Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

terabit

Federation-registry: the renovated Configuration Management Database for dynamic cloud federation

Savarese Giovanni, INFN-BA

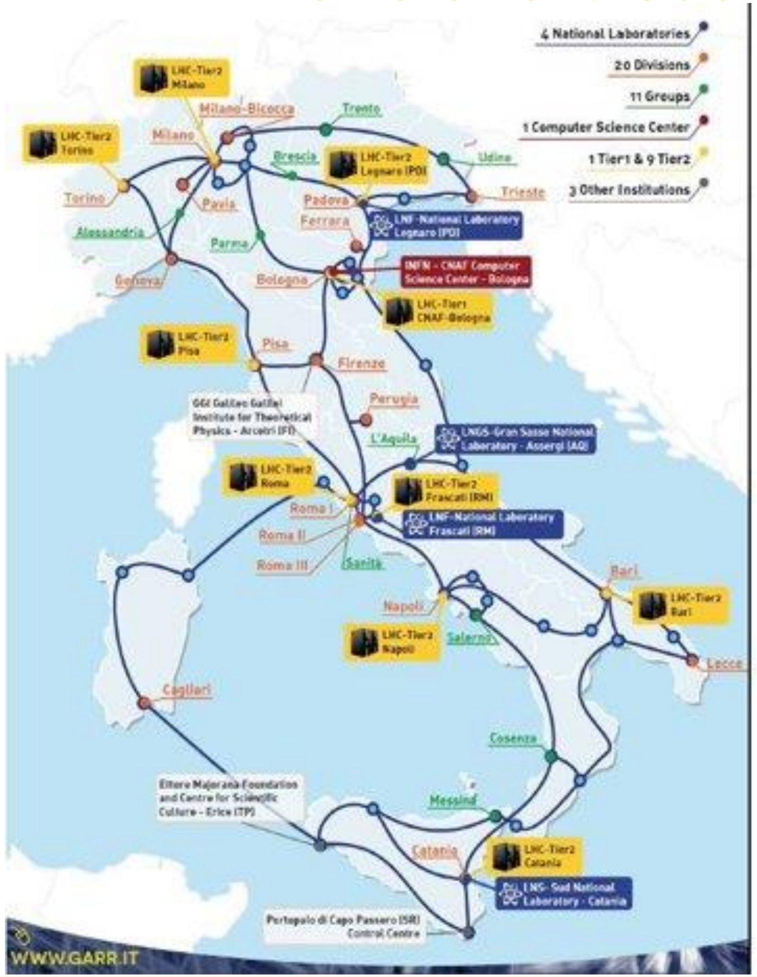
M. Antonacci – INFN-BA

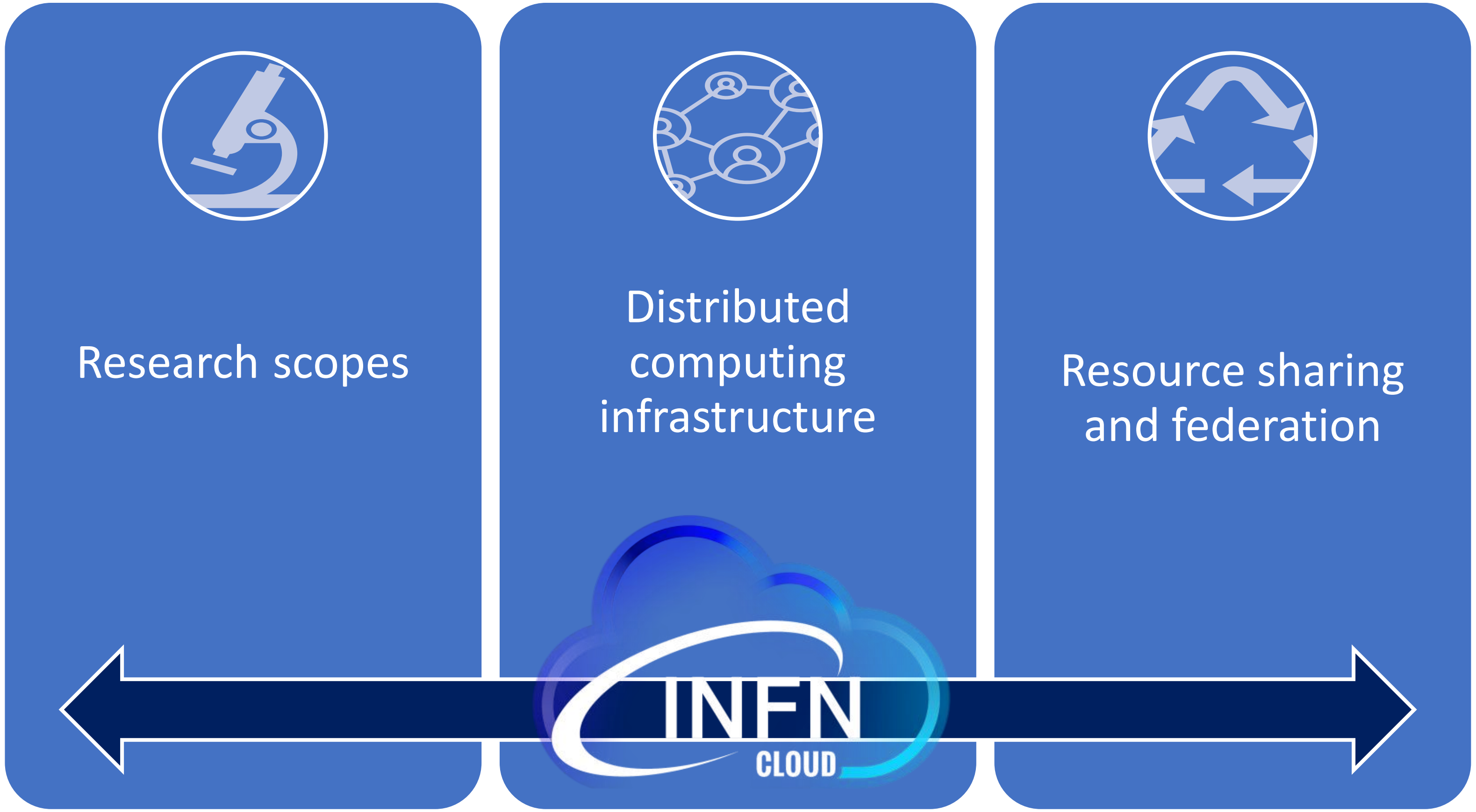
L. Giommi – INFN-CNAF

Taipei (TW), 03.26.2024

Internation Symposium on Grids & Clouds (ISCG) 2024

INFN - Grid and Cloud



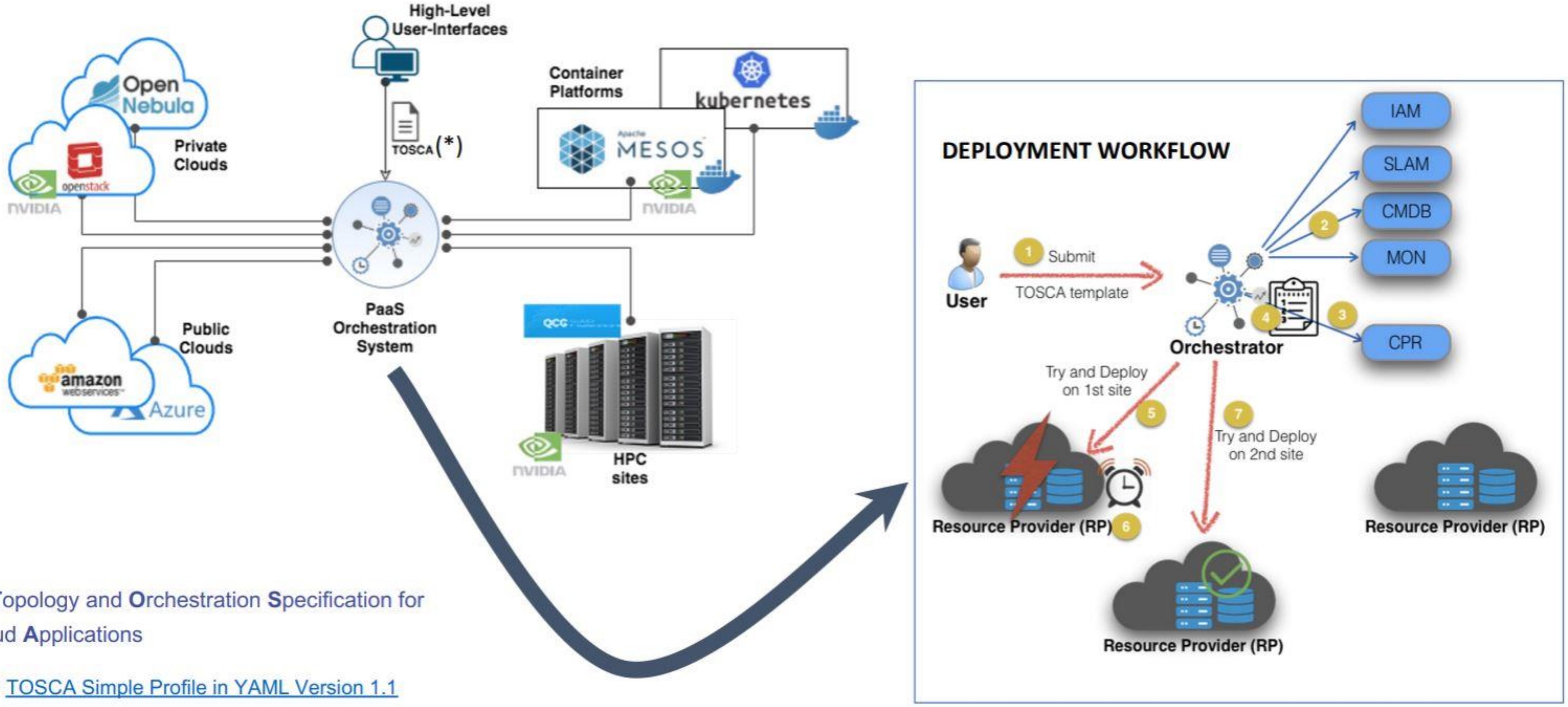


Research scopes

Distributed computing infrastructure

Resource sharing and federation

INFNCloud Orchestrator



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

Ref: TOSCA Simple Profile in YAML Version 1.1

Configuration Management Database (CMDB) and Cloud Info Provider (CIP)

REST API to manage information about federated providers and their services

Java Based (Java 8)

Integration with Indigo-IAM to authenticate and authorize users

Exploits Apache CouchDB

- No schema definition
- Document Storage model
- REST API expecting JSON compliant data

Docker based. Image on Docker Hub

Population script (CIP)

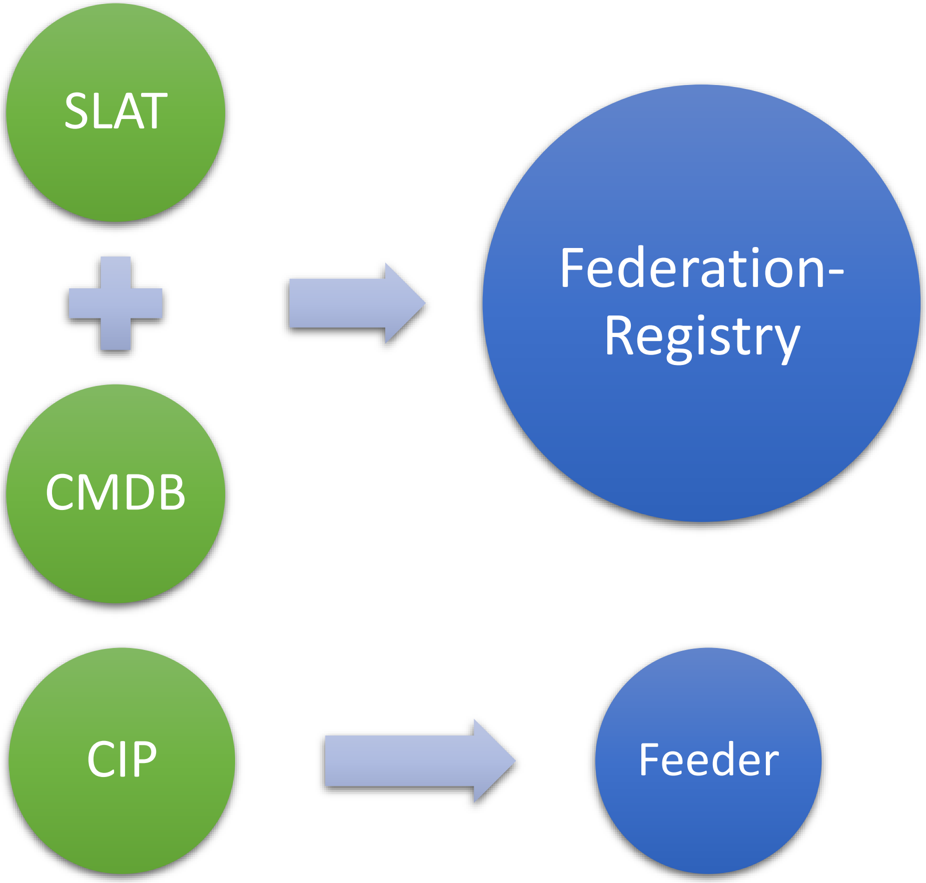
- Python 2.7
- Periodically execute a container with the script
- Based on YAML files

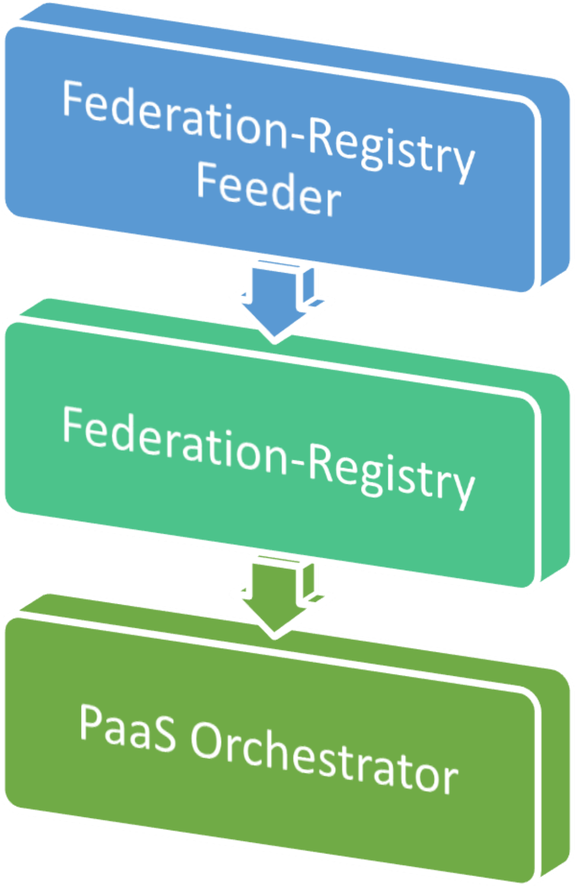
No longer maintained!

Service Level Agreement Tool (SLAT)

- REST API to manage providers' resources usage granted to specific user groups
- Python3 Flask app with SQLAlchemy integration
- Jinja2 templates for the Graphical User Interface
- Integration with Indigo-IAM to authenticate and authorize users
- MySQL database to store Service Level Agreements (SLA)
  - Schema based
  - Relational Database Model
  - Integration through SQLAlchemy library
- Docker based. Image on Docker Hub
- Shared or highly coupled information with CMDB

Federation-Registry replacement strategy





Federation-Registry

REST API to manage federated providers configurations and resources usage grants

- Provider's available services and federated resources
- Quotas assigned to each user group with per user limitations
- Trusted identity providers configurations

Poetry tool to manage package dependencies

Python3 FastAPI app with pydantic integration

Integration with Indigo-IAM to authenticate and authorize users

- Support for any other OIDC based identity providers with the flaat module

Neo4j Graph Database

- No schema definition
- Graph Model
- Integration through neomodel library

Docker based. Image on Docker Hub

Implementation choices (1)

Python

- Requests frequency and response time are neglectable with respect to Orchestrator **deployment** procedure overall time
- Easier to learn and generally more appreciated than Java

FastAPI

- Performance equals to NodeJS and Go
- Easy to use and learn
- Minimize code duplication and requires less code
- Huge documentation and examples
- Open standards for APIs (OpenAPI and JSON Schema)
- Automatic API documentation with possibility to execute query

Neo4j

- No schema definitions. Model flexibility
- Efficient horizontal scaling to handle high-throughput and very large data sets
- More simple queries (cypher language)

Implementation choices (2)

Entity redefinition merging SLA details and providers configuration

- Reduce number of requests executed by the orchestrator
- Consistency with federated providers resources and permissions
- Avoid information spread and subsequent merge

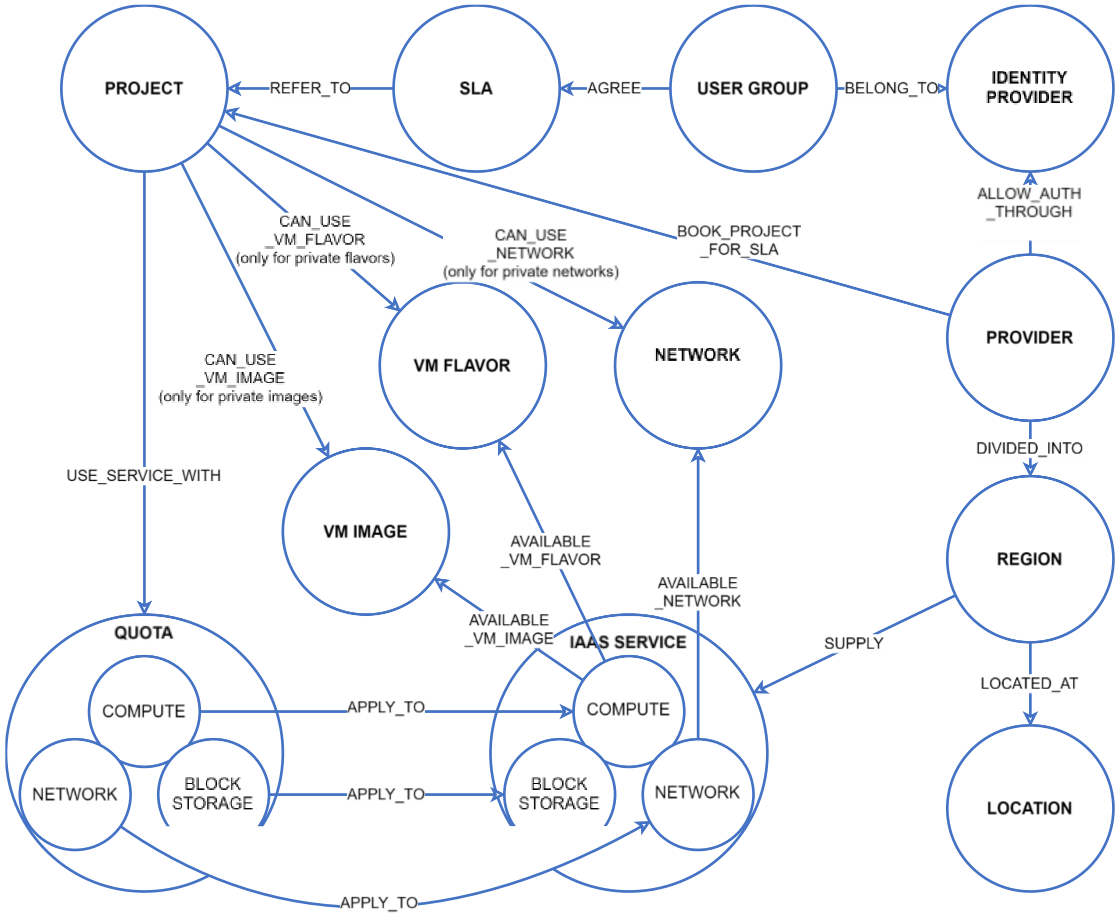
Model definitions through pydantic and neomodel objects

- Add validation check on input values
- Automatic data categorization based on node labels
- Keep the neo4j model flexibility

Each REST request is enveloped in a single database transaction

Data visibility and operations restrictions

- Non authenticated users can read only a subset of the data stored for each entity
- Authentication happens through Bearer Token which is validated by flaat against the trusted identity providers.
- Only authorized users can perform write operations



Federation-Registry-Feeder

Python3 script

Poetry tool to manage package dependencies

Based on YAML files

- Trusted identity providers details (url, group claim…)
- Signed SLAs details (target document, start and end dates)
- Providers configuration details (url, support emails, regions…)
- Federated projects details (preferred default network…)
- Only per user quotas definition (possibility to specify target region)

Federated provider inspection to retrieve, services, quotas and resources

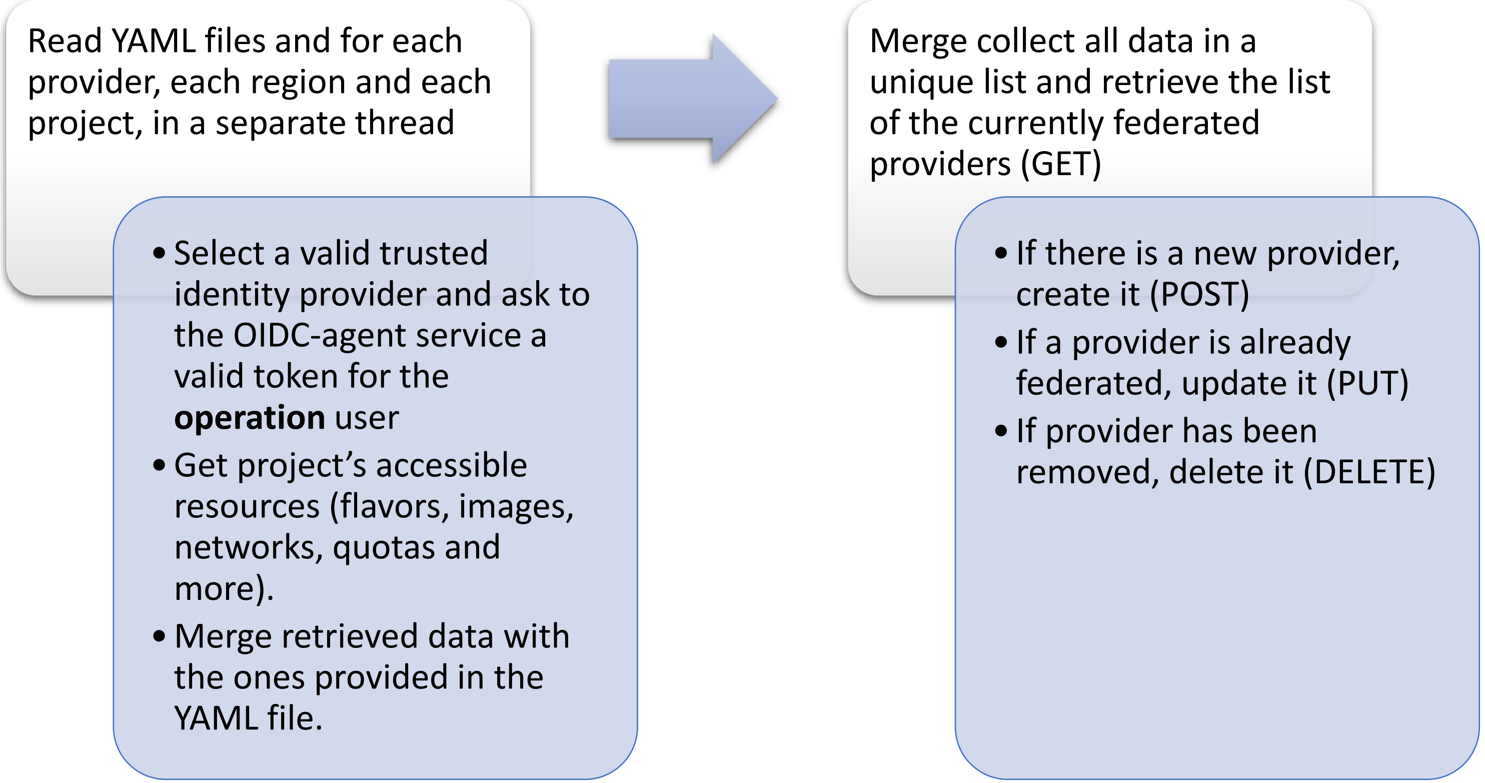
- OIDC-Agent service to generate token
- The **operation** user must have access to the federated projects

Permission to execute read and write operations on Federation-Registry

- OIDC-Agent service to generate token
- The **operation** user must have write access to the Federation-Registry

A cron job periodically starts a docker container executing this script

Flow description

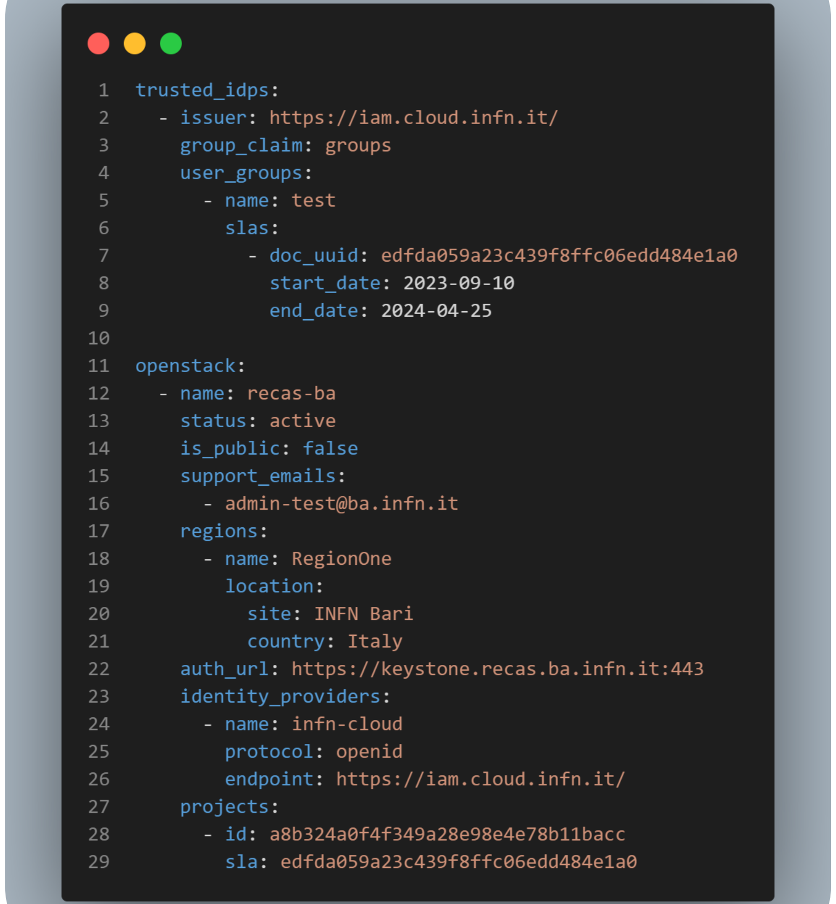


Read YAML files and for each provider, each region and each project, in a separate thread

- Select a valid trusted identity provider and ask to the OIDC-agent service a valid token for the **operation** user
- Get project’s accessible resources (flavors, images, networks, quotas and more).
- Merge retrieved data with the ones provided in the YAML file.

Merge collect all data in a unique list and retrieve the list of the currently federated providers (GET)

- If there is a new provider, create it (POST)
- If a provider is already federated, update it (PUT)
- If provider has been removed, delete it (DELETE)



```yaml
trusted_idps:
  - issuer: https://iam.cloud.infn.it/
    group_claim: groups
    user_groups:
      - name: test
        slas:
          - doc_uuid: edfda059a23c439f8ffc06edd484e1a0
            start_date: 2023-09-10
            end_date: 2024-04-25

openstack:
  - name: recas-ba
    status: active
    is_public: false
    support_emails:
      - admin-test@ba.infn.it
    regions:
      - name: RegionOne
        location:
          site: INFN Bari
          country: Italy
    auth_url: https://keystone.recas.ba.infn.it:443
    identity_providers:
      - name: infn-cloud
        protocol: openid
        endpoint: https://iam.cloud.infn.it/
    projects:
      - id: a8b324a0f4f349a28e98e4e78b11bacc
        sla: edfda059a23c439f8ffc06edd484e1a0
```

Source code, docker images, tests, CI and more

Source code on Github	• https://github.com/indigo-paas/federation-registry • https://github.com/indigo-paas/federation-registry-feeder
Docker images on Docker Hub	• https://hub.docker.com/r/indigopaas/federation-registry • https://hub.docker.com/r/indigopaas/federation-registry-feeder
Test suites and coverage	• Usage of *pytest, pytest-cov, pytest-mock* and *pytest-cases* libraries
Github Actions	• CI to build and push docker images • CI to execute tests and analyze code with SonarCloud
Features available for VSCode users	• Devcontainer support • Launch files

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

terabit

Thank you

Savarese Giovanni, INFN-BA
giovanni.savarese@ba.infn.it