INDIGO IAM

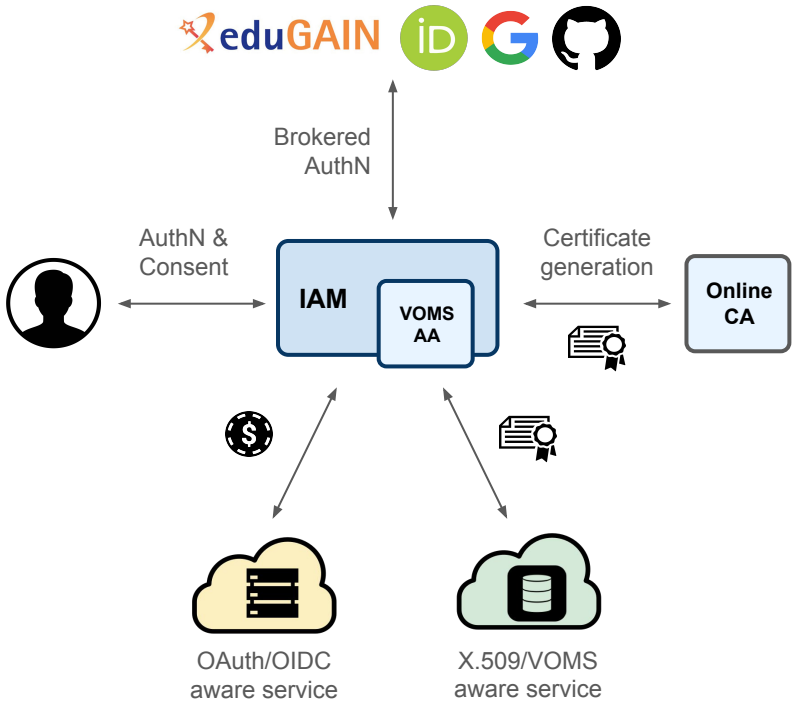
Corso di formazione “Panoramica su OAuth2/OpenID Connect e sue applicazioni tramite il servizio INDIGO IAM”, 12-14 Maggio 2025, LNF

Roberta Miccoli, INFN CNAF

Introduction

INDIGO Identity and Access Management Service

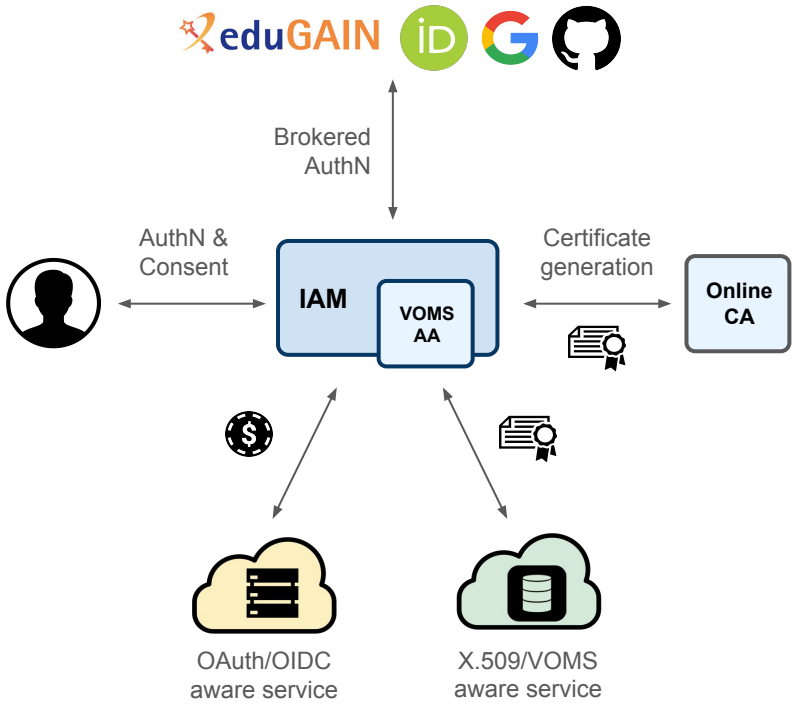
- Standard **OAuth2 Authorization Service** and **OpenID Connect Provider**
  - based on the MITREid Connect library*
- Easy integration with (web) applications
- Integration with command-line tools
  - oidc-agent, curl, gfal2
- Support for **multiple profiles**, i.e. how content is expressed in JWTs
  - IAM, WLCG, AARC, Keycloak
- Rich set of **APIs**



*no longer actively maintained;
extended and maintained in IAM for research infrastructure needs

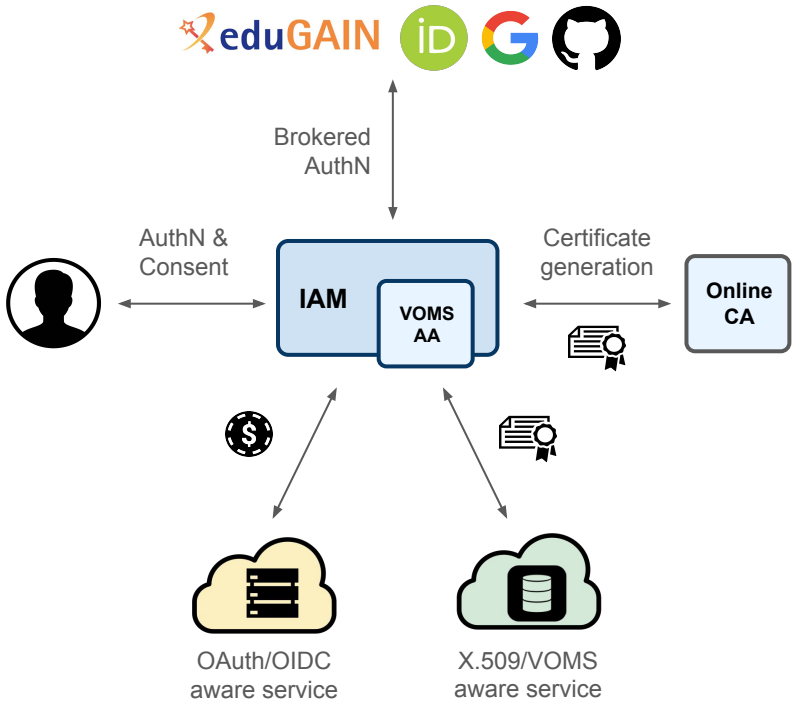
INDIGO Identity and Access Management Service

- Design choices rely on 20+ years of experience with **VOMS** (Virtual Organization Membership Service)
  - The service used on the Grid to manage a VO
  - Based on X.509 certificates, proxies and attribute certificates
- **Multiple authentication mechanisms**
  - SAML, OpenID Connect, X.509, username/password
- **Multi-Factor Authentication** (MFA)
  - *experimental* feature, available for local credentials
  - *soon* also for SAML, OIDC external providers and X.509



INDIGO Identity and Access Management Service

- **Account linking**
- Moderated and automatic user enrollment
- Enforcement of **AUP acceptance**
- VO membership management
  - Groups, roles, attributes
- Smooth transition from a VOMS-based infrastructure
- Issuance of **JWT** tokens and VOMS attribute certificates with **identity** and **membership information**, **attributes** and **capabilities**

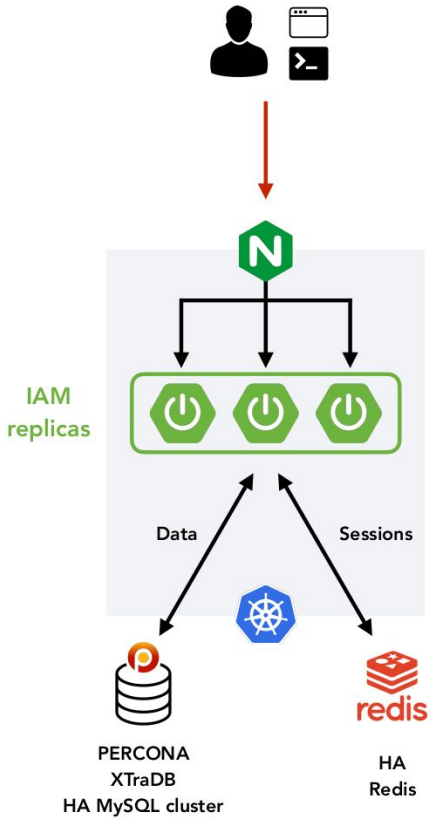


Synergies with other projects

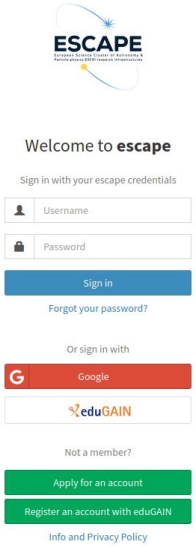
- First developed in the context of the H2020 INDIGO DataCloud project
- **Selected by the WLCG management board** to be the core of the future, token-based WLCG AAI
- INFN commitment for the foreseeable future, with additional support of several **Italian and European projects**

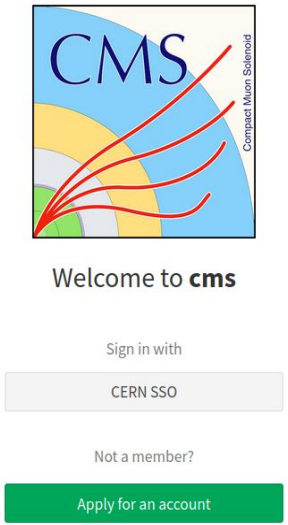
Development and deployment

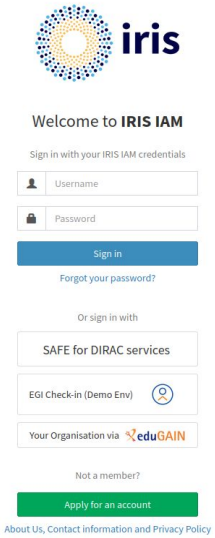
- **Java** application based on the **Spring Boot** framework
  - MITREid library to be replaced by Spring Authorization Server
- Typically deployed as a **Docker container** in K8S
- Load tests have shown that a single IAM instance can issue tokens up to a few hundred Hz, with a latency of about 1s
  - Not enough for some scenarios
  - Work ongoing to improve scalability
- Deployment in **HA** is possible

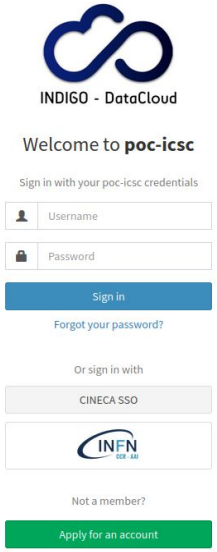


INDIGO IAM deployments









~ 20 instances inside CNAF for internal purposes (INFN Cloud, CNAF Cloud, INFN T1 services, etc.) and support collaborations (ILDG, Belle-II, HERD, JUNO, etc.)

4 instance at CERN for LHCb, ATLAS, CMS and ALICE experiments and other instances for VOs management (e.g. dteam)

1 instance at STFC for IRIS project

3 instances at IN2P3 for MesoNET, EURO-LABS, GRANDMA projects

Development roadmap

- **Security**
  - Full support of MFA
- **Superseded obsolete dependencies**
  - MITREid → Spring Authorization Server
  - AngularJS → React JS
- **Interoperability focus**
  - Support OIDC Federation
  - Improve compliance 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)

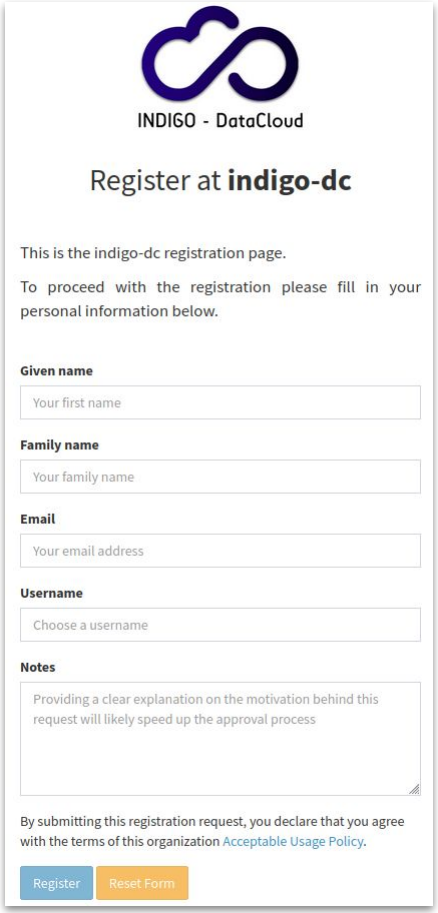
Latest release IAM v1.11.0 - released on 2024-12-19

Key features

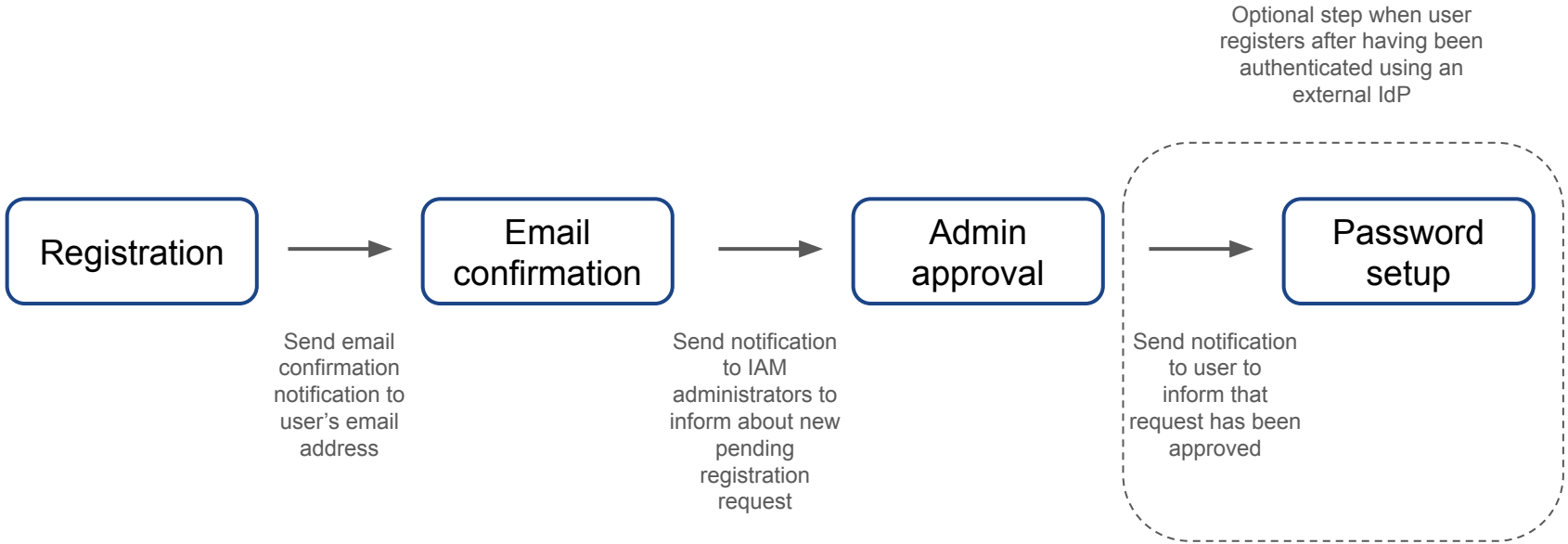
User enrolment & registration service

- **Admin-moderated** flow
  - The user fills basic registration information, accepts AUP, proves email ownership
  - IAM administrators are informed by email and can approve or reject incoming membership requests
  - The user 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

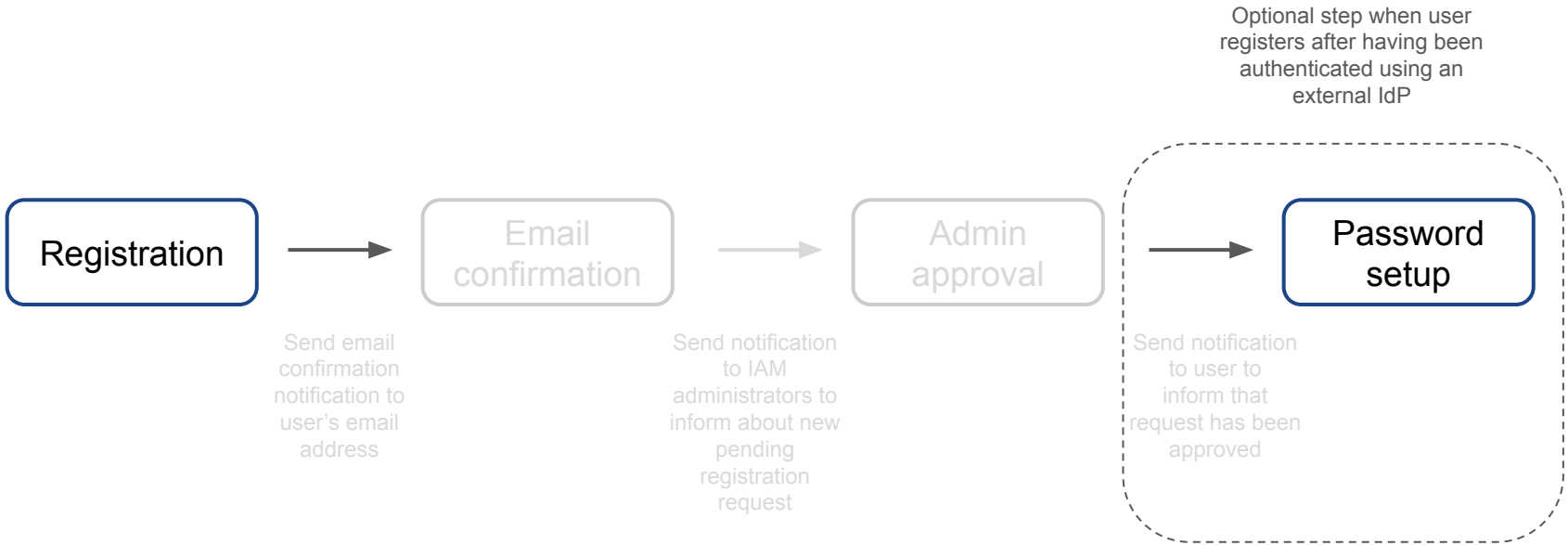
*It will also work for trusted OIDC IdPs by the next release



Moderated enrolment flow

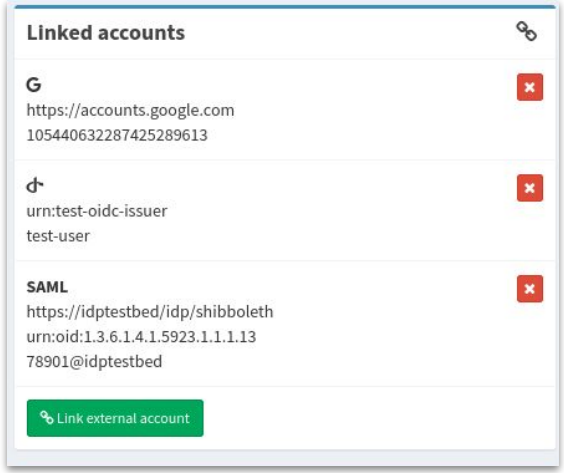


Automatic enrolment flow



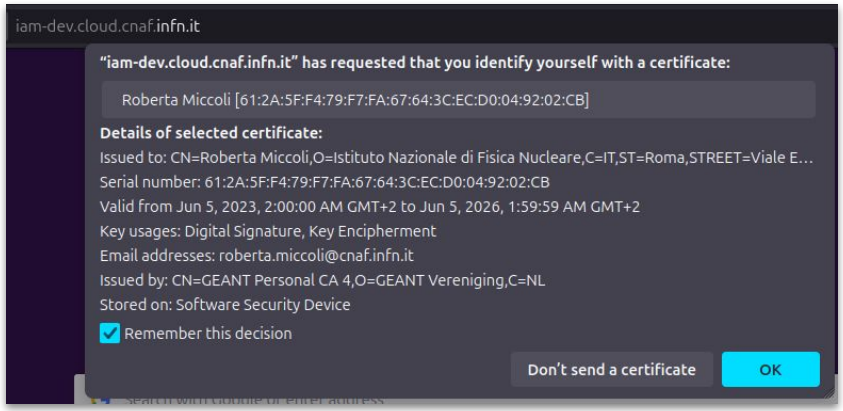
Flexible authentication & account linking

- Authentication supported via
  - local **username/password** (created at registration time)
  - **SAML** Home Institution IdP (e.g., eduGAIN)
  - **OpenID Connect** IdP (e.g., Google, 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 they own such account



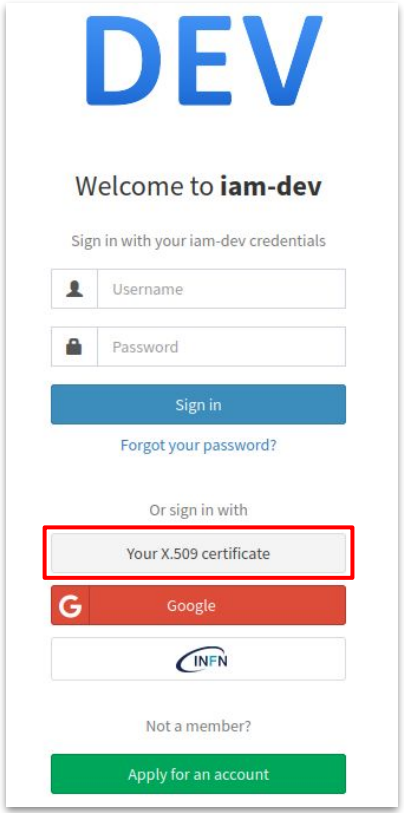
X.509 certificate linking

- The X.509 certificates must be correctly installed in the browser
- The X.509 certificates must be trusted by IAM (i.e. signed by a Certificate Authority known and trusted by IAM)
- At the IAM login page, a pop-up is prompted asking which X.509 certificate you wish to use



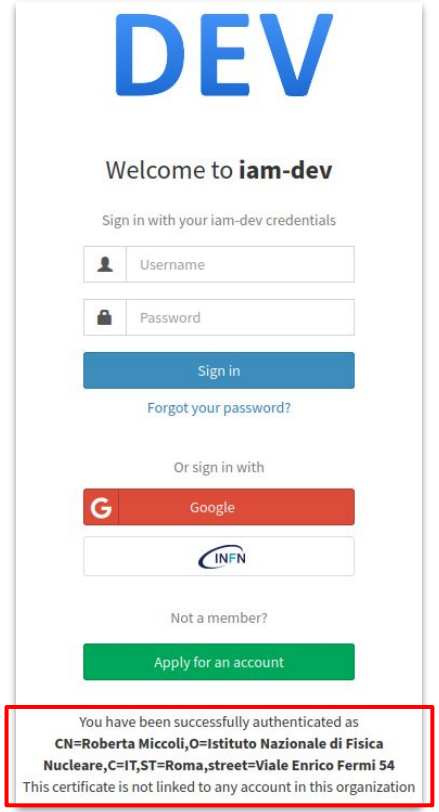
X.509 certificate linking

If the X.509 certificate used for the authentication is already linked to an account, the IAM login page will display a "*Sign in with your X.509 certificate*" button



X.509 certificate linking

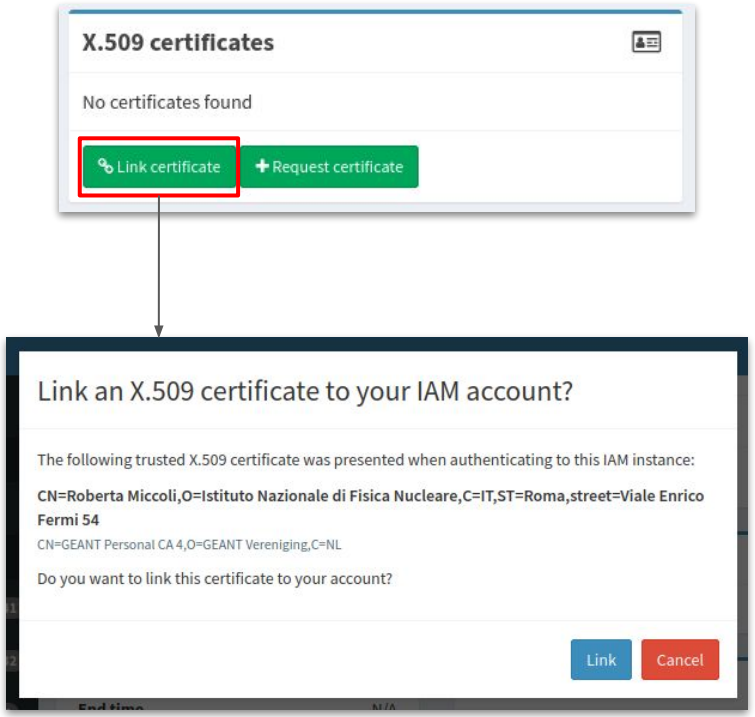
If the X.509 certificate was recognized but is not linked to any account in IAM, the user will have to authenticate with other credentials (local username/password or external authentication mechanisms)



X.509 certificate linking

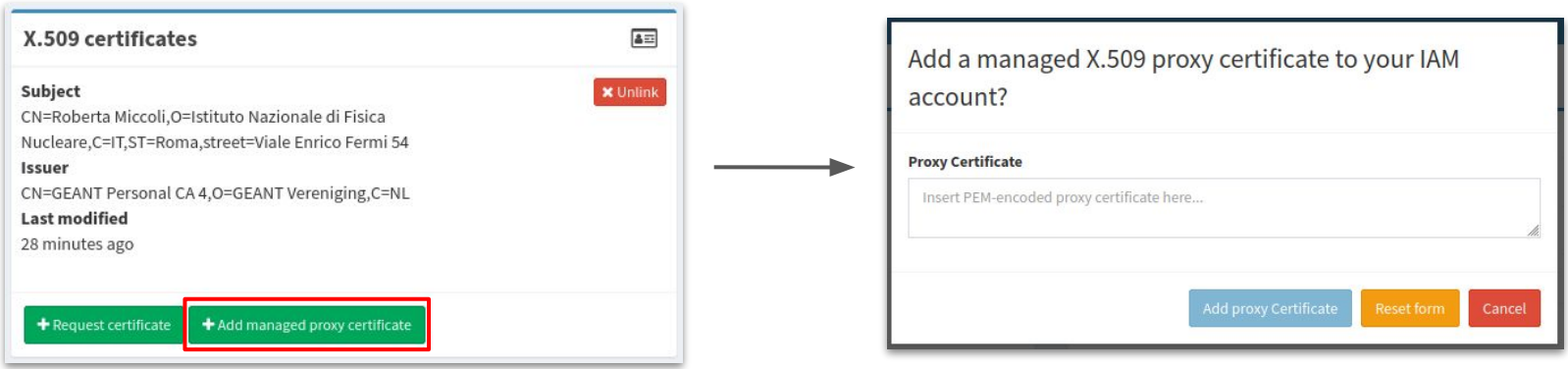
If the X.509 certificate was recognized but is not linked to any account in IAM, the user will have to authenticate with other credentials (local username/password or external authentication mechanisms)

After having logged in, the user can link the X.509 certificate by clicking on the "*Link Certificate*" button



X.509 proxy certificate

An **X.509 plain proxy** (i.e. without VOMS extensions), previously obtained from a linked certificate in IAM by running the `voms-proxy-init*` command, can be upload on IAM




```
*$ voms-proxy-init -cert <path-to-certificate> -key <path-to-key> -valid <hours:minutes>
```

X.509 proxy certificate

IAM provides a RESTful API to download the X.509 proxy certificate previously uploaded in IAM dashboard

```
$ curl -s -X POST -H "Authorization: Bearer ${AT}" -d
client_id=${CLIENT_ID} -d client_secret=${CLIENT_SECRET} -d
lifetimeSecs=${PROXY_CERT_LIFETIME_SECS}
http://localhost:8080 /iam/proxycert
{
  "subject": "CN=2050153474,CN=813462308,CN=John Doe
jdoe@infn.it,O=Istituto Nazionale di Fisica
Nucleare,C=IT,DC=tcs,DC=terena,DC=org",
  "issuer": "CN=813462308,CN=John Doe jdoe@infn.it,O=Istituto
Nazionale di Fisica Nucleare,C=IT,DC=tcs,DC=terena,DC=org",
  "identity": "CN=John Doe jdoe@infn.it,O=Istituto Nazionale
di Fisica Nucleare,C=IT,DC=tcs,DC=terena,DC=org",
  "not_after": "1663991299000",
  "certificate_chain": "-----BEGIN
CERTIFICATE-----\nMI...lCU=\n-----END CERTIFICATE-----\n"
}
```

The **`proxy:generate`** OAuth scope is required to access this API

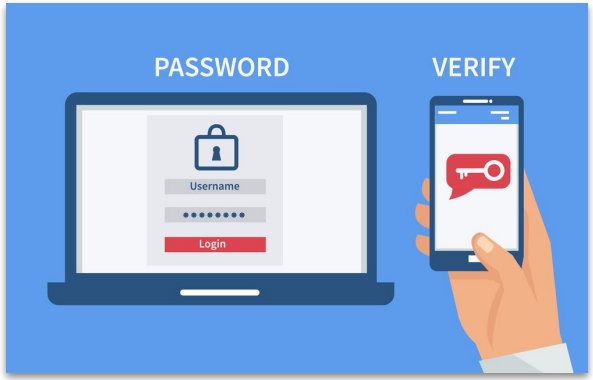
The user can request a specific lifetime of the proxy that is less than or equal to the maximum lifetime of the proxy itself

From this X.509 proxy certificate, the user can obtain a **VOMS proxy** - a proxy with VOMS extensions, i.e., VO-related attributes

```
$ voms-proxy-init --voms <voname>
```

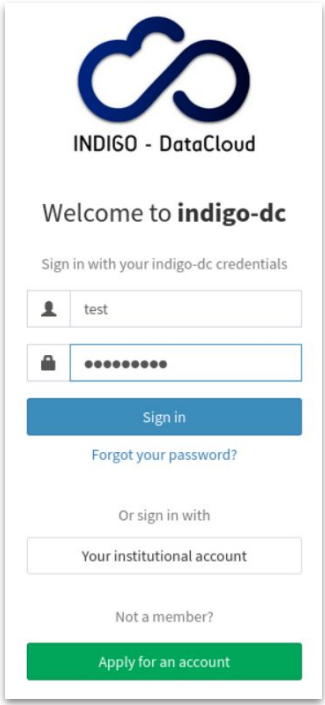
Two-Factor Authentication (2FA)

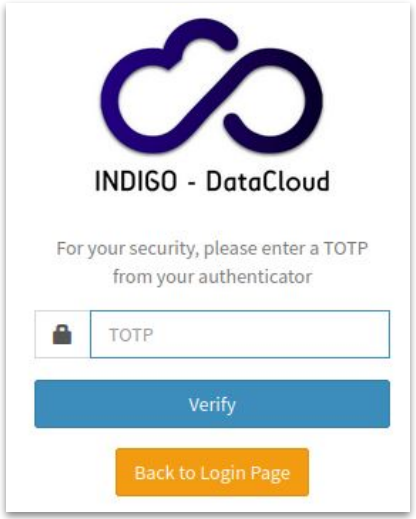
- Introduced as an *experimental* feature in the latest IAM release, v1.11.0
  - applicable only to **login with username and password**
  - integration with SAML, OIDC remote providers and X.509 certificates is ongoing
- 2FA enabled by configuration
  - `mfa` spring profile
- Multi-factor settings menu available on user dashboard
- Each user can enable/disable it for themselves; IAM admin can disable it for each user



Enabling 2FA for local credentials

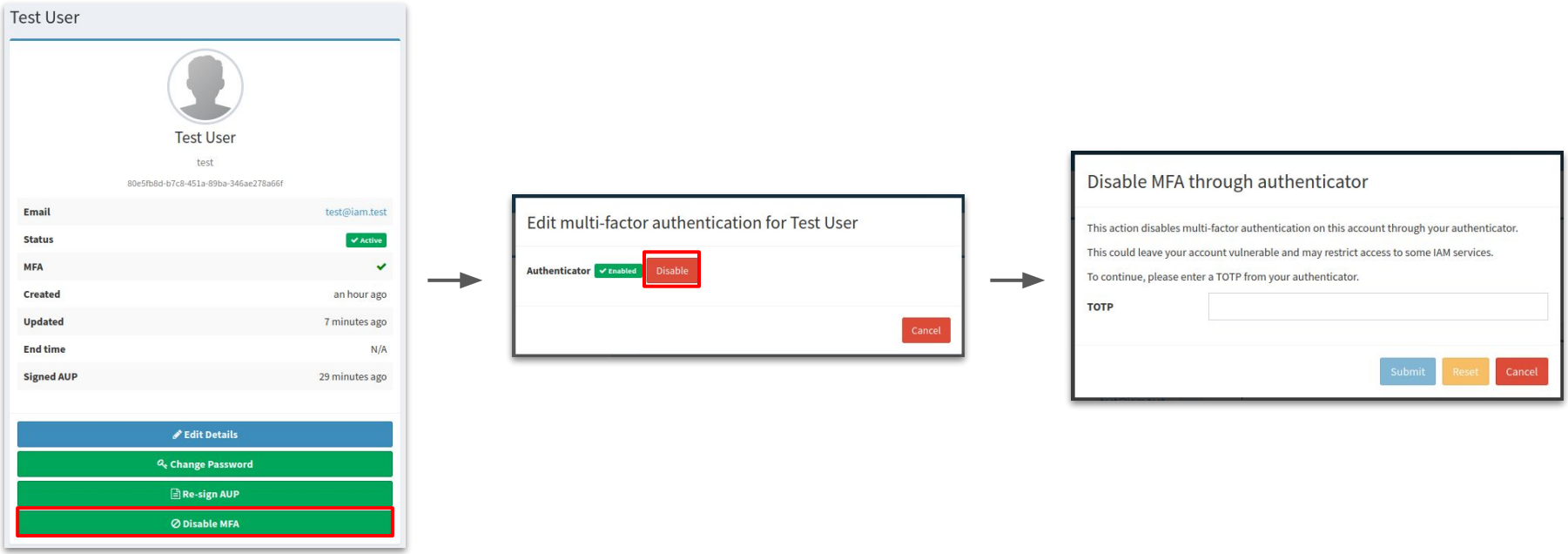
Signing in and verification





Authenticated!

Disabling 2FA



Integrating 2FA with external SAML/OIDC providers

- PR in progress/review
- Work also carried out for the EOSC Beyond project
- Based on REFEDS MFA profile and RFC 9470

Ongoing implementation

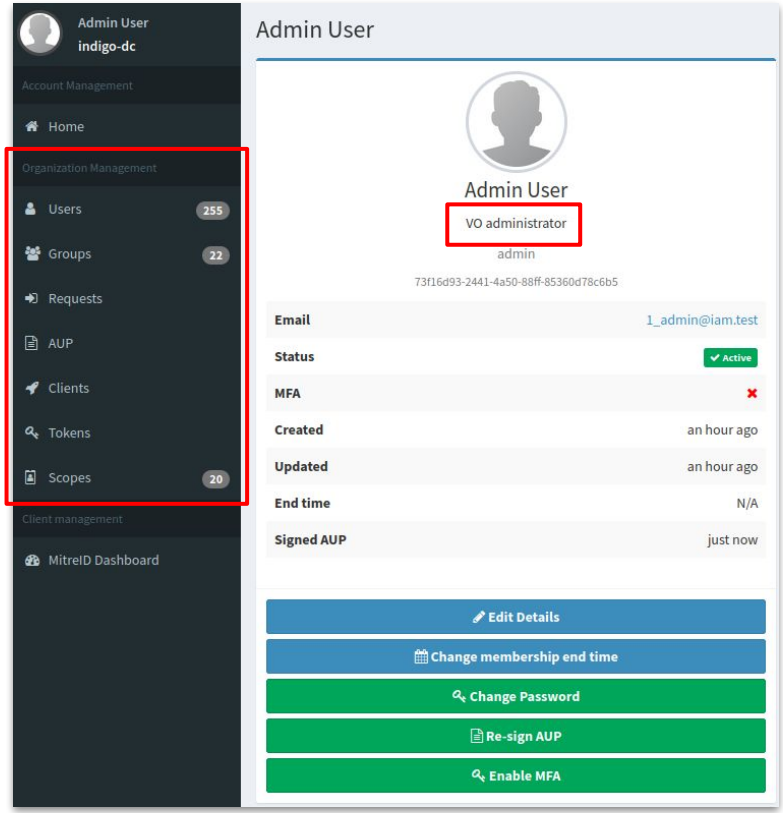
- When the user authenticates via an external OIDC/SAML provider
  - if MFA was used, IAM skips second-factor authentication, based on the information received from the remote IdP
    - **acr** claim (for OIDC) in the ID token
    - **AuthnContextClassRef** (for SAML) in the SAML assertion
  - if MFA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

Management tools

IAM provides a **mobile-friendly** dashboard for

- User management
- Group management
- Membership request management
- AUP management
- Client management
- Token management
- Scopes management
- Account linking and personal details editing

All management functionality is also exposed by **REST APIs**

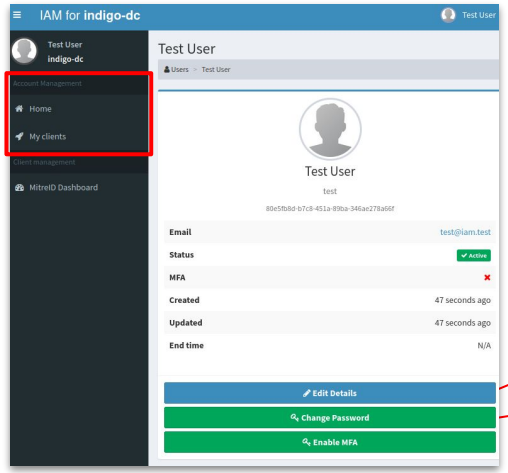


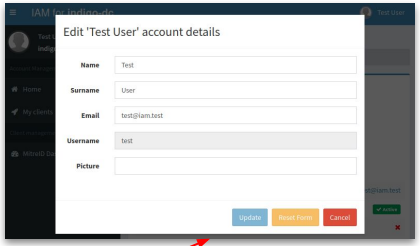
IAM roles

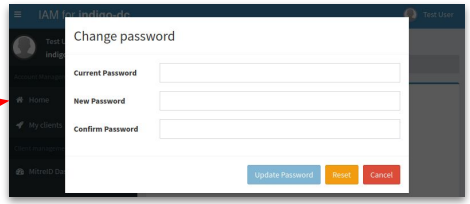
Functionality	User	Group Manager	Admin
Access to services (SSO)	✅	✅	✅
Personal profile management	✅	✅	✅
Group membership request	✅	✅	✅
Approval/rejection of account membership requests	❌	❌	✅
Approval/rejection of group membership requests	❌	✅	✅
Removal of members from groups	❌	✅	✅
Creating/deleting groups	❌	❌	✅
User management (suspension, roles)	❌	❌	✅
Full access to APIs	❌	❌	✅

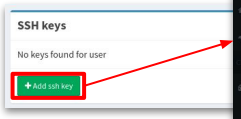
Personal profile management

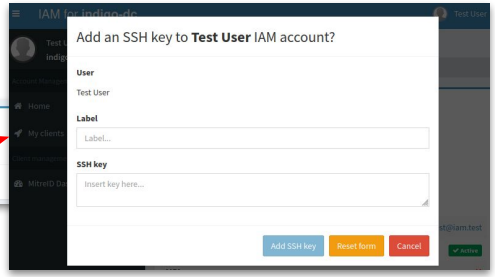
User perspective

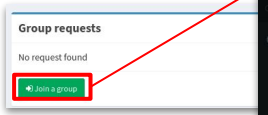


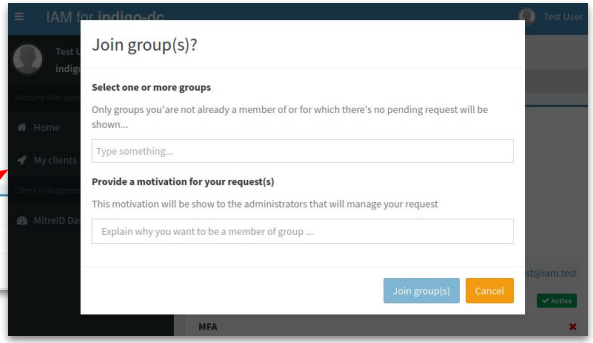






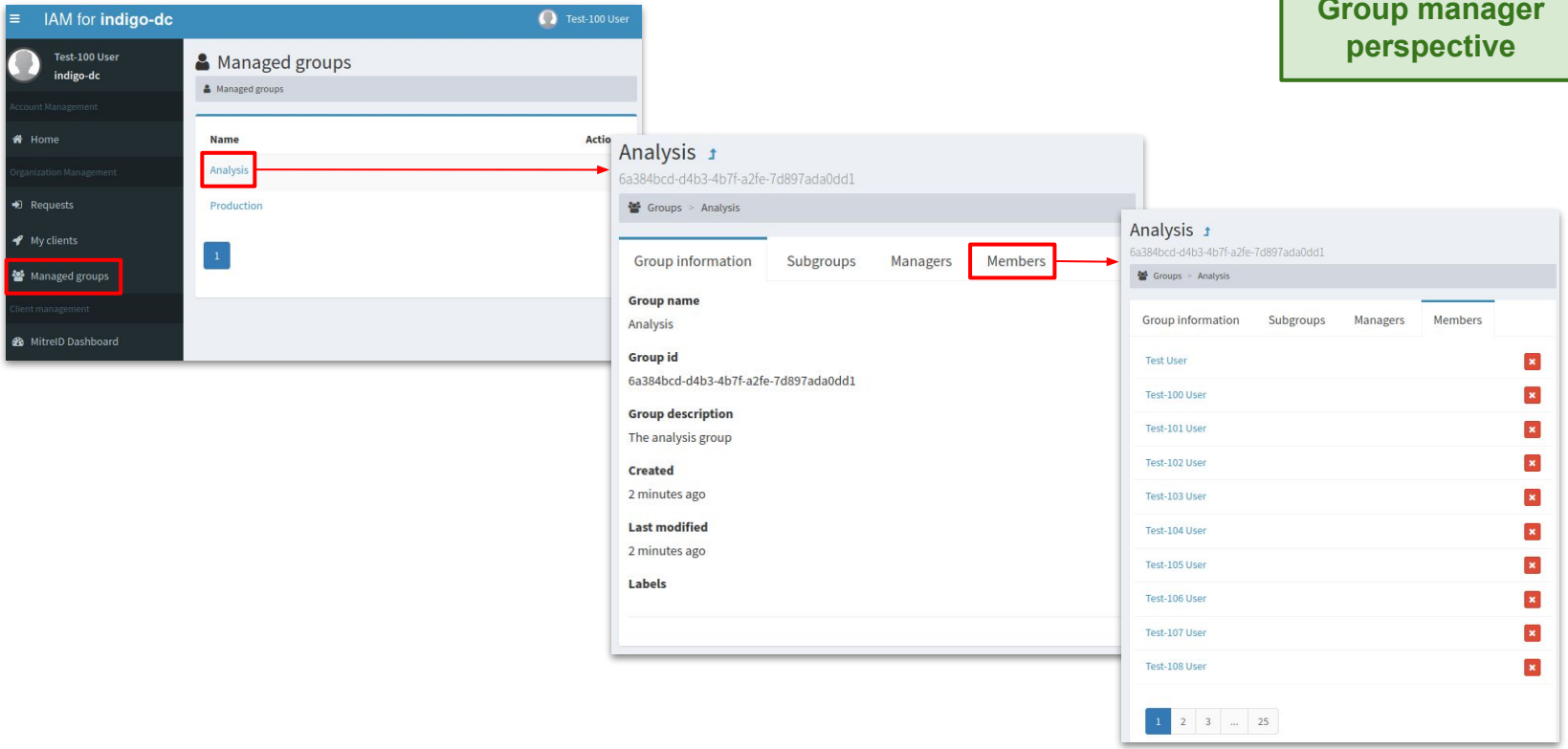






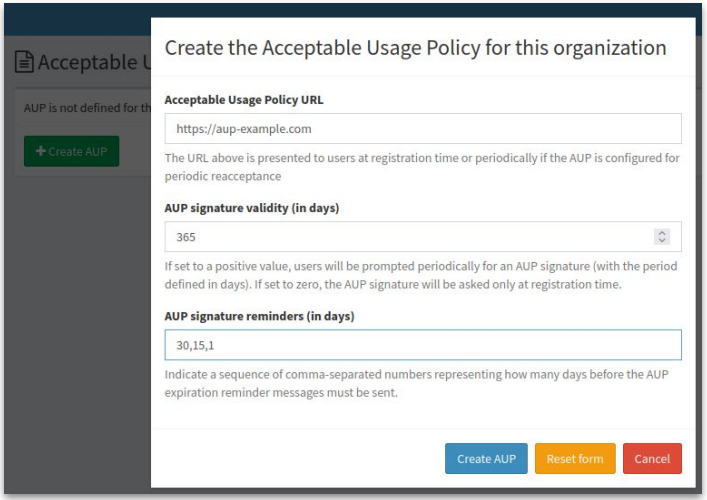
Group management

Group manager perspective



AUP enforcement support

- **AUP acceptance**, if enabled, can be configured by admins to be
  - requested once at user registration time
  - requested periodically, with configurable period
- AUP reminders can be set up so that a few days before the AUP signature expires, emails are sent to users to invite them to re-subscribe
- User cannot login to the system (and as such be authenticated and authorized at services) unless the AUP has been accepted



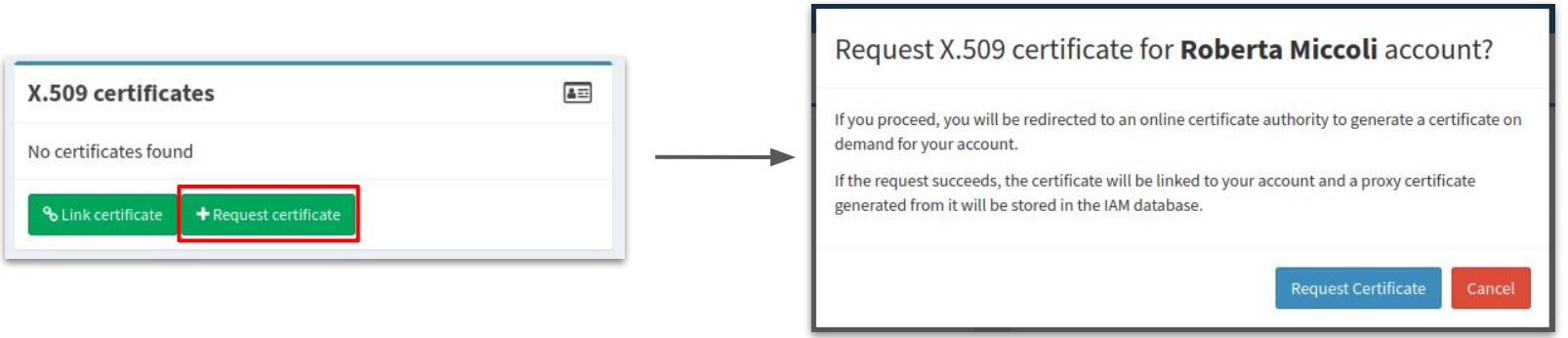
On-demand X.509 certificate generation

- INDIGO IAM integrates with the RCAuth.eu online certificate authority so that **users without an X.509 certificate can easily request one and link it to their membership**, via the IAM dashboard

- Must be enabled by configuration
- An IAM OAuth client has to be registered on the RCAuth service that will issue an X.509/proxy certificate

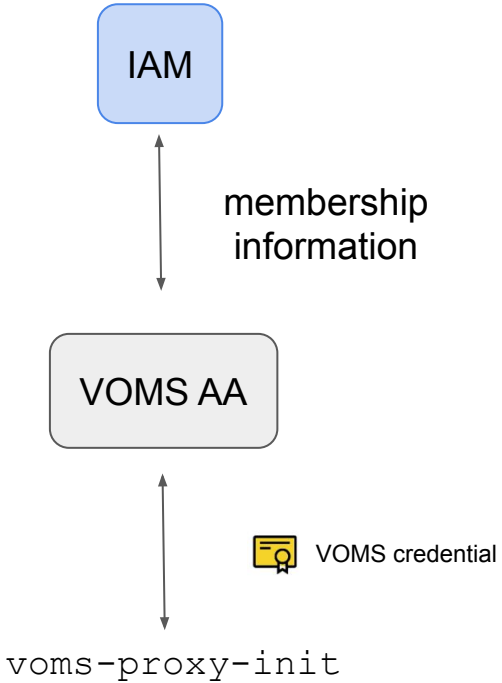
On-demand X.509 certificate generation

- **A long-lived X.509 proxy certificate** is generated from the certificate obtained from RCAuth and stored in the IAM database
- A **RESTful API** provides access to the certificate to **trusted clients**



VOMS provisioning

- Knowing that the transition from X.509 to tokens will take time, IAM was designed to be backward-compatible with our existing infrastructure
- IAM provides a **VOMS Attribute Authority** (VOMS AA) micro-service that can encode IAM membership information in a **standard VOMS Attribute Certificate**
- VOMS AA can issue VOMS credentials (`voms-proxy-init`) understood by existing clients



IAM JWT profiles

A **JWT profile** is a named set of rules that defines which information is included in access tokens, ID tokens, userinfo and introspection responses issued by IAM in an OAuth/OIDC message exchange

- IAM allows to define a default profile that is used for all clients
- It is also possible to define the profile per client, using scopes
  - the scope, in this context, does not identify a set of permissions but an encoding of authentication/authorization information in tokens and responses issued by IAM
  - this mechanism enables integration of the same IAM instance with resources relying on different profiles

IAM JWT profiles

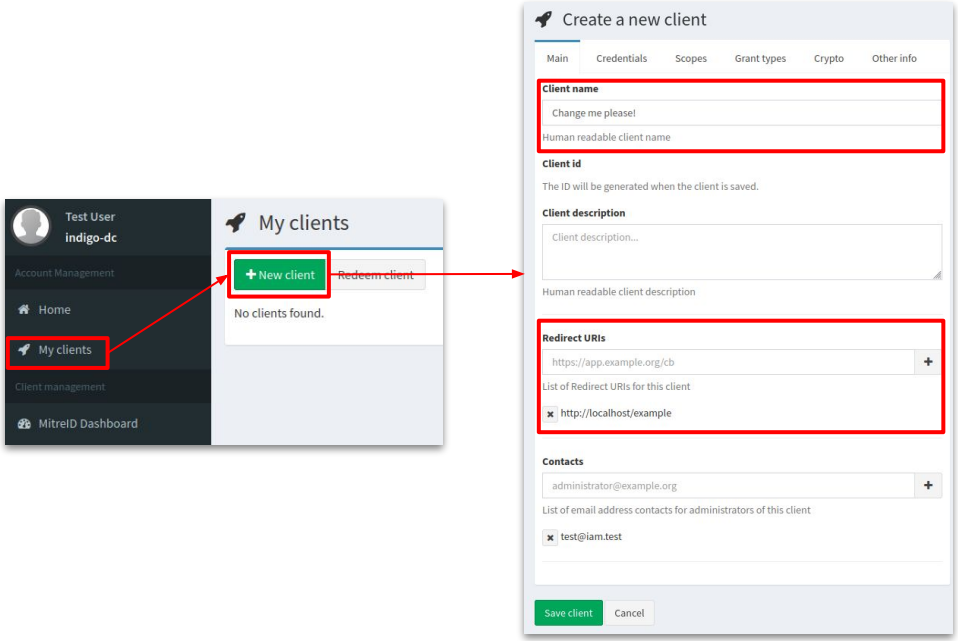
IAM currently supports four JWT profiles

- the **`iam`** profile (default)
  - groups are encoded in the `groups` claim
- the **`wlcg`** profile
  - compliant with the WLCG JWT profile
  - (*default*) groups can be requested using the `wlcg.groups` scope and are encoded in the `wlcg.groups` claim
  - *optional groups* (labelled with `wlcg.optional-group`) are groups whose membership is only asserted in tokens on explicit request coming from a user
- the **`aarc`** profile
  - compliant with AARC-G002/AARC-G069 guidelines
  - organisation name can be requested using the `eduperson_scoped_affiliation` scope and it's encoded in the `eduperson_scoped_affiliation` claim
  - groups can be requested using the `eduperson_entitlement` scope and they're encoded as URN in the `eduperson_entitlement` claim
- the **`keycloak`** profile
  - groups are encoded in the `roles` claim
    - using `roles` instead of `groups` enables proper user mapping in Keycloak (acting as a resource), which is required for OIDC client integration – for example, with a containerised MISP deployment

Client registration

- A client is an application that requests access to a user's protected resources on their behalf. It authenticates with the authorization server and uses OAuth tokens to obtain the necessary permissions
- Anyone can register a client, even anonymous users (default behaviour)
- IAM has the ability to enable or disable the client registration, and to limit the registration to certain users (registered users, administrators)
- IAM exposes the **OpenID Connect/OAuth dynamic client registration** functionality on its own dashboard
- A new client can be registered in the IAM in two ways
  - using the dynamic client registration API
  - via the IAM dashboard (which simply acts as a client to the API mentioned above)

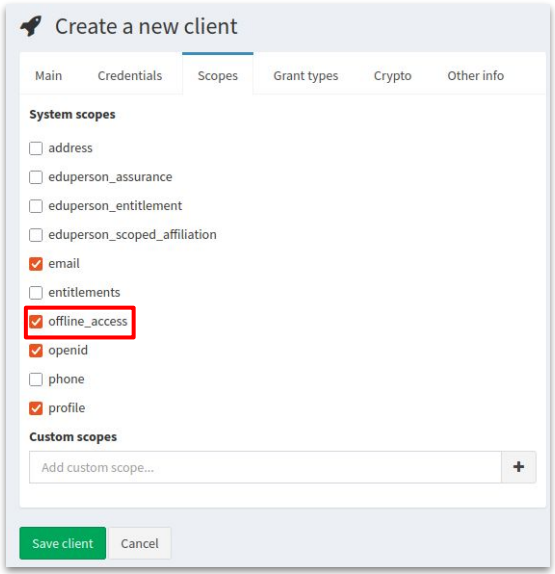
Registering a client using the dashboard



The minimum information you have to provide is

- *Client name*
- *Redirect URI(s)*: one or more redirect URIs, required if authorization code flow is selected

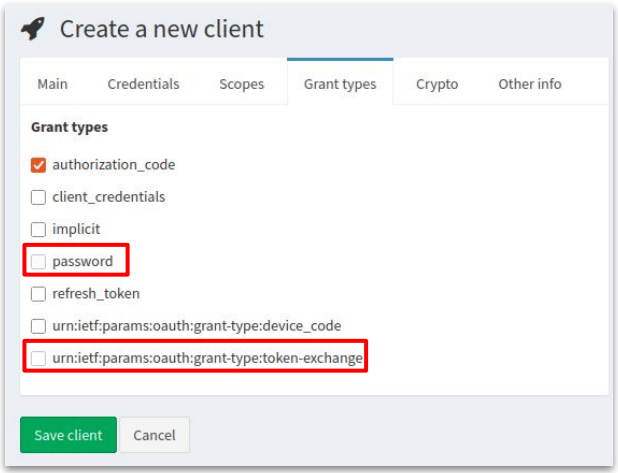
Registering a client using the dashboard



Select the `offline_access` scope from the *Scopes* tab if you want to request refresh tokens for the client being created

Restricted scopes can only be assigned to clients by admin users. As for the **admin** (`iam:admin.read` and `iam:admin.write`) and **SCIM** (`scim:read` and `scim:write`) scopes, only admin users can delegate clients to use them. *Client credentials* clients may obtain access tokens with these privileged/restricted scopes if and only if they have been previously assigned by the administrator.

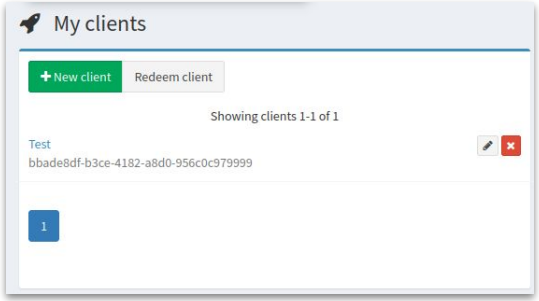
Registering a client using the dashboard



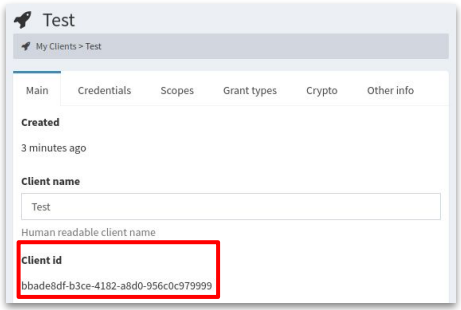
Select the *grant types* that determine how the application will obtain access tokens

- Token exchange & password - restricted (assigned only by admins)
- Implicit - deprecated

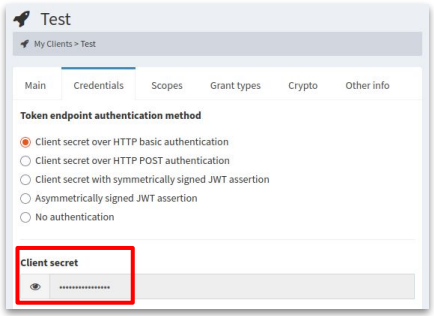
Registering a client using the dashboard



After saving the client, IAM generates the client credentials (client id and client secret) for the client



Random UUID



Strong alphanumeric string, similar to a password

Requesting an AT

```
$ curl -u <client-id>:<client-secret> -d
grant_type=client_credentials
http://localhost:8080/token | jq
{
  "access_token":
"eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiI2Z
GU3NTFkYi01YTY3LTRhMWYtO…0OS02OTliNjc4MTg2ODUifQ.rnM
Bg1yKo…lVxmzrbuae9SHPXge4y_fA",
  "token_type": "Bearer",
  "expires_in": 3599,
  "scope": "openid email offline_access profile"
}
```

Registering a client from CLI

- IAM supports the OpenID Connect Dynamic Client Registration API at the `/iam/api/client-registration` endpoint
- Client registration does not require authentication

```
{
  "redirect_uris": [
        "https://another.client.example/oidc"
  ],
  "client_name": "another-example-client",
  "contacts": [
        "test@iam.test"
  ],
  "token_endpoint_auth_method": "client_secret_basic",
  "scope": "address phone openid email profile
offline_access",
  "grant_types": [
        "refresh_token",
        "authorization_code"
  ],
  "response_types": [
        "code"
  ]
}
```

JSON file: client-req.json

```
$ curl -X POST -d @client-req.json
https://iam-dev.cloud.cnaf.infn.it/iam/api/
client-registration
```

Token exchange and **password** grant types are NOT allowed for registered users; for anonymous users not even **client credentials**!

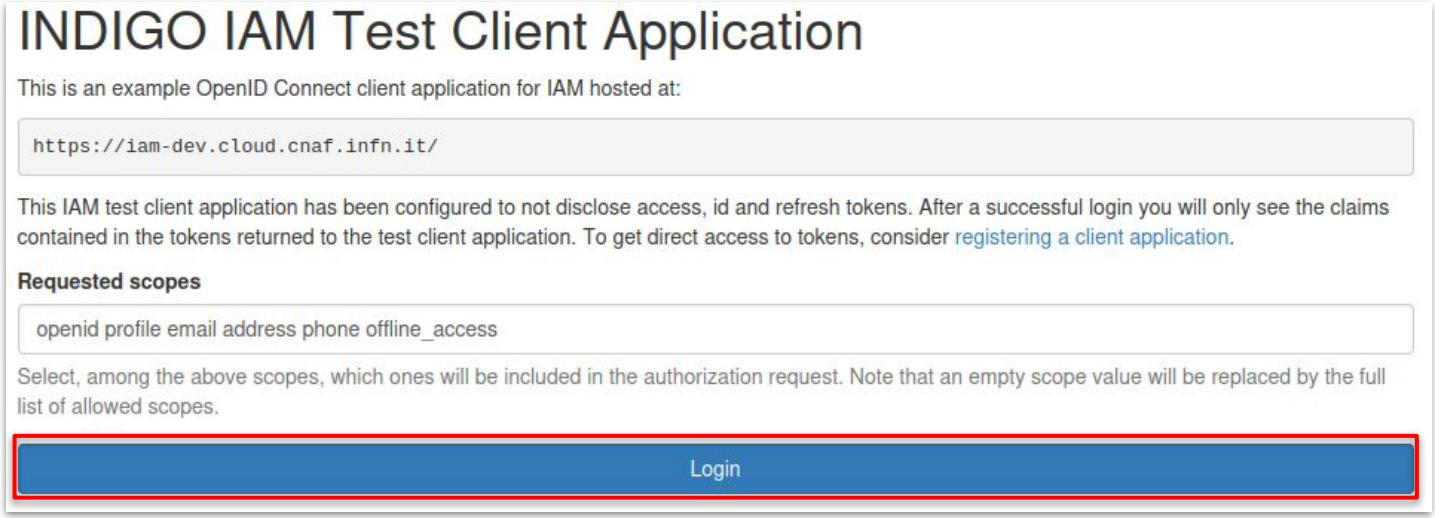
IAM Test Client application

The IAM Test Client application is designed to simulate OAuth 2.0 and OpenID Connect authorization flows, allowing users to test and verify their identity and access management configurations

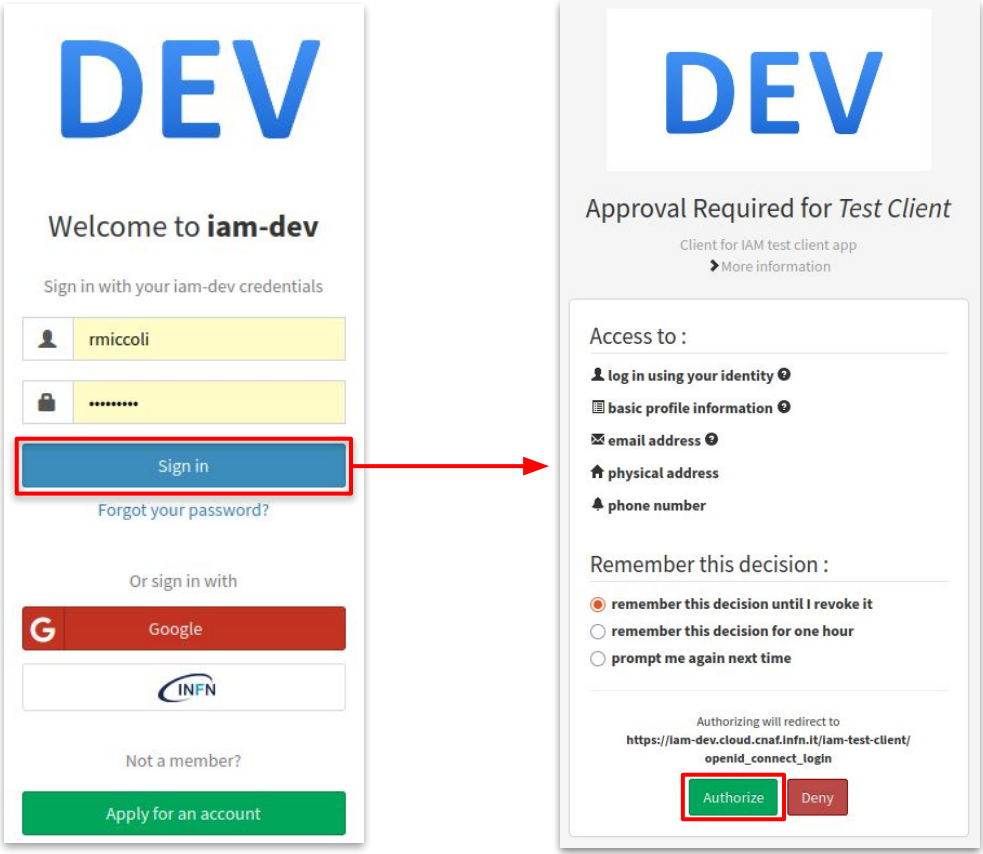
- It lives at `https://your-IAM-instance/iam-test-client`
- To run the Test Client application you need to setup a minimal configuration

```
# Public identifier for client application
IAM_CLIENT_ID=client

# Client application's own password
IAM_CLIENT_SECRET=secret

# Default scopes allowed to the client application (optional)
IAM_CLIENT_SCOPES=openid profile email

# Use forwarded headers from reverse proxy. Set this to native when deploying the
# service behind a reverse proxy
IAM_CLIENT_FORWARD_HEADERS_STRATEGY=none
```

IAM Test Client application



IAM Test Client application



Since Test Client uses OAuth **Authorization Code** Grant Type, after a successful login you will be redirected to the consent page that the authorization server displays to confirm if the requested scopes can be shared with the client

IAM Test Client application

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/",
  "exp": 1746189158,
  "iat": 1746185558,
  "jti": "3b48b6b0-bc7c-4428-9a22-1fa15354d64d",
  "client_id": "42999a63-7449-43fb-952e-42f2d75b865b"
}
```


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


```
{
  "active": true,
  "scope": "address openid profile phone email",
  "expires_at": "2025-05-02T14:32:38+0200",
  "exp": 1746189158,
  "sub": "8b7b42fd-0e42-43c5-8254-729aa8f6a12d",
  "user_id": "rmiccoli",
  "client_id": "42999a63-7449-43fb-952e-42f2d75b865b",
  "token_type": "Bearer",
  "name": "Roberta Miccoli",
  "given_name": "Roberta",
  "family_name": "Miccoli",
  "email": "roberta.miccoli@cnaf.infn.it"
}
```


Try the Test Client app following the AAI tutorials!

- id_token (claims):


```
{
  "sub": "8b7b42fd-0e42-43c5-8254-729aa8f6a12d",
  "aud": "42999a63-7449-43fb-952e-42f2d75b865b",
  "kid": "rsa1",
  "iss": "https://iam-dev.cloud.cnaf.infn.it/",
  "exp": 1746186158,
  "iat": 1746185558,
  "nonce": "177a34ab4d7a",
  "jti": "c4b70e48-377c-4f3c-9d40-b04efef9c612"
}
```


- OpenID-Connect user info endpoint response (authorized via access_token):


```
{
  "sub": "8b7b42fd-0e42-43c5-8254-729aa8f6a12d",
  "name": "Roberta Miccoli",
  "preferred_username": "rmiccoli",
  "given_name": "Roberta",
  "family_name": "Miccoli",
  "picture": "http://t0.gstatic.com/licensed-image?q=tbn:ANd9GcQdVrDbX5tCAQ6lX9axvmA12
  "updated_at": 1745426059,
  "email": "roberta.miccoli@cnaf.infn.it",
  "email_verified": true
}
```

APIs

IAM APIs - a subset

- **SCIM API** - IAM provides a RESTful API, based on the **S**ystem for **C**ross-domain **I**dentity **M**anagement (**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**
    - for example, the SCIM API is used in the integration with the HTCondor batch system to do UNIX account pre-provisioning based on IAM account information

IAM APIs - a subset

- **IAM account API** - it's a RESTful API used to manage user attributes, authorities, labels, clients, group membership, etc.
- **IAM client management & registration API** - this API solves several scalability and usability limits of old MITREid Connect API
  - **pagination** → no pagination on MITREid client management APIs causes the management dashboard to be unavailable with a large number of clients
  - **server-side search functionality** → no client search API on MITREid
  - **clients ownership** → on MITREid managing a client requires to use registration access tokens, making it hard for users to have a clear view of their registered clients; now users own their created clients and old registration access token can be used to **redeem** and link an owned client

Scope policy API

- In OpenID Connect and OAuth, scopes are used to determine the privileges granted to a client application for a given session
- IAM implements two levels of access control on OAuth scopes
  - a client-level vetting, implemented through the MITREid Connect library, so that each registered client has a list of allowed scopes
  - an identity-related vetting, implemented through the concept of **scope policies**, that can be used to limit access to scopes based on user identity
- The Scope Policy API is a REST API that allows to manage scope policies
  - access to the API is restricted to administrators authenticated via web interface or OAuth clients that have access to the admin OAuth scopes
  - endpoint at `/iam/scope_policies`

Scope policy API

- IAM scope policies provide a mechanism to control access to OAuth scopes
  - a `rule`, PERMIT or DENY, which determines the behaviour of the policy
    - PERMIT policies are used to allow access to scopes
    - DENY policies are used to block access to scopes
  - a `scopes` selector, i.e. a set of scopes for which the policy applies (and a scope `matchingPolicy` used to determine the scope matching algorithm used)
  - an `account` or `group` selector, used to determine for which user account or group of accounts the policy should apply

Example of PERMIT policy that allows access to all the scopes to any account → **default scope policy**

Scope policy API

- IAM policy engine evaluates the policy in a specific order
  - account-level policies are applied first, then group-level policies and finally those not bound to any specific account or group
- The policy engine implements a deny-override policy composition logic, which means that a deny policy matching a request at a given level wins over a permit policy at that same level

Scope policy API

- IAM currently supports three scope matching algorithms
  - `EQ` (default): uses string equality when comparing requested scopes to scopes allowed by the client configuration or by the scope policies
  - `REGEXP`: uses a regular expression evaluation when comparing requested scopes to scopes allowed by the client configuration or by the scope policies
  - `PATH`: uses a WLCG-specific path-matching logic to compare requested scopes to scopes allowed by the client configuration or by the scope policies

Scope policy API

REGEXP and PATH matching algorithms are configured by adding a `scope.matchers` section to the IAM configuration, which defines the scope matching algorithm for WLCG profile scopes

```
scope:
  matchers:
        - name: storage.read
        type: path
        prefix: storage.read
        path: /
        - name: storage.create
        type: path
        prefix: storage.create
        path: /
        - name: storage.modify
        type: path
        prefix: storage.modify
        path: /
        - name: wlcg.groups
        type: regexp
        regexp: ^wlcg\.groups(?::((?:\/[a-zA-Z0-9][a-zA-Z0-9_.-]*)+))?$
```

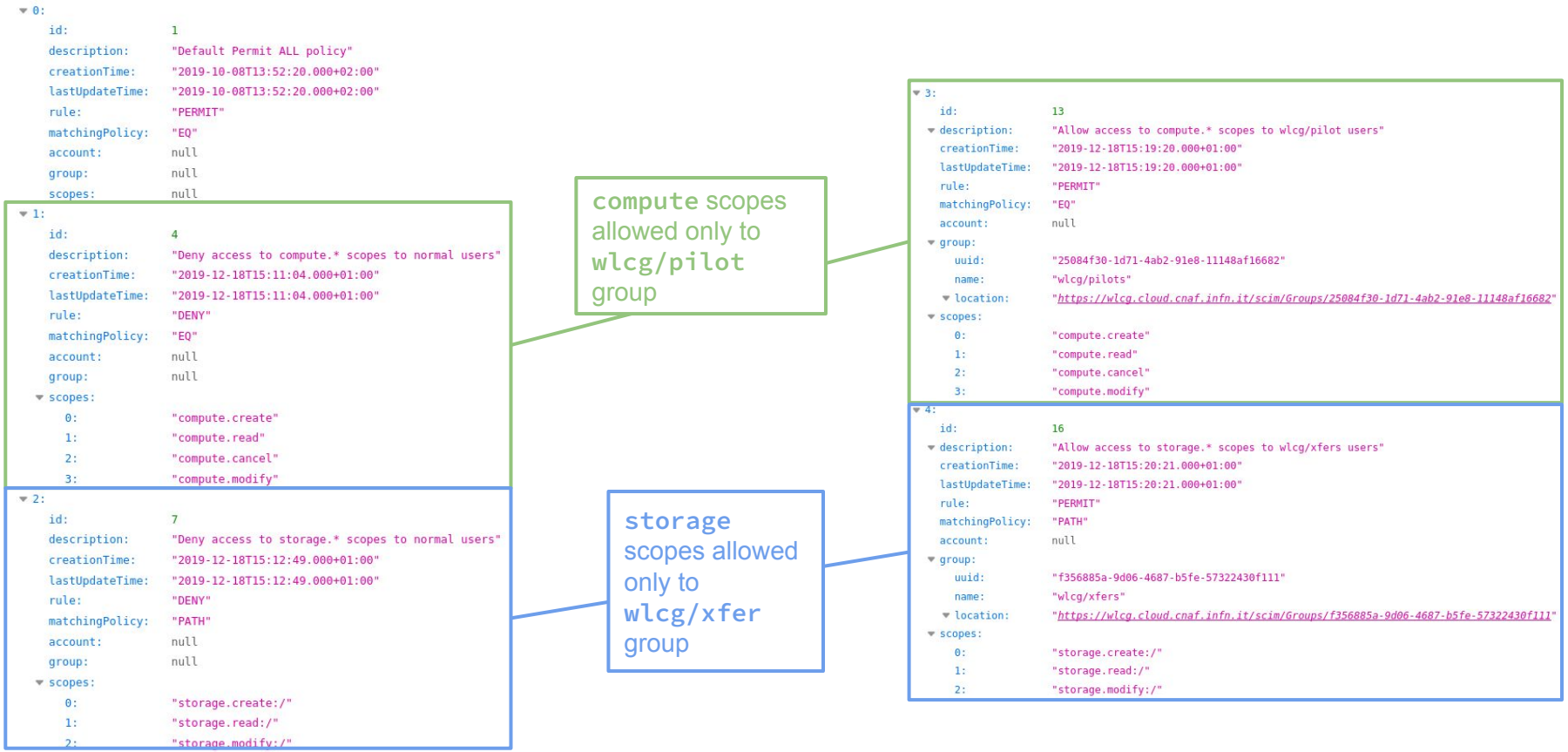
PATH scope matching

- Follows the SciTokens model
- Enables defining scopes with hierarchical paths, e.g. `storage.read:/cms`
- Permissions apply recursively to all subpaths

REGEXP scope matching

- Uses regular expressions to define which scopes are allowed for a client or user
- In this example, a client authorized for `wlcg.groups` is also allowed to request `wlcg.groups:/a/group` and any scope that matches the regular expression
- Flexible matching for scopes that include dynamic values, like group names

Example of IAM scope policies



Support and resources

Useful references

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

Contacts:

- iam-support@lists.infn.it

Q&A

Bkp

Open Policy Agent (OPA)

Exploring AuthZ with OPA

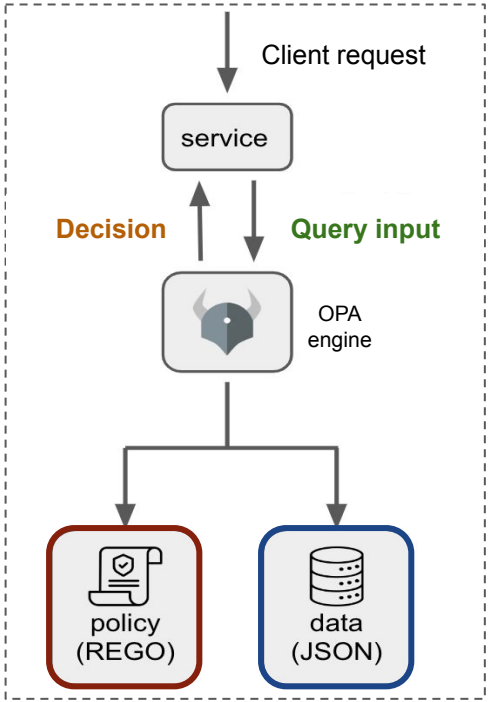
Open Policy Agent (OPA) is an open-source authorization engine based on a high-level declarative language (***Rego***) that allows the definition of policies as code

Rego is designed to express policies over complex hierarchical data structures

- policy authors can focus on what queries should return rather than how they should be executed
- Rego ensures high performance policy decisions, even with increasing number of rules

A service which needs to take policy decisions can **query** OPA with arbitrary structured data (JSON or YAML) as **input**

- OPA evaluates the query input against **policies** and optionally **data**
- OPA **decision** is not limited to a simple allow/deny answer, but can generate arbitrary structured data as output



Integration with INDIGO IAM

OPA is going to replace and evolve the IAM Scope Policy API

- more readable policies
- policies are also applied to clients to support the OAuth *client credentials* flow (not bound to a user)
- backward compatible with current IAM scope policies syntax

An OPA query took **~130 ms** to parse 10k policies, which in IAM reached the client timeout!

More details here

New IAM dashboard

Motivation

- Current dashboard is based on **JavaServer Pages** (JSP) and **Angular.js** which is in EOL since January 2022
- Drop deprecated libraries in order to increase security
- Decouple the frontend logic from the backend service
- Modern web development
- Lightweight and responsive
- Customization (anyone can fork and extend/modify the dashboard for their own needs)

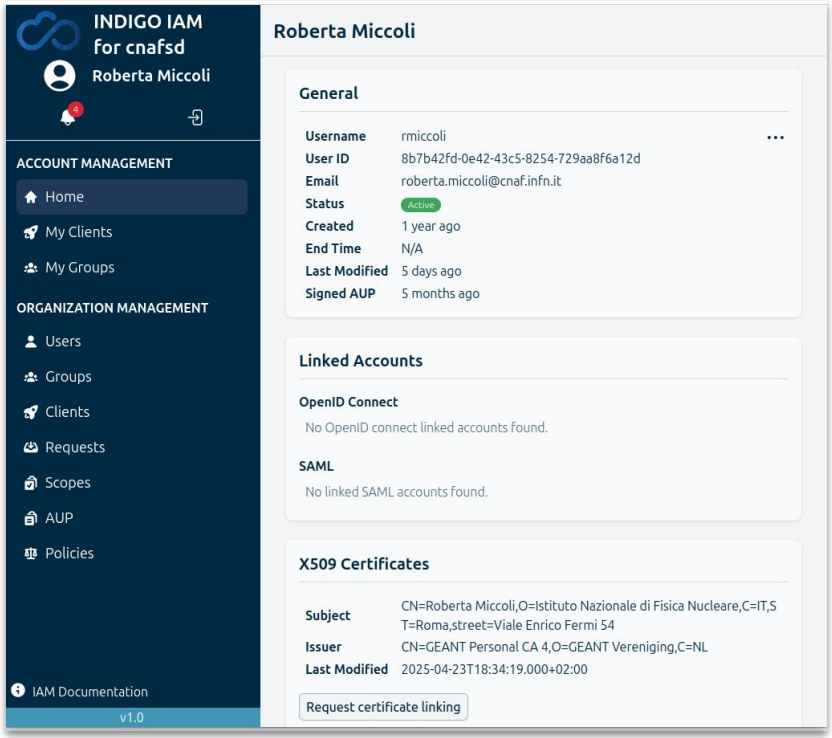
IAM dashboard: a Next.js/React web application

- The new IAM dashboard is a web application written in **TypeScript** on top of the Next.js framework, the official framework indicated by the React developers
  - in other words, the new IAM Dashboard is a Next.js application whose components are written in React
- It is both a Node.js web server which serves static and dynamically rendered content (compiled html, js and css) AND a real API that the browser interacts with
- It is based on the Backend for Frontend (BFF) pattern

More details here

Development status

- Most of the current features have been successfully implemented
- Final User Interface and User Experience (UI/UX) not yet defined
- Changes to improve UI/UX are still under investigation
- Code review needed
- End-to-end tests are being written to increase coverage
- We hope to receive feedback from the users!



https://iam-dashboard.cloud.cnaf.infn.it/