

# WISDOM Data Manager

Last review date	Reviewer
2009-09-15	Marco Bencivenni Enrico Fattibene

---

## Table of Contents

[WISDOM Data Manager](#)

[WISDOM Production Environment](#)

[WISDOM Data Manager](#)

## WISDOM Data Manager

Use of the Data Management service in the WISDOM Production Environment.

### WISDOM Production Environment

WISDOM can be seen through 2 aspects, the drug- discovery application against emergent and neglected diseases, or the production environment that was developed (and that is still in continuous development) to manage jobs and data on the Grid.

Even though WISDOM has been initially an application on large scale drug discovery, it tends now to cover more ground through the WISDOM Production Environment, a generic meta- middleware allowing the automatic management of jobs and data on a multi- Grid- infrastructure, for simple jobs or large- scale deployment alike.

The WISDOM Production Environment has been developed (and that is still in continuous development) to allow automatic and efficient submission of jobs and data management. The goal of the WISDOM environment is to manage all the Grid specific operation for the users. The environment provides a set of services that virtualizes the Grid complexity. The users finally interacts with the system through high- level business specific web services, through which they don't submit jobs, but instead send simulations.

### WISDOM Data Manager

The WISDOM Data Manager, is the "service" that provides all the mechanisms to automatically deploy data on the Grid, replicate them a given number of time, provide meta- data on those data, resolve filenames into physical locations, etc. The particularity of WISDOM Data Manager is that it does not rely on a specific infrastructure. It can be used and integrated with basically any middleware, or batch queuing system. It can be used to deploy automatically files on multiple infrastructures, and replicate them on those infrastructures and resolve the physical locations for each of these infrastructures.

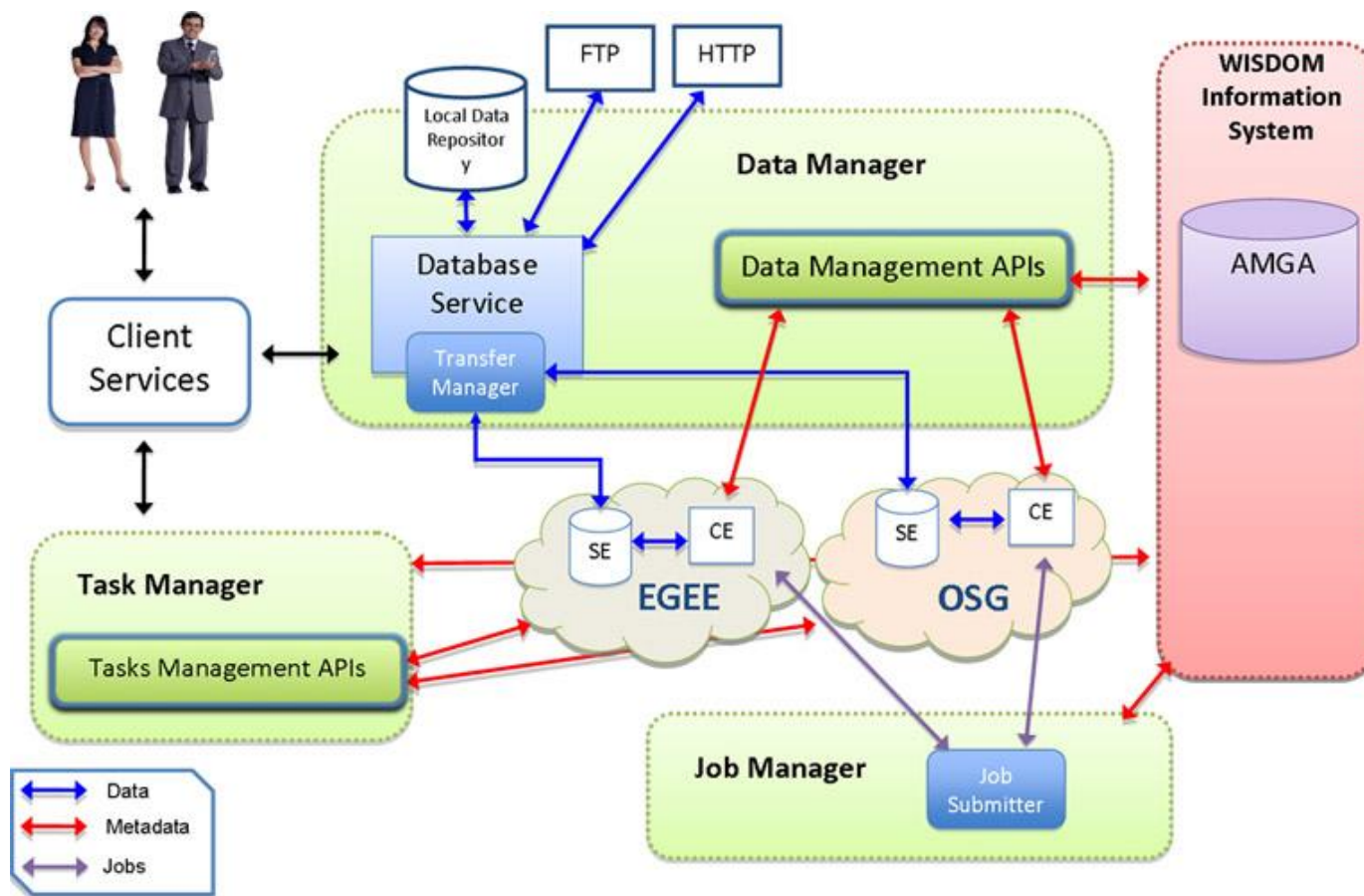
#### WISDOM Data Manager in EGEE

On the EGEE/ gLite middleware, WISDOM Data Manager can use LFC to store files, or directly copy the files on the storage elements taking advantage of the SRM interface. For performances issues, we tend to use the SRM interfaces. For the moment the system simply uses the existing SRM command- line APIs on the gLite UI, but in the future it is not excluded that more generic APIs will be used. A random url is generated and the files are put in the VO specific "generated" path since all the ACL are correctly setup already on the SE. Files are then registered into the WISDOM information system based on AMGA (some metadata can be added to this to make extended query on the files).

The use of SRM offers more flexibility since the replicas can be handled with the Data Manager, and provides more fault tolerance and more features:

- Making the jobs automatically share the multiple replica instead of using the same one.
- Providing to the job a location that is uncorrupted and accessible.
- Automatically and dynamically increasing the number of replica with the workload.

- Providing an access to SE that is completely transparent. The system can test the SE on the fly, and blacklist the failing SE. Since all the SE are registered in the WISDOM information system, when the user wants to perform a replica on a specific SE, he can change its status depending on the success of the replication, to avoid trying to store a file on a failing SE, or avoid sending back replica located there.



Thanks to the Data Manager the Grid users have only to specify a file repository they want to deploy on the Grid (either in their local account, or on an ftp server) and indicate how many times they want the files to be replicated and on which infrastructures. The system then automatically deploys everything, and replicates the files.

The system can be used to deploy and replicate single files or huge databases alike, even though it was initially made to automatically deploy, replicate and update biological databases that are repositories of flat files (not relational databases). It was tested successfully on databases that contained around 10000 files.