



Finanziato  
dall'Unione europea  
NextGenerationEU



Ministero  
dell'Università  
e della Ricerca



Italiadomani  
PIANO NAZIONALE  
DI RIPRESA E RESILIENZA



# PaaS State of the Art

Luca Giommi, INFN-CNAF

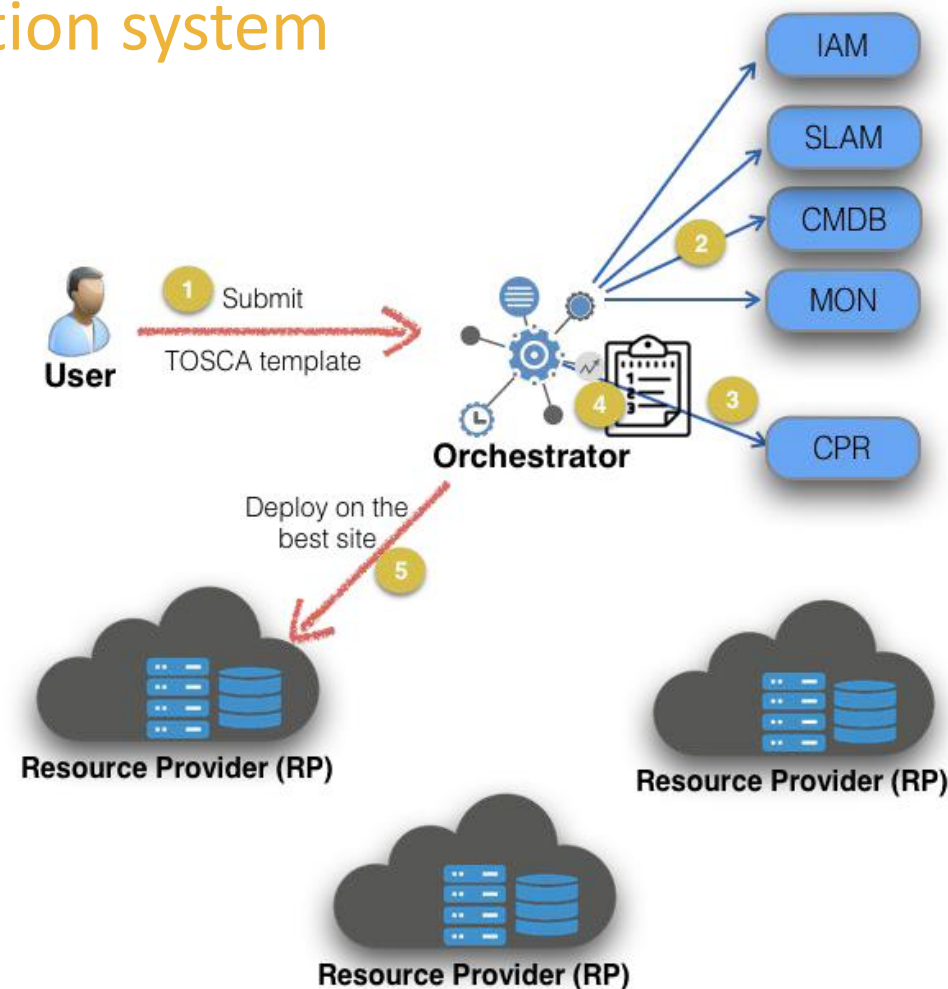
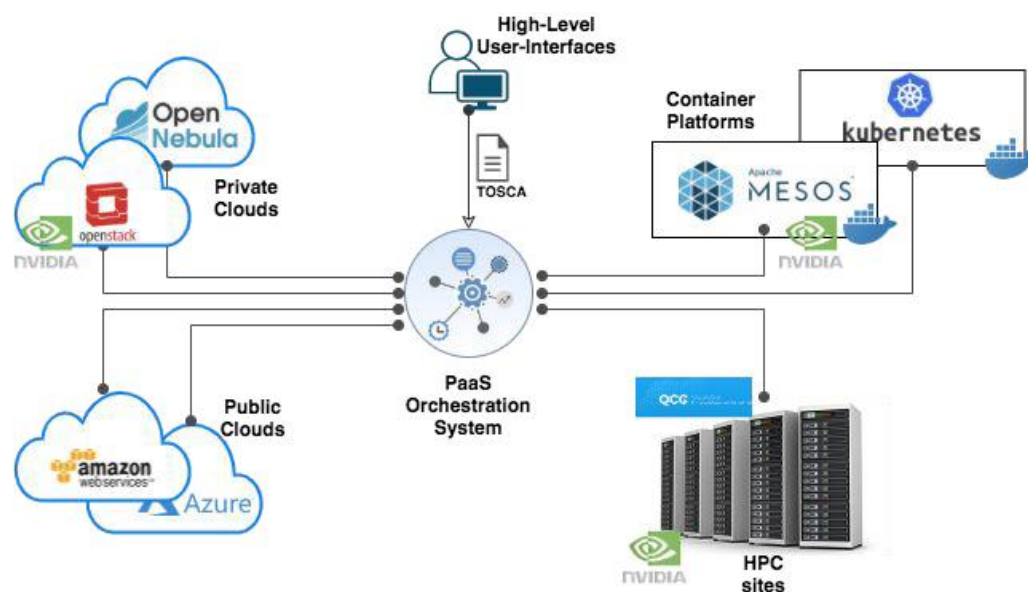
Giovanni Savarese, INFN-BA

Bari, 08.09.2025

Sviluppi PaaS, Bari, 08-10.09.2025



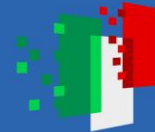
## A bit of history of the INDIGO PaaS Orchestration system



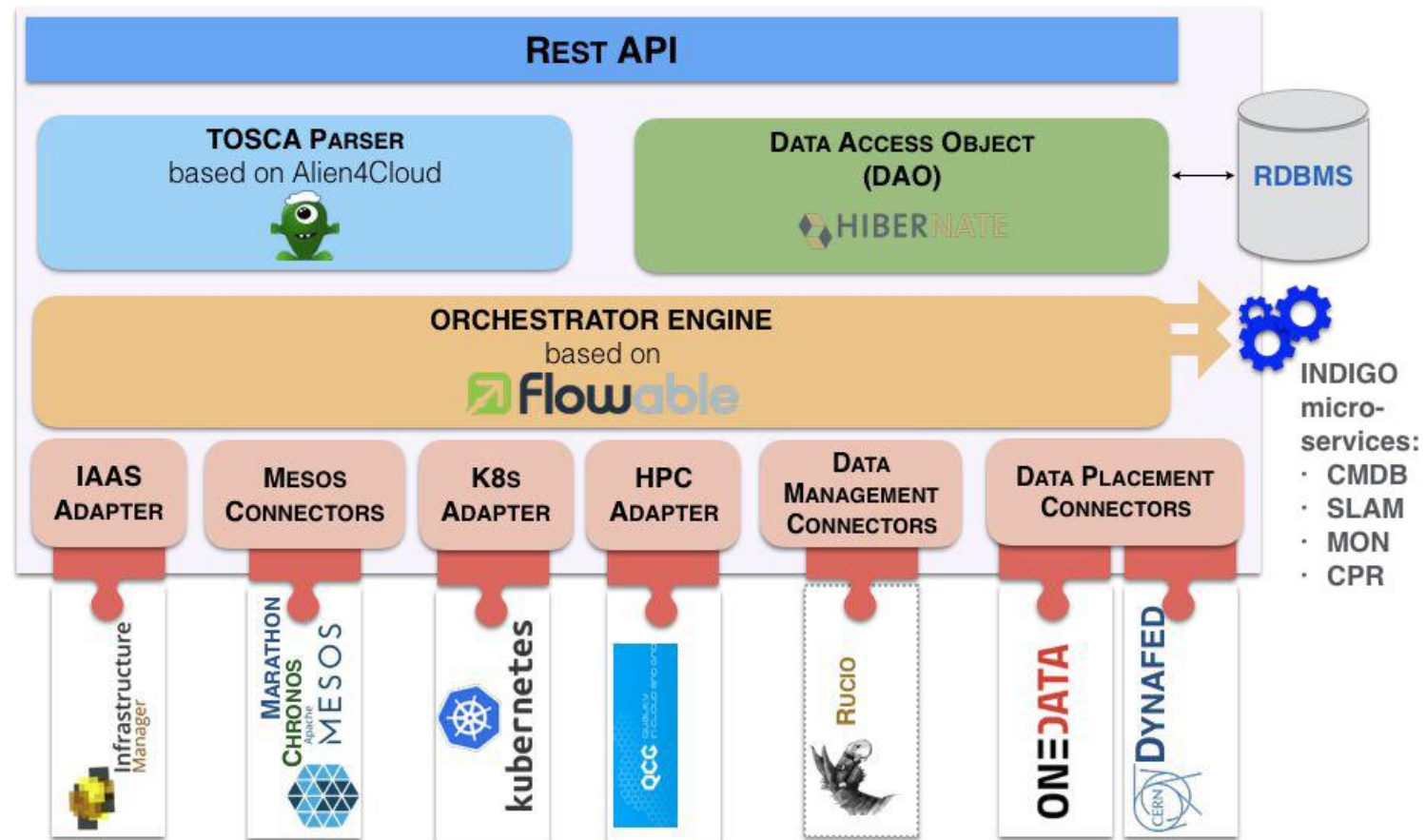
The development started during the European H2020 project "INDIGO-DataCloud" (2015-2017) and continued during the DEEP-Hybrid DataCloud, eXtreme-DataCloud and EOSC-Hub projects

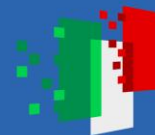
TOSCA: Topology and Orchestration Specification for Cloud Applications

REF: [TOSCA Simple Profile in YAML version 1.0](#)



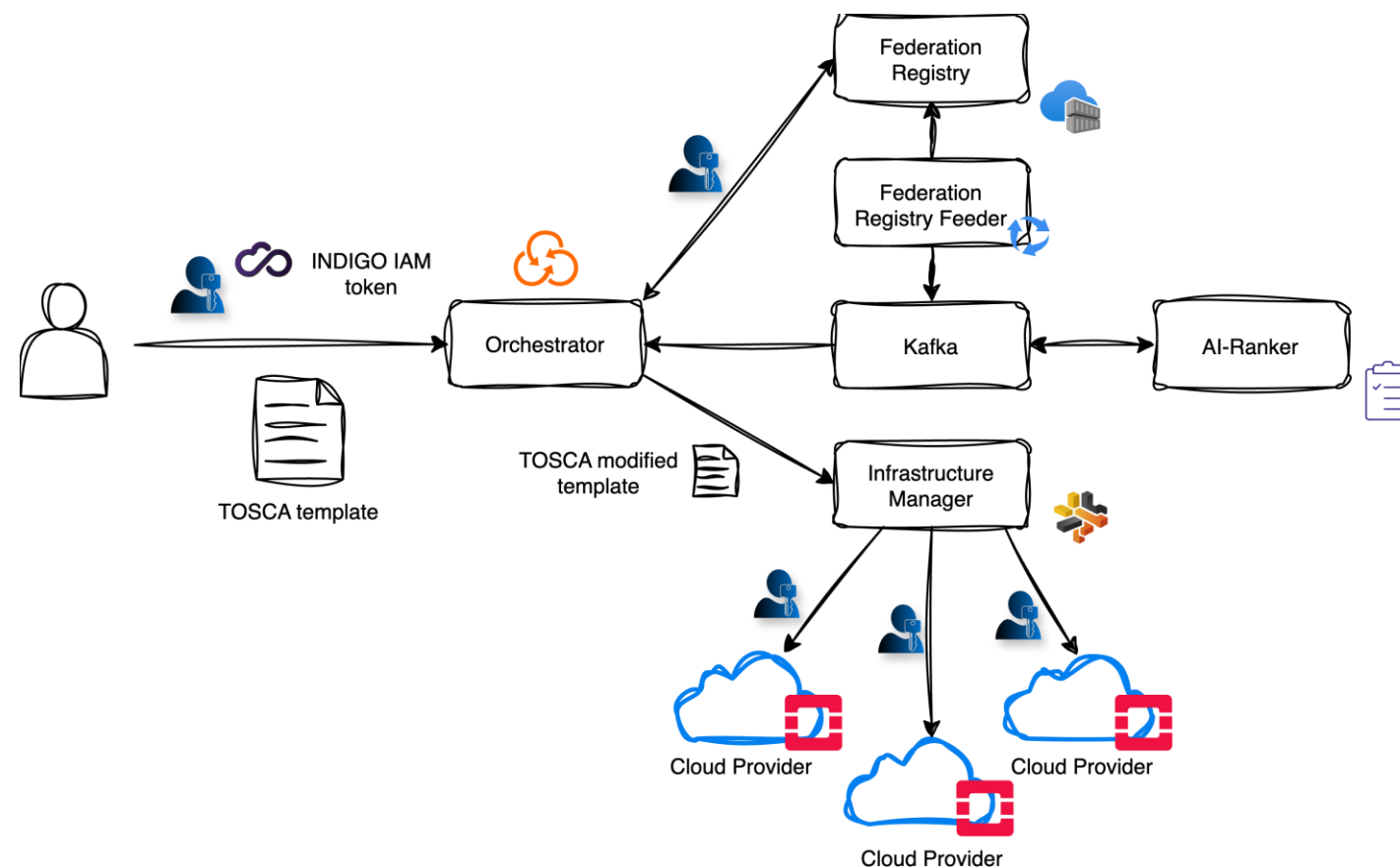
## PaaS Orchestrator high-level architecture





## What is changed in the last two years

- Development of the Federation Registry and Feeder
- Development of the AI-Ranker
- Development of a data streaming platform based on Kafka
- Features added in the Orchestrator
  - Management of creation/deletion of IAM clients and S3 buckets
  - Integration with the Federation Registry
  - Integration with Kafka and AI-Ranker
- Renovation and introduction of many features in the Dashboard







## Data collection and aggregation using Kafka

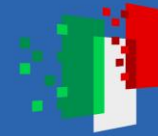
We decided to reorganize the PaaS Orchestration system using messages exchange between components



**Apache Kafka** is a **distributed streaming platform** designed for building real-time data pipelines and streaming applications. It provides a highly scalable, fault-tolerant, and durable mechanism for publishing and subscribing to streams of records.

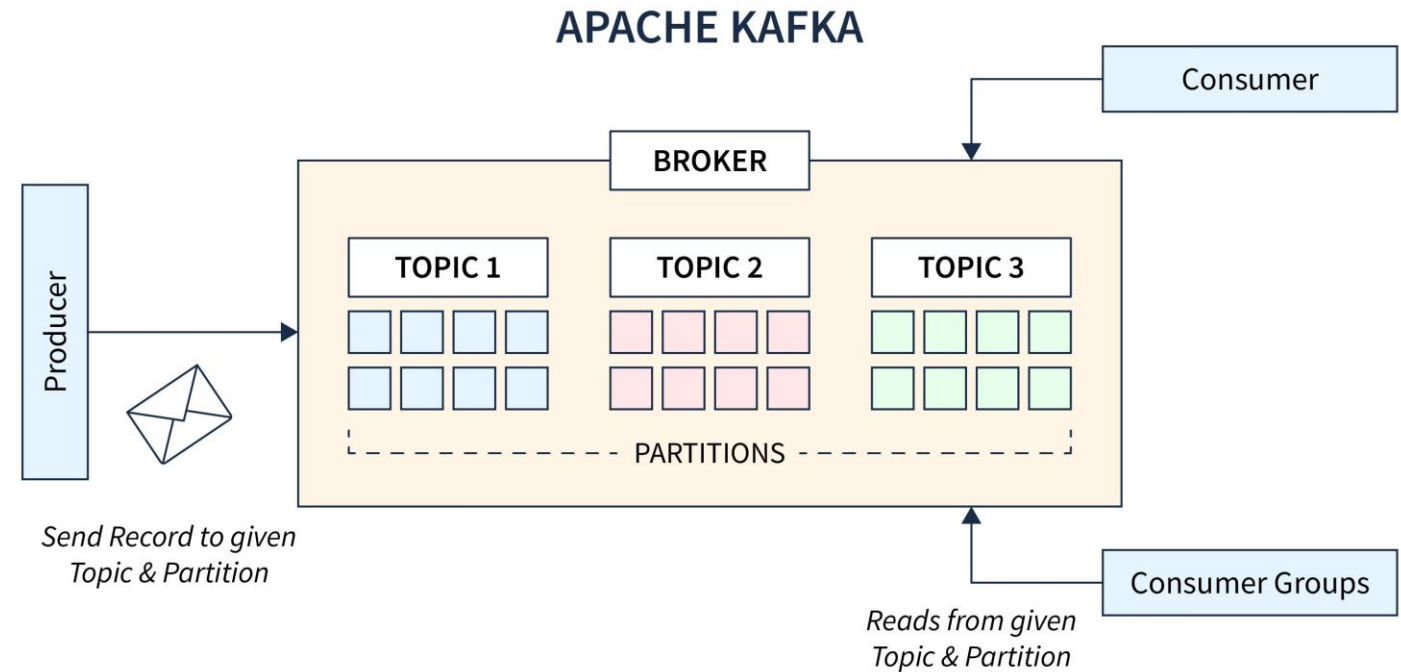
- Data is organized into **topics**, further divided into partitions and replicated across the cluster for fault tolerance
- Actors that write data are called **producers**, while those that read data are called **consumers**
- A **Kafka cluster** can consist of one or more nodes (typically an odd number) to ensure high availability
- Kafka supports high-throughput and low-latency processing, ideal for log aggregation, stream processing, and **event-driven microservices**

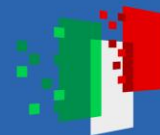
An internal evaluation compared Kafka and RabbitMQ, as both offer core features suitable for our needs. Kafka was ultimately chosen due to the team's existing expertise with this technology.



## Our current solution based on Kafka

- Created many topics for the different components (more in the next slides)
- Cluster in HA with authentication (SSL), authorization (ACL), and encryption and ciphering (TLS/SSL) of messages





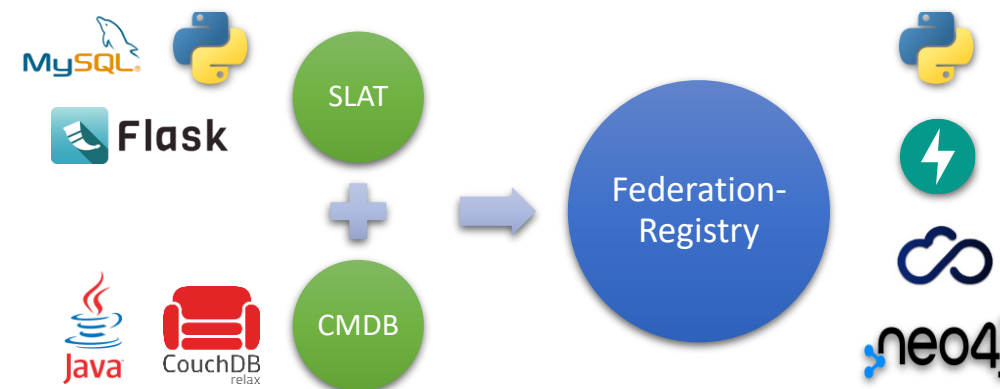
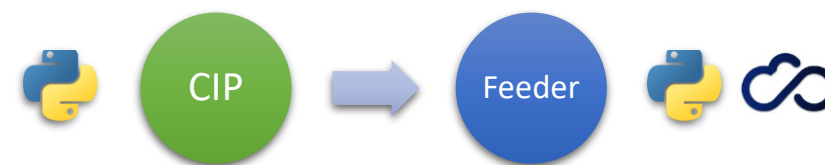
## Federation Registry

### Federation Registry Feeder

- Periodic Python script
- Based on YAML configuration file to connect to federated providers
- Update the Federation Registry with up to date information (flavors, images, networks, quotas and more) retrieved directly from federated providers

### Federation Registry

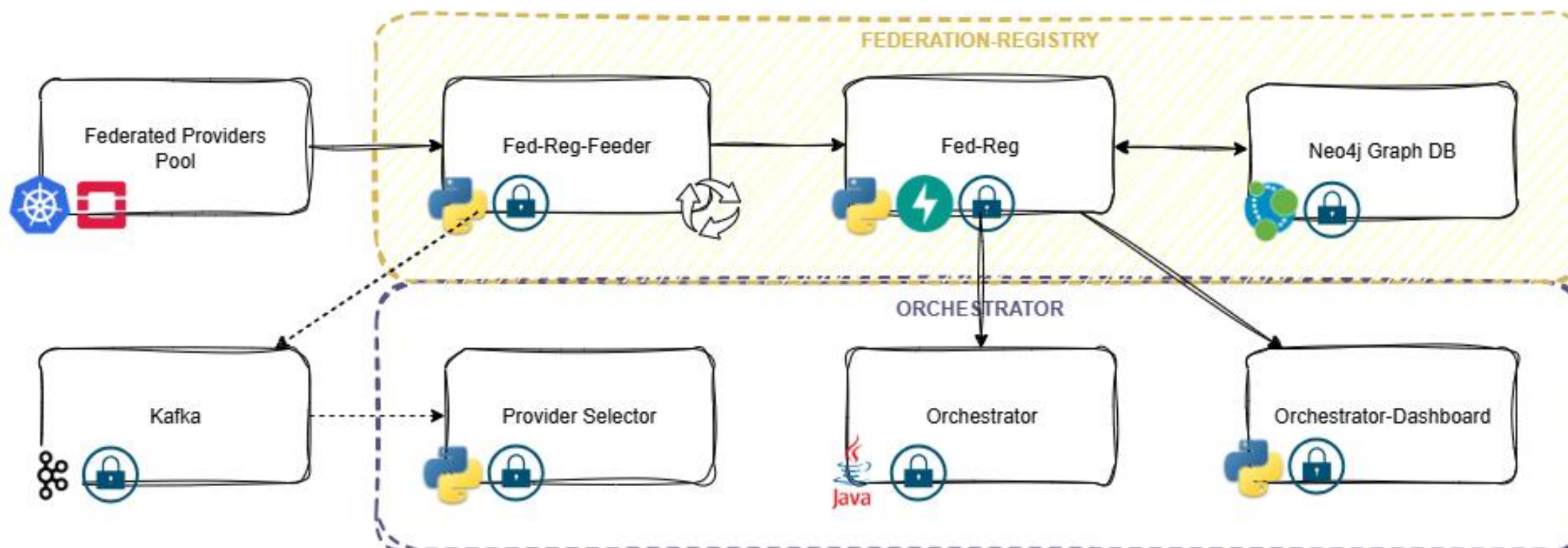
- Micro-service used to store federated provider configurations and SLAs details
- Python REST API based on FastAPI
- Support for OAuth2/OIDC authentication and authorization
- Uses Neo4j as graph database





## Federation-Registry

Micro-service used to store federated provider configurations and SLAs details



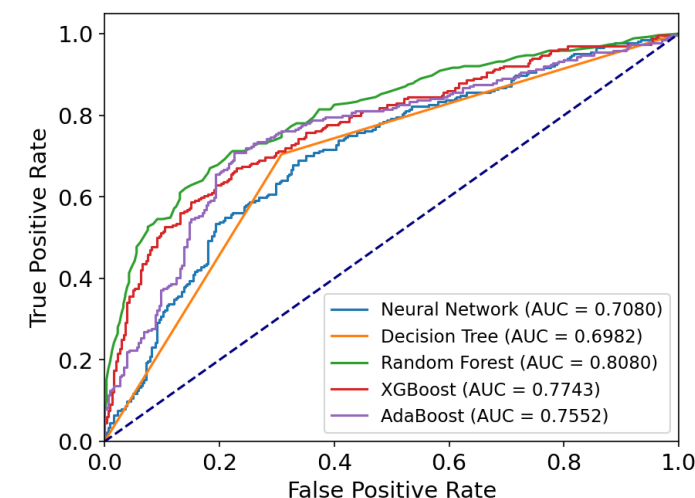
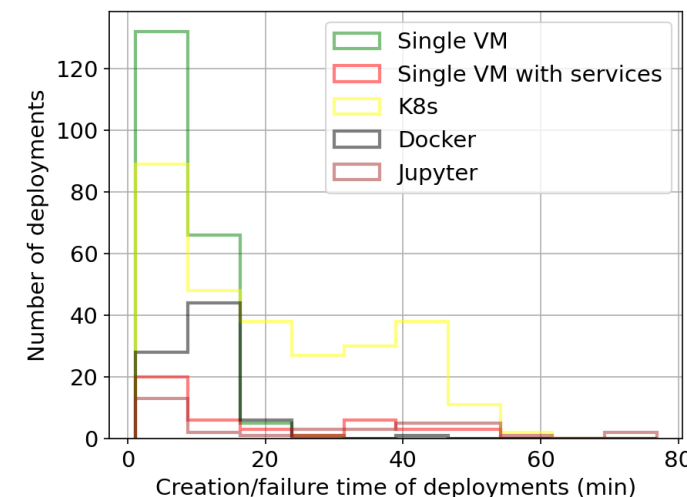




## AI-Ranker: a first study in 2024

**Target:** Improve the Cloud Provider ranking system by identifying and using more appropriate information and metrics through an **AI-based approach**

- 1) Identification of data sources and available information
- 2) Data collection (manual) and dataset creation
  - **6 months of data** used: 08.2023 – 01.2024, **643 entries** (**very few!**)
- 3) Data exploration, data cleaning, data transformation, and feature engineering
  - Categorized the service type of the deployment according to the **complexity**
  - **Reduced the number of features**. Finally used **11**
- 4) Model and training design, and performance evaluation
  - **Two ML models: classification** for success/failure of a deployment, **regression** for creation/failure time of a deployment
  - Defined the training procedure using data of recent and **sliding time windows with fixed size**

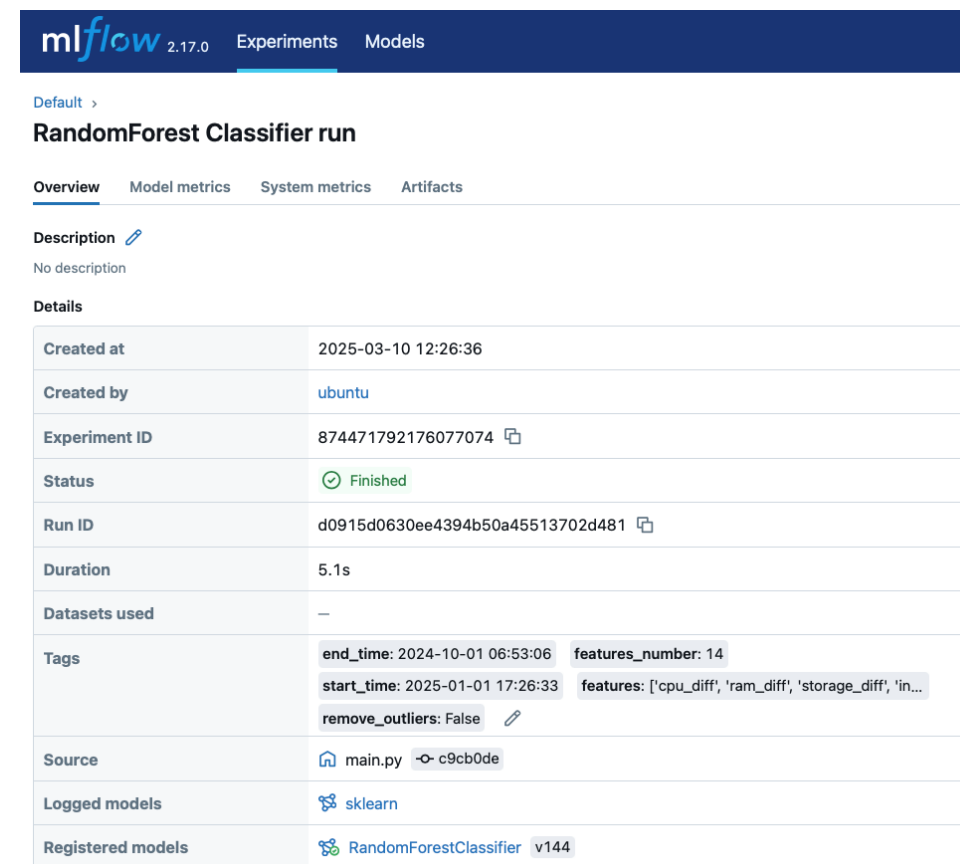




## AI-Ranker

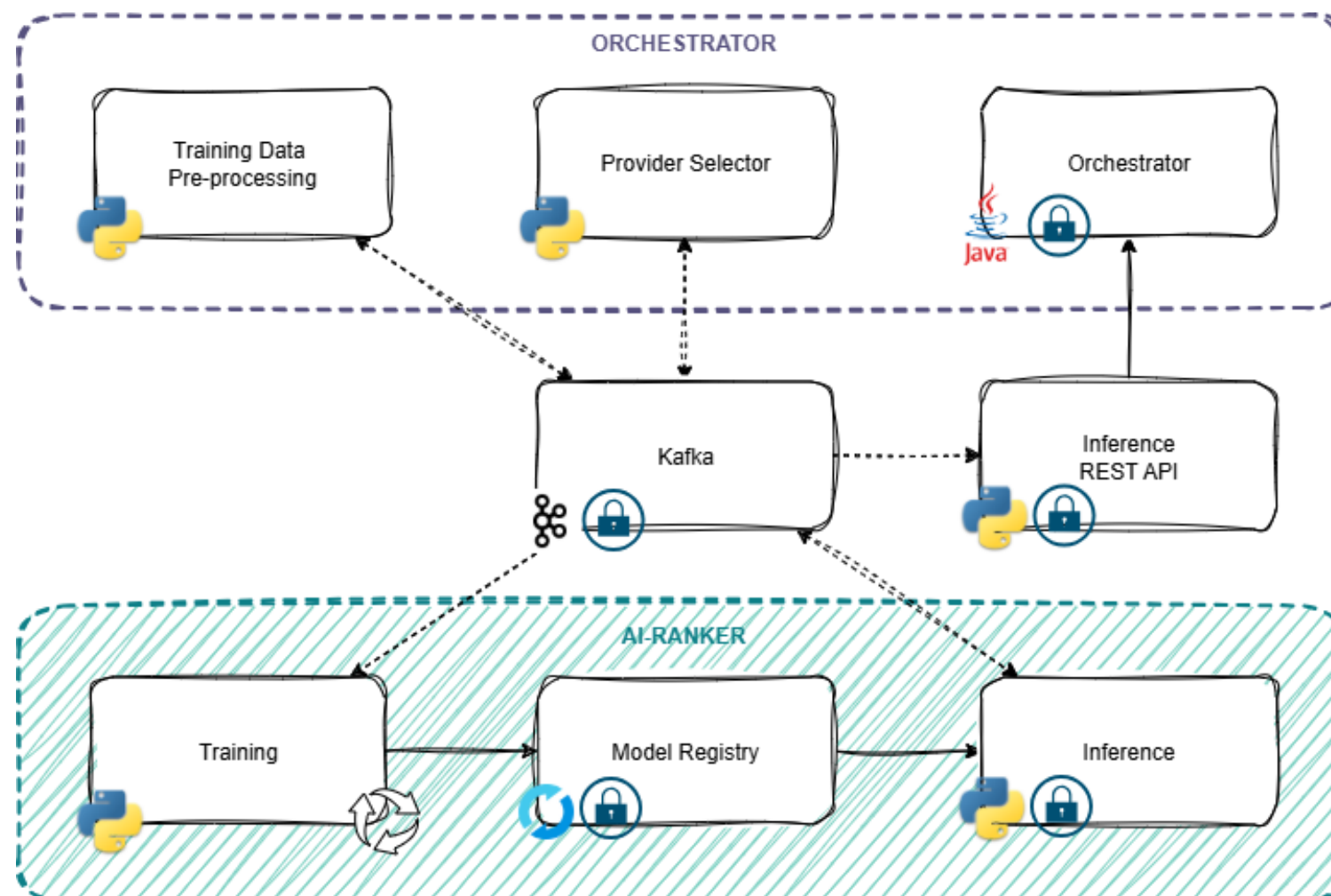


- The AI-Ranker service is composed by three different services deployed as Docker containers
  - **AI-Ranker training**
  - **AI-Ranker inference**
  - **AI-Ranker model registry**
- The AI-Ranker service uses ML models implemented with the *scikit-learn* library. Two types of models used:
  - **Classification** for success/failure of a deployment
  - **Regression** for creation/failure time of a deployment
- The AI-Ranker service uses the **MLflow** framework for managing the ML lifecycle. Mlflow helps us to
  - track experiments (log parameters, metrics, and artifacts from training runs)
  - store models in a standardized format
  - register and version models



The screenshot shows the MLflow web interface for a 'RandomForest Classifier run'. The interface includes tabs for 'Overview', 'Model metrics', 'System metrics', and 'Artifacts'. The 'Overview' tab is selected, displaying a table with the following details:

Created at	2025-03-10 12:26:36
Created by	ubuntu
Experiment ID	874471792176077074
Status	Finished
Run ID	d0915d0630ee4394b50a45513702d481
Duration	5.1s
Datasets used	—
Tags	<div>end_time: 2024-10-01 06:53:06 features_number: 14</div> <div>start_time: 2025-01-01 17:26:33 features: ['cpu_diff', 'ram_diff', 'storage_diff', 'in...]</div> <div>remove_outliers: False</div>
Source	main.py c9cb0de
Logged models	sklearn
Registered models	RandomForestClassifier v144





## Integration of the AI-Ranker with Kafka

### ➤ **Template parser processor**

- Reads from the orchestrator-logs topic
- Writes to the validated-template topic

### ➤ **Provider selector processor**

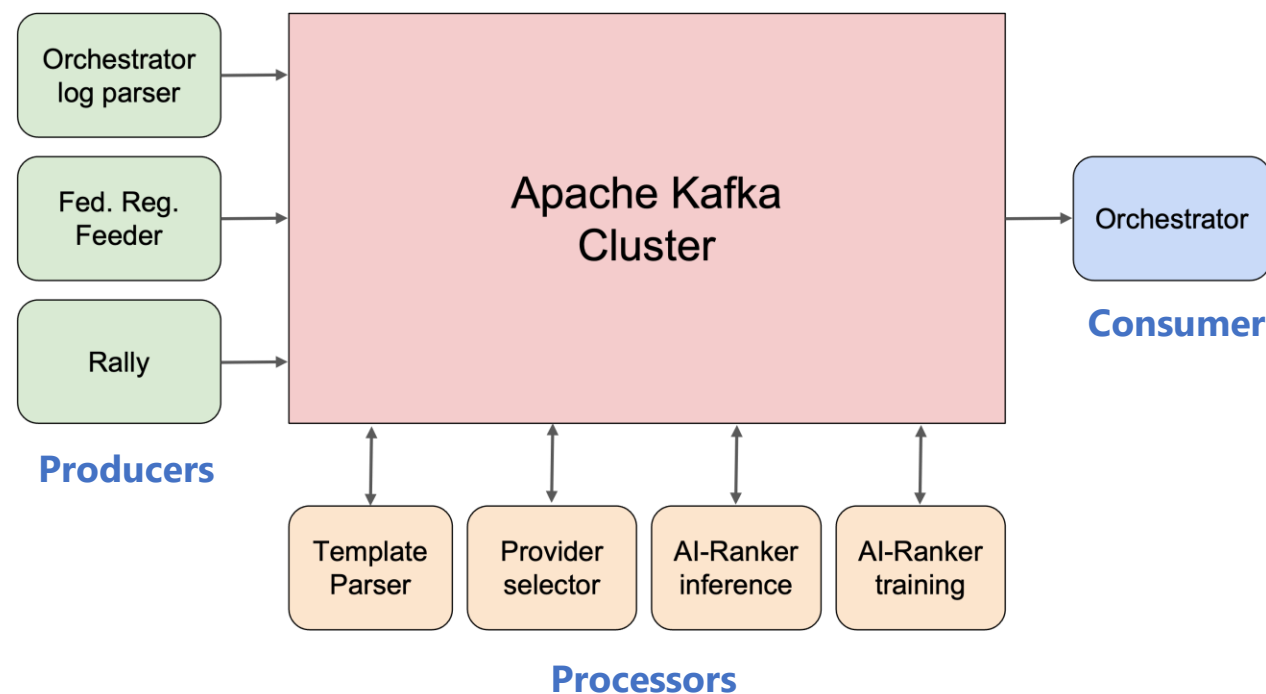
- Reads from the validated-template, federation-registry-feeder, and rally
- Writes to the ai-ranker-inference topic

### ➤ **AI-Ranker inference service processor**

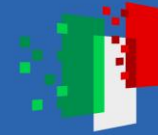
- Reads from the ai-ranker-inference and ai-ranker-training topics
- Writes to the ranked-providers topic

### ➤ **AI-Ranker training data processor**

- Reads from the orchestrator-logs and ai-ranker-inference topic
- Writes to the ai-ranker-training topic

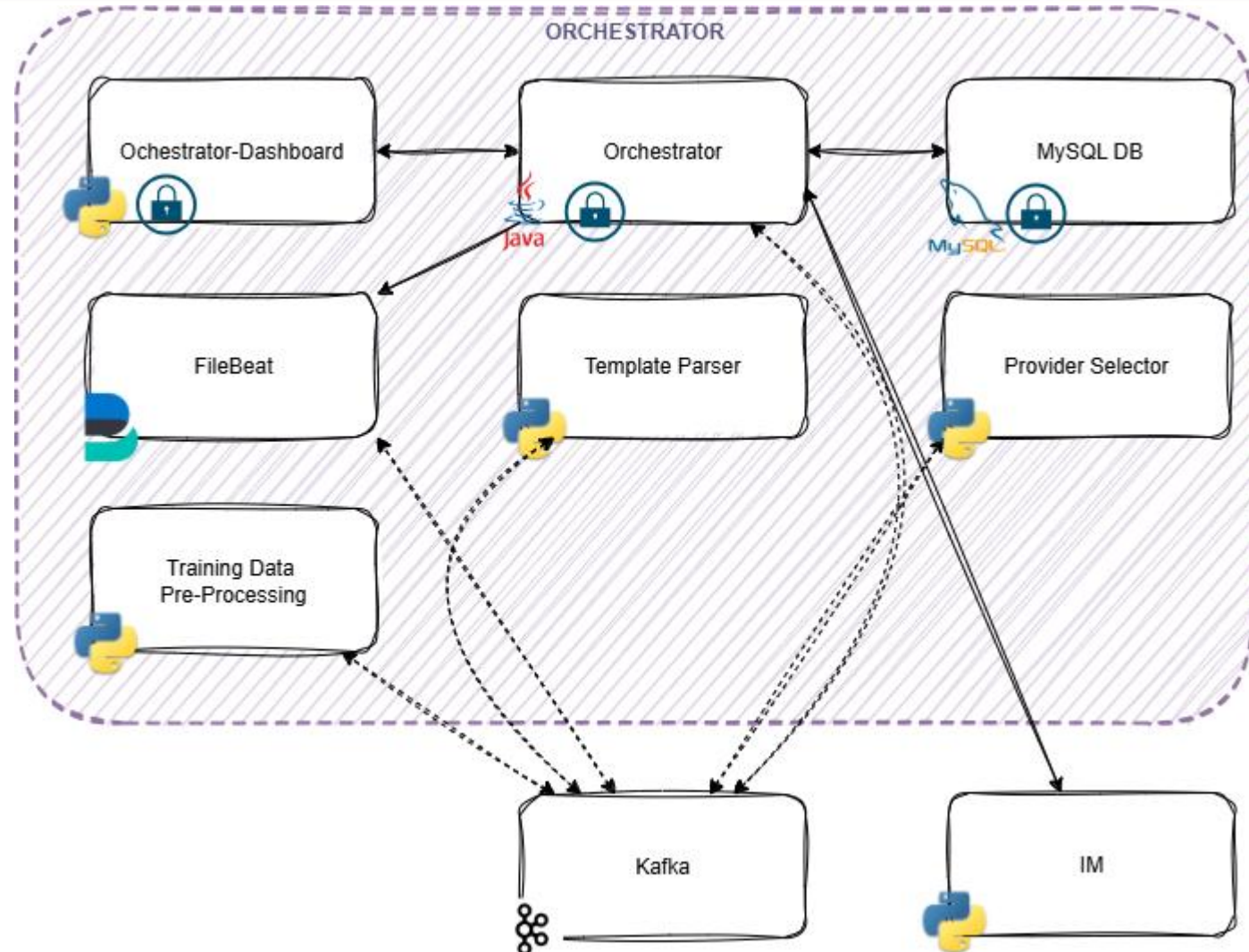






## Orchestrator

The log parser component extracts information about the TOSCA templates, user's inputs, selected provider and status of the deployment







Finanziato  
dall'Unione europea  
NextGenerationEU



Ministero  
dell'Università  
e della Ricerca



Italiadomani  
PIANO NAZIONALE  
DI RIPRESA E RESILIENZA



# The PaaS Orchestrator Dashboard

The screenshot shows the INFN Dashboard. On the left is a dark sidebar with navigation links: HOME, DEPLOYMENTS, ADVANCED, EXTERNAL LINKS, and ADMIN. At the bottom of the sidebar is the user profile 'Luca Giommi' with the role 'admins/catchall'. The main content area has a 'REPORT' section with three status indicators: 'CREATION COMPLETED' (2), 'CREATION IN PROGRESS' (0), and 'CREATION FAILED' (2). Below this is the 'SERVICES' section with a search bar and 'CENTRALISED SERVICES'. Two services are listed: 'INFN Cloud object storage' (described as 'the centrally managed service based on Ceph Rados-Gateway') and 'Notebooks as a Service (Naas)' (described as 'Jupyter Notebooks as a Service'). Each service has a 'GO TO SERVICE' button.

<https://my.cloud.infn.it>

The screenshot shows the 'ON-DEMAND SERVICES' section. The 'Virtual machine' service is highlighted with a description: 'Launch a compute node getting the IP and SSH credentials to access via ssh'. A 'CONFIGURE' button is visible. Below this is the 'Scheduling' section with 'SCHEDULING TYPE' options: 'Automatic' and 'Manual' (selected). A note says 'Select a deployment provider or let the system choose automatically'. The 'PROVIDER' section lists three options: 'RECAS-BARI: org.openstack.nova' (checked), 'BACKBONE - bari: org.openstack.nova', and 'CLOUD-CNAF-T1: org.openstack.nova'. A 'CANCEL' button is present. On the right, a 'Virtual machine' configuration form is shown, labeled 'STEP 3/4'. It has tabs for 'DEPLOYMENT DESCRIPTION (0/50)' and 'CONFIGURATION'. The 'CONFIGURATION' tab is active, showing fields for 'HOSTNAME' (vnode0), 'PORTS' (with an '+ Add rule' button), 'FLAVOR' (a dropdown menu), and 'OPERATING SYSTEM' (a dropdown menu). At the bottom of the form are 'CANCEL' and 'CONTINUE' buttons.



## Features for the users in the PaaS Orchestrator dashboard

- See the **list of created deployments**
- See the **logs** of the deployment configuration
- The **deployments can be managed**, for example by starting and stopping the VMs and managing ports. For deployments of type Kubernetes, nodes can also be added or removed
- in case of a deployment creation failure, a **"Retry"** option will be available in the drop-down menu, enabling users to resubmit the deployment request with the same parameters

### ☰ My deployments

🔄 Refresh

New deployment +

Show 10 ▾ entries

☐ Show deleted deployments Search:

DEPLOYMENT IDENTIFIER	DESCRIPTION	STATUS	CREATION AT	DEPLOYED AT	REGION	Actions
11eeccd1-1dd0-ac5f-8be4-56fce75e0bfa	MLaaS Giommi	CREATE_COMPLETE	2024-02-16 13:41:00	CLOUD- CNAF-T1	tier1	☰ Details ▾

Showing 1 to 1 of 1 entries

Previous 1 Next

### 🖥 Virtual Nodes

← Back 🔄 Refresh + Add Node +

Show 10 ▾ entries

Search:

NAME	HARDWARE CONFIGURATION	NETWORK INTERFACES	STATUS	ACTIONS
k8s-master-server-ffe0239a-1aaf-11f0-a32a-fa163e537d21	cores: 2 ram: 3906.25 MB disk: 18.62645149230957 GB Operating System: debian 12	net_interface_1_ip: 131.154.99.243 net_interface_0_ip: 192.168.12.91	STARTED	⏏ Stop ▶ Start 🗑 Delete
k8s-node-server-fe5ad1aa-1aaf-11f0-a743-fa163e537d21	cores: 2 ram: 3906.25 MB disk: 37.25290298461914 GB Operating System: debian 12	net_interface_0_ip: 192.168.12.221	STARTED	⏏ Stop ▶ Start 🗑 Delete

☰ Details ▾

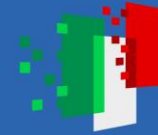
🔄 Retry

🔍 Show template

📄 Log

🖥 Manage Nodes

🗑 Delete



# Features for the admins in the PaaS Orchestrator Dashboard

List and manage other users's deployments: deletion of deployments and full logs visualization

## Deployments full list

Show 10 entries

Refresh Group

Show deleted deployments Search:

DEPLOYMENT IDENTIFIER	DESCRIPTION	STATUS	USER	CREATION TIME	DEPLOYED AT	REGION	GROUP	Actions
11f02cdd-1bbd-2395-8ecb-02424a612ab9	iam-dev	CREATE_COMPLETE	017d3540-a151-464e-bf13-fc7152bb7088	2025-05-09 13:54:00	BACKBONE	bari	admins/training	Details
11f02cdb-bf5c-df70-8ecb-02424a612ab9	iam-dev	CREATE_FAILED	017d3540-a151-464e-bf13-fc7152bb7088	2025-05-09 13:44:00	BACKBONE	bari	admins/training	Details
		CREATE_COMPLETE	564f8033-4025-4fad-889f-83d01fec157c	2025-05-09 08:57:00	BACKBONE	bari	admins/beta-testers	Details

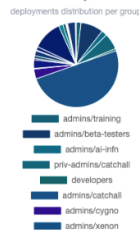
- Show template
- Log
- Manage Ports
- Manage Nodes
- Delete

## Deployments Overview

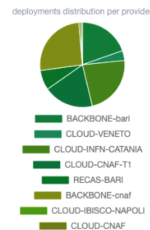
Deployments status



Groups



Providers



## Templates Usage

Show 10 entries

Search:

TEMPLATE NAME	INSTANCES
single-vm/single_vm.yaml	91
single-vm/single_vm_with_volume.yaml	75
kubernetes/k8s_cluster.yaml	46
jupyter/jupyter_vm.yaml	43

See the **Deployments statistics** section to visualize the number of deployments per type, user group, and provider



## Integration of the Orchestrator Dashboard with RUCIO

Implementation of scenario 0: the user can choose the provider to submit the deployment depending on which RSE has a copy of the dataset (storage-binded)

### Scheduling

STEP 2/4

**SCHEDULING TYPE**

☐ Automatic

☒ Manual

Select a deployment provider or let the system choose automatically

☒ Storage binded (Rucio)

**FILE NAME**

test\_TPC

✔ RSE found! Choose one provider from the list above

**PROVIDER**

CLOUD-CNAF-T1: org.openstack.nova

List of allowed providers

CANCEL

← Back

CONTINUE →



## Deploy a TOSCA-based service with IM using a Kubernetes provider

- We did tests with TOSCA templates to create deployments with IM using a Kubernetes provider
  - We enabled the access to the Kubernetes cluster through IAM tokens
  - We deployed a containerized application as a POD on the Kubernetes cluster
  - We successfully tested the use of storage class, PVC, and access point (nodeport and ingress)
  
- Now we are working on the IM connector that the Orchestrator will contact to create deployments with IM using a Kubernetes provider





## Portfolio of services

- Notebook as a Service
- INFN Cloud Registry (Harbor)
- INFN Cloud object storage (RGW)
- INFN Cloud monitoring (Grafana)
- Healthchecks
- Status

SaaS



- Virtual Machine
- Docker Compose
- Run Docker
- Kubernetes cluster (w/o Kafka or Interlink)
- Spark + Jupyter cluster
- HTCondor (mini or cluster)
- Jupyter (w/o Matlab) with persistence
- INDIGO IAM as a Service
- Elasticsearch & Kibana
- Sync & Share
- ML-INFN working station
- CYGNO working station

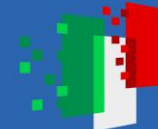
PaaS



- Start & Stop
- Hostname choice
- Open ports

IaaS





## PaaS services (1)

All PaaS services are defined using an **Infrastructure as Code** paradigm, based on a procedural paradigm that aims to reduce manual processes and increase flexibility and portability across environments, via a combination of:

- **TOSCA** (Topology and Orchestration Specification for Cloud Applications) templates, to model an application stack
- **Ansible** roles, to manage the automated configuration of virtual environments
- **Docker** containers, to encapsulate high-level application software and runtime
- **Helm** charts, to manage the deployment of an application in Kubernetes clusters

```
node_templates:
  ml_install:
    type: tosca.nodes.DODAS.single-node-jupyterhub
    properties:
      contact_email: { get_input: contact_email }
      iam_url: { get_input: iam_url }
      iam_subject: { get_input: iam_subject }
      iam_groups: { get_input: iam_groups }
      iam_admin_groups: { get_input: iam_admin_groups }
      monitoring: { get_input: enable_monitoring }
      jupyter_hub_image: dodasts/snj-base-jhub:v1.1.1-snj
      jupyter_images: { get_input: jupyter_images }
      jupyterlab_collaborative: { get_input: jupyterlab_collaborative }
      jupyter_post_start_cmd: "/usr/local/share/dodasts/script/post_script.sh"
      jupyterlab_collaborative_image:
        { get_input: jupyterlab_collaborative_image }
      dns_name: { concat: [get_attribute: [HOST, public_address, 0],
      cert_manager_type: { get_input: certificate_type }
    requirements:
      - host: vm_server
```

**TOSCA**

```
artifacts:
  ml_role:
    file: git+https://github.com/DODAS-TS/ansible-role-jupyterhub-env,v2.4.1
    type: tosca.artifacts.AnsibleGalaxy.role
```

```
- name: prepare compose file
  ansible.builtin.template:
    src: jupyter_hub-compose.j2
    dest: /usr/local/share/dodasts/jupyterhub/compose.yaml
  vars:
    iam_client_id: "{{ iam_response.json.client_id }}"
    iam_client_secret: "{{ iam_response.json.client_secret }}"
  when: cert_manager_type != "self-signed"
```

**Ansible**

```
- name: Run Jupyter Hub
  ansible.builtin.shell:
    cmd: docker-compose up -d
    chdir: /usr/local/share/dodasts/jupyterhub
  when: (run_jupyter | bool)
```



## PaaS services (2)

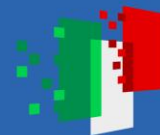
- TOSCA templates has the reference to a specific version of the TOSCA types used where we define new custom types and the types we have extended from the normative types
- Known critical aspects
  - Maintenance/updates of currently supported PaaS services is difficult, in particular for some of them ([TOSCA services inventory](#))
  - Some services requires renovation (e.g. IAMaaS)
  - Addition of new services
  - Ansible version used by the IM to deploy the PaaS services is old (2.10.7)

Version	Support	End Of Life	Control Node Python	Target Python / PowerShell
2.10	GA: 13 Aug 2020 Critical: 26 Apr 2021 Security: 08 Nov 2021	EOL 23 May 2022	Python 2.7 Python 3.5 - 3.9	Python 2.6 - 2.7 Python 3.5 - 3.9 PowerShell 3 - 5.1
2.18	GA: 04 Nov 2024 Critical: 19 May 2025 Security: 03 Nov 2025	May 2026	Python 3.11 - 3.13	Python 3.8 - 3.13 PowerShell 5.1



## DevOps Procedure

- Develop code and write related tests
- Run service and tests on your local machine
- Write and build the docker image on your local machine
- Write and configure Jenkins pipelines
- Write and test a docker-compose example on your local machine
  - *(As soon as ArgoCD will be available replace with)* Convert the docker-compose in kubernetes configurations example ([kompose](#))
- Write the ansible role and playbook based on the docker-compose example
  - *(As soon as ArgoCD will be available replace with)* Write ArgoCD YAML configurations based on the kubernetes configurations example
- Deploy in pre-production with integration tests and debugging (unexpected crashes, wrong behaviors...)
- Deploy in production



## Code repositories management

- Github (INFN-Datacloud) [Github Organization INFN-DataCloud](#)
  - Libraries and main code (fed-mgr, fed-reg, ai-ranker, orchestrator, ...)
- INFN Baltig ([INFN Cloud group](#))
  - INFN Cloud specific configurations (indigopass-deploy, TOSCA templates, ...)
- Development/Collaboration common strategies
  - No forks - multiple branches (as much as possible: one task=one branch)
  - Branch **main** protected and PR with at least one reviewer
  - README.md
    - Description on how to install and start the service, env variables, requirements, ...
  - Apache v2.0 LICENSE





## CI/CD, code analysis and security

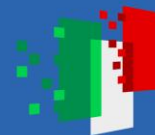
- [Jenkins](#) pipelines
  - Code test and analysis
  - Docker images creation: Upload on both [INFN Harbor](#) and [DockerHub](#)
  - [Shared libraries](#) (repo with shared groovy functions used by jenkins pipelines)
  - [Tutorial Jenkins](#) and [Slides](#)
- [SonarCloud](#)
  - Analyzes code and shows coverage details
  - Explicit projects addition
- [Snyk](#) and Github code scanning
  - Checks code security (libraries deprecation or security problems)
  - Explicit projects addition
- [GitGuardian](#) **DO NOT PUBLISH UNCRYPTED PASSWORDS AND SECRETS**
  - Check password and secrets exposure
  - Explicit projects addition



**GitGuardian**



**Jenkins**



## Repository tools and others

- Useful tools (and VSCode extensions) to develop with python
  - [Ruff](#): code linting and formatting
  - [Poetry](#): dependency management
- Useful tools for commit management
  - [Pre-commit](#): automatically validate custom rules on staged code before committing
- [INFN-CNAF Nexus](#)
  - Service hosted by INFN storing RPM and other compiled packages
- [Bitwarden](#)
  - Service hosted by INFN with shared secrets and passwords



## Documentation and tasks management

- Documentation links
  - [WP5 - INFN Confluence](#)
    - Software and service instances inventory
    - WP5 meetings and lessons' notes
  - [PaaS services documentation](#)
    - Architecture and design choices
  - [IaaS/PaaS/SaaS endpoints list](#)
- [INFN Jira Software](#)
  - Tasks list (tasks up to one week, issues, discussions on specific relevant topics)
  - Track events or updates on target issues
  - We will work mainly through these. Keep them up to date
- Communication channels
  - Team **Sviluppi PaaS** in Teams
  - Mailing list [sviluppi-paas@lists.infn.it](mailto:sviluppi-paas@lists.infn.it)



Finanziato  
dall'Unione europea  
NextGenerationEU



Ministero  
dell'Università  
e della Ricerca



Italiadomani  
PIANO NAZIONALE  
DI RIPRESA E RESILIENZA



# Thank you

luca.giommi@cnafe.infn.it,  
giovanni.savarese@ba.infn.it

