: Roberta Miccoli

This application has received the following information:

- access_token (claims):


```
{
  "sub": "8b7b42fd-0e42-43c5-8254-729aa8f6a12d",
  "iss": "https://iam-dev.cloud.cnaf.infn.it/",
  "preferred_username": "rmiccoli",
  "client_id": "42999a63-7449-43fb-952e-42f2d75b865b",
  "wlcg.ver": "1.0",
  "acr": "https://refeds.org/profile/mfa",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "nbf": 1748123629,
  "scope": "address phone openid profile offline_access email",
  "name": "Roberta Miccoli",
  "exp": 1748127229,
  "iat": 1748123629,
  "jti": "9ada8420-d948-4875-b9cf-806513035686",
  "email": "roberta.miccoli@cnaf.infn.it"
}
```


- OAuth2 token introspection endpoint response (invoked on access_token, authorized by client credentials):


```
{
  "active": true,
  "scope": "address phone openid profile offline_access email",
  "expires_at": "2025-05-25T00:53:50+0200",
  "exp": 1748127230,
  "sub": "8b7b42fd-0e42-43c5-8254-729aa8f6a12d",
  "user_id": "rmiccoli",
  "client_id": "42999a63-7449-43fb-952e-42f2d75b865b",
  "token_type": "Bearer",
  "iss": "https://iam-dev.cloud.cnaf.infn.it/",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "acr": "https://refeds.org/profile/mfa"
}
```

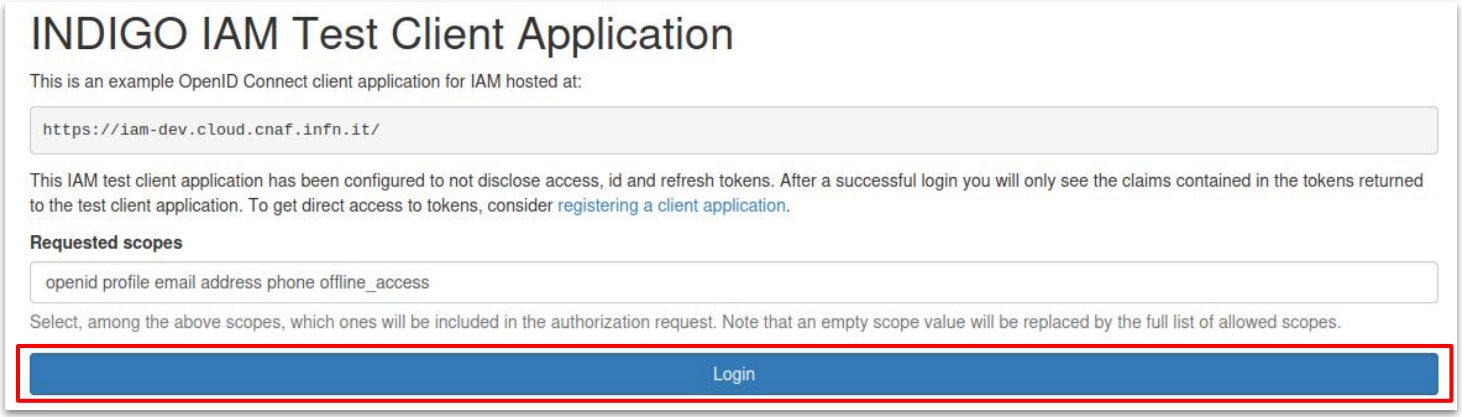

- id_token (claims):


```
{
  "sub": "8b7b42fd-0e42-43c5-8254-729aa8f6a12d",
  "amr": [
    "pwd",
    "otp"
  ],
  "kid": "rsa1",
  "iss": "https://iam-dev.cloud.cnaf.infn.it/",
  "preferred_username": "rmiccoli",
  "organisation_name": "iam-dev",
  "nonce": "39965e5e4d5ef",
  "wlcg.ver": "1.0",
  "aud": "42999a63-7449-43fb-952e-42f2d75b865b",
  "acr": "https://refeds.org/profile/mfa",
  "name": "Roberta Miccoli",
  "exp": 1748124229,
  "iat": 1748123629,
  "jti": "3dac647a-d197-4169-bee4-3cf5924dffdb",
  "email": "roberta.miccoli@cnaf.infn.it"
}
```

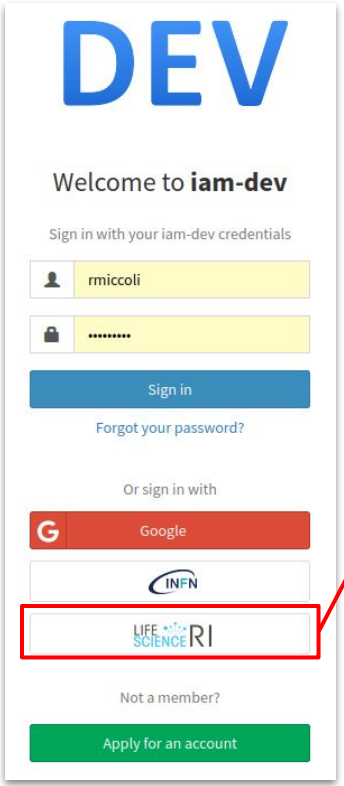

IAM **skips the MFA** but **signals it** by including the `acr` value received from the INFN IdP in the access token, ID token and introspection response

MFA demo

MFA at Proxy (IAM)

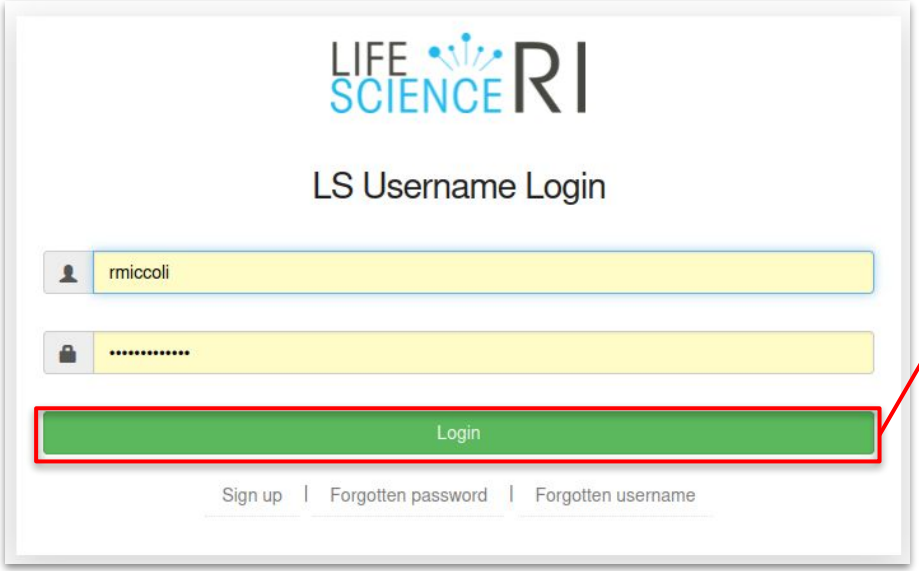


MFA demo



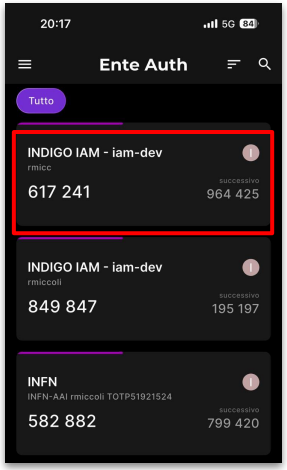
Login with LS AAI (SAML IdP) - **2FA disabled**

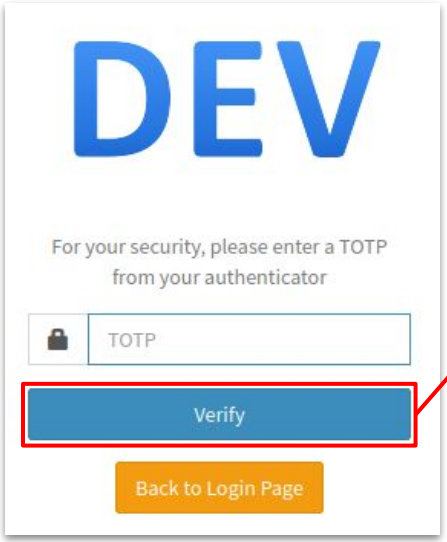
MFA demo



Login with username & password (first authentication factor)

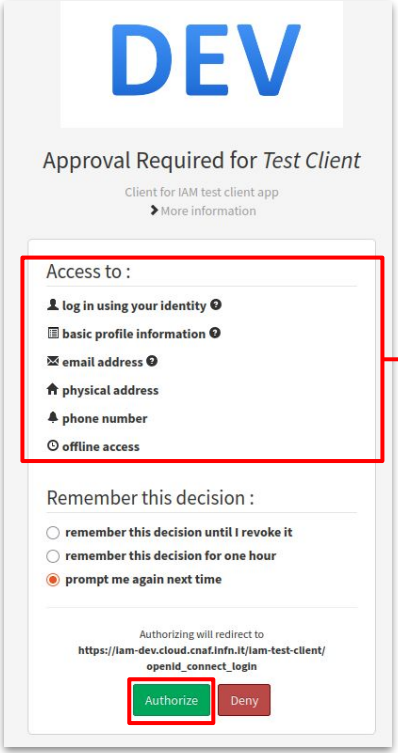
MFA demo





Insert the OTP (second authentication factor)

MFA demo



Consent page:
the user is asked to authorize the Client to access this information!

MFA demo

INDIGO IAM Test Client Application

You're now logged in as: Roberta Miccoli

This application has received the following information:

- access_token (claims):


```
{
  "sub": "760eaa3c-0537-4ca9-8939-e4e7123db868",
  "iss": "https://iam-dev.cloud.cnaf.infn.it/",
  "preferred_username": "rmicc",
  "client_id": "42999a63-7449-43fb-952e-42f2d75b865b",
  "wlcg.ver": "1.0",
  "acr": "https://refeds.org/profile/mfa",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "nbf": 1748123781,
  "scope": "address phone openid profile offline_access email",
  "name": "Roberta Miccoli",
  "exp": 1748127381,
  "iat": 1748123781,
  "jti": "877561f8-bd49-41b0-bb27-abfc8451abb8",
  "email": "rmiccoli@infn.it"
}
```


- OAuth2 token introspection endpoint response (invoked on access_token, authorized by client credentials):


```
{
  "active": true,
  "scope": "address phone openid profile offline_access email",
  "expires_at": "2025-05-25T00:56:22+0200",
  "exp": 1748127382,
  "sub": "760eaa3c-0537-4ca9-8939-e4e7123db868",
  "user_id": "rmicc",
  "client_id": "42999a63-7449-43fb-952e-42f2d75b865b",
  "token_type": "Bearer",
  "iss": "https://iam-dev.cloud.cnaf.infn.it/",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "acr": "https://refeds.org/profile/mfa"
}
```

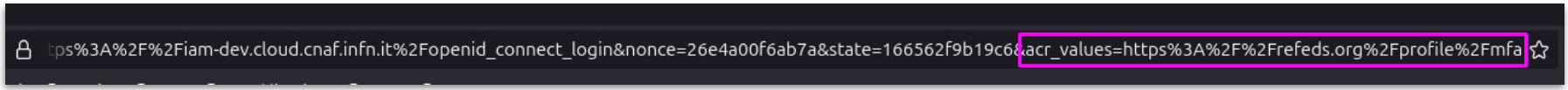

- id_token (claims):


```
{
  "sub": "760eaa3c-0537-4ca9-8939-e4e7123db868",
  "amr": [
    "otp",
    "saml"
  ],
  "kid": "rsa1",
  "iss": "https://iam-dev.cloud.cnaf.infn.it/",
  "preferred_username": "rmicc",
  "organisation_name": "iam-dev",
  "nonce": "290d58bcefcab",
  "wlcg.ver": "1.0",
  "aud": "42999a63-7449-43fb-952e-42f2d75b865b",
  "acr": "https://refeds.org/profile/mfa",
  "name": "Roberta Miccoli",
  "exp": 1748124381,
  "iat": 1748123781,
  "jti": "48ec2a16-986a-4f84-aa87-9f0cff14c6d1",
  "email": "rmiccoli@infn.it"
}
```


IAM **signals MFA** by including the `acr` claim in the access token, ID token and introspection response

MFA demo

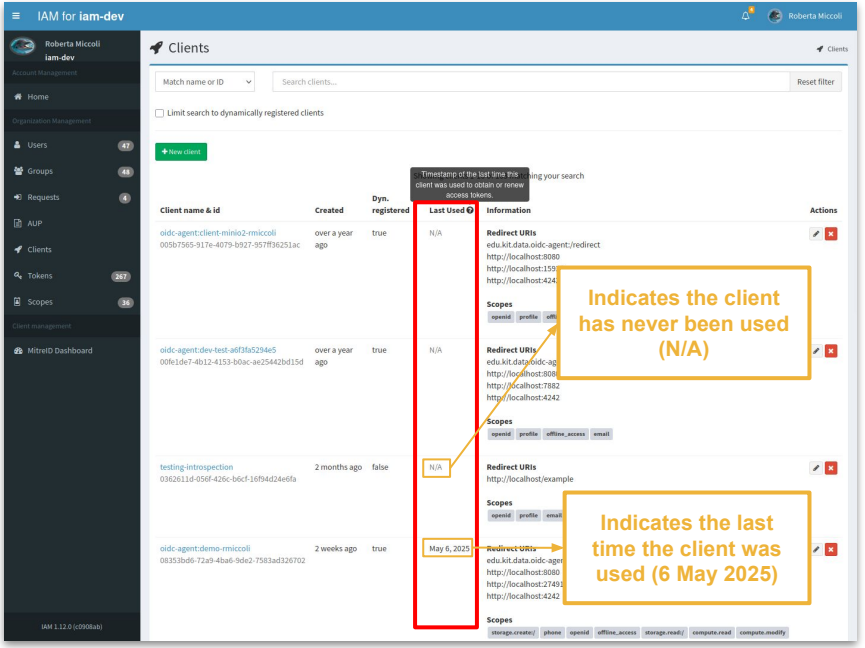
The behavior with OIDC providers is the same; the only difference is how IAM requests MFA from the OIDC provider during the OIDC authentication request



Clients cleanup

How to remove obsolete clients

- Problem: (in DataCloud) accumulation of unused clients can compromise security and efficiency
- Solution: keep track of the last time (*date only, no timestamp*) a client has been used, i.e.
  - the last time the **client was used to obtain or renew access tokens**
- Implementation: a `last_used` column has been added to the database clients table (`client_details`), and the API and dashboard have been updated accordingly
  - this behaviour is enabled by configuration by setting `IAM_CLIENT_TRACK_LAST_USED = true` (default is `false`)



How to remove obsolete clients

- By exploiting the tracking of the last use of the existing client, external tools such as **scripts** that use the **API** to remove obsolete clients can be realised
  - these scripts can only be executed by IAM administrators, since access to the API is only authorised to clients presenting an access token that contains admin scopes (i.e. `iam:admin.read` & `iam:admin.write`)
- Here an example of Python script which
  - queries an IAM API endpoint (`/iam/api/clients`) to retrieve a list of registered clients using paginated HTTP requests, authenticated via an access token with admin scopes
  - filters the clients based on a user-specified date
    - **dynamically registered clients** are selected if they have **never been used** and were **created before** the given date
    - **statically registered clients** are selected if they were **last used before** the given date

How to remove obsolete clients

```
$ python3 mastrolindo.py  https://iam-dev.cloud.cnaf.infn.it 2022-03-12

# client_id                   005b7565-917e-4079-b927-957ff36251ac
# client_name                 oidc-agent:client-minio2-rmiccoli
# created_at                  2020-02-12 12:30:19
# dynamically_registered      True
# last_used                   2021-05-28 11:01:20
https://iam-dev.cloud.cnaf.infn.it/iam/api/clients/005b7565-917e-4079-b927-957ff36251ac

# client_id                   f23fb711-c7ff-4cce-bcf0-1fc73a7de4d8
# client_name                 oidc-agent:dev-2feee830c1a4
# created_at                  2022-03-09 18:18:44
# dynamically_registered      True
# last_used                   None
https://iam-dev.cloud.cnaf.infn.it/iam/api/clients/f23fb711-c7ff-4cce-bcf0-1fc73a7de4d8
```

The output of the Python script run against IAM dev instance indicates that two clients should be deleted!

To effectively delete clients from INDIGO IAM, run

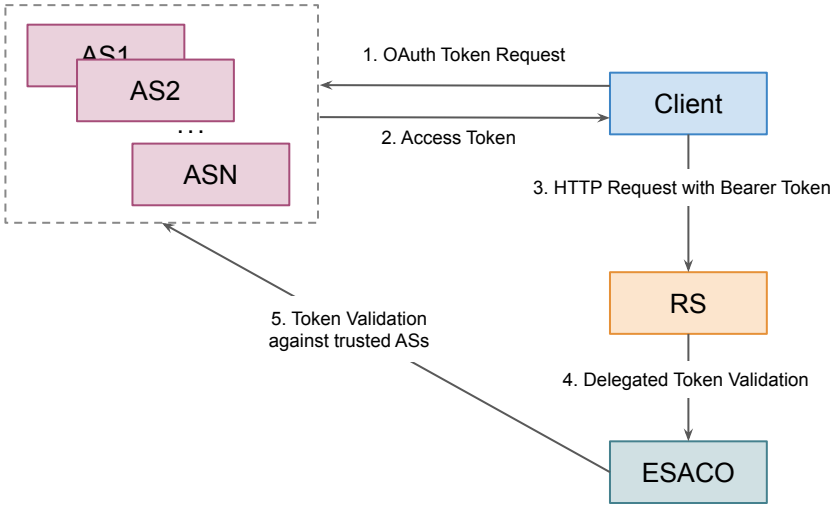
```
$ python3 mastrolindo.py https://iam-dev.cloud.cnaf.infn.it 2022-03-12 > output
$ cat output | grep ^https | xargs  curl -X DELETE -H "Authorization: Bearer $BEARER_TOKEN"
```

Proxied token introspection

Proxied token introspection

- The possible adoption of INDIGO IAM in the Italian EOSC node requires the development of the ***proxied token introspection*** (AARC-G052 guideline)
  - an extension of OAuth2 that delegates to a local infrastructure proxy the introspection requests to remote authorization servers
  - one of the building blocks needed for OpenID Federation
- We could start from ESACO, a software developed by INFN CNAF
  - it is a daemon that has the responsibility of checking validity and signatures of OAuth tokens for registered trusted OAuth authorization servers
  - it exposes an OAuth token introspection endpoint compliant with RFC 7662 that can be used by authenticated clients to inspect access tokens

ESACO



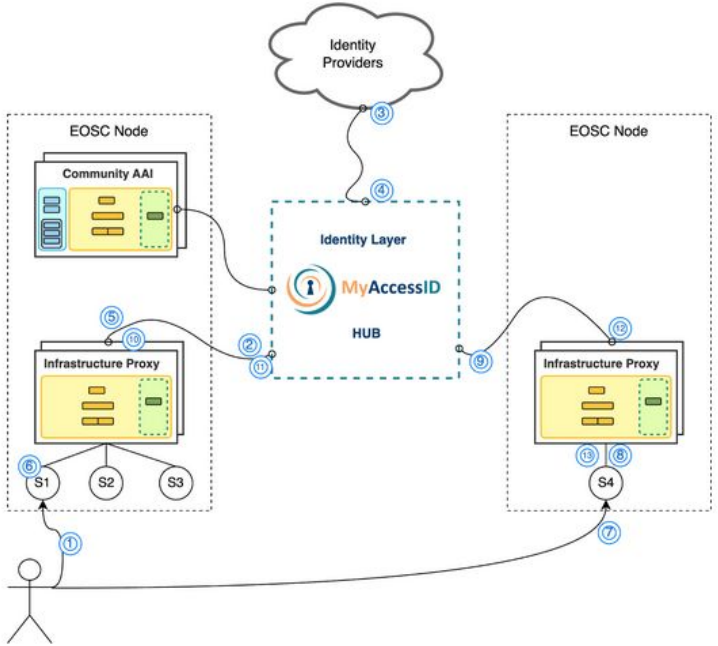
ESACO is registered as a client at one or more trusted OAuth Authorization Servers, and is used by Resource Servers (e.g., StoRM) as a gateway for token validation and introspection

ESACO performs local JWT validation checks and leverages the introspection endpoint at trusted ASs to inspect a submitted access token

Proxied token introspection with INDIGO IAM

ESACO could be extended and integrated into INDIGO IAM to enable *proxied token introspection*, supporting its role as a participating node in the EOSC AAI Federation

INDIGO IAM can act both as a Community AAI and an Infrastructure Proxy



Thanks for your attention!

Useful references

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

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

For general information:

- OAuth 2.0: https://oauth.net/2/ and OAuth 2.1: https://oauth.net/2.1/
- OpenID Connect: https://openid.net/connect/

Contacts:

- iam-support@lists.infn.it

Backup

INDIGO IAM in the computing federation

ICSC/TeRABIT

Role

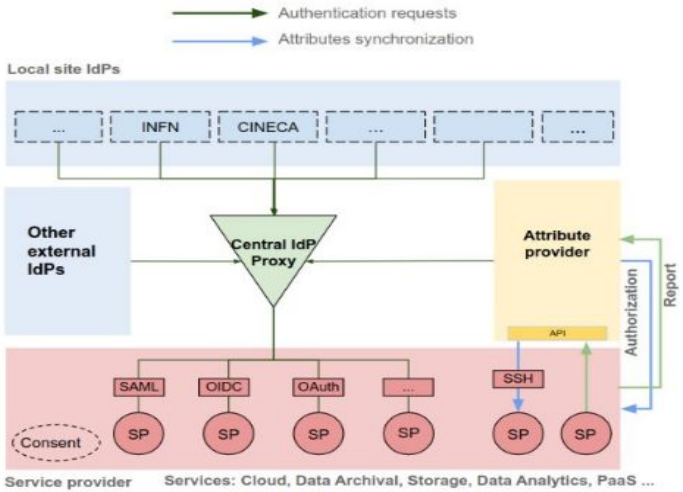
- Entrypoint for the computing federation, IAM acts as Attribute Authority and IdP proxy for the whole infrastructure

Objective

- Federate INFN and CINECA IdPs
- CINECA and INFN users can register into INDIGO IAM through their IdP and be authorized to access the resources of both institutions

Advantages

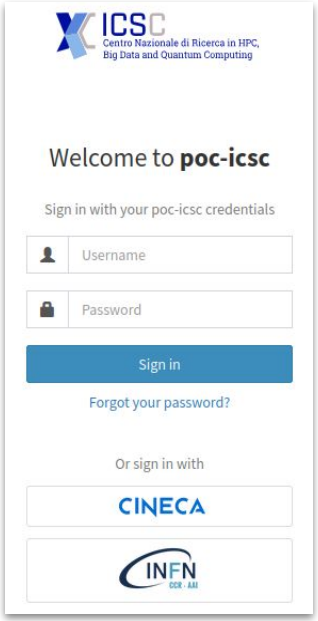
- In-house development (mainly by INFN CNAF), born to satisfy the needs of scientific community
- Easy integration with third-party applications
- Backward-compatible with Grid-based authorization
- Support for capability-based authorization
- Allows the definition of policies for fine-tuned access privileges



PoC IAM: technologies

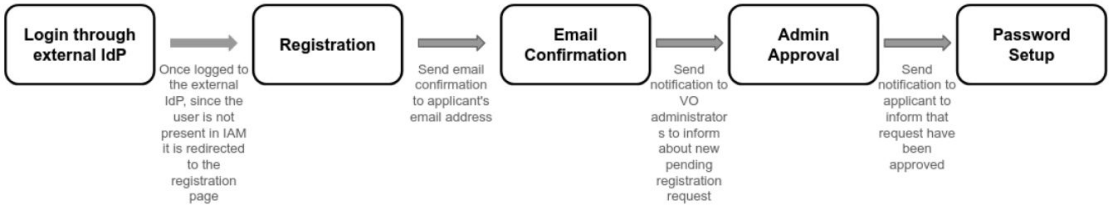
- The PoC IAM instance is deployed using **Docker Compose** on a Virtual Machine within the **INFN Cloud** infrastructure
  - deployed behind an **NGINX**
  - stores data in a **MySQL** database
- All the services belonging to the PoC infrastructure (the INDIGO PaaS orchestrator, RUCIO, etc.) that support authentication with the PoC IAM have been integrated by registering them as **clients** in IAM, exploiting the OpenID Connect technology

PoC IAM: state of the art



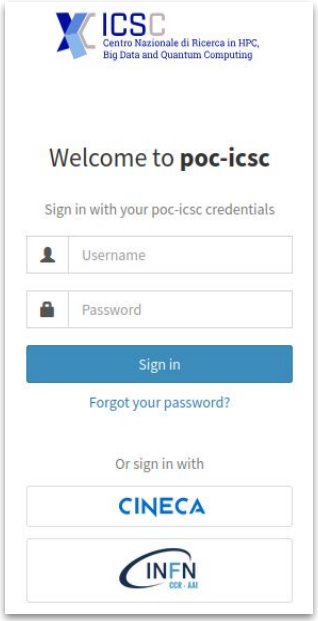
https://iam-poc-icsc.cloud.infn.it/

- Defined a Virtual Organization (VO), called *poc-icsc*
- Moderated user enrollment
  - it requires manual approval by IAM admins



- Authentication methods
  - external IdPs: **CINECA** dev instance of keycloak (*OIDC*) and **INFN AAI** (*SAML*)
  - X.509 certificates (if linked to the account)

PoC IAM: state of the art

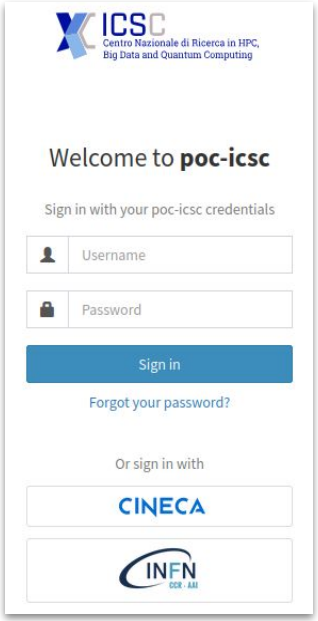


https://iam-poc-icsc.cloud.infn.it/

Attribute-based authorization

- Defined a set of **IAM groups** to enforce a more controlled access to federated resources
  - *poc-icsc/prod*: optional group (or VOMS role), necessary to submit third-party transfer jobs to FTS in the infrastructure, authenticating and authorizing with a proxy
  - *poc-icsc/admins/poc-icsc*: mapped to an OpenStack project on the federated Cloud providers used to instantiate services on the public network
  - *poc-icsc/priv-admins/poc-icsc*: not yet defined, will be mapped to an OpenStack project on the federated Cloud providers used to instantiate services on a private network

PoC IAM: state of the art

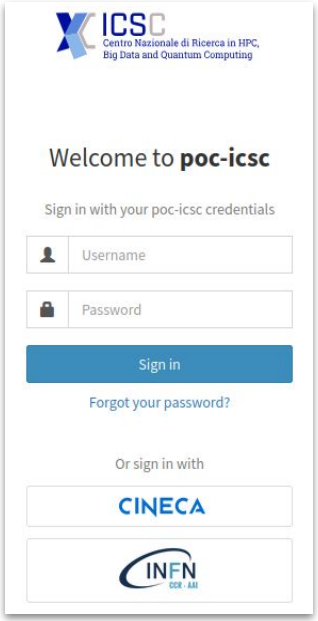


https://iam-poc-icsc.cloud.infn.it/

Scope-based authorization

- Defined **IAM scope policies** applied to each storage system for finer-grained read/write permissions in the federated namespace
  - read access to the entire namespace (`/`) is granted to users of the *poc-icsc* group
  - write access to the `/user/<iam-username>` namespace is granted to the user `<iam-username>`

PoC IAM: state of the art



https://iam-poc-icsc.cloud.infn.it/

The PoC IAM instance has been successfully integrated with

- PaaS orchestrator and its dashboard
- OAuth2 proxy services used by interLink
- RUCIO + FTS
- INFN Storage systems and INFN federated Cloud services involved in the PoC

Current constraints

Identity Federation

- Due to existing certification policies and processes, a federated user must already be registered as a user at CINECA to access CINECA resources
- As a workaround for the PoC, all federated users must first be statically registered in CINECA LDAP and then registered in the PoC IAM instance using the same username from CINECA LDAP
- On the other hand, any federated CINECA user can transparently access both CINECA and INFN resources
- Ongoing discussions in ICSC spoke 0 and TeRABIT

Integration with CINECA

- Limited to a few processes and test endpoints (i.e., offloading part of the workflow of a Cloud application to HPC (CINECA) resources and data transfer between a test CINECA S3 endpoint and INFN resources)