AMGA
Metadata Service

Vladimir Dimitrov
IPP-BAS

"gLite middleware Application Developers Course",
Plovdiv, Bulgaria, 4.04.2008

Acknowledgements

- **This presentation primarily consists in slides from:**
  - Mike Mineter
    - 2006, 2007
  - Tony Calanducci
    - Third EELA Tutorial for Managers and Users
    - Rio de Janeiro, 26-30 June 2006
  - Nuno Santos, Birger Koblitz
    - 20 June 2006
    - Workshop on Next-Generation Distributed Data Management
  - Patricia Méndez Lorenzo: UNOSAT application using AMGA
    - User Forum
    - CERN, 1st March 2006
    - http://indico.cern.ch/materialDisplay.py?contribId=23&sessionId=11&materialId=slides&confId=286
  - Documents and examples from AMGA web site.

Contents

- Background and Motivation for AMGA
- Interface, Architecture and Implementation
- Metadata Replication on AMGA
- Examples
- Further information

Metadata on the GRID

- **Metadata is data about data**
  (a formal definition)

- **On the Grid: information about files**
  - Describes files
  - Locate files based on their metadata

AMGA Implementation

- **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…
  - More on this later

Metadata Concepts

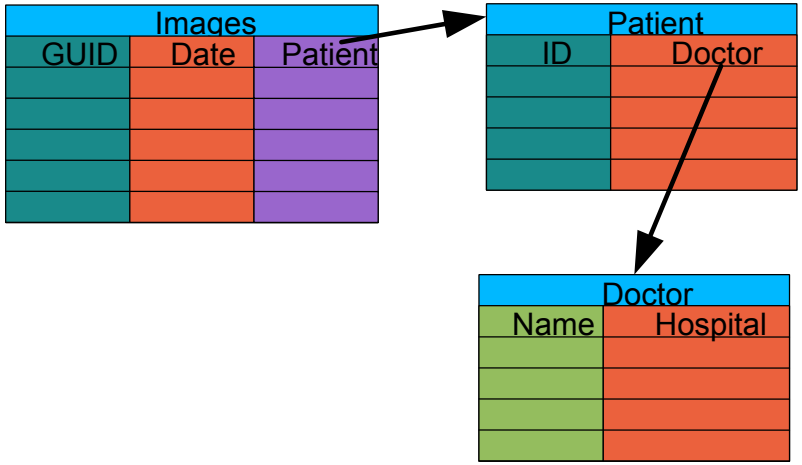
- **Some Concepts**
  - Metadata - List of attributes associated with entries
  - Attribute – key/value pair with type information
    - Type – The type (int, float, string,…)
    - Name/Key – The name of the attribute
    - Value - Value of an entry's attribute
  - Schema – A set of attributes
  - Collection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

Examples

- **gLibrary**
  - Files are saved on SEs and registered into LFC file catalogues
  - The AMGA Metadata Catalogue is used to archive and organize metadata and to answer users' queries.
- **LHCb-bookkeeping**
  - 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**
  - Job management system
    - Developed jointly by **Atlas** and **LHCb**
  - Uses AMGA for storing information about job status
    - Small amount of highly dynamic metadata

Biomed VO

- **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 4th EGEE conference in 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://twiki.cern.ch/twiki/bin/view/EGEE/DMEncryptedStorage

UNOSAT Presentation

UNOSAT is a United Nations Initiative

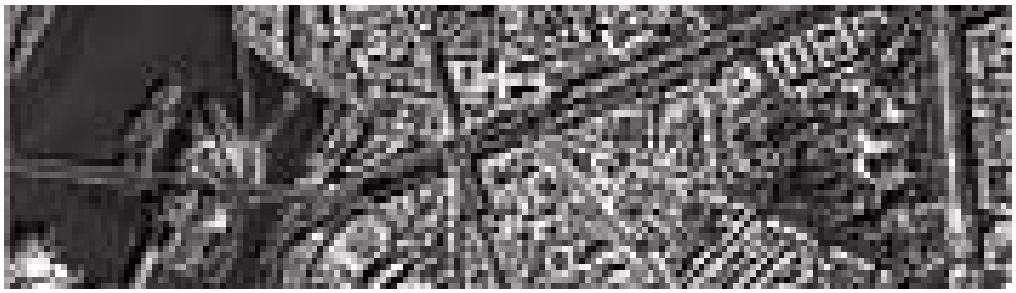
Objectives

- 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

One step further: GRID

- Potential Bottlenecks:
  - UNOSAT beginning to suffer from limited capacity and processing power
  - Multiple satellites being launched
  - Larger and larger storage capacity needed
-
  - 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 in summer 2005 for UNOSAT

- 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 [49]) has been adapted for UNOSAT needs

UNOSAT Production inside GRID

- UNOSAT provided us with 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

A GRID Metadata Catalogue

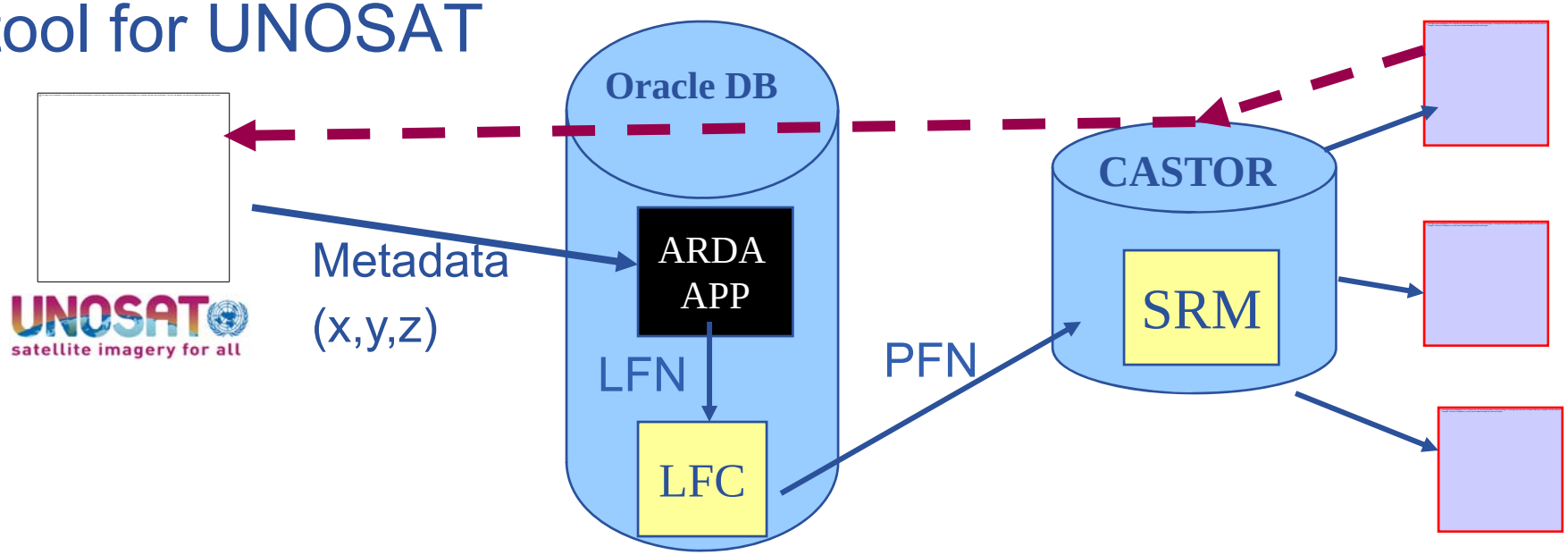
LFC Catalogue

- Mapping of LFN to PFN

UNOSAT requires

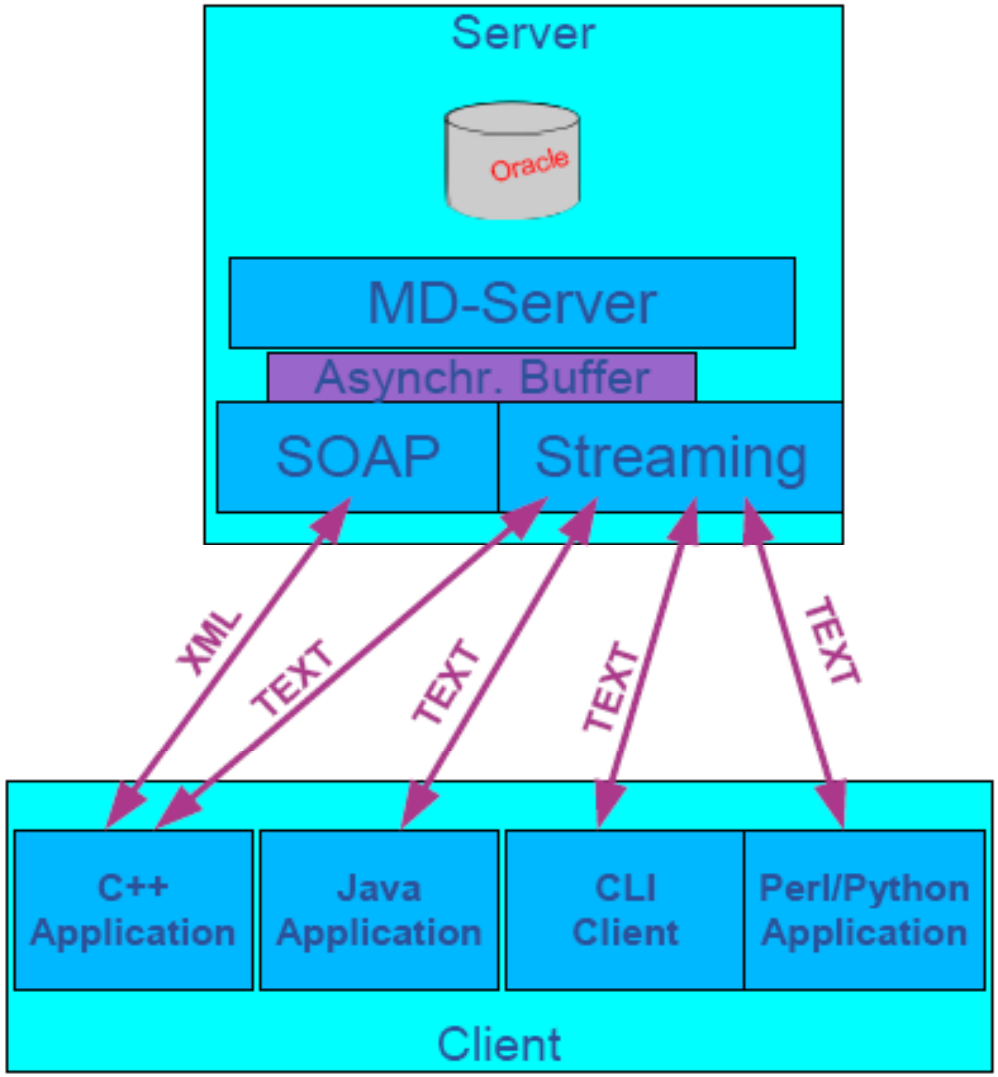
- User will give as input data certain coordinates
- As output, want the PFN for downloading

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



AMGA Implementation

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

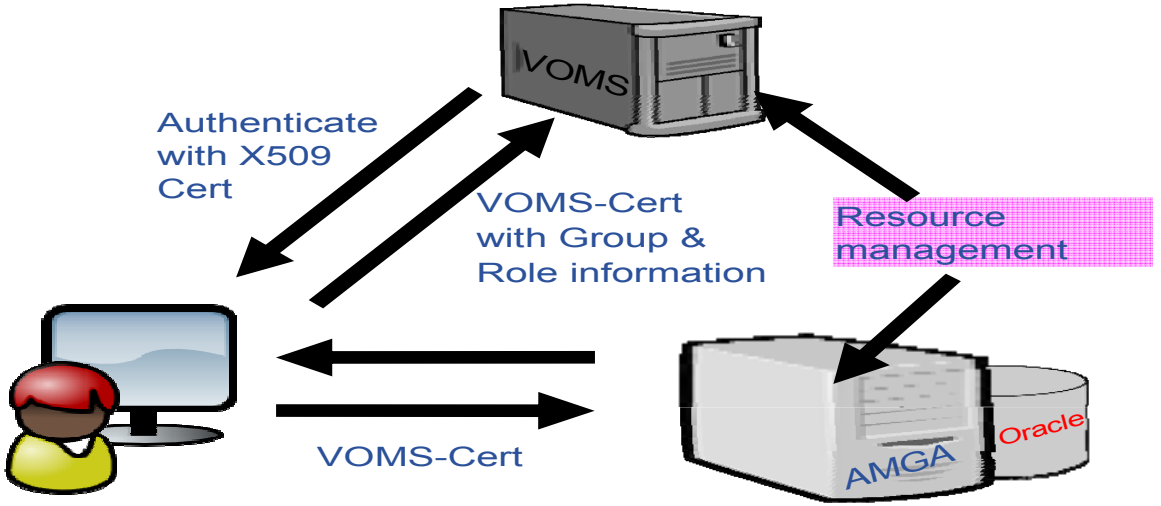


AMGA Features

- **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

Security

- **Unix style permissions**
- **ACLs – Per-collection or per-entry.**
- **Secure connections – SSL**
- **Client Authentication based on**
  - Username/password
  - General **X.509** certificates
  - Grid-proxy certificates
- **Access control via a Virtual Organization Management System (VOMS):**

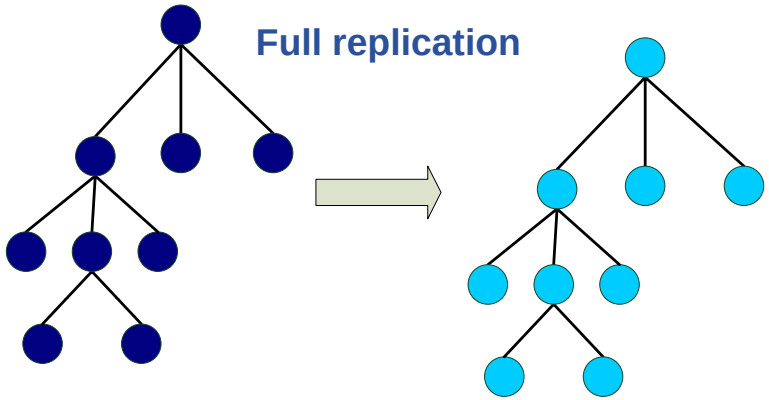


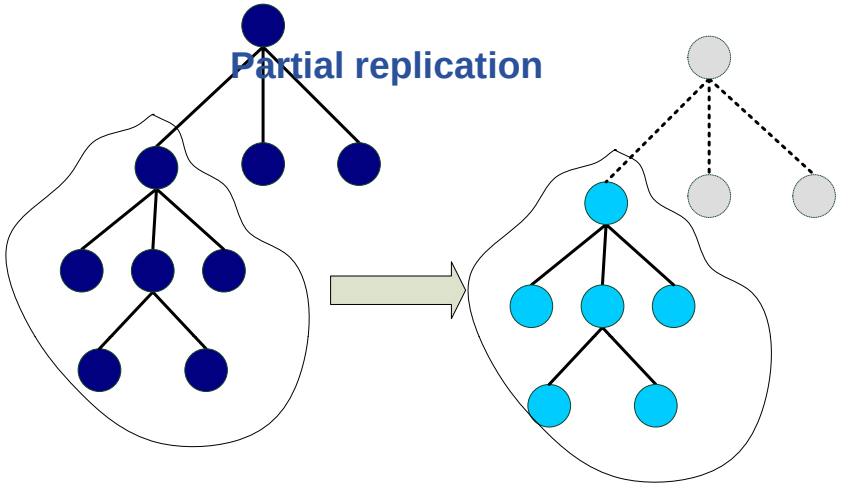
Metadata Replication

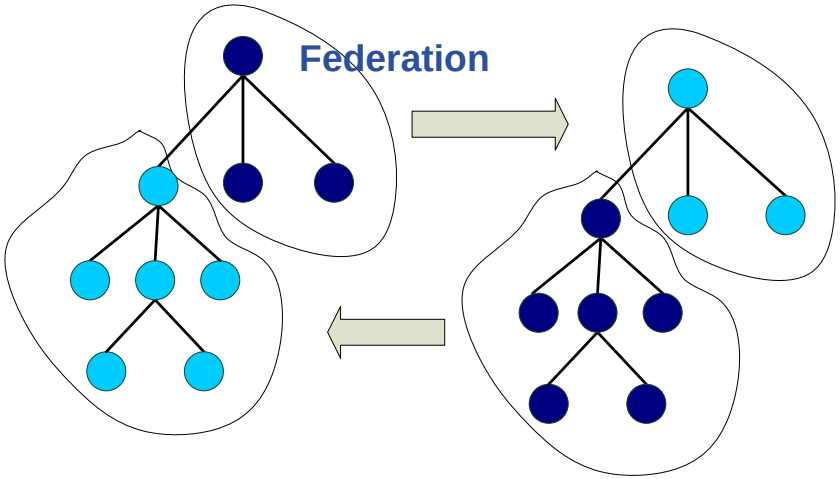
- **Currently working on replication/federation mechanisms for AMGA**
- **Motivation**
  - Scalability – Support hundreds/thousands of concurrent users
  - Geographical distribution – Hide network latency
  - Reliability – No single point of failure
  - DB Independent replication – Heterogeneous DB systems
  - 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
- http://amga.web.cern.ch/amga/publications/nsantos2006AMGAReplication.pdf

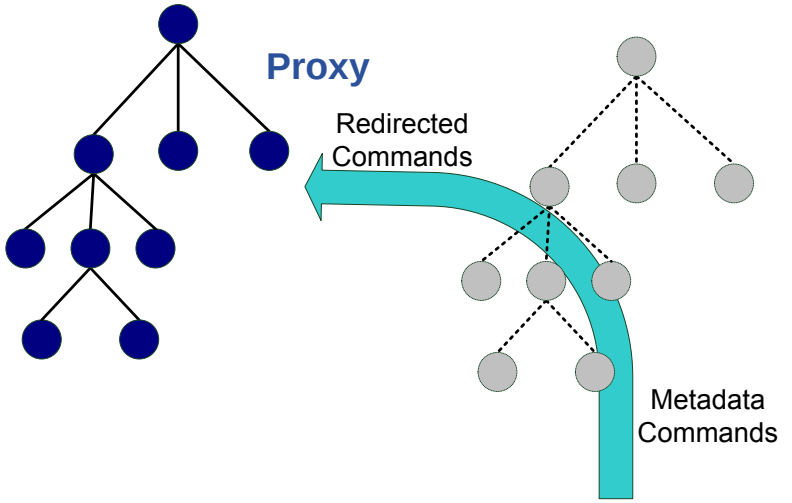
Metadata Replication

Some use cases









Not only metadata….

- **But also….**

  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

gLibrary prototype implementation

- **Files are saved on SEs and registered into LFC file catalogues**
- **The AMGA Metadata Catalogue is used to archive and organize metadata and to answer users' queries.**
- **gLibrary is built using the following AMGA collections:**
  - **/gLibrary** contains generic metadata for each entry
  - /gLAudio, /gLImage, /gLVideo, /gLPPT, /EGEEPPT, /gLDoc, … are examples of collections of "additional features" (shown later)
  - /gLTypes
    - keeps the associations between document types and the names of the collection that contains the "additional features"
    - is used by gLibrary to find out where it has to look when new document types are added into the system (extensibility)
  - /gLKeys is used to store Decryption Keys

gLibrary Example of entries

Collection	/gLibrary			
Entry Names	**Attributes**			
	FileName	**PathName**	**Type**	**Submitter**
4ffaffc8-26e7-4826-b460-3d5bf08081a4	DedicatoAte.mp3	/grid/gilda/calanducci	Audio	Tony Calanducci
00454dca-a269-4b93-8a45-c4012af05600	ardizzonelarocca_is_231005.ppt.gpg	/grid/gilda/calanducci/EGEE	EGEEDOC	Tony Calanducci

/gLibrary (continuum)

Attributes

SubmissionDate	Encryption	Description	Keywords	CreationDate
2006-01-05 00:00:00	false	Canzone delle vibrazioni che ha ricevuto un enorme successo tra i teenagers nel 2003	Vibrazioni	2004-02-05 00:00:00
2005-01-05 16:44:22	true	gLite Information System	R-GMA, RGMA, BDII, IS	2005-10-05 23:40

Example of gLibrary collections

Collection	/gLTypes
Entry names	Attributes
	Path *(refers to a collection)*
Audio	/gLAudio
Image	/gLImage
Video	/gLVideo
Documents	/gLDOC
PowerPoint	/gLPPT
EGEEDOC	/EGEEPPT

Collection	/gLKeys
Entry names	Attributes
	Passphrase
00454dca-a269-4b93-8a45-c4012af05600	ardizzo

"additional features"

Collection	/EGEEPPT							
Entry names	Attributes							
	Title	Runtime	Author	Type	Date	Event	Speaker	Topic
00454dca-a269-4b93-8a45-c4012af05600	Information Systems	00:30:00	Valeria Ardizzione, Giuseppe La Rocca	Theorical	2005-10-23	4th EGEE Conference	Giuseppe La Rocca, Valeria Ardizzone	R-GMA, BDII

Collection	/gLAudio					
Entry names	Attributes					
	SongTitle	Duration	Album	Genre	Singer	Format
4ffaffc8-26e7-4826-b460-3d5bf08081a4	Dedicato A Te	00:03:27	Dedicato A Te	Pop	Le Vibrazioni	MP3

gLibrary Security

- **User Requirements:**
  - a valid proxy with VOMS extensions
  - VOMS Role and Group needed to be recognized by **gLibrary** as a contents manager.
- **3 kinds of users:**
  - gLibraryManager: (s)he can create new content type and allows a generic VO user to become gLibrarySubmitter
  - gLibrarySubmitters: they can add new entries and define access rights on the entries they create.
    - Fine-grained permission (reading, writing, listing, decrypting) settings on each entry: whole VO members, VO groups, list of DNs
  - generic VO users: browse and make queries (on entries they have access to)
- **Basic level of cryptography:**
  - New files saved on SEs can be encrypted beforehand with a symmetric passphrase that will be saved in /gLKeys. Only selected users (that have a specific DN in the subject of their VOMS proxy) can access the passphrase and decrypt the file.

AMGA client installation

- **The clients (C++, Java and Python) are provided as RPMs packages here:**
  http://project-arda-dev.web.cern.ch/project-arda-dev/metadata/downloads/

```
rpm -i glite-amga-cli-1.3.0-1.i386.rpm
(for C++ API)
  or
rpm -i glite-amga-api-java-1.2.5.rpm
(for Java API)
```

- Copy the **/opt/glite/etc/mdclient.config** client configuration file into the directory from which you intend to work or into **~/.mdclient.config** and customize it according to the instructions in the manual.

- *There is also a **Python** Client API module available as an RPM package.*

Example: gLibrary queries

Initialize your VOMS proxy asking to be member of the gilda VO

Edit your .mdclient.config setting Login=NULL (user will be retrieved from your proxy extensions)

Log into AMGA using mdclient

```
$ voms-proxy-init --voms gilda
$ voms-proxy-info -fqan
/gilda/Role=NULL/Capability=NULL
$ grep Login .mdclient.config
Login = NULL
```

Suppose we want to look for all contents about VOMS

```
Query> whoami
>> gilda
Query> selectattr /gLibrary:FILE /gLibrary:FileName /gLibrary:Type
'like(/gLibrary:Keywords, "%VOMS%")'
>> 1f6e9ac6-5c86-4599-b03b-560e0e7ea38a
>> VOMS_server_Installation.ppt.gpg
>> EGEEDOC
```

EGEEDOC attributes

```
Query> getattr /DLTypes/EGEEDOC Path
>> EGEEDOC
>> /EGEEPPT
```

Example: gLibrary queries (II)

JOIN between the 2 tables to extract all the information we like

```
Query> selectattr /gLibrary:FILE /gLibrary:FileName /gLibrary:Description
/EGEEPPT:Author /EGEEPPT:Title /EGEEPPT:Event '/gLibrary:FILE=/EGEEPPT:FILE and
like(/gLibrary:Keywords, "%VOMS%")'
>> 1f6e9ac6-5c86-4599-b03b-560e0e7ea38a
>> VOMS_server_Installation.ppt.gpg
>> VOMS Server installation tutorial done in Venezuela
>> ziggy, Giorgio
>> Installing a gLite VOMS Server
>> First Latin American Workshop for Grid Administrators
```

Decrypting

```
Query> selectattr /gLibrary:FILE DecryptKeyDir 'FILE="1f6e9ac6-5c86-4599-b03b-
560e0e7ea38a"'
>> 1f6e9ac6-5c86-4599-b03b-560e0e7ea38a
>> /DLKeys/gildateam
```

But …

```
Query> getattr /gLKeys/gildateam/1f6e9ac6-5c86-4599-b03b-560e0e7ea38a
Passphrase
Error 4: Permission denied
```

Because gilda is not a member of the gildateam group

Practicals using AMGA mdclient

```
Query>
dir /gilda/plovdiv

listattr /gilda/plovdiv

getattr /gilda/plovdiv/ Author
getattr /gilda/plovdiv/ Title
getattr /gilda/plovdiv/ Date

find /gilda/plovdiv 'like(Author, "Vla%")'
find /gilda/plovdiv 'like(Author, "P%")'

cd /gilda/plovdiv
setattr Test01 Author 'Pesho G. Petrov'
getattr Test01 Author
quit
```

Using the C++ Client API

- **Two different C++ client APIs are available for the AMGA metadata service:**
  - `md_api` - ***many API functions***
  - `MDClient` - C++ *class. A direct interface but does not parse the responses of the server into suitable structures, while this is done by the* md_api*.*

Both ways to access the metadata service from C++ depend on an existing and accessible mdclient.config

Example of using md_api

```
#include "client/md_api.h"
#include <iostream>

int main (int argc, char *argv[])
{
  std::cout << "Listing attributes of /test\";
  std::list < std::string > attrList;
  std::list < std::string > types;
  if( (res=listAttr("/test", attrList, types)) == 0){
    std::cout << "  Result:" << std::endl;
    std::list< std::string >::iterator I=attrList.begin();
    while(I != attrList.end())
      std::cout << "  >" << (*I++) << "<" << std::endl;
  } else {
    std::cout << "  Error: " << res << std::endl;
  }

/* more code here ... */

  return 0;
}
```

Example of using MDClient C++ class

```
#include <MDClient.h>
#include <iostream>

int main (int argc, char *argv[])
{
  int res;

  MDClient client;
  //  client.setDebug(true);

  if(client.connectToServer()){
    std::cout << client.getError() << std::endl;
    return 5;
  }

  std::string command="pwd";
  if( ( res=client.execute(command)) ){
      std::cout << "  ERROR: execute failed"
                << "   (" << res << "): "
                << client.getError() << std::endl;
      return res;
  }

/* more code here ... */

return 0;
}
```

Further information

- **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`

Conclusion

- **AMGA – Metadata Service of gLite**
  - Useful for simplified DB access
  - Integrated in the Grid environment (Security)
- **Replication/Federation under development**
- **Tests show good performance/scalability**

- **Already deployed by several Grid Applications:**
  - LHCb, ATLAS, Biomed, …
  - DLibrary