Authentication and Authorization with INDIGO IAM

Enrico Vianello, Andrea Ceccanti
INFN CNAF

Corso Nazionale CCR Big Data Analytics
Bologna, 9 Dicembre 2019
Agenda

AAI challenges

OAuth & OpenID connect overview

IAM overview

A token-based AAI - in practice

Beyond X.509: migrating from X.509/VOMS to tokens

Hands on
HANDS ON
Apply for an IAM account

Please point your browser to:

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

and apply for an account.
Apply for an IAM account

Use a valid e-mail address

Note: Use a username without special characters

In the notes field put "Corso Big Data 2019"

You will use that account later in the tutorial
Apply for an IAM account

Check your e-mails and click on confirmation link:

iam-demo@cloud-vm195.cloud.cnaf.infn.it
Confirm your iam-demo registration request
A: Enrico Vianello

Dear Enrico Vianello,

you have requested to be a member of iam-demo.

In order for the registration to proceed, please confirm this request by going to the following URL:

https://iam-demo.cloud.cnaf.infn.it/registration/verify/e222995d-fb38-4855-a60

The iam-demo registration service

The following user has submitted a membership request:
Name: Enrico Vianello
Username: test
Email: en.vianello@gmail.com
Notes: test

You can approve or reject this request by following the link below:
https://iam-demo.cloud.cnaf.infn.it/dashboard/#/requests

The iam-demo registration service

Request confirmed successfully

Your registration request has been confirmed successfully, and is now waiting for administrator approval. As soon as your request is approved you will receive a confirmation email.

Back to Login Page
Apply for an IAM account

Wait for admins approval
Apply for an IAM account

Check your e-mails again and click on password-reset link:

iam-demo@cloud-vm195.cloud.cnaf.infn.it
Your iam-demo account is now active
A: Enrico Vianello

Dear Enrico Vianello,
your registration request has been approved.
You can set your password by following this link:
https://iam-demo.cloud.cnaf.infn.it/iam/password-reset/token/17f26a14-5deb-4d38-aa

The iam-demo registration service
Apply for an IAM account

Try to login:

You will use that account later in the tutorial
A novel AAI: main challenges
A novel AAI: main challenges

Authentication

• **Flexible**, able to accommodate 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
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
A brief introduction to OAuth and OpenID Connect
IAM 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
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. API)
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**

```
Qb0fPrha9kp4e7TknXe88
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
OAuth roles

Resource owner

- A user that owns resources hosted at a service

Client

- An application that wants to have access to user resources

Authorization server

- A service that authenticates users and client applications and issues access tokens according to some policy

Resource server

- A service that holds protected resources and grants access based on access tokens issued by the authorization server
OAuth client registration

In OAuth clients that interact with an Authorization Server (AS) need to be registered.

When a client is registered, it typically receives the client credentials:

- **client_id**: the client “username”
- **client_secret**: the client “password”

Credentials are required in some OAuth flows or to access specific endpoints, where different privileges may be assigned to different clients.
OAuth client types

https://tools.ietf.org/html/rfc6749#section-2.1

**confidential:** Clients capable of maintaining the confidentiality of their credentials (e.g., client implemented on a secure server with restricted access to the client credentials), or capable of secure client authentication using other means.

**public:** Clients incapable of maintaining the confidentiality of their credentials (e.g., clients executing on the device used by the resource owner, such as an installed native application or a web browser-based application), and incapable of secure client authentication via any other means.
Handling client credentials

Client credentials must be maintained confidential

- **not** stored in Docker images or source code
  - use ENV variables or other secret management mechanisms to pass down these secrets to your application

Follow recommendations in the client app security section of the OAuth security recommendations

- [https://tools.ietf.org/html/rfc6819#section-5.3](https://tools.ietf.org/html/rfc6819#section-5.3)
OAuth/OpenID Connect grant types

Authorization grant types = Authorization Flows = Ways for an application to get tokens
# OAuth/OpenID Connect grant types

<table>
<thead>
<tr>
<th>Grant Type</th>
<th>Context</th>
<th>Client type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization code</td>
<td>Server-side apps</td>
<td>Confidential</td>
</tr>
<tr>
<td>Implicit</td>
<td>Client-side, Javascript apps</td>
<td>Public</td>
</tr>
<tr>
<td>Device code</td>
<td>Limited-input devices, CLIs</td>
<td>Confidential</td>
</tr>
<tr>
<td>Resource owner password credentials</td>
<td>Trusted apps, CLIs</td>
<td>Confidential</td>
</tr>
<tr>
<td>Client credentials</td>
<td>Server-side apps</td>
<td>Confidential</td>
</tr>
<tr>
<td>Refresh token</td>
<td>Server-side apps</td>
<td>Confidential</td>
</tr>
<tr>
<td>Token exchange</td>
<td>Server-side apps</td>
<td>Confidential</td>
</tr>
</tbody>
</table>
OAuth/OpenID Connect provider metadata

OAuth & OpenID Connect provide a standard way to expose the authorization server/OpenID provider configuration to clients.

Information is published at a well-known endpoint for the server, e.g.:

- https://iam-demo.cloud.cnaf.infn.it/.well-known/openid-configuration

Clients can use this information to know about:

- supported grant types/authorization flows
- endpoint locations
- supported claims
- ...

and implement automatic client configuration.
OAuth/OpenID Connect provider metadata

```json
{
  "request_parameter_supported": true,
  "claims_parameter_supported": false,
  "introspection_endpoint": "https://dodas-iam.cloud.cnaf.infn.it/introspect",
  "scopes_supported": ["openid", "profile", "email", "address", "phone", "offline_access"],
  "issuer": "https://dodas-iam.cloud.cnaf.infn.it/",
}
```
OAuth/OpenID Connect provider metadata

"claims_supported": [
  "sub",
  "name",
  "preferred_username",
  "given_name",
  "family_name",
...
  "zoneinfo",
  "locale",
  "updated_at",
  "birthdate",
  "email",
  "email_verified",
  "phone_number",
  "phone_number_verified",
  "address",
  "organisation_name",
  "groups",
  "external_authn"
],
...

OAuth/OpenID Connect provider metadata

{
   "authorization_endpoint": "https://dodas-iam.cloud.cnaf.infn.it/authorize",
   "claim_types_supported": ["normal"],
   "claims_parameter_supported": false,
   "claims_supported": [
      "sub",
      "name",
      "preferred_username",
      "given_name",
      "family_name",
      "middle_name",
      ...
   ],
   "code_challenge_methods_supported": [
      "plain",
      "S256"
   ],
   "grant_types_supported": [
      "authorization_code",
      "implicit",
      "refresh_token",
      "client_credentials",
      "password",
      ...
   ]
}
OAuth/OpenID Connect provider metadata

"password",
"urn:ietf:params:oauth:grant-type:jwt-bearer",
"urn:ietf:params:oauth:grant-type:redelegate",
"urn:ietf:params:oauth:grant-type:token-exchange"
],
"id_token_encryption_alg_values_supported": [ "RSA-OAEP",
"RSA-OAEP-256",
"RSA1_5"
],
"id_token_encryption_enc_values_supported": [ “A256CBC+HS512",
...,
],
"id_token_signing_alg_values_supported": [ "HS256",
“HS384",
...,
],
"introspection_endpoint": "https://dodas-iam.cloud.cnaf.infn.it/introspect",
"issuer": "https://dodas-iam.cloud.cnaf.infn.it/",
"jwks_uri": "https://dodas-iam.cloud.cnaf.infn.it/jwk",
"op_policy_uri": "https://dodas-iam.cloud.cnaf.infn.it/about",
"op_tos_uri": "https://dodas-iam.cloud.cnaf.infn.it/about",
OAuth/OpenID Connect provider metadata

"registration_endpoint": "https://dodas-iam.cloud.cnaf.infn.it/register",
"request_object_encryption_alg_values_supported": [
  "RSA-OAEP",
  ...
],
"request_object_encryption_enc_values_supported": [
  "A256CBC+HS512",
  ...
],
"request_object_signing_alg_values_supported": [
  "HS256",
  ...
],
"request_parameter_supported": true,
"request_uri_parameter_supported": false,
"require_request_uri_registration": false,
"response_types_supported": [
  "code",
  "token"
],
"revocation_endpoint": "https://dodas-iam.cloud.cnaf.infn.it/revoke",
"scopes_supported": [
  "openid",
  "profile"
"scopes_supported": [ "openid", "profile", "email", "address", "phone", "offline_access" ],
"service_documentation": "https://dodas-iam.cloud.cnaf.infn.it/about",
"subject_types_supported": [ "public", "pairwise" ],
"token_endpoint": "https://dodas-iam.cloud.cnaf.infn.it/token",
"token_endpoint_auth_methods_supported": [ "client_secret_post", "client_secret_basic", "none" ],
"token_endpoint_auth_signing_alg_values_supported": [ "HS256", ...,
OAuth/OpenID Connect provider metadata

"userinfo_encryption_alg_values_supported": [
  "RSA-OAEP",
  ...
],
"userinfo_encryption_enc_values_supported": [
  "A256CBC+HS512",
  ...
],
"userinfo_endpoint": "https://dodas-iam.cloud.cnaf.infn.it/userinfo",
"userinfo_signing_alg_values_supported": [
  "HS256",
  ...
]
IAM, relying parties & OAuth roles

Resource owner

StoRM
WebDAV

Resource Server

OneData

Client Resource Server

IAM

Authorization Server
Resource Server
IAM, relying parties & OpenID Connect roles
IAM overview
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 mechanisms.

The IAM Web appearance can be customized to include a community logo, AUP and privacy policy document.
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.
User enrollment & registration service

IAM supports two enrollment 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-enrollment flow

• Users authenticated at trusted, configurable SAML IdPs are automatically on-boarded, without administrator approval
Management tools

IAM provides a mobile-friendly dashboard for:

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

All management functionality is also exposed by REST APIs
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
Easy integration with services

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

We have successfully integrated IAM with minimal effort with:

- Openstack
- Atlassian JIRA & Confluence
- Kubernetes
- Moodle
- RocketChat
- Grafana
- JupyterHub
IAM Software Quality

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

- now 24k lines of code, 86% branch coverage, >900 tests

Open, test-driven development process

Static analysis tools

- SonarCube IAM page

Multiple test suites

- Unit tests
- Frontend test suite (based on Selenium and Robot framework)
- Deployment tests (in CI)
IAM evolution: porting to Keycloak

IAM 2 (in development) will be based on Keycloak

- Powerful RedHat SSO solution
- Vibrant community: > 250 GitHub contributors
- LDAP/Kerberos integration
- Multi-tenancy

We will focus on what not already provided by Keycloak

- flexible registration service
- X.509 and VOMS authentication support

Improved flexibility and sustainability
A token-based AAI - in practice
A token based AAI - WLCG model

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
- **OAuth scopes**: capabilities linked to access tokens at token creation time
The central authorization servers provide attributes that can be used for authorization at services, e.g.:

- groups/roles, e.g.: cms, production-manager
- capabilities, e.g.: storage.read:/cms, 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 functionality based on this information. Examples:

- allow read access on the /cms to all members of the cms group
- allow read access on the /atlas namespace to anyone with the capability read:/atlas
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.

**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.
Identity-based vs Scope-based Authorization

The two models can co-exist, even in the context of the same application!

The two models can co-exist:

- scope-based authZ
- identity-based authZ

* Slide courtesy of B. Bockelman
OAuth bearer token usage

There’s a standard that defines how to send tokens to resource servers. Typically, tokens are sent in the Authorization HTTP header, following the rules defined in RFC 6750, as in the following example HTTP request:

```
GET /shared-oauth HTTP/1.1
Host: apache.test.example
Authorization: Bearer eyJraWQiOiJy...rYI
User-Agent: curl/7.65.3
Accept: */*
```

The token!
Refresh tokens are the credentials that can be used to acquire new access tokens.

The lifetime of a refresh token is much longer compared to the lifetime of an access token. Refresh tokens can also expire.
Beyond X.509
Migrating from X.509/VOMS to tokens

IAM implements **VOMS provisioning** to expose authentication and authorization information in the form of a **VOMS attribute certificate**, compatible with existing VOMS clients.

IAM integrates with **RCAuth.eu** online CA to generate X.509 certificates on-demand and link them to IAM user memberships.

A **gradual transition** towards token-based authn/authz is thus possible.
Migrating from X.509/VOMS to tokens

Beyond X.509: Token-based Authentication and Authorization for HEP (CHEP 2018)
https://indico.cern.ch/event/587955/contributions/3012583/
attachments/1685421/2709996/CHEP-2018-Beyond-X.509-AC.pdf

WLCG AuthZ Working Group Demos: https://indico.cern.ch/event/791175/attachments/1806605/2948665/demos.mp4 (IAM starts at minute 46)
Web application integration scenario
A Web App integrates with IAM to delegate user authentication management and obtain authorization information.
Web application: authorization code flow

OAuth and OpenID connect provide the authorization code flow in support of this integration use case.
Authorization code flow

User points its browser to web app, which redirects back to IAM for authentication.
Authorization code flow

User points its browser to web app, which redirects back to IAM for authentication
Authorization code flow

User does not have a valid session at IAM, so IAM shows the login page.
Authorization code flow

User does not have a valid session at IAM, so IAM shows the login page authorization request.
Authorization code flow

User selects EduGAIN, and chooses his home IDP for authentication
Authorization code flow

Home IdP

User does not have a valid session at IAM, so IAM shows the login page.

Sign in with your IdP

You will be redirected for authentication to:

INFN - Istituto Nazionale di Fisica Nucleare

Proceed?

Sign in with IdP

Remember this choice on this computer

Search again
Back to login page
Authorization code flow

User is redirected to home IDP for authentication
Authorization code flow

Home IdP

User is redirected to home IDP for authentication

INFN Identity Check

Username

Password

LOGIN

Come ottenere un accesso ad INFN-AAI
Cambio o Rigenerazione Password - Recupero Username

X.509 Certificate
Accesso tramite certificato.

ACCEDI

Kerberos5 GSS-API
Accesso tramite Kerberos 5.
Authorization code flow

Home IDP authenticates user and sends back an authentication assertion, via redirection and possibly other interactions between IAM and the IDP.
Authorization code flow

IAM validates the assertion, the user is a registered one, so IAM shows a “Give consent” page.
Authorization code flow

IAM validates the assertion, the user is a registered one, so IAM shows a "Give consent" page.

---

Approval Required for *Web App*

**Access to:**
- log in using your identity
- basic profile information
- email address
- physical address
- telephone number
- offline access

**Remember this decision:**
- remember this decision until I revoke it
- remember this decision for one hour
- prompt me again next time

Do you authorize "webapp"?

Authorize  Deny
IAM generates an **authorization code** and sends it back to web app using an HTTP redirect.
Authorization code flow

The Web App exchanges the authorization code with a couple of tokens: an access token and an id token.
Authorization code flow

In the IAM implementation, both tokens are JWT tokens.
Authorization code flow

The access token provides (mainly) authorization information

```
{
    "sub": "e1eb758b-b73c-4761-bfff-adc793da409c",
    "iss": "https://dodas-iam.cloud.cnaf.infn.it/",
    "scope": "openid profile email webapp:admin",
    "exp": 1554142904,
    "iat": 1554139304,
    "jti": "70ca3f64-7595-43b9-84f3-bba7bd34e14a"
}
```
Authorization code flow

```
{
  "sub": "e1eb758b-b73c-4761-bfff-adc793da409c",
  "kid": "rsa1",
  "iss": "https://dodas-iam.cloud.cnaf.infn.it/",
  "groups": [
    "cms",
    "cms/admins"
  ],
  "preferred_username": "andrea",
  "organisation_name": "dodas",
  "nonce": "1b4514004ffd2",
  "aud": "webapp",
  "auth_time": 1554138126,
  "name": "Andrea Ceccanti",
  "exp": 1554141104,
  "iat": 1554139304,
  "jti": "fa9551bc-0898-4770-9b9f-60737bc6e76a",
  "email": "andrea.ceccanti@cnaf.infn.it"
}
```

The **id token** provides (mainly) authentication information

Home IdP

Web App

IAM

The **id token** provides (mainly) authentication information
Authorization code flow

Both tokens are validated following the OpenID Connect guidelines, checking temporal validity, token signature, audience, etc…
Additional information about the user can be requested by querying the `/userinfo` endpoint and providing the just obtained access token for authentication/authorization purposes.
Authorization code flow

The returned JSON object contains authentication information that can overlap with the contents of the id token, depending on the IAM configuration.
Authorization code flow in practice

In practice, decent OAuth/OpenID Connect client libraries implement all the above **behind the scenes.**

As an example, **Apache mod_auth_openidc** requires the following information to enable a working OpenID Connect integration

- The OpenID Connect provider discovery/metadata URL
- Client credentials

The library then takes care of exchanging messages with the OpenID provider, implementing verification checks, and provides the obtained authentication/authorization information to the protected web application

- typically via env variables or HTTP headers
Demo setup

HTTPD is an Apache server configured with mod_auth_openidc.

The /shared directory is only accessible to users authenticated by iam-demo.

demo.cloud.cnaf.infn.it

iam-demo.cloud.cnaf.infn.it
Demo setup

HTTPD is an Apache server configured with mod_auth_openidc

The /ibergrid directory is only accessible to users authenticated by iam-demo in the ibergrid group

demo.cloud.cnaf.infn.it

iam-demo.cloud.cnaf.infn.it
Apache mod_auth_openidc configuration

ServerName demo.cloud.cnaf.infn.it

<VirtualHost _default_:80>

  OIDCProviderMetadataURL https://iam-demo.cloud.cnaf.infn.it/.well-known/openid-configuration
  OIDCClientID demo_client
  OIDCClientSecret *****
  OIDCScope "openid email profile"
  OIDCRedirectURI https://demo.cloud.cnaf.infn.it/oidc/redirect_uri
  OIDCCryptoPassphrase *****

  <Location /shared>
    ...
    AuthType openid-connect
    Require valid-user
    LogLevel debug
  </Location>
  ...
</VirtualHost>
IAM client configuration

Note that the redirect uri above matches with the one in the Apache configuration
HANDS ON
Authorization code flow

Point your browser to: https://demo.cloud.cnaf.infn.it

IAM demo

/shared The /shared directory is accessible to all authenticated users.
/ibergrid The /ibergrid directory is accessible only users in the ibergrid group.

Click on /shared protected path
Authorization code flow

Use your IAM credentials at https://iam-demo.cloud.cnaf.infn.it/login
Hi Enrico Vianello

This is the /shared section of this demo website.

You're now logged in as: vianello

This application has received the following information:

- access_token (JWT):

  eyJraWRQOljyc2ExliwiYVxnljoiUNlMyNTYqQ.eyJzdWliOiJlMzU0My05NDM2LTQ1MGUOTFiZi00MzImN2VhMTg2MjliLCJpc3MiOiJodHRwczp...wvaWFlWRIj

- access_token (decoded):

  
  
  
  "sub": "e3373543-9436-450e-91bf-439f3ea18622",
  "iss": "https://iam-demo.cloud.cnaf.infn.it/",
  "name": "Enrico Vianello",
  "groups": [
    "ibergird",
    "demo",
    "ibergird/feudal"
  ]

$ export ACCESS_TOKEN=[paste your copied token]
Access to /userinfo endpoint

Get more user info from /userinfo endpoint:

```
$ curl https://iam-demo.cloud.cnaf.infn.it/userinfo -H "Authorization: Bearer ${ACCESS_TOKEN}" | jq '.
```

```
{
  "sub": "e3373543-9436-450e-91bf-439f3ea18622",
  "name": "Enrico Vianello",
  "preferred_username": "vianello",
  "given_name": "Enrico",
  "family_name": "Vianello",
  "updated_at": 1575733577,
  "email": "enrico.vianello@cnaf.infn.it",
  "email_verified": true,
  "groups": [
    "ibergrid",
    "demo",
    "ibergrid/feudal"
  ],
  "organisation_name": "iam-demo"
}
```
Follow IAM documentation and create your client:

https://indigo-iam.github.io/docs/v/current/user-guide/client-registration.html

client name convention: demo_<surname>
Get info from introspection endpoint

Get more user info from /introspect endpoint:

```
$ export CLIENT_ID=[paste your client id]
$ export CLIENT_SECRET=[paste your client secret]

$ curl -H "Content-Type: application/x-www-form-urlencoded"
   -u ${CLIENT_ID}:${CLIENT_SECRET} -d "token=${ACCESS_TOKEN}"
   https://iam-demo.cloud.cnaf.infn.it/introspect | jq `.
```
Get info from introspection endpoint

Get more user info from /introspect endpoint:

```json
{
    "active": true,
    "scope": "openid profile email",
    "expires_at": "2019-12-08T20:25:36+0100",
    "exp": 1575833136,
    "sub": "e3373543-9436-450e-91bf-439f3ea18622",
    "user_id": "vianello",
    "client_id": "demo.cloud.cnaf.infn.it",
    "token_type": "Bearer",
    "iss": "https://iam-demo.cloud.cnaf.infn.it/",
    "groups": ["ibergrid", "demo", "ibergrid/feudal"],
    "name": "Enrico Vianello",
    "preferred_username": "vianello",
    "organisation_name": "iam-demo",
    "email": "enrico.vianello@cnaf.infn.it"
}
```
Identity based Authz

Point again your browser to: https://demo.cloud.cnaf.infn.it

Click on /ibergrid protected path

IAM demo

/shared The /shared directory is accessible to all authenticated users.
/ibergrid The /ibergrid directory is accessible only users in the ibergrid group.
Identity based Authz

Approval Required for demo.cloud.cnaf.infn.it

Do you authorize "demo.cloud.cnaf.infn.it"?

401 Unauthorized

You are not authorized to access the requested resources

Back to index
Apache mod_auth_openidc configuration

ServerName demo.cloud.cnaf.infn.it

<VirtualHost _default_:80>
  OIDCProviderMetadataURL https://iam-demo.cloud.cnaf.infn.it/.well-known/openid-configuration
  OIDCClientID demo_client
  OIDCClientSecret *****
  OIDCScope "openid email profile"
  OIDCRedirectURI https://demo.cloud.cnaf.infn.it/oidc/redirect_uri
  OIDCCryptoPassphrase *****

  <Location /ibergrid>
    ...
    AuthType openid-connect
    Require claim groups:ibergrid
    LogLevel debug
  </Location>

  ...
</VirtualHost>
Ask for Group Membership

Go to https://iam-demo.cloud.cnaf.infn.it/dashboard#/home

and wait for admins approval
Identity based Authz

Open a new Anonymous Navigation window on your browser in order to avoid using your previous session.

Point again your browser to: https://demo.cloud.cnaf.infn.it

EOSC-hub

IAM demo

/shared The /shared directory is accessible to all authenticated users.
/ibergrid The /ibergrid directory is accessible only users in the ibergrid group.

Click on /ibergrid protected path
Hi Enrico Vianello

This is the /ibergrid section of this demo website.

You're now logged in as: vianello

This application has received the following information:

- **access_token (JWT):**
  eyJraWQiOiJyc2ExliwiYWxnjoiUIMyNTYifQ.eyJzdWIiOiJIMzM3MzU0My05NDM2LTQ1MGUtOTFj

- **access_token (decoded):**

```json
{
  "sub": "e3373543-9436-450e-91bf-439f3ea18622",
  "iss": "https://iam-demo.cloud.cnaf.infn.it/",
  "name": "Enrico Vianello",
  "groups": [
    "ibergrid",
    "demo",
    "ibergrid/feudal"
  ],
  "preferred_username": "vianello",
}
```
Thanks for your attention.
Questions?
Useful references

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

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

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
- enrico.vianello@cnaf.infn.it
- indigo-aai.slack.com