Summary item 3

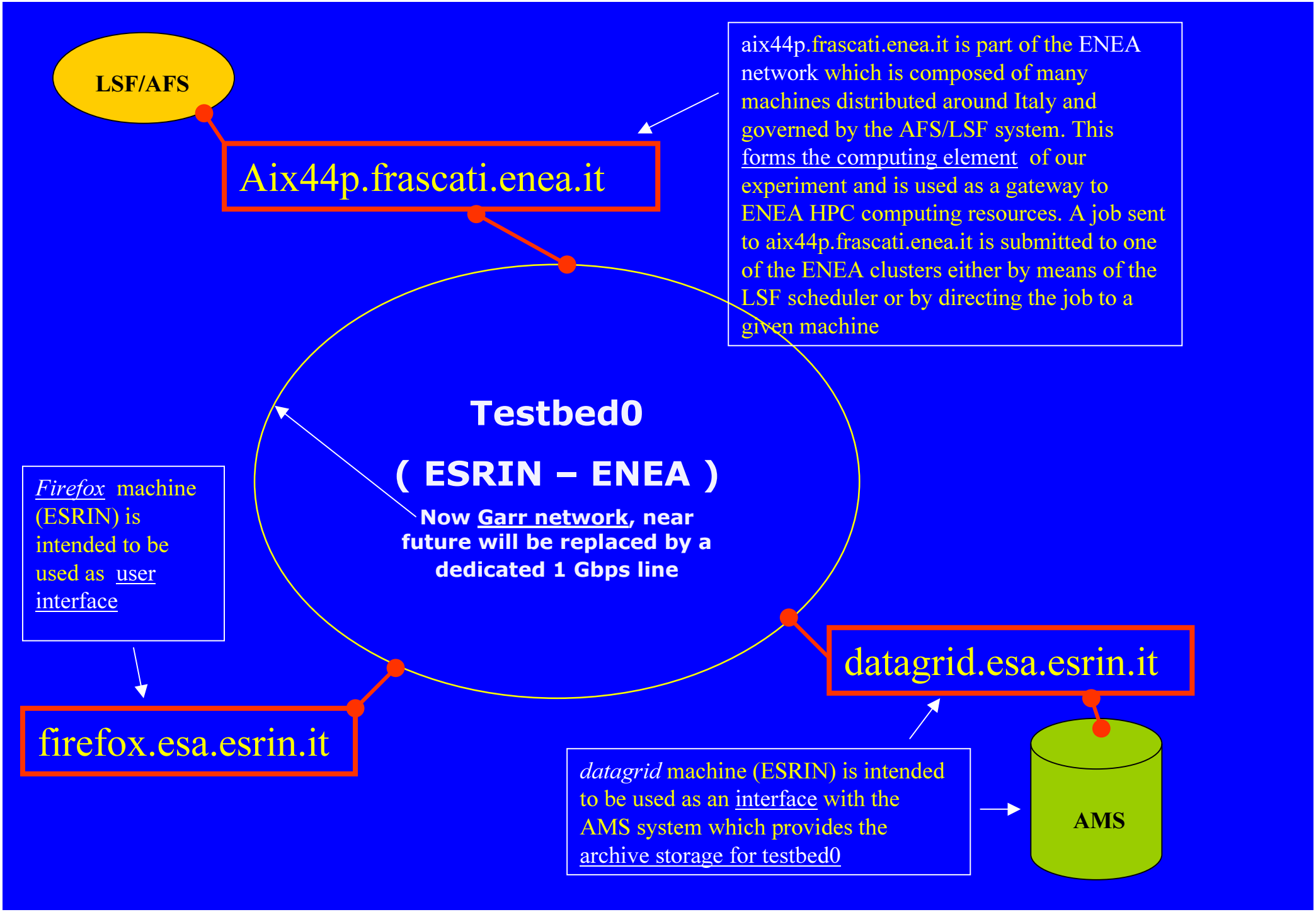
- ESA and WP9 part 1 (45m)
  - DataGrid Frascati infrastructure (AT, 7.5m)
  - ESA GOME application (SC, 7.5m)
  - ESA experience in using DataGrid testbed 0 and testbed 1 (JL, 15m)

Summary

- Experiences with Testbed0
- Experiences with Testbed1
- GOME Application Use Case
- EO Application Components
- Issues

Testbed0

- Installation and testing of Globus
- Globus 1.1.3 and Condor installed on several ESRIN machines and tested (Task 00 Activity Report by Fabrizio Pacini)
- First GOME processing prototype
- Globus 1.1.3 installed at ENEA
- Solved security issues using GSIKLOG workaround
- Demonstration and report at EDGC3 Frascati (October 2001)


LSF/AFS
Aix44p.frascati.enea.it
aix44p.frascati.enea.it is part of the ENEA network which is composed of many machines distributed around Italy and governed by the AFS/LSF system. This forms the computing element of our experiment and is used as a gateway to ENEA HPC computing resources. A job sent to aix44p.frascati.enea.it is submitted to one of the ENEA clusters either by means of the LSF scheduler or by directing the job to a given machine
Testbed0
(ESRIN – ENEA)
Now Garr network, near future will be replaced by a dedicated 1 Gbps line
Firefox machine (ESRIN) is intended to be used as user interface
firefox.esa.esrin.it
datagrid.esa.esrin.it
datagrid machine (ESRIN) is intended to be used as an interface with the AMS system which provides the archive storage for testbed0
AMS

The following solution has been adopted to allow Globus to acquire the AFS token. First, the Globus Security Inftrastructure (GSI) software has been installed both the client (Aix44p) and server (aixfs, AFSfileServer) at ENEA. Second, two background tasks based on the GSI software have been configured: the GSIKLog client, invoked by the Globus software each time a a job is submitted, and GSIKLogD server which authenticates the Globus user by checking the Grid certificate associated with the token.

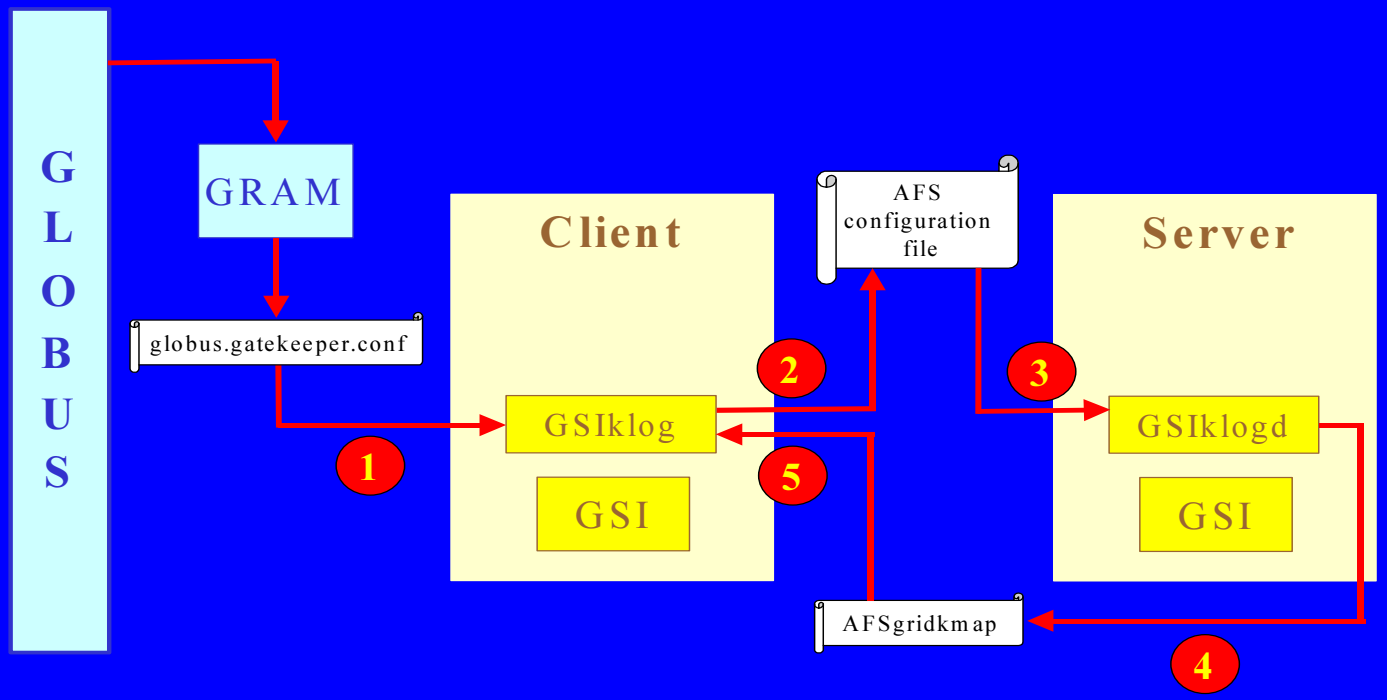


Figure : Globus acquires the token to access AFS. *Step 1:* having submitted a job request to the Globus Resource Allocation Manager (GRAM), the configuration file *globus.gatekeeper.conf* is sent to GSIklog client, step 2: a fileserver is found in the AFS configuration file, *step 3:* the request for a token is sent to the server, *step 4:* the AFSgridkmap file is created, *step 5:* an AFS token is returned to the client.

• The user saves the files *infostart.txt* on firefox. It contains high level information related to the starting and stopping date elapsing the period which he or she needs to study. This file is sent from firefox to datagrid by means of the *globus-rcp* command:

```
# globus-rcp /home_firefox/user_home/infostart.txt
datagrid.esrin.esa.it/home_datagrid/esagrid/Grid_Data_Proj/info/in
fostart.txt
```

• Then the shell script on datagrid is launched from firefox using the globus *globusrun* command:

```
# globusrun  - s  - r datagrid.esrin.esa.it/jobmanager - f
/home_firefox/user_home/get.rsl
```

• This command copy the data from AMS to datagrid.

• After having retrieved the files from the AMS they are transferred to the machine aix44p using:

```
# globus-rcp
/home_datagrid/esagrid/Grid_Data_Proj/archive/00220202.lv1

aix44p.frascati.enea.it:/afs/enea.it/frascati/info/grid/0022
0202.lv1
```

- Now the user is ready run its application on aix44p by IDL

```
# globusrun  - s  - r  datagrid.esrin.esa.it/jobmanager  - f
/home_firefox/user_home/gome.rsl
```

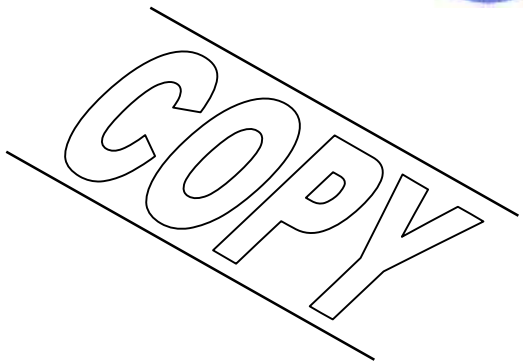
```
/afs/enea.it/software/rsi/idl/bin/idl</afs/enea.it/frascati/info/grid/gome.es
```

- Finally, the resulting fileis transferred back to firefox by means of *globus-rcp*.

Testbed1

- Integration team assembled at CERN (September)
- First presentation of Integration Testbed (October)
- WP9 Validation Plans drawn up (October)
- Loose Cannons formed
- Validation plans presented at CERN (November)
- Demonstration of integrated testbed (December)
  - Officially opened to the applications
- IDL packaged, delivered and tested (January)
- Report to PTB (January)

Report to PTB (January)

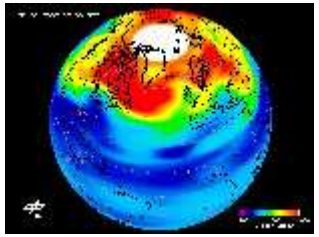


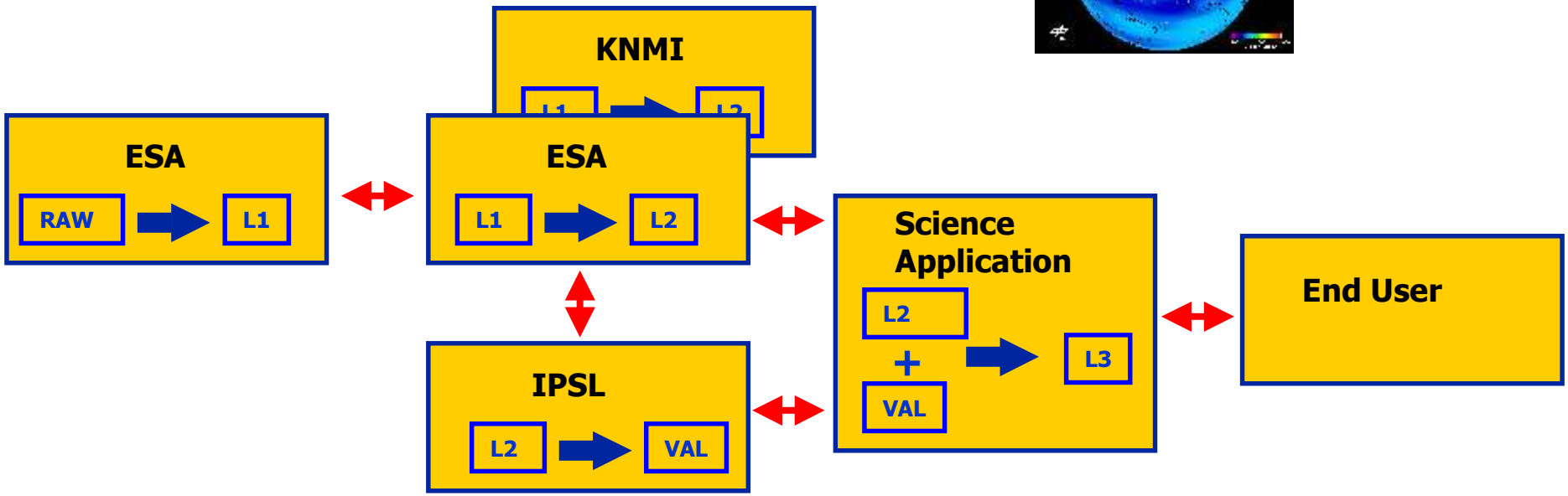
- Tested: Basic Job Submisson OK
  - High throughput simultaneous jobs NOT tested
  - dg-job-get-output problem
- Not Tested: RC / RM / GDMP
  - No clear procedure; end-to-end demo by ITeam not seen
  - Access to SEs: we assume its not ready for us to test yet
- CEs/SEs for EO VO installed at few sites: CERN, Lyon, NIKHEF
- Testbed1 sites lacking in conformity
- Preparing to test IDL: Application Kits currently being installed
  - Grid-wide installation procedure?
- Documentation quality: verify that examples actually work
  - Prevented IPSL from testing GDMP
- Integration effort underestimated
  - Missing integration layer; installation procedures unclear; not automated; needs help of an expert

GOME Use Case

L1 4724 files = 66 Gb

L2 9,448,000 files = 108 Gb

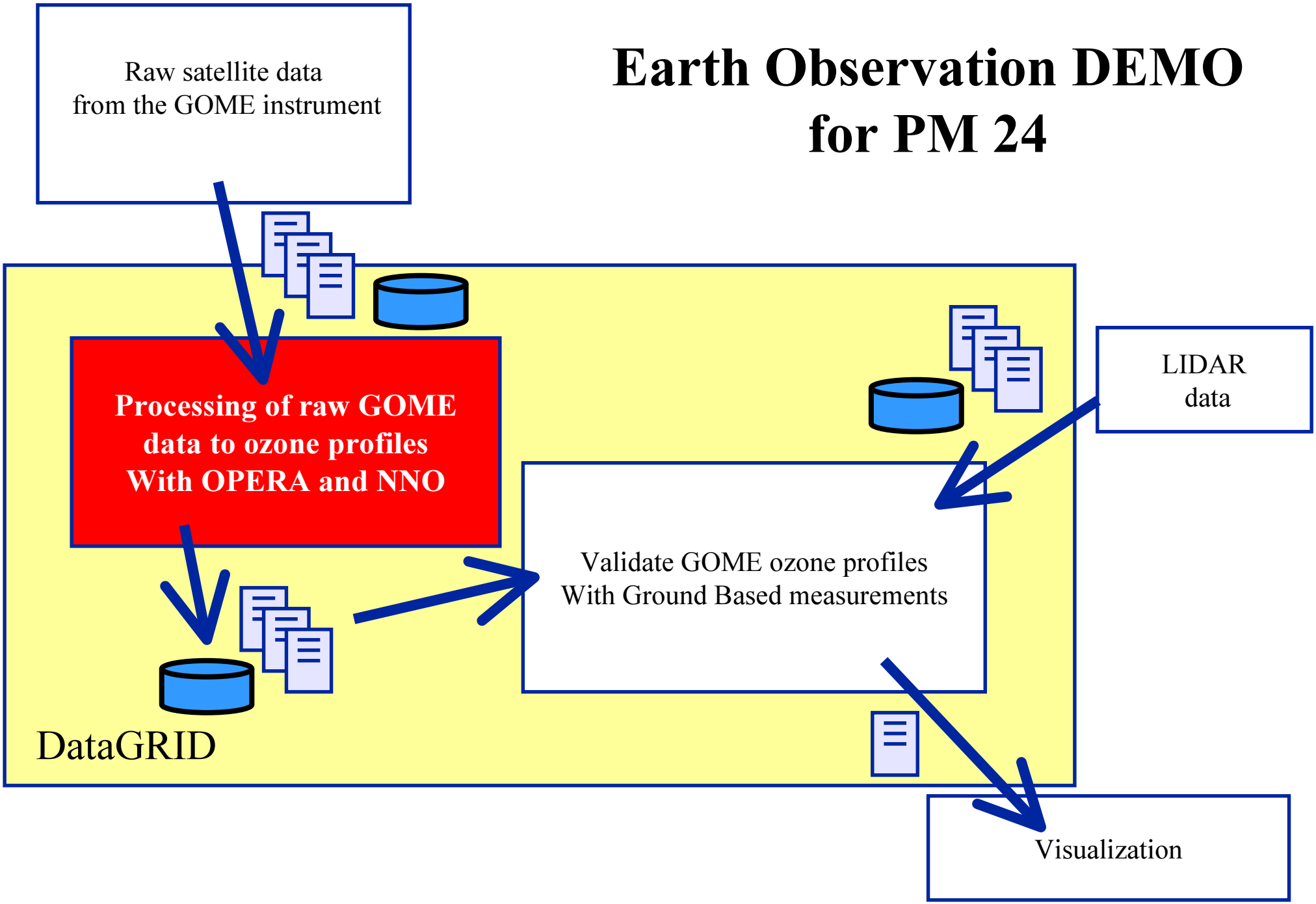




Regulated Access to Grid processing power

Secure access to Grid-registered high-volume data storage

Earth Observation DEMO for PM 24



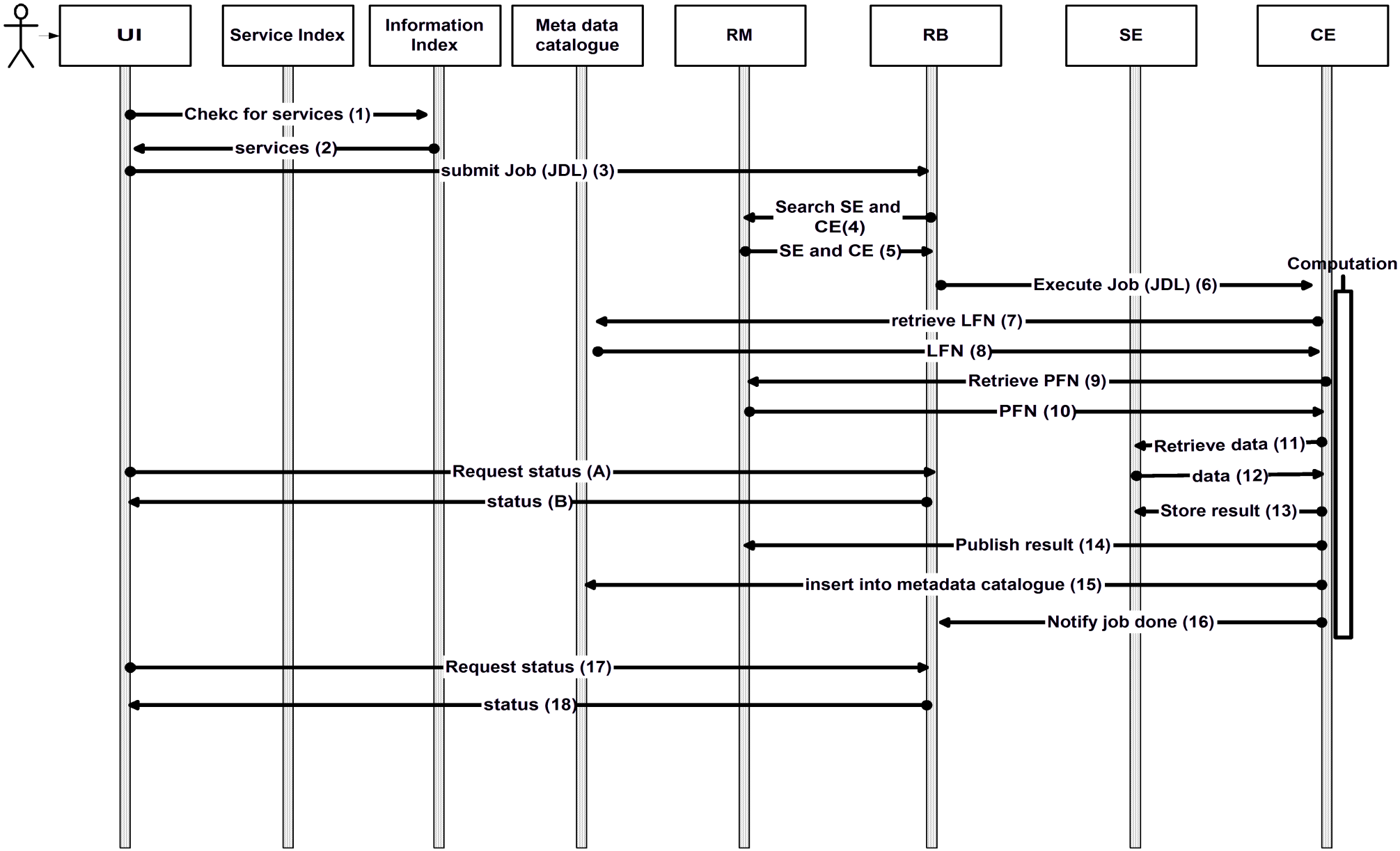
File numbers for one year of GOME data :

	Number of files (to be stored and replicated):	File Size
Level 1:	4,724	15 Mb
NNO:	9,448,000	10 kb
Opera :	9,448,000	12 kb
Lidar	12	2.5 Mb
Total:	18,900,736	267 Gbyte

Gome has a data set of 5 years

Gome is relatively small (in both size and number of files)

Use case flow for profile processing:



EO Application Interface Components

- DataGrid provides a series of low-level grid commands
- In order to support a wide range of EO applications we need generic components embedded in a Grid Application Layer
- Provides functions such as:
  - Information Services: Obtaining information about EO products and services
  - Job Execution Cycle: Composing JDL scripts, submitting them, monitoring, retrieving job results
  - MSS interface: Moving files between MSS archives and Grid SEs
  - Data Replication: Replicating SE datasets to other Grid locations
  - Metadata Services: Registering file metadata with Grid catalogs
  - File retrieval directly from metadata keys: Search Grid catalogs to retrieve file names for the dataset to be processed
- Provide support for higher level application interfaces e.g. repository, workflow

Issues (1)

- Very-large-scale, complex system with large numbers of participants
  - Dealing with new concepts and technology
  - Communication and coordination in de-centraised, multi-cultural, multi-institutional development group
  - Ongoing agressive deployment of middleware releases
  - Huge turnover in output: INFORMATION OVERLOAD risk
- Driven by needs of HEP
  - With EO & Biology contributions
  - Discussion forum is rather crowded
  - Lack of EO scientists in other WPs (1-8, 10-12)
  - Terms of reference in meetings & discussions :
    - Physicist : Scientist; Experiment : Application; Loose Cannons : Validation Team; CASTOR : Archive Storage; etc.

Issues (2)

- Yet the basic requirements are mostly in common
  - Need first to develop the basic functional layer
  - Can then be extended for application-specific requirements
- Testbed stability, usability, performance and scalability
  - Application Grid interfacing layer needs to be developed
  - APIs needed
  - Limited functionality
  - Limited performance
- Ongoing rapid prototyping and development
  - Keeping step with code & documentation in a rapidly changing distributed environment
- Integration with existing environment

First collaborative scientific result using GRID!