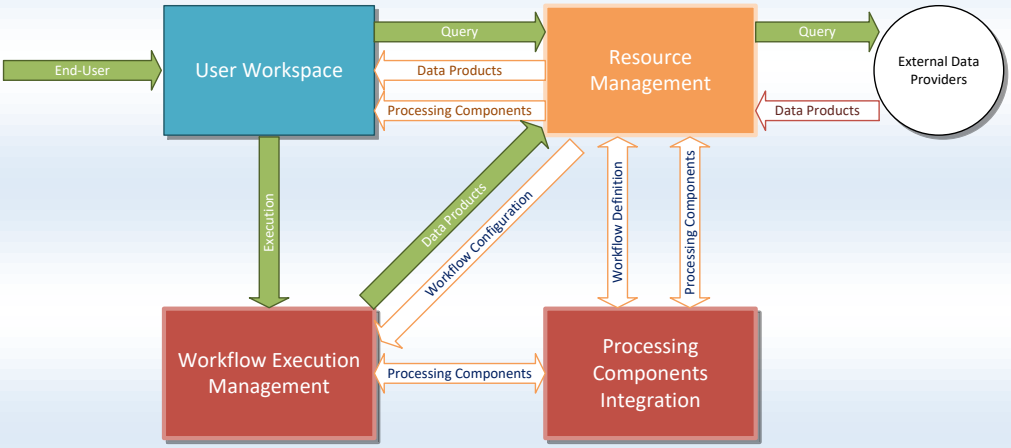
TAO (stands for Tool Augmentation by user enhancements and Orchestration) is a **lightweight, generic integration and distributed orchestration framework**. It allows to integrate commonly used toolboxes (such as, but not limited to, **SNAP, Orfeo Toolbox, GDAL, PolSARPro**, etc.). This framework allows for processing composition and distribution in such a way that end users could define by themselves processing workflows and easily integrate additional processing modules (by processing module it is understood either a standalone executable or a script).
In terms of use, the TAO platform provides a mean for orchestration of heterogeneous processing components and libraries in order to process scientific data. This is achieved in following steps:

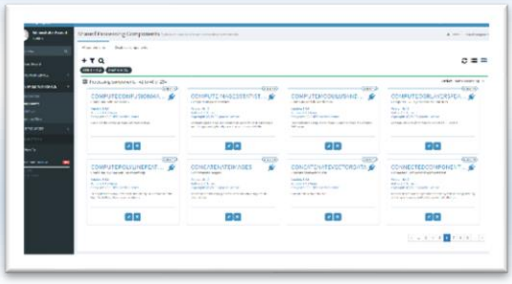
- Preparation of resources (including processing components) and data input
- Definition of a workflow as a processing chain
- Execution of workflows
- Retrieval / visualization of the results.

To have a simple view of the TAO platform, the platform model is split among four main macro-components. Such a macro-component is a logical collection of components with related functions. It has no direct relationship to the software implementation.



The key features of the TAO framework consist in:

- Visual **integration of EO processing toolboxes** (the user can perform such an integration by him-/her-self, without programming knowledge)
- A pre-configured set of **Docker containers** for **Orfeo Toolbox**, **SNAP**, **GDAL** and **Python 2.7**
- **Visual definition of processing workflows** by simple drag-and-drop operations and easy parametrization of the workflow elements

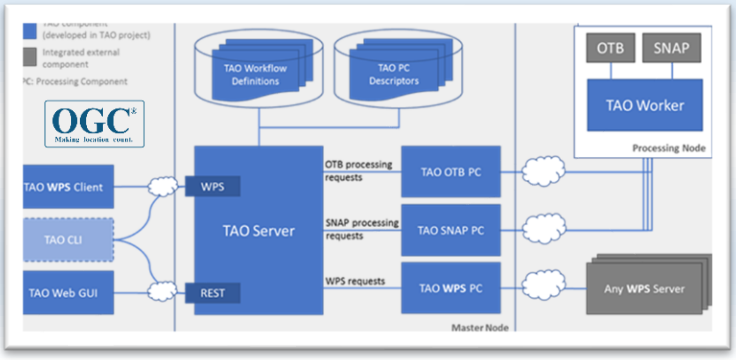


- **Integration of user-defined algorithms**, written in Python (also in R is possible), in the processing workflows
- Visual definition of **execution topologies** (collection of machines onto which components are executed)
- **Orchestrated execution of workflows**, employing different **DRMAA**-compliant Cluster Resource Manager software, such as Torque, SLURM or directly via SSH
- Self-contained **containerized execution** of components on remote nodes so that they do not interfere with other components
- Basic visual **monitoring of executions** and topology nodes resources

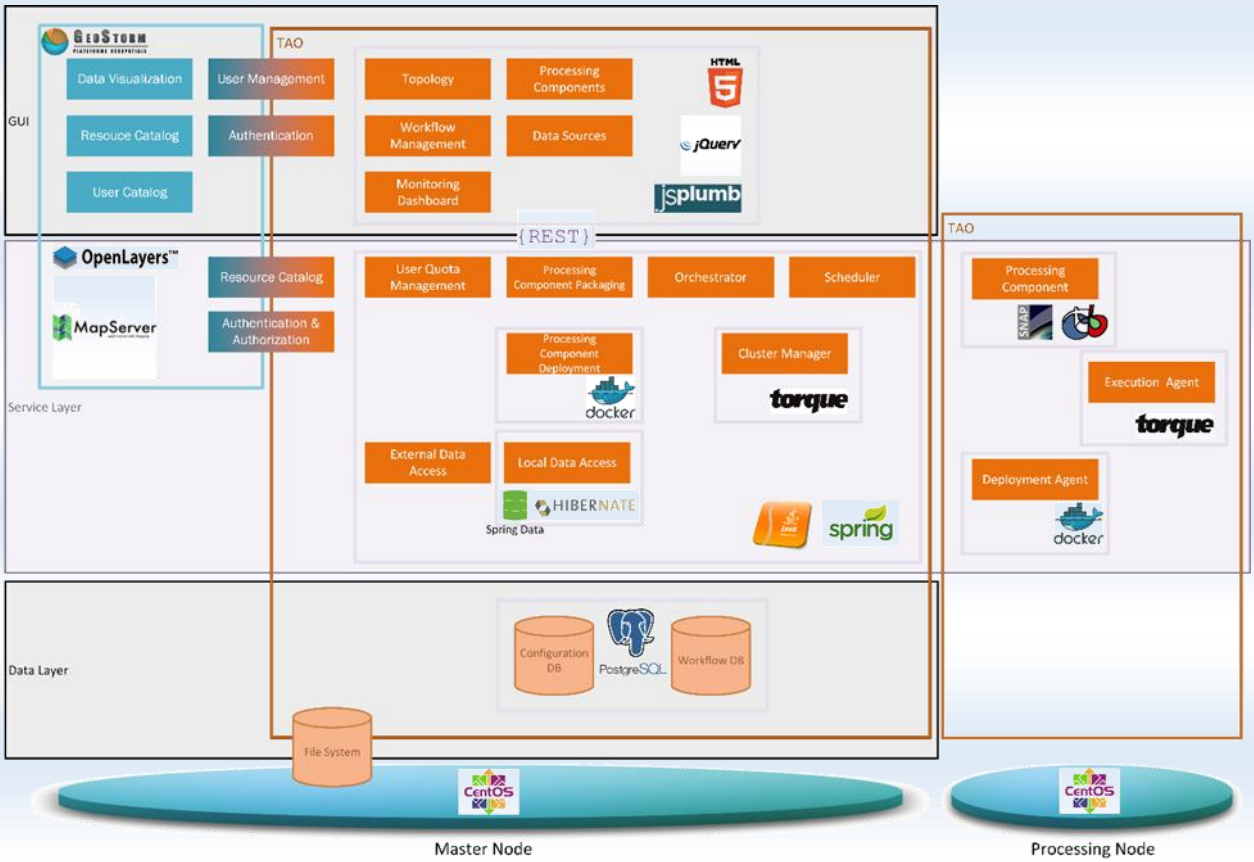
- **User and shared workspaces** to allow the visualization of the execution **results**, but also to allow users to upload various files that can be used in the workflow execution (such as **model files**, **shape files**, etc.)

- **EO data sources abstraction framework** (plugin-based) that allows the querying and retrieval of EO data from different providers. Out-of-the-box, TAO comes with plugins for ESA's **Scientific Data Hub, Amazon Web Services, USGS Earth Explorer, PEPS, ESA's FedEO**, **Alaska Satellite Facility**, for a various collection of sensors. Additionally, it also comes with such plugins for 3 of the recent **DIASes** (**CreoDIAS**, **Mundi** and **Onda**), allowing the re-use of local DIAS product archives

- **OGC standard interfaces** for exposing workflows as WPS services
- A rich **RESTful API** to control and manage all the TAO entities and services
- **Plugins** for integration with external visualization platforms, such as CS SI's GeoStorm.



- The framework is **open source** under **GPLv3** license and it is entirely written in the Java language. It was developed on the following technology stack:



The source code is available at: https://github.com/tao-org