



HTC/HPC TCOM

Miguel CABALLER
UPV



eosc-hub.eu



@EOSC_eu



EOSC-hub receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 777536.

- Summary of the TCOM HPC/HTC
- Requirements and tickets
- Topics for discussion



- User community requirements
 - 8 communities need HTC (Marine, ELIXIR, GEOSS, DODAS, WeNMR, OpenCoastS)
 - 2 explicitly mention the need for running HPC applications (EOS, ECAS/ENES)
- Several communities already identified the services
 - WeNMR, OpenCoastS already identify an HTC job scheduler (DIRAC4EGI)
 - ELIXIR, GEOSS state the need for a container-based or even function-based HTC model which can be supported with an Elastic Kubernetes as a Service plus additional services.
 - ECAS/ENES use its own scheduling system to manage jobs.
 - DODAS use HTCondor.
- The identified services are:
 - EGI Workload Manager (ex DIRAC4EGI).
 - EGI High-Throughput Compute, EGI Cloud Compute and EGI Container Compute.
 - Advanced IaaS, a set of common solutions including Docker support for OpenStack and OpenNebula and on HPC clusters, such as uDocker.
 - PaaS Orchestrator, TOSCA-based deployment orchestration on multiple IaaS, as ECAS/ENES already have their scheduling system based on this technology.

Analysis of Requirements

<https://jira.eosc-hub.eu/secure/Dashboard.jspa?selectPageId=10300>

Filter Results: TCOM HTC/HPC Compute			
T	Key	Summary	P
+	EOSCWP10-86	Raman spectroscopy: provide jupyter notebooks environment	↓
+	EOSCWP10-85	CWL workflow support for LOFAR/RACC	↑
+	EOSCWP10-84	Singularity container support for LOFAR/RACC	↑
+	EOSCWP10-83	CVMFS service for LOFAR/RACC	↑
+	EOSCWP10-68	ECAS/IM Integration	↑
+	EOSCWP10-29	WeNMR RQ1: DIRAC4EGI usage should be integrated into the portals	↑
+	EOSCWP10-18	Multi-site job submission	↑
1-7 of 7			

- Related to the User Communities of ENES, WeNMR, LOFAR/RACC, RAMAN spectroscopy and OpenCoasts.
- Summary of Requirements:
 - Deployment of specific clusters for Data Analytics (EOSCWP10-68) & workflow engines (EOSCWP10-85)
 - Close relation to Cloud Compute, partial relation to workflows and data analytics.
 - Integration of job submission services (DIRAC4EGI), both (even mixed) on HTC Compute and Cloud Compute resources (EOSCWP10-29, EOSCWP10-18).
 - Deployment of container-based jobs (EOSCWP10-83/4).
 - Integration of Jupyter notebooks as front-ends for job submission (EOSCWP10-86).



Within the HTC/HPC TCOM, we identify two macro-features:

1. Capability of submitting HTC jobs with predefined constraints (resources+software) without deploying virtual infrastructure
 - It will require multi-tenant platform services.
 - It will require a transparent management for the allocation of resources in cloud IaaS.
2. Support for the deployment of single-tenant customized HTC or HPC virtual infrastructures.
 - Where a user can instantiate its own cluster, with the need for:
 - Elasticity - requiring credential delegation and secure storage and renewal of credentials if needed;
 - Customization - supporting specific virtual hardware configurations, such as GPUs and of course software configuration.

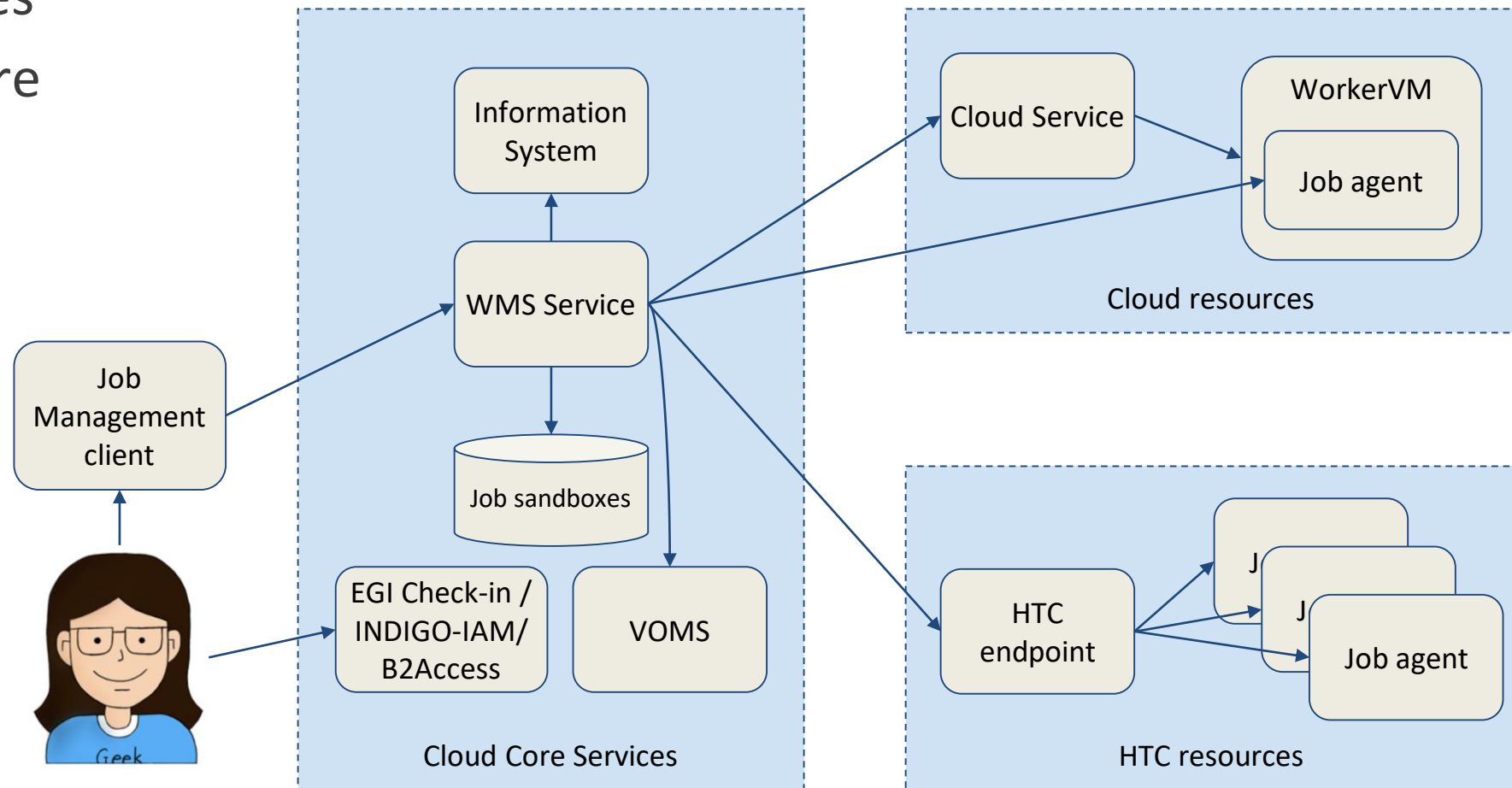


- A service to run a batch job on HTC and cloud compute resources
 - Capable of submitting bunch of jobs to both HTC & cloud compute resources without an explicit deployment.
 - Interface with storage solutions.
 - Seamlessly integrated with the authentication mechanisms.
- Standards
 - Job description: JDL specification.
 - HTC interfaces:
 - CREAM? Others?
 - Cloud interfaces
 - OpenStack Nova API and VOMS with FEDERATION-OS (Keystone API v3) or Tokens OpenID.
- API (REST)
 - Workload management - at least: submit, list, status, kill, log, get-output, describe.
 - Proxy interaction - at least: init, info, destroy, upload.

Macro Feature 1 (a)- Multitenant job submission job submission

Samples of services offering the feature

- DIRAC4EGI
- INDIGO-DC Orchestrator + QCG-Computing.
- HTCONDOR.



 **EOSC-hub**

Macro Feature 1 (b)- Multitenant (container-based) job submission

Add the capability of running containerised jobs to any HTC / Cloud resource, eventually considering public clouds

- A service with similarities to SCAR (FaaS on Batch / Lambda), EGI Fed. Containers or AWS Fargate.
- Jobs described as containers which should run on resources even if there is no container engine installed.
- Eventually accessing data in external storages.

b) Standards, protocols and API

- Container image format: Docker or Singularity images.
- REST API: **XXX**.
- Job Description: K8s Jobs in YAML/JSON?, JDL? OASIS TOSCA Jobs?.
- AAI: OpenID.

EOSC-hub Macro Feature 1 (b)- Multitenant (container-based) job submission

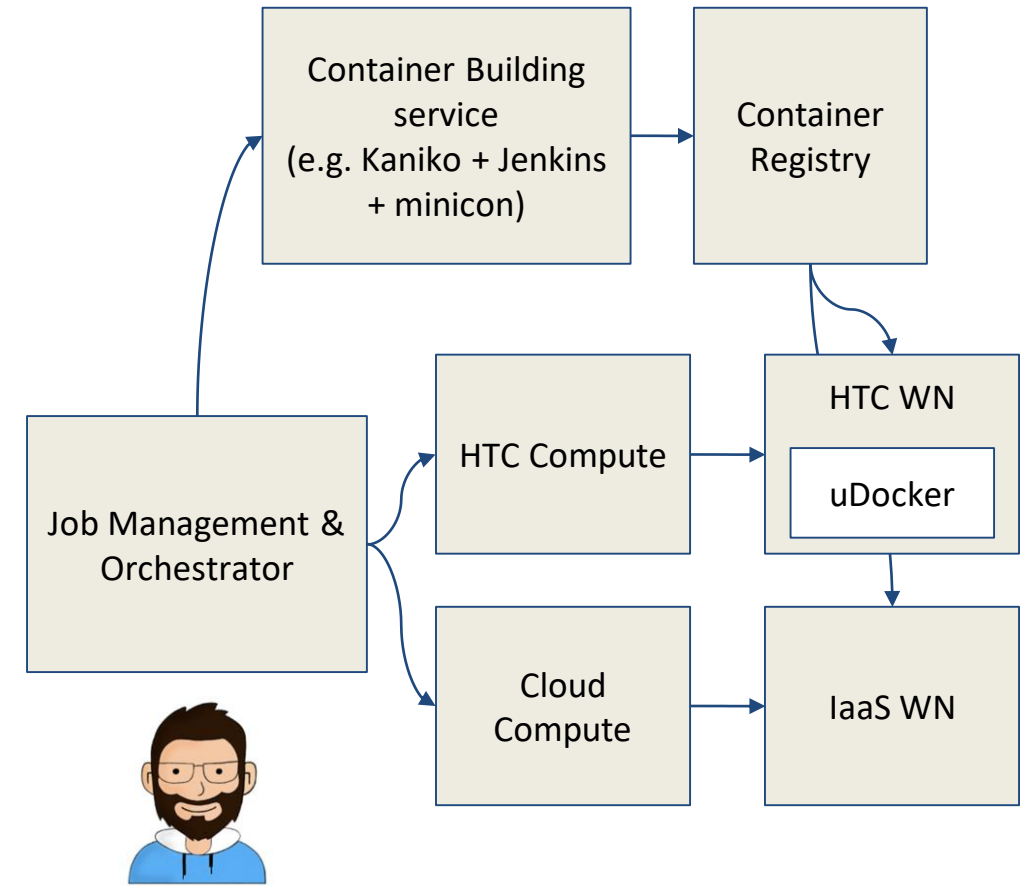
c) Interoperability guidelines

- Support for Singularity or no Container engine (uDocker).
- Potentially supporting public clouds.

d) samples of services offering the feature

- SCAR - Serverless Container-aware ARchitectures
(<https://github.com/grycap/scar>)

Architecture (additional wrt 1a)



Several user communities are demanding a convenient way of deploying a back-end cluster on demand

- Similar to Amazon EKS.
- However more general as there are different load balancers.
- This complements the previous use case as the environment is isolated and single-tenant.

Description

- A convenient service for defining the features of the cluster among a set of choices for the LRMS and base software configurations.
- Ideally self-managed and extensible by the user.
- Tailored to deal with specific HPC hardware if available, including both computing accelerators and low-latency and high-bandwidth networks.

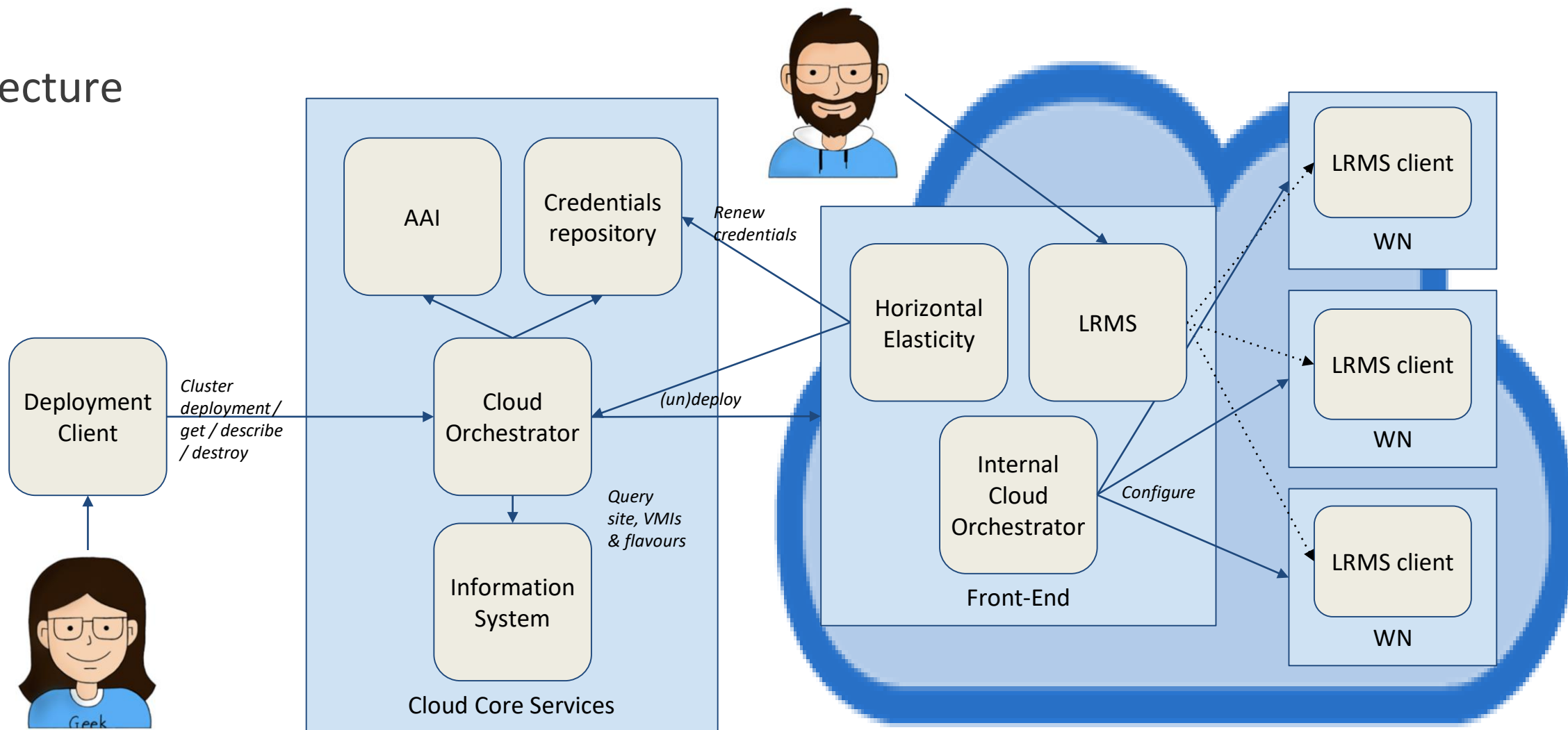
b) standards and protocols

- For the AAI: OpenID - EGI Check-in, INDIGO-IAM, B2Access.
- For the IaaS: OpenStack Nova + OpenNebula.
- For the description of the Clusters: OASIS TOSCA
- API:
 - REST + CLI and GUI using the REST API.
 - At least: Deploy, list, describe, destroy, reconfigure, update.

c) interoperability guidelines

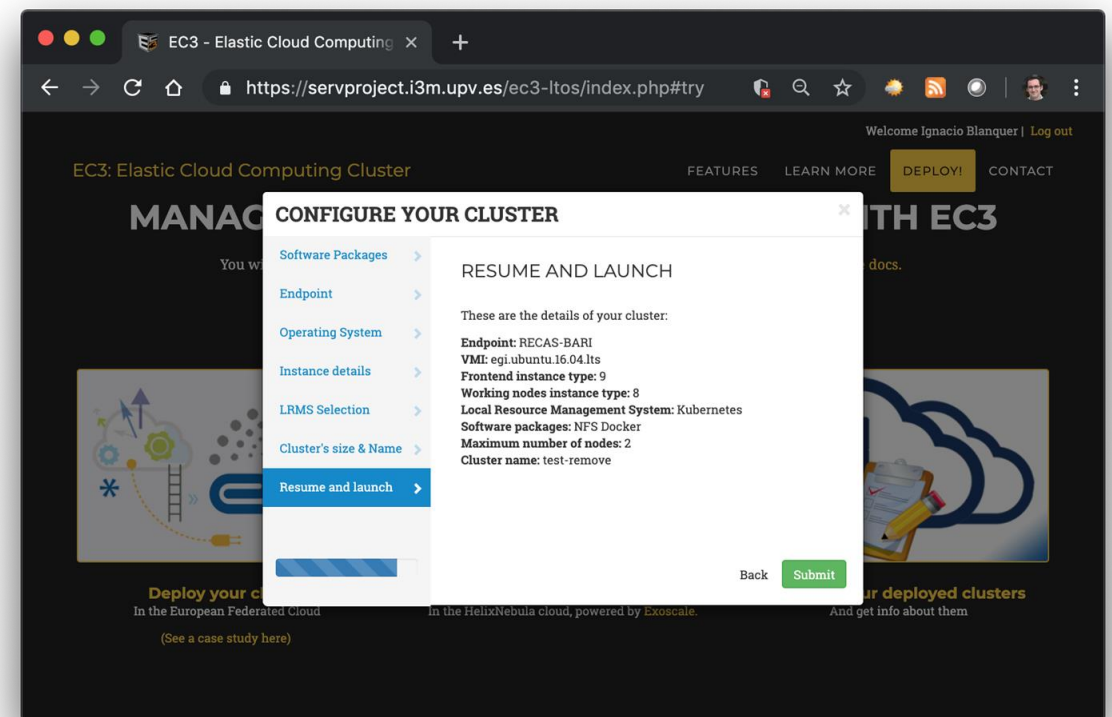
- Integrated with the AAI.
- Integrated with the WMOps Dashboard
- Integrated with the IaaS plugins.

Architecture



d) samples of services offering the feature

- EC3 Dashboard in EGI Access (<https://servproject.i3m.upv.es/ec3-ltos/index.php>)





Integration of HTC services on top of the Federated cloud resources

- To increase the usability of IaaS as well as provide seamless transition between different infrastructure types
 - Integration of HTC workload management services on Federated Cloud resources.
 - Homogeneous support of containers for HTC and Cloud infrastructures.
 - Workflow engines for HTC and Cloud infrastructures.
 - Integration of specific resources to build up HPC clusters (e.g. Infiniband networks, GPUs).