AMGA Metadata Catalog

Distributed AMGA Metadata Service

KISTI / NISN (National Institute of Supercomputing and Networking)

Soonwook Hwang

EGI Webinar, 17 July 2013

Contents

- Background and Motivation
- Interface, Architecture and Implementation
- Metadata Replication/Distribution on AMGA
- Use Cases in Scientific Applications
- Ongoing Works and Plans

Background and Motivations

Why Metadata on the Grid?

- The Worldwide LHC Computing Grid (WLCG) is the world largest production grid infrastructure ever built
  - > 150,000 CPUs
  - > 150 computing centers in ~40 countries
  - ~ 25 PBs of data annually generated by Large Hadron Collider (LHC
    - > millions of files to be stored, distributed, and analyzed
- Catalogs are essential to populate, discover and locate data (e.g., millions of files) among the numerous sites of the Grid
  - File Catalog
    - Maps logical filenames to the physical locations of one or more replicas of a file
    - LFC is the most popular file catalog service on the Grid
  - Metadata Catalog
    - Describe the contents of files (e.g., who to create, when to create, etc)
    - Help to search for files based on their description
    - AMGA is the most popular and widely used grid-enabled metadata service

Motivation and History (1/2)

- AMGA provides:
  - Access to metadata for files stored on the Grid
  - A simplified DB access on the Grid
- 2004 – the ARDA project evaluated existing Metadata Services from HEP experiments
  - AMI (ATLAS), RefDB (CMS), Alien Metadata Catalogue (ALICE)
  - Similar goals, similar concepts
  - Each designed for a particular application domain
    - Reuse outside intended domain difficult
  - Several technical limitations: large answers, scalability, speed, lack of flexibility
- ARDA proposed an interface for Metadata access on the GRID
  - Based on requirements of LHC experiments
  - But generic - not bound to a particular application domain
  - Designed jointly with the gLite/EGEE team
  - Incorporates feedback from GridPP

※ ARDA Project (A Realisation of Distributed Analysis for LHC)

Motivation and History (2/2)

- Began as prototype to evaluate the Metadata Interface
  - Evaluated by community since the beginning:
  - Matured quickly thanks to users feedback

- Requirements from HEP community
  - Millions of files, 6000+ users, 200+ computing centres
  - Mainly file metadata
  - **Main concerns : scalability, performance, fault-tolerance, Support for Hierarchical Collection**

- Requirements from Biomed community
  - Smaller scale than HEP
  - **Main concerns : Security**

- Now as part of the EMI product, AMGA is available for download and installation with the latest AMGA 2.4.0 release from the EMI repository

Metadata Catalogue

-Basic Concept

Metadata User Requirement on the Grid

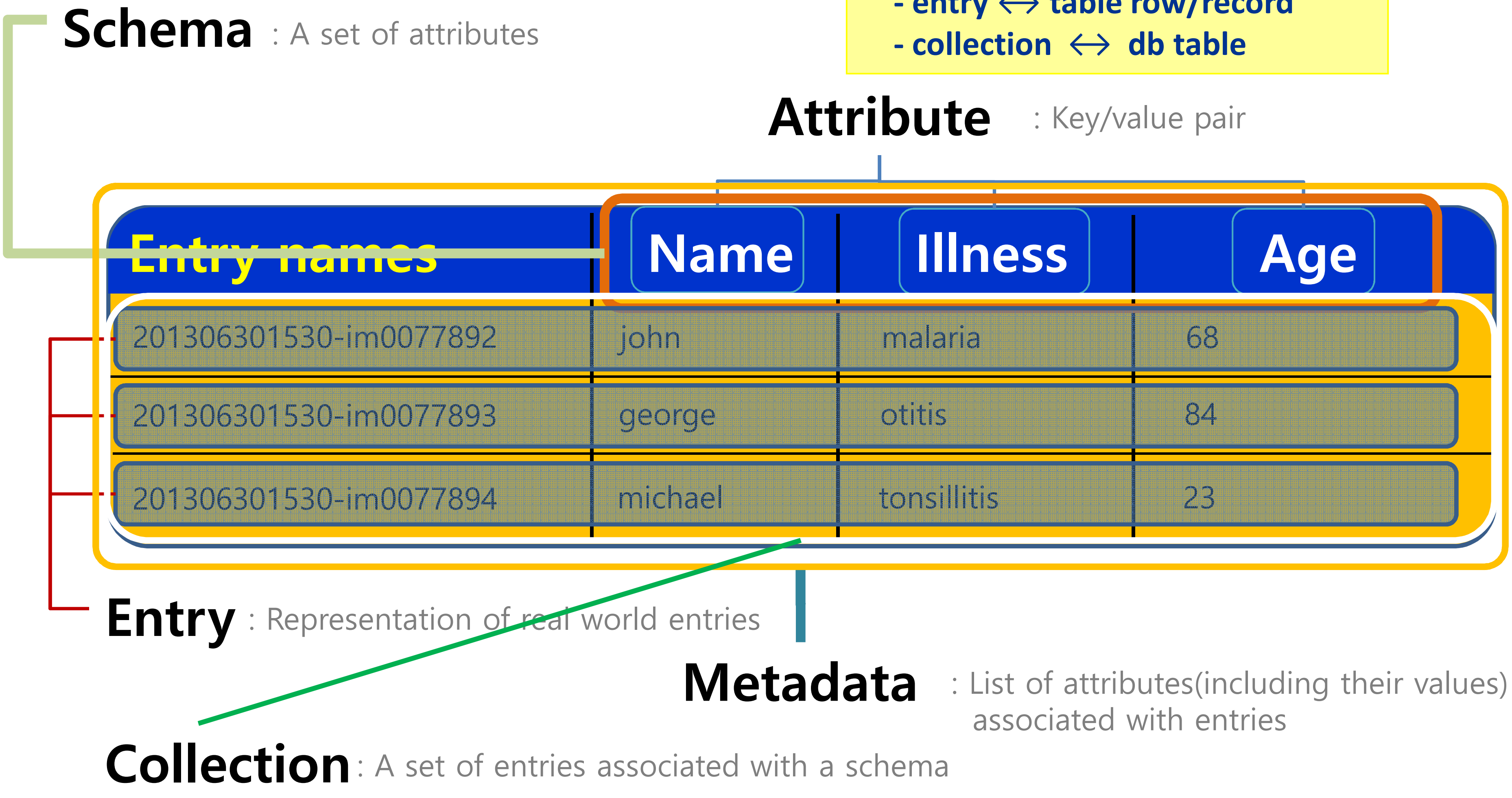
- I want to
  - store some information about files
    - in a structured way
  - query a system about those information
  - Have simplified DB for keeping information about jobs running on the Grid.
    - my job to have direct access to the metadata service using my proxy certificate
    - to have read/write access to job status information
  - Direct use of database system is no choice on the Grid
    - Traditional DB is not considered Grid-enabled in terms of its no support for grid-aware authentication and authorization (e.g., VOMS certificates)

Metadata Concept in AMGA

- Entries
  - Representation of real world entities, which we are attaching metadata to describe them
- Attributes : key/value pairs /w type information
  - Type : Int, float, string
  - Name/Key : the name of the attribute
  - Value : Value of an entry's attributes
- Schema
  - **a set of attributes**
- Collections
  - **A set of entries associated with schema**
- Query
  - **SELECT … WHERE … clause in SQL-like or SQL query language**

AMGA analogy to RDBMS

- attribute ↔ schema column
- schema ↔ table schema
- entry ↔ table row/record
- collection ↔ db table



Schema : A set of attributes

Attribute : Key/value pair

Entry names	Name	Illness	Age
201306301530-im0077892	john	malaria	68
201306301530-im0077893	george	otitis	84
201306301530-im0077894	michael	tonsillitis	23

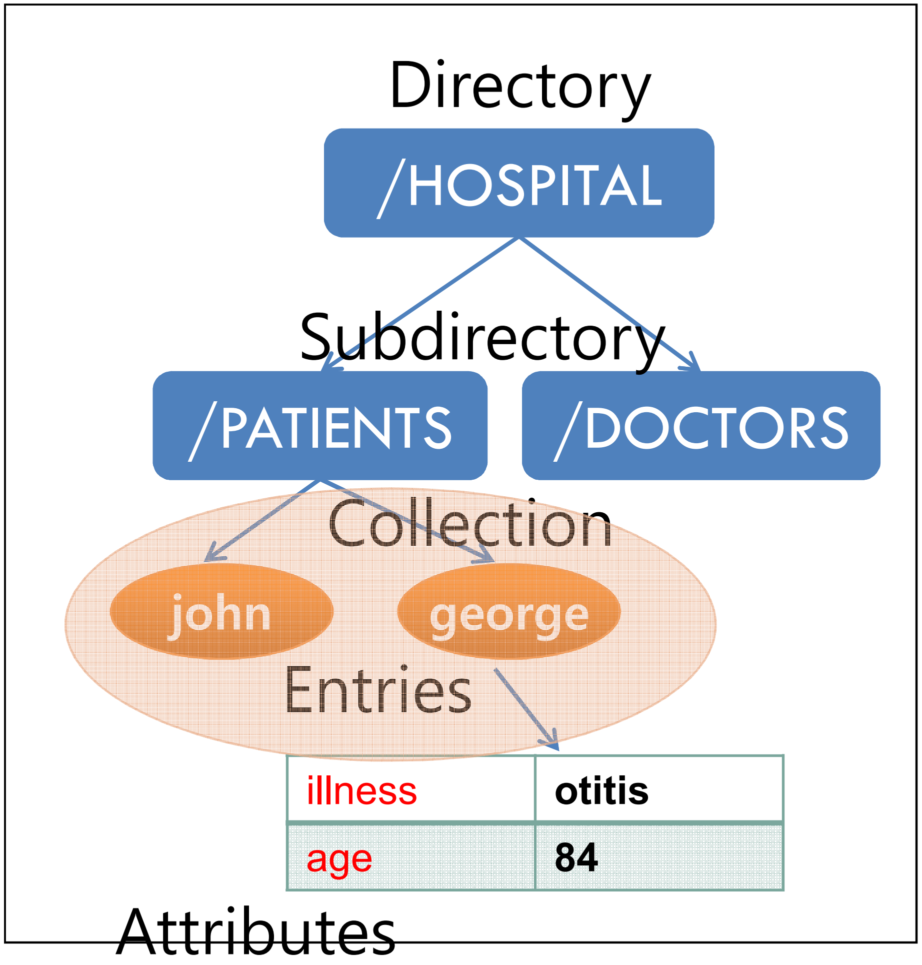
Entry : Representation of real world entries

Metadata : List of attributes(including their values) associated with entries

Collection : A set of entries associated with a schema

AMGA Analogy to file system

- Collection ↔ Directory
- Entry ↔ File
- Attribute ↔ File Attribute



illness	otitis
age	84

AMGA Metadata Organization

- Relational schema

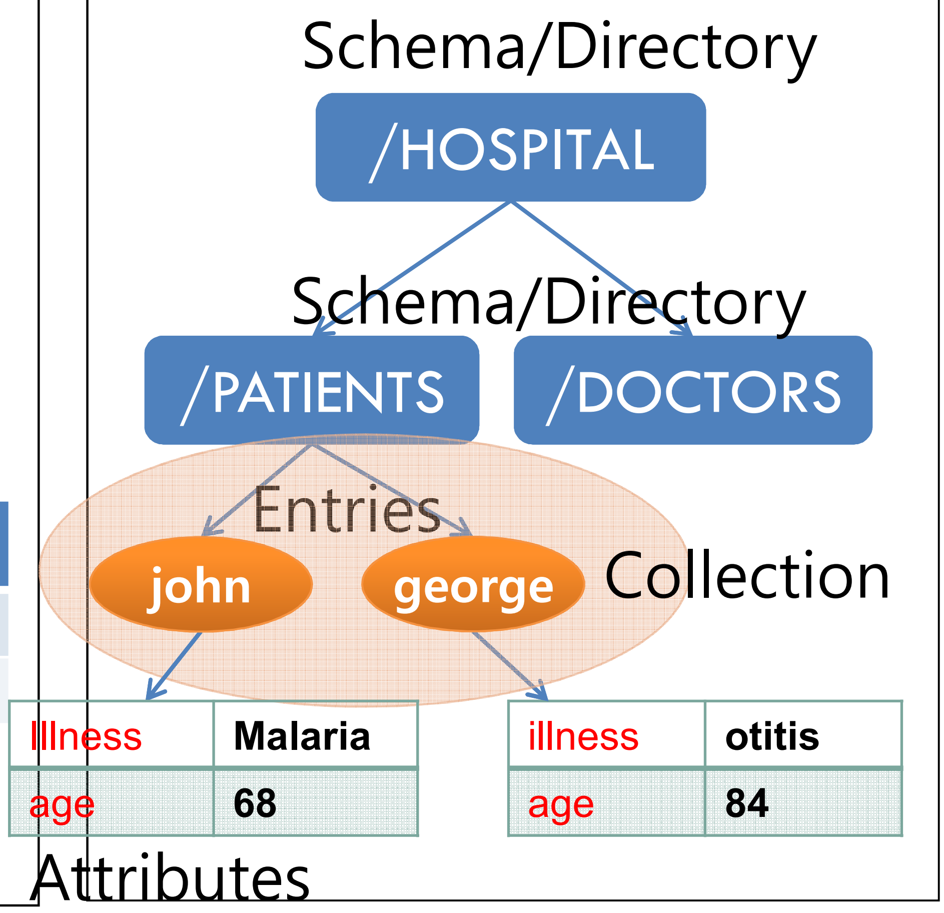
TABLE: HOSPITAL

#name	#type
PATIENTS	**people_group**
DOCTORS	**people_group**

TABLE: PATIENTS

#name	sickness	age
john	**malaria**	**68**
george	**otitis**	**84**

- AMGA(hierarchy)

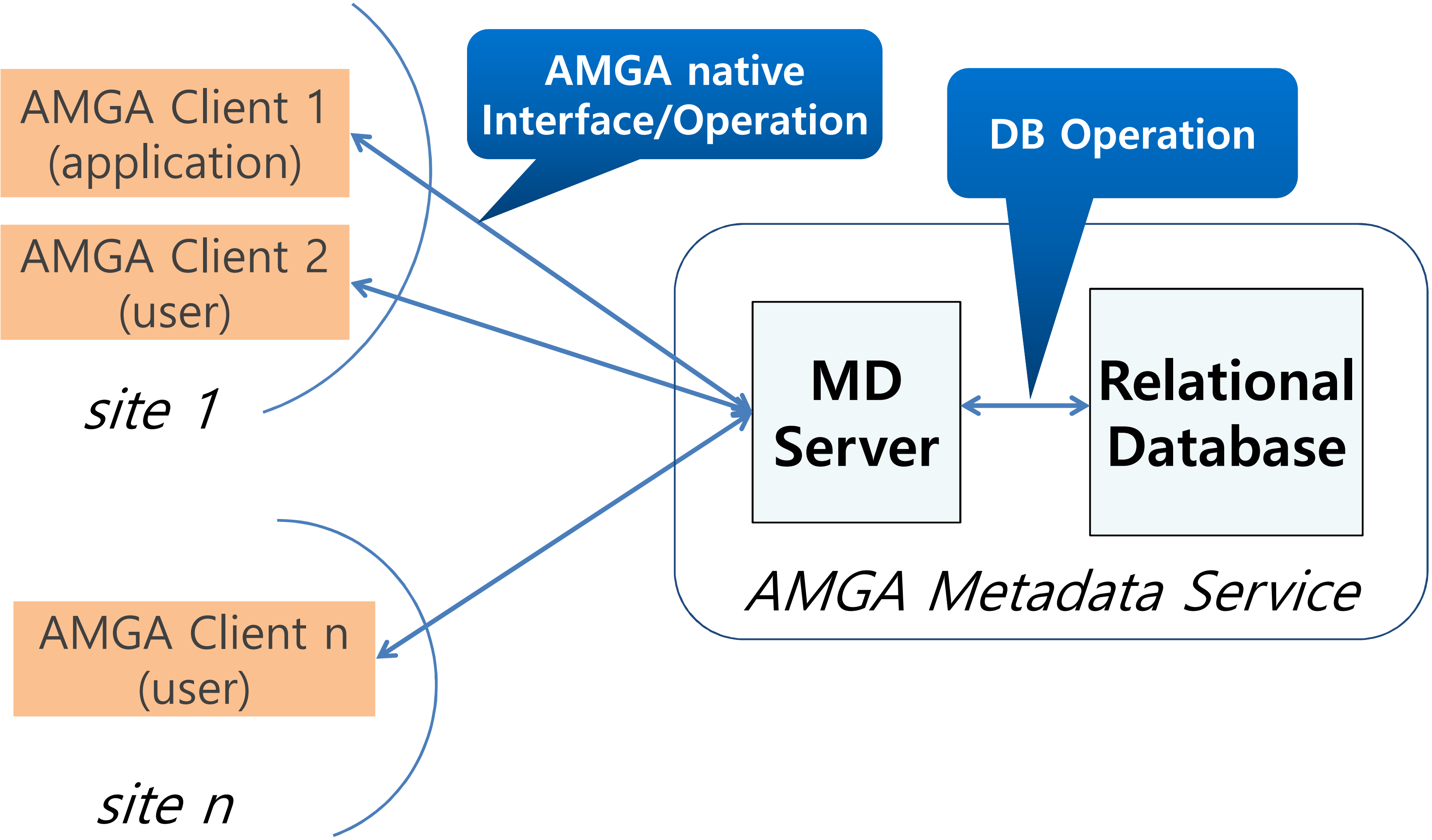


Illness	**Malaria**
age	**68**

illness	**otitis**
age	**84**

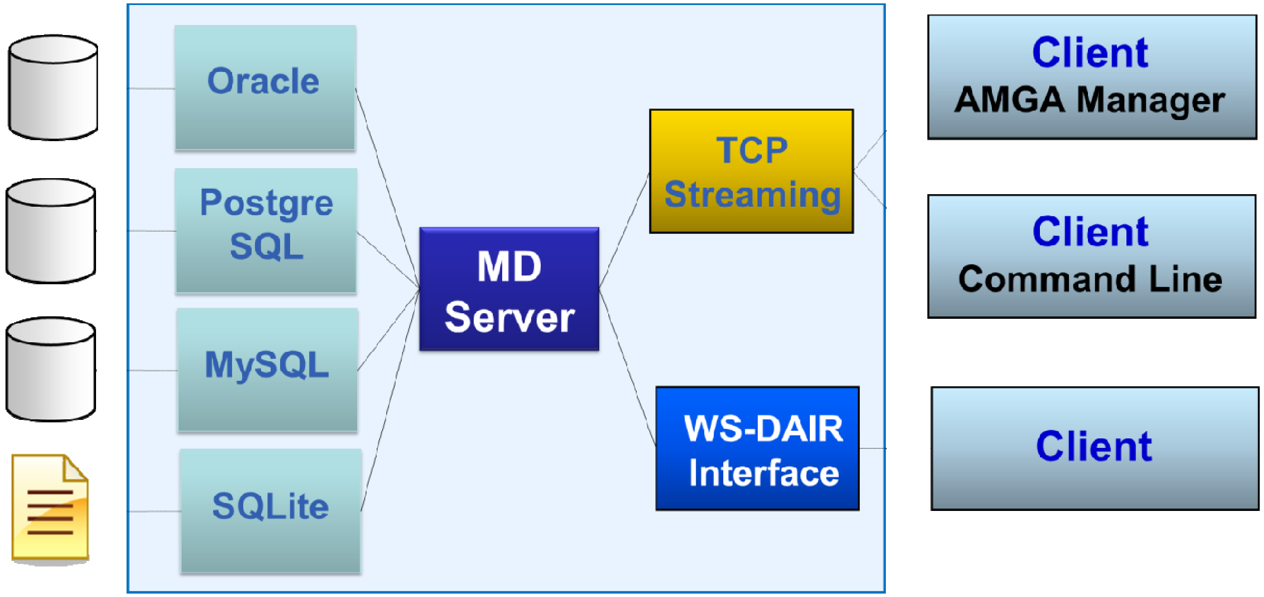
AMGA Features

AMGA Implementation(1/2)



AMGA Implementation(2/2)

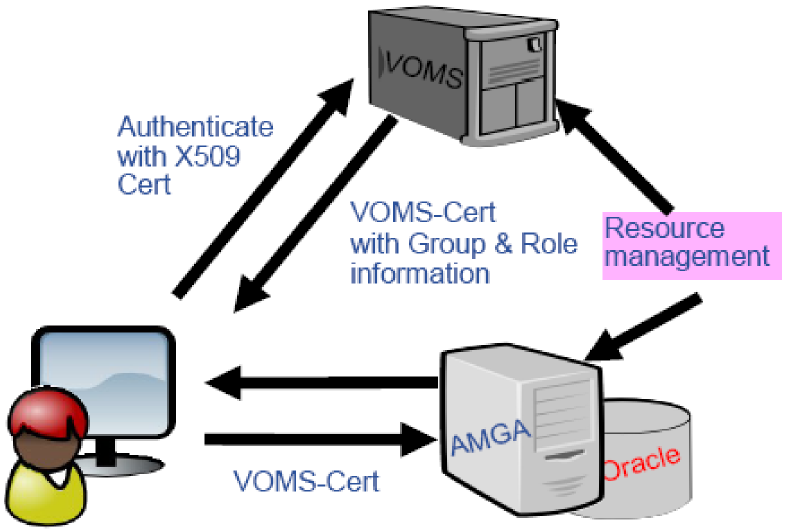
- Modular back-end : Oracle, PostgreSQL, MySQL, SQLite
- Modular front-end : TCP Streaming, WS-DAIR (SOAP)



- Streamed Bulk Operations
- Import existing databases
- Language : Native SQL Query & AMGA Language Query
- Platform : SLC3/4/5/6, Fedora Core, Gentoo

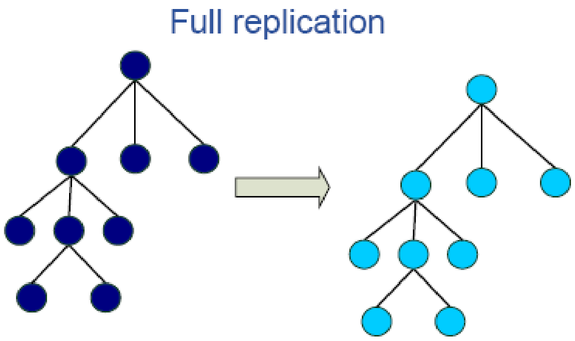
AMGA Security

- Client Authentication based on
  - Username/password
  - X509 Grid certificates
  - VOMS certificates
- Secure connections - SSL
- Access Control is supported
  - Unix style permission
    - User-group-others (e.g., rwxr--r--)
  - Fine-grained ACLs
    - per-collection (default)
    - per-entry

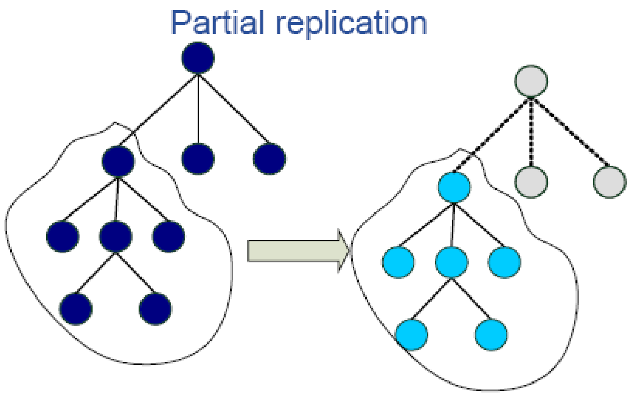

VOMS
Authenticate with X509 Cert
VOMS-Cert with Group & Role information
Resource management
AMGA
Oracle
VOMS-Cert

Metadata Replication on AMGA

- Why Replication of metadata catalog?
  - With hundreds of geographically distributed sites accessing a metadata catalog service, a centralized catalog service doesn't provide the required scalability, performance or fault-tolerance.
- In HEP applications, write rates are an order of magnitude lower than read rates
  - Write operations carried out on a central catalog
  - Read operations offloaded to read-only replicas that are closer to the applications in order to avoid network latency
- Partial replication support is necessary



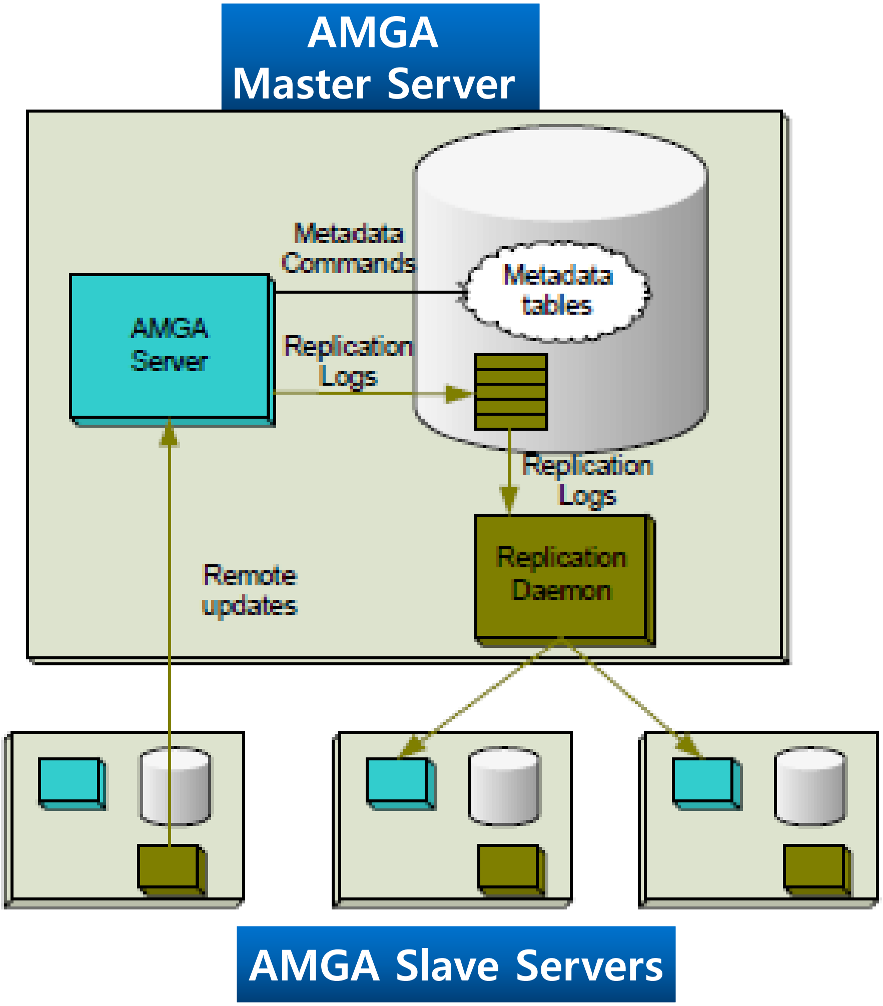
Full replication



Partial replication

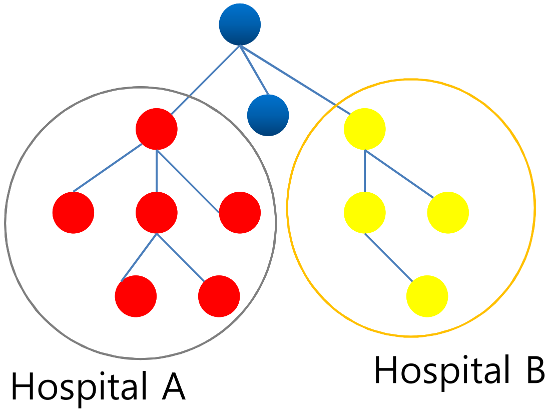
AMGA Replication Implementation

- 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



Metadata Federation on AMGA

- Why federation/distribution of metadata catalog?
  - The idea of federation of metadata comes from the requirement of the biomedical community, as their metadata often contains confidential information about patients
  - The metadata is often created in different geographical locations (hospital or laboratories)
  - Replicating the sensitive metadata either to a central catalog or to other replicas would increase the exposure to attacks
- AMGA approach is the federation/distribution of individual catalogs into a single virtual catalog
  - Allowing metadata to remain secure at its origin site

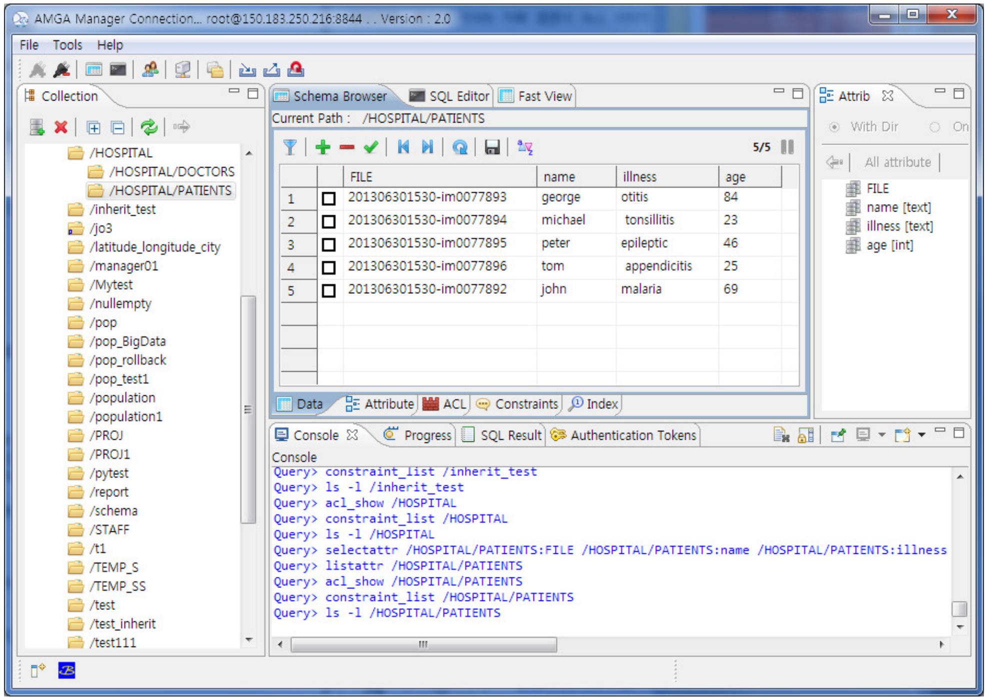


AMGA Clients

- CLI – Shell-like interactive client (MD client)

```
createdir /HOSPITAL
createdir /HOSPITAL/PATIENTS
addattr /HOSPITAL/PATIENTS name text illness text age int
addentry /HOSPITAL/PATIENTS/ 201306301530-im0077892 name john illness malaria age 67
update /HOSPITAL/PATIENTS age 69  'FILE="201306301530-im0077892"‘
selectattr /HOSPITAL/PATIENTS:FILE /HOSPITAL/PATIENTS:name /HOSPITAL/PATIENTS:illness
        /HOSPITAL/PATIENTS:age ‘/HOSPITAL/PATIENTS:age>40‘
```

- GUI – AMGA Manager



Current Path :   /HOSPITAL/PATIENTS

		FILE	name	illness	age
1	□	201306301530-im0077893	george	otitis	84
2	□	201306301530-im0077894	michael	tonsillitis	23
3	□	201306301530-im0077895	peter	epileptic	46
4	□	201306301530-im0077896	tom	appendicitis	25
5	□	201306301530-im0077892	john	malaria	69

Console

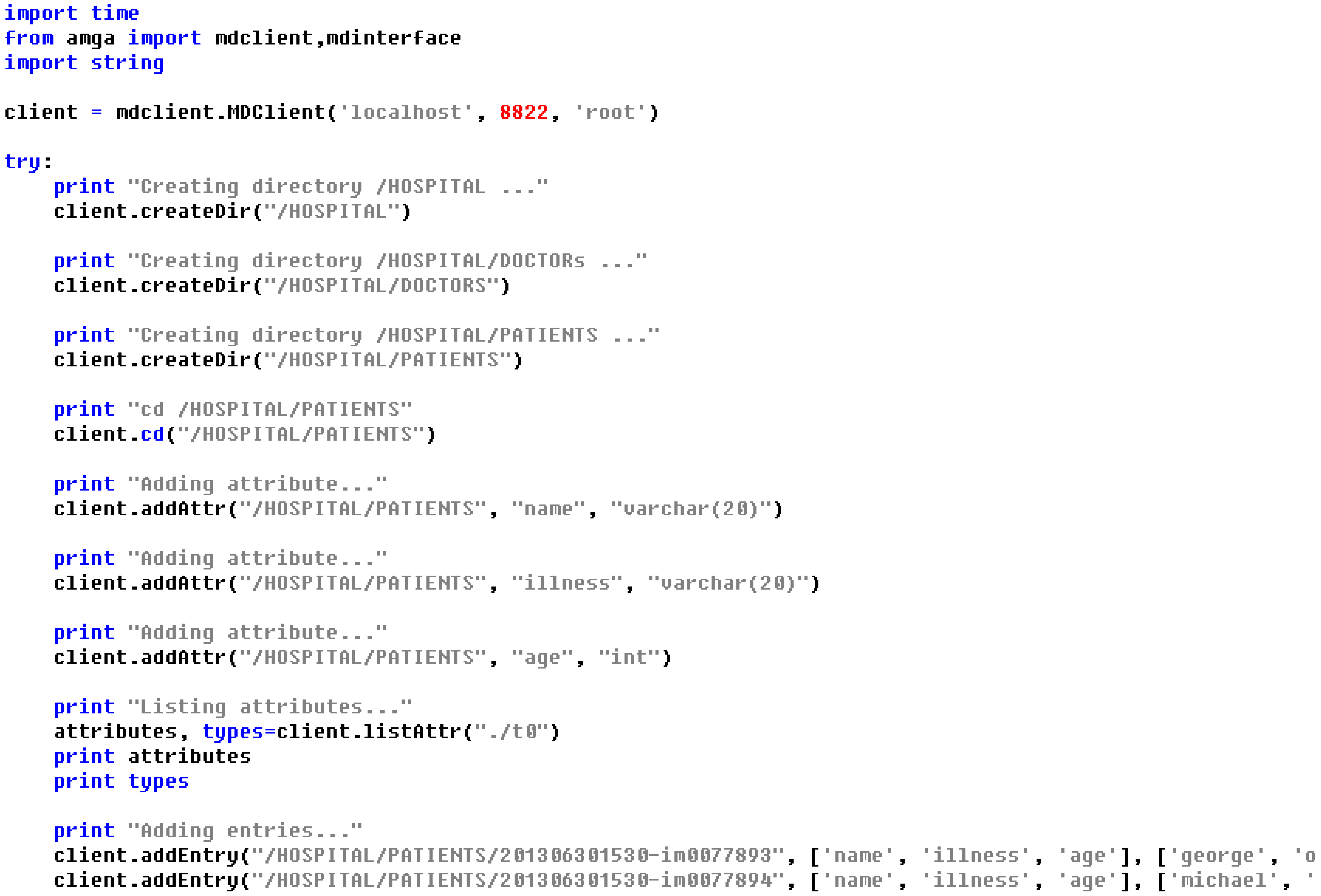
```
Query> constraint_list /inherit_test
Query> ls -l /inherit_test
Query> acl_show /HOSPITAL
Query> constraint_list /HOSPITAL
Query> ls -l /HOSPITAL
Query> selectattr /HOSPITAL/PATIENTS:FILE /HOSPITAL/PATIENTS:name /HOSPITAL/PATIENTS:illness
Query> listattr /HOSPITAL/PATIENTS
Query> acl_show /HOSPITAL/PATIENTS
Query> constraint_list /HOSPITAL/PATIENTS
Query> ls -l /HOSPITAL/PATIENTS
```

- Many Programming APIs
  - C/C++
  - Python
  - Java
  - Perl
  - PHP

MD Client – AMGA CLI Client

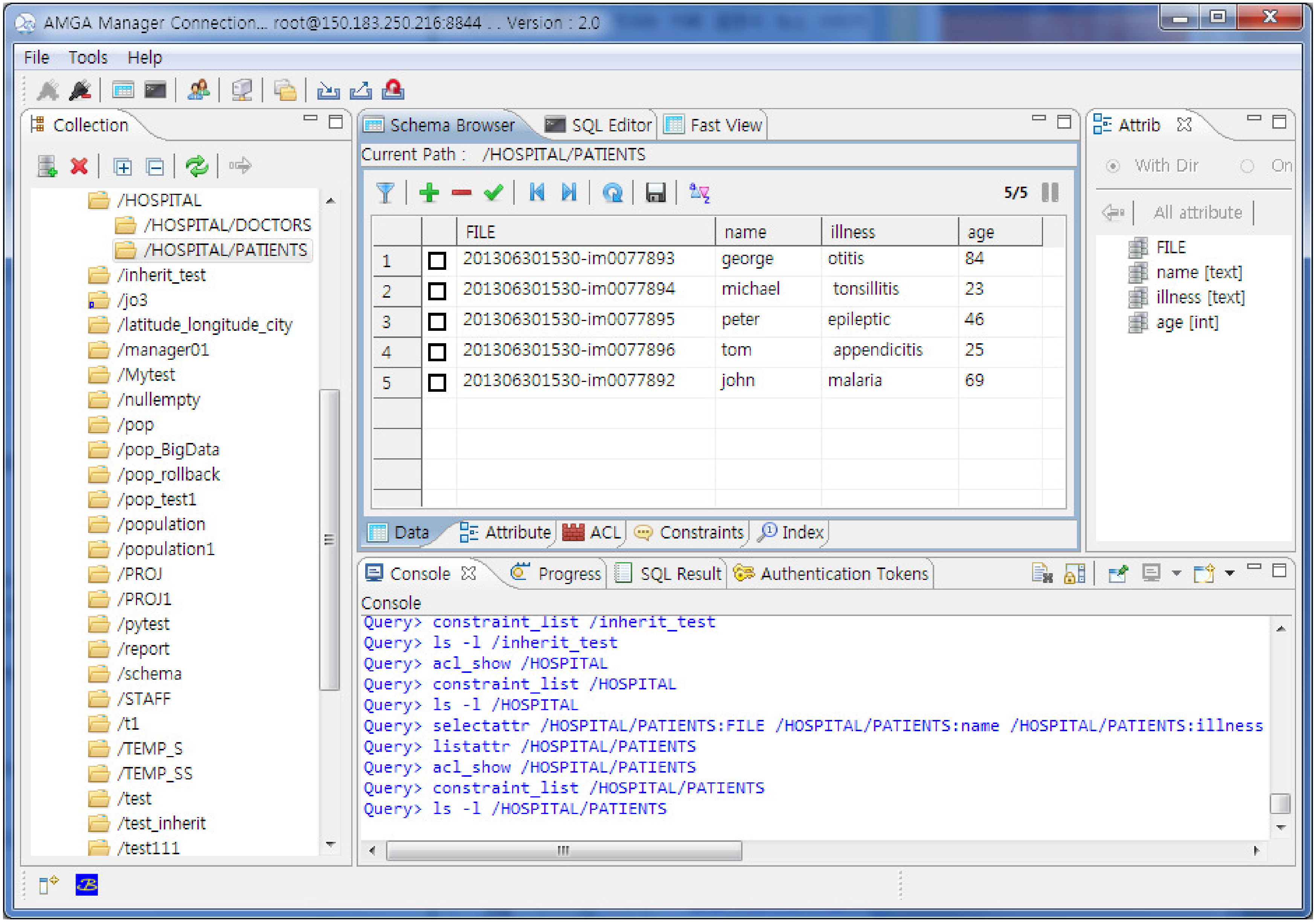
```
% createdir /HOSPITAL

% createdir /HOSPITAL/PATIENTS

% addattr /HOSPITAL/PATIENTS name text illness text age int

% addentry /HOSPITAL/PATIENTS/201306301530-im0077892

              name  john illness malaria age 67

% update /HOSPITAL/PATIENTS age 69  'FILE="201306301530-im0077892"'‘

% selectattr /HOSPITAL/PATIENTS:FILE /HOSPITAL/PATIENTS:name

             /HOSPITAL/PATIENTS:illness  /HOSPITAL/PATIENTS:age

             ‘/HOSPITAL/PATIENTS:age>40‘
```

AMGA Python API Example



```
import time
from amga import mdclient,mdinterface
import string

client = mdclient.MDClient('localhost', 8822, 'root')

try:
    print "Creating directory /HOSPITAL ..."
    client.createDir("/HOSPITAL")

    print "Creating directory /HOSPITAL/DOCTORs ..."
    client.createDir("/HOSPITAL/DOCTORS")

    print "Creating directory /HOSPITAL/PATIENTS ..."
    client.createDir("/HOSPITAL/PATIENTS")

    print "cd /HOSPITAL/PATIENTS"
    client.cd("/HOSPITAL/PATIENTS")

    print "Adding attribute..."
    client.addAttr("/HOSPITAL/PATIENTS", "name", "varchar(20)")

    print "Adding attribute..."
    client.addAttr("/HOSPITAL/PATIENTS", "illness", "varchar(20)")

    print "Adding attribute..."
    client.addAttr("/HOSPITAL/PATIENTS", "age", "int")

    print "Listing attributes..."
    attributes, types=client.listAttr("./t0")
    print attributes
    print types

    print "Adding entries..."
    client.addEntry("/HOSPITAL/PATIENTS/201306301530-im0077893", ['name', 'illness', 'age'], ['george', 'o
    client.addEntry("/HOSPITAL/PATIENTS/201306301530-im0077894", ['name', 'illness', 'age'], ['michael', '
```

AMGA Manager – AMGA GUI Client



/HOSPITAL
/HOSPITAL/DOCTORS
/HOSPITAL/PATIENTS
/inherit_test
/jo3
/latitude_longitude_city
/manager01
/Mytest
/nullempty
/pop
/pop_BigData
/pop_rollback
/pop_test1
/population
/population1
/PROJ
/PROJ1
/pytest
/report
/schema
/STAFF
/t1
/TEMP_S
/TEMP_SS
/test
/test_inherit
/test111

Current Path : /HOSPITAL/PATIENTS

		FILE	name	illness	age
1	□	201306301530-im0077893	george	otitis	84
2	□	201306301530-im0077894	michael	tonsillitis	23
3	□	201306301530-im0077895	peter	epileptic	46
4	□	201306301530-im0077896	tom	appendicitis	25
5	□	201306301530-im0077892	john	malaria	69

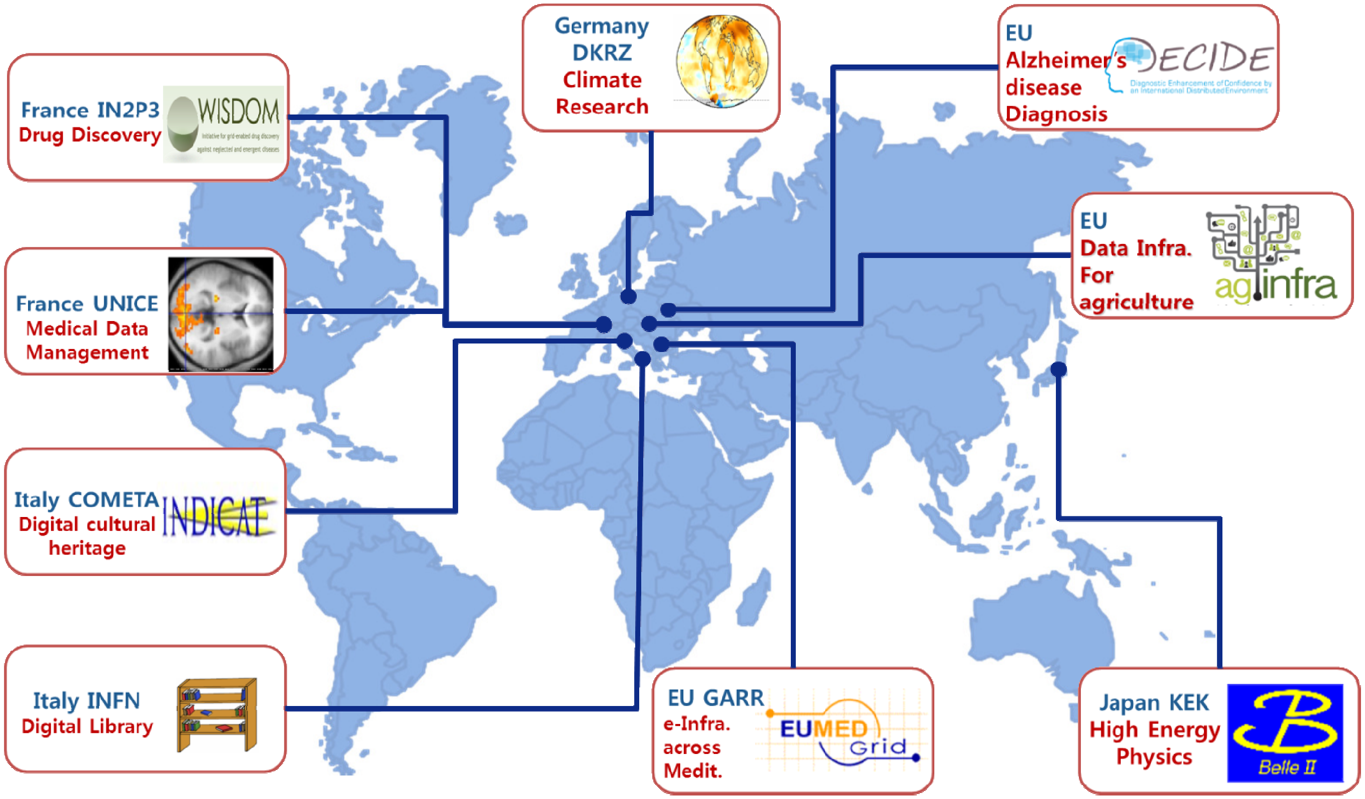
FILE
name [text]
illness [text]
age [int]

Console

```
Query> constraint_list /inherit_test
Query> ls -l /inherit_test
Query> acl_show /HOSPITAL
Query> constraint_list /HOSPITAL
Query> ls -l /HOSPITAL
Query> selectattr /HOSPITAL/PATIENTS:FILE /HOSPITAL/PATIENTS:name /HOSPITAL/PATIENTS:illness
Query> listattr /HOSPITAL/PATIENTS
Query> acl_show /HOSPITAL/PATIENTS
Query> constraint_list /HOSPITAL/PATIENTS
Query> ls -l /HOSPITAL/PATIENTS
```

AMGA Use Cases

Projects and Applications using AMGA



- Others: Health e-Child(EU), GISELA(EU), neuGRID, outGRID, SEEGRID, GAP(TW) etc

Early Adopters of AMGA

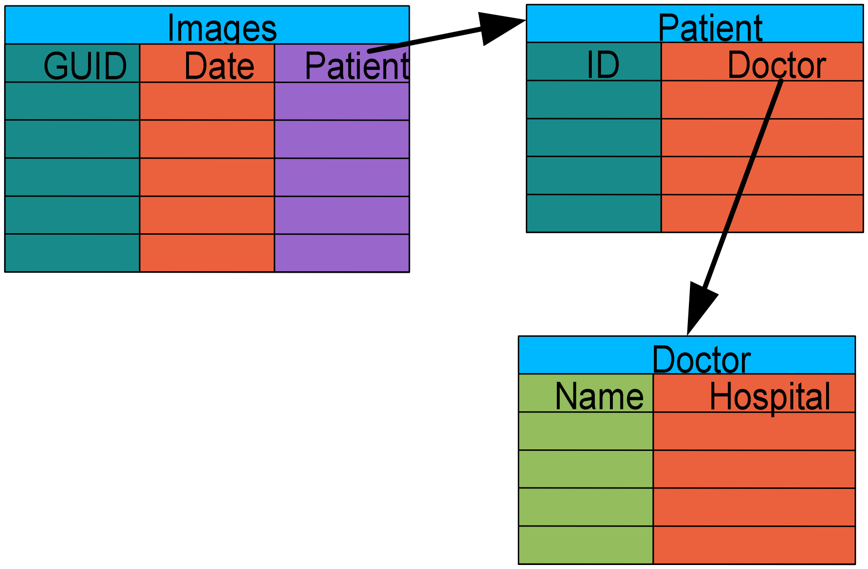
- LHCb-bookkeeping (keep additional information from executed jobs)
  - Migrated bookkeeping metadata to ARDA prototype
    - 20M entries, 15 GB
    - Large amount of static metadata
  - Feedback valuable in improving interface and fixing bugs
  - AMGA showing good scalability

- Ganga
  - Grid Job submission and management system
    - Developed jointly by Atlas and LHCb
  - Uses AMGA for storing information about job status
    - Small amount of highly dynamic metadata

Biomed: Medical Data Manager (MDM)

- Store and access medical images exploiting metadata on the Grid

- 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



Images

GUID	Date	Patient

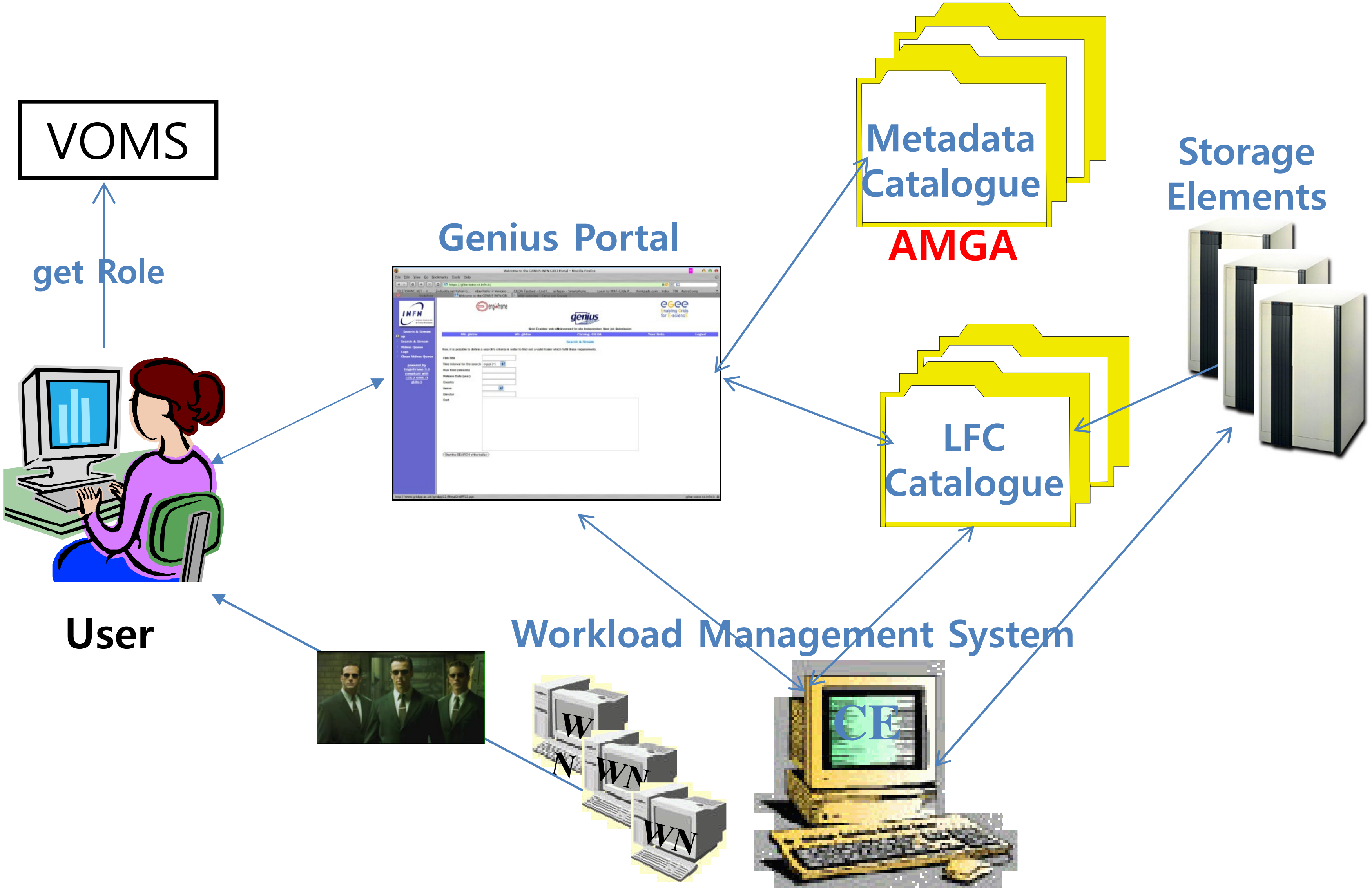
Patient

ID	Doctor

Doctor

Name	Hospital

gMOD: grid Movie On Demand Service



VOMS

get Role

User

Genius Portal

Metadata Catalogue

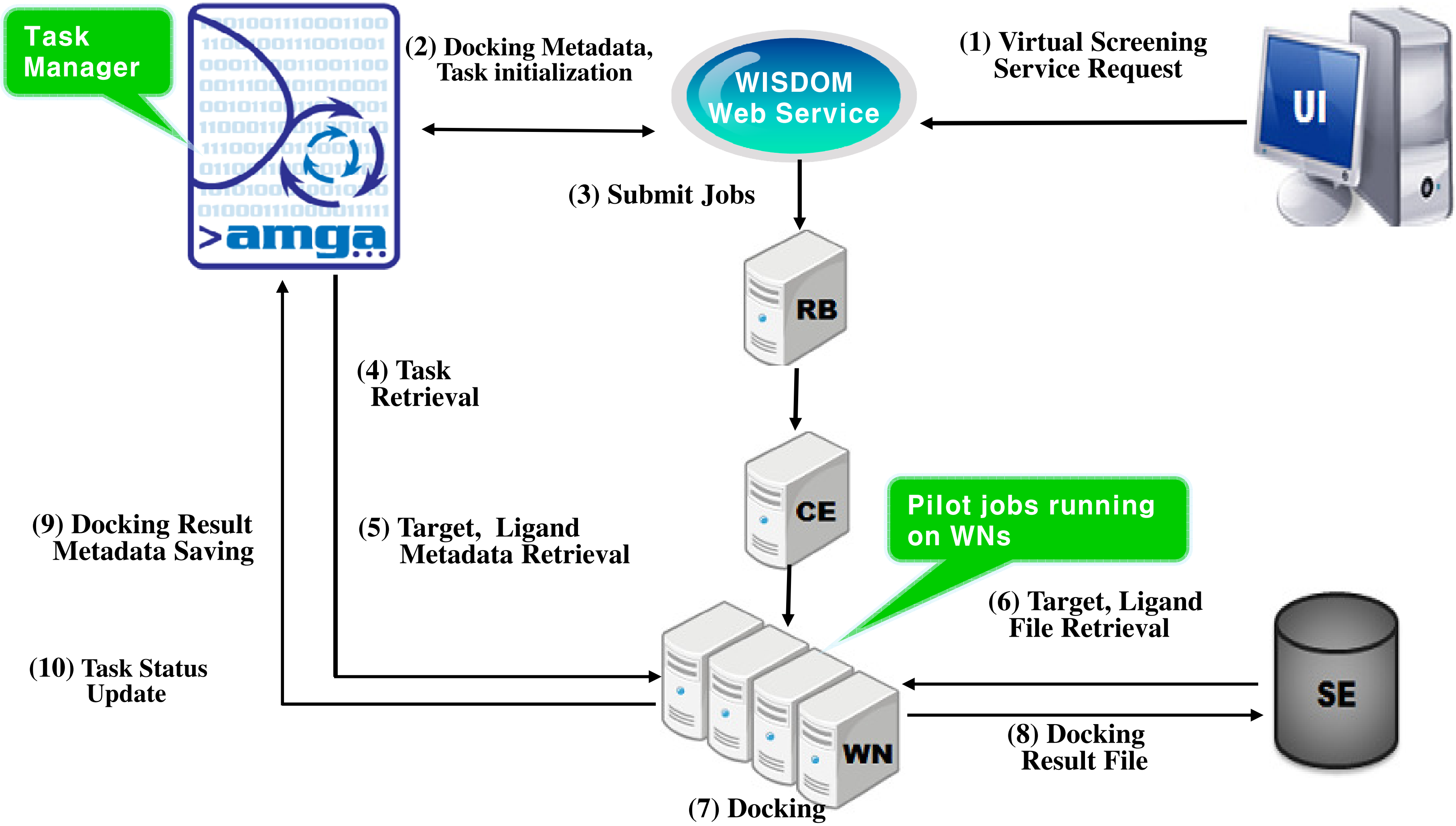
AMGA

LFC Catalogue

Storage Elements

Workload Management System

WISDOM: Worldwide In Silico Docking on Malaria



Task Manager

(1) Virtual Screening Service Request

(2) Docking Metadata, Task initialization

(3) Submit Jobs

(4) Task Retrieval

(5) Target, Ligand Metadata Retrieval

(6) Target, Ligand File Retrieval

(7) Docking

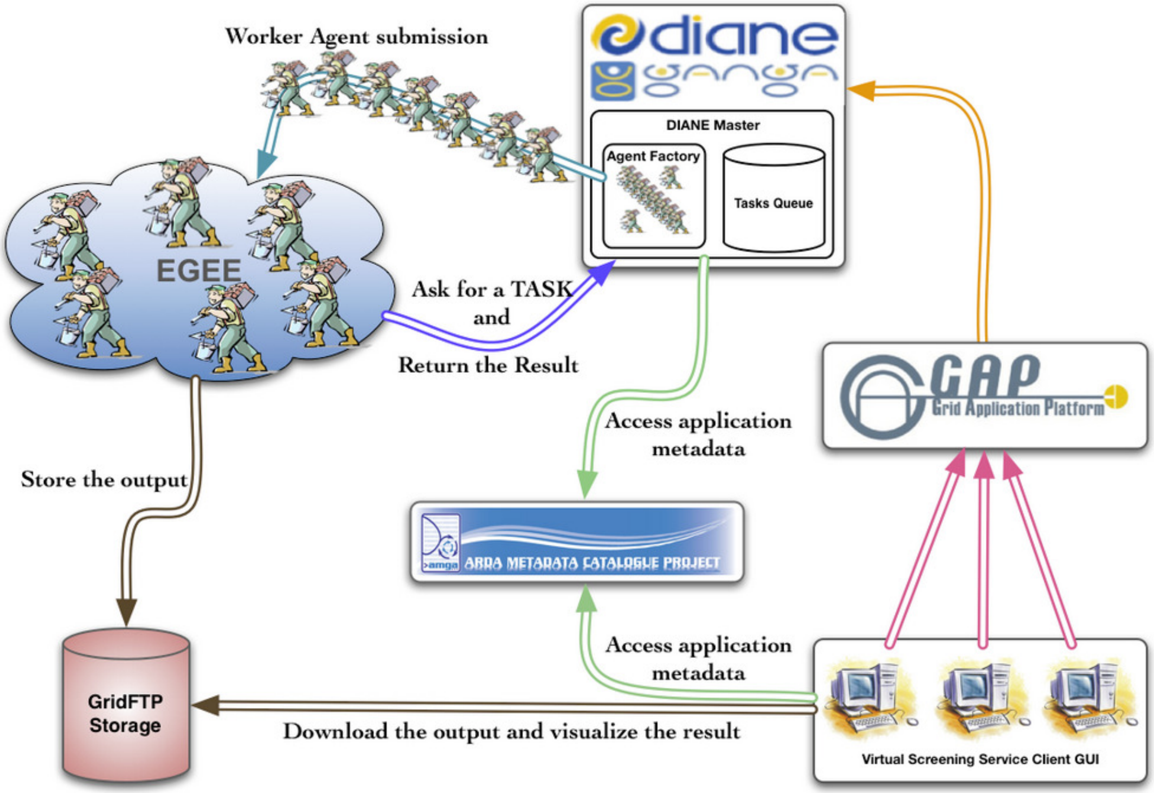
(8) Docking Result File

(9) Docking Result Metadata Saving

(10) Task Status Update

Pilot jobs running on WNs

GAP Virtual Screening Service


Worker Agent submission
diane
ganga
DIANE Master
Agent Factory
Tasks Queue
EGEE
Ask for a TASK
and
Return the Result
Store the output
Access application metadata
GAP
Grid Application Platform
ARDA METADATA CATALOGUE PROJECT
amga
Access application metadata
GridFTP Storage
Download the output and visualize the result
Virtual Screening Service Client GUI

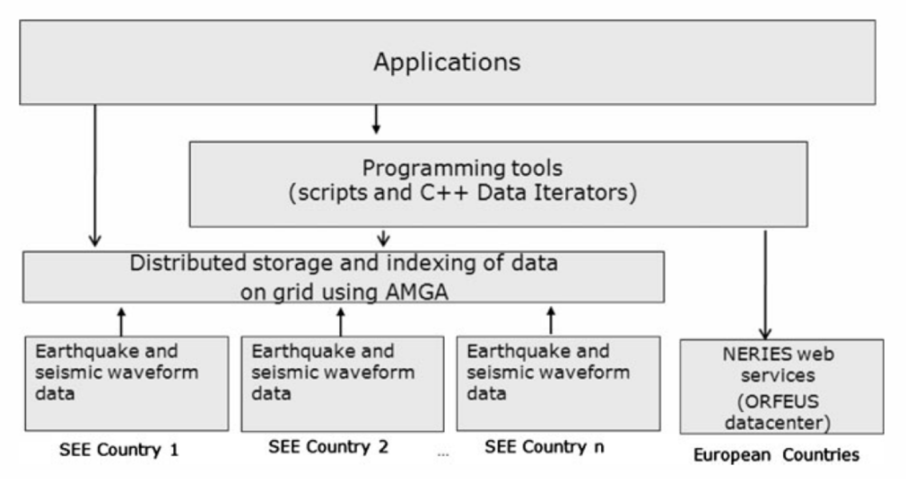
- ASGC has developed the GVSS application package integrated with the gLite software DIANE2 and AMGA, and used the Autodock as the simulation docking engine.

- Virtual Screening Service (VSS) based the Grid Application Platform (GAP) for the Avian Flu DC2 Refine drug discovery in EUAsia VO in March 2009
- a total of 1,111 CPU-days was run over the EUAsiaGrid infrastructure
- more than 160,000 output files with a data volume of 12.8 Gigabytes were created and stored
- now aiming for more scientific collaboration with EUAsiaGrid partners: to seek for solutions for Dengue Fever via using the GVSS (GAP Virtual Screening Service).

http://gap.grid.sinica.edu.tw

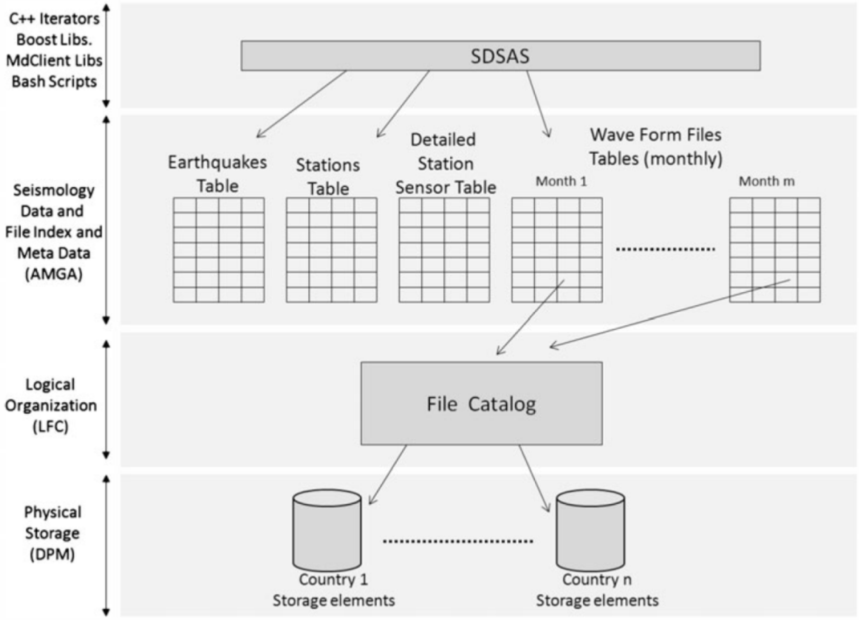
SDSAS (Seismic Data Server Application Service)

Seismic Data Server Application Service (SDSAS) serves official lists of earthquakes, stations and waveform data collected from various South Eastern European (SEE) countries – **SEEGRID Project**



Applications
Programming tools
(scripts and C++ Data Iterators)
Distributed storage and indexing of data
on grid using AMGA
Earthquake and seismic waveform data
SEE Country 1
Earthquake and seismic waveform data
SEE Country 2
...
Earthquake and seismic waveform data
SEE Country n
NERIES web services
(ORFEUS datacenter)
European Countries

Software / data stack for seismology VO



Architecture of the SDSAS

Reference [1]

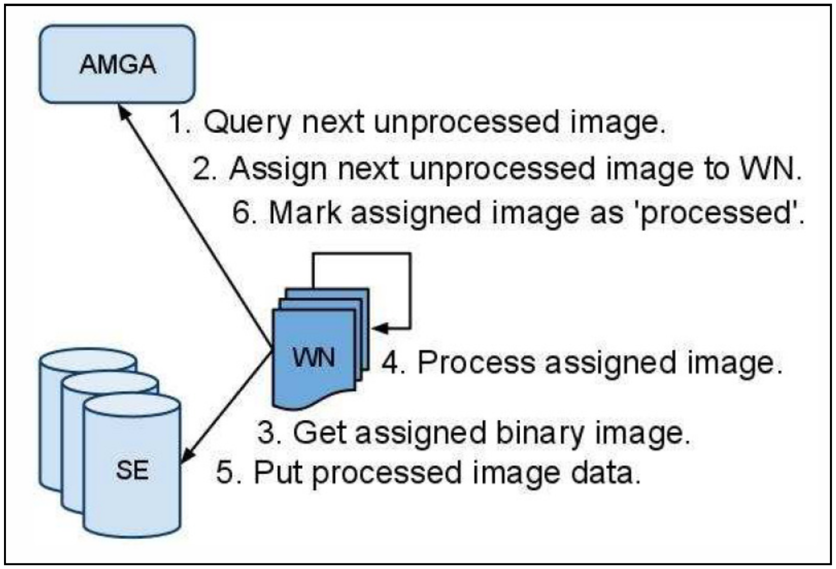
Can Özturan, Bilal Bektaş, Mehmet Yilmazer, "**Seismic data server application service for SEEGRID seismology virtual organization**", Earth Science Informatics, Vol 3, Issue 4, pp 219-228, 2010

Digi-Clima

pilot jobs are

Digi-Clima: an Octave/Matlab application for the semi-automatic processing of historical graphical rain records. - **GISELA Project**

Aims: digitalizing the pluviographic records to preserve the data and to allow an easier access to it.


AMGA
1. Query next unprocessed image.
2. Assign next unprocessed image to WN.
6. Mark assigned image as 'processed'.
WN
4. Process assigned image.
SE
3. Get assigned binary image.
5. Put processed image data.

Pilot job sequence diagram

- Pilot jobs are used.

  ∴ The time to set up the CE workspace to run Digi-clima is not negligible, and a large number of images are to be processed

- AMGA is used for accounting on the images metadata (status and name-results mappling)

Reference [1]

S. GARCÍA, Sebastián; ITURRIAGA, Santiago; NESMACHNOW, Sergio (Universidad de la República, Uruguay).**Scientific computing in the Latin America-Europe GISELA Grid infrastructure**. EProceedings of the High-Performance Computing Latin America Symposium, Cordoba, Argentina, 2011

AMGA Use Case in the Belle II Computing

Belle II Overview

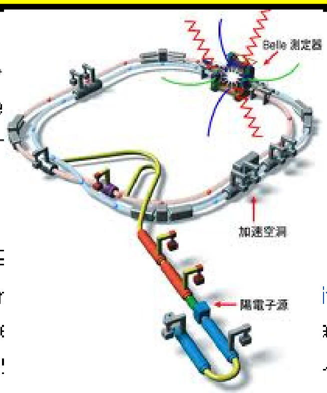
Belle II

- HEP Experiment to confirm of the theory of Kobayashi and Maskawa
- Awarded the 2008 Nobel Prize in Physics

The **Belle experiment** is a particle physics experiment conducted by the **Belle Collaboration**, an international collaboration of more than 400 physicists and engineers investigating CP-violation effects at the High Energy Accelerator Research Organisation (KEK) in Tsukuba, Ibaraki Prefecture, Japan.

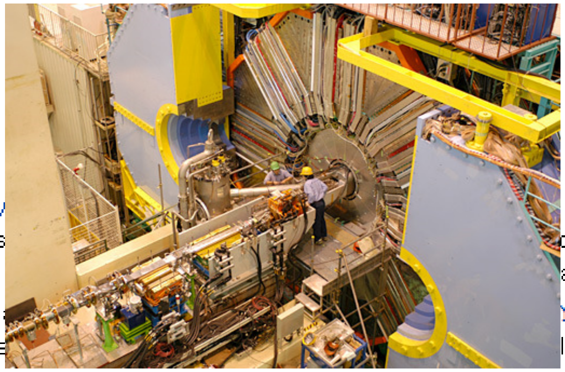
Belle II experiment is expected to produce ~200 petabytes of data in ~10 years starting 2015
- 2GB/s DAQ rate
- 10s of millions of files distributed across multiple grid sites

- observation of: $B \to K^* l^+ l^-$ and $b \to s l^+ l^-$
- measurement of ϕ_3 using the $B \to DK, D \to$



ut special runs at the Υ(5S) reson

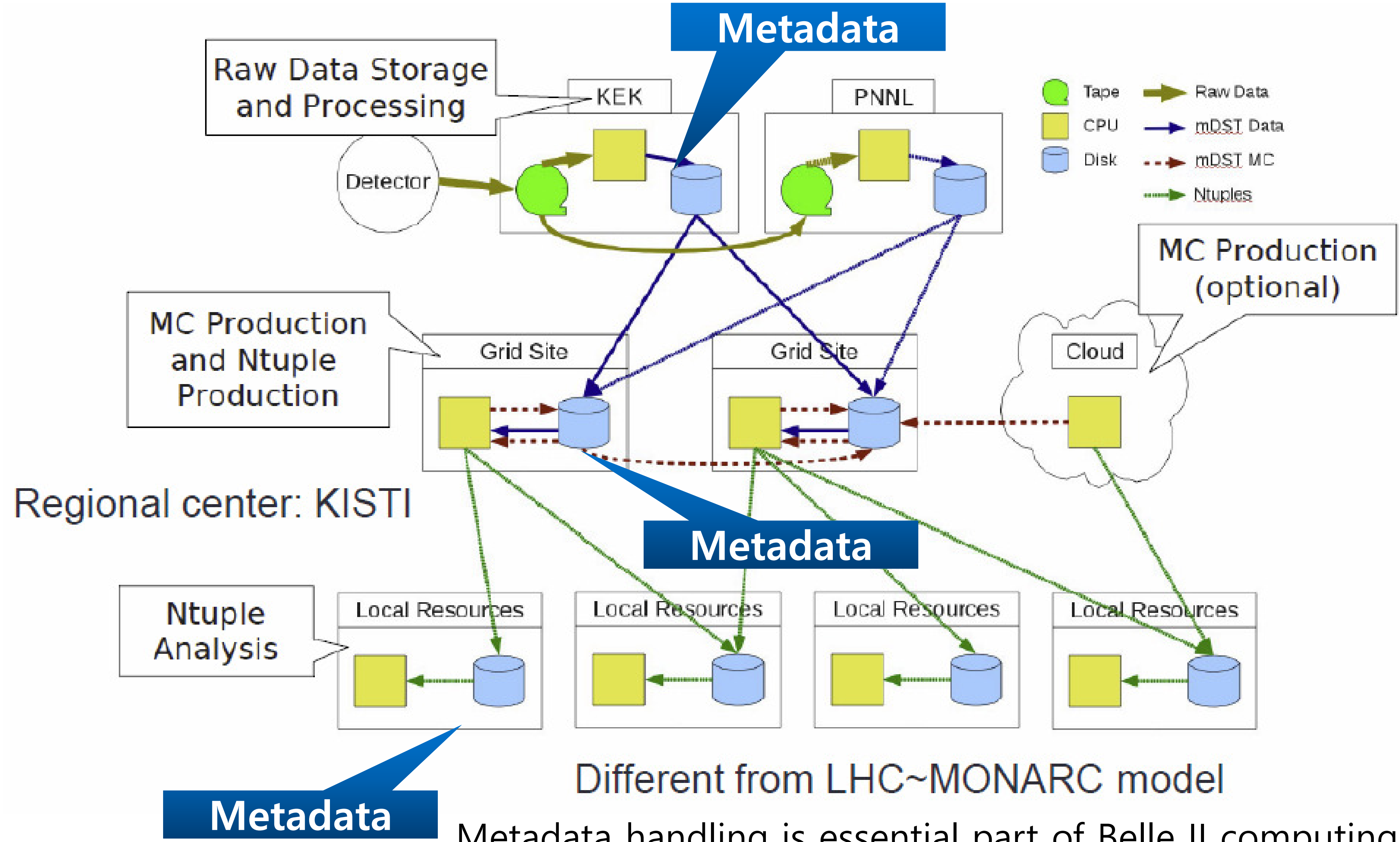
nd the Higgs Boson. Actually the s



The Belle II B-factory, an upgraded facility with two orders of magnitude more luminosity, has been approved in June 2010.[1] The design and construction work is ongoing.

Source: Wikipedia

Belle II Computing Model



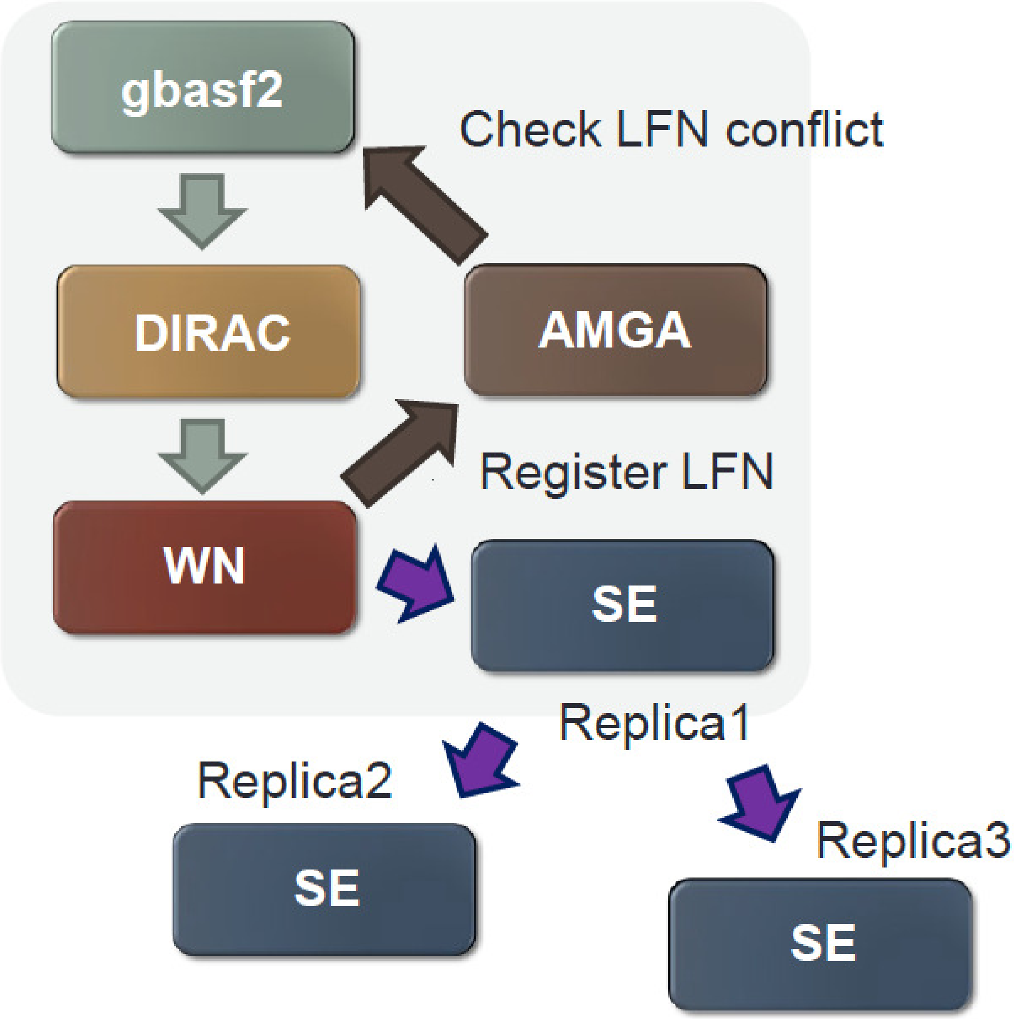
Different from LHC~MONARC model

Metadata handling is essential part of Belle II computing

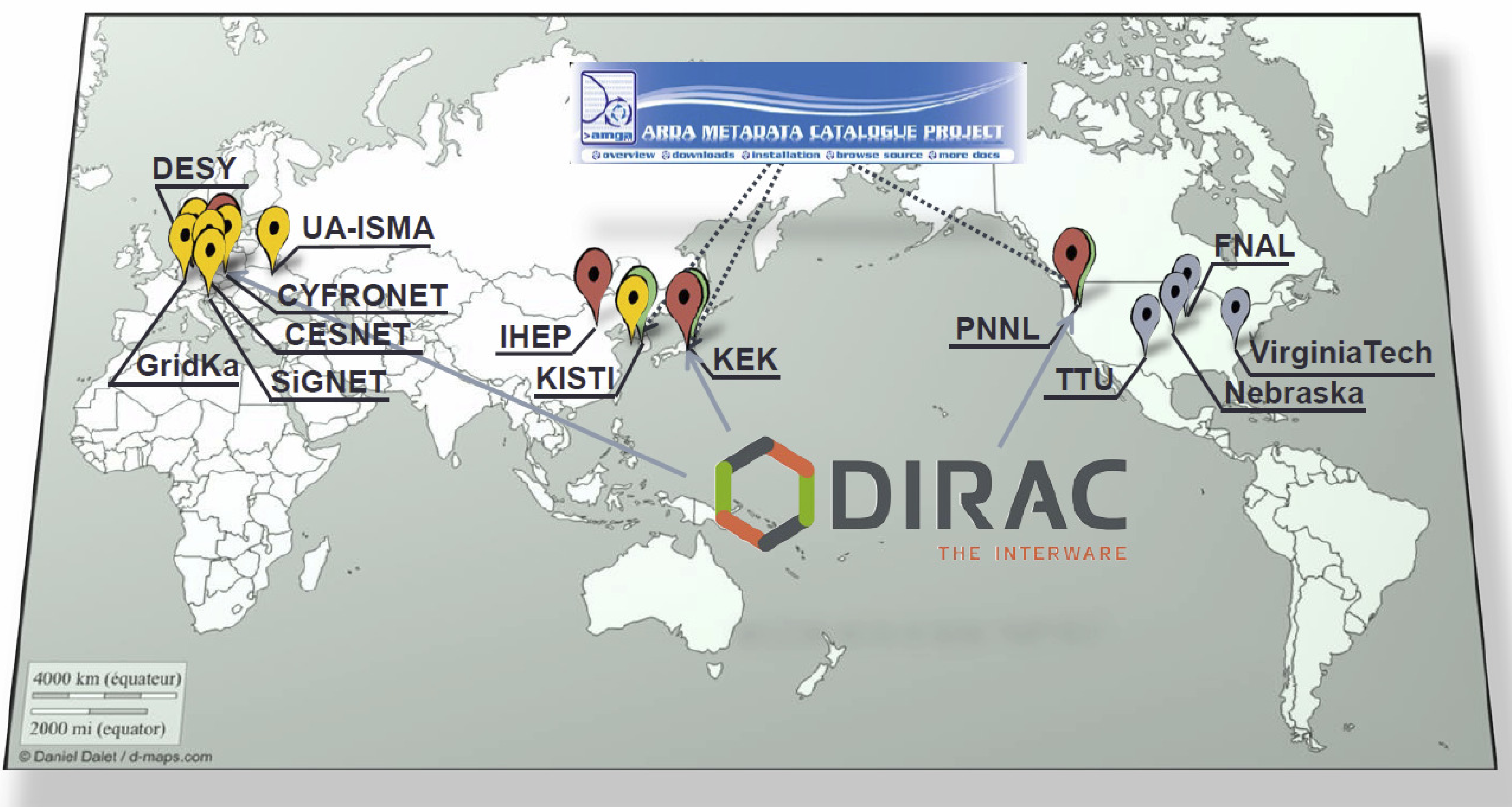
Source: Hideki Miyake(KEK)

Belle II Distributed Computing System

- One of the Belle II Computing Software Infrastructure
  - DIRAC (distributed workload management)
  - AMGA (metadata catalog)
  - gBasf2 (job submission client)
- AMGA master server at KEK
  - amga01.cc.kek.jp
- Several AMGA slave server at world-wide
  - amga02.cc.kek.jp (KEK)
  - amga.pnl.gov (PNNL)
  - belledh.kisti.re.kr (KISTI)



Belle II Distributed Computing Grid



4 DIRAC server
4 AMGA server

+

LCG sites
OSG sites

Source: Hideki Miyake(KEK)

Metadata Schema for Belle II (1/2)

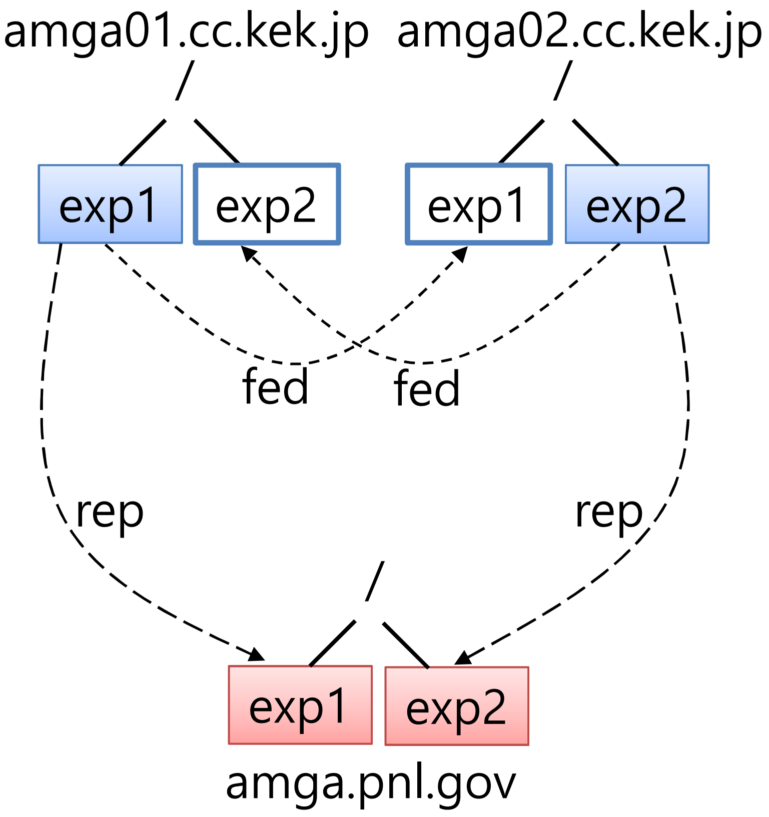
Source: Wenjing Wu(IHEP)

- File-level Metadata Schema
  - Example: /belle/user/wuwj/project1/outfile1

Attribute	Description	AMGA datatype
id	unique identifier	int
lfn	LFN of the logical file	varchar(1024)
guid	guid of the logical file	varchar(32)
status	(good or bad, maybe further values, e.q. for MC files that still have to be validated)	varchar(128)
events	Number of events	int
experiment	Experiment numbers (array)	int
runL	Lowest run number	smallint
runH	Highest run number	smallint
eventL	Lowest event number	int
eventH	Highest event number	int
parentid	IDs of parent files (array)	int
date	Data and time of creation	date
site	ID of site where the file was created	varchar(32)
software	software build number	varchar(32)
versionid	svn version of user source code	smallint
user	user id	varchar(32)
stream		smallint

Distributed AMGA Deployment for Belle II

❖ Example Scenario



Load balancing (Federation)

If a client connects amga01 and tries to read or write on /exp2, the request will be redirected to amga02

Data redundancy (Replication)

/exp1 and /exp2 will be replicated to the same dir on amga.pnl.gov

🟦 hosted data □ mounted data 🟥 replicated data

Current Status of AMGA usage at Belle II

- The performance of AMGA was evaluated during the 1st mass MC data challenge
  - Period: Feb 28th ~ Mar 20th
  - Generate 60M BB events (one B→Dπ, the other B→anything)
    - 1000 events/Job → 60000 jobs
  - No critical problems with AMGA in terms of stability and performance
  - Valuable feedback in improving the AMGA python APIs and some minor bugs fixing
- AMGA Integration with DIRAC is under investigation
  - Can handle the AMGA metadata by DIRAC-API level
  - Evaluated the possibility of AMGA integration with DIRAC with a prototype implementation from Hideki Miyake(KEK)
- Some federated/replicated metadata catalog scenarios are under discussion
  - Federation for metadata load balancing
  - Replication for metadata redundancy

Future Plans on AMGA Support

- Post-EMI Activity Plan
  - AMGA level of service support through GGUS
    - As the AMGA product is included in the EGI UMD (Unified Middleware Distribution), AMGA team will continue provide a base level of service through the EGI GGUS
    - A response time of 5 working days regardless of the ticket priority level
  - Participation to the MeDIA Initiative (Middleware Development and Innovation Alliance)
    - Open, lightweight collaboration on the coordination of distributed middleware technologies
- The evolutional development and maintenance of AMGA will continue in KISTI with its internal budget (~2FTE)
- AMGA support for Belle II
  - Continuing participation to the future Belle II mass MC campaigns
  - Technical support for the future Federated AMGA service deployment and the AMGA Integration with DIRAC

References

- AMGA Main Homepage
- AMGA 2.4.0 User Manual
- AMGA GILDA Wiki pages
- Belle II AMGA Wiki pages

Q & A

AMGA Metadata Catalog