

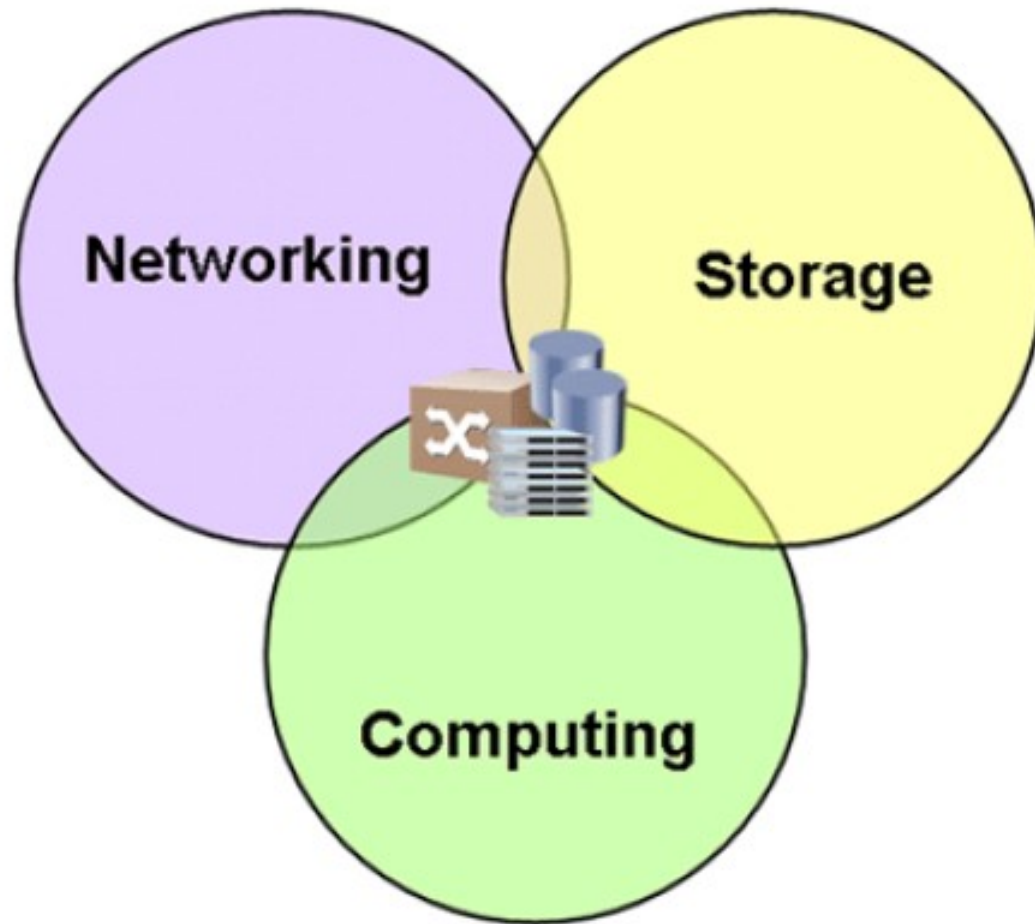
E-science Applications on the Grid

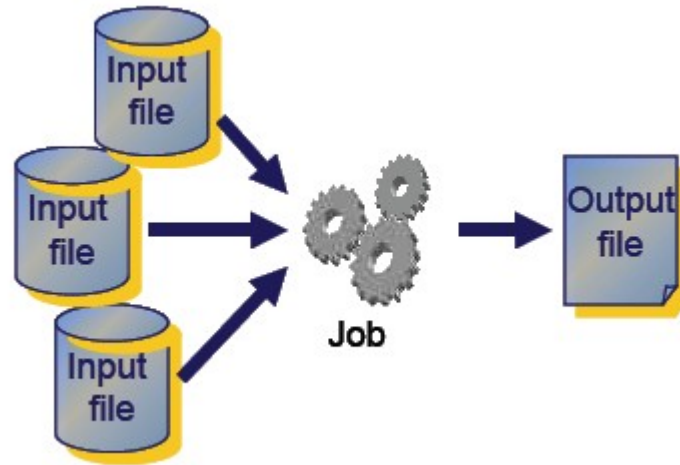
*Fotis Georgatos <gef@grnet.gr>
GRNET*

*Grid Applications Development
February 21st, 2008
Demokritos, Athens, Greece*

- **Portion of slides (derived from those) prepared by:**
 - Mike Mineter, NESC
 - Charles Loomis, LAL-Orsay
 - Roberto Barbera and his GILDA team
University of Catania and INFN
 - GRNET colleagues
 - EGEE-II NA2/NA3/NA4 Activity Members

- **Basics; the necessary concepts**
- **The HellasGrid & SEE environment**
- **Workflows and parametric jobs**
- **Portals and job submission tools**
- **Credential management**
- **AMGA & Metadata management**
 - A demonstration of a UNOSAT application
- **Trade-offs and special topics**
 - Performance Optimization
 - Security vs Performance
 - Information System consistency
 - Checkpointing
- **Other resources that could be useful**

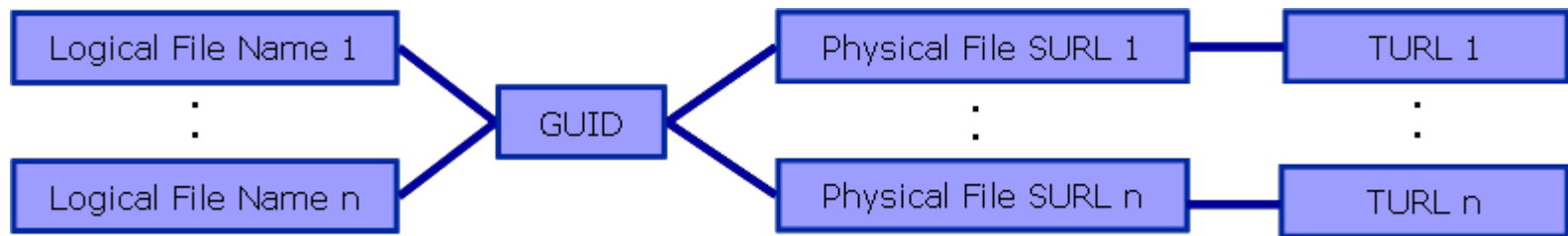




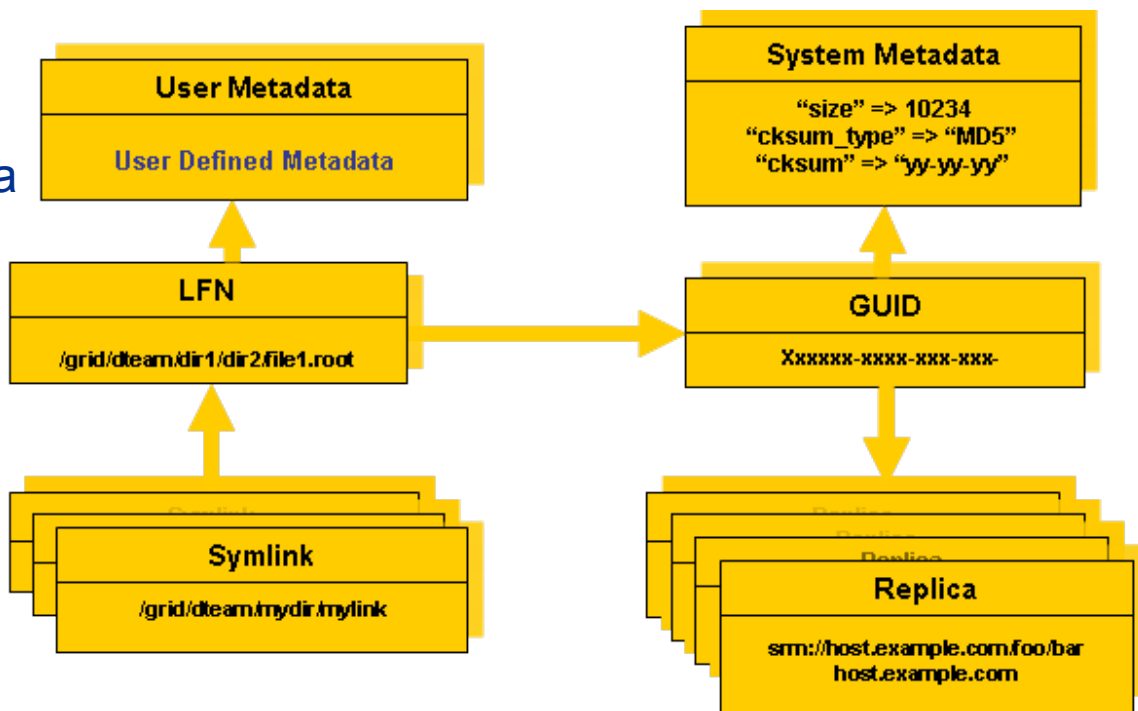
The user asks for the execution of an application at a remote system. It is typically described with a “.jdl”. The application handles input data (that perhaps exist on the Grid already) and produces some output data, which are also stored on the Grid, quite likely on the Storage Element or the UI itself.

- **UI: User Interface**
 - This is the system through which a User submits her jobs
 - It might be any Linux system, including a portable notebook.
- **WMS: Workload Management System (or RB)**
 - This is the node where User's jobs arrive, before being sent to a CE
- **CE: Computing Element**
 - This is the front-end node that provides access to a set of WNs
- **WN: Worker Node**
 - These nodes provide the computational services to the Grid
- **SE: Storage Element**
 - These nodes provide access to disk and tape subsystems
- **LFC: LHC File Catalog**
 - Technically it is the file index, the “parent” of all Storage Elements.
- **BDII: Berkeley Database Information Index (IS)**
 - It provides the information «Where/who/what is the Grid»

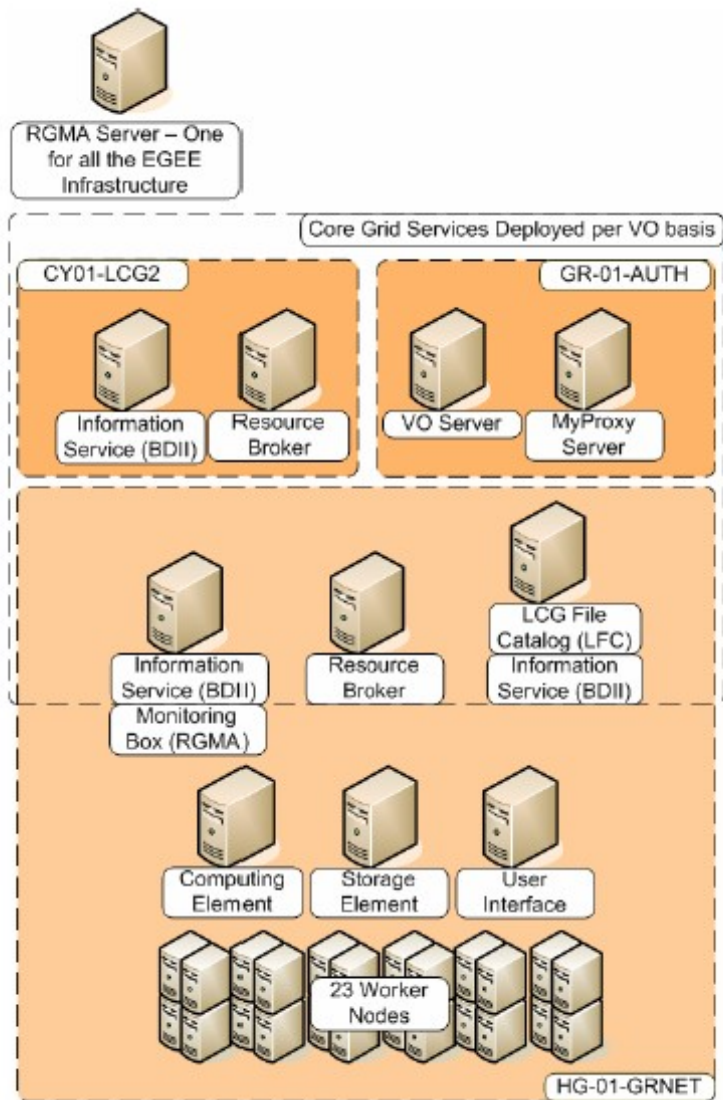
- Logical File Name (**LFN**)
 - An alias created by a user to refer to some item of data, e.g. “lfn:cms/20030203/run2/track1”
- Globally Unique Identifier (**GUID**)
 - A non-human-readable unique identifier for an item of data, e.g. “guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”
- Site URL (**SURL**) (or Physical File Name (**PFN**) or Site FN)
 - The location of an actual piece of data on a storage system, e.g.
 - “srm://pcrd24.cern.ch/flatfiles/cms/output10_1” (SRM)
 - “sfn://lxshare0209.cern.ch/data/alice/ntuples.dat” (Classic SE)
- Transport URL (**TURL**)
 - Temporary locator of a replica + access protocol: understood by a SE, e.g. “gsiftp://lxshare0209.cern.ch//data/alice/ntuples.dat”



- Manages the identification, sharing and replication of data in the gLite Grid.
- LFN acts as main key in the database. It has:
 - Symbolic links to it (additional LFNs)
 - Unique Identifier (GUID)
 - System metadata
 - Information on replicas
 - One field of user metadata





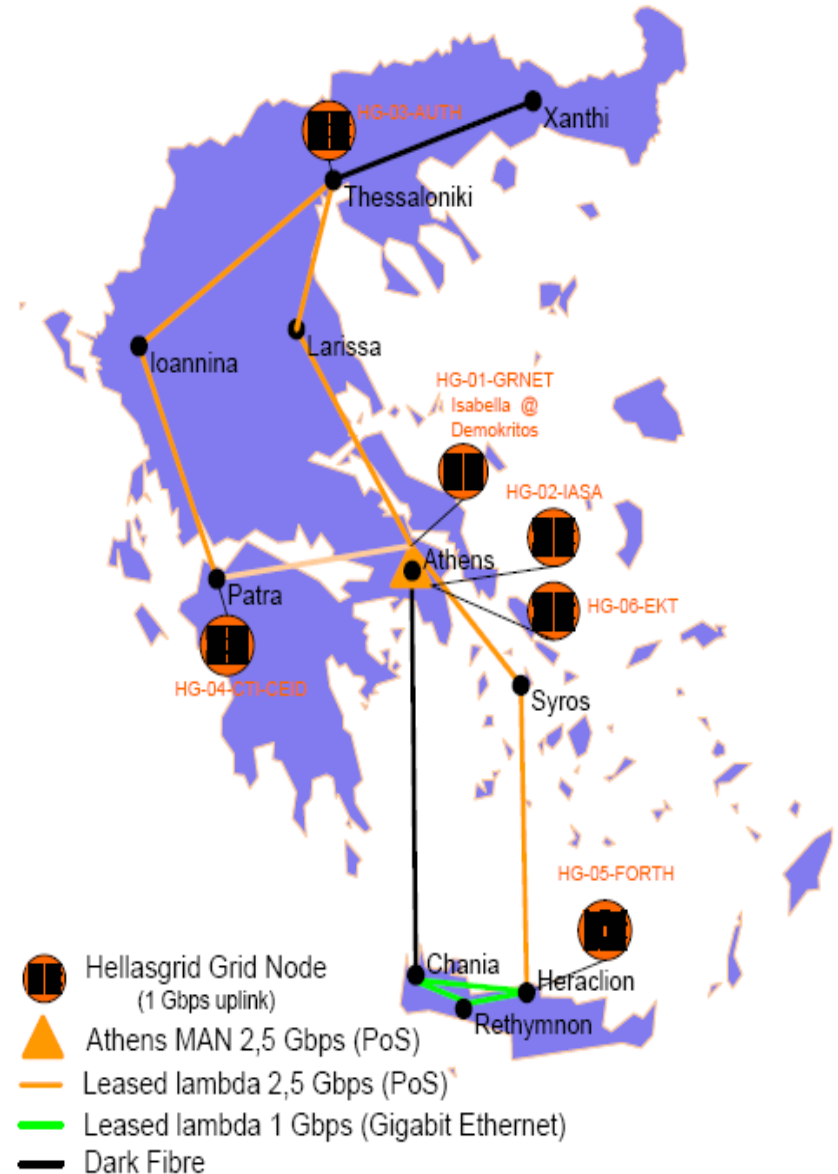


- **User directory:**
 - VO server & Myproxy
- **Resources directory:**
 - BDII (LDAP based!)
- **Computational Resources:**
 - Resource Broker (RB)
- **Storage Resources:**
 - LCG File Catalog (LFC)
- **Local infrastructures:**
 - CE & WNs, SE, UI etc.

- >700 64bit CPUs, 2GBs RAM systems
- >100 Tbytes storage area
- Gigabit interconnects, end-to-end

Design based on:

- GRNET2 backbone planning
- Placement of grid nodes within academic institutes
- Creation of human networks
- Creation of knowledge pools within universities and research facilities
- Use of AccessGrid for communication and management

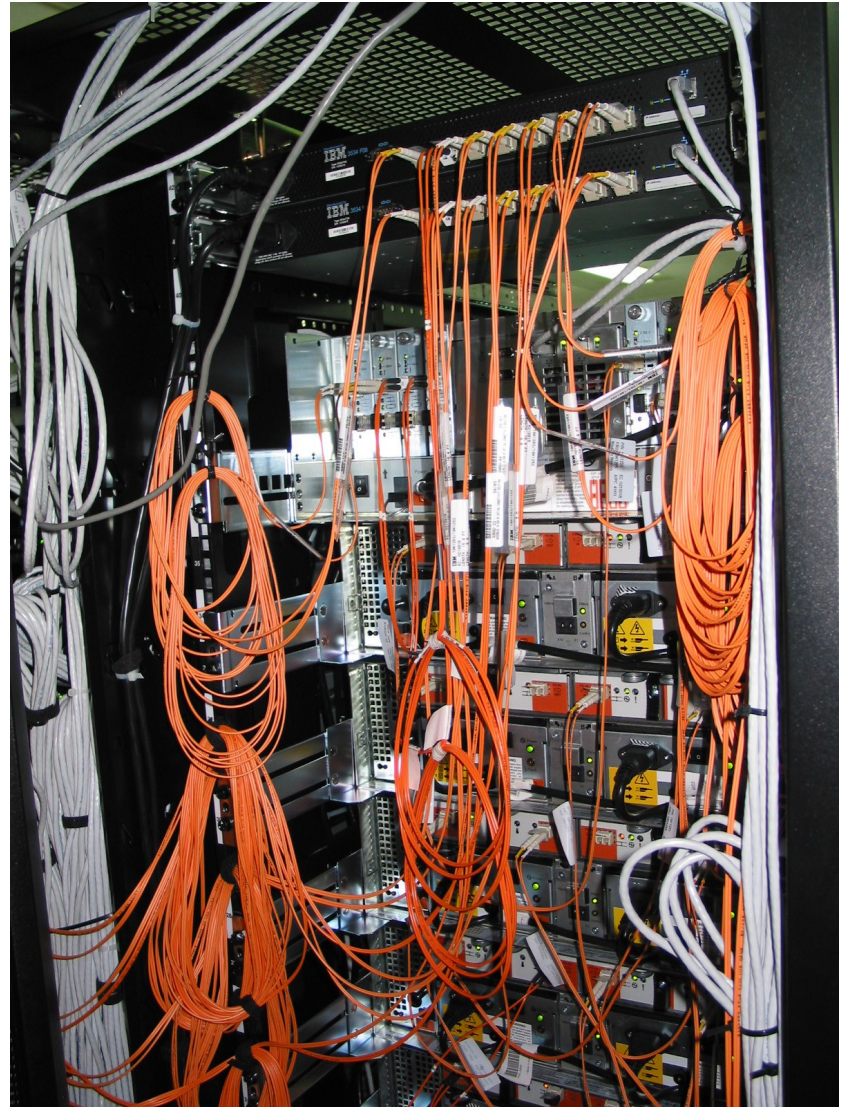


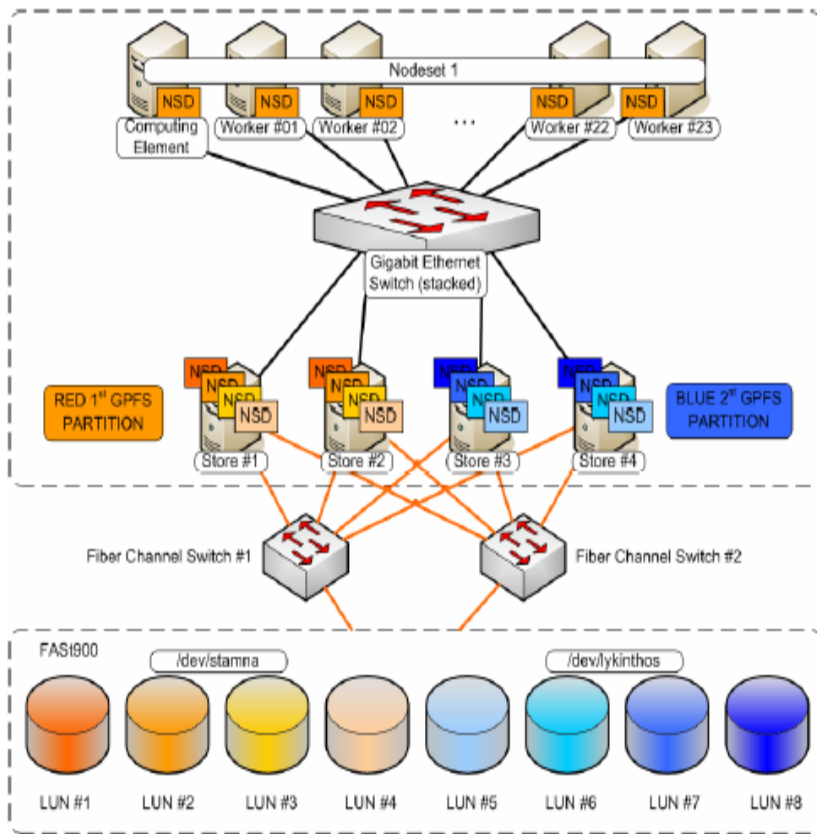
- **HellasGrid I**

- Located at Demokritos, Aghia Paraskevi, Athens (aka. Isabella)
- 34 dual Intel Pentium Xeon @ 2.8GHz, 1GB RAM, 2x70GB SCSI RAID1 HDD, 2x Gigabit
- IBM FAStT900 Storage Area Network, integrated system
 - Redundant Fibre Channel Controllers with 1GByte Cache
 - 70x146.8GB= **10,276TB raw storage capability**
 - Fully automated solution, **hot spare + hot swap**
- Tape Library with a capacity up to ~30 TBytes
- Delivered by IBM to GRNET during December 2004

- **HellasGrid II**

- 5 more physical nodes: EKT, IASA, AUTH, FORTH, UPATRAS
- ~700 CPUs x86_64, 2 GB RAM, 80GB SATA HDD, 2x Gigabit
- ~20 TBytes total storage capacity provided by SAN solutions
- ~50 TBytes Tape Library



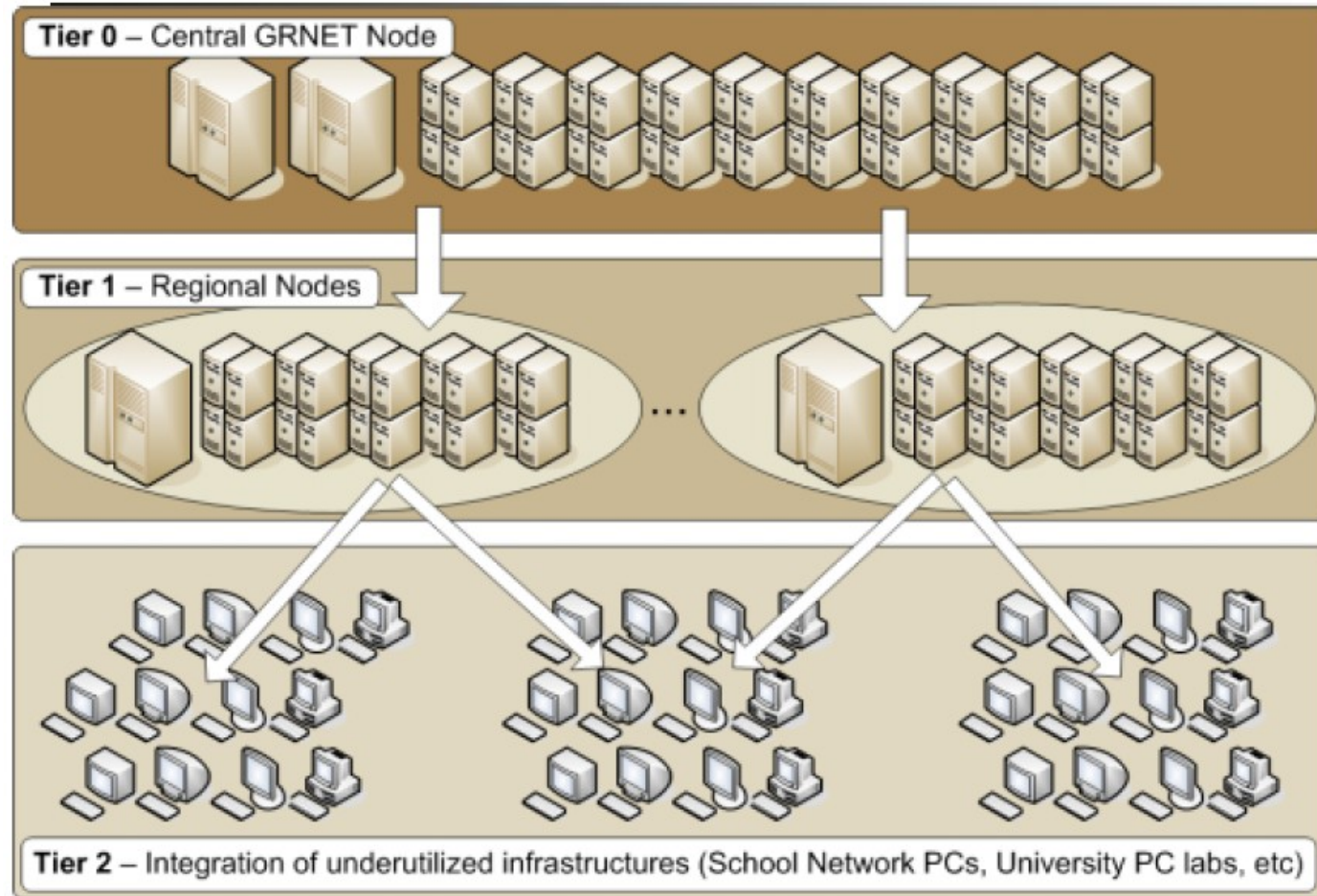


- The first node of the HellasGrid infrastructure has been a great tool for building a knowledge base.
- The experience with it has been exploited during the second phase of the project, in benefit of the newer nodes and users.
- Outstanding and very unconventional organization of the SAN system and its filesystems.

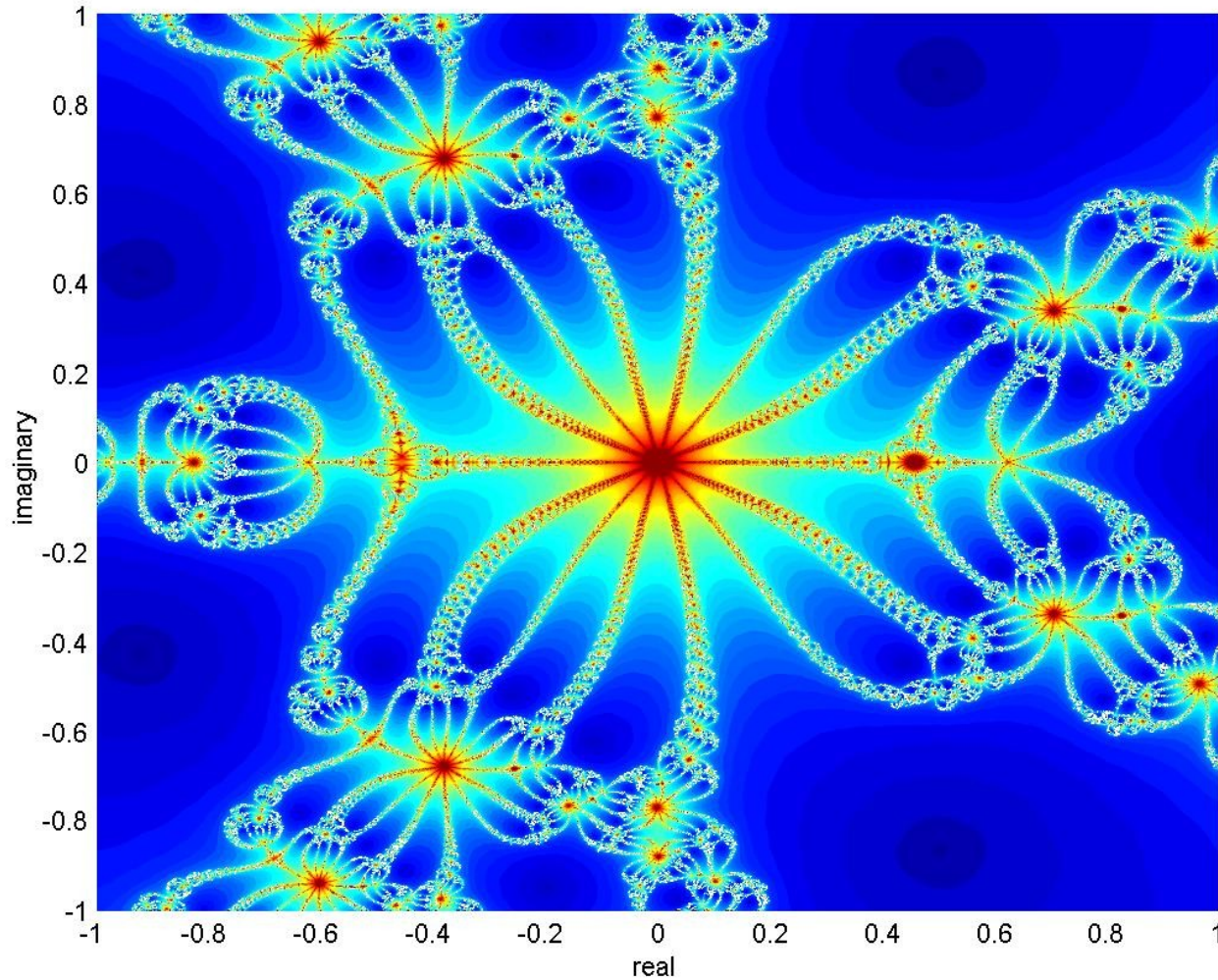
**Tier 0 –
Central GRNET nodes
(Computing Power,
Storage resources
& Core Grid Services)**

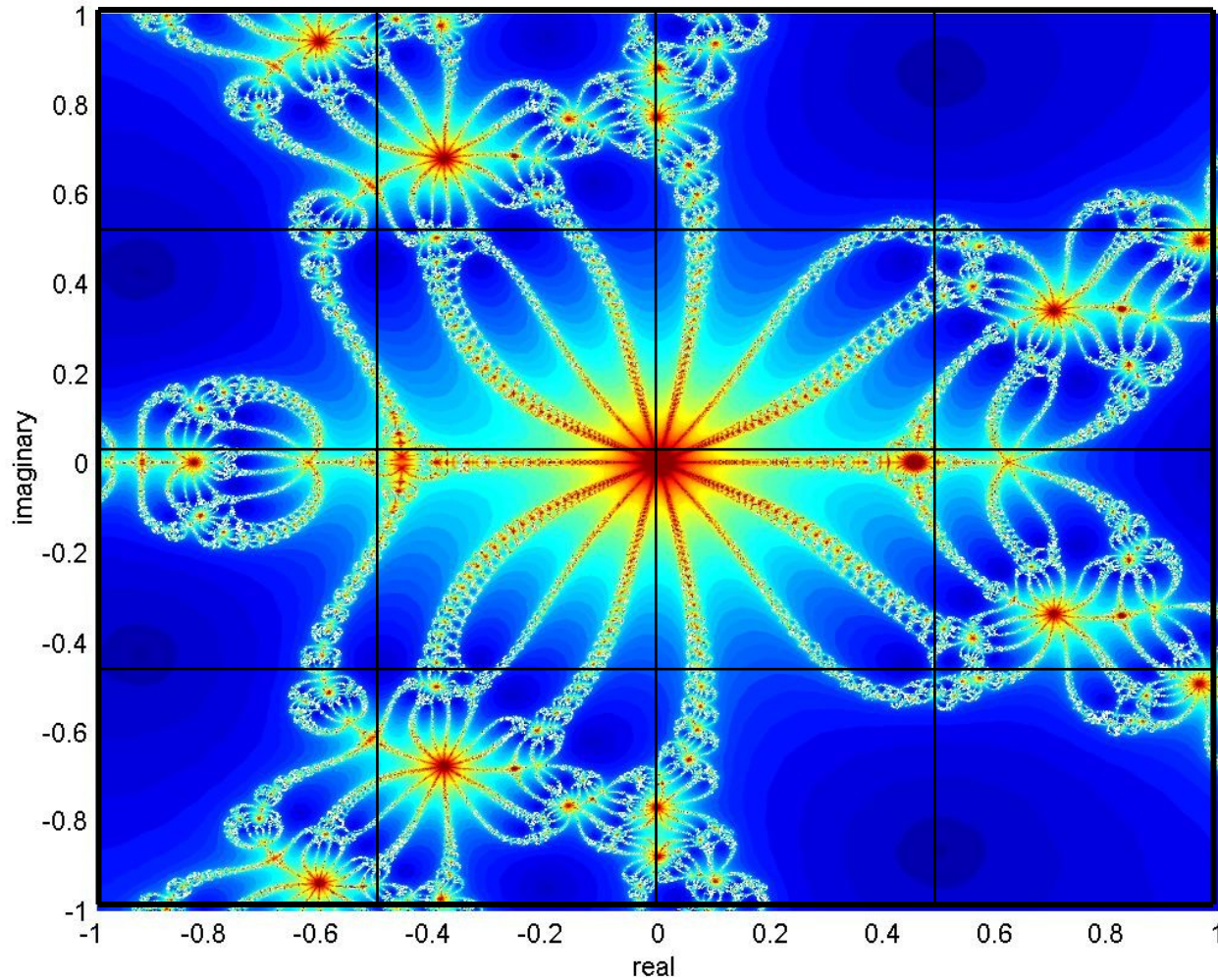
**Tier 1 –
5 regional nodes:
Athens (2)
Patras,
Thessaloniki,
Creta**

**Tier 2 – Integration of
other infrastructures
(sch.gr, Laboratories,
other scavenged CPUs)**



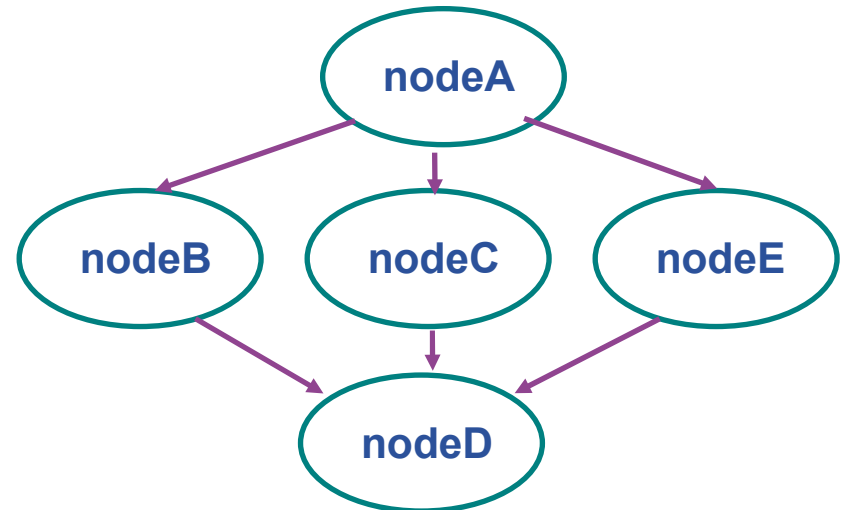
- **Simulations (CPU intensive, need lots of CPUs)**
- **Bulk data processing (lots of storage, lots of CPU)**
- **Parallel jobs (typically a few MPI supporting clusters)**
 - MPI fully supported in many (huge) clusters within SEE (40-200)
 - Interconnects are typically Gigabit, but some sites have Myrinet!
 - Constellation-like calculations should be possible in the future
 - HellasGrid is designed with Gbit provision e2e. Take advantage!
- **We'd like to see: workflows of ~medigrid applications**
 - Weather models, Fire behaviour, Flooding, Landslides etc
 - Anything with direct or indirect impact on level-of-living for SEE
- **We'd like to see: Scavenged resources & Applications**
 - We can do that, LiveWN proved that it is technically possible
 - Needs lots of experimentation on resource & job matchmaking
 - Needs lots of evolution in the Information System (Glue Schema)



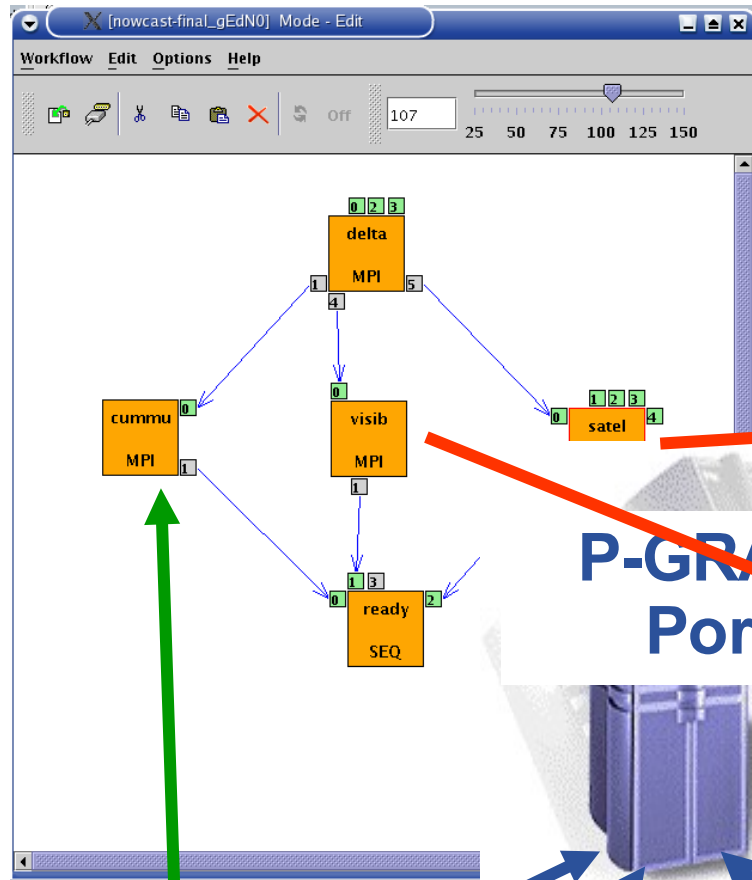


- **Workflow** is a set of **related jobs**, whose execution results are interdependent, either because they have to finish concurrently or, because the output of one job is input for another -in a **tree** or **chain** setup- etc
- They are also known as **DAGs**, standing for **Directed Acyclic Graphs**, which is indicating their structure
- A **flat workflow** is the common case of **embarrassingly parallel** tasks, which are the most trivial grid usage
- Flat workflows can often be described as **parametric jobs** which means they are defined in domain (1..N).
- **Enumerable sets** from **mathematical set theory** would help you understand if your application is parametric; A surprisingly high number of applications are such!
-

- **Direct Acyclic Graph (DAG)** is a set of jobs where the input, output, or execution of one or more jobs depends on one or more other jobs
- **A Collection** is a group of jobs with no dependencies
 - basically a collection of JDL's
- **A Parametric job** is a job having one or more attributes in the JDL that vary their values according to parameters
- **Using compound jobs** it is possible to have one shot submission of a (possibly very large, up to thousands) group of jobs
 - Submission time reduction
 - Single call to WMPProxy server
 - Single Authentication and Authorization process
 - Sharing of files between jobs
 - Availability of both a single Job Id to manage the group as a whole and an Id for each single job in the group

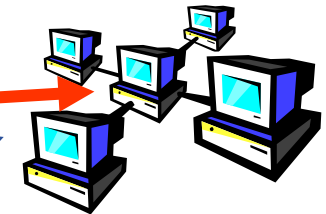


- **gLite WMS** includes a special facility for execution of **DAGs and parametric jobs**, which make the life of a developer much easier and turn the grid into a powerful automatic tool working on the background.
- You will either have to execute your workflows with **your own custom scripts under Unix** or, use one of:
 - **P-Grade portal**, which can be used as a workflow design and management environment; it is being setup in Patras University
 - **Genius portal** from INFN/Catania provides similar functionality as above, but appears somewhat more closed project
 - **Ganga** is a job submission framework widely accepted within the LHC VOs. **Panda** is being added to it for ease-of-use.
 - **Taverna** is a workflow design environment, language AND enactment engine from **BIOMEDICAL community** people
 - ...there are more, we find a new one every month!



Different jobs of a workflow can be executed in different grids

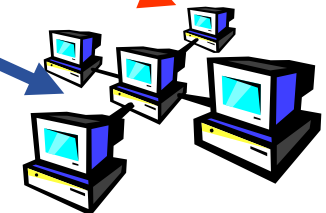
EGEE Grid
e.g. VOCE



P-GRADE-Portal

The portal can be connected to multiple grids

UK NGS

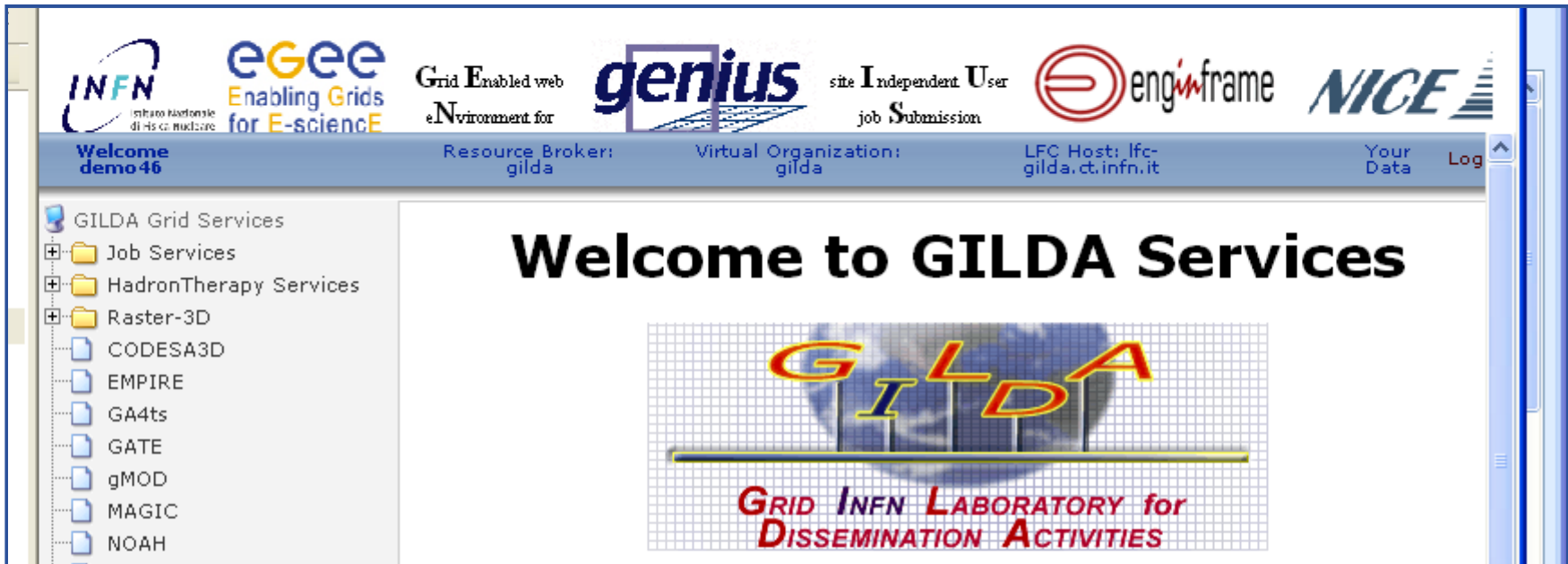


London

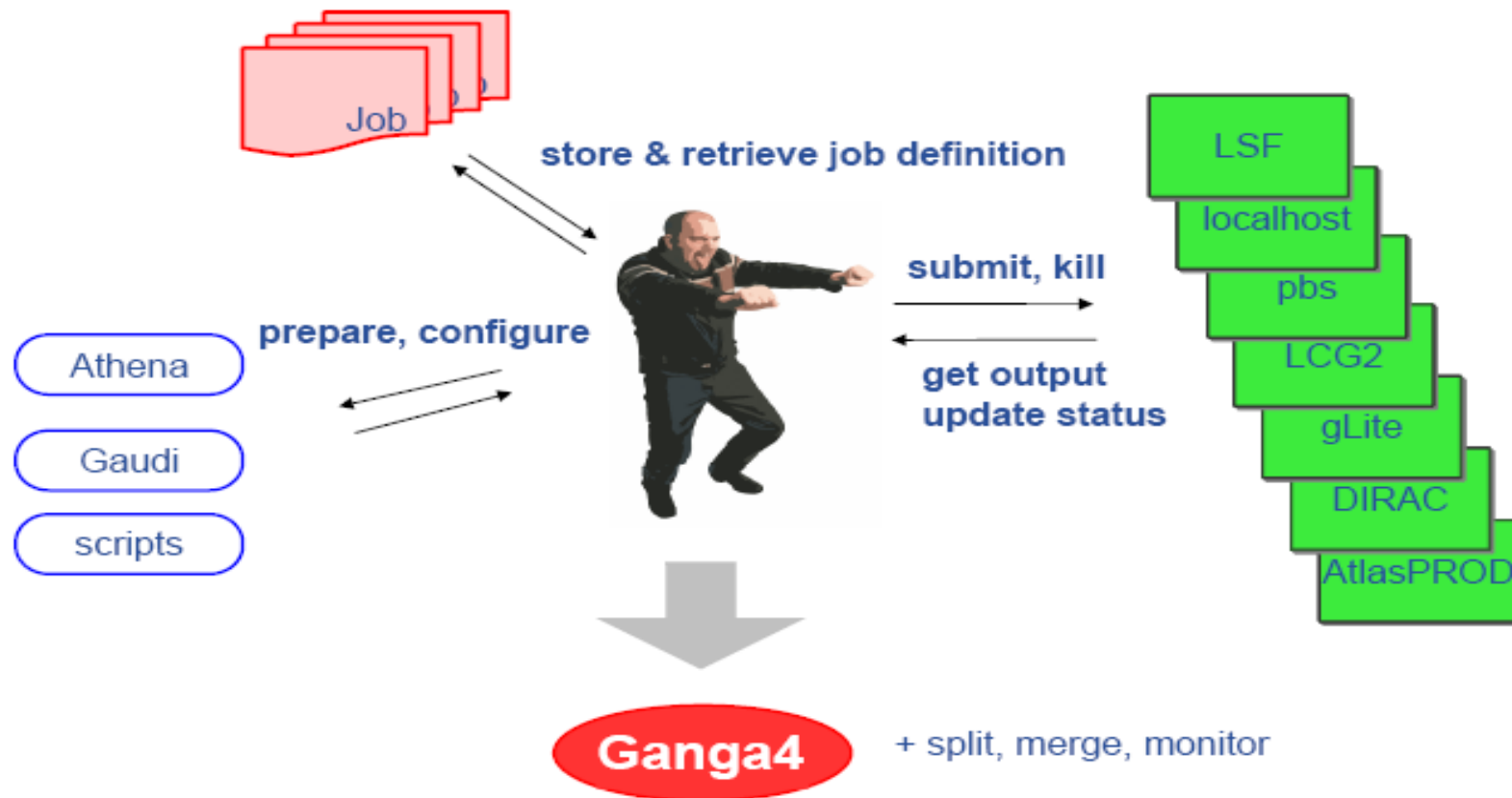
Rome

Athens





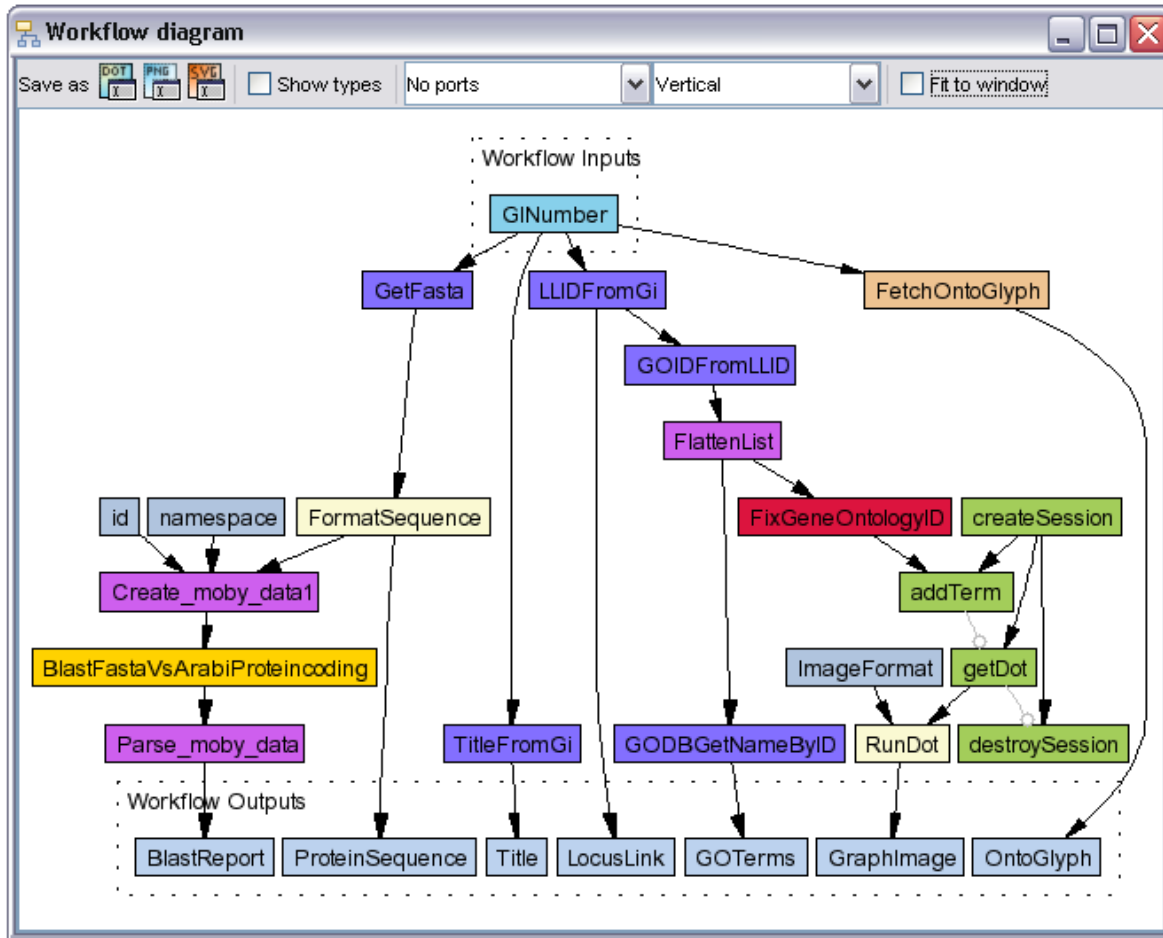
- **For many application communities**
 - Interface can be tailored for specific requirements
- **For demonstration purposes**
 - <https://glite-demo.ct.infn.it/>
 - Available for anyone to use
 - <https://glite-tutor.ct.infn.it/>
 - Fuller functionality for users who have stored long-lived proxy in MyProxy server



- **Ganga is a lightweight user tool**
ganga.web.cern.ch/

- **But also: Ganga is a developer framework**

- ▶ Taverna in MyGrid <http://www.mygrid.org.uk/>
- ▶ “allows the e-Scientist to describe and enact their experimental processes in a structured, repeatable and verifiable way”
- ▶ GUI
- ▶ Workflow language
- ▶ Enactment engine

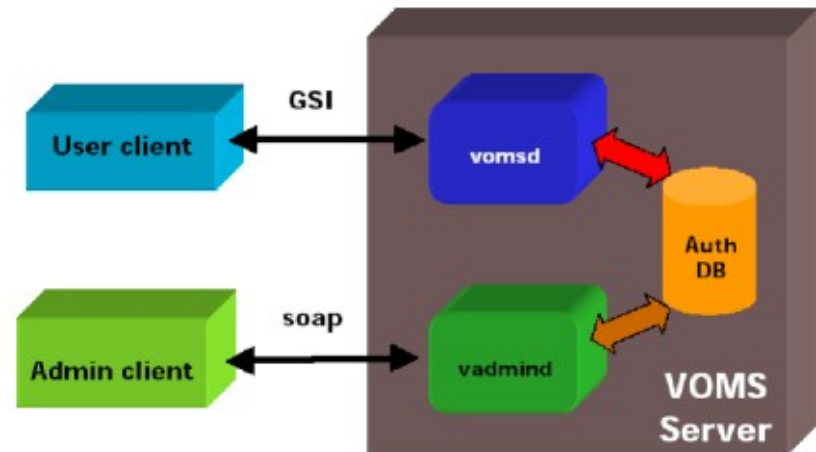


- **Credential** is a token you use for **authentication**; typically on a grid this is a **proxy X.509 certificate**
- Every VO provides a **VOMS service** where your **certificate** should be listed, once you are a member
- **MyProxy** is a service of **online credential repository**
- **MyProxy** is necessary for **credential renewal** which is particularly useful for **long-running jobs**
- It is a much safer storage for long-term certificates

Ref: <http://grid.ncsa.uiuc.edu/myproxy/> (++)

Virtual Organization Membership Service

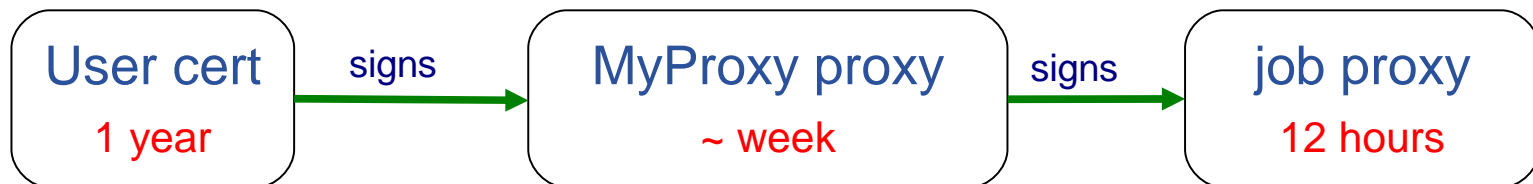
- Multiple **VOs**
- Multiple **roles** in VO
 - compatible X509 extensions
 - signed by VOMS server
- Web admin interface
- Supports MyProxy
- Resource providers grant access to **VOs** or **roles**
- Sites map VO members/roles to local auth mechanism (unix user accounts)
 - allows for local policy



Layer of abstraction: individual members irrelevant

MyProxy...

- **...allows longer lived jobs / increases security**
 - W M S renews proxy
 - users should not produce long lived proxies :-)
- **...allows for secure user mobility**
 - user does not need to copy gbus-keys around
- **...stores medium-lived proxy (days ~ weeks)**



- **AMGA is a Metadata service for gLite-based grids**
- **Metadata is data about data**
- **On the Grid: information about files**
 - Describe files
 - Locate files based on their contents, categories, keywords, etc
- **But also we need simplified DB access on the Grid**
 - Many Grid applications need structured data
 - Many applications require only simple schemas
 - Can be modelled as metadata
 - Main advantage: better integration with the Grid environment
 - Metadata Service is a Grid component
 - **Grid security**
 - Hide DB heterogeneity

- **AMGA – ARDA Metadata Grid Application**
 - ARDA: A Realisation of Distributed Analysis for LHC
- **Now part of gLite middleware**
 - Official Metadata Service for EGEE
 - Also available as standalone component
- **Expanding user community**
 - HEP, Biomed, UNOSAT...
 - Nearly anyone who needs an RDBMS solution

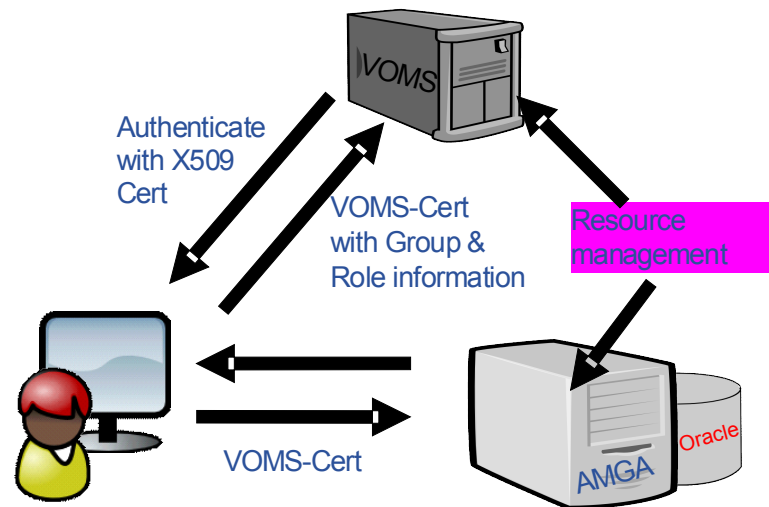
- **AMGA** provides access to metadata for files stored on the Grid, as well as a simplified general access to relational data stored in database systems
- **AMGA** is a service in front of a relational database back-end. Currently Oracle, PostgreSQL, MySQL and SQLite are supported
- **AMGA** has two different front-ends: a web service front-end implemented using gSOAP and a text-protocol (TCP, SSL) based front-end, which is able to stream data back to the client

- **Some Concepts**

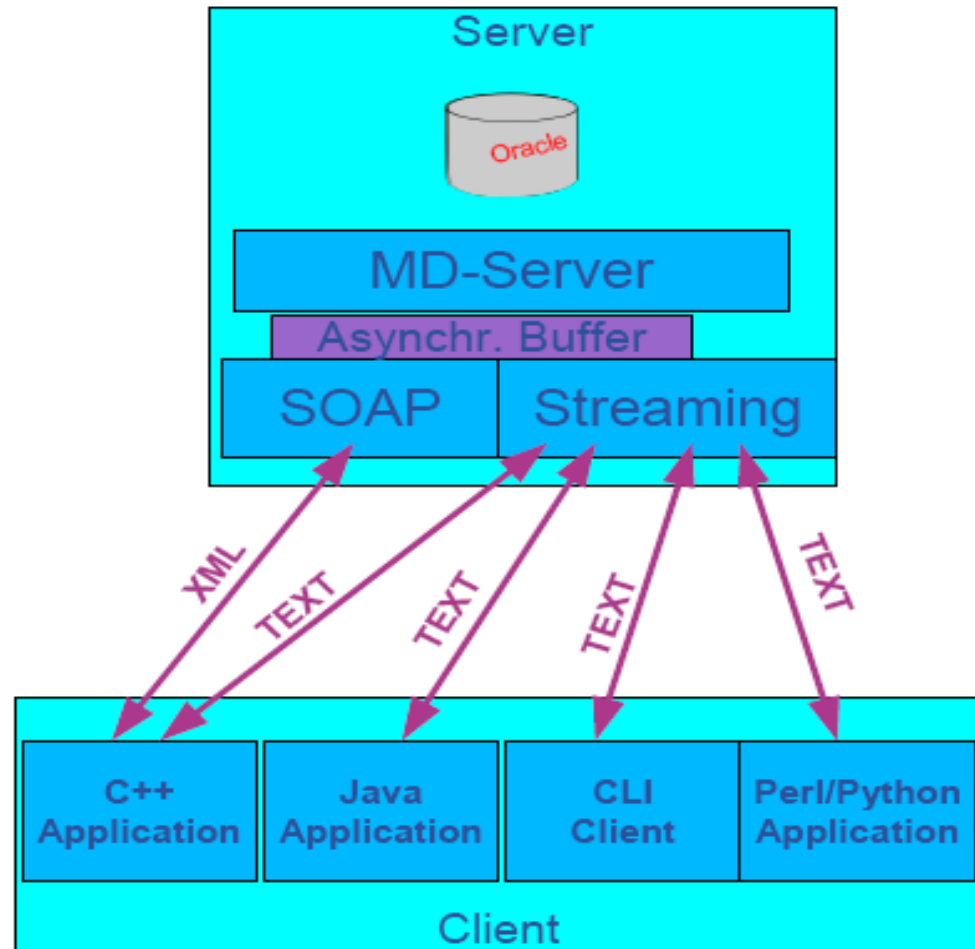
- **Metadata** - List of attributes associated with **entries**
- **Attribute** – key/value pair with type information
 - **Type** – The type (int, float, string,...)
 - **Name/Key** – The name of the attribute
 - **Value** - Value of an entry's attribute
- **Schema** – A set of attributes
- **Collection** – A set of entries associated with a schema
- Think of schemas as tables, attributes as columns, entries as rows

- **LHCb-bookkeeping examples**
 - Migrated bookkeeping metadata to ARDA prototype
 - 20M entries, 15 GB
 - Large amount of static metadata
 - Feedback valuable in improving interface and fixing bugs
 - AMGA showing good scalability
- **Ganga is using it, too**
 - Job management system
 - Developed jointly by ATLAS and LHCb VOs
 - Uses AMGA for storing information about job status
 - Small amount of highly dynamic metadata

- **Unix style permissions**
- **ACLs – Per-collection or per-entry.**
- **Secure connections – SSL**
- **Client Authentication based on**
 - Username/password
 - General X509 certificates
 - Grid-proxy certificates
- **Access control via a Virtual Organization Management System (VOMS):**



- **AMGA Implementation:**
 - SOAP and Text frontends
 - Streamed Bulk Operations
 - Supports single calls, sessions & connections
 - SSL security with grid certs (negotiated by client)
 - Own User & Group management + VOMS
 - PostgreSQL, Oracle, MySQL, SQLite backends
 - Works alongside LFC
 - C++, Java, Perl, Python clients

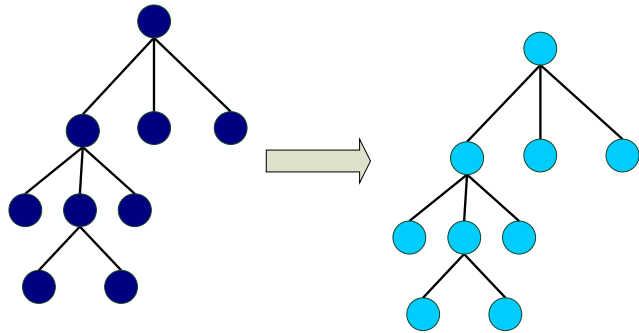


- **Dynamic Schemas**
 - Schemas can be modified at runtime by client
 - Create, delete schemas
 - Add, remove attributes
- **Metadata organised as an hierarchy**
 - Schemas can contain sub-schemas
 - Analogy to file system:
 - Schema ⇔ Directory; Entry ⇔ File
- **Flexible Queries**
 - SQL-like query language
 - Joins between schemas

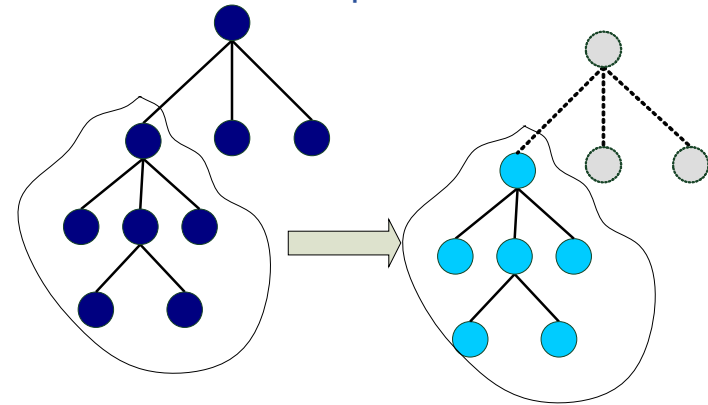
- Currently working on **replication/federation** mechanisms for AMGA
- **Motivation**
 - **Scalability** – Support hundreds/thousands of concurrent users
 - **Geographical distribution** – Hide network latency
 - **Reliability** – No single point of failure
 - **DB Independent replication** – Heterogeneous DB systems
 - **Disconnected computing** – Off-line access (laptops)
- **Architecture**
 - Asynchronous replication
 - Master-slave – Writes only allowed on the master
 - Replication at the application level
 - Replicate Metadata commands, not SQL → DB independence
 - Partial replication – supports replication of only sub-trees of the metadata hierarchy

Some use cases

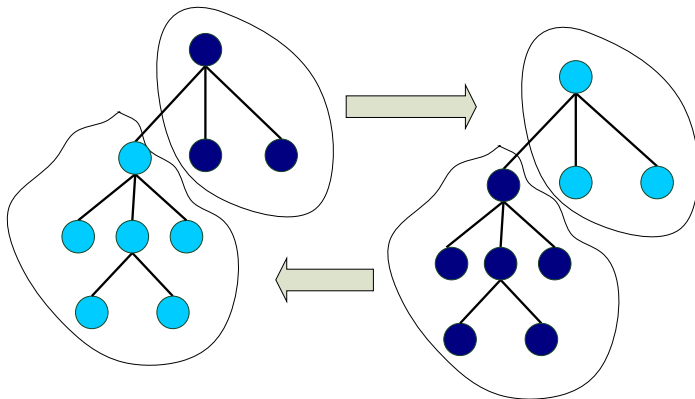
Full replication



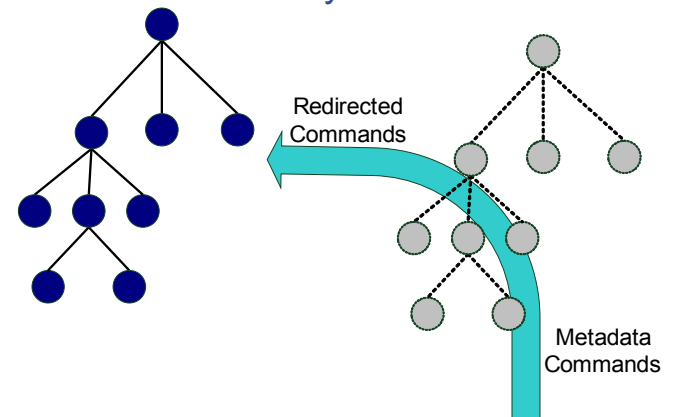
Partial replication



Federation



Proxy



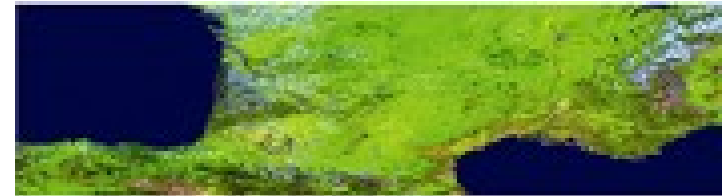
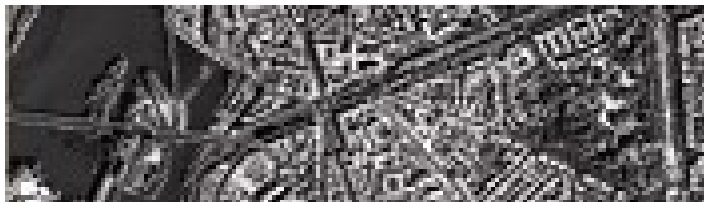
UNOSAT is a United Nations Initiative

Objectives

- ➔ Provide the humanitarian community with access to satellite imagery and Geographic Information System services
 - ▶ Reduce disasters and plan sustainable development
- ➔ Ensure cost-effective and timely products

Core Services

- ➔ Humanitarian Mapping
- ➔ Image Processing



VEGETATION – 1 Km

IKONOS – 1m

- **Potential Bottlenecks:**

- UNOSAT beginning to suffer from limited capacity and processing power
 - Multiple satellites being launched
 - Larger and larger storage capacity needed

- **In summer 2005 CERN provided a whole structure for UNOSAT:**

- UNOSAT Virtual Organization (VO)
 - 3.5TB in CASTOR
 - Computing Elements, Resource Brokers
 - Collaboration with ARDA group
 - AFS area of 5GB

We have provided
the whole GRID
infrastructure
At CERN

- **We have run some UNOSAT tests (images compression) inside the GRID environment (quite successful)**

- **The framework developed for in principle for Geant4 (See Alberto Ribon's presentation) has been adapted for UNOSAT needs**

- UNOSAT provided a set of images for testing
- Associated to each image a metadata file was included
File name, directory path, geographical coordinates

▫ Steps:

STORAGE LEVEL

- ➔ Copy and registration of the images in Castor@CERN
 - ▶ Use of the LFC Catalog
- ➔ Parse the metadata files to extract the different metadata
- ➔ Use of the AMGA tool to parse metadata to location of the files

COMPUTING LEVEL

- ➔ Use of compression tools to compress images inside LCG resources
- ➔ Use of the general submission tool adapted to UNOSAT needs

☐ LFC Catalogue

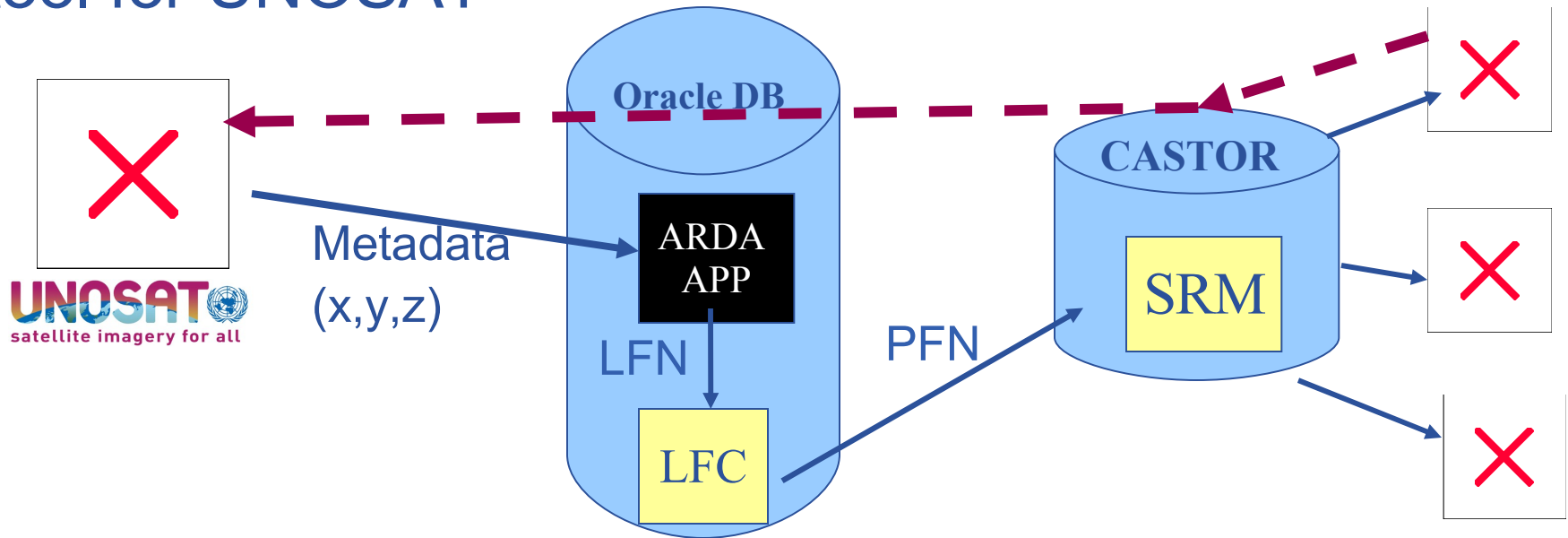
➔ Mapping of LFN to PFN

☐ UNOSAT requires

➔ User will give as input data certain coordinates

➔ As output, want the PFN for downloading

☐ The ARDA Group assists us setting up the AMGA tool for UNOSAT



Medical Data Manager – MDM

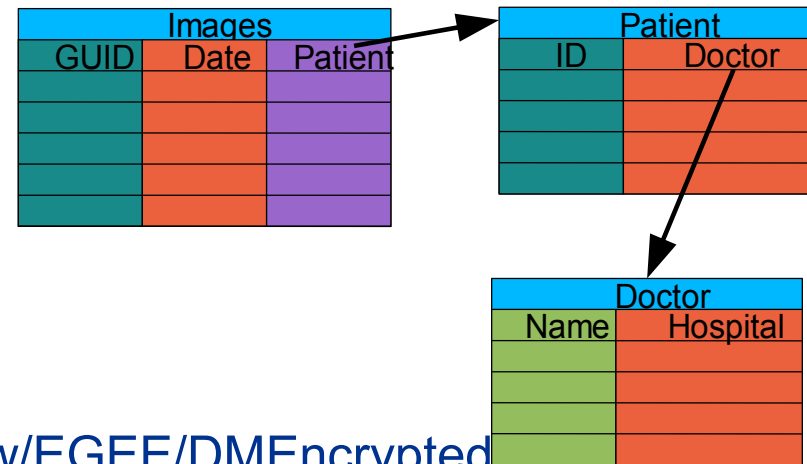
- Store and access medical images and associated metadata on the Grid
- Built on top of gLite 1.5 data management system
- Demonstrated at last EGEE conference (October 05, Pisa)

Strong security requirements

- Patient data is sensitive
- Data must be encrypted
- Metadata access must be restricted to authorized users

AMGA used as metadata server

- Demonstrates authentication and encrypted access
- Used as a simplified DB



More details at

- <https://uimon.cern.ch/twiki/bin/view/EGEE/DMEncryptedStorage>

- **on AMGA and gLibrary:**

- <http://indico.eu-eela.org/conferenceTimeTable.py?confId=37>
- (go to day 3 for the AMGA tutorial)

- **AMGA Web Site**

<http://project-arda-dev.web.cern.ch/project-arda-dev/metadata/>

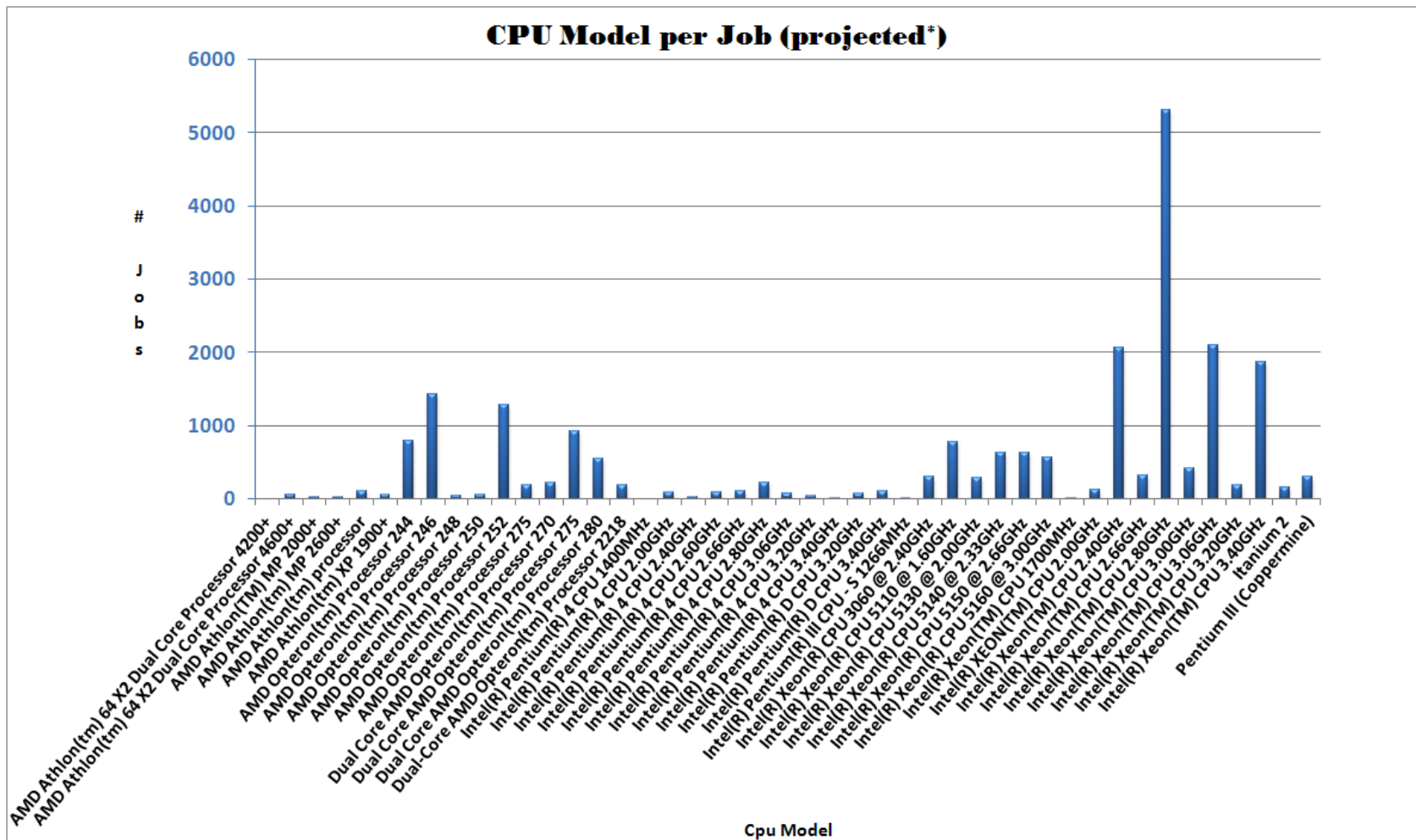
- **Simulations (CPU intensive, SEE has >1000 of CPUs)**
 - CPU-intensive, they have to be profiled & optimized
 - >700 64bit provided by HellasGrid II clusters alone,
 - twice as many systems (mostly 32bit) in whole SEE region
 - HOW: In SEE we have a range of x86 execution environments
 - all are 32bit i386 compatible: IA32, IA64, Athlons, Opterons etc.
 - They can vary considerably in performance, see next slide
- **Bulk data processing (Storage, SEE has >100TBs+CPU)**
 - Involve the management of huge datasets
 - Best performance occurs if data is distributed/replicated
 - They might involve workflow and/or meta-data techniques
 - Plus, similar computing needs as simulations
 - HOW: Create replicas and initiate a distributed workflow
 - We have plenty of Storage >100TBs by HellasGrid
 - Create *at least* three replicas of each dataset "slice", on HG sites.

- **Responsive Applications**
 - Interactive applications have been attempted in the past
 - During CrossGrid there has been success, if Grid was unloaded
 - Short response time, requires low-latency, QoS
 - In the future we may be able to pause-reanimate jobs on demand
 - Necessary feature to support Urgent Computing (EQ, floods, fires)
 - Middleware has to evolve further, in order to support them well
 - HOW: Not simple, but not impossible. It requires VO coordination.
- **Workflows**
 - It can involve:
 - a GUI to design it
 - a language to represent it
 - Enactment engine
 - Typically used to tackle any sufficiently complex problem
 - HOW: Look for tools like Ganga, Taverna, P-Grade, Genius etc.

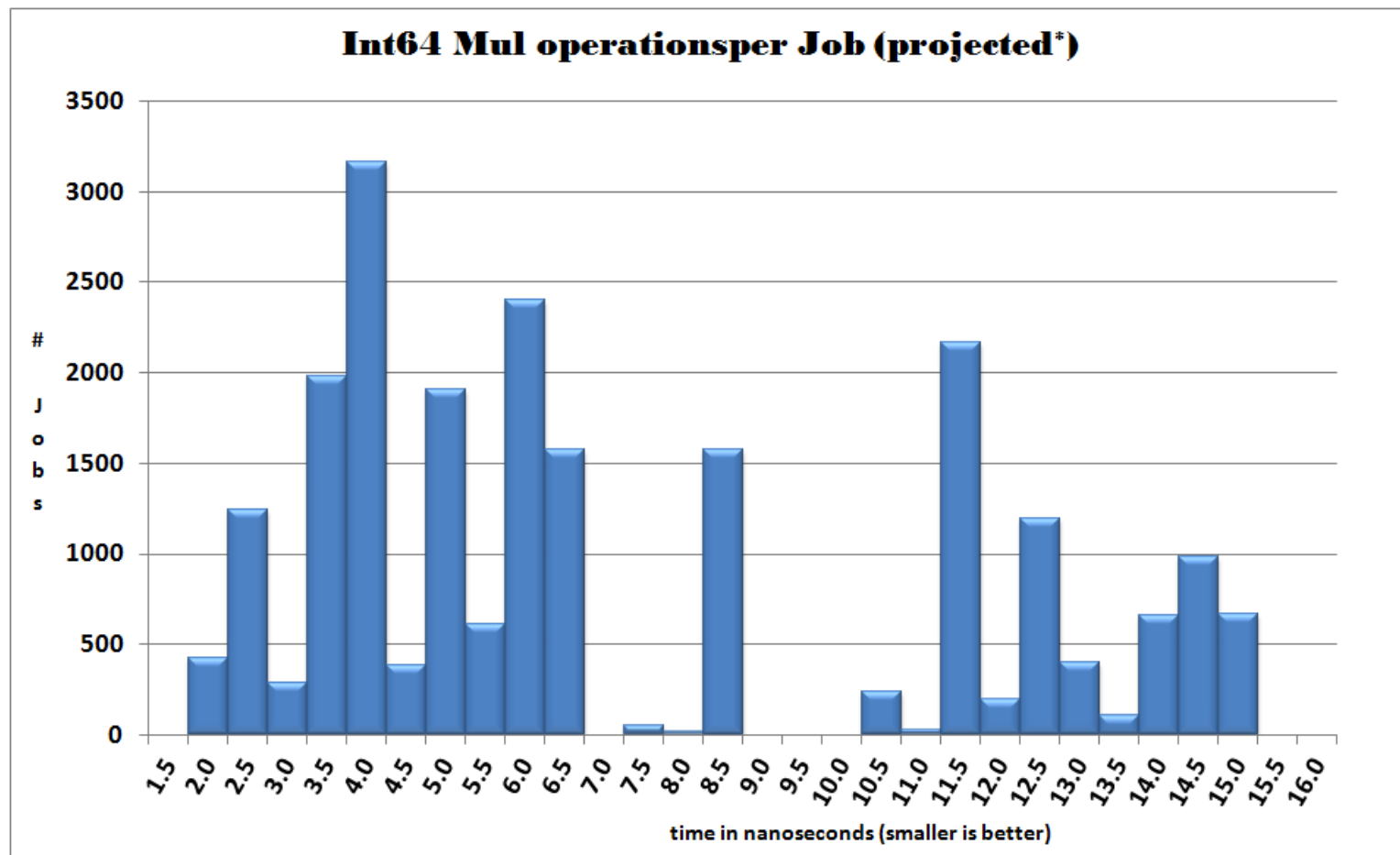
- **Parallel jobs**
 - Interdependent, communicating tasks
 - They have to use an interconnect (hardware) and MPI (software)
 - Very critical to ensure proper MPI libraries are there
 - Applications might have to be recompiled
 - It should be possible (in the future) to do constellation calculations
 - HOWTO: Request support from NA4 activity, ASAP.
- **Legacy applications**
 - Commercial or closed source binaries
 - User must ensure that there is at least one compatible license model, that can be applied on the grid (or request for it!)
 - It is a serious issue that they can't be reengineered for the grid
 - FUSE/ELFI, flexlm somewhere on a central server
 - HOWTO: Try to deploy it (Cross your fingers that it will run OK...)

Basic Number Operations - in nanoseconds (smaller is better)																	
SiteName	Specifications	Float			Double			Integer				Integer 64bit					
		add	mul	div	add	mul	div	bit	add	mul	div	mod	bit	add	mul	div	mod
ce.hep.ntua.gr	Intel,P4,2.8GHz,1024MB,L.SATA	2,26	3,65	16,48	2,25	3,67	16,42	0,38	0,36	3,59	23,26	30,07	3,86	5,61	6,29	94,44	92,97
ce.phy.bg.ac.yu (lcg)	Intel,Xeon,2.8GHz,1024MB,NFS	3,00	3,41	16,56	3,96	2,85	15,93	0,42	0,28	5,26	21,48	25,51	3,55	4,24	11,85	127,71	154,17
ce.ulakbim.gov.tr	Intel,P4,2.66GHz,1024MB,NFS	1,88	2,64	16,26	1,88	2,63	16,26	0,19	0,19	5,30	21,79	24,26	3,24	3,41	13,00	80,82	95,61
ce001.imbm.bas.bg	Intel,P4,3GHz,1024MB,NFS	1,78	2,50	15,43	1,78	2,50	15,43	0,18	0,18	5,03	20,68	23,01	3,08	3,25	12,36	76,68	90,76
ce001.ipp.acad.bg (glite)	Intel,P4,3GHz,1024MB,L.SATA	2,00	2,67	15,11	2,01	2,68	15,14	0,33	0,33	3,34	20,72	27,80	3,22	5,58	5,46	79,15	70,70
ce002.ipp.acad.bg (lcg)	Intel,P4,2.4GHz,1024MB,L.SATA	2,06	2,84	17,10	2,06	2,84	17,10	0,25	0,25	5,05	23,08	27,36	3,58	4,39	11,49	85,22	94,42
ce01.afroditi.hellasgrid.gr	Intel,Xeon,3.4GHz,2048MB,GPFS	1,76	2,35	13,30	1,76	2,36	13,31	0,29	0,29	2,94	18,24	24,63	3,13	4,91	4,85	67,56	62,02
ce01.ariagni.hellasgrid.gr	Intel,Xeon,3.4GHz,2048MB,GPFS	1,76	2,35	13,30	1,76	2,35	13,30	0,29	0,29	2,94	18,24	24,64	3,16	4,91	4,84	67,81	62,06
ce01.athena.hellasgrid.gr (l)	Intel,Xeon,3.4GHz,2048MB,GPFS	1,76	2,35	13,31	1,76	2,35	13,31	0,29	0,29	2,98	18,24	24,52	3,13	4,91	4,85	66,10	61,91
ce01.athena.hellasgrid.gr (s)	Intel,Xeon,3.4GHz,2048MB,GPFS	1,76	2,35	13,31	1,76	2,35	13,30	0,29	0,29	2,94	18,24	24,47	3,14	4,91	4,83	65,07	61,93
ce01.isabella.grnet.gr	Intel,Xeon,2.8GHz,1024MB,GPFS	2,54	10,89	15,54	2,85	5,14	15,52	0,27	0,22	5,18	22,19	25,14	3,43	3,74	12,91	91,98	108,10
ce01.kallisto.hellasgrid.gr	Intel,Xeon,3.4GHz,2048MB,GPFS	1,76	2,35	13,31	1,76	2,39	13,31	0,32	0,31	2,96	18,59	24,64	3,20	5,20	5,25	77,58	61,94
ce01.marie.hellasgrid.gr	Intel,Xeon,3.4GHz,2048MB,GPFS	1,77	2,36	13,30	1,76	2,35	13,30	0,29	0,31	2,94	18,24	24,54	2,98	5,11	4,82	67,94	62,18
ce02.athena.hellasgrid.gr (l)	Intel,Xeon,3.4GHz,2048MB,GPFS	1,76	2,35	13,30	1,76	2,35	13,30	0,29	0,29	2,94	18,24	24,48	2,99	4,91	4,81	69,68	62,07
ce02.athena.hellasgrid.gr (s)	Intel,Xeon,3.4GHz,2048MB,GPFS	1,76	2,35	13,30	1,76	2,35	13,30	0,30	0,29	2,95	18,24	24,67	3,10	4,92	4,90	65,04	62,03
ce02.marie.hellasgrid.gr	Amd,Athlon,1.6GHz,1024MB,NFS	2,51	2,50	15,05	2,50	2,50	15,06	0,63	0,63	2,50	25,67	26,90	3,64	2,62	7,37	82,58	80,89
ce101.grid.ucy.ac.cy	Amd,Opteron,2.5GHz,1024MB,L.S	1,54	1,54	9,30	1,55	1,54	9,29	0,39	0,39	1,16	15,83	16,21	2,57	1,72	3,33	48,02	45,70
ctb31.gridctb.uoa.gr	Intel,Xeon,3GHz,2048MB,L.SATA	2,00	2,67	15,07	2,00	2,67	15,07	0,33	0,33	3,33	20,67	27,72	3,53	5,57	5,44	78,99	70,20
cox01.grid.metu.edu.tr	Intel,Xeon,1.6GHz,4096MB,NFS	1,88	3,13	23,89	1,88	3,13	23,90	0,63	0,63	1,88	20,88	12,71	3,76	2,75	5,99	70,59	62,16
g02.phy.bg.ac.yu (glite)	Intel,Xeon,2.8GHz,1024MB,NFS	4,68	5,79	16,64	4,69	5,62	16,69	0,43	0,27	5,13	21,66	24,18	3,67	4,07	11,46	115,52	126,76
grid-ce.ii.edu.mk	Intel,Xeon,2.8GHz,1024MB,NFS	4,79	5,46	15,52	4,23	5,81	15,54	0,51	0,29	5,25	22,75	24,57	4,08	3,94	12,14	114,79	124,91
grid001.ics.forth.gr	Intel,Xeon,2.8GHz,512MB,L.SATA	2,15	2,87	16,21	2,15	2,87	16,21	0,36	0,36	3,58	22,20	29,99	3,69	5,99	5,86	84,92	75,57
grid01.cu.edu.tr	Intel,Xeon,1.6GHz,4096MB,L.SATA	1,88	3,13	23,85	1,88	3,13	23,85	0,63	0,63	1,88	20,85	12,71	3,77	2,74	6,00	70,51	62,04
grid01.erciyes.edu.tr	Intel,Xeon,1.6GHz,4096MB,L.SATA	1,88	3,13	23,85	1,88	3,14	23,85	0,63	0,63	1,88	20,84	12,71	3,77	2,74	5,99	70,51	62,03
kalkan1.ulakbim.gov.tr (lcg)	Intel,Xeon,1.6GHz,4096MB,NFS	1,88	3,22	23,85	1,88	3,13	23,85	0,64	0,63	1,88	20,85	12,72	3,90	2,75	5,98	70,92	63,28
kalkan2.ulakbim.gov.tr (glite)	Intel,Xeon,1.6GHz,4096MB,NFS	1,88	3,13	23,85	1,88	3,13	23,86	0,63	0,63	1,88	20,85	12,71	3,78	2,74	5,99	70,54	62,07
node001.grid.auth.gr	Intel,P4,2.8GHz,1024MB,NFS	1,80	2,66	15,75	1,81	2,63	15,74	0,24	0,23	4,39	20,87	24,25	3,25	4,22	10,37	87,37	98,69
testbed001.grid.ici.ro	Intel,Xeon,2.4GHz,1024MB,NFS	1,96	4,96	18,03	5,97	4,39	18,02	0,27	0,22	5,56	23,06	25,64	3,34	4,08	13,63	101,39	108,73
wipp-ce.weizmann.ac.il	Intel,Xeon,3.2GHz,2048MB,LATA	2,78	4,42	32,02	2,78	4,42	32,00	0,82	0,82	3,91	34,44	38,94	5,75	4,67	8,96	127,22	123,45

- **VERY wide performance spectrum; be aware**
 - A Technical Report On Grid Benchmarking using SEE V.O
 - J. Kouvakis & F. Georgatos, <http://arxiv.org/abs/cs/0703086>



- **VERY wide performance spectrum; be aware**
 - A Technical Report On Grid Benchmarking using ATLAS V.O
 - J. Kouvakis & F. Georgatos, <http://www.arxiv.org/abs/0705.1915>



- **VERY wide performance spectrum; be aware**
 - A Technical Report On Grid Benchmarking using ATLAS V.O
 - J. Kouvakis & F. Georgatos, <http://www.arxiv.org/abs/0705.1915>

- **Maybe you are lucky, and the application is there.**
- **Ask the NA4 about it! They are doing this job for you!**
 - If it is Free software+RPMs (like gcc, octave, anything in Quantian) should be easily accepted for evaluation.
- **or, you could find the source code, compile it, and send it upon job submission**
- **If it is a big executable, you can use DM to replicate it.**
- **You could also compile it at run-time on the WN!**
 - This is quite appropriate for MPI programs on Opterons.
- **Since HellasGrid has Intel processors, and is half of the infrastructure, it is best to use their own compiler, which might imply having your own interface. It is a big overhead, but you might even see 100% speedup (*2).**

Final advice: If you don't know what to do, do what you know, what is simple and works for you.

- **Do clusters have Worker Nodes with...**
 - Same OS? Same CPU? Same clockrate? Same RAM? ...
- **There is some heterogeneity in the resources provided by a given cluster (eg. slightly different clockrates).**
- **It is imposed by the combination of hardware diversity, the capabilities of the IS, ie. GLUE schema v1.x, and above all, the exact middleware implementation!**
- **GLUE schema v2 is a future proposal, which will allow the description of "subclusters", so this issue should be overcome; should be decided within 2007.**
- **For the time being, ask users to put their minimum job requirements in a .jdl, and if they see an error, to report immediately to Operations (SA1)'s help-desk tools.**

- **Check-pointing is the technique of storing the state of your job, typically in a set of files, so that you can later "reanimate" it. It is very critical for "long jobs".**
- **If jobs take more than 24 hrs, it is imperative for users to consider check-pointing for a number of reasons:**
 - User: software engineering reasons, ie. debugging a checkpointable job is simpler to debug on longer-term issues, since it is possible to restart it on eg. the 6th day of execution
 - User: overcome the limit of grid queues (infinite jobs!) and be able to pause and reanimate the job arbitrarily
 - User: by limiting the execution time of your jobs, you can now tap much more resources (queues), which will decrease the total time
 - Systems administration: cluster downtimes can happen for either scheduled reasons (m/w upgrade, hardware installation or reconfiguration etc) or various unscheduled reasons (power/network/airconditioning outages, security issues)

- **There are currently two well-known techniques to implement checkpointing (c/p):**
 - Update some central service eg. some database like AMGA, about the current state of the job (easy, if state==integer)
 - Save the state of the job in a set of files, make a tarball (.tar.gz) & register it on a nearby SE with a predefined LFN!
- **Middleware is supposed to be able to assist, but in gLite v3.0 this is currently unsupported function (?)**
- **In effect, workflow and job management tools (ganga?) do state-maintenance at the large-scale of a gridified application, can this be integrated with checkpointing?**
- **Maybe future tools that are able to do workflow management, will be able to implement this internally.**
- **A known trade-off: high-throughput vs low latency c/p**

- **Medical applications might want to set-up special encryption or anonymization techniques for their data, eg. Hydra, and perhaps enter negotiations with GRNET for Confidentiality Agreements (this will not be easy)**
- **Security and flexibility are well known trade-offs**
- **Beware of IEEE 754 arithmetic; compilers and archs**
- **Beware of architecture-specific optimizations, eg. AMD**
- **Beware of cache-related performance tuning (BLAS!!!)**
- **Beware of current and future network evolution (IPv6!)**
- **Be aware of highly relevant research literature (LSSC...)**
- **FTS/FPS are at your disposal for Data Management...**

- A ***whole*** 2-day training activity has been video-taped and put on-line, for your pleasure to enjoy at your own pace, on any single-digit early morning hours:
 - <http://indico.cern.ch/conferenceDisplay.py?confId=a063525>
(Requirements to use it: FireFox & RealPlayer)
 - <http://vod.grnet.gr>
-> Αποθηκευμένα video -> Αναζήτηση - keyword: grid
- There are some important documents to read, in order to learn shell and language (python) scripting:
 - **Unix Shell Scripting** Tutorial, Ashley J.S Mills, 2002
 - **Advanced Bash-Scripting** Guide; An in-depth exploration of the art of shell scripting, Mendel Cooper, 2004
 - **An Introduction to Python**, David M. Beazley, 2000
 - **Advanced Python** Programming, David M. Beazley, 2000

HellasGrid A4 handout

- Just a tiny double page guide, but with surprisingly important info
- Infrastructure highlights
- Useful email addresses

- Available at:
<http://indico.cern.ch/conferenceDisplay.py?confId=21243>



The HellasGrid infrastructure of GRNET is comprised of 6 computer clusters. They are located in Athens (3), Thessaloniki (1), Patras (1) and Crete (1). They provide more than 700 64-bit CPUs and 100 TBs of on-line storage (disks and tapes) interconnected over an end-to-end Gigabit backbone.

The HellasGrid infrastructure is fully integrated within the pan-European Grid infrastructure EGEE <http://www.eu-egee.org> offering more than 30,000 CPUs and 10 PetaBytes of Storage. This infrastructure is available free of charge for the Research and Academic community, for their day-to-day application needs, projects, etc.

More information on the infrastructure can be found at <http://www.hellasgrid.gr/infrastructure>

Access to the infrastructure

Access to the infrastructure is available for the users and the operators with a certain procedure and Acceptable Usage Policy. Instructions for all interested parties are given at <http://www.egee-see.org/>

More information on how to access the HellasGrid infrastructure according to the level of users' expertise (experimental, production etc.) can be found at: <http://www.egee-see.org/users/>

- The pages above will guide you through the process of using the HellasGrid. The instructions cover three different types of potential users:
1. **New experimental users:** users interested in gridifying their existing application(s).
 2. **New production users:** those ready to run their application(s) in the production mode.
 3. **Experienced production users:** (High Energy Physics, Bio-Medical, Computational Chemistry, Finance, Computer Vision, Distributed Rendering, etc.) that want to use the regional Grid

Whatever category of user you belong to, please first register with the HellasGrid infrastructure at www.hellasgrid.gr/user-request

The computing power and storage resources are being shared among all HellasGrid users independently of whether they are located inside or outside the institutes hosting the resources, with common access policies for all institutes and users. The policy is safeguarded by GRNET headquarters' team.

Application Software

Parallel library MPICH G2 is installed on the HellasGrid infrastructure. Additional third party software may be installed with the approval of the HellasGrid Management Boards. A formal request must be sent to: user-request@hellasgrid.gr

The way to install the application software is the temporary installation through a job submission or to request advice by experts through: application-support@hellasgrid.gr

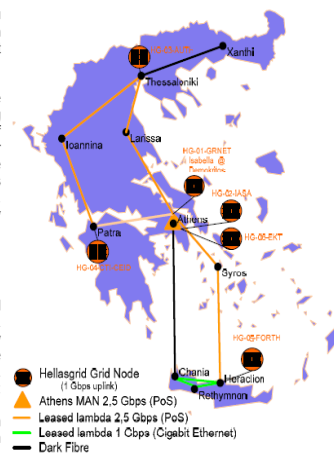


Figure 1 - The GRNET HellasGrid infrastructure



GRNET S.A.
 Greek Research & Technology Network
 56, Mesogion Ave. 11527 Athens
 TEL: +30 2107474274 FAX: +30 2107474490
<http://www.grnet.gr>



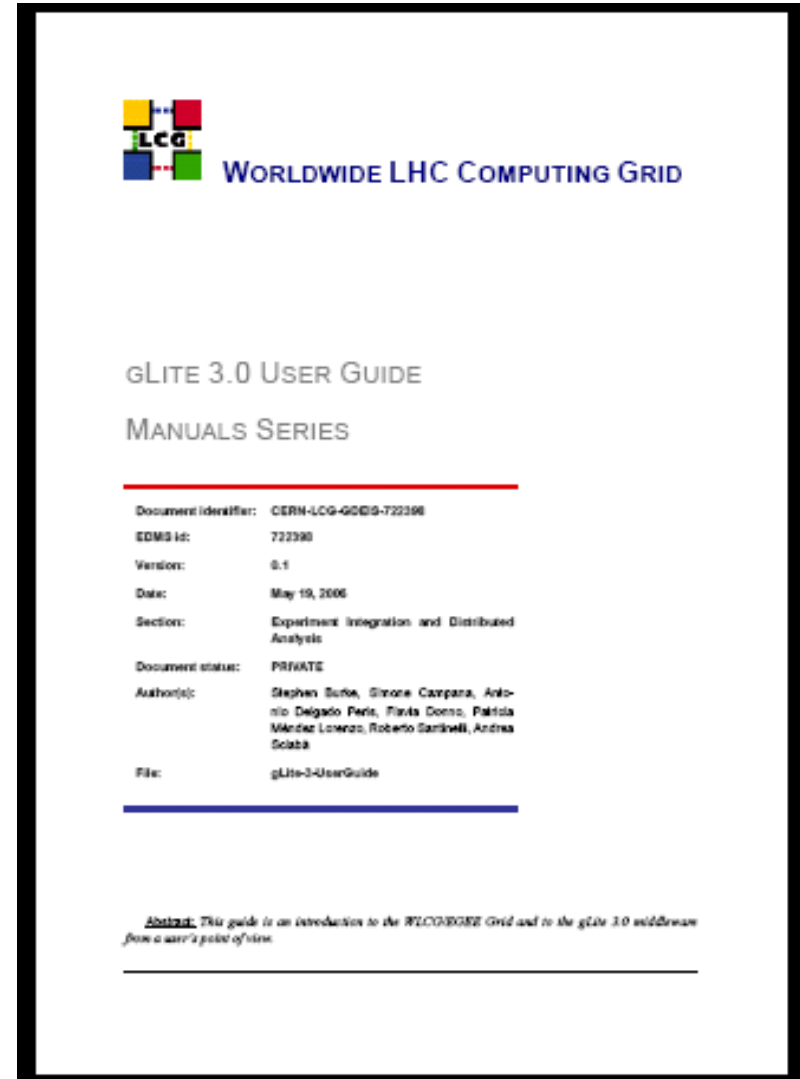
GLITE 3.0 USER GUIDE

- **Glite instruction manual, with multiple middleware commands, 166 pages .pdf:**
 - Getting started
 - Information Service
 - Workload Management & .jdl
 - Data Management
 - User tools

• **RTFM!**

(=Read The Fine Manual!)

<http://glite.web.cern.ch/glite/documentation/>



Medical/Healthcare

- Imaging
- Diagnosis & Treatment
- Drug design (d2ol, for SARS, anthrax, embola etc)

Bioinformatics

- Study of the human and other genomes (genome@home)
- Protein folding (folding@home, predictor@home)

Geological and climate applications

- Weather Forecasting
- Climate Simulation (climate@home)
- Ocean current analysis
- Oil and Gas Exploration
- Seismic Signal Analysis

Pharmaceutical, Chemical, Biotechnology

- atmospheric chemistry
- systems biology
- materials science
 - material interaction simulations
 - catalysis investigations)
- molecular modeling
- nanotechnology

Mathematics and Basic Research

- prime numbers (gimps/mprimes effort)
- The (partial) verification of [Riemann's Hypothesis](#)

Business decision support

- Financial analysis
- Portfolio optimization
- Risk management applications
- Route Optimization
 - Transportation
 - LAN and WAN Networking
- Supply Chain and Demand Chain Optimization
- Search and Retrieval (huge databases, data mining)

Electrical, Mechanical and Civil Engineering

- Energy production and distribution strategy optimization
- Engineering and digital design
- CAD / CAM
- Construction verification against earthquakes
 - eg. finite elements method
- Aerodynamic simulation (wind tunnel simulation)
- Digital Rendering (raytracing, digital video synthesis)

Physics & Astrophysics

- High Energy Physics simulations and signal analysis
- N-body problem simulation
- space probe signal analysis (einstein@home)
- radio telescope signal analysis (seti@home)

Computer Science

- Cryptography (distributed.net)
- Search Engines (grud, a distributed Internet crawler)

Many! <http://distributedcomputing.info/distrib-2003/distrib-projects.html>



Thank You
gef@grnet.gr