

Orchestrating complex deployments with the PaaS Orchestrator

Ibergrid 2019

Santiago de Compostela, Spain
September 2019

MARICA ANTONACCI, ALESSANDRO COSTANTINI
marica.antonacci@ba.infn.it, alessandro.costantini@cnae.infn.it

Istituto Nazionale di Fisica Nucleare (INFN)

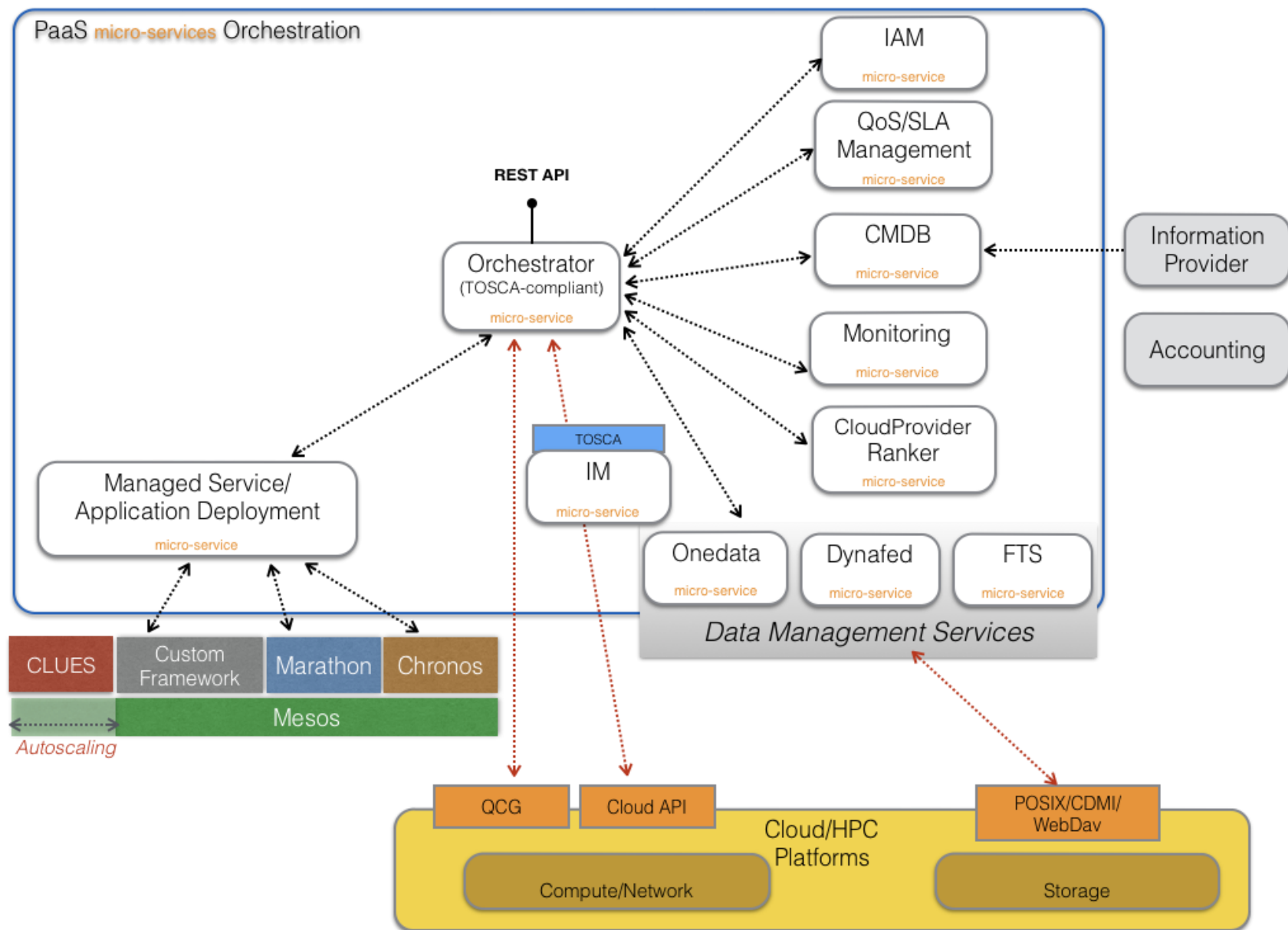
DEEP-Hybrid-DataCloud is funded by the Horizon 2020 Framework Programme
of the European union under grant agreement number 777435



- PaaS layer
- Deployment workflow
- Orchestrator architecture
- Usage scenarios
- APIs and tools

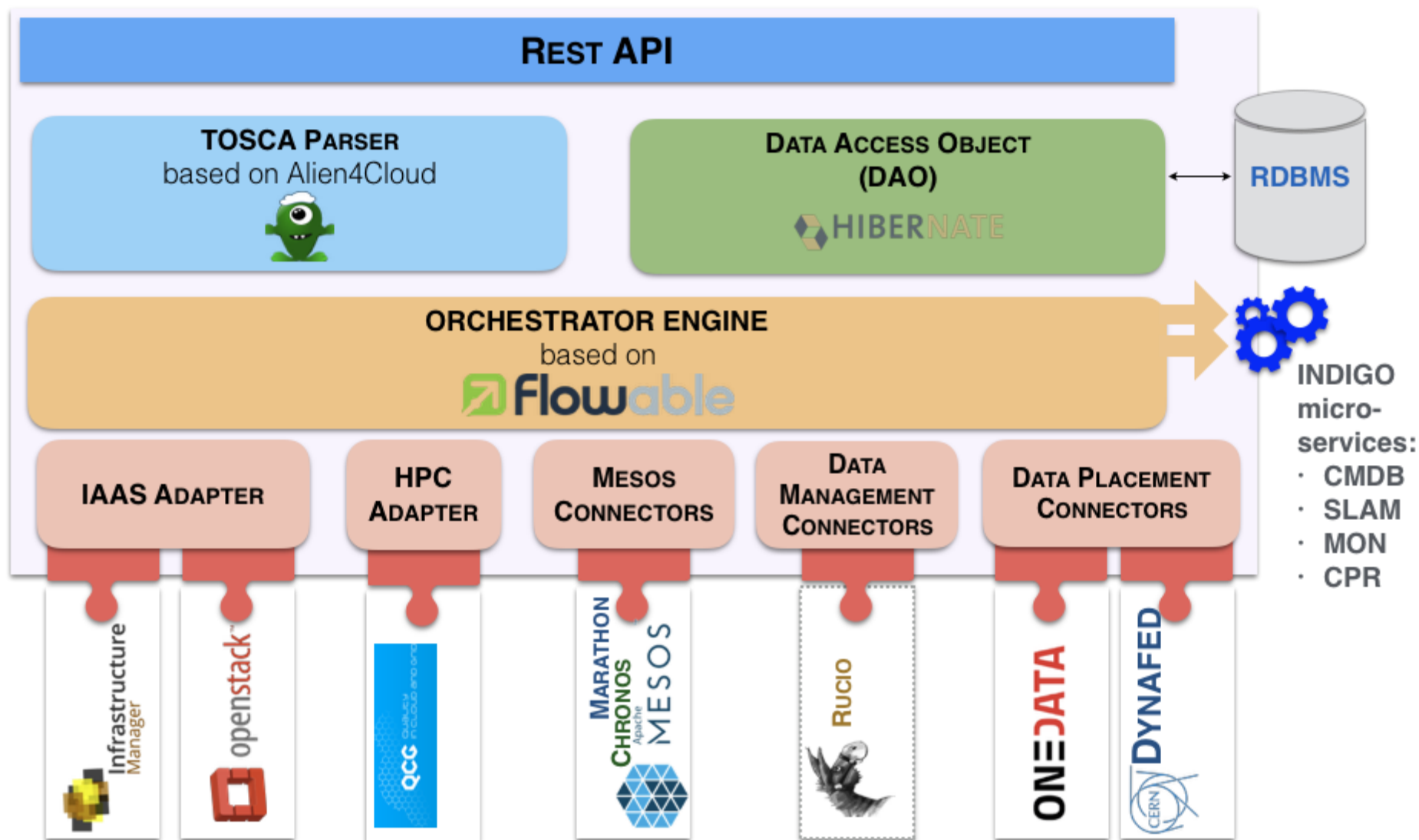
- The PaaS Orchestrator is based on the developments carried out during the **INDIGO-DataCloud** project
 - advanced features and important enhancements are being implemented in the framework of three projects: **DEEP-Hybrid DataCloud**, **eXtreme-DataCloud** and **EOSC-Hub**
- It allows to coordinate 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.
- The PaaS orchestrator 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

INDIGO Platform as a Service Layer



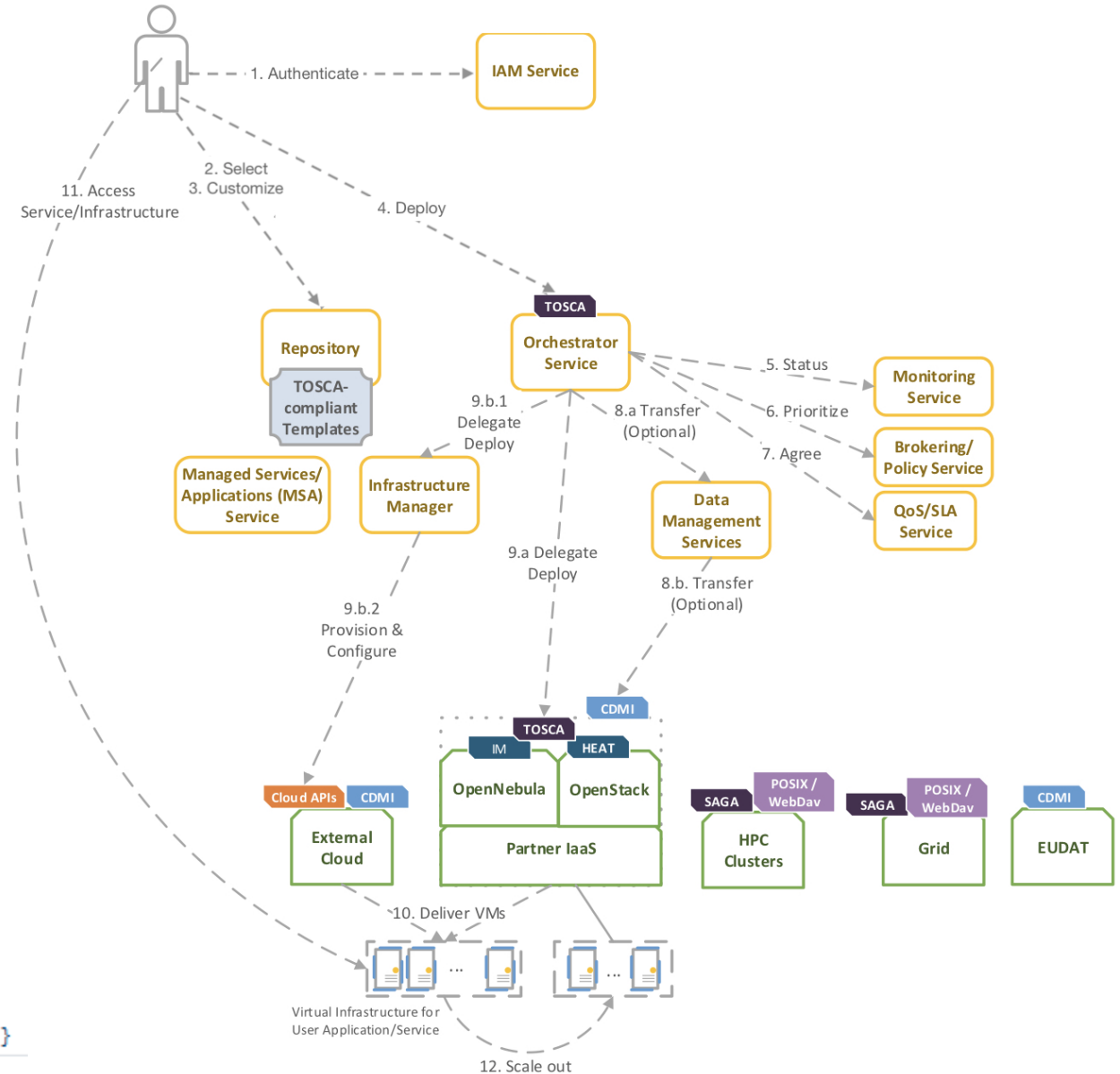
- The Orchestrator receives the deployment request (TOSCA template)
- The Orchestrator collects all the information needed to deploy the virtual infra/service/job consuming others PaaS μ Services APIs:
 - **SLAM Service**: get the prioritized list of SLAs per user/group;
 - **Configuration Management 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 configurable rules;
- The orchestrator delegates the deployment to **IM, Mesos or QCG-Computing** based on the TOSCA template and the list of sites.
- Cross-site deployments are also possible.

PaaS Orchestrator Architecture

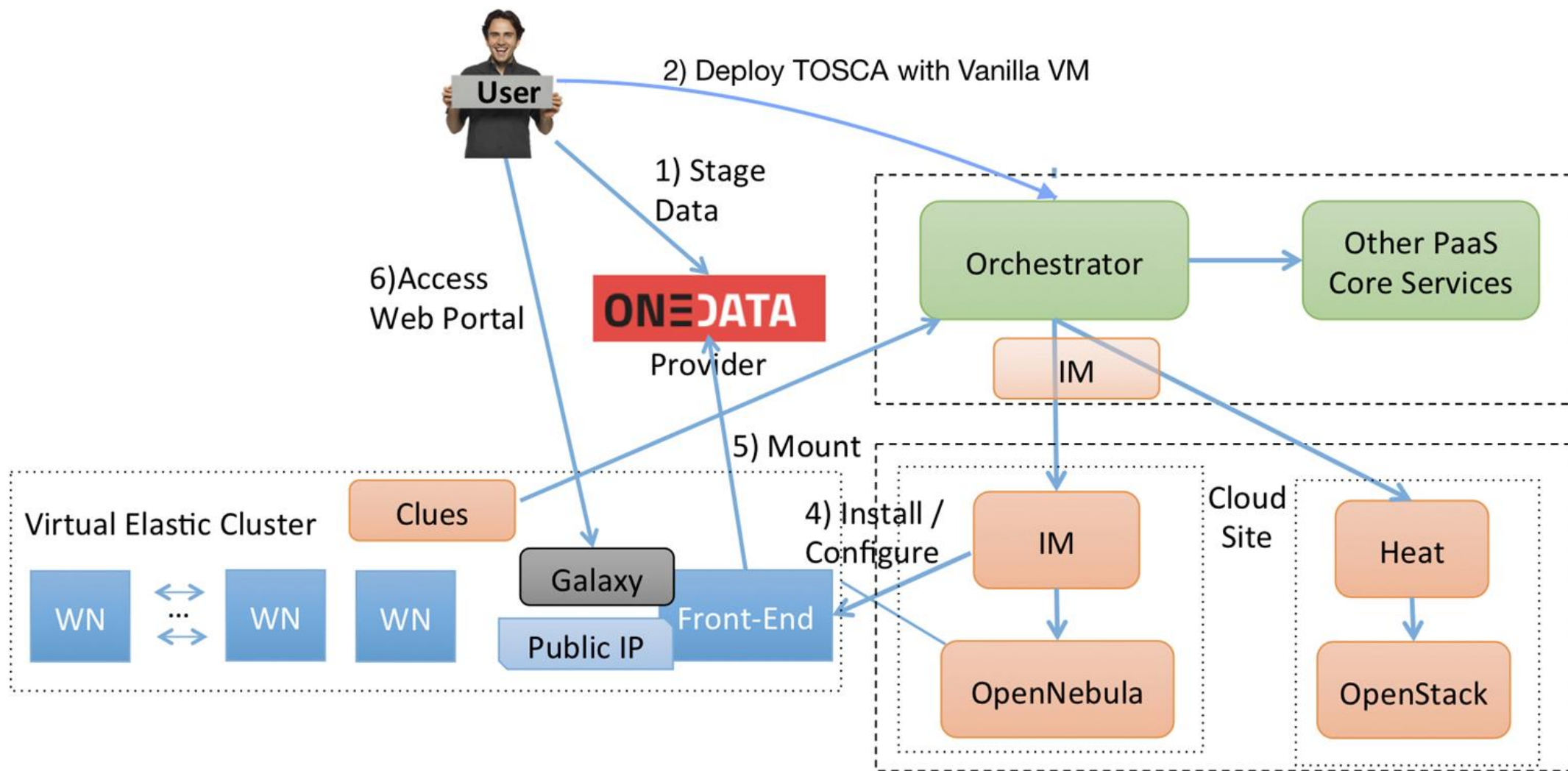


Scenario I: deployment of Virtual Infrastructure

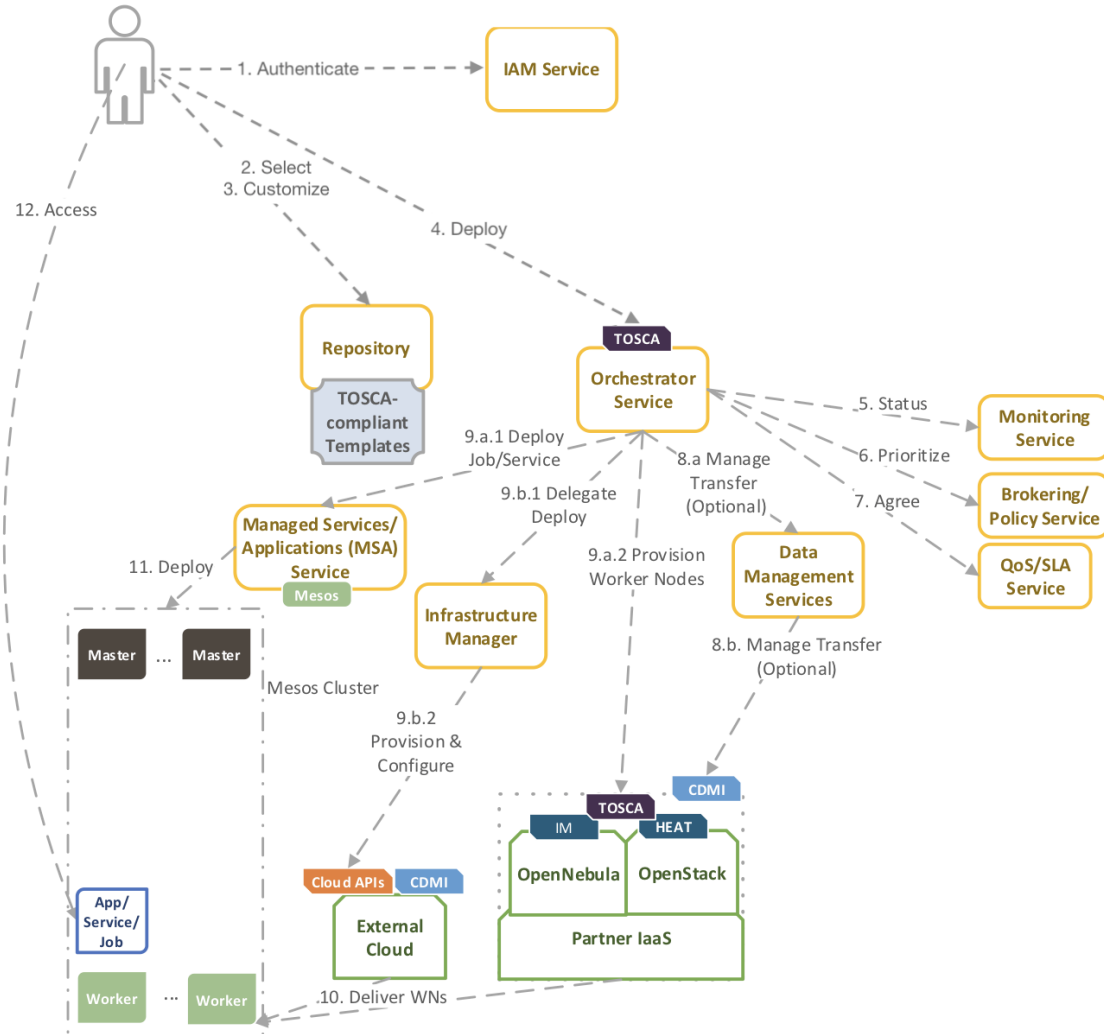
```
1  toska_definitions_version: toska_simple_yaml_1_0
2
3  imports:
4    - indigo_custom_types: https://raw.githubusercontent.com
5
6  description: TOSCA test for a node with SSH listening :
7
8  topology_template:
9
10   node_templates:
11
12     simple_node:
13       type: toska.nodes.indigo.Compute
14       capabilities:
15         endpoint:
16           properties:
17             network_name: PUBLIC
18           ports:
19             ssh_port:
20               protocol: tcp
21               source: 10022
22               target: 22
23         scalable:
24           properties:
25             count: 1
26         host:
27
34   outputs:
35     node_ip:
36       value: { get_attribute: [ simple_node, public_address, 0 ] }
37     node_creds:
38       value: { get_attribute: [ simple_node, endpoint, credential, 0 ] }
```



Use case: frontend + elastic batch system

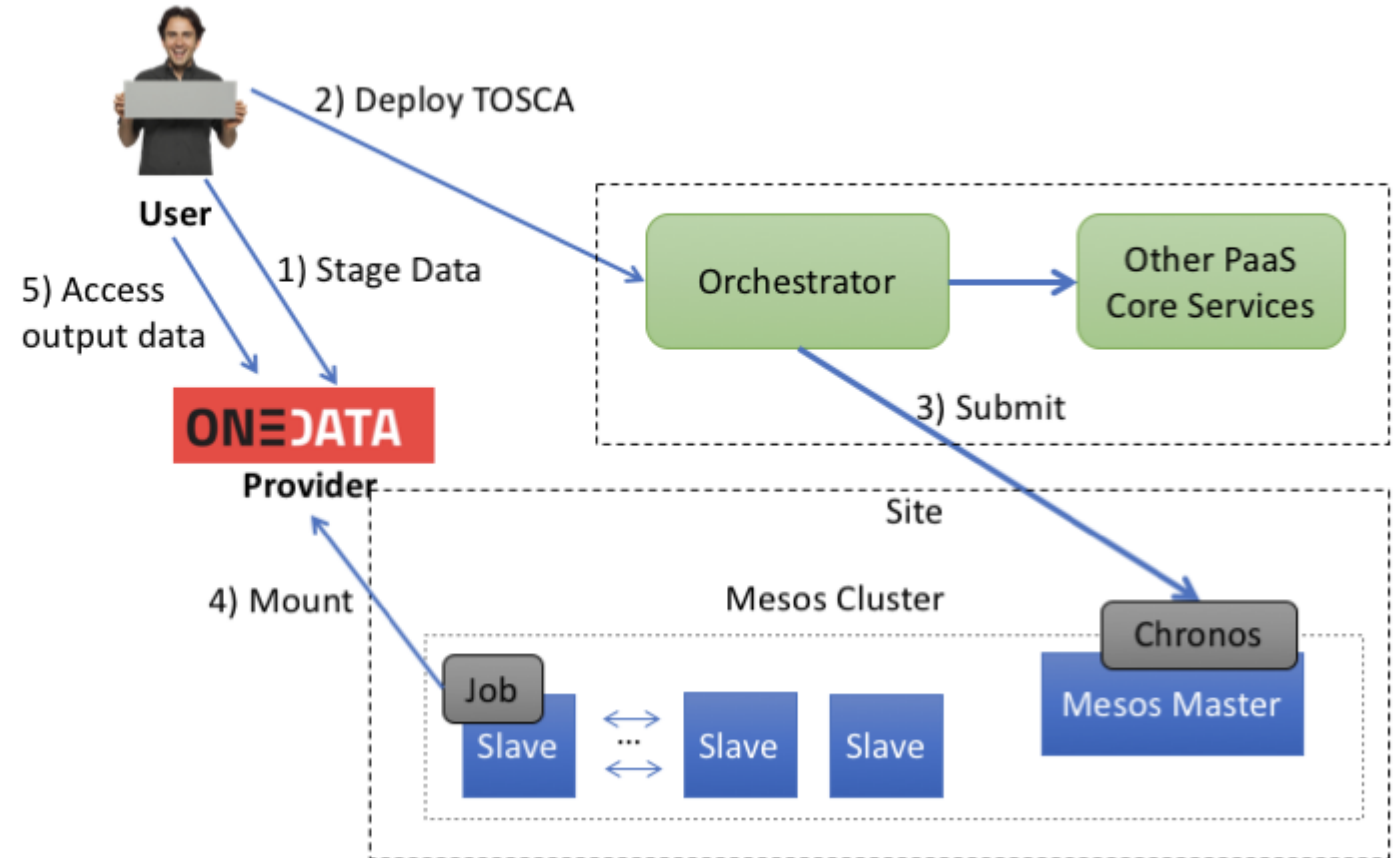
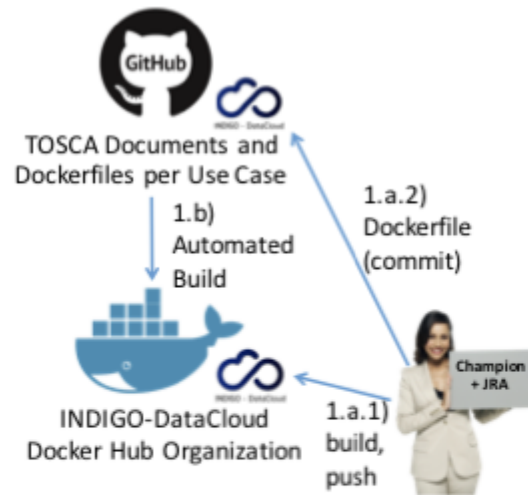


Scenario II: Deployment of managed services/jobs



- The PaaS orchestrator interacts with:
 - Marathon to deploy, monitor and scale **Long-Running services**, ensuring that they are always up and running.
 - Chronos to run user **applications** (jobs), taking care of fetching input data, handling dependencies among jobs, rescheduling failed jobs.
- **Marathon** and **Chronos** are two powerful frameworks that can be deployed on top of a Mesos Cluster.
- **Mesos** is able to manage cluster resources (cpu, mem) providing *isolation* and *sharing* across distributed applications (frameworks)

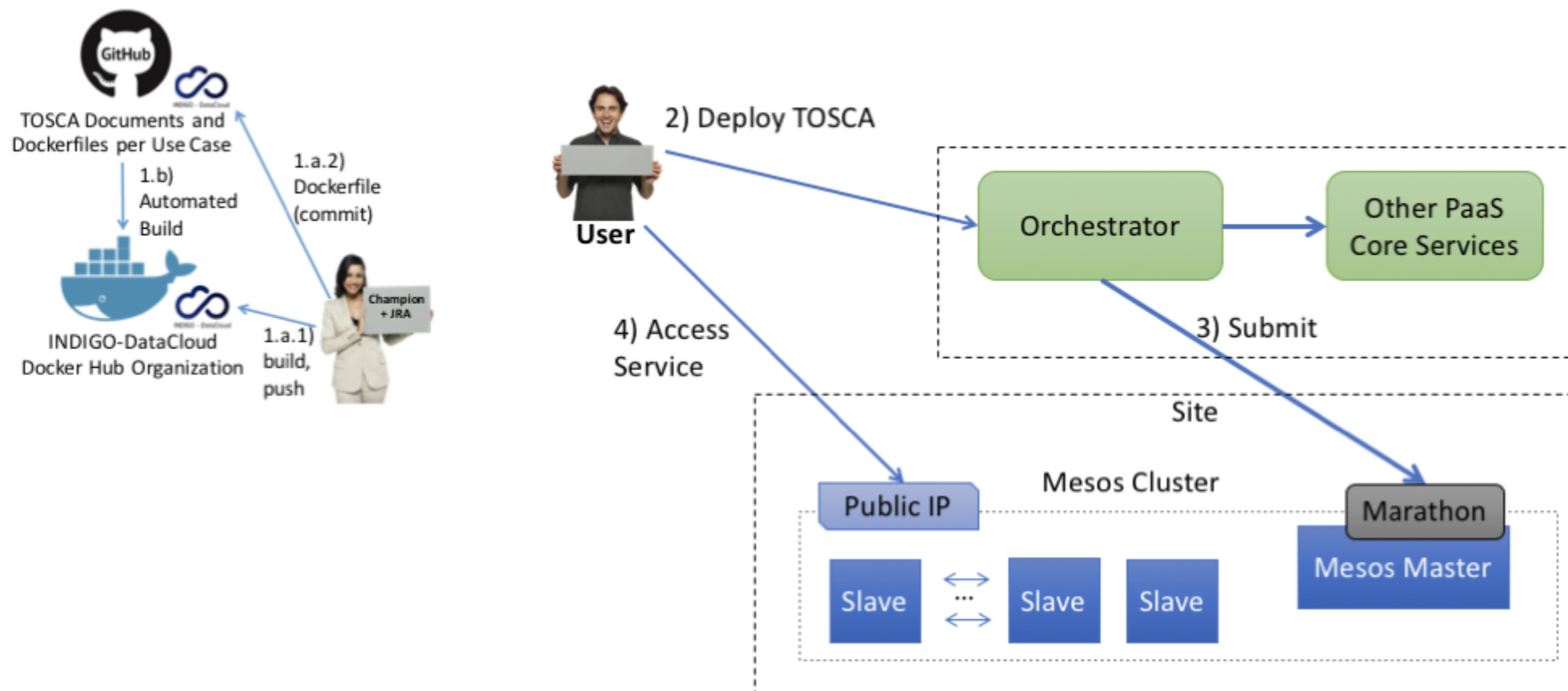
Use-case: execution of batch-like jobs



Features using TOSCA Templates

- **Parameter sweep**
 - Multiple job submission
- **Job Resubmission**
 - Configurable
- **Job dependencies**
 - managed automatically
- **Data-aware Scheduling**
 - Based on the data location

Use-case: deployment of a long-running service



Long-running service: generic template

```
tosca_definitions_version: tosca_simple_yaml_1_0

imports:
- indigo_custom_types: https://raw.githubusercontent.com/indigo-dc/tosca-types/master/custom_types.yaml

description: >
  TOSCA examples for specifying Marathon applications to enable the
  specification of long-running services in INDIGO.

topology_template:

  inputs:

    cpus:
      type: float
      description: Amount of CPUs for this service
      required: yes
      default: 1.0

    mem:
      type: scalar-unit.size
      description: Amount of Memory for this service
      required: yes
      default: 1 GB

    docker_image:
      type: string
      description: Docker image to be used to run the container application
      required: yes
      default: ""

    port:
      type: integer
      description: service port (exposed by the docker container)
      required: yes
      default: 8080

    data_path:
      type: string
      description: container path for persistent data
      required: no
      default: "/data"
```

outputs:

```
  endpoint:
    value: { concat: [ { get_attribute : [ marathon-app, load_balancer_ips, 0 ] }, ':', { get_attribute : [ docker_runtime, host, publish_ports, 0, target ] } ] }
```

Use-case: DEEPaaS service deployment

Deep Learning prediction modules included in the DEEP-HybridDataCloud Open Catalog (<https://marketplace.deep-hybrid-datacloud.eu/>) can be deployed through a TOSCA template

- the DEEPaaS API is deployed as long running service on Mesos cluster
- the API can be accessed from the web browser and used to make predictions

DEEP as a Service API endpoint 0.5.0

[Base URL: /]
<http://mesos-lb-recas.ba.infn.it:10019/swagger.json>

DEEP as a Service (DEEPaaS) API endpoint.

models Model information, inference and training operations ▼

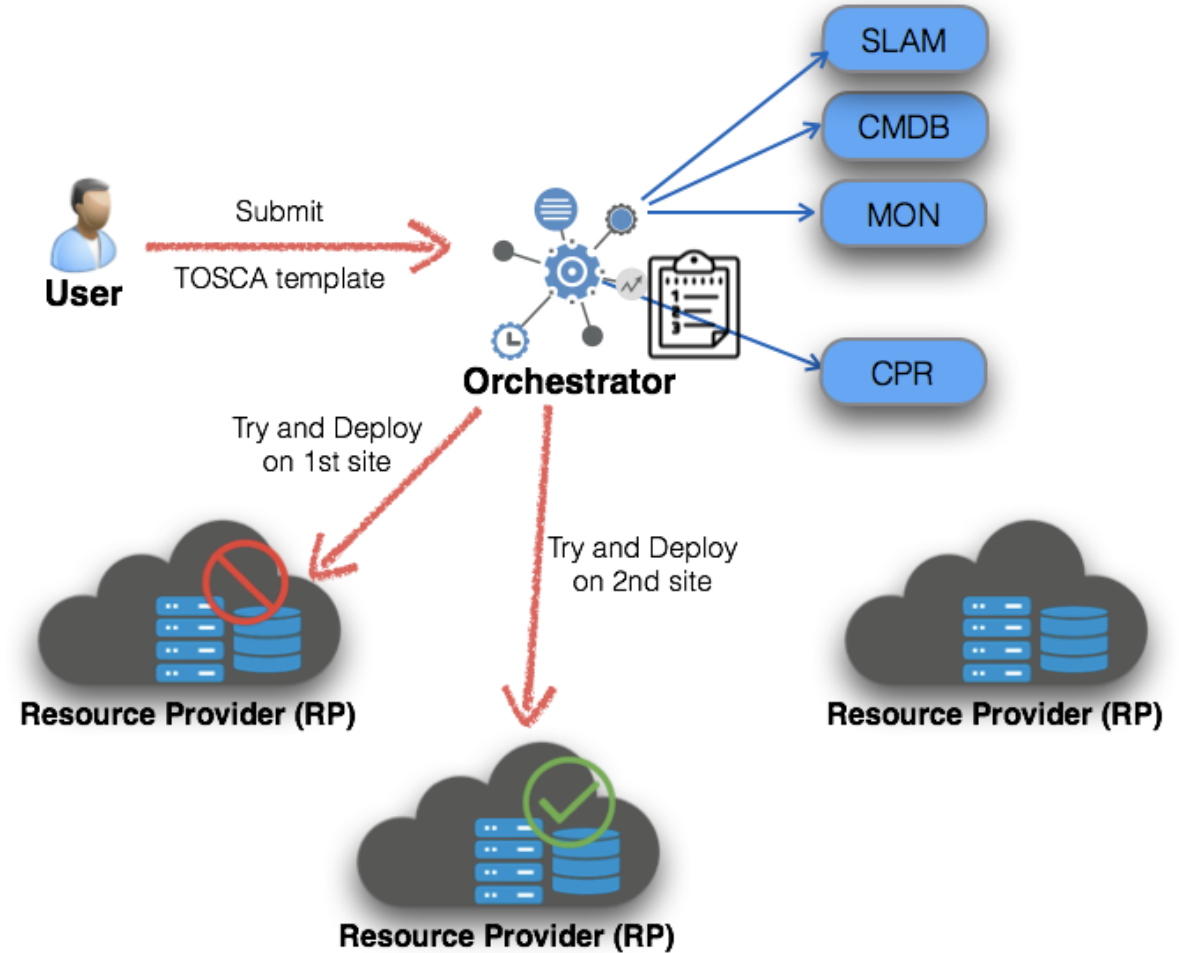
GET	/models/	Return loaded models and its information
GET	/models/imgclas	Return model's metadata
POST	/models/imgclas/predict	Make a prediction given the input data
PUT	/models/imgclas/train	Retrain model with available data

- **Elastic Galaxy Cluster**
 - a Galaxy portal is automatically deployed from TOSCA and configured to use a SLURM elastic cluster
- **Elastic Mesos Cluster**
 - a complete HA Mesos cluster with Chronos/Marathon framework is automatically deployed from a TOSCA template
- **Jupyter with K8s Cluster**
- **HTCondor cluster on Mesos (DODAS)**
- **Big-data Analysis Cluster (Spark on Mesos)**
- **Deep Learning training/prediction services (DEEPaaS)**

Advanced features

Deployment retry strategy

- 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



The problem: user applications need access to sensitive data (e.g. password for dbs, storage service credentials, etc.)

The solution: introduce a secrets manager, **Hashicorp Vault**, in the PaaS architecture to store the sensitive information safely.

The Orchestrator writes the secrets in Vault on behalf of the user, then the Marathon plugin retrieves the secrets from Vault and makes them available to the application container (via environment variables)



- The PaaS Orchestrator supports the deployment of virtual machines and containers that need to **access specialised hardware devices**, namely GPUs, to provide the processing power required by tasks like **Machine Learning** algorithms
 - the GPU requirements (num, vendor, model) can be specified in the TOSCA template
 - the Orchestrator automatically selects the sites/services that provide the needed capabilities (flavors, gpu support)
- The Orchestrator includes a plugin for **submitting jobs to HPC** facilities
 - exploits the **QCG-Computing** service (PSNC) that exposes REST APIs to submit jobs to the underlying batch systems

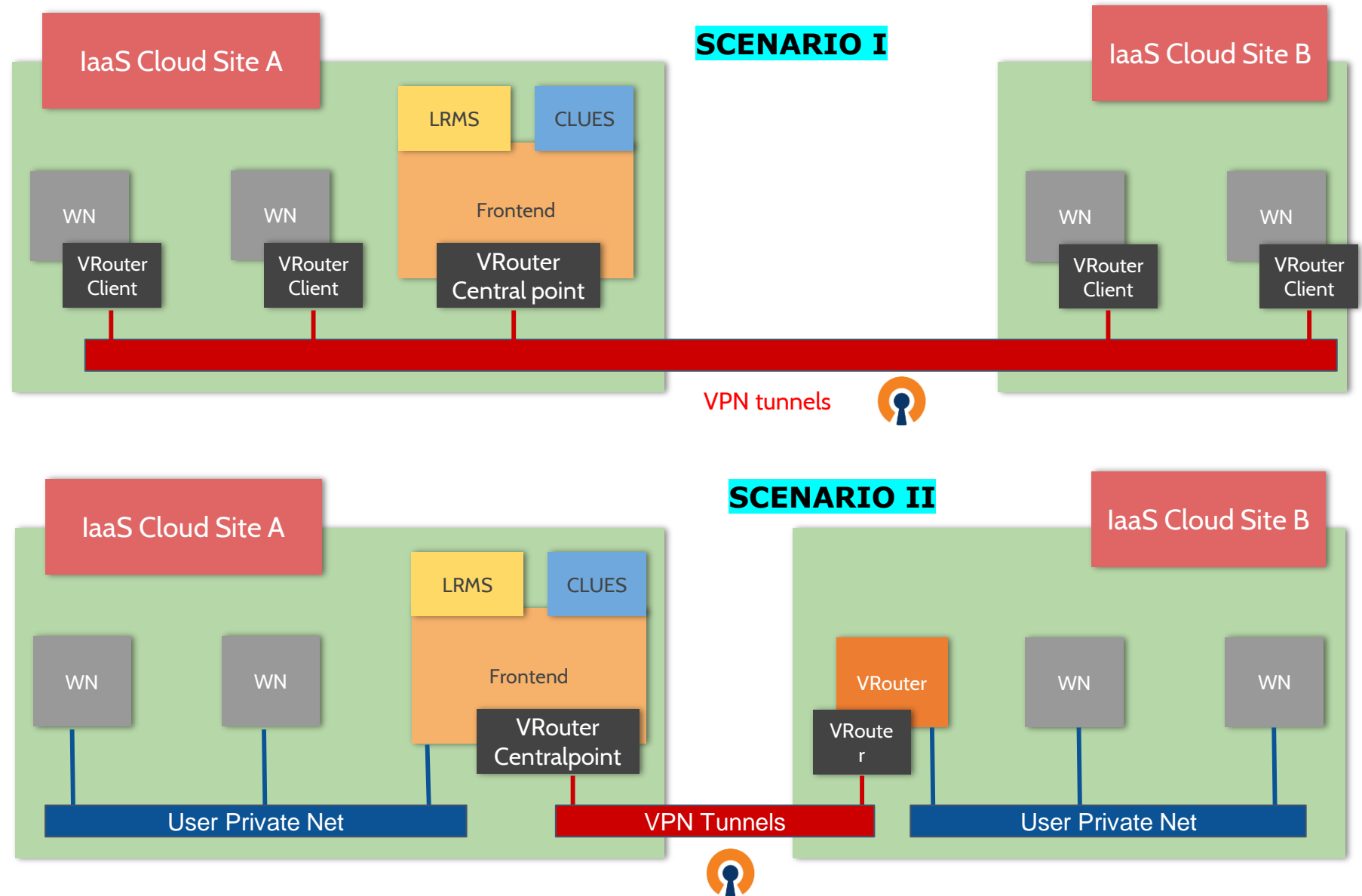
Support for hybrid deployments of elastic clusters

Scenario I:

- exploits L2 network provided by the sites

Scenario II:

- dedicated private nets are automatically provisioned



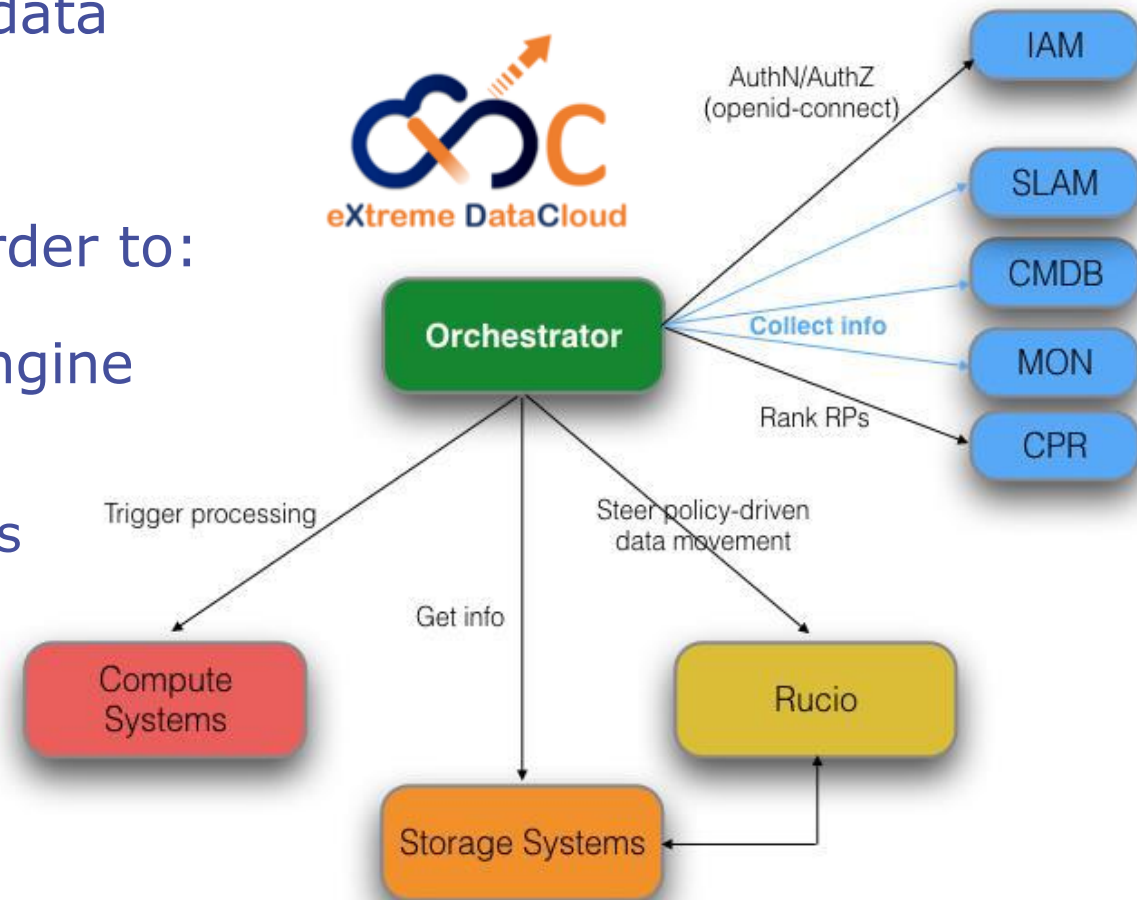
Further features and enhancements

The PaaS Orchestrator has been enhanced to:

- schedule the processing jobs near the data

The PaaS Orchestrator is being extended in order to:

- Integrate a data management policy engine (QoS and Data Life Cycle)
 - move data between distributed storages
 - specify different QoS for replicas
- Support workflows for data pre-processing at ingestion



- **Create a deployment:**
 - POST request to /deployments - parameters:
 - template: string containing a TOSCA YAML-formatted template
 - parameters: the input parameters of the deployment (map of strings)
- **Get deployment details:**
 - GET request to /deployments:
 - `curl 'http://localhost:8080/deployments/<uuid>'`
- **Delete deployment:**
 - DELETE request
 - `curl 'http://localhost:8080/deployments/<uuid>'`
- **Documentation:** <http://indigo-dc.github.io/orchestrator/restdocs/#overview>

Orchent: The Orchestrator CLI

```
export ORCHENT_TOKEN=<your access token>
export ORCHENT_URL=<orchestrator_url>
```

```
usage: orchent <command> [<args> ...]
```

Commands:

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

```
depls
    list all deployments
```

```
depshow <uuid>
    show a specific deployment
```

```
decreate [<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
```

Installation guide:


<https://indigo-dc.gitbooks.io/orchent/content/admin.html>

User guide:

<https://indigo-dc.gitbooks.io/orchent/content/user.html>


The Orchestrator Dashboard


Simple graphical user interface for the Orchestrator


PaaS Orchestrator Dashboard Deployments SLAs Settings  Authentication via INDIGO IAM


Search...

1 Select deployment type


Lifewatch Algae Bloom



DisVis



Galaxy Elastic Cluster



Dariah Data Repository


Kepler
TOSCA template for deploying an instance for Kepler
[Configure](#)

Galaxy Elastic Cluster - Elixir Italy


Galaxy - Elixir Italy


Mesos Cluster


Kepler batch job


2

Template: kepler.yaml

Configure input parameters

Description: TOSCA template for deploying an instance for Kepler

Input Values [Advanced](#)

number_cpus
1
number of cpus required for the instance


memory_size
1 GB
ram memory required for the instance

3 [Submit](#) [Cancel](#)

Submit deployment request

List your deployments

My deployments

 Refresh

New deployment

Show

10 ▾

 entries

Search:

Deployment uuid	Status	Creation time	Deployed At	Actions
11e9ddcc-74e1-e98e-b691-067adb74d617	CREATE_COMPLETE	2019-09-23T06:36+0000	provider-RECAS-BARI	<div>Delete ▾</div>
11e9dc71-fe4a-a242-b691-067adb74d617	CREATE_COMPLETE	2019-09-21T13:16+0000	provider-RECAS-BARI	<div>Details Show template Log</div>
11e9d915-526c-bfb4-8967-067adb74d617	CREATE_COMPLETE	2019-09-17T06:35+0000	provider-RECAS-BARI	
11e9aeaa-8fea-48c9-83c8-067adb74d617	CREATE_COMPLETE	2019-07-25T07:05+0000	provider-RECAS-BARI	<div>Delete ▾</div>
11e9ac63-6f8e-095d-9841-067adb74d617	CREATE_COMPLETE	2019-07-22T09:31+0000	provider-RECAS-BARI	<div>Delete ▾</div>
11e9a7b2-172a-5c41-8c92-067adb74d617	DELETE_FAILED	2019-07-16T10:11+0000	provider-IFCA-LCG2	<div>Delete ▾</div>

Showing 1 to 6 of 6 entries

Previous

1

Next

Get deployment details and outputs

My deployments

Show 10 ▾ entries

Deployment uuid

11e9ddcc-74e1-e98e-b691-067adb74d617

11e9dc71-fe4a-a242-b691-067adb74d617

11e9d915-526c-bfb4-8967-067adb74d617

11e9aeaa-8fea-48c9-83c8-067adb74d617

11e9ac63-6f8e-095d-9841-067adb74d617

11e9a7b2-172a-5c41-8c92-067adb74d617

Deployment details

Overview Outputs

UUID: 11e9ac63-6f8e-095d-9841-067adb74d617

STATUS: CREATE_COMPLETE

CREATED AT: 2019-07-22T09:31+0000

UPDATED AT: 2019-07-22T09:32+0000

DEPLOYED AT: provider-RECAS-BARI

Close

 Refresh

+ New deployment

Search:

Deployed At

Actions

provider-RECAS-BARI


 Delete ▾

provider-RECAS-BARI

≡ Details
🔍 Show template
📄 Log

provider-RECAS-BARI


provider-RECAS-BARI

 Delete ▾

provider-RECAS-BARI

 Delete ▾

provider-IFCA-LCG2

 Delete ▾

Showing 1 to 6 of 6 entries

Previous 1 Next

View the deployment log

Deployment log

Refresh

← Back

```
2019-09-23 06:37:05.617885: Select master VM
2019-09-23 06:37:05.619607: Wait master VM to boot
2019-09-23 06:37:05.624133: Wait master VM to have the SSH active.
2019-09-23 06:37:50.835489: Creating and copying Ansible playbook files
2019-09-23 06:37:51.765238: Galaxy role indigo-dc.zabbix-agent detected setting to install.
2019-09-23 06:37:51.765432: Performing preliminary steps to configure Ansible.
2019-09-23 06:37:52.395987: Configure Ansible in the master VM.
2019-09-23 06:42:08.735649: Ansible successfully configured in the master VM.
2019-09-23 06:42:16.319374: Copying YAML, hosts and inventory files.
VM 0:
Contextualization agent output processed successfullyGenerate and copy the ssh key

Sleeping 0 secs.
Launch task: wait_all_ssh
Waiting SSH access to VM: 90.147.75.27
Testing SSH access to VM: 172.30.99.81:22
Remote access to VM: 90.147.75.27 Open!
Changing the IP 172.30.99.81 for 90.147.75.27 in config files.
Task wait_all_ssh finished successfully
Process finished
Contextualization agent output processed successfullyGenerate and copy the ssh key
Sleeping 0 secs.
Launch task: basic
Waiting SSH access to VM: 90.147.75.27
Testing SSH access to VM: 172.30.99.81:22
Remote access to VM: 90.147.75.27 Open!
Requiretty successfully removed
Install indigo-dc.zabbix-agent with ansible-galaxy.
Galaxy dependencies file: [{src: indigo-dc.zabbix-agent}]

Call Ansible

PLAY [90.147.75.27_0] *****

TASK [Check Python is installed] *****
Monday 23 September 2019 06:42:34.044102
ok: [90.147.75.27_0]

TASK [Bootstrap with python] *****
Monday 23 September 2019 06:42:34.083232
skipping: [90.147.75.27_0]

TASK [Install libselinux-python on redhat systems] *****
Monday 23 September 2019 06:42:34.110847
fatal: [90.147.75.27_0]: FAILED! => {"changed": false, "failed": true, "module_stderr": "sudo: unable to resolve host simple-nc
...ignoring

TASK [Set the hostname of the node] *****
Monday 23 September 2019 06:42:34.540385
changed: [90.147.75.27_0]
```

Useful for debugging purposes.

You can also download the log file clicking on the ‘download’ button at the end of the page.

- **TOSCA Templates**

- Use-cases templates: <https://github.com/indigo-dc/tosca-templates>
- Example templates: <https://github.com/indigo-dc/tosca-types/tree/master/examples>

- **Ansible Roles**

- Ansible Galaxy: <https://galaxy.ansible.com/indigo-dc/>

- **Docker images**

- Docker hub:
<https://hub.docker.com/u/indigodatacloudapps/dashboard/>

Building and Deploying Complex Applications with Alien4Cloud

Ibergrid 2019

Santiago de Compostela, Spain

September 2019

Marica Antonacci, Alessandro Costantini

on behalf

Andy S. Alic, Miguel Caballer, Germán Moltó

asalic@upv.es, micafer1@upv.es, gmolto@dsic.upv.es

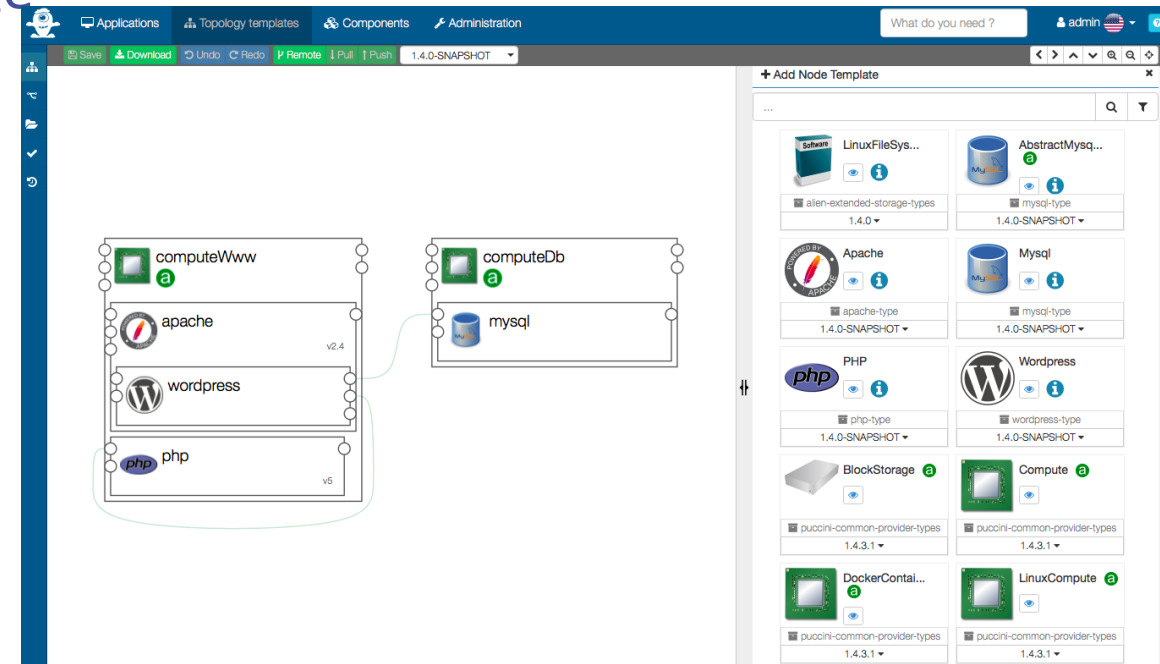
Universitat Politècnica de València (UPV)

Building/composing TOSCA Topologies

Alien4Cloud Overview (1)



- Web app for Composition/management of TOSCA topologies
 - Scratch or Existing Template
- Open Source under Apache 2.0
 - <https://github.com/alien4cloud/alien4cloud>
- Very low barrier for non-experts
 - Topologies with minimal TOSCA knowledge
 - Easy deployment/interaction with orchestrator deploying/managing the actual infrastructure
- Java REST backend (SpringBoot) / HTML5 frontend (AngularJS)
- Extensible via Plugins
 - Easy to add new orchestrators to create topologies built/composed via the UI



Building/composing TOSCA Topologies Alien4Cloud in DEEP



- Extension of the Existing Version
 - DEEP fork @ <https://github.com/indigo-dc/alien4cloud>
 - Improvements like outputs handling, TOSCA functions parsing,
- Implementation of a plugin to connect to the IndigoDC Orchestrator
 - Orchestrator @ <https://github.com/indigo-dc/orchestrator>
 - Plugin @ <https://github.com/indigo-dc/alien4cloud-deep>
 - Repo with Dockerized A4C, plugin, TOSCA normative types, and DEEP custom types
- Support for OAuth2
 - Via A4C dependency @ <https://github.com/indigo-dc/spring-social-oidc>

Building/composing Alien4Cloud UI - Configure Orchestrator Endpoint



Applications Catalog Administration admin

Information
 Deployments
 Configuration
 Locations
 Artifacts

Global configuration

Naming policy of deployment

`(application.name + '-' + environment.name).replaceAll(['^w\\-'], '_')`

Driver configuration

clientId	none
clientSecret	none
tokenEndpoint	https://iam.deep-hybrid-dat...
clientScopes	openid profile email offlin...
orchestratorEndpoint	https://deep-paas.cloud.ba....
iamHost	https://iam.deep-hybrid-dat...
user	none


Building/composing Alien4Cloud UI - Create New Users


The image shows a screenshot of the Alien4Cloud UI with a 'New User' modal form open. The background interface includes a top navigation bar with 'Applications', 'Catalog', and 'Administration' tabs, and a user profile 'admin' in the top right. The main content area has tabs for 'Users' and 'Groups', with a 'New User' button and a search bar. The modal form, titled 'New User', contains the following fields and options:

- Username:** A text input field with a red border and a red error message.
- Password:** Two text input fields labeled 'Password' and 'Confirm', both with red borders and red error messages.
- First Name:** A text input field.
- Last Name:** A text input field.
- Email:** A text input field.
- Roles:** A list of roles with checkboxes:
 - ☐ ADMIN
 - ☐ APPLICATIONS_MANAGER
 - ☐ ARCHITECT
 - ☐ COMPONENTS_BROWSER
 - ☐ COMPONENTS_MANAGER

At the bottom right of the modal are 'Create' and 'Cancel' buttons.



Building/composing Alien4Cloud UI - Components




































 Applications Catalog Administration

admin 

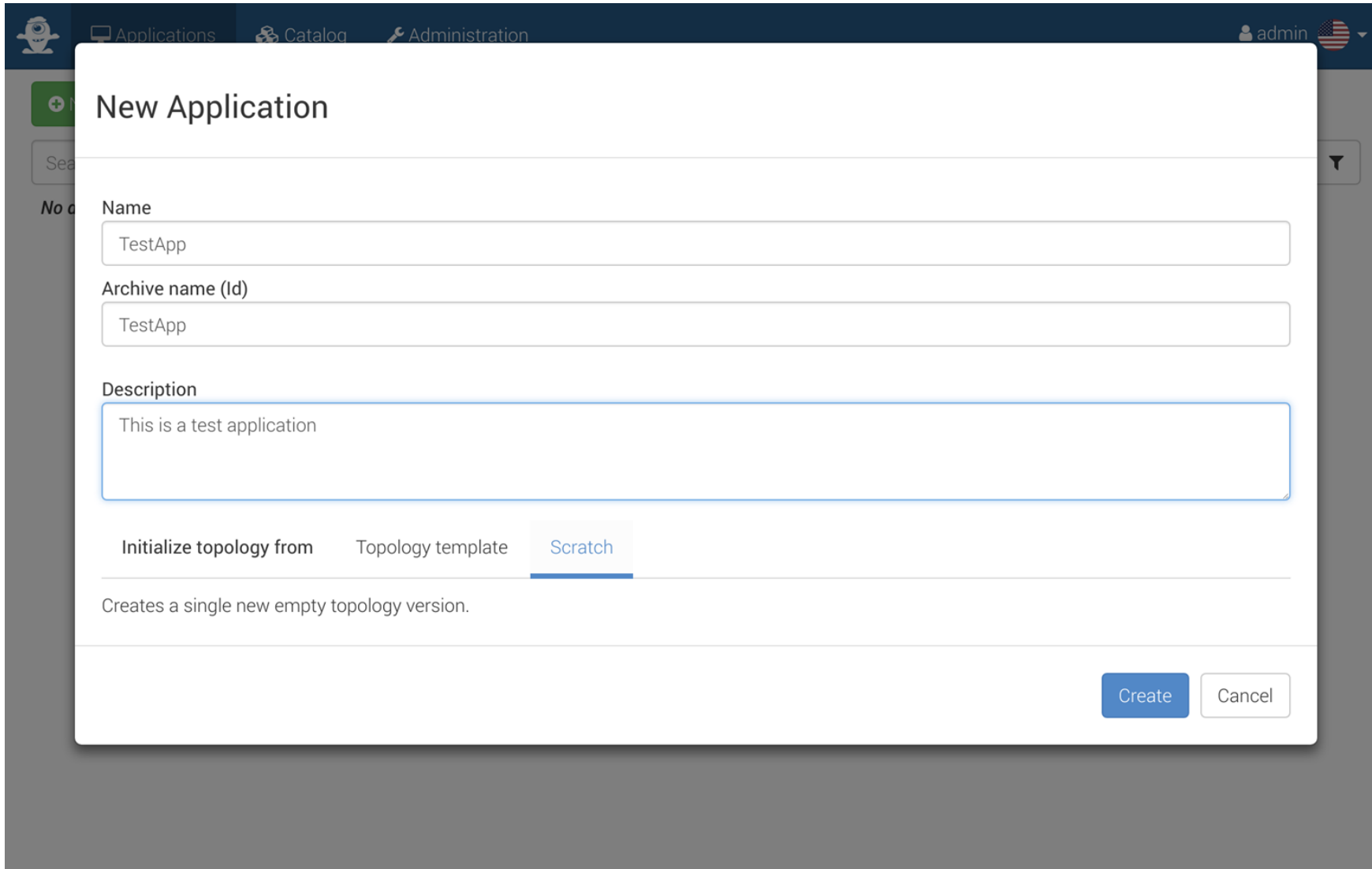
Components

Manage archives Browse components Browse topologies Browse policies

Search...  

 PersistentVo... PersistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts  org.alien4cloud.kubernetes.api 2.0.0 ▼	 PersistentVo... An extension of PersistentVolumeClaimSource that allows to specify a storageClass. If no volume that match this claim is found in the PersistentVolume pool, a volume  org.alien4cloud.kubernetes.api 2.0.0 ▼	 VolumeBase   org.alien4cloud.kubernetes.api 2.0.0 ▼	 DockerExtVol...   docker-types 2.0.0 ▼	 BlockStorage  The TOSCA BlockStorage node currently represents a server-local block storage device (i.e., not shared) offering evenly sized blocks of data from which raw storage volumes can be  tosca-normative-types 1.0.0-ALIEN20 ▼
 Compute  The TOSCA Compute node represents one or more real or virtual processors of software applications or services along with other essential local resources. Collectively, the resources the  tosca-normative-types 1.0.0-ALIEN20 ▼	 Application  The TOSCA Container Application node represents an application that requires Container-level virtualization technology.  tosca-normative-types 1.0.0-ALIEN20 ▼	 Docker  indigo-types 1.0.0 ▼	 DockerContai...  The TOSCA Container Application Docker node represents an application running in a Docker container. Properties defined in the node will be interpreted as the endpoint's  docker-types 2.0.0 ▼	 Runtime  The TOSCA Container Runtime node represents operating system-level virtualization technology used to run multiple application services on a single Compute host.  tosca-normative-types 1.0.0-ALIEN20 ▼
 DBMS  The TOSCA DBMS node represents a typical relational, SQL ...	 MySQL ...	 Database  The TOSCA Database node represents a logical database that ...	 MySQL ...	 LoadBalancer  The TOSCA Load Balancer node represents logical ...

Building/composing Alien4Cloud UI - New TOSCA Topology



The screenshot shows the 'New Application' dialog box in the Alien4Cloud UI. The dialog has a dark blue header with navigation tabs: 'Applications', 'Catalog', and 'Administration'. The 'Applications' tab is active. The user is logged in as 'admin' with a US flag icon. The dialog title is 'New Application'. It contains three input fields: 'Name' (containing 'TestApp'), 'Archive name (Id)' (containing 'TestApp'), and 'Description' (containing 'This is a test application'). Below the description field, there are three tabs for 'Initialize topology from': 'Topology template' and 'Scratch'. The 'Scratch' tab is selected and highlighted with a blue underline. Below the tabs, a note states: 'Creates a single new empty topology version.' At the bottom right, there are two buttons: 'Create' (blue) and 'Cancel' (white).

Applications Catalog Administration admin

New Application

Name
TestApp

Archive name (Id)
TestApp

Description
This is a test application

Initialize topology from Topology template Scratch

Creates a single new empty topology version.

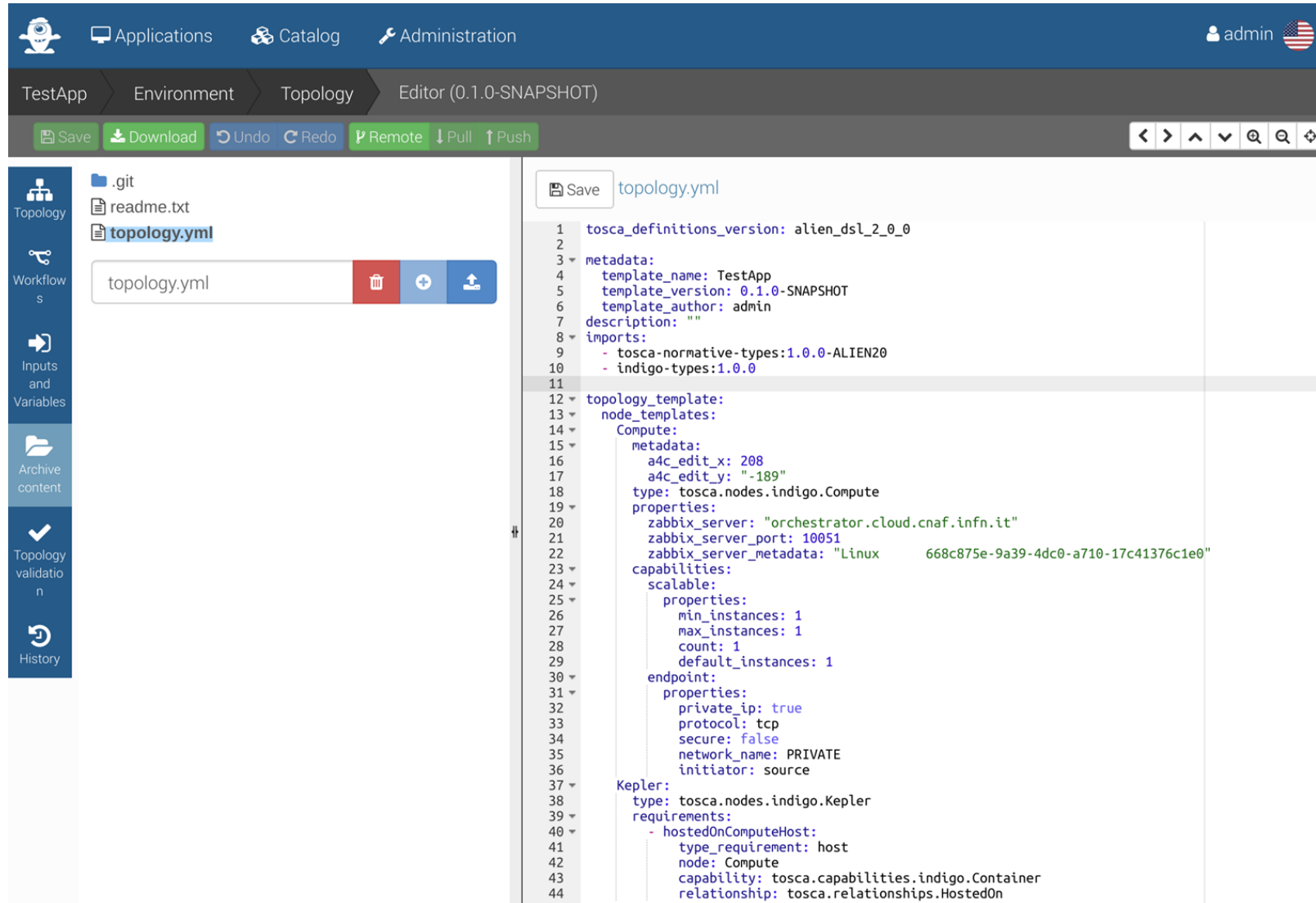
Create Cancel

Building/composing Alien4Cloud UI - GUI composition

The screenshot displays the Alien4Cloud UI interface. The top navigation bar includes 'Applications', 'Catalog', and 'Administration'. The main header shows 'TestApp', 'Environment', 'Topology', and 'Editor (0.1.0-SNAPSHOT)'. Below the header is a toolbar with 'Save', 'Download', 'Undo', 'Redo', 'Remote', 'Pull', and 'Push' buttons. The left sidebar contains icons for 'Topology', 'Workflow', 'Inputs and Variables', 'Archive content', 'Topology validation', and 'History'. The main workspace shows a topology diagram with two nodes: 'Compute' (represented by a green square icon) and 'Kepler' (represented by a blue square icon). The right panel, titled 'Selected node', displays the properties of the selected 'Compute' node. It includes a 'Type' field set to 'Compute', a 'Properties' section with three rows of data, and a 'Capabilities' section with a 'scalable' property set to 'Scalable' and a list of instance-related properties.

Selected node			
Compute			
Type: Compute			
▼ Properties			
zabbix_server	orchestrator.clou...		
zabbix_server_...	10051		
zabbix_server_...	Linux 668c875e-9a39-4d...		
▼ Capabilities			
▼ scalable		Scalable	∞ (∞)
min_instances	1		
max_instances	1		
count	1		
removal_list			
default_instan...	1		



Building/composing Alien4Cloud UI - Text composition




The screenshot displays the Alien4Cloud UI Editor interface. The top navigation bar includes links for Applications, Catalog, and Administration, along with a user profile (admin) and a US flag. The main breadcrumb trail shows TestApp > Environment > Topology > Editor (0.1.0-SNAPSHOT). Below this, a toolbar contains buttons for Save, Download, Undo, Redo, Remote, Pull, and Push. The left sidebar features icons for Topology, Workflow, Inputs and Variables, Archive content, Topology validation, and History. The central workspace shows a file explorer with .git, readme.txt, and topology.yml. The topology.yml file is open in the editor, displaying a YAML configuration for a topology template. The configuration includes metadata, imports, a topology template with node templates (Compute and Kepler), and various properties and capabilities.

```
1  tosca_definitions_version: alien_dsl_2_0_0
2
3  metadata:
4    template_name: TestApp
5    template_version: 0.1.0-SNAPSHOT
6    template_author: admin
7    description: ""
8  imports:
9    - tosca-normative-types:1.0.0-ALIEN20
10   - indigo-types:1.0.0
11
12 topology_template:
13   node_templates:
14     Compute:
15       metadata:
16         a4c_edit_x: 208
17         a4c_edit_y: "-189"
18       type: tosca.nodes.indigo.Compute
19       properties:
20         zabbix_server: "orchestrator.cloud.cnaf.infn.it"
21         zabbix_server_port: 10051
22         zabbix_server_metadata: "Linux        668c875e-9a39-4dc0-a710-17c41376c1e0"
23       capabilities:
24         scalable:
25           properties:
26             min_instances: 1
27             max_instances: 1
28             count: 1
29             default_instances: 1
30         endpoint:
31           properties:
32             private_ip: true
33             protocol: tcp
34             secure: false
35             network_name: PRIVATE
36             initiator: source
37     Kepler:
38       type: tosca.nodes.indigo.Kepler
39       requirements:
40         - hostedOnComputeHost:
41             type_requirement: host
42             node: Compute
43             capability: tosca.capabilities.indigo.Container
44             relationship: tosca.relationships.HostedOn
```


Building/composing Alien4Cloud deploy


ApplicationsCatalogAdministrationadmin 


TestApp>Environment>Info

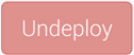
 Deployment init


[Prepare next deployment 0.1.0-SNAPSHOT](#)[Manage current deployment 0.1.0-SNAPSHOT](#)[Deployment History](#)


Info



Runtime view


Workflow




 Reminder of current deployment

Application :	TestApp	Version :	0.1.0-SNAPSHOT
Environment :	Environment	Environment type :	OTHER

 Ressources installed

Compute	(<i>tosca.nodes.indigo.Compute</i>)
Kepler	(<i>tosca.nodes.indigo.Kepler</i>)

 Output properties

No output properties or attributes

Deployed at	Jun 18, 2018 11:23:32 PM
Duration	a few seconds



Thank You

All names, logos, and brands are property of their respective owners in the European Union member states and/or other countries. All company, product and service names used on this presentation can be considered fair use for teaching purposes.