

GAMAS

A GENERIC AND MULTIPURPOSE ARCHIVE SYSTEM

Etienne Lyard⁽¹⁾, Roland Walter⁽¹⁾, Vitalii Sliusar⁽¹⁾, Dominik Neise⁽²⁾, Adrian Biland⁽²⁾ and Pablo Fernandez⁽³⁾

(1): Université de Genève, Département d'Astronomie; (2): ETH Zurich, Institute for Particle Physics and Astrophysics; (3): Swiss National Supercomputing Centre

OVERVIEW

There exist many distributed storage systems that are mature and reliable, for instance DIRAC [1], iRods [2] and OneData [3]. These systems lack compliance with the Open Archival Information Systems (OAIS) standard and require data centres that run them to accommodate for their specific needs. We introduce GAMAS, a novel distributed OAIS. GAMAS is a lightweight python package that can be interfaced with any storage via a plugin system. It is modular and can be used stand-alone or in conjunction with pre-existing distributed storage systems. It ensures data integrity and provides high-level data management based on experiments metadata.

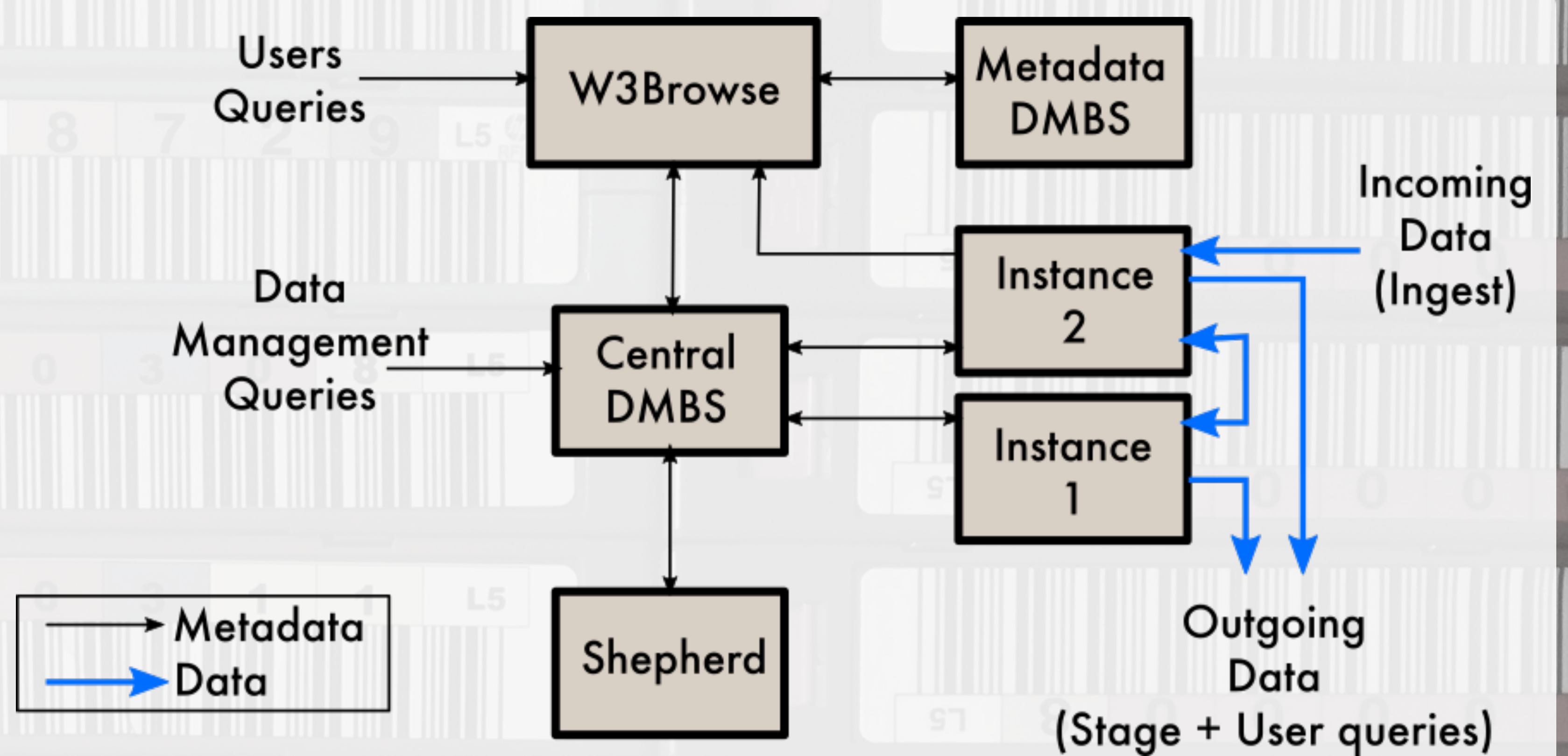


Figure 1: Overall architecture of GAMAS with 2 instances.

ARCHITECTURE

GAMAS is composed of a set of instances orchestrated by a Shepherd, via a central database, as seen on figure 1. It can be interfaced to W3Browse [4] for experiments metadata queries and retrieval. Each instance is interfaced to a specific set of storages as seen on figure 2. Each instance provides the same set of functionalities, namely Ingest, Expose, Copy and Stage. At least one instance must run an ingest process, which verifies, archives and registers incoming datasets. Ingested datasets are then exposed and copied until the requested number of copies exists in the system. Archived datasets can be staged to a scratch storage for processing.

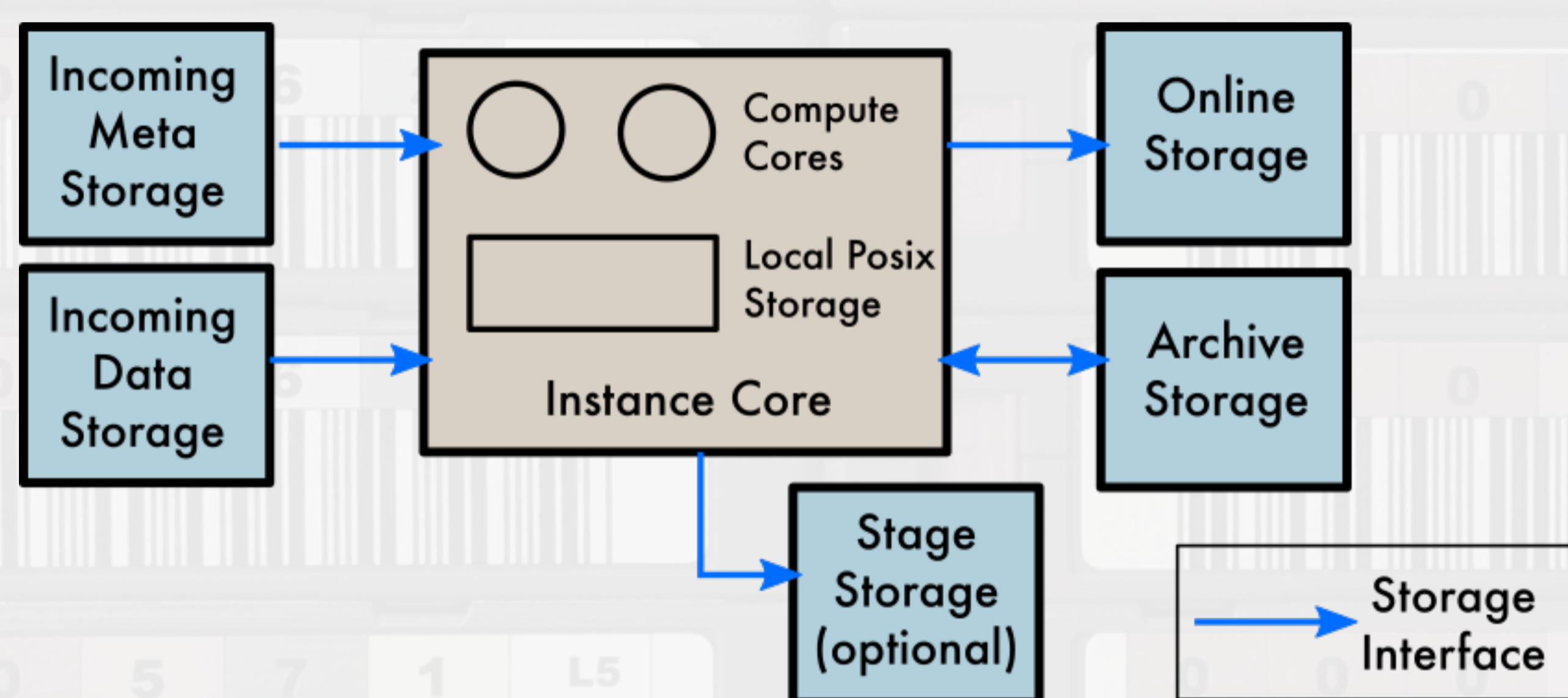


Figure 2: Internal architecture of a GAMAS instance.

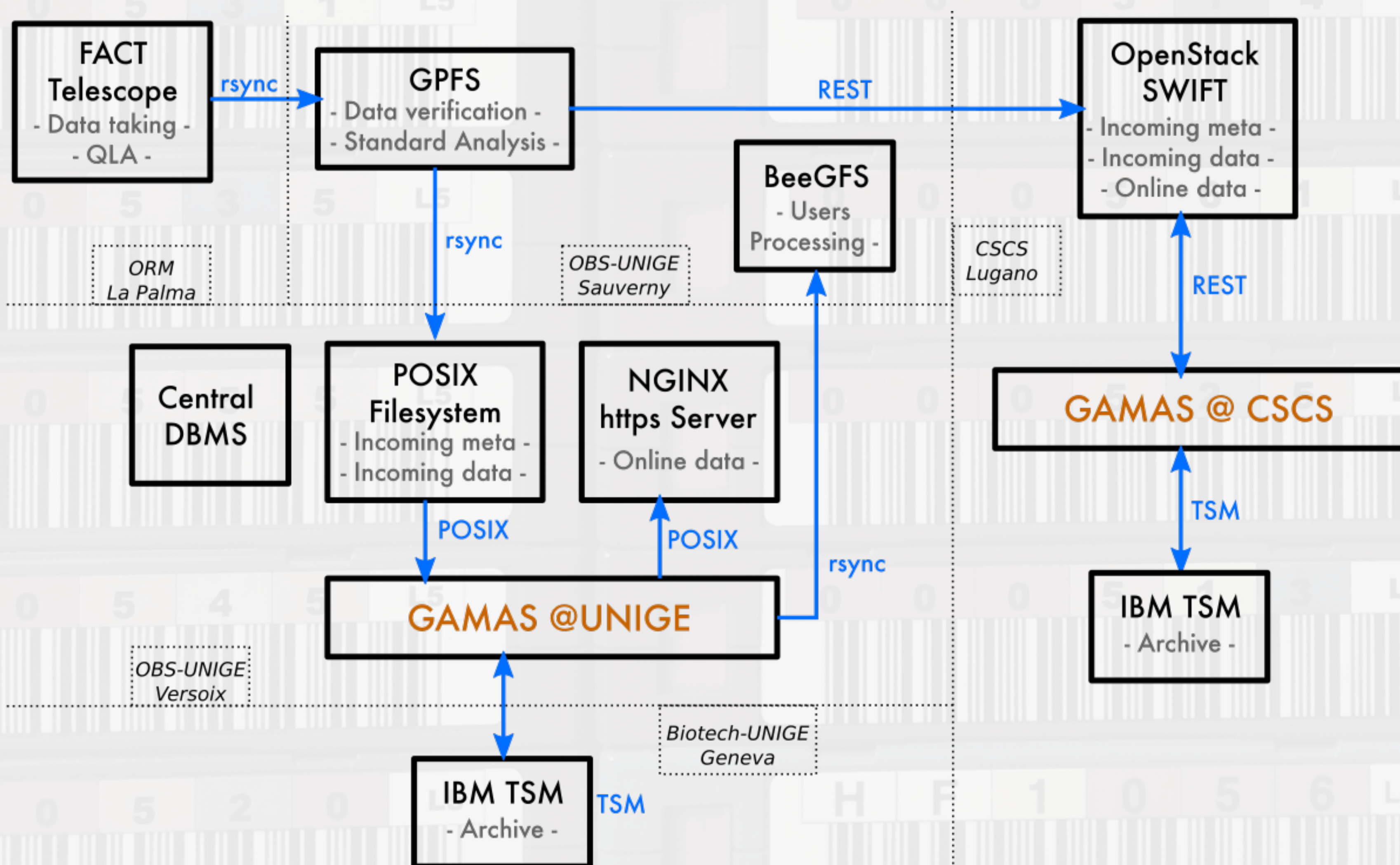


Figure 3: Overview of the current GAMAS prototype deployment.

CURRENT PROTOTYPE

The current prototype handles approximately 400TB of data from the FACT experiment [5] and 200TB more is being added. A schematic view of the current prototype can be seen on figure 3. The exact same code runs on both instances, though with different configurations. The instance at UNIGE runs on bare metal, while the one at CSCS runs in a virtual machine. The ingest process can run at up to 200MB/s as long as enough throughput is available from the various storage interfaces. Datasets can be staged back to a BeeGFS [6] filesystem within the internal network of the University of Geneva.

REFERENCES

- [1] DIRAC | The INTERWARE. <http://diracgrid.org/> accessed June 2019
- [2] IRODS. <https://irods.org/> accessed June 2019
- [3] ONEDATA - Global data access solution for science. <https://irods.org/> accessed June 2019
- [4] W3Browse - NASA's archive system. <https://heasarc.gsfc.nasa.gov/w3browse/> accessed June 2019
- [5] H. Anderhub et al. Design and operation of FACT. Journal of Instrumentation, vol. 8, 2013.
- [6] BeeGFS - The Parallel Cluster File System. <https://www.beegfs.io/content/> accessed June 2019



UNIVERSITÉ
DE GENÈVE

FACULTÉ DES SCIENCES

