



INDIGO PaaS Overview

Marica Antonacci - INFN



eosc-hub.eu



@EOSC_eu

Dissemination level: Public/Confidential *If confidential, please define:*

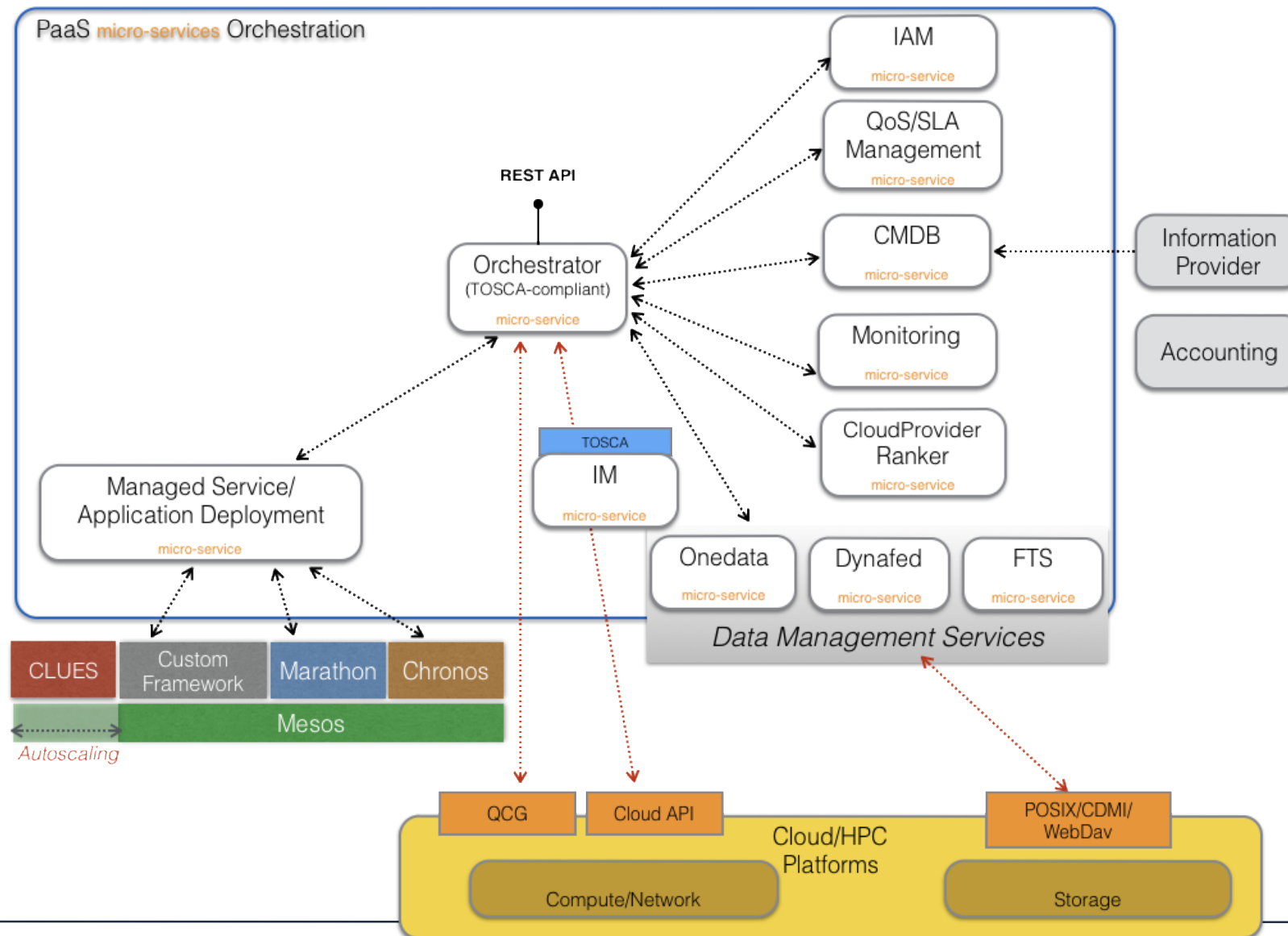
Disclosing Party: (those disclosing confidential information)

Recipient Party: (to whom this information is disclosed, default: project consortium)



The **INDIGO-DC PaaS** is a suite of micro-services that

- allows to coordinate the **provisioning** of virtualized compute and storage resources on different Cloud Management Frameworks (like OpenStack, OpenNebula, AWS, etc.) and the **deployment** of dockerized services and jobs on Mesos clusters;
- features advanced **federation** and **scheduling** capabilities ensuring the transparent access to heterogeneous cloud environments and the selection of the best resource providers based on criteria like user's SLAs, services availability and data location

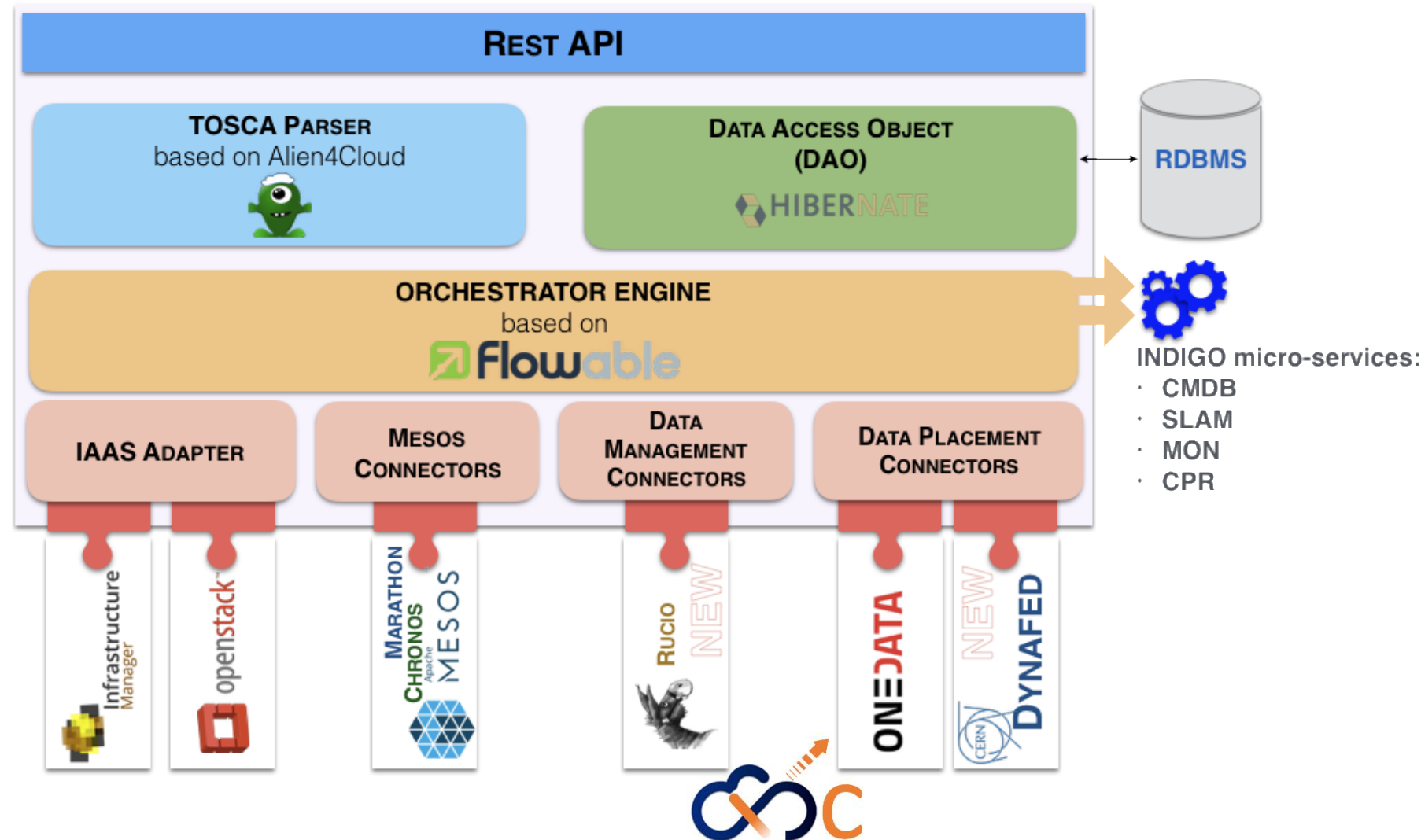


The Orchestrator is built with Java Technologies

- Spring Boot Framework
- Alien4Cloud TOSCA Parser
- Flowable

It can run three main workflows:

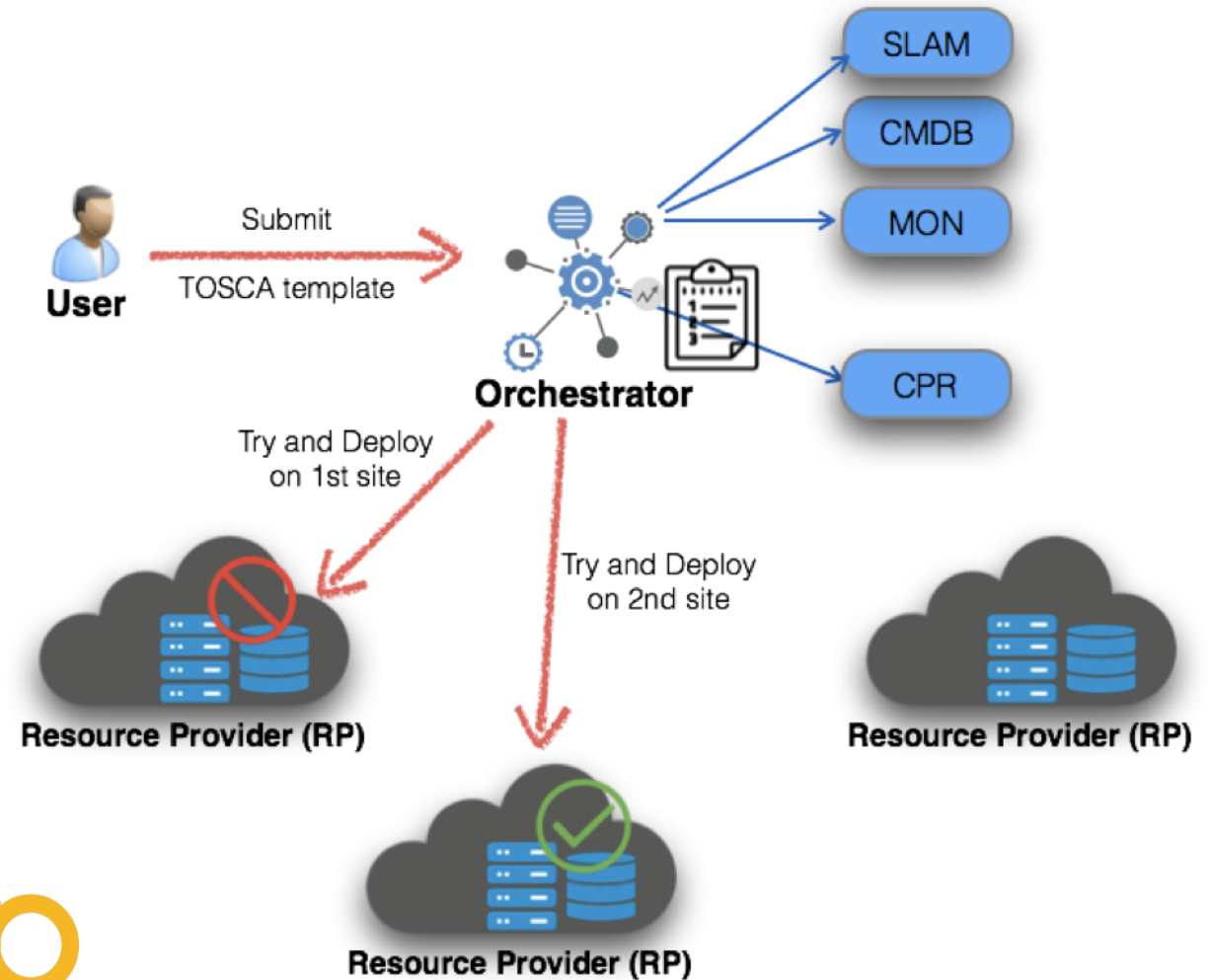
- Deploy
- Undeploy
- Update





- The Orchestrator receives the deployment request (**TOSCA** template)
- The Orchestrator collects all the information needed to deploy the virtual infra/service/job consuming others PaaS μ Services APIs:
 - **SLAM Service**: get the prioritized list of SLAs per user/group;
 - **Configuration Management DB**: get the the capabilities of the underlying IaaS platforms;
 - **Data Management Service**: get the status of the data files and storage resources needed by the service/application;
 - **Monitoring Service**: get the IaaS services availability and their metrics;
 - **CloudProviderRanker Service** (Rule Engine): sort the list of sites on the basis of rules defined per user/group/use-case;
- The orchestrator delegates the deployment to **IM**, **HEAT** or **Mesos** based on the TOSCA template and the list of sites.
- Cross-site deployments are also possible.

- The Orchestrator implements a **trial-and-error mechanism** that allows to re-schedule the deployment on the next available cloud provider from the list of candidate sites.
- Example: deployment fails because of exceeding the quota on the chosen site



- Support the deployment of **complex virtual infrastructures** (e.g. Mesos, Kubernetes, HTCondor clusters) and their **elasticity**
 - **TOSCA templates** are available in the [indigo-dc github](https://github.com/indigo-dc) repository
- Support for deployments that require the use of **specialized hw resources**
 - GPU already supported (**DEEP-1**)
 - Infiniband will be included in **DEEP-2**
- Support for deployments that span across **multiple sites**
 - Implementing **overlay networks** (INDIGO vRouter)





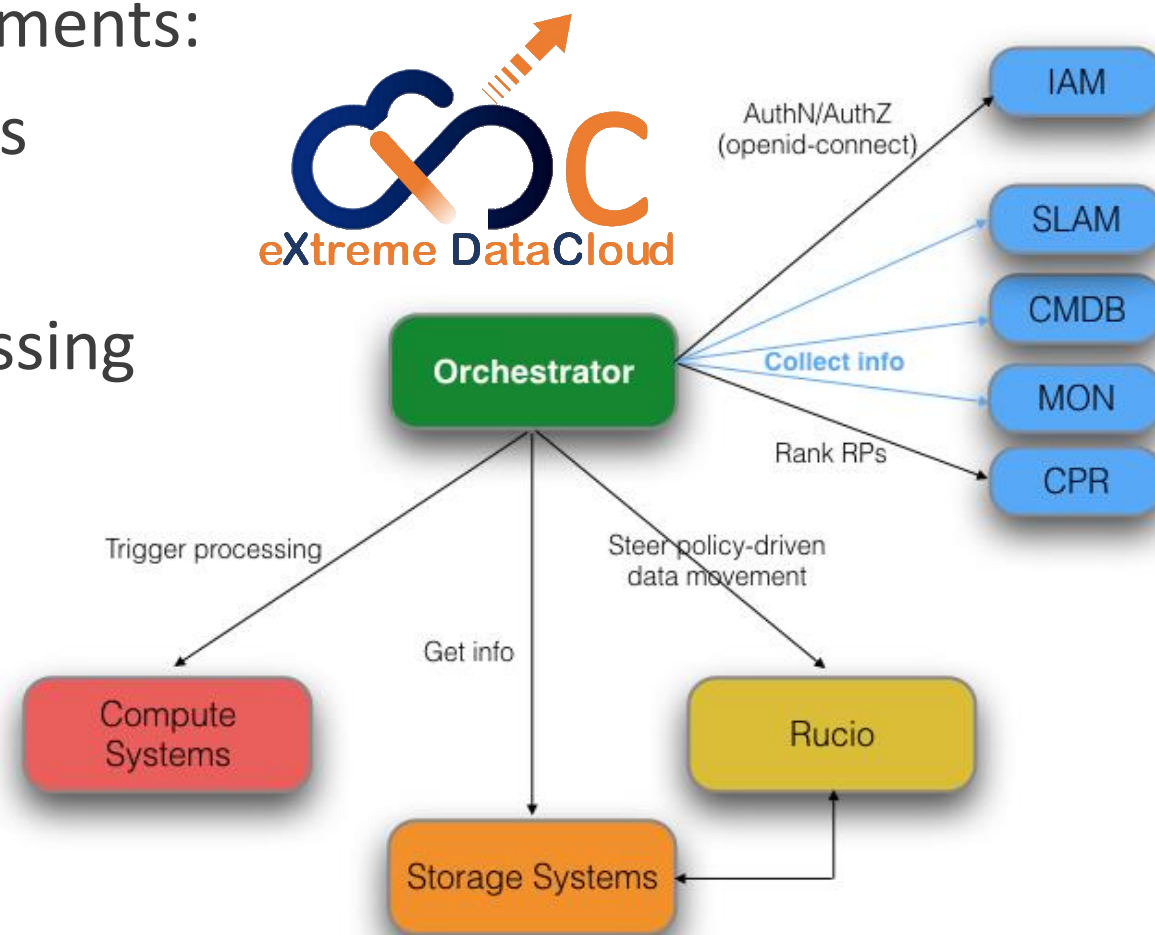
- The Orchestrator is able to select automatically the best compute cluster based on the data location
- **Plugin for Onedata and Dynafed**
 - The user specifies the "logical" path/name of the data needed by the application (e.g. onetada "space")
 - The Orchestrator interacts with the federator (Onezone or Dynafed) service that knows the geographical distribution of storage providers and monitors the availability of storage providers that host the replicas
- Intersecting the information about the storage providers and the compute providers (from CMDB), **the Orchestrator is able to schedule the processing jobs near the data**

In the framework of the **XDC project**, the Orchestration layer is being extended in order to address new requirements:

- move data between distributed storages
- specify different QoS for replicas
- launch and monitor user defined processing jobs at ingestion time

Rucio

- Initially developed by the High-Energy Physics experiment ATLAS
- Manages multi-location data in a heterogeneous distributed environment



• Alien4Cloud

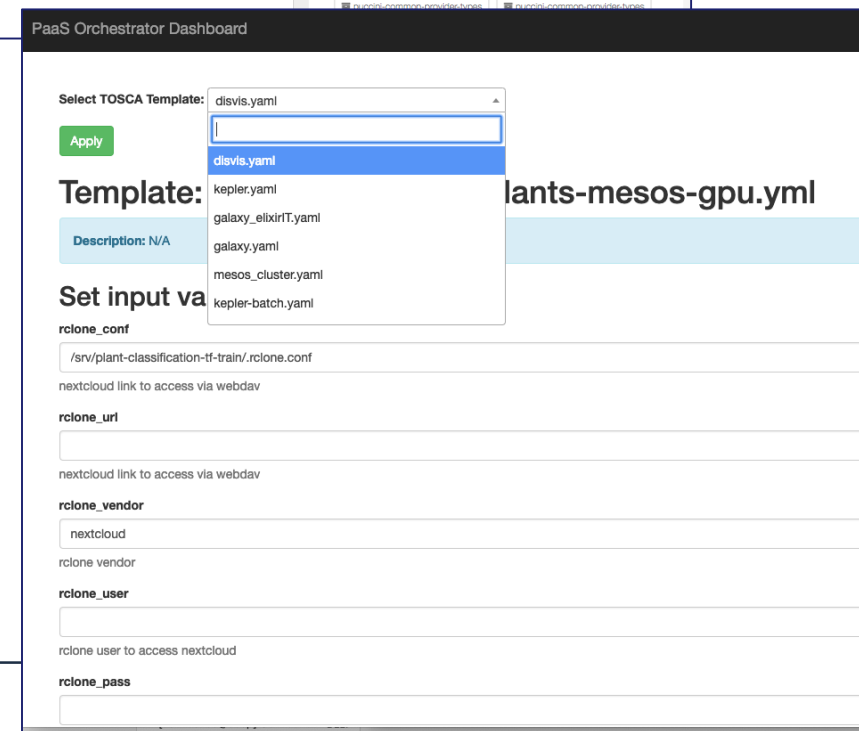
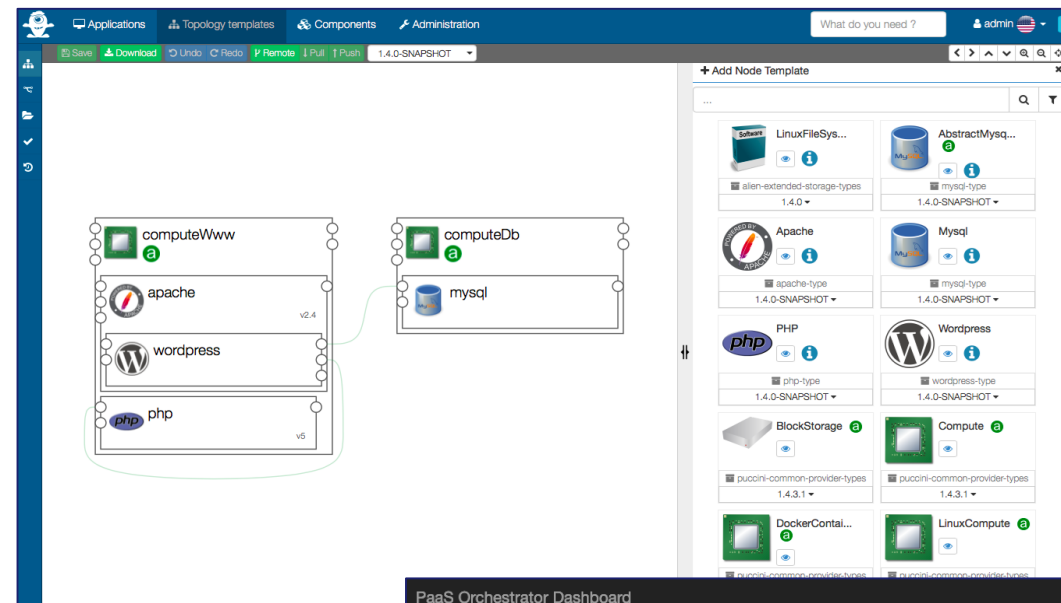
- Open-source tool to graphically compose templates

• Orchestrator dashboard

- easy to use web portal for submitting tosa templates to the Orchestrator

• Orchest

- Command line interface



```
Commands:
help [<command>...]
  Show help.

depls [<flags>]
  list deployments

depshow <uuid>
  show a specific deployment

depcreate [<flags>] <template> <parameter>
  create a new deployment

depupdate [<flags>] <uuid> <template> <parameter>
  update the given deployment

deptemplate <uuid>
  show the template of the given deployment

depdel <uuid>
  delete a given deployment

resls <deployment uuid>
  list the resources of a given deployment

resshow <deployment uuid> <resource uuid>
  show a specific resource of a given deployment
```

- Implement a plugin in the Orchestrator for secrets management
 - Built around the **Hashicorp Vault**
- Support the **interaction with HPC resources** with PaaS approach
 - Via REST APIs (reference implementation based on QCG from PSNC)

● DODAS

- allows to instantiate on-demand container-based clusters, both HTCondor batch systems, and platforms for Big Data analysis (based on Spark and/or Hadoop) (DOI: 10.22323/1.327.0024)

● Dariah

- Digital repository as a service: automatic deployment of an Invenio-based repository using TOSCA orchestration and Apache Mesos
(<https://indico.cern.ch/event/587955/contributions/2937204/>)

● ELIXIR-ITALY

- Laniakea: an open solution to provide Galaxy "on-demand" instances over heterogeneous cloud infrastructures (DOI: 10.1101/472464)

DEMO

Submitting a GPU application through the INDIGO PaaS stack

Day 3, Friday, 12th April 2019, from 9:00 to 14:30 CET

**Thank you
for your attention!**

Questions?



EOSC-hub

 eosc-hub.eu  [@EOSC_eu](https://twitter.com/EOSC_eu)



This material by Parties of the EOSC-hub Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).