INFN CLOUD
Istituto Nazionale di Fisica Nucleare

The INFN Cloud AAI: how IAM supports INFN Cloud use cases

Marica Antonacci
marica.antonacci@ba.infn.it

IAM Users Workshop

27-28 January 2021 - Online

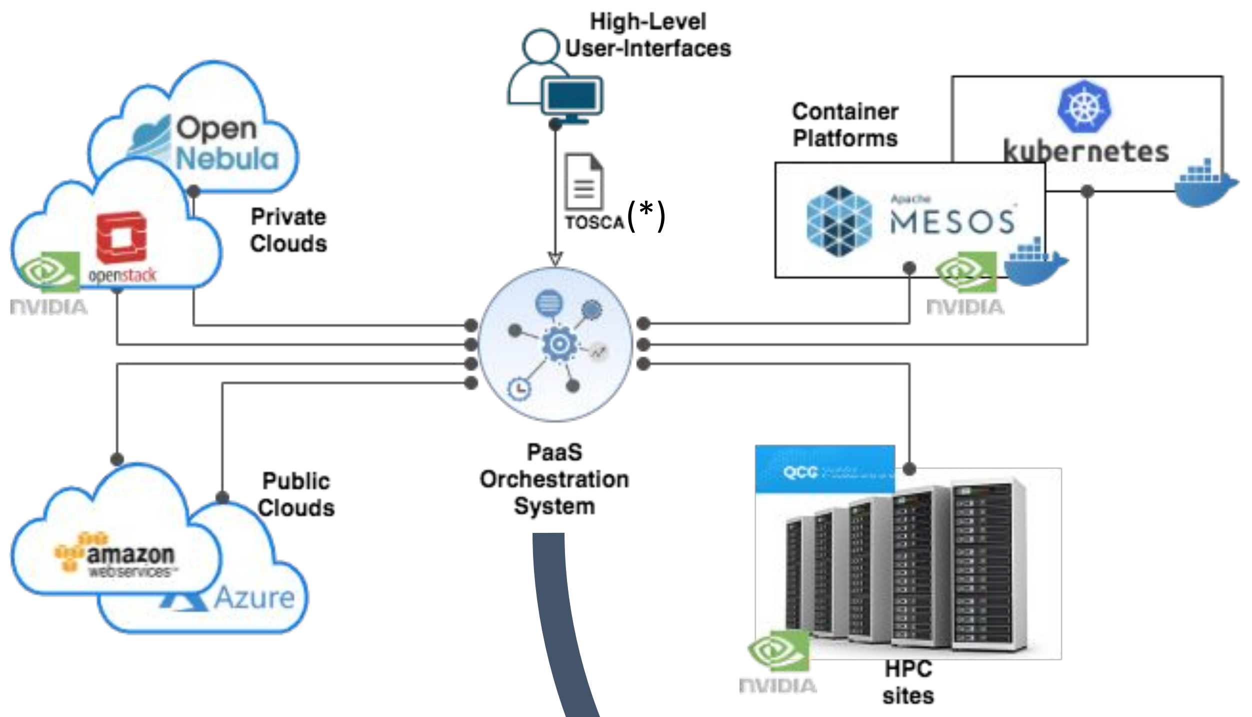
Outline

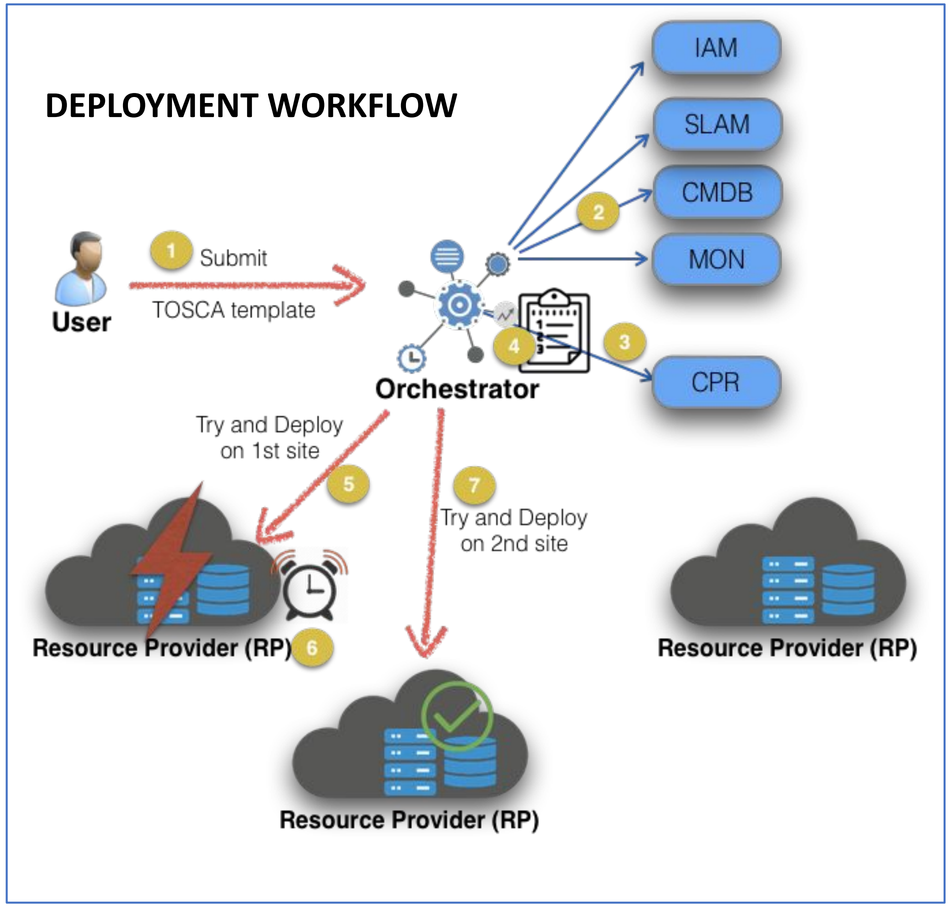
- INFN Cloud Overview
- PaaS Orchestration
- Access model & implementation
- IAM integration scenarios
  - to support automated deployments on Openstack, Mesos & Kubernetes
  - to manage users' secrets
- Conclusions

The INFN Cloud

- **INFN Cloud aims to offer a full set of high-level cloud services to INFN user communities**
  - the service catalogue is not static: new applications are included through a defined “on-boarding” process for new use-cases
- **Architecturally INFN Cloud is a federation of existing infrastructures**
  - the *INFN Cloud backbone*, consists of two tightly coupled federated sites: BARI and CNAF
  - a scalable set of satellite sites, geographically distributed across Italy, and loosely coupled.
- **Key enabling factors for the federation**
  - leverage the same authentication/authorization layer based on **INDIGO-IAM**
  - agree on a consistent set of policies and participation rules (user management, SLA, security, etc.)
  - transparent and dynamic orchestration of the resources across all the federated infrastructures through the **INDIGO PaaS Orchestrator**

PaaS Orchestration System (from 10Km)





(*) **T**opology and **O**rchestration **S**pecification for **C**loud **A**pplications

Ref: TOSCA Simple Profile in YAML Version 1.1

High-level services portfolio

- Creation of VMs with different flavors and sizes.
- Creation of containers (specify the container name) or of services via docker- compose files.
- Building blocks “as a service” for example for container orchestration (e.g. creation of a Mesos cluster or of a Kubernetes cluster as a service).
- Pre-configured environments for data analytics (e.g. using ElasticSearch and Kibana or Spark).
- Non volatile, object storage and Posix-compliant virtual file system solutions transparently connected to higher-layer services (e.g Jupyter notebooks as a service with permanent, replicated storage).
- Dynamic clusters tailored to specific experiments (e.g. an automated full HTCondor installation realized on a k8s cluster, or a GPU-based Machine Learning-optimized environment).
- Services leveraging transparent user-level encryption of disk volumes.

The service catalogue can be easily extended with the simple addition/customization of TOSCA templates.

The access model

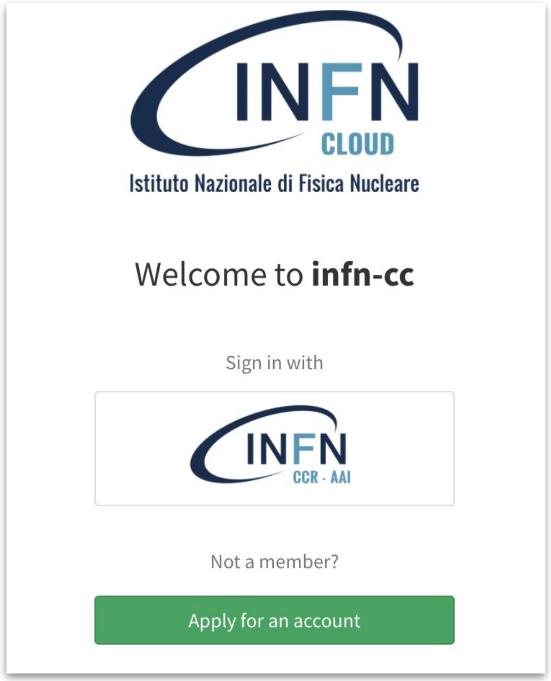
- Users are grouped in research communities (experiments/projects)
- Each Site supports a subset of communities
- Resources available at each site are partitioned into quotas assigned to each research community
  - *SLA* (site, community, quota)

- These entities and their relationships are not static: a site can extend the quota dedicated to a group or can support a new group at any time:
  - opportunistic use of the available resources
  - exploit agreements, regional projects/fundings, national or international grants

Model implementation

The INFN Cloud AAI is based on an instance of INDIGO IAM (https://iam.cloud.infn.it)

- In order to access the services and the resources of the INFN Cloud, users must be registered in the INFN Cloud IAM.
  - The registration process is simplified since it is managed through the institutional account → IAM acts as a proxy for INFN AAI
- Users in IAM are organized in groups: members of the same research community belong to the same IAM group
- Consistent group-based authorization policies are applied at all Cloud levels (IaaS, PaaS, SaaS)



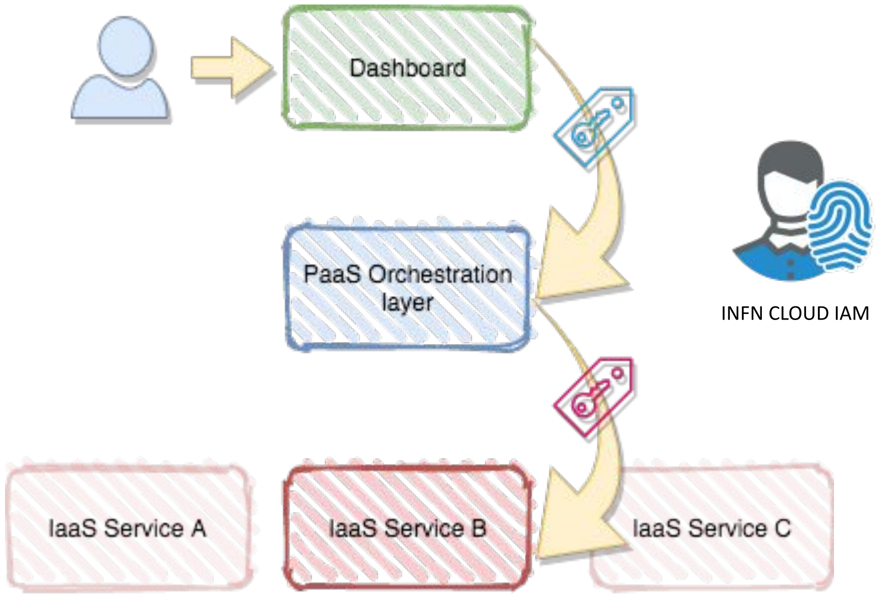
Model implementation

We are exploiting the following main capabilities provided by IAM:

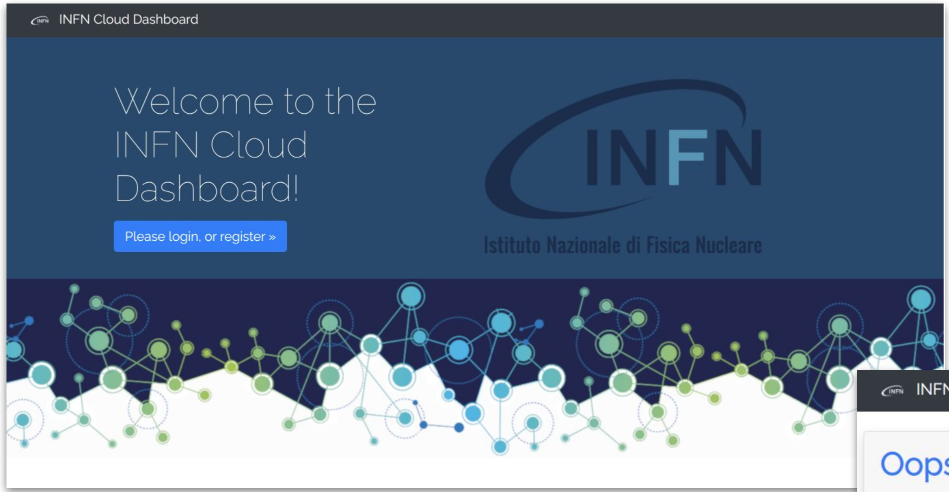
- Authorization & Membership: orthogonal to authentication, attribute-based (as described in the previous slide)
- Provide the ability for services to act on behalf of users (with token exchange, if needed)
- Support for long-running applications (token renewal)

Advantages and objectives:

- Allow the federated access to the distributed resources
  - throughout the full stack
- Allow to trace the user and link resources to its user
  - important for accounting & in case of security incidents



INFN Cloud Dashboard

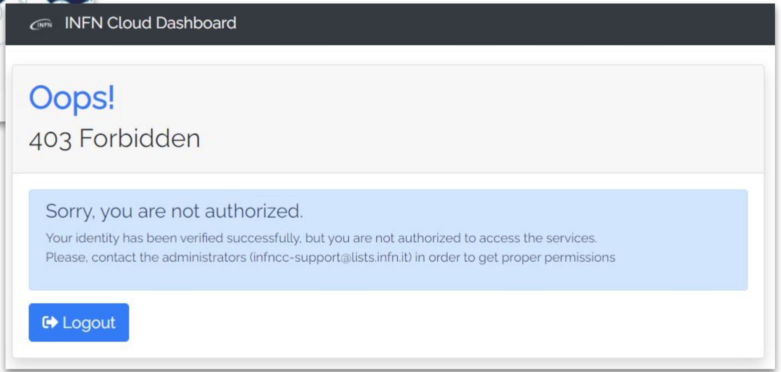

INFN Cloud Dashboard
Please login, or register »
INFN
Istituto Nazionale di Fisica Nucleare

Welcome to the INFN Cloud Dashboard!

The dashboard is written in Python3 using the *flask* framework.

The integration with IAM has been implemented leveraging the *flask-dance* extension.

After authentication via IAM, the user can access a specific set of services (personalized view) according to the group the user belongs to.



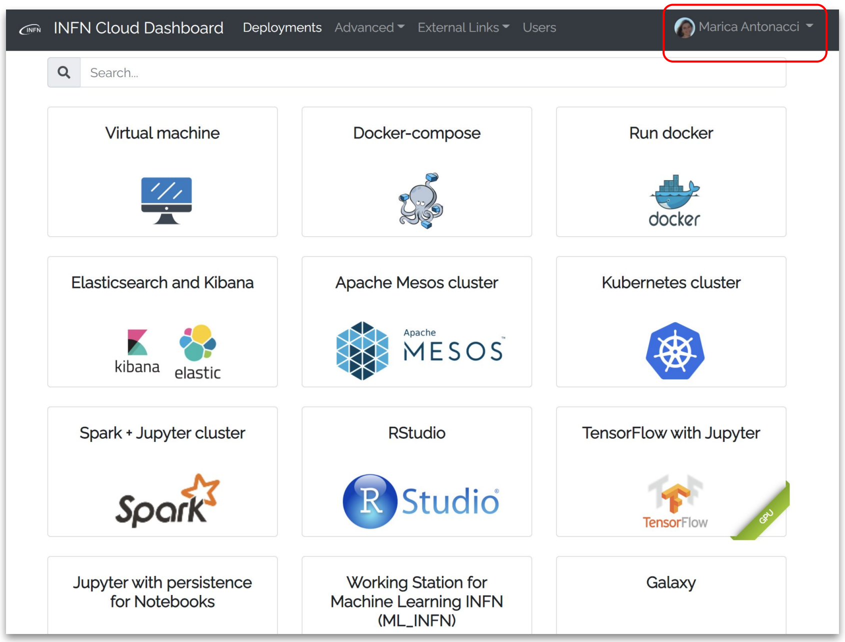
Oops!

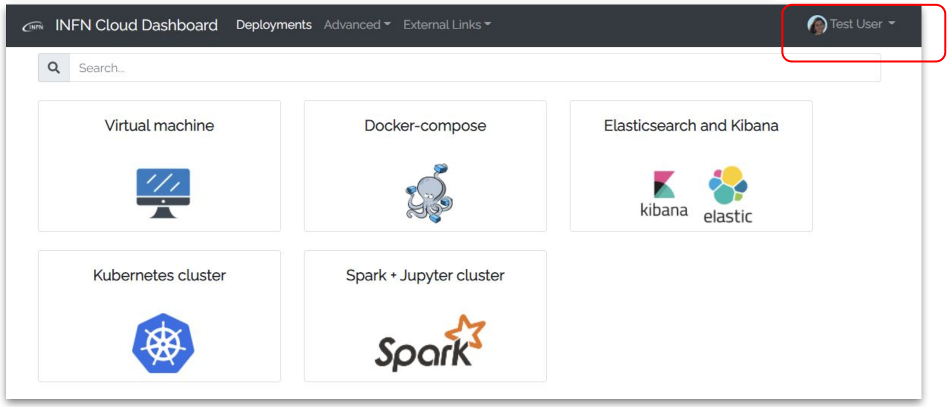
403 Forbidden

Sorry, you are not authorized.

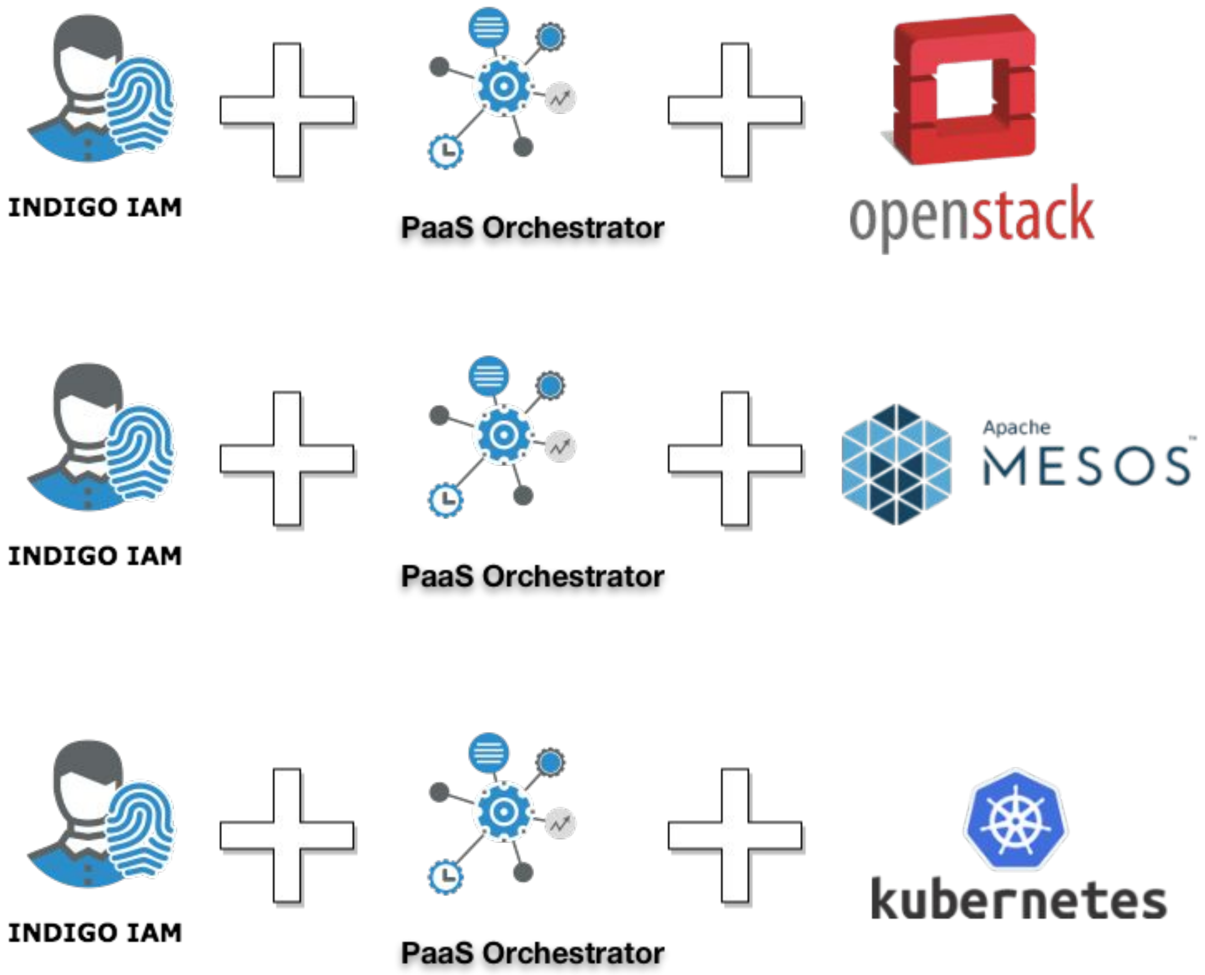
Your identity has been verified successfully, but you are not authorized to access the services.
Please, contact the administrators (infncc-support@lists.infn.it) in order to get proper permissions

The dashboard hides the complexity of the interaction with the PaaS Orchestrator, including the token management.





IAM Integrations at PaaS/IaaS level



PaaS Orchestrator & IAM

The PaaS Orchestrator is built using open-source Java Technologies:

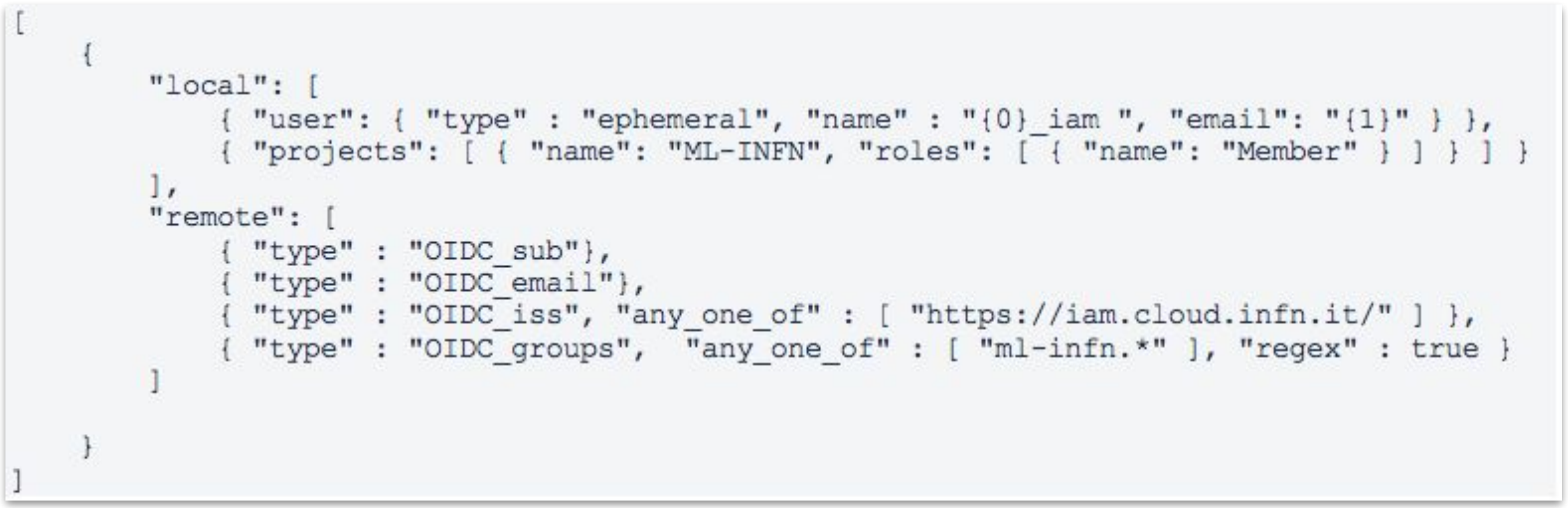
- Spring Boot framework
- Flowable (BPMN2.0 Workflow Manager)

The integration with IAM has been implemented using the OpenID Connect client filter in Spring Security (org.mitre/openid-connect-client)

The Orchestrator exposes REST APIs protected by OAuth2; it valides and introspects the received access token and then exchanges it with a couple of tokens: an access token and a refresh token, that will be used during the deployment workflow.
When interacting with other protected resource servers, on behalf of the user, the Orchestrator asks IAM to issue tokens with specific audiences or downscoped tokens for mininal access.

Openstack-based scenario

- The oidc integration is fully supported by the Identity Service Keystone:
  - run keystone under Apache
  - configure Apache enabling the module mod_auth_openidc
    - if you need to integrate multiple oidc identity providers, you can use esaco
  - configure the federation in keystone
    - create groups and roles
    - define the mapping, i.e. a set of rules that allow to map user claims to keystone users/groups/projects



```
[
    {
        "local": [
            { "user": { "type" : "ephemeral", "name" : "{0}_iam ", "email": "{1}" } },
            { "projects": [ { "name": "ML-INFN", "roles": [ { "name": "Member" } ] } ] }
        ],
        "remote": [
            { "type" : "OIDC_sub"},
            { "type" : "OIDC_email"},
            { "type" : "OIDC_iss", "any_one_of" : [ "https://iam.cloud.infn.it/" ] },
            { "type" : "OIDC_groups",  "any_one_of" : [ "ml-infn.*" ], "regex" : true }
        ]

    }
]
```

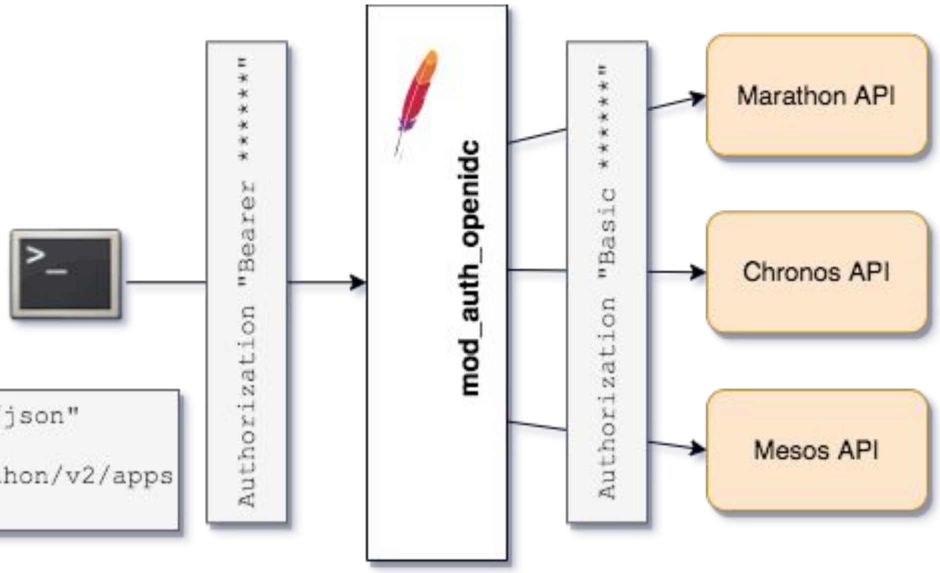
Mesos-based scenario

OIDC Authentication is not supported natively in the Community Edition.

An easy way to enable it is to setup an Apache reverse proxy with the module auth_openidc

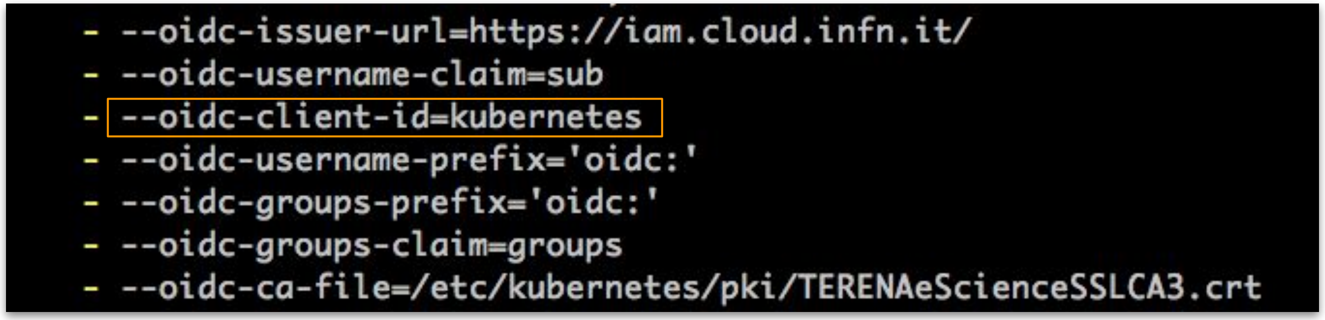
```
 OIDCResponseType                code
 OIDCScope                       "openid profile"
 OIDCProviderMetadataURL         https://iam.cloud.infn.it/.well-known/openid-configuration
 OIDCOAuthVerifyJwksUri          https://iam.cloud.infn.it/jwk
 OIDCClientID                    442....aaece
 OIDCClientSecret                ****
 OIDCProviderTokenEndpointAuth   client_secret_basic


<Location /marathon>
   AuthType oauth20
   Require valid-user
   LogLevel debug
   RequestHeader set Authorization "Basic YWRtaW46OGFXQ0E2a2VVag=="
</Location>

…
ProxyPass /marathon/ http://172.30.0.16:8080/
ProxyPassReverse /marathon/ http://172.30.0.16:8080/
```

```
curl -i -H "Content-Type: application/json"
-H "Authorization: Bearer ***"
-X POST https://mesos.example.it/marathon/v2/apps
-d@app.json
```

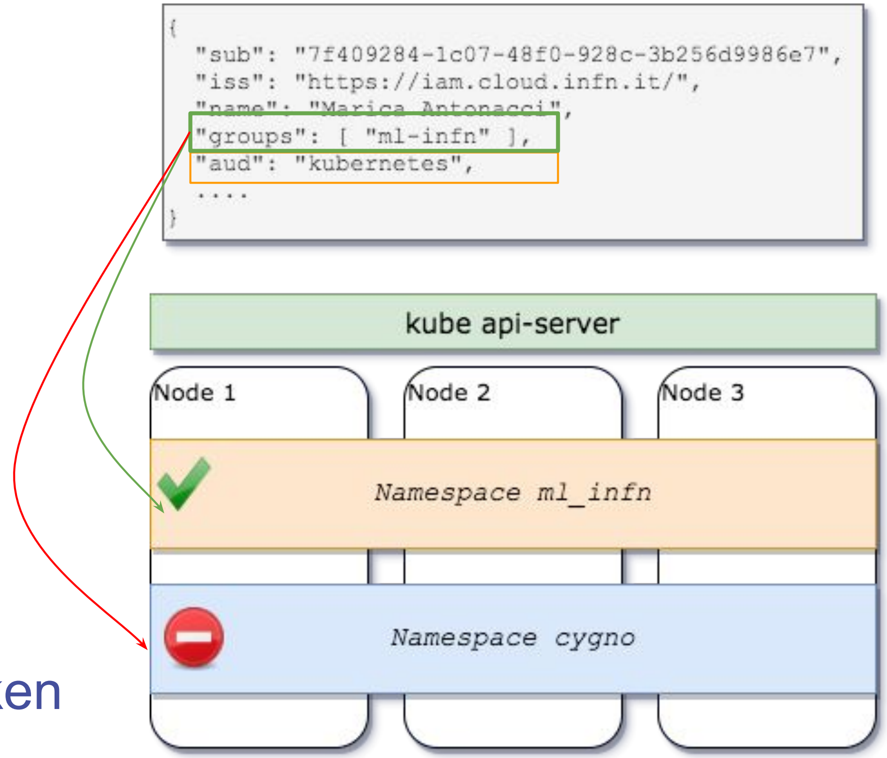


Kubernetes-based scenario

The kube-apiserver can be easily configured to support OpenID Connect Authentication via INDIGO IAM.



```
- --oidc-issuer-url=https://iam.cloud.infn.it/
- --oidc-username-claim=sub
- --oidc-client-id=kubernetes
- --oidc-username-prefix='oidc:'
- --oidc-groups-prefix='oidc:'
- --oidc-groups-claim=groups
- --oidc-ca-file=/etc/kubernetes/pki/TERENAeScienceSSLCA3.crt
```



```
{
  "sub": "7f409284-1c07-48f0-928c-3b256d9986e7",
  "iss": "https://iam.cloud.infn.it/",
  "name": "Marica Antonacci",
  "groups": [ "ml-infn" ],
  "aud": "kubernetes",
  ....
}
```

We have then configured RBAC in order to authorize users of a specific IAM group to access a dedicated namespace in order to isolate their deployments.

In this case the Orchestrator exchanges the user access token with a new token with the requested audience "*kubernetes*"

...and yet another IAM integration

HashiCorp Vault

+

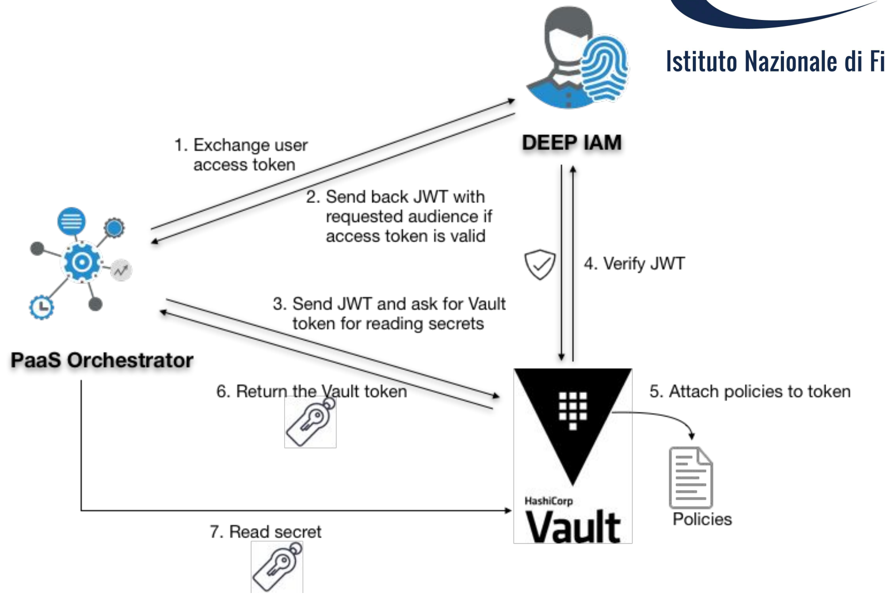
INDIGO IAM

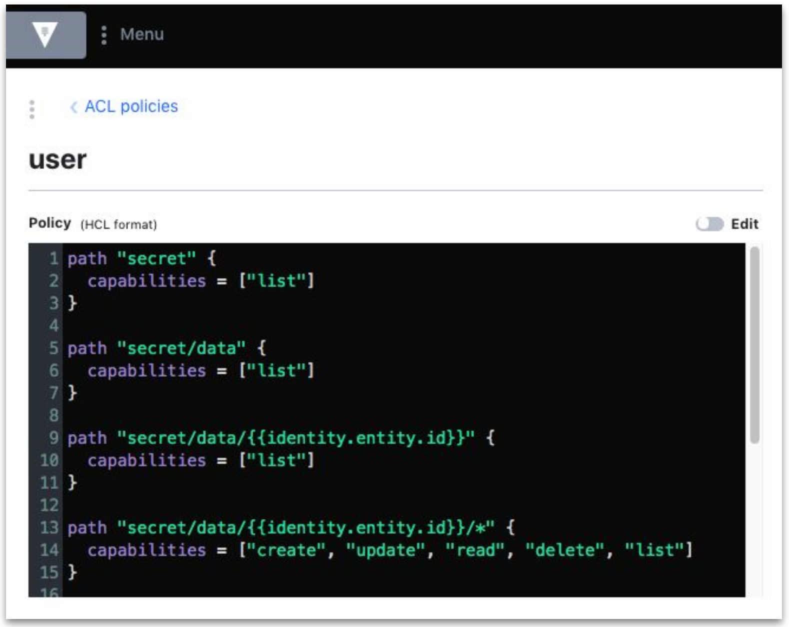
Secrets management

Both the Orchestrator and the Dashboard are integrated with Hashicorp **Vault** (Secrets Manager) to support some functionalities, e.g.

- ssh key pair management
- service sensitive data

The Vault has been integrated with **INFN Cloud IAM** (jwt auth) and proper policies grant read and/or write permissions to specific Vault paths depending on the user claims.


DEEP IAM
1. Exchange user access token
2. Send back JWT with requested audience if access token is valid
4. Verify JWT
3. Send JWT and ask for Vault token for reading secrets
PaaS Orchestrator
6. Return the Vault token
5. Attach policies to token
HashiCorp Vault
Policies
7. Read secret



```
path "secret" {
  capabilities = ["list"]
}

path "secret/data" {
  capabilities = ["list"]
}

path "secret/data/{{identity.entity.id}}" {
  capabilities = ["list"]
}

path "secret/data/{{identity.entity.id}}/*" {
  capabilities = ["create", "update", "read", "delete", "list"]
}
```

Conclusions

- The INFN Cloud AAI is based on an instance of INDIGO IAM configured to easily manage the INFN users community.
- Client applications and services (at all the three levels of the stack, IaaS/PaaS/SaaS) have been integrated with this IAM instance via standard OAuth/OpenID Connect mechanisms.
- We are exploiting, in particular, the following functionalities:
  - Authorization & membership
  - Impersonation (with proper constraints allow services to act on behalf of the user)
- This presentation was mainly focussed on the PaaS/IaaS integration aspects with IAM, but most of the high-level services that can be automatically deployed through the INFN Cloud PaaS (e.g. JupyterHub and the access to the storage) support the IAM authentication.
  - See the talk from D. Spiga & D. Ciangottini (tomorrow at 15.20)

Thank you

for your attention!