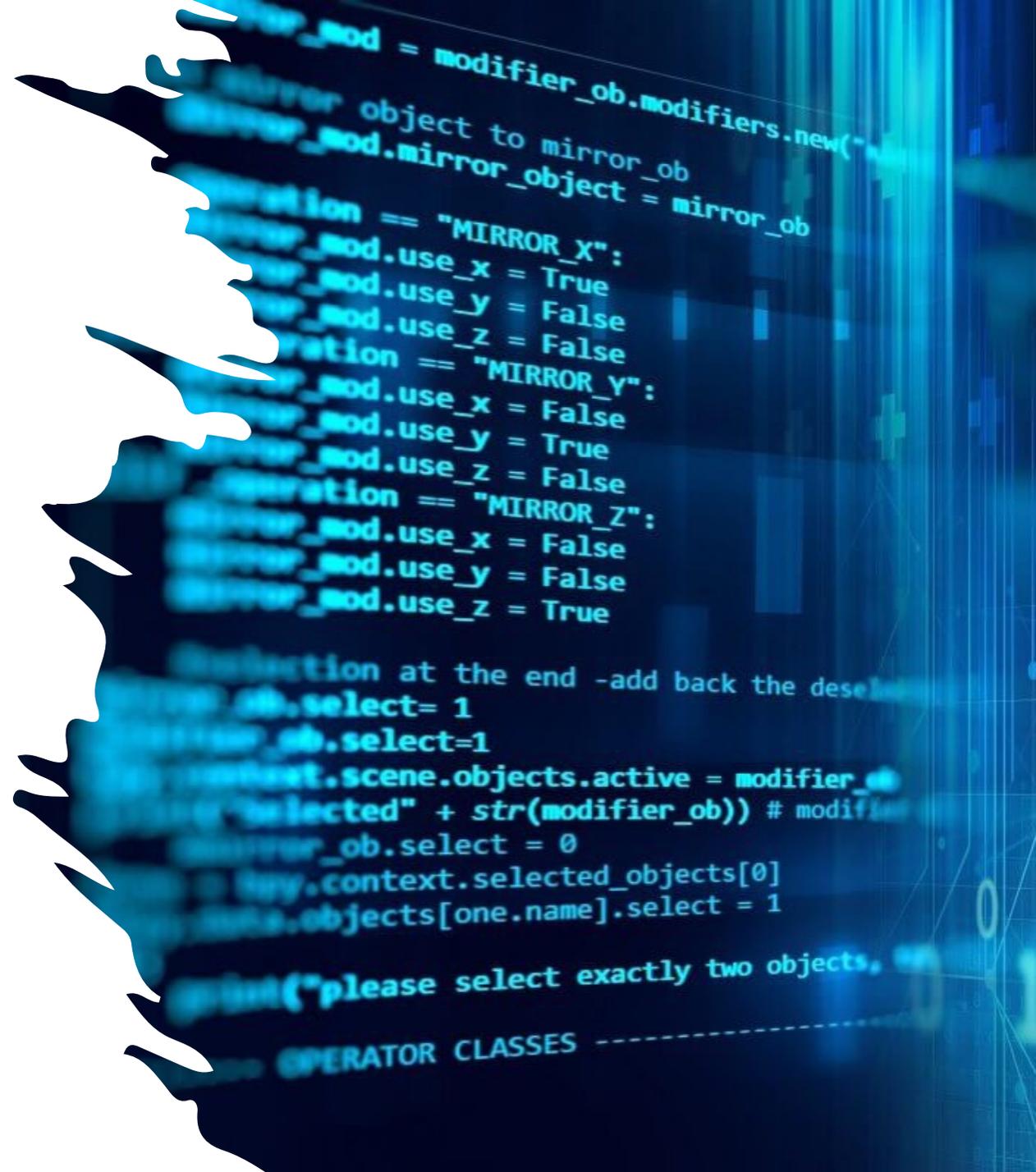


Evoluzione del sistema di orchestrazione PaaS: stato e criticità

MARICA ANTONACCI

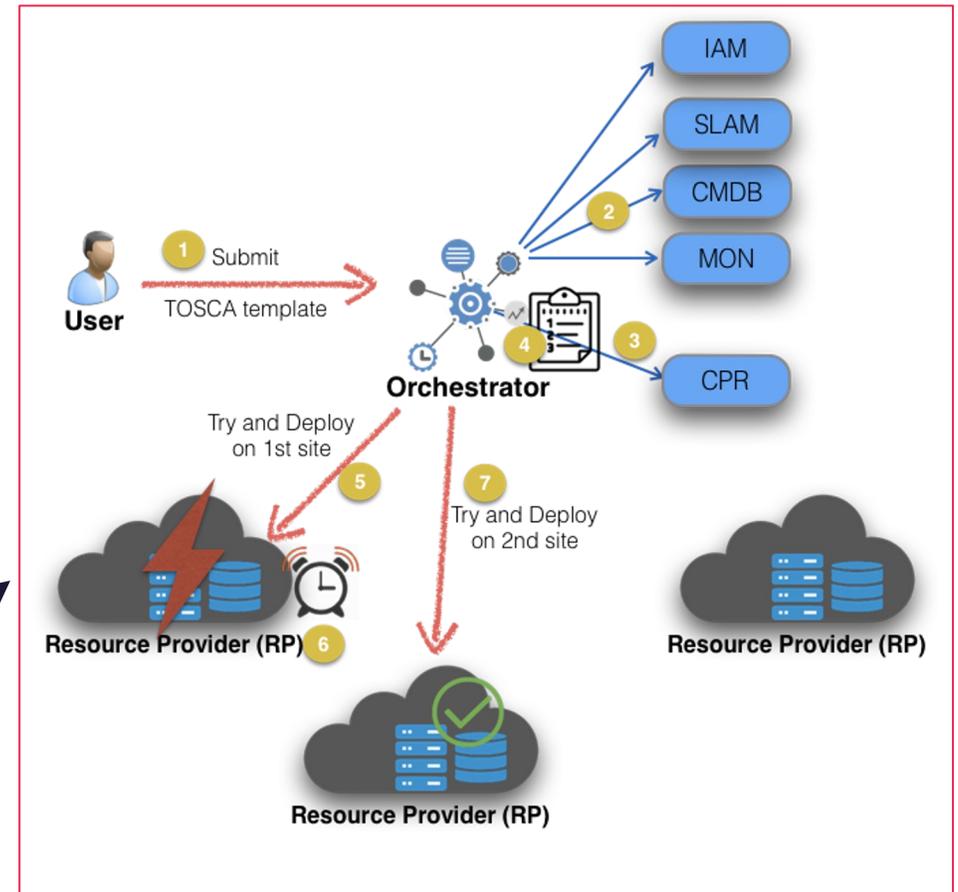
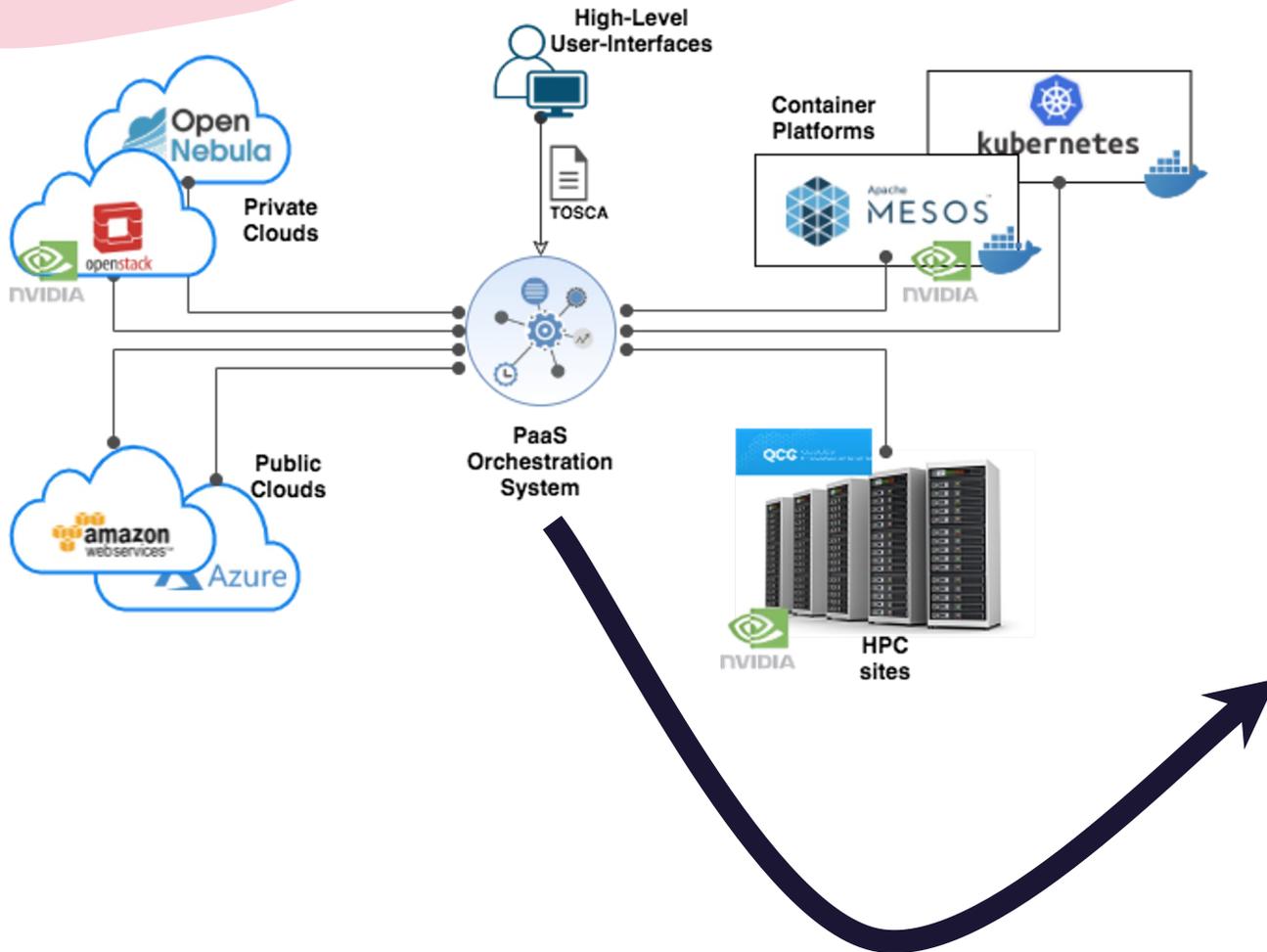
Meeting 10/04/2024



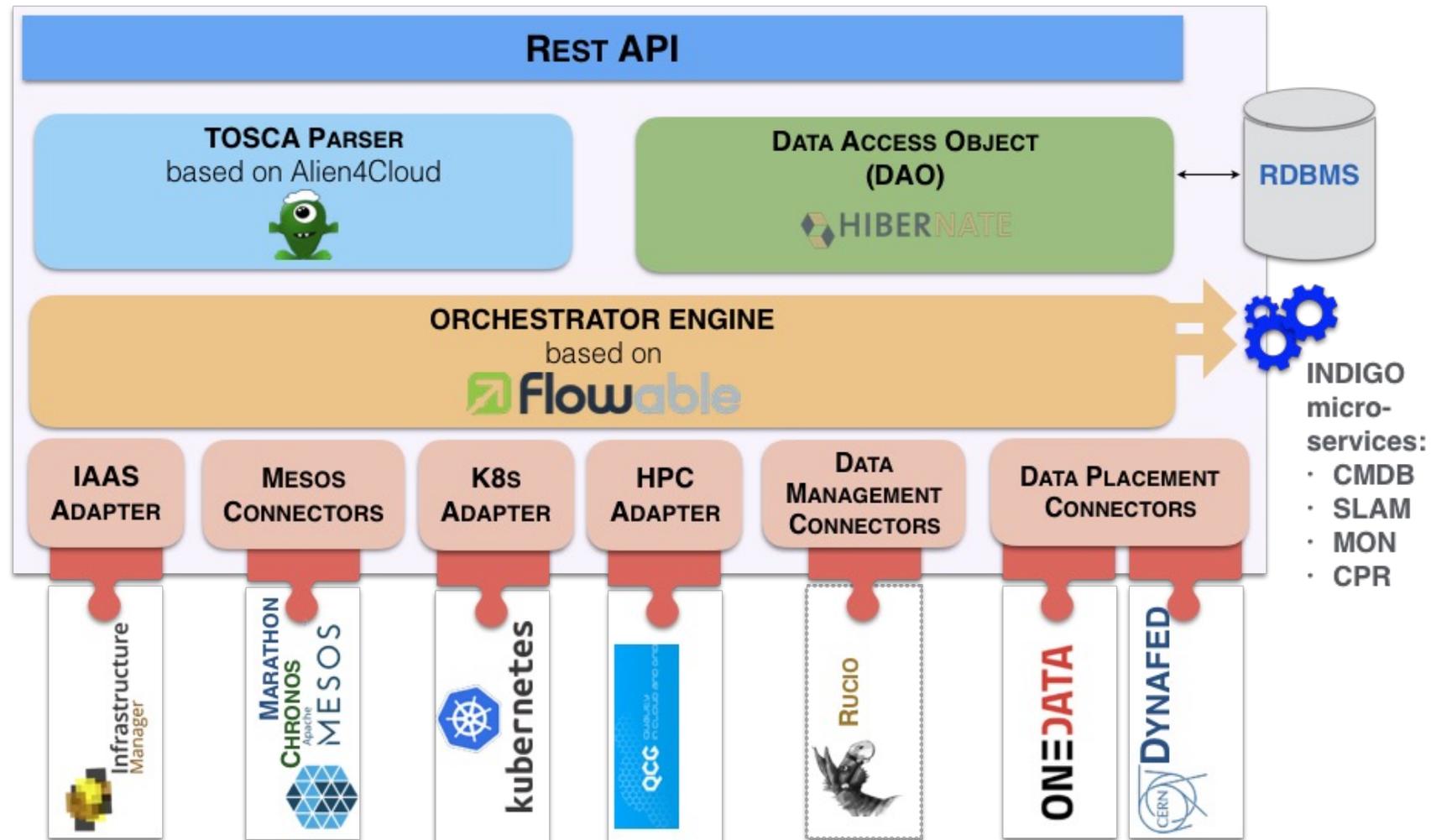
Background

- PON PRISMA (2012-2015)
 - OR1: Studio e realizzazione infra IaaS (INFN)
 - **OR2: Studio e realizzazione infra PaaS (Santer Reply)**
- INDIGO-DataCloud (2015-2017)
 - Reply (partner) rimane il principale responsabile dello sviluppo dell'Orchestratore, ma l'architettura evolve verso una a micro-servizi
- DEEP/XDC/EOSC-Hub (2017-2019)
 - Reply è subcontractor. INFN Bari comincia a contribuire allo sviluppo e nasce la dashboard dell'orchestratore
- INFN Cloud/interTwin/AI4EOSC (2019 - oggi)
 - Gli sviluppi continuano in DataCloud WP5

Il sistema di orchestrazione PaaS



Orchestrator high-level architecture



Stato attuale dei componenti PaaS

CMDB (java)/CIP (python 2.7) → Reimplementati in python

SLAM (java) → in fase di riscrittura

Monitoring (Zabbix + Java) → in fase di progettazione

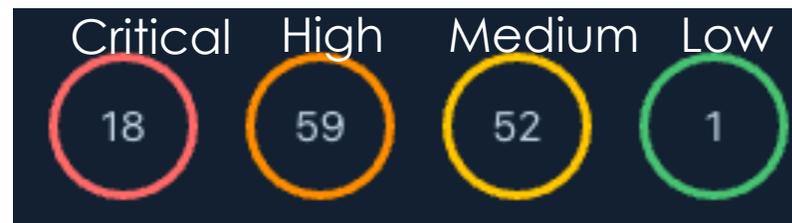
Orchestratore:

Java Spring Boot (1.5.21) Application

261 componenti (40 direct, 31 out of 41 outdated)

TOSCA Parser basato su un fork di alien4cloud (3.0.x)

Java 1.8 EOL November 2026



Attività in corso sull'orchestratore

- AI-based orchestrator
 - Progetto interTwin
- Piccole migliorie
 - Gestione automatica client iam e bucket S3
- Correzione di bug

Plugin per kubernetes

- Necessità di federare siti che gestiscono le risorse NON tramite Openstack, MA tramite k8s
- Il PoC implementato in EOSC-hub si basa su componenti legacy (deprecati) per cui è necessario studiare una soluzione basata su tecnologie più nuove
- Working group su kubernetes
 - Kick-off meeting 16/02/2024
 - Meeting bi-settimanali
 - Partecipazione circa 25 persone

Risorse attuali

- Al momento sull'**Orchestratore** è stata allocata **una sola risorsa (CNAF)** - non era uno sviluppatore java
- Per gli **altri sviluppi in python** è stata allocata **una risorsa a Bari**
- Si riscontra, in generale, scarso interesse nel contribuire allo sviluppo dell'Orchestratore (forse perchè in Java) e dei servizi (TOSCA e ansible – nonostante seminari e guide/howto)
 - Attività più semplici come i test sono facilmente coperte – richiedono meno competenze!
- Mancanza di informazioni complete sull'effettivo effort che le persone possono fornire alle attività di sviluppo, essendo coinvolte in altri progetti locali, nazionali, europei, ...

Possibili strategie?

- Porting del codice a Python
 - Python è più semplice di Java ed è più facile (forse) trovare persone che abbiano esperienza con il linguaggio
 - Opportunità di usare lo stesso toska-parser usato da IM
- Investire in altro, per esempio l'Infrastructure Manager (python), contribuendo allo sviluppo delle funzionalità che ci servono
 - UPV ha anche «copiato» la dashboard dell'orchestratore 😊
- Altro?

Cosa ci serve in ogni caso...sono le RISORSE UMANE e...

- Evitare di lavorare in silos
 - condivisione di codice, esperienze, informazioni per garantire tempistiche e avanzamento attività
- Sviluppo di competenze software e non solo (p.e. security per lo sviluppo sicuro → certificazione)
 - A volte ci sono gravi carenze anche nell'utilizzo di tool come git → necessità di corsi di formazione
- Sviluppo di soft skills come autonomia e attitudine al team-working