

# Earth Observation on Clouds

Use case: **GISHEO** (as a legacy app.) using  
mOSAIC

leAT mOSAIC Team:

Prof. Dana Petcu [petcu@info.uvt.ro](mailto:petcu@info.uvt.ro) - Leader

Ciprian Craciun [ccraciun@info.uvt.ro](mailto:ccraciun@info.uvt.ro)

Marian Neagul [marian@info.uvt.ro](mailto:marian@info.uvt.ro)

**Silviu Panica** [silviu@info.uvt.ro](mailto:silviu@info.uvt.ro)



# Outline

- GiSHEO Project
- GiSHEO current architecture
  - components
  - technologies
- GiSHEO proposed migration levels
- proof of concept: L3 GiSHEO Migration

## **GiSHEO Platform DEMO**

# GiSHEO Project – Technical Objectives

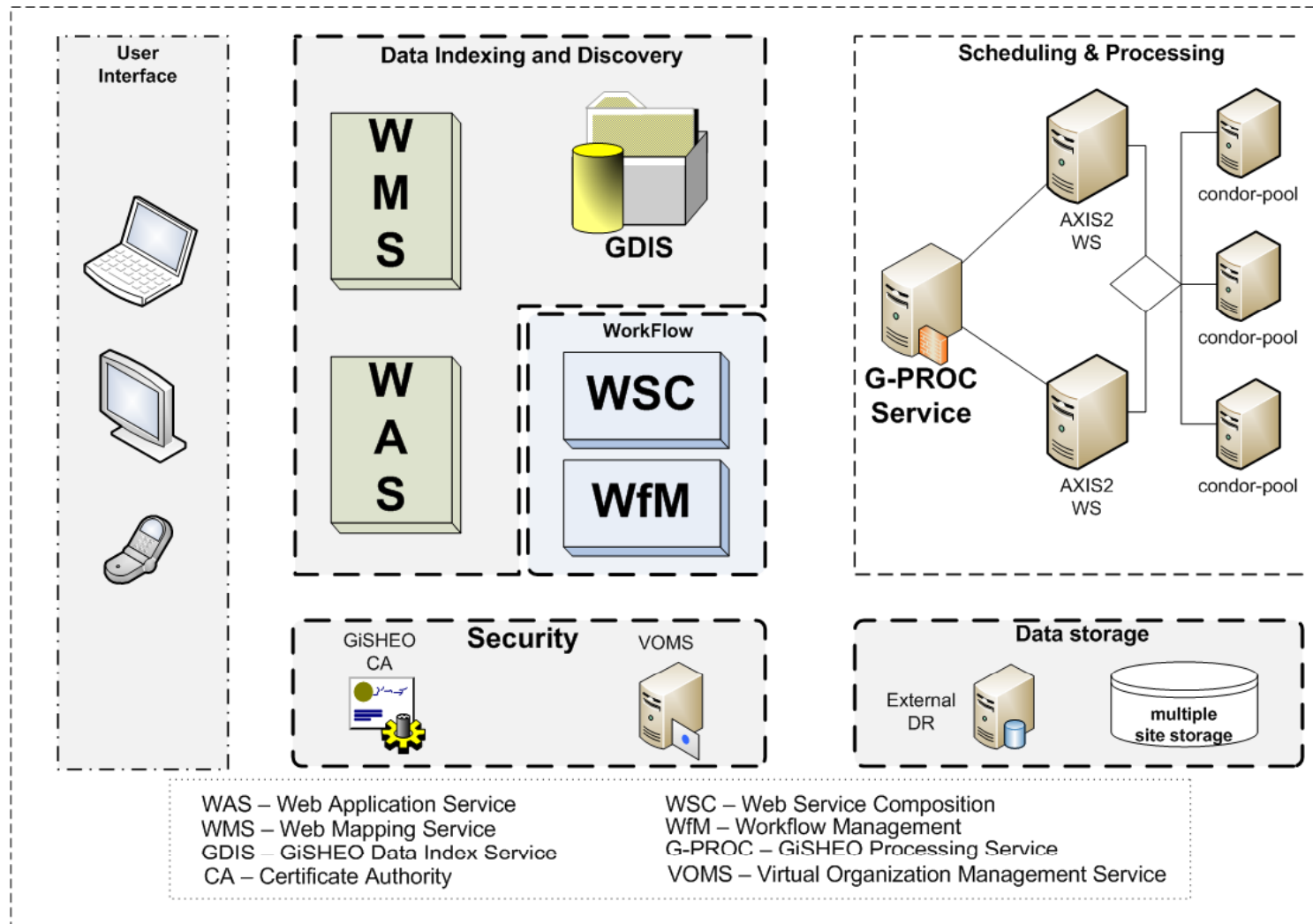
- set-up and organize a virtual organization (VO) based on Grid technology for education, training and knowledge dissemination for Earth observation (EO)
- development of specific instruments dedicated to on-demand services for education activities based on Earth observation information;
- facilitate specific access for the academic and scientific community to on-demand services related to specific applications using ESA database and facilitate synergies;
- Correlation and harmonization of resources with specific ESA projects dedicated to on demand services for earth observation (GPOD).

**<http://gisheo.info.uvt.ro/>**

# GiSHEO Platform

- grid-enabled platform
- components
  - security
  - scheduling and processing
  - workflow engine
  - data indexing and discovery
  - data storage

# GiSHEO Architecture



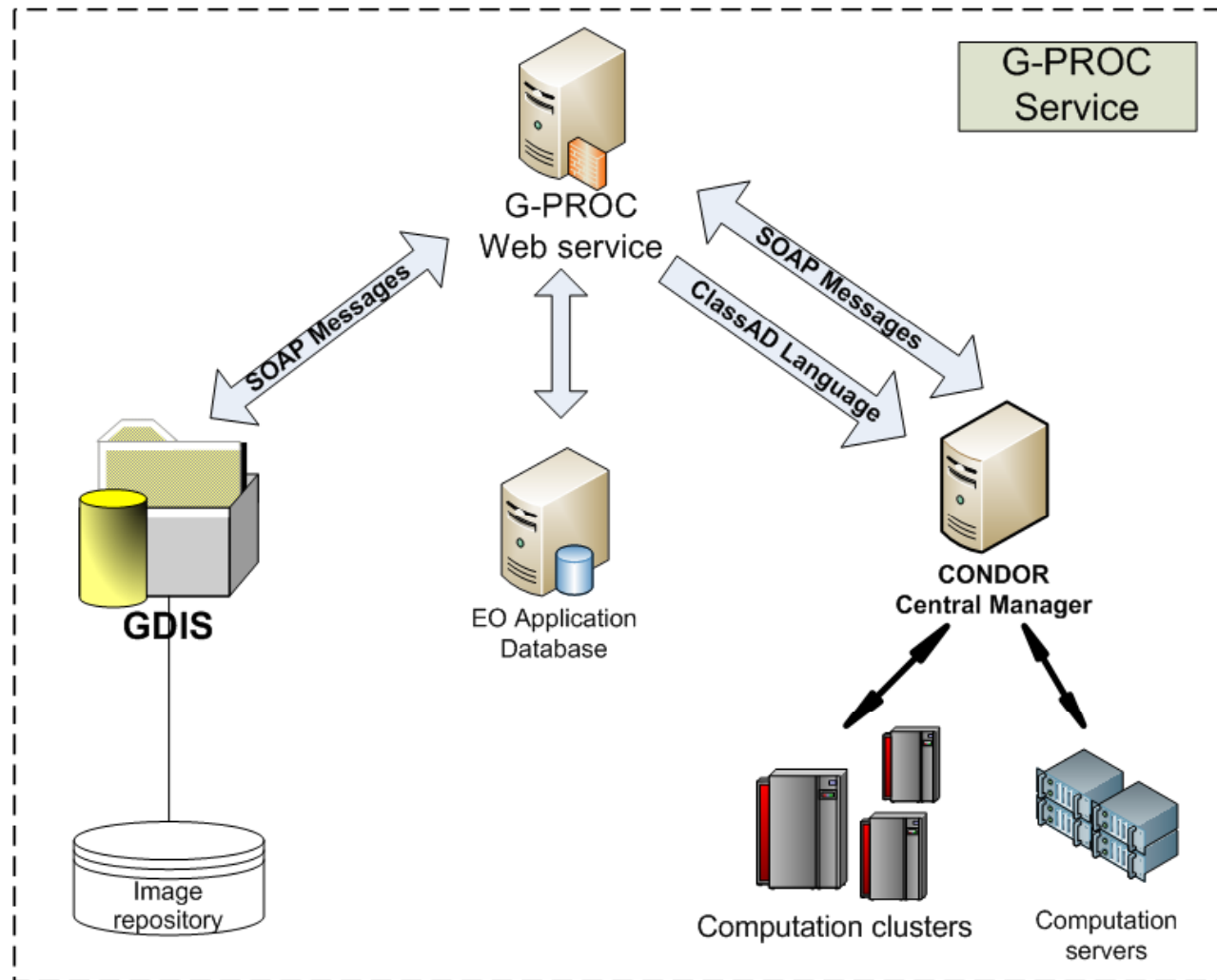
# Security component

- authentication
  - X.509 digitally signed certificates;
  - issued by a trusted CA (GiSHEO CA or EUGridPMA affiliated CA);
- authorization
  - VO Membership;
  - extended attributes support: embedded into the X.509 certificate;
  - VOMS Service;
- credentials delegation
  - using MyProxy service;

# Scheduling and processing

- G-PROC (GiSHEO Processing Service): the interface
  - access to EO application database;
  - prepares and submits jobs to the Resource Manager;
  - interacts with other internal services;
  - WS compliant (using AXIS2 engine)
- WMS
  - Condor HTC (using multiple pools)
  - two resource managers supported:
    - Condor-WS;
    - GT4 GRAM (deprecated);

# Processing architecture



# Workflow composition engine

- Each image processing **operation** is viewed as a single **task**
- Several tasks can be linked together to form a **workflow** whose order is decided at client side
- The place where tasks are executed is decided by a **Discovery Service** whose role is to *select* appropriate services (which can run the desired task) and to return the list to the engine
- The engine is based on an ECA (Event-Condition-Action) approach which offers much more **dynamism** and **adaptability** to changes in workflow tasks and resource states than other classic workflow engines. It is also less verbose than classic XML approaches.
- DROOLS was chosen as the main rule based inference engine

# Storage architecture

- data distribution
  - local disk storage:
    - local FS: ext3;
  - distributed storage:
    - Hadoop FS (distributed FS);
- unique access interface: FTP with SSL auth (like VSFTPd)
  - also GridFTP Server with GSI is supported;

# Storage: data indexing

- Data Indexing component;
- using a RDBMS: PostGRES with PostGIS support;
- what offers:
  - stores processing result information;
  - stores satellite images meta information;
  - offers information about satellite images (geographical information);

# Storage: data query

- using a custom made query language: LLQL (Lisp Like Query Language)
- syntax is inspired from the LISP language and partially from LDAP filters;
- for regular usage a simple query language is available;
- it is similar to the language used by the search engines:

*vendor:NASA type:DEM*

*place:Timisoara, Timis, Romania*

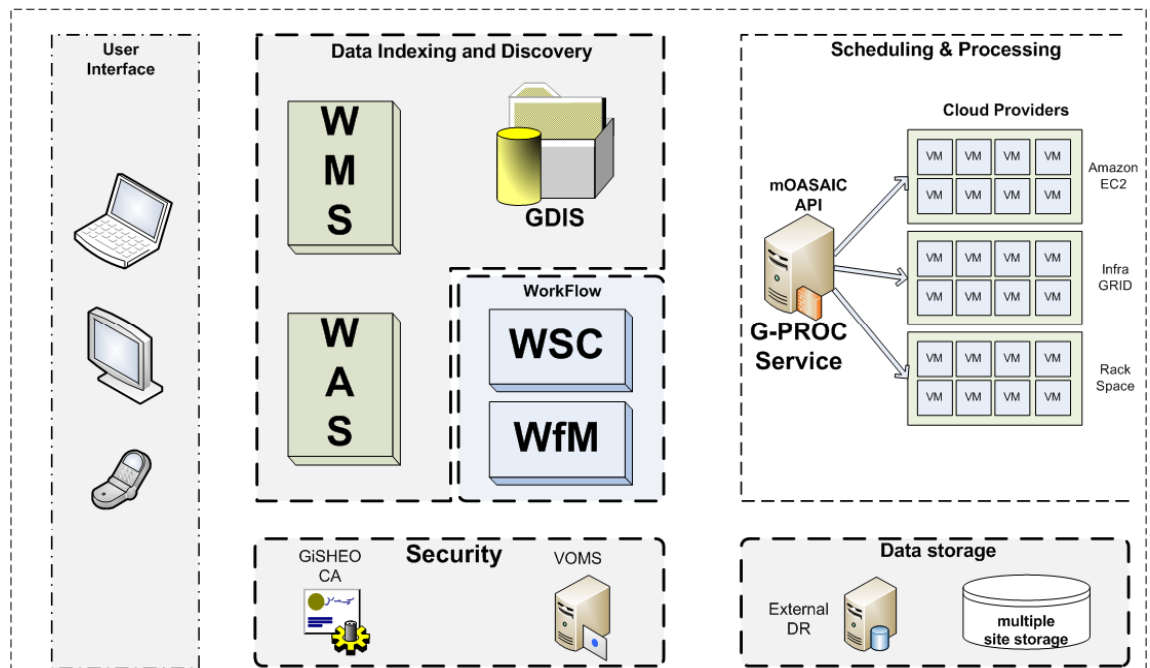
- each query regarding PostGIS information is translated into PostgreSQL query;

# GiSHEO2Cloud: Migration

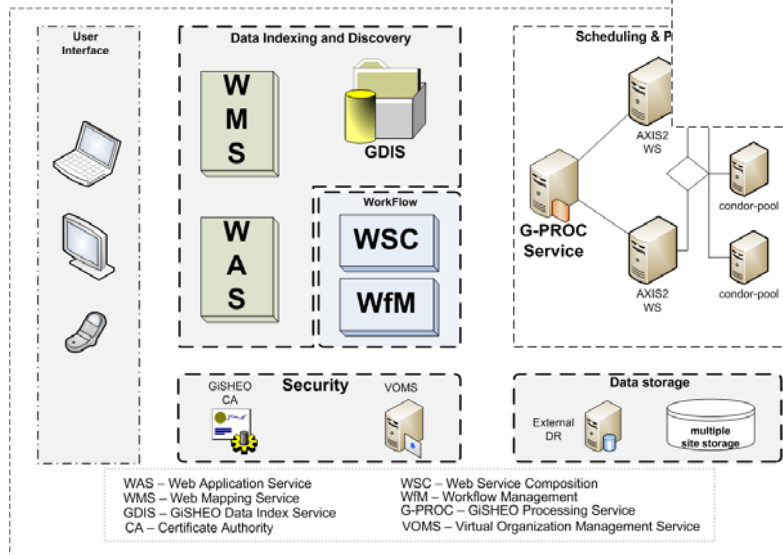
- Level 1 : GiSHEO Processing Resources on clouds
  - only worker nodes;
  - entire WMS is moved on clouds;
- Level 2 : GiSHEO using mOSAIC resource broker for setting up a scalable infrastructure;
  - entire processing platform is moved on clouds;
- Level 3: GiSHEO on Clouds using mOSAIC
  - re-written GiSHEO components to suit the mOSAIC legacy cloud model;

# G2Cloud: Level 1

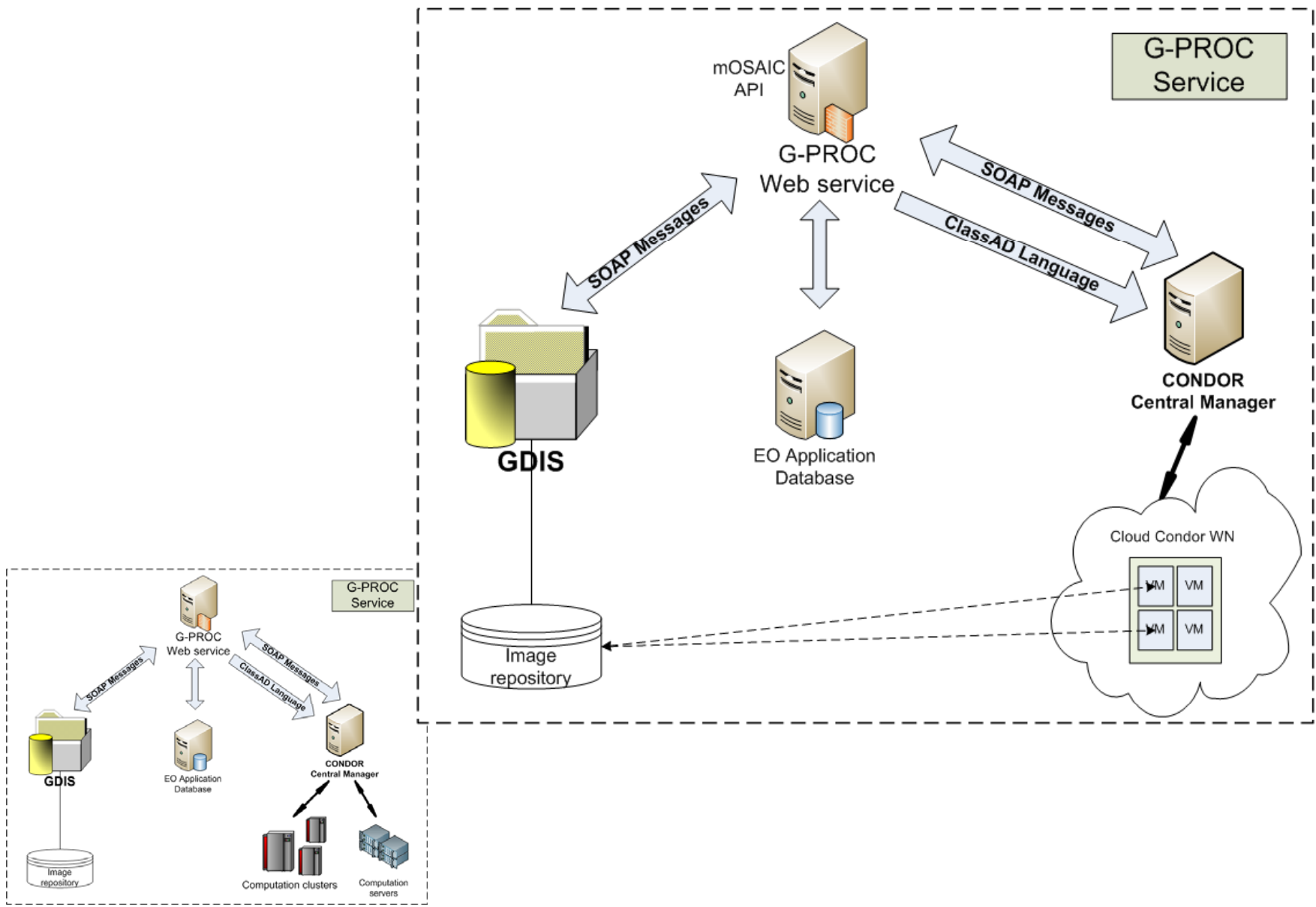
- first approach: classical worker nodes are moved in clouds for scalability;
- second approach: entire processing platform is moved into the clouds (WMS and worker nodes);

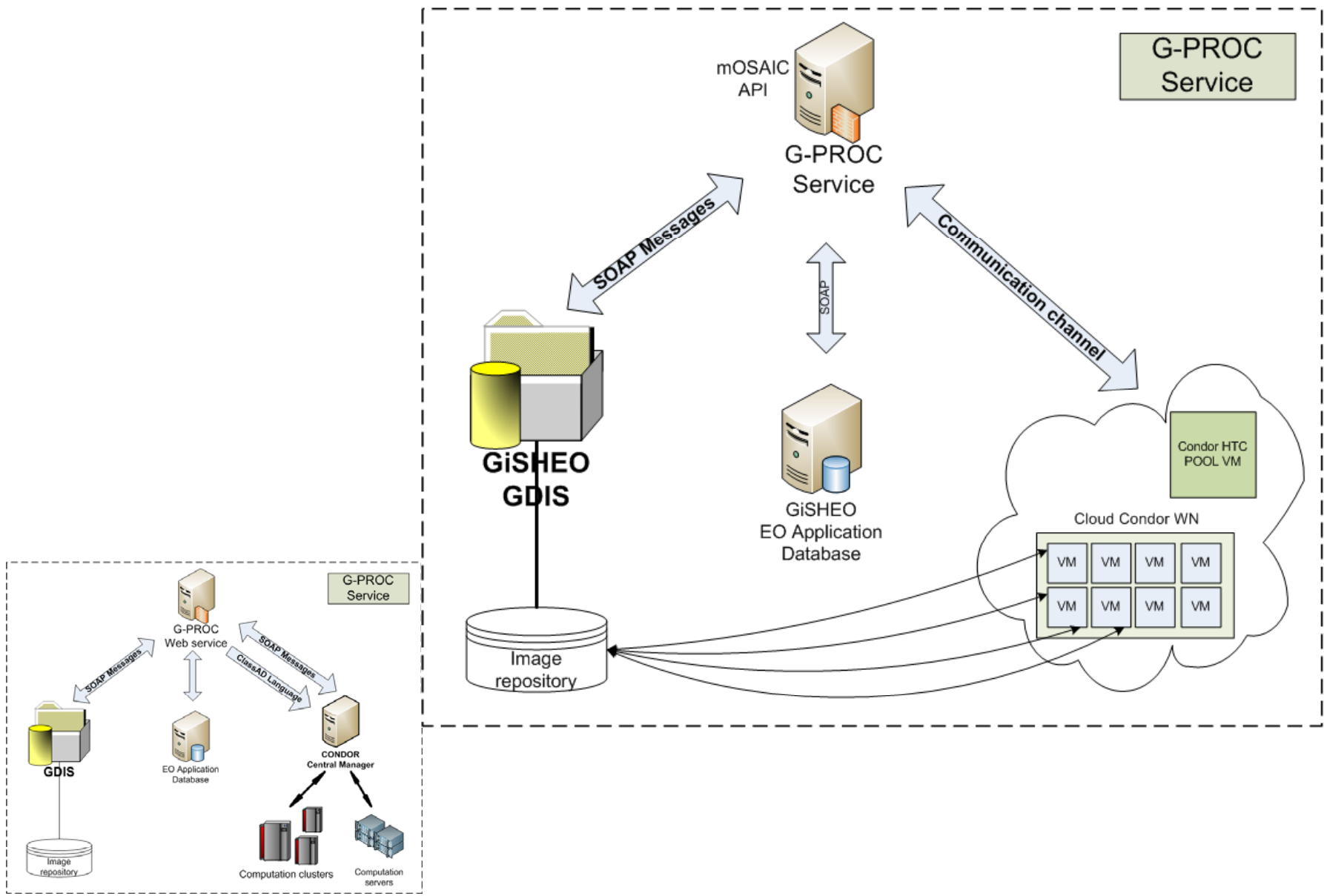


WAS – Web Application Service  
 WMS – Web Mapping Service  
 GDIS – GISHEO Data Index Service  
 CA – Certificate Authority  
 WSC – Web Service Composition  
 WfM – Workflow Management  
 G-PROC – GISHEO Processing Service  
 VOMS – Virtual Organization Management Service



WAS – Web Application Service  
 WMS – Web Mapping Service  
 GDIS – GISHEO Data Index Service  
 CA – Certificate Authority  
 WSC – Web Service Composition  
 WfM – Workflow Management  
 G-PROC – GISHEO Processing Service  
 VOMS – Virtual Organization Management Service

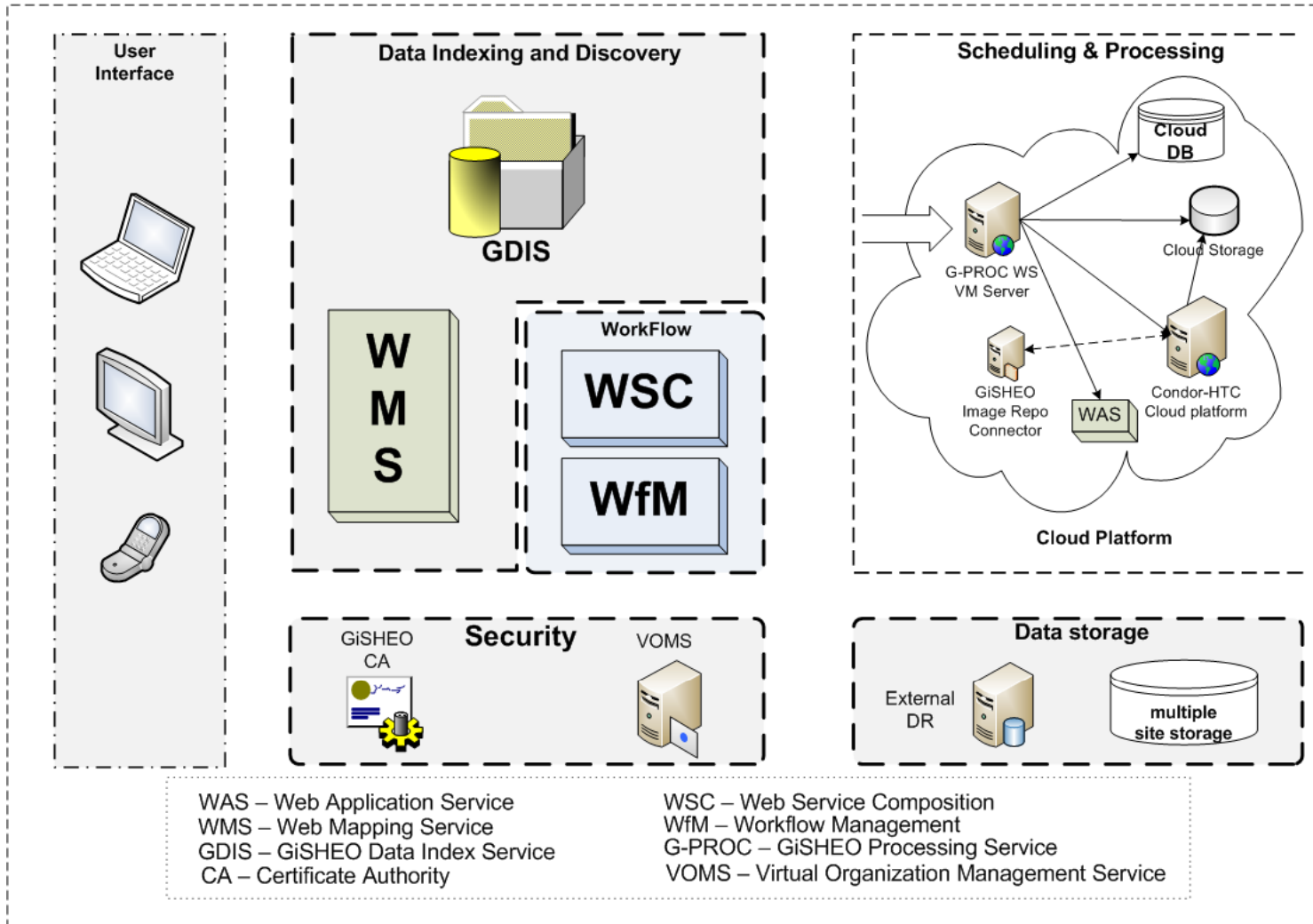




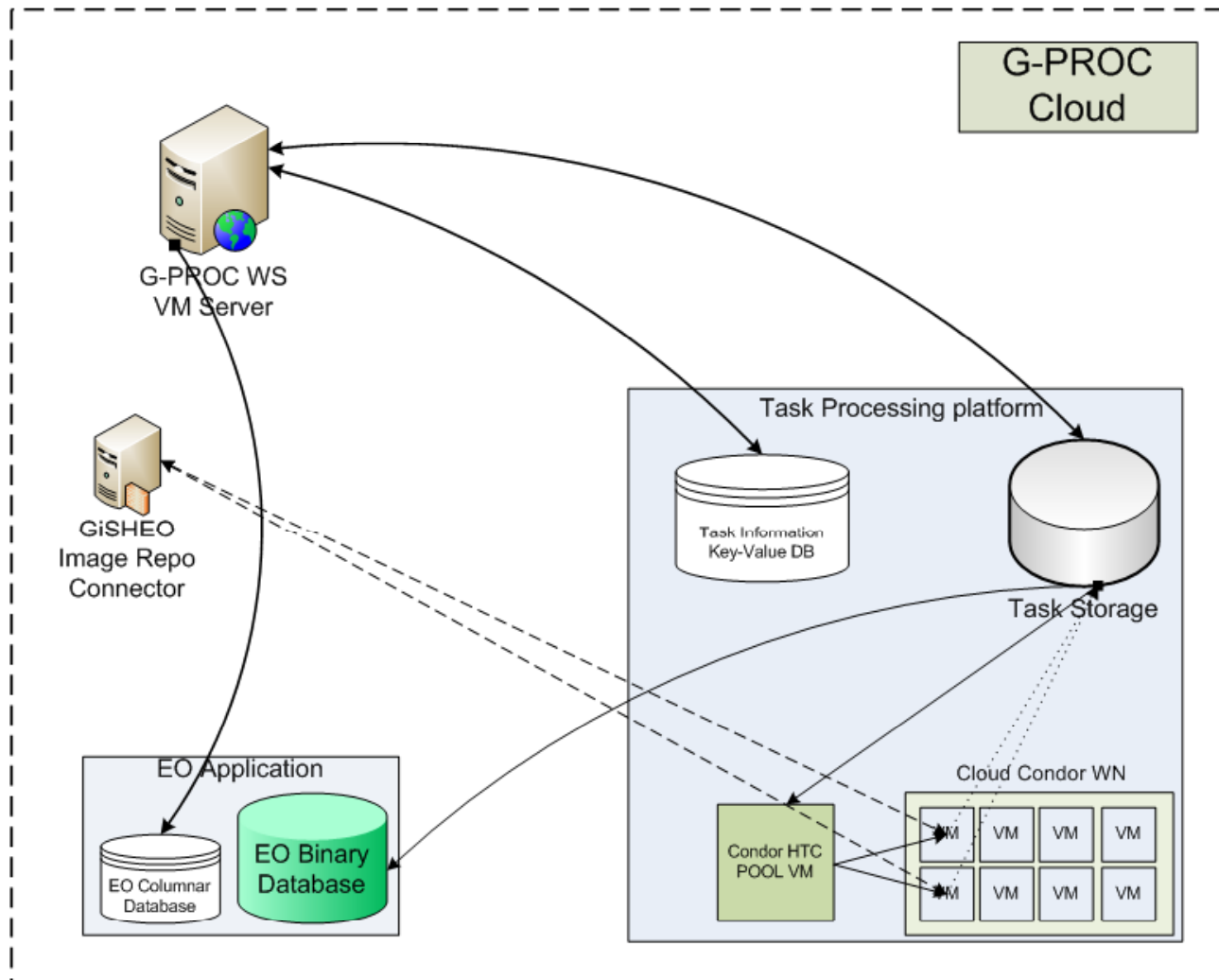
# G2Cloud Level 2

- at this level GiSHEO's G-PROC service and Web Application Service (WAS - EO Applications) are moved completely into the clouds.
- This can be done in several steps because we need to migrate and adapt (rewrite) some components to be suitable for clouds technologies:
  - G-PROC Service
    - G-PROC Task database from SQLite to Key-Value Cloud Database (Store);
    - G-PROC Processing storage (task directories and output) to a cloud storage;
    - GiSHEO Image repository connector for downloading datasets from GiSHEO Database;
  - Web Application Service (EO Application)
    - cloud storage for storing the binaries;
    - key-value store for storing applications parameters
  - Processing platform as a VMs infrastructure (Condor HTC instances are in the cloud);

# G2Cloud L2 Architecture



# G2Cloud G-PROC Service



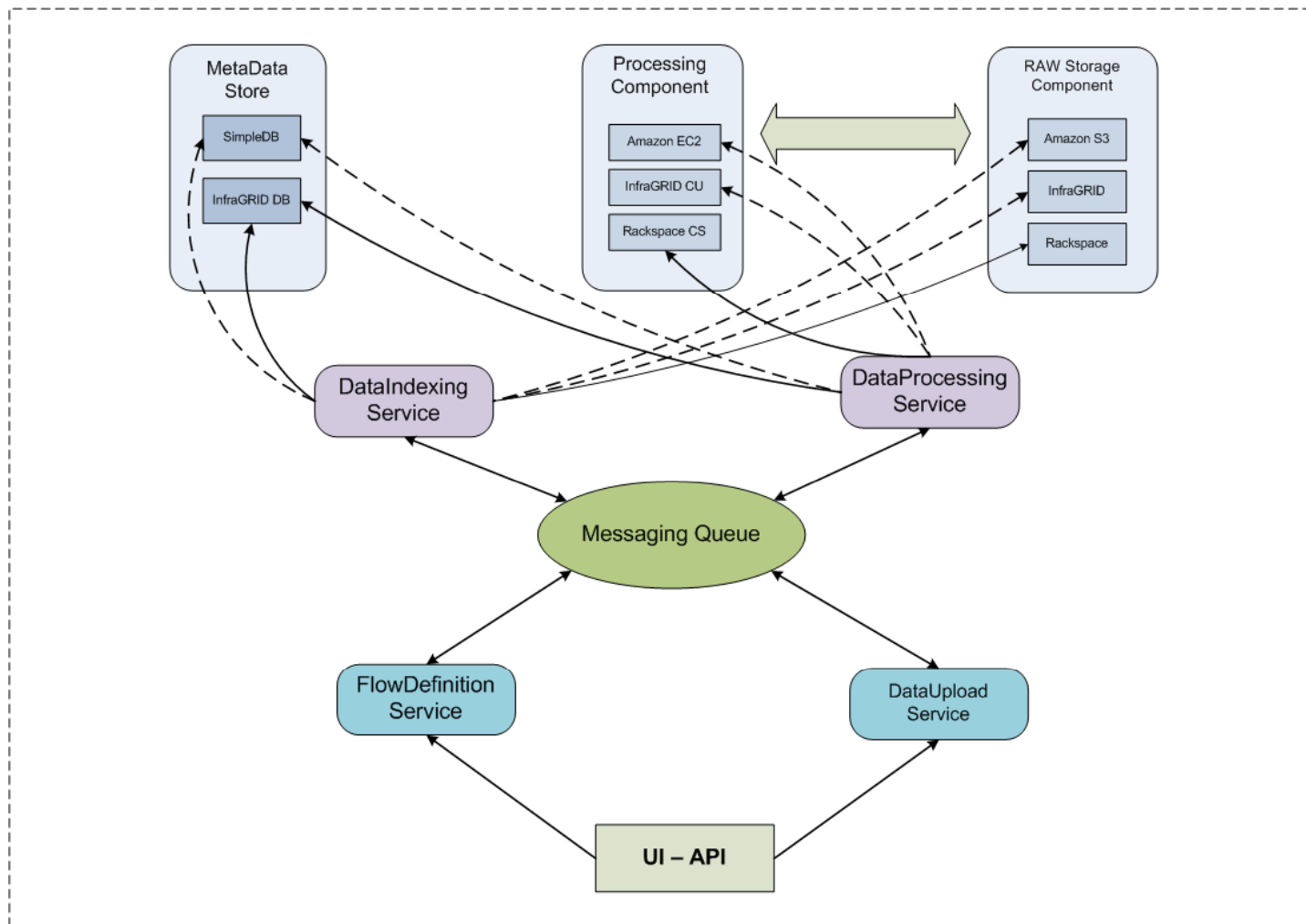
# G2Cloud Level 3

- at this level we propose to transform the most important parts of the GiSHEO Platform to fit the cloud model proposed by mOSAIC legacy app. support;
- New GiSHEO components:
  - MetaDataStore - a cloud database (columnar or key-value; depending on the type of data that are stored and the operations we want to support) for storing datasets metadata and EO application information;
  - Processing component - a cloud VM infrastructure where EO application can be processed;
  - Raw Storage Component - a cloud storage service for storing EO related data (datasets, tasks desc., app. binaries);

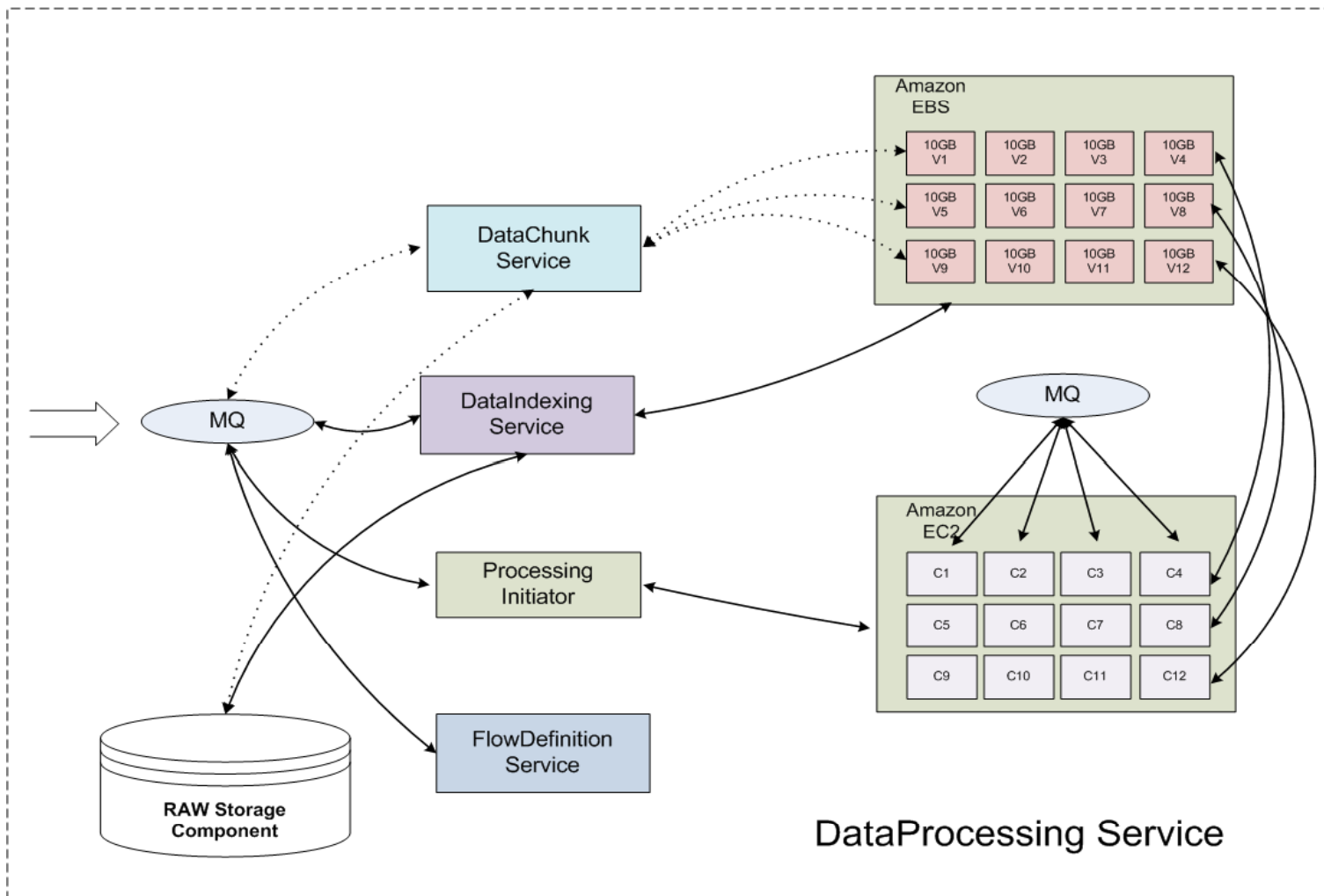
# G2Cloud Level 3

- New GiSHEO services:
  - DataProcessing Service - a service responsible with data processing (old G-PROC);
  - DataIndexing Service - a service similar to GDIS;
- New GiSHEO Front-ends:
  - FlowDefinition Service - a service where the UI will deploy the workflow description for data processing;
  - DataUpload Service - a service for uploading datasets into clouds;

# G2Cloud Level 3 Architecture



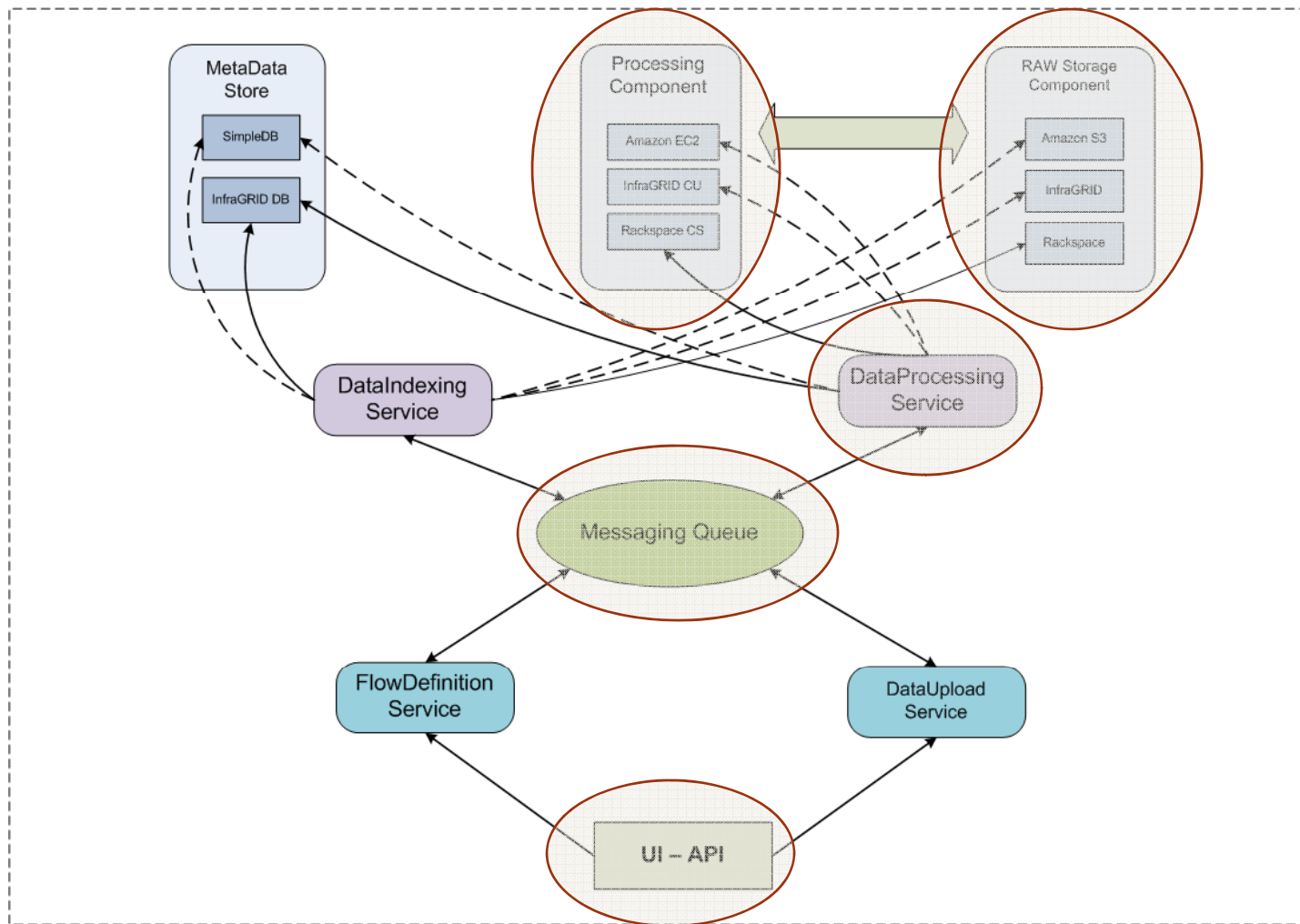
# G2Cloud Data Processing service



# G2Cloud DPS

- a new batch execution approach is proposed;
- centralized communication model is replaced by asynchronous decentralized message queues driven model;
- each processing resources (previous CondorHTC workers) are now message consumers and batch executors;
- all the services are build using the same model producer-consumer;
- service VMs will have block devices (EBS for Amazon and Eucalyptus cloud technologies) attached for data persistency:
  - message queue server is installed on an EBS (bootstrap) and it is attached to a pre-configured VM; if the VM dies then a new one is started and the EBS is attached to it and the MQ server is back online;
  - same approach for all the others core services of the platform;

# G2Clouds – Proof of concept



# Used technologies

- AXIS2 Web Service for exposing the Data Processing Service;
- Rabbit-MQ for messaging layer;
  - Java API;
- Eucalyptus as a VM scheduler;
- Eucalyptus Walrus (S3 compatible) for raw storage;
- Eucalyptus Block Storage (EBS compatible) for service data persistency;

# Simple Cloud Batch Execution platform

- batch execution platform for clouds to support legacy distributed applications;
- self-discovery protocol for network topology lookup (core services and neighbors);
- each node will be able to “talk” with it’s cloud neighbors for getting “useful” information (in case of algorithms that need communication for exchanging data);
- coupled with a scheduling platform capable of optimizing task scheduling; ( for example to reduce the overhead mostly introduced by the data transfer; )

# GISHEO UI DEMO

Questions ?!

**Thank you!**