



GridPP

UK Computing for Particle Physics

Metadata

(data about data)

Status and Plans

GridPP-16

Thomas Doherty

GridPP Physicist Programmer



UNIVERSITY
of
GLASGOW



- Introduction
 - The metadata team
 - MonAMI
 - VOMS
 - SQLite
 - glite Metadata Interfaces
 - Event-Level Metadata
 - Conclusion





GridPP

UK Computing for Particle Physics

The Metadata Team

- The Metadata Team (based in Glasgow) consists of:
 - Tony Doyle - a.doyle@physics.gla.ac.uk
 - Team leader and involvement in weekly meetings
 - Paul Millar - p.millar@physics.gla.ac.uk
 - MonAMI and AMI
 - Caitriana Nicholson - c.nicholson@physics.gla.ac.uk
 - Event-Level Metadata
 - Tom Doherty - t.doherty@physics.gla.ac.uk
 - VOMS
 - SQLite
 - glite Metadata Interfaces (Steven Hanlon)
 - Helen McGlone (Student based in CERN)
 - Event-Level Metadata



UNIVERSITY
of
GLASGOW



- Why Monitor?

- Site-admins, Developers, End-users and VO/site want to know why:
 - a service is not working
 - a job is blocked
 - if hardware needs an upgrade
- We want an **easy** way to monitor and send information to **multiple** information sources: a universal sensor!

- What is a Universal Sensor?

- Metadata service has many components:
 - database, application-server, metadata application, the hardware...
- Different groups want different subsets of available information,

Different groups might be using different information system

- RGMA, MonaLisa, SAM(E), Ganglia, Nagios, ...



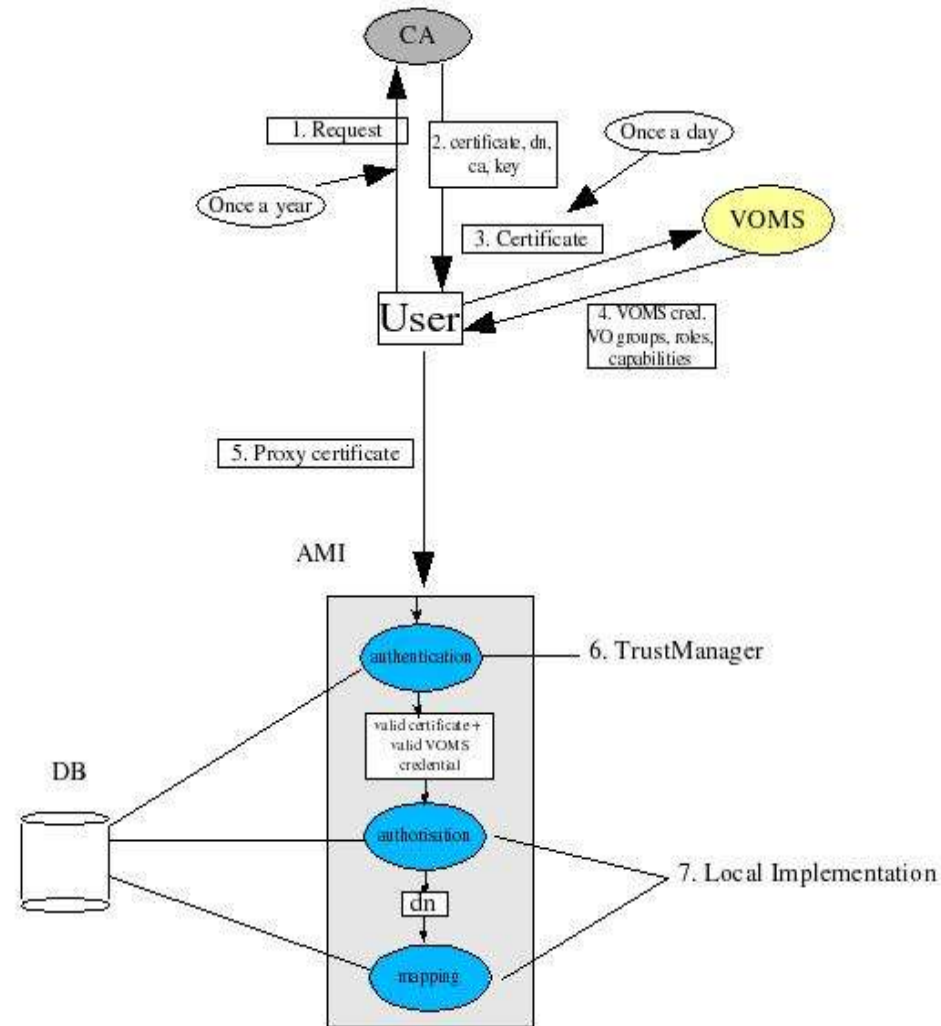


- MonAMI: a small, light-weight daemon.
- Gathers information and sends it somewhere.
- Uses plugins for **monitoring** and **reporting**.
- For a more in depth look into how it works see slides for talk given in monitoring discussion session yesterday on MonAMI
- See <http://monami.sourceforge.net/>
- Future work: adding more plugins (DPM & FTS planned), supporting ATLAS db monitoring, adding support for monitoring-on-demand





- Obtain grid-certificate from CA
- Request VOMS proxy certificate
- Access AMI using this certificate
- User authenticated using TrustManager
- Authorisation achieved depending on user task mapping correctly from their VOMS group/role to their registered user role within AMI
- This poses a problem for AMI





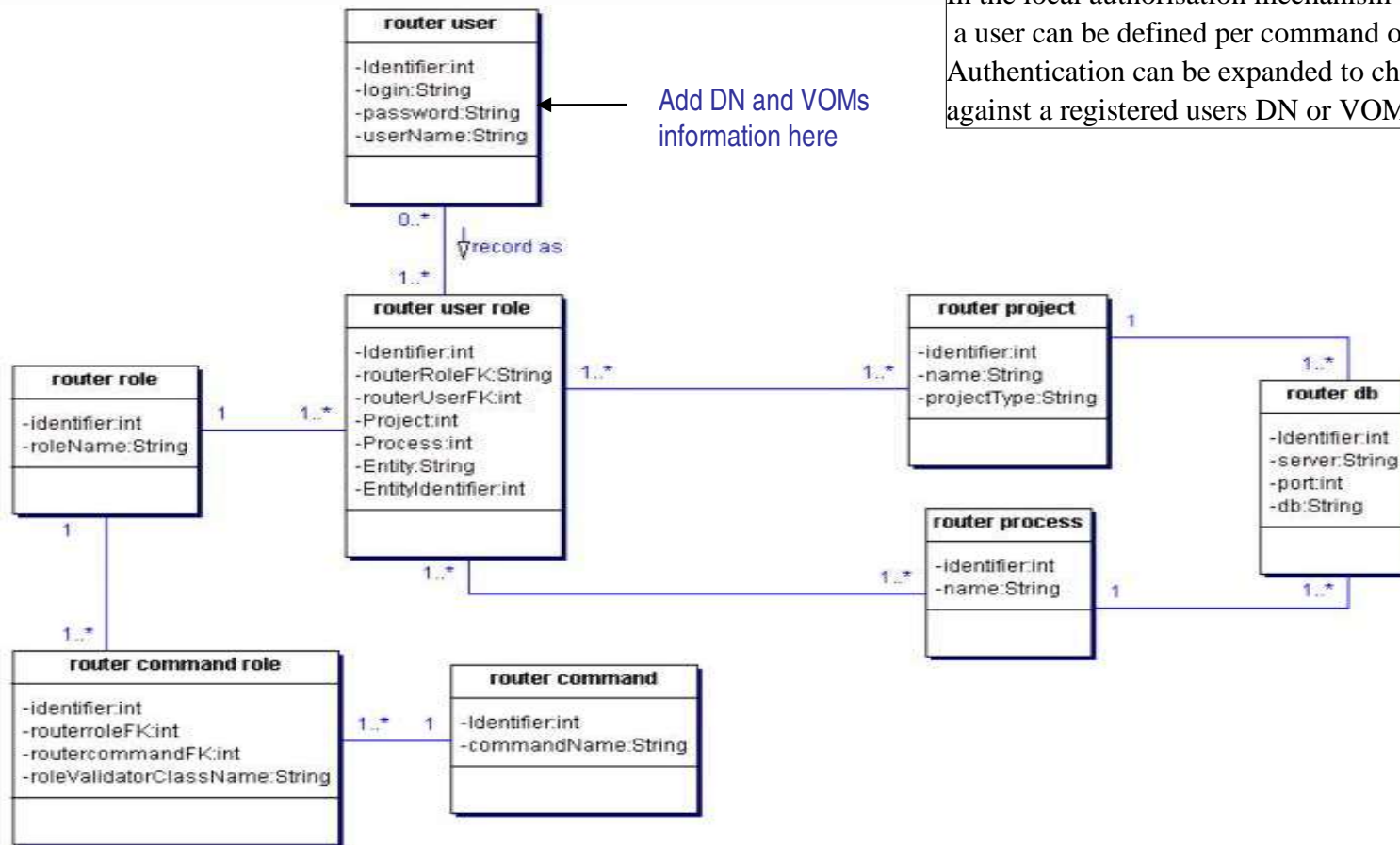
Problems with VOMS and possible solutions

- The main search interface for AMI is browser based
- Browsers at the moment cannot handle proxy certificates
- There are many possible solutions:
 - Mod_gridsite – this method fetches lists of DNs from the VOMS system per group/role
 - MyProxy – this method allows for the storing of proxy-certificates on a remote server
- A purely Java/Tomcat prototype is being developed at the moment





Authorisation and DNs



Add DN and VOMs information here

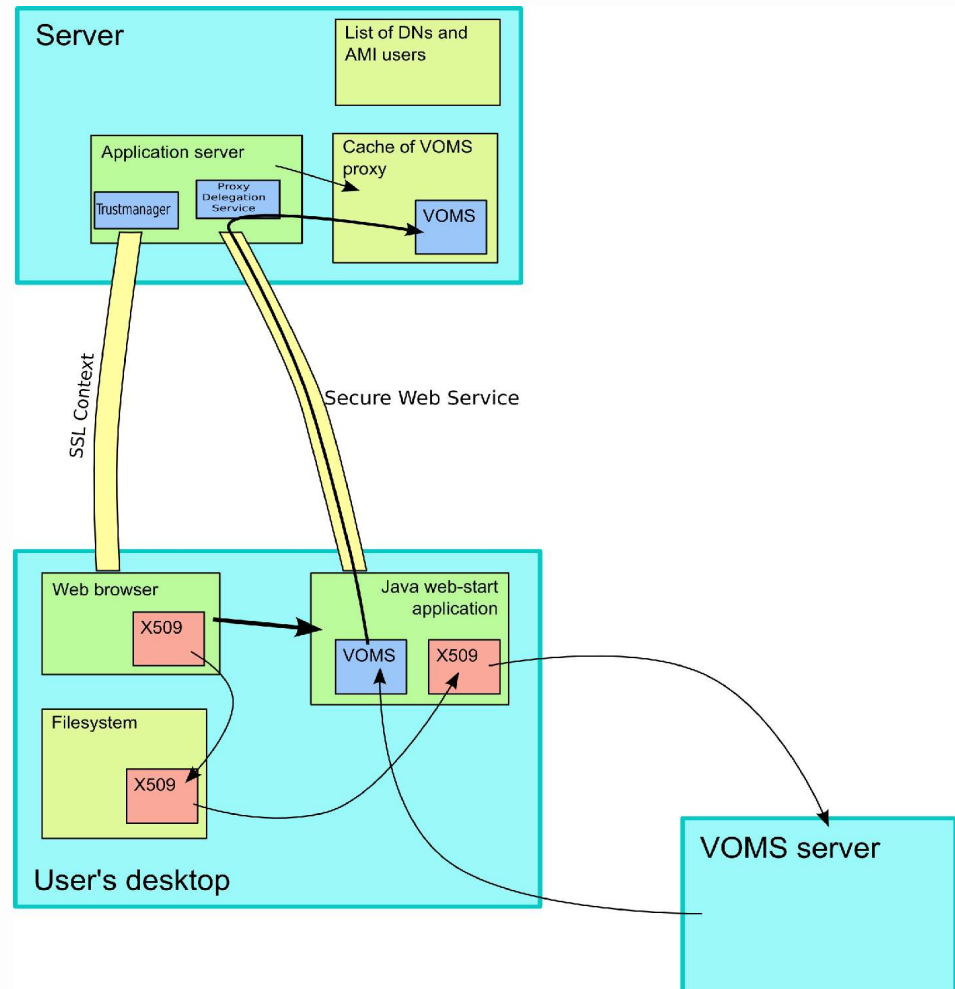
In the local authorisation mechanism the rights of a user can be defined per command or entity. Authentication can be expanded to check against a registered users DN or VOM group/role.





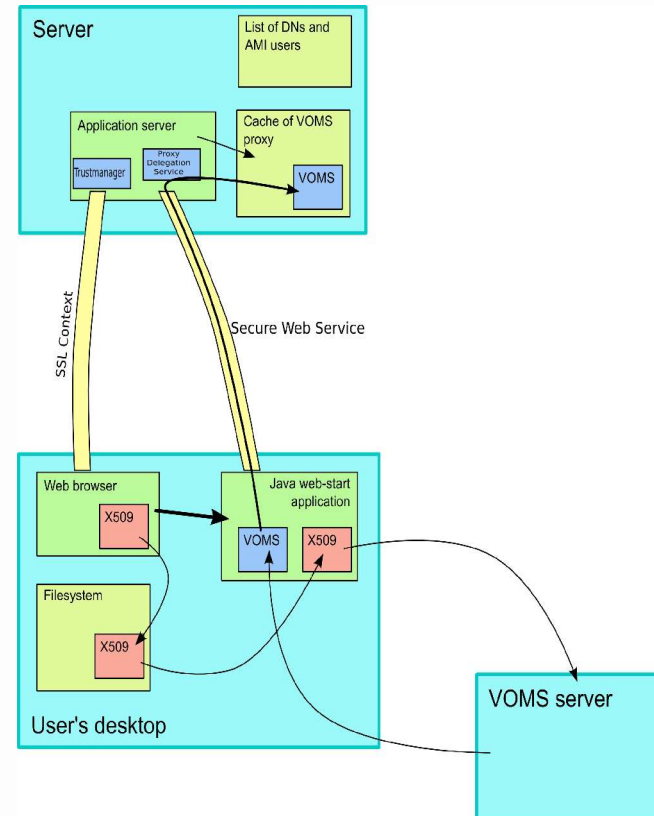
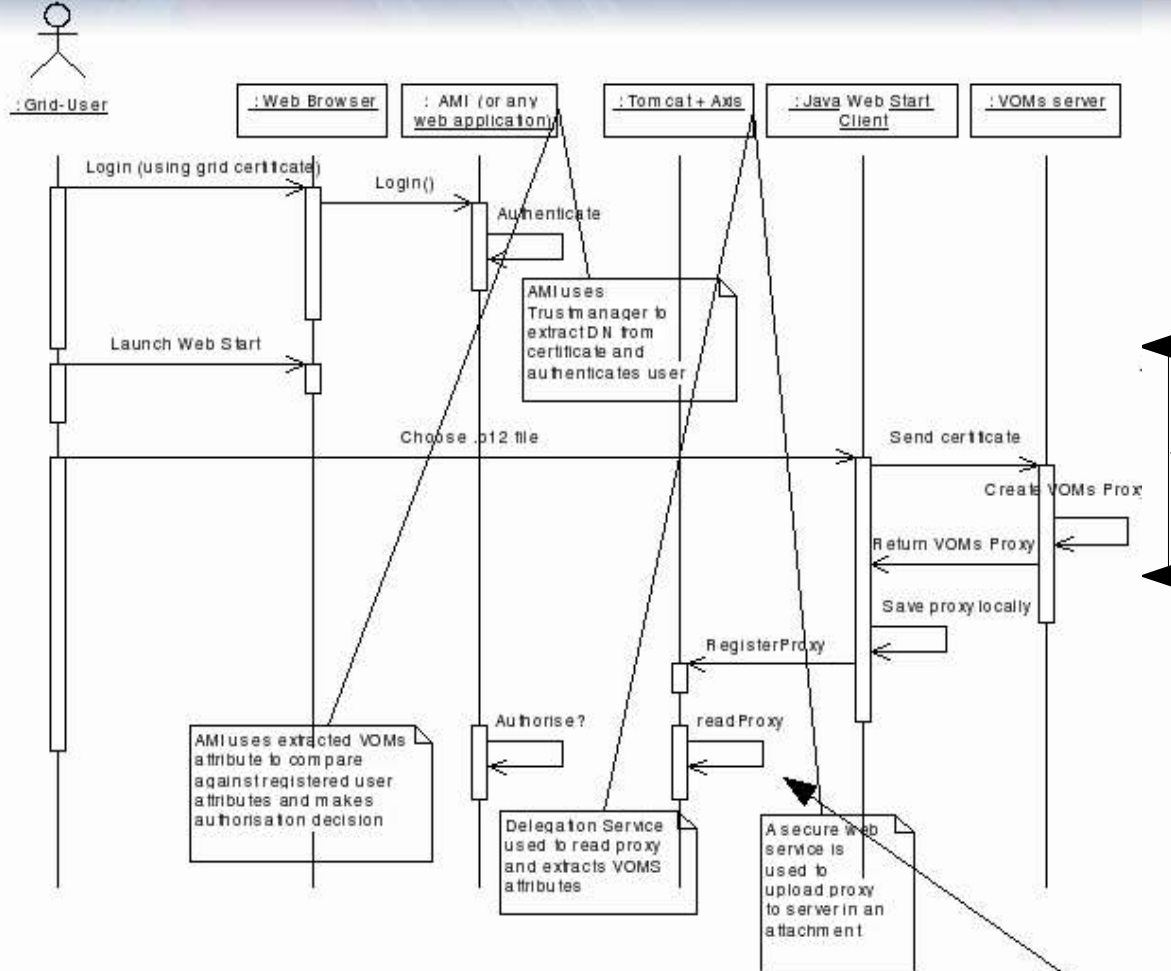
Java VOMS solution

- The Java VOMS solution briefly consists of the following steps:
 - 1) The grid user authenticates themselves to AMI (web app.) using a grid certificate
 - 2) AMI allows the user to launch a Java web start application to create a VOMS proxy
 - 3) The user chooses the x509 certificate to be sent to the VOMS server
 - 4) The VOMS server sends back a VOMS proxy certificate to the W.S application
 - 5) Using Axis set up in Tomcat the VOMS proxy file is uploaded to the server via a secure web service attachment
 - 6) A proxy delegation service is available on the server
 - 7) The users VO and role can be extracted from the proxy and used for authorisation purposes





VOMS Authorisation

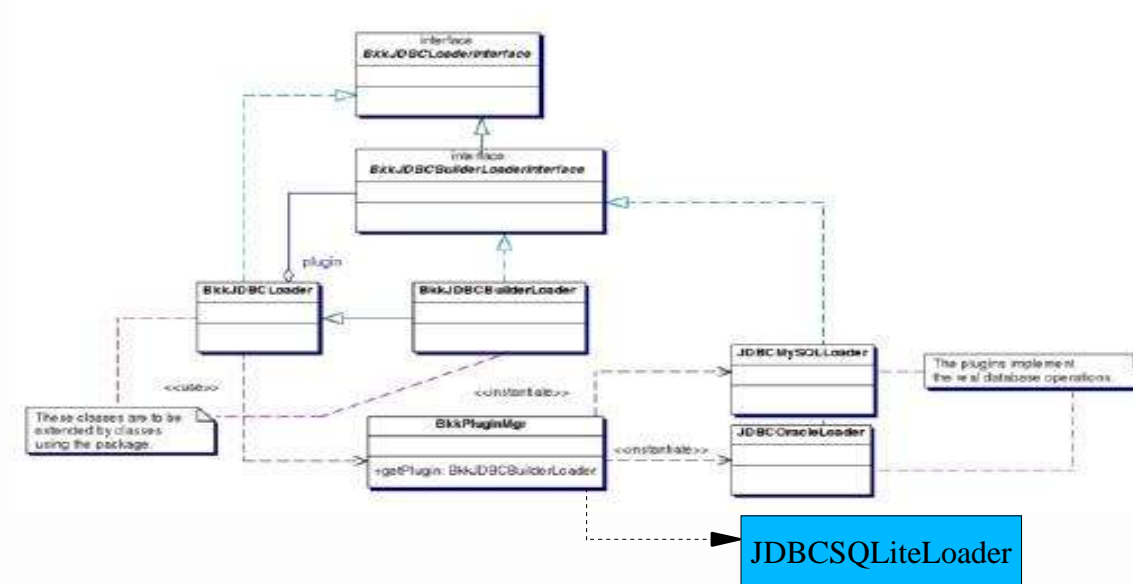


Functionality provided
by Gidon Moont



SQLite plug-in

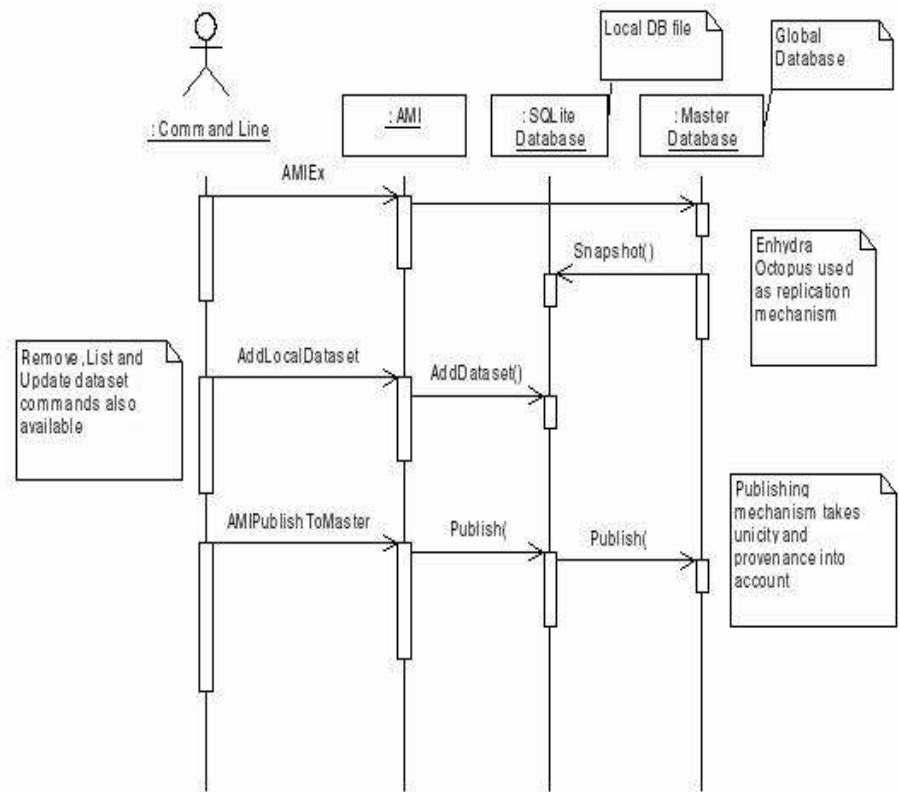
- SQLite plugin developed for AMI to allow for offline/private work
- AMI Architecture allowed for easy integration of SQLite code as it plugged in alongside Oracle and MySQL.
- An interface called BkkJDBCBuilderLoader was implemented to match the functionality of the other two backends.





Publishing mechanism

- The SQLite implementation provides the following:
 - 1) Additional AMI commands introduced to allow for this offline/private work
 - 2) A snapshot mechanism using Octopus functionality that takes a copy of a global DB and saves it to an SQLite file.
 - 3) And a mechanism that allows the private work to be published back to other DB backends for global ATLAS viewing





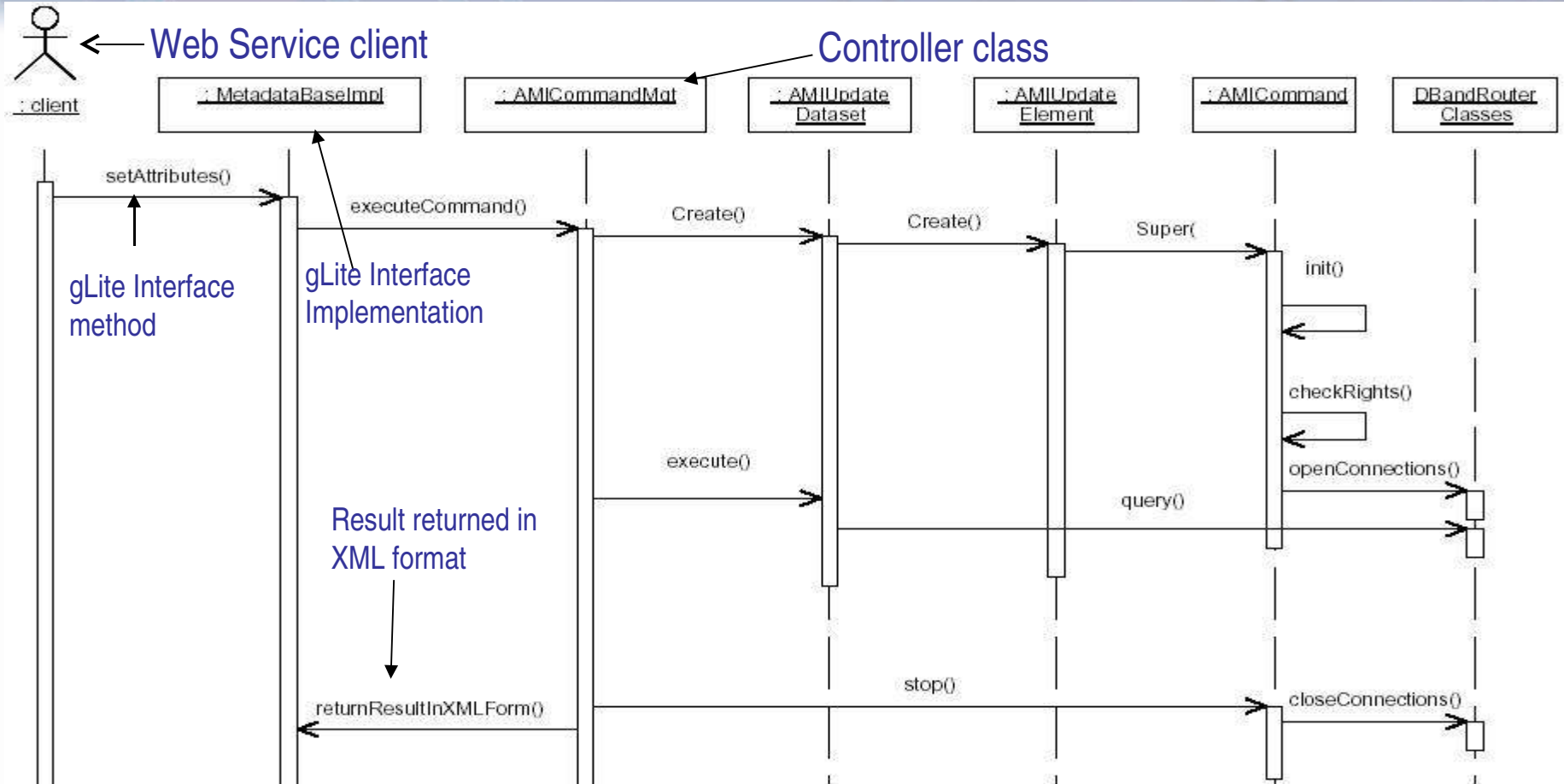
Why use gLite?

- Problem
 - Want AMI to be not only useful to physicists but to any grid user
- Solution
 - Adapt AMI to handle the gLite Metadata Interface.



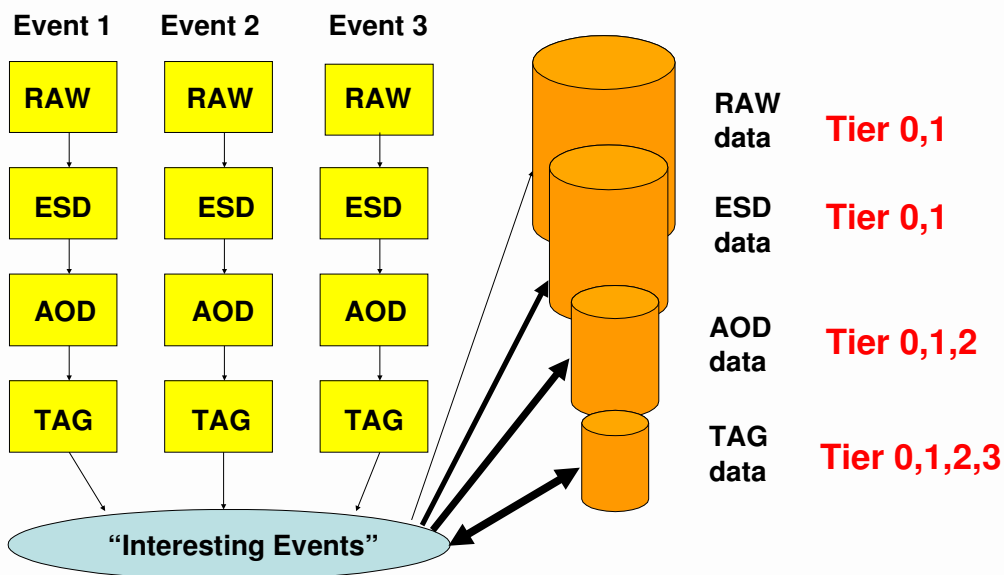


Adaptation of AMI Architecture for gLite Interfaces





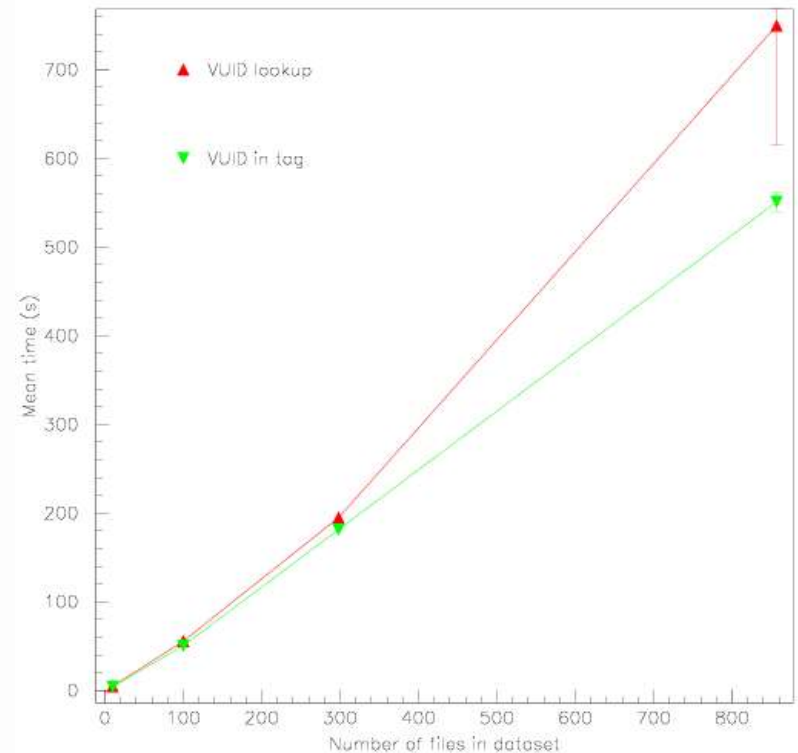
- Event level metadata (Tags) used to allow efficient access to interesting data
- Written as ROOT files at AOD production then loaded into relational database at CERN and replicated to Tier 1 sites (and Tier-2s depending on capabilities)
- ~ 1kB/event, total size ~ few TB
- Currently have Rome tags in central database at CERN (Oracle / MySQL) with some test copies at BNL and Glasgow (MySQL)





Tag/DDM integration

- ATLAS Distributed Data Management system DQ2 (*dataset-based*) must work with Tag system (*file-based*) to distribute data
- Integration work proceeding - can:
 - query Tag database from a site
 - define a new DQ2 dataset with required files, possibly at different sites
 - subscribe to dataset and have it delivered to local site
- Performance tests underway to evaluate inclusion of dataset information in Tags
 - Preliminary results indicate this may not be necessary





Basic Usability Tests

- Series of tests to show basic Tag usability before handing over for user-level testing:
 - reading from Tag database and finding correct file when running
 - on lxplus
 - on non-CERN site
 - from a grid job, on all grid flavours (OSG/LCG/NG)
 - generation of appropriate debugging information in case of failure
 - extraction job which generates new files based on Tag selection
 - make these into a dataset and deliver to site





Tag Distribution and Cataloging

- As well as relational database, have Tag files which are isomorphic to AOD – useful as indices to AOD files
- Want jobs to run with *either* database *or* dataset of files
- When many copies of Tag database / datasets exist, need cataloguing mechanism
- Eventually need to connect event-level metadata with other ATLAS metadata
 - will discuss these and other issues at Metadata Workshop, Oxford, July 4th - 7th (<http://www.physics.ox.ac.uk/ukmetadata/>)





GridPP

UK Computing for Particle Physics

Conclusion

- Gave a brief overview of the team and their areas of involvement
- MonAMI status and plans
- VOMS and work on a prototype to solve browser vs proxy certificate problem
- SQLite plug-in
- glite Metadata Interfaces
- Event-Level metadata



UNIVERSITY
of
GLASGOW



GridPP

UK Computing for Particle Physics

Extra slides



UNIVERSITY
of
GLASGOW



GridPP

UK Computing for Particle Physics

What is AMI?

- It is a developing application, which stores and allows access to dataset metadata for the ATLAS experiment
- It fulfils the need of many database-backed applications by offering a generic web service and servlet interface, through the use of self-describing databases
- supports geographical distribution with the use of web services and secure access with the use of grid-certificates

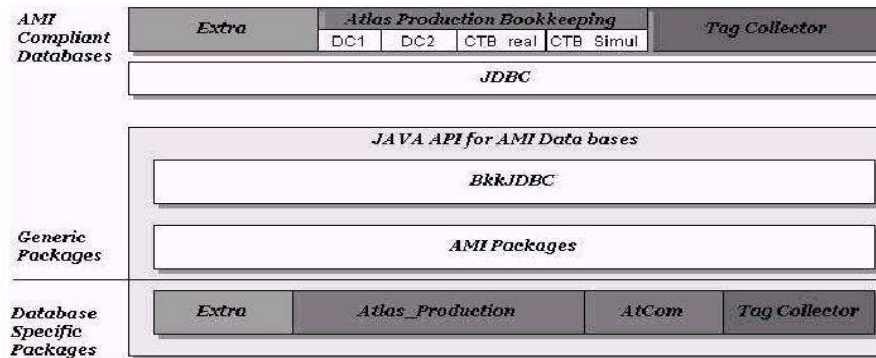
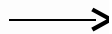


UNIVERSITY
of
GLASGOW



Fundamental Architecture of AMI

Command classes used in managing the databases



allows for connection to MySQL, Oracle and SQLite



Generic web search page for example



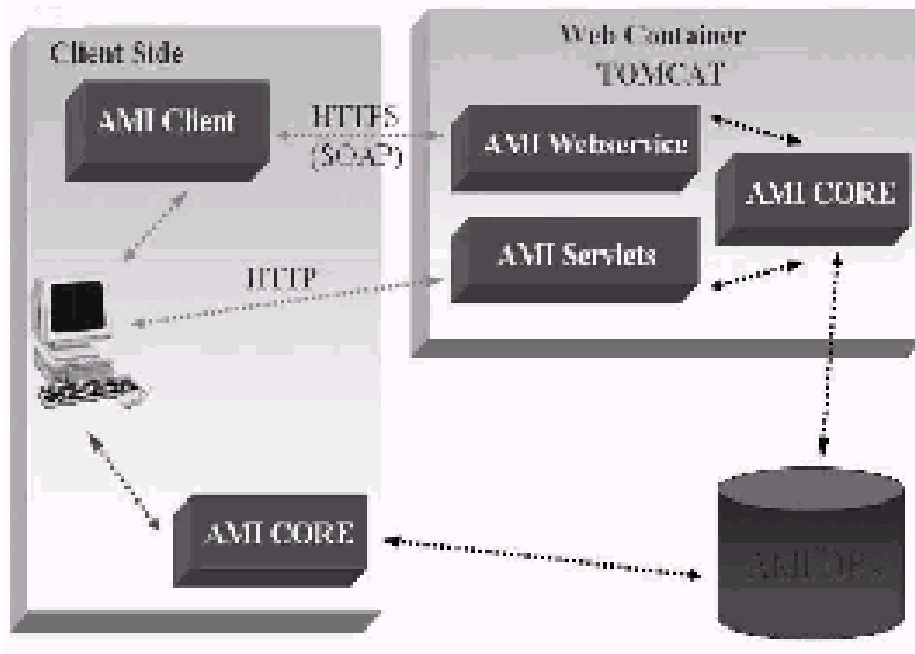
- Three-tier architecture - generic packages - middle layer
- command classes placed within generic packages key to the implementation of the gLite metadata interface methods





Fundamental Architecture of AMI and AMI clients

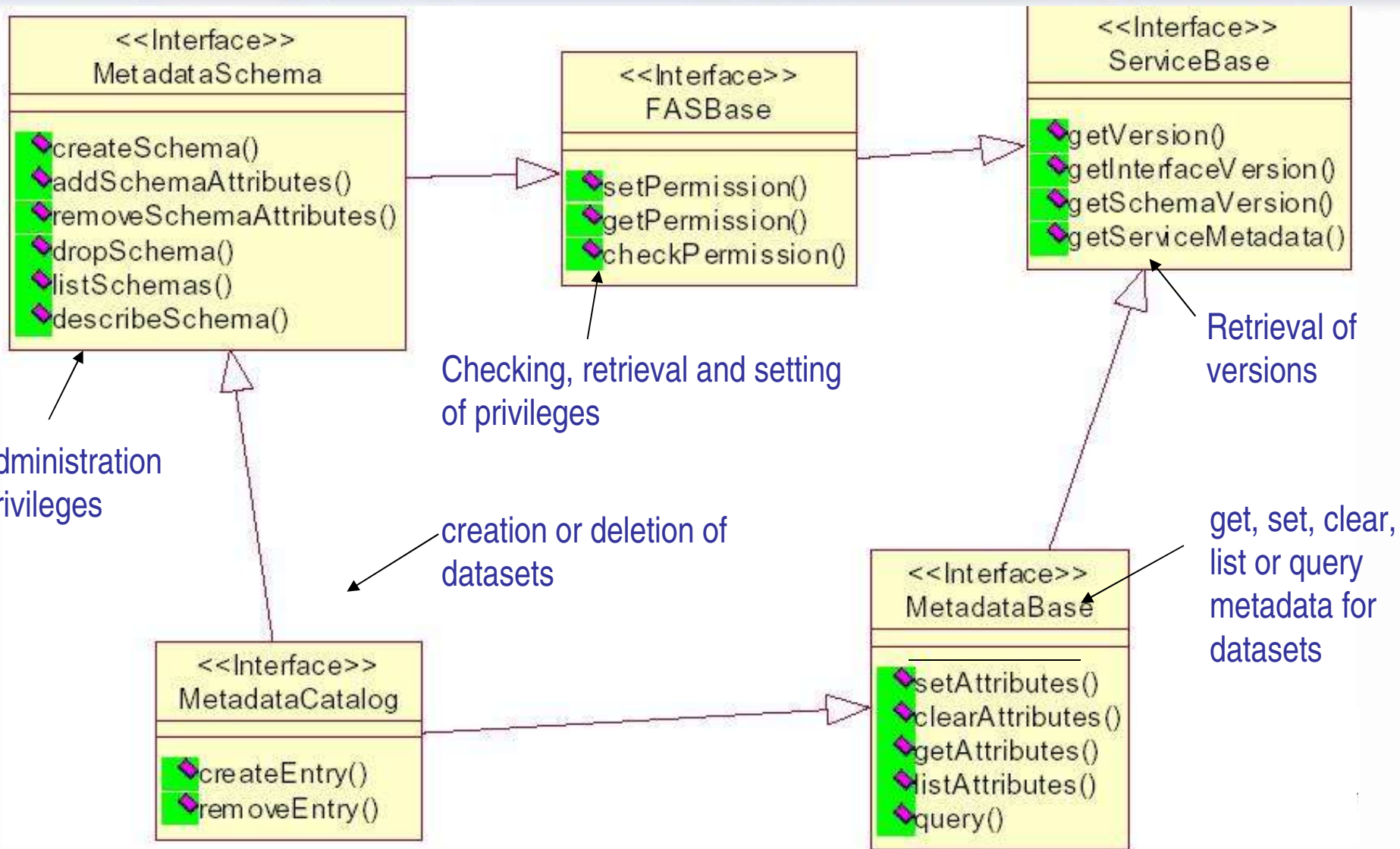
- A controller class directly interfaces with the command classes in the middle layer and with the client tier (illustrated later)
- choose corresponding AMI command equivalent to the basic requirements of each of the gLite Interface methods
- There are three possibilities for the client:



- relevant client is Web Services client
- web services deployed on Tomcat with Axis framework plug-in



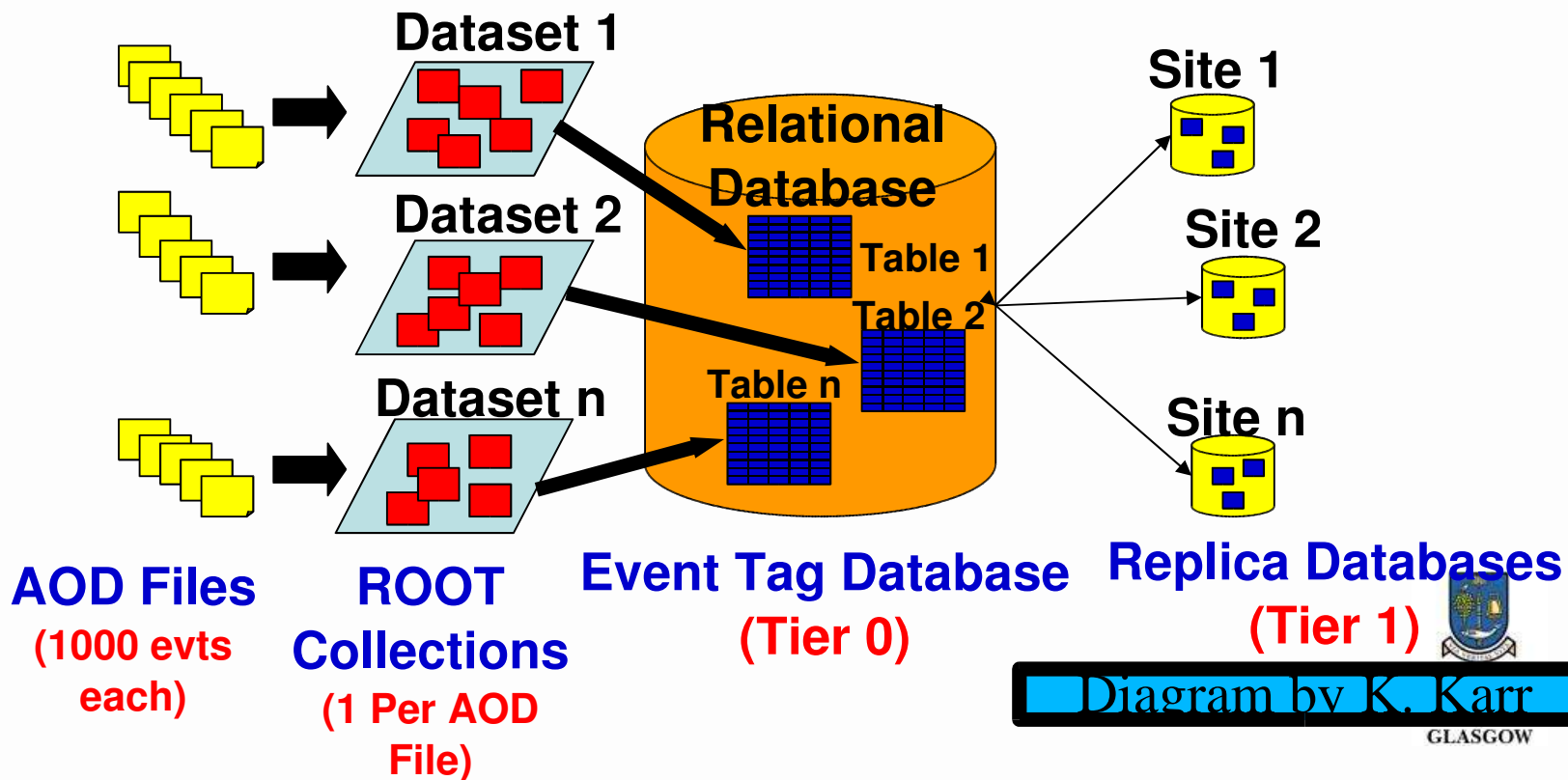
What are the gLite Metadata Interface methods?





ATLAS Event Level Metadata

- Event level metadata (Tags) used to allow efficient access to interesting data
- Written as ROOT files at AOD production then loaded into relational database at CERN and replicated to Tier 1 sites
- ~ 1kB/event





The proposed VOMS Groups and Roles that will be used in AMI are as follows:

Groups: det-indet, det-larg, det-muon, det-tile, gen-user, perf-egamma, perf-flavtag, perf-jets, perf-muons, perf-tau, phys-beauty, phys-exotics, phys-gener, phys-hi, phys-higgs, phys-lumin, phys-sm, phys-susy, phys-top, soft-prod, soft-test, soft-valid and trig-pesa

Roles: Editor (write access across group)

Writer (write access to own data only)

Reader (read access)

