



INDIGO - DataCloud

Global Architecture

Alfonso Pérez, Germán Moltó

Universitat Politècnica de València



INDIGO-DataCloud is co-funded by the
Horizon 2020 Framework Programme

Index

- Overview of modules and components
- Deployment of customized virtual infrastructures
- Deployment of a managed service/application (Mesos clusters)
 - Long Running Services on Marathon
 - Batch jobs on Chronos
- User interface

Overview of modules and components



INDIGO - DataCloud

- **Orchestrator:**
 - Core component of the PaaS layer.
 - Receives high-level deployment requests
 - Coordinates the deployment process over the IaaS platforms.
- **Identity and Access Management (IAM) Service:**
 - Provides a management layer for:
 - Identities
 - Enrolment
 - Group membership
 - Attributes and policies to access distributed resources and services

Overview of modules and components

- **Monitoring Service:**

- Collects monitoring data from the targeted clouds.
- Analyze and transform the data into information to be consumed by the Orchestrator.

- **Brokering/Policy Service:**

- Rule-based engine that allows managing the users priority, fair-share, etc. per site.
- Prepares the list of IaaS sites where the requested resources might be deployed.
- The final decision will be made by the Orchestrator taking into account also the monitoring data and the QoS/SLA information.

Overview of modules and components

- **QoS/SLA Management Service:**

- Allows the handshake between a users and a site on a given SLA.
- Describes the QoS that a specific user/group has both over a given site or generally in the PaaS as a whole:
 - Priorities for a given user, i.e. the capability to access different levels of QoS at each site (Gold, Silver, Bronze services);

- **Managed Service/Application (MSA) Deployment Service:**

- In charge of scheduling, spawning, executing and monitoring applications and services on a distributed infrastructure.
- Composed by an elastic Mesos cluster with slave nodes dynamically provisioned and distributed on the IaaS sites.

Overview of modules and components



- **Infrastructure Manager (IM):**

- Deploys complex and customized virtual infrastructures on IaaS Cloud.
- Provides an abstraction layer to define and provision *resources* in different clouds and virtualization platforms.

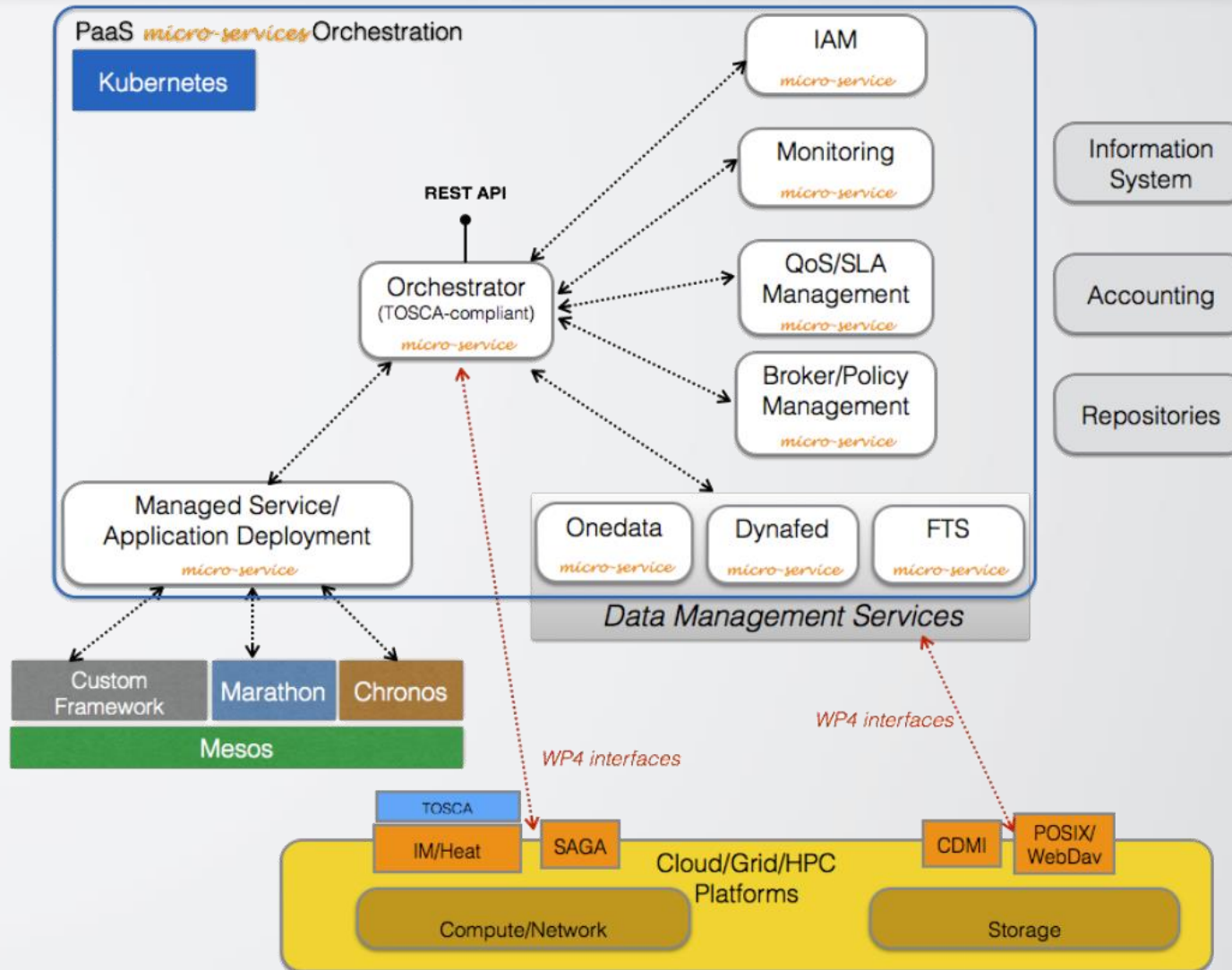
- **Data Management Services:**

- Collection of services that provide an abstraction layer for accessing the data storage in a unified and federated way.

Overview of modules and components



INDIGO - DataCloud



Deployment of customized virtual infrastructures



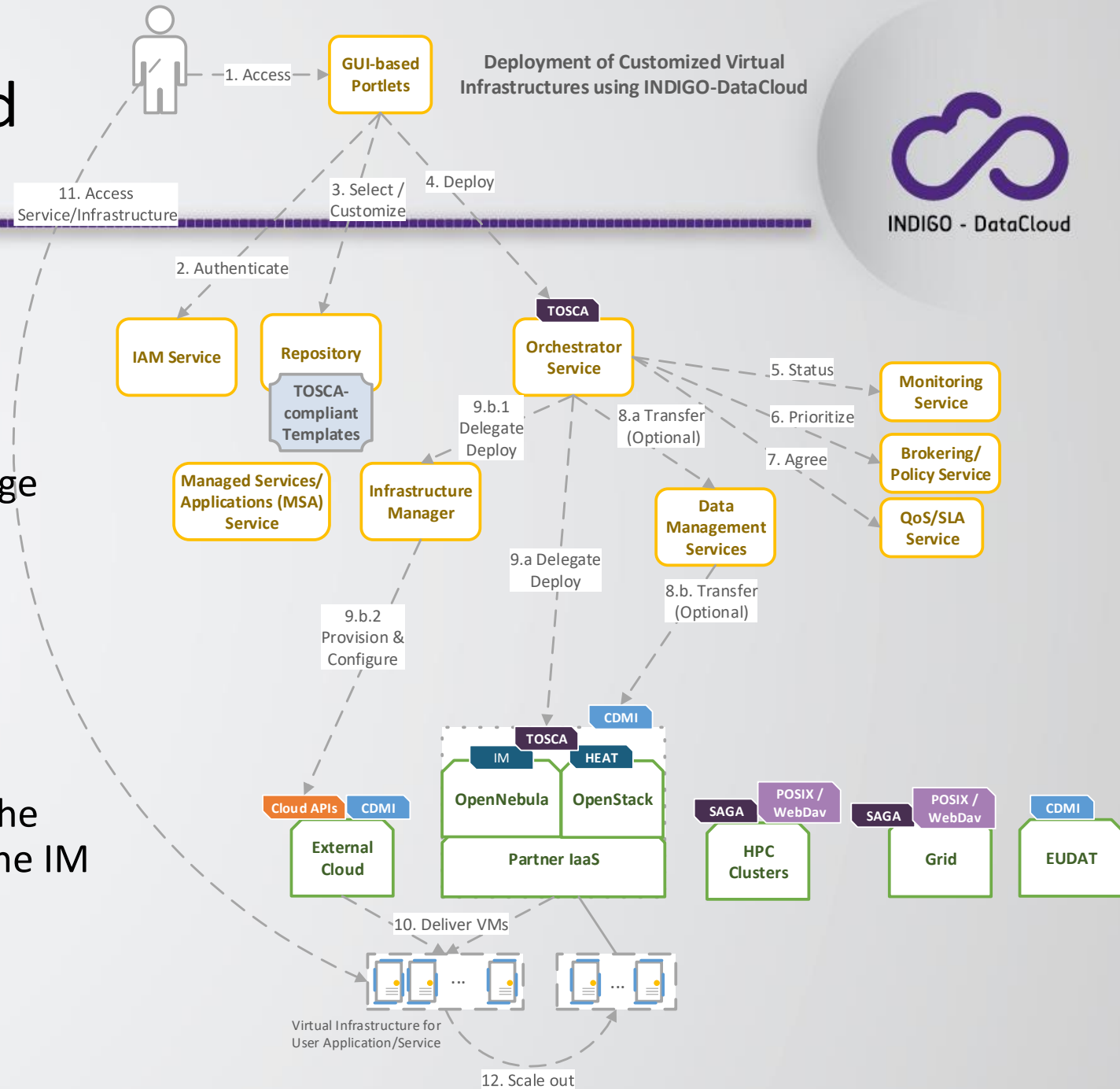
- The components cooperate together in order to provide the ability to describe and deploy services/applications on heterogeneous environments.
- The user will access the PaaS layer services through high-level interfaces.
- The INDIGO repository will provide a catalogue of pre-configured TOSCA templates:
 - Allows the deployment of a wide range of applications and services.
 - Customizable with different requirements of scalability, reliability and performance.
- The selected template will be submitted to the PaaS Orchestrator using its REST API endpoint.

Deployment of customized virtual infrastructures

- The Orchestrator collects all the information needed to generate the deployment workflow:
 - Health status and capabilities of the underlying IaaS platforms and their resource availability from the Monitoring Service.
 - Priority list of sites sorted by the Brokering/Policy Service.
 - QoS/SLA constraints from the SLA Management System.
 - The status of the data files and storage resources needed by the service/application and managed by the Data Management Service.
- This information will be used to perform the matchmaking process and to decide where to deploy each service.

Deployment of customized virtual infrastructures

- The user is able to access the deployed customized virtual infrastructure:
 - The user can run/administer/manage applications running on it
- When a customized virtual infrastructure deployment is requested:
 - The Orchestrator manages the instantiation and configuration of the required resources using HEAT or the IM



Deployment of a managed service/application



- When a managed PaaS service deployment is requested:
 - The Orchestrator interacts with the Managed Service/Application (MSA) Deployment Service.
 - It supervises the deployment on the *elastic* Mesos cluster that hosts the user application/service.
- The Mesos cluster consists of:
 - One or more master nodes (one is elected as leader in high-availability mode).
 - Slave nodes that register with the master and offer resources, i.e. capacity to be able to run tasks.

Deployment of a managed service/application



- The master node knows the state of the whole cluster:
 - It can share and assign resources to the different applications (i.e. frameworks) based on a pluggable scheduling policy (fair share, strict priority, etc.).
 - The Automatic Scaling Service, based on EC3/CLUES, ensures the elasticity and scalability.
- When additional computing resources (worker nodes) are needed:
 - The Orchestrator is requested to deploy them on the underlying IaaS matching the QoS/SLA, health and user/group/use-case policies agreed by the Broker.
 - These new worker nodes are configured in order to join the cluster as new Mesos slaves where the services/applications will run.

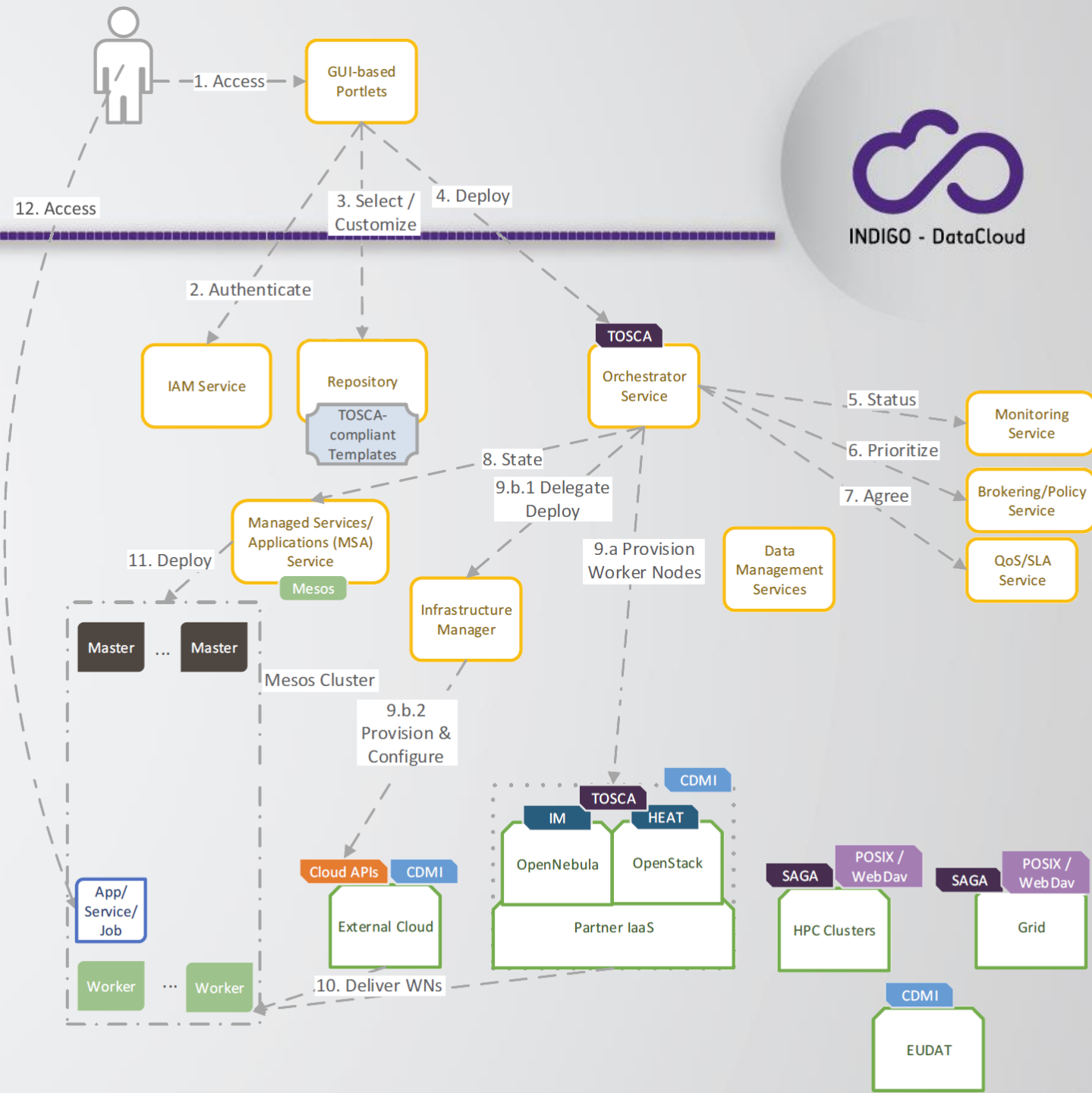
Deployment of a managed service/application



- In case of Long Running Services:
 - The MSA will use Marathon to ensure that the services are always up and running.
 - Marathon is able to deal with different issues:
 - Restart of the services.
 - Migration in the event of problems.
 - Dependency among them.
 - Load-balancing.
 - ...
- The MSA Deployment Service will use Chronos to execute a given application with the definition of input/output/dependency, etc.
 - Chronos may also handle the rescheduling of failed applications, or simple workflow made by different applications.
- New Frameworks can be developed in order to meet specific use-cases:
 - E.g. one able to deploy a batch cluster (like HTCondor) on demand.

Deployment of a managed service/application

- In this scenario the user will be returned the list of endpoints to access the deployed services.



User Interface

- The PaaS Core is not intended to be directly accessed by users.
- The user should use the Graphical User Interfaces or simple APIs.
- The PaaS Core provides an entry point to its functionality via the Orchestrator service:
 - RESTful API that receives a TOSCA-compliant description of the application architecture to be deployed.
- Regarding Data Management Services, interfaces are provided to advanced users for specific data management tasks:
 - OneData provides its web based interface for managing user spaces (virtual folders) and controlling access rights to files on a fine grained level.