

VOX Project Status

T. Levshina

Task List and Schedule for Virtual Organization and Related Work for USCMS vs. 1.0 2002-12-17 (CD Briefing - January 24, 2003)

VO Package 1: Registration of Users

- **Integration Phase (WBS 1.3.3.1, 1.3.3.4)**
 - Deliverable 1: (Jan 2003)
 - Registration Schema Definition for USCMS - **DONE**
 - Deliverable 2: (Jan 2003)
 - A USCMS VO membership service to hold information about members according to the above schema and their authorizations. - **IN PROGRESS**
 - Deliverable 3: (April 2003)
 - The USCMS VO secretariat will have to develop with the resource providers, the mechanisms and interfaces to register members with the various resource providers that require registration -**IN PROGRESS**

VO Package 1: Registration of Users

- **Production Phase (WBS 1.1, 1.2)**
 - Deliverable 1: (June 2003)
 - Ongoing support in place for the USCMS registration system, including some fractions of trained DBA and Web support. - **NOT PROVIDED**
 - Deliverable 2: (June 2003)
 - A secretariat in place which will be the public face of the USCMS registration system. Must maintain communication with international CMS and relevant institutional authorities (including Fermilab) to make sure that user information is up to date. - **NOT STARTED, OUT OF VOX SCOPE**

VO Package 2: Authentication

- **Integration Phase (WBS 1.3)**
 - **Deliverable 2: (Jan 2003)**
 - Deploy the KCA at Fermilab with appropriate attention to documentation for support. Document the maintenance requirements of the KCA, and maintain contacts with KCA providers for ongoing supports and upgrades. – **DONE (OUT OF SCOPE OF VOX)**
 - **Deliverable 3: (January/February 2003)**
 - Pilot deployment of KCA with VOMS on the development grid testbed. We should be able to authenticate users with KCA both onsite and offsite FNAL and authorize them to submit jobs from onsite and offsite to run on FNAL resources. - **DONE**
 - Deliverable 5: (April 2003)
 - An interface or process by which a member of the Virtual Organization outside of Fermilab can obtain KCA certification in a timely manner. -**IN PROGRESS**

CD Briefing - January 24, 2003 (II)

VO Package 3: Authorization

• Integration Phase (WBS 1.3)

– Deliverable 1: (Jan 2003)

- Deploy the EDG gatekeeper (with LCAS hooks) and provide LCAS modules to implement:
- a) enforcement of VO policy (use VOMS contents of proxy) -NOT DONE
- b) enforcement of resource provider policy (site authorization callout) - DONE
- c) enforcement of resource policy (locally permitted users) - DONE
- d) SAS modules need to be written – DONE

– Deliverable 2: (January/February 2003)

- Pilot deployment of KCA with VOMS on the development grid testbed. - DONE

– Deliverable 4: (Jan 2003)

- Maintenance interfaces are needed for the appropriate people and roles to update/access the VOMS information, integrated with registration interfaces. – IN PROGRESS

– Deliverable 5: (April 2003)

- Redundancy, and distribution of VOMS system design needs to be done. VOMS will have to be a high availability service for the collaboration's Grid. – NOT DONE

VO Package 3: Authorization

• Production Phase (WBS 1.1, 1.2)

– Deliverable 2: (June 2003)

- Final LCAS modules would replace the pilot ones used to facilitate the installation of the EDG gatekeeper. – LRAS, SAZ are using Globus 2.4 callouts

Goals, team and collaborators

- Purpose:

To facilitate the remote participation of US based physicists in effective and timely analysis of data from the LHC experiments during DC04 by designing, developing, and deploying Virtual Organization Management Service eXtension (VOX) for US CMS

- Stakeholders:

- US CMS (L. Bauerdick)
- Fermilab Security Team (D. Skow)
- iVDGL (R. Gardner)

- Team:

- T.Levshina – CD/CCF
- V. Sekhri – SDSS/iVDGL
- Y. Wu – CD/CMS
- L. Grundhoefer – iVDGL

- Collaborators

- BNL – VOMRS architecture, registration process, common interfaces - R. Baker, D. Yu
- EDG – VOMS – V. Ciaschini, A. Frohner
- VDT (U of Wisconsin) - ongoing communication and agreements with Globus.

Any Problems?

New kind of project:

- international collaboration
 - multiple stakeholders
 - everybody has a clear understanding of the outcome of the project:
Registration service that holds users' personal data as well as some information about VO itself and automated registration process with grid resources
 - but very vague understanding about:
registration work flow,
substance of personal data and access to it
hierarchy of administrators and their interaction
 - Not only the requirements are not fully specified but it is unclear who can specify them
- Lack of people interesting in discussion of lower level design
- CMS is pushing for the project, but it seems nobody from CMS is interested to participate in technical discussions about requirements, design etc
- Lack of man power to do development (see next slide)
- Lack of skills required for this project (*very good for personal growth!*)
- familiarity with Grid/Globus environment
 - Java
 - Web services
 - Servlets
 - Database

Scope of VOX Project (I)

- Provide US CMS registration service that
 - allows single access to registration with US CMS VO
 - facilitates, negotiates and monitors the process of member's acceptance to grid resources
 - provides centralized storage of
 - members DNs and their personal data
 - US CMS VO institutions and their representatives
 - US CMS VO affiliated grid resource administrators
 - provides means to query this information
 - performs authentication and authorization of VO members based on member's proxy, group, role and status and generated VOMS extended proxy (core VOMS service)

Scope of VOX Project (II)

- Local Resource Authorization Service (LRAS) – automates and facilitates the process of managing fine grain access to local grid element
 - provides storage of subset of VO membership information and mapping to local accounts
- LCAS callouts (in agreement with standard agreed by Globus, EDG, FNAL, and Virginia Tech)
- Site authentication and authorization service (SAZ)
 - allows security authorities of the local site to control access to the site
 - provides storage of all grid users that can potentially use the site resource
 - provides means to retrieve the information about users and their access

Current Status of the Project (summary)

Module Name	Original Milestone	Current Status (% of completion)	Estimated Deployment Time	Comments
VOMRS	July 2003	30%	End of September (?)	Requirements are still not finalized, Database schema is not reviewed, not enough developer (<1 FTE)
LRAS	January 2003	70%	Mid September	More requirements were added that demand some changes in database schema and some code changes (Who will be working on this?)
LCAS plugins	April 2003	80%	End of September	Timeslot plugin and plugin to extract information from extended proxy need to be implemented (Who will be working on this?)
SAZ	June 2003	90%	Deployment version (alpha) was given to Security team	New requirements were added last month, documentation is needed (Who will be working on this?)
VOMS EDG/DATA TAG installation/testing	February 2003	100%	Deployed (part of DPE)	VOMS core and admin services are installed, build and testes on CMS testbed. Will be part of Grid3 ran. VOMS admin is interim solution while VOX is not ready. Will try to work with VOM admin developers to come up with common solution

Efforts and current responsibilities (I)

T.Levshina – CD/CCF - (55%) – project leader

high level software design for VOX project

low level design and implementation of VOMRS server

VOMRS database schema design and deployment

VOMRS API, client

writing documentation

working with collaborators

V. Sekhri – SDSS/iVDGL - (65 %) - 0% between now and August 13th

50% -afterwards if CD provides funding

SAZ redesign and reimplementaion

LRAS, LAMS low level design and implementation

Common authentication classes based on Java COG

LCAS plugins

working with collaborators on Globus gatekeeper callouts

writing documentation

Efforts and current responsibilities (II)

Y. Wu – CD/CMS - (30%)

VOMS-EDG (Proxy Server) installation, building and testing

VOMS-EDG client (voms-proxy-init) installation, building and testing

VOMS-EDG client code modification for DOESG and KCA certificates acceptance

bug reports/ close collaboration with voms developers

VOMS-EDG admin installation testing

packaging VOMS for DPE

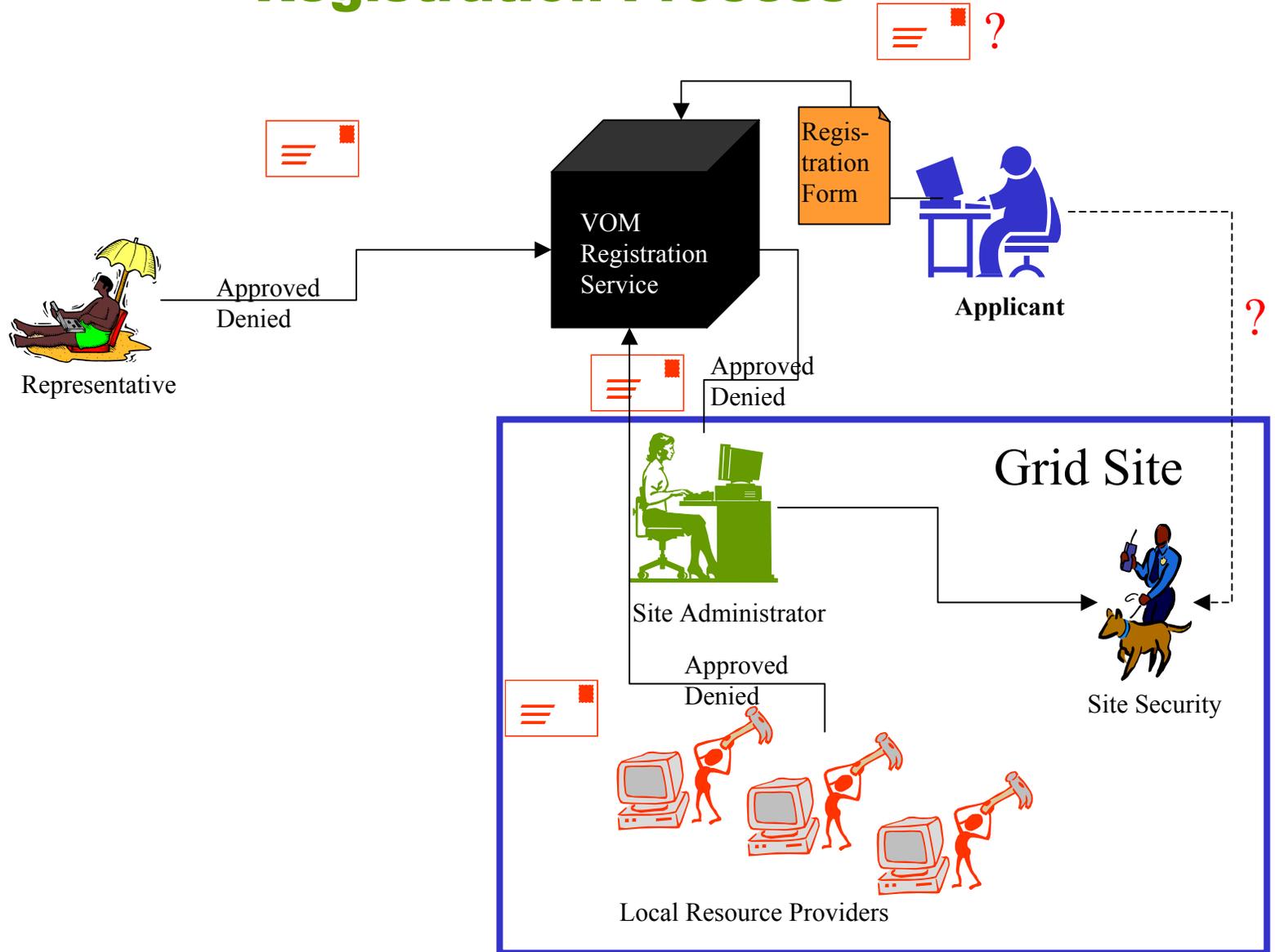
L. Grundhoefer – iVDGL (40 %)

VOMS-EDG (Proxy Server) installation, building and testing

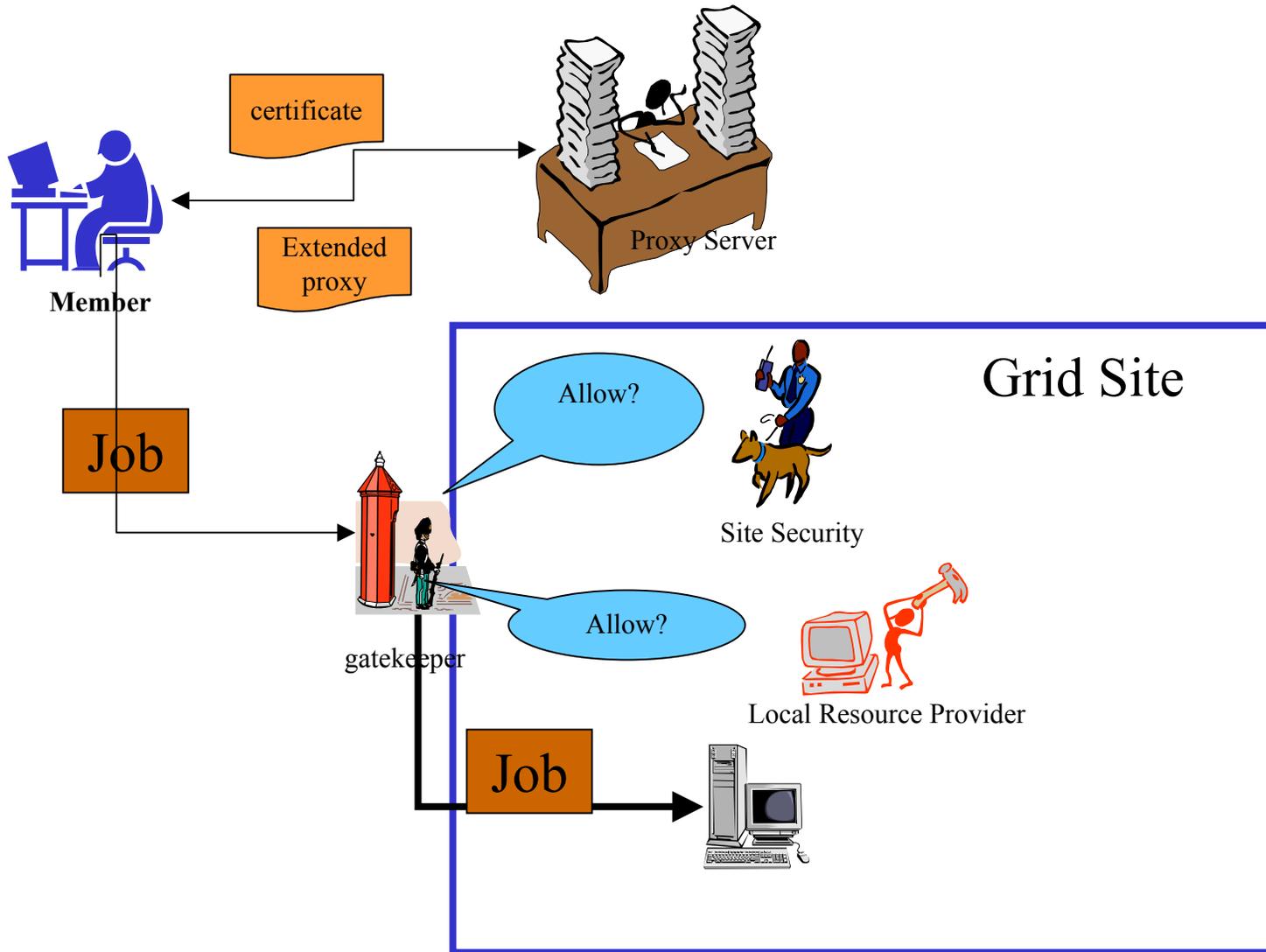
VOMS-EDG client (voms-proxy-init) installation, building and testing

bug reports to voms developers

Registration Process



Job Submission Process

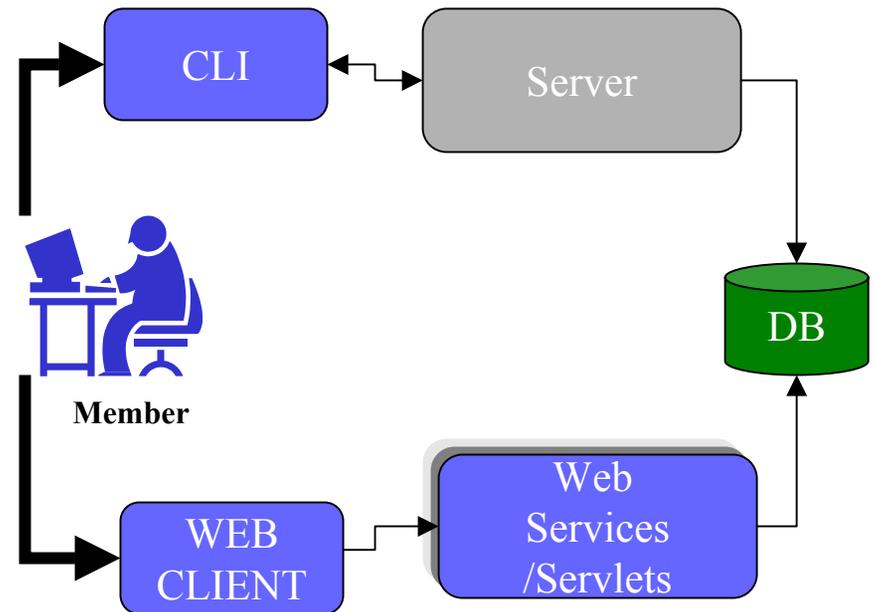


VOM Registration Service

The service provides means for member registration and coordination of registration procedure among various VO and grid administrators.

It consists of:

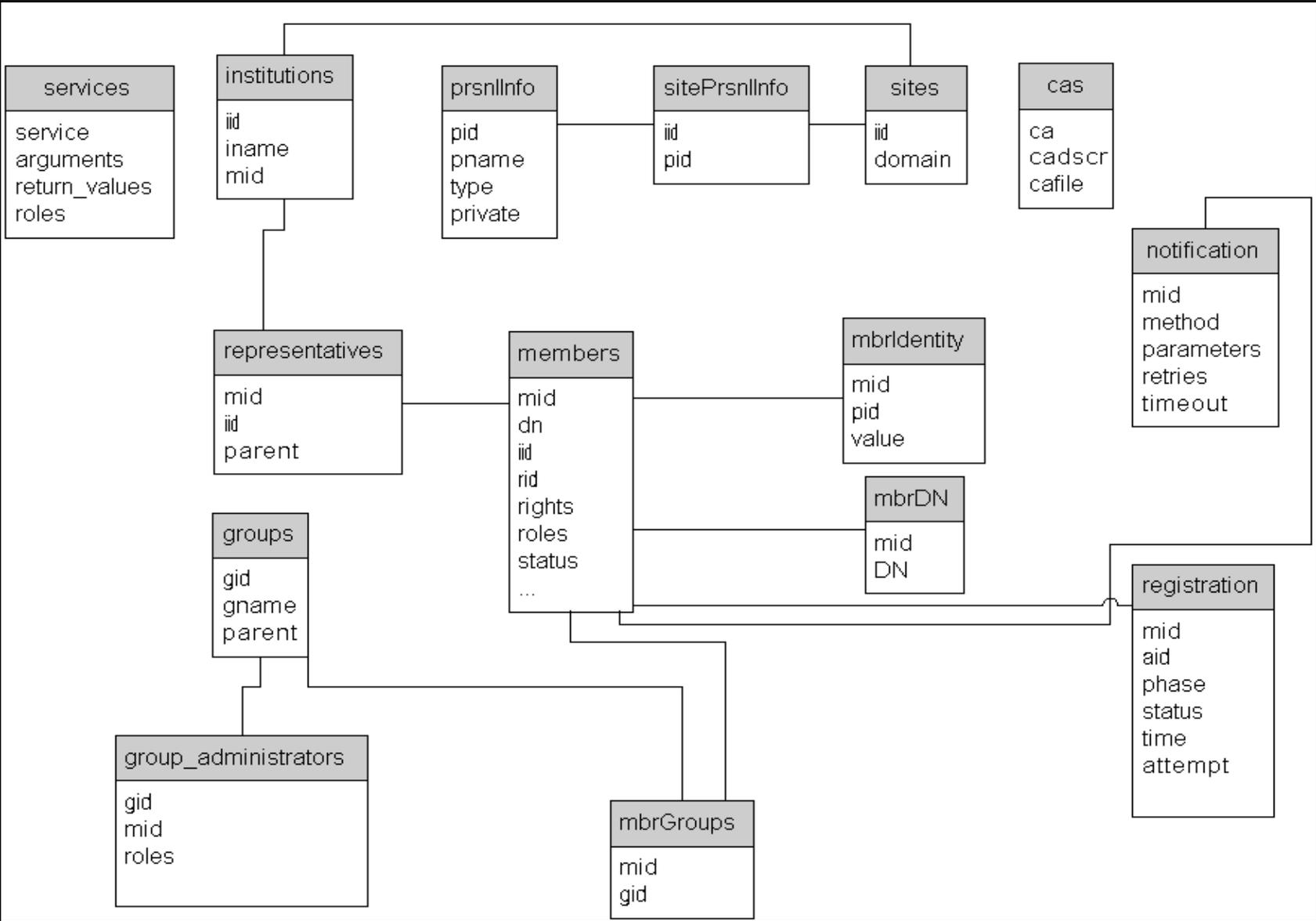
- VOM Registration Server
- Web services
- Web GUI/Servlets
- Command line interface
- Registration database



Member roles in VO

- There are eight distinguished roles within VO:
 - VO Admin
 - Institutional Representative
 - Site Administrator
 - Local Resource Provider
 - Group Owner
 - Group Manger
 - Member
 - Applicant
- A VO Member can have one or more roles.
- A VO Member can perform a set of predefined actions that correspond to his roles.
- Members with particular roles (Institutional Representatives, Group Owner, Group Manager) are organized in hierarchies. The hierarchical standing limits the set of actions that a member is entitled to perform within the VO.
- The other limitation is defined by the members' affiliation with a particular institution, group etc

Database Schema



Client/Server Protocol

VOMR Server exchanges messages with clients. The message client send has the following format:

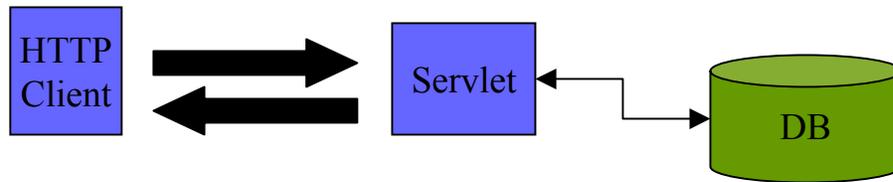
SERVICE=service_name [PARAMETER=value ...]

The message the client receives back has the following format:

***STATUS={OK,ERROR}
MESSAGE={error_message,
warning_message, ...}
[PARAMETER=value...]***

```
bash-2.03$ java fnal/vox/vomrs/MyClient  
hotdog62 14000 "getInstitutions"  
STATUS=OK MESSAGE=SUCCESS  
INSTITUTION=Boston University  
INSTITUTION=California Institute of Technology  
INSTITUTION=Carnegie Mellon University  
INSTITUTION=Fairfield University, Fairfield  
INSTITUTION=Fermi National Accelerator Laboratory  
INSTITUTION=Florida Institute of Technology  
INSTITUTION=Florida International University  
INSTITUTION=Florida State University  
INSTITUTION=Iowa State University  
INSTITUTION=Johns Hopkins University  
INSTITUTION=Kansas State University  
INSTITUTION=Massachusetts Institute of  
Technology  
INSTITUTION=Northeastern University  
...  
bash-2.03$ java fnal/vox/vomrs/MyClient  
hotdog62 14000 "doSomething"  
STATUS=ERROR MESSAGE=AuthorizationError:  
service doSomething is unknown
```

Web GUI example



First example:

<http://ngop.fnal.gov:8080/axis/vox/html/registration.html>

VOM Registration Service

(Status of component specification and code development)

- VOMS DB:
 - Design is done
 - Schema is deployed
 - mysql
 - 17 tables
 - Would like to have a thorough review from DB developers
- VOM Registration Server, Registration API
 - Design is done
 - Prototype is deployed
 - New requirements are emerging all the time. Interactive process!
- VOM Registration Client
 - Developed primitive client to exercise server functionality
 - Started requirements collection (No clear understanding who should provide them)
- Web GUI
 - Work has been just started
 - Again: no requirements, we have to invent them!

Site authentication and authorization service (SAZ)

SAZ consists of the following components:

- SAZServer (GSS/GSI)
 - Handles new site users
 - Performs user authorization
 - by verifying user access status in SAZDB
 - by analyzing user certificate chain
- SAZDB stores user's principal, dn, status etc
- SAZClient (GSI/GSI)
 - invoked as LCAS plugins to communicate with SAZ server to check user authorization
- Admin Server (GSS/Kerberos)
 - allows admin to add,delete and list any DN and principal in SAZDB
 - user to add,delete or list any DN associated with his own principal in SAZDB
- AI/UIClient(GSI/Kerberos)
 - provides front end for the admin/user to do operation on SAZDB

Local Resource Administration Service

Local Resource Administration Service allows local resource administrators to manage fine grain access control to local resources. It perform the following actions:

- associates VO member with the local account based on the user groups
- can create local account(s) on the appropriate grid cluster(s)
- can notify VO Registration Server that local site is ready for user to submit jobs.

LRAS consists of following components:

- LRA Server that communicates with VOMRS. It acquires information about new VO member and set member registration status.
- LRA DB that stores mapping between VO members and their local accounts, the member's current access status to the local resources,etc
- LRA Admin that allows to modified information stored in LRA Database
- LAM Server/Client that automatically creates user account on the local cluster and update gridmap file

LRA Admin example

LRA Client

Info Account

Fields

DN
ID
Allow(Y,N)
Deny(Y,N)
Time From
Time To
New
Role
Group

Clause

AND
 OR

Action

Query Clear
Save Exit

Result

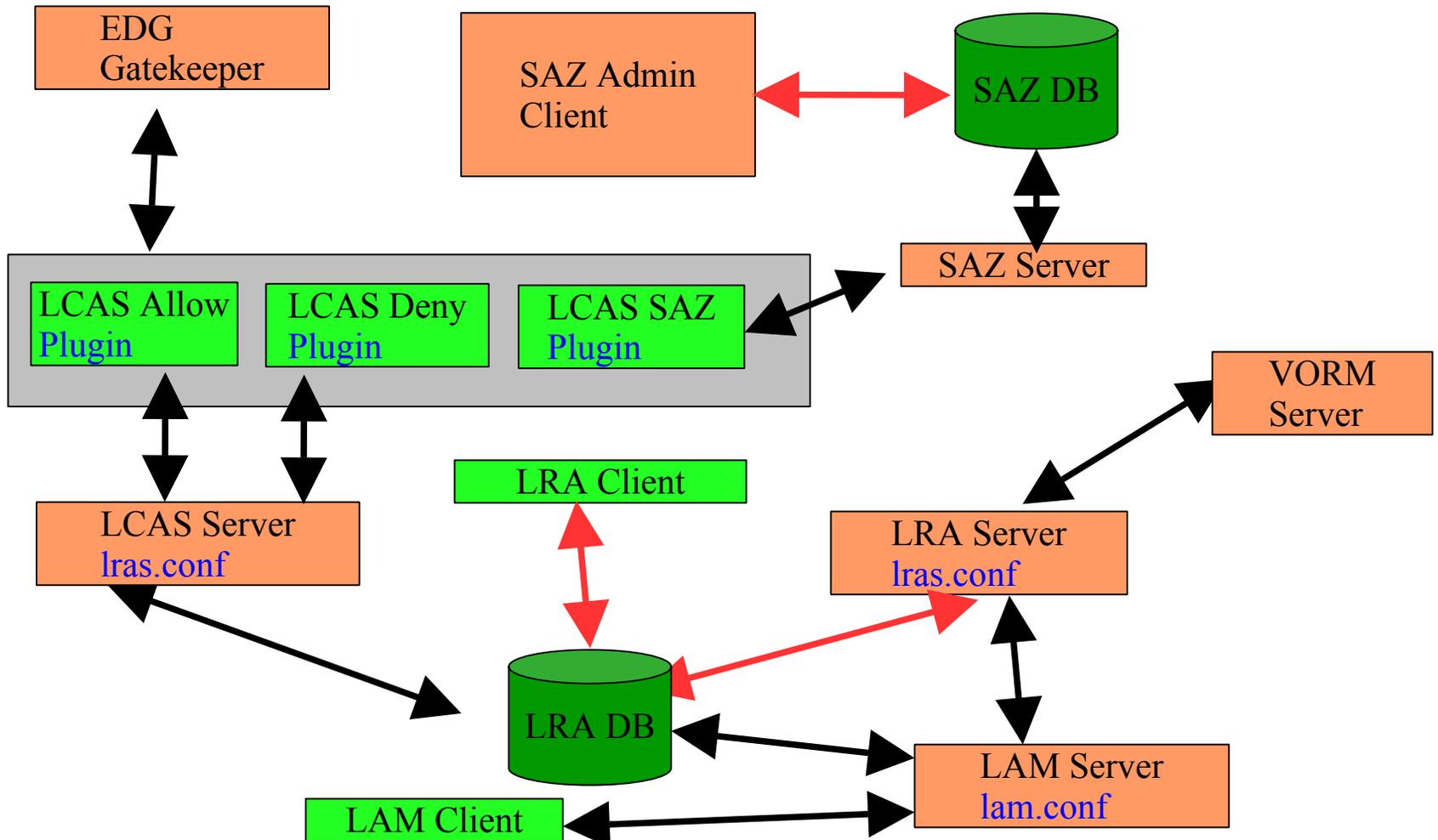
DN	ID	Allow	Deny	Time From	Time To	New	Role	Group
/C=US/ST...	annis	N	N	0000-00...	0000-00...	YNY	user	sdss
/C=US/ST...	sekhri	Y	N	0000-00...	0000-00...	YNN	user	sdss
/DC=gov/...	sekhri	N	N	0000-00...	0000-00...	YNY	user	sdss
DC=gov/...	tanya	Y	N	0000-00...	0000-00...	YNY	user	sdss
DC=gov/...	vjay	Y	N	0000-00...	0000-00...	YNY	user	sdss
DC=gov/...	vjay	Y	N	0000-00...	0000-00...	YNY	user	sdss

LRAS

(Status of component specification and code development)

- LRAS Server, API and Client (GUI)
 - Design is done
 - Coding is done
 - (* needs some modification due to new requirements coming from local grid SE)
 - Had Code review
- LRAS DB
 - Design is done
 - Schema is deployed
- LAMS (Local Admin Manager Server responsible for updating gridmap file and optionally creating user account on grid cluster)
 - Design is done
 - Coding is done
- LAMS GUI Client
 - Design is done
 - Coding is done

Job submission on the local grid cluster



LCAS Plug-ins and SAZ

(Status of component specification and code development)

- LCAS Plug-ins:
 - Allow/Deny plug-in checks with LRAS if user has access to local grid cluster **Done**
 - Timeslot plug-in check with LRAS if user is allowed to run during current timeslot **Not started**
 - SAZ plug-in check with SAZ server if user has been authorized to use grid cluster
 - **Done**
- ANAM (Authorization and Authentication Module) –based upon GSS library comes from Java Cog (version cog-1.1a)
 - **Design is done**
 - **Coding is done**
- SAZ
 - **Alpha version is released , testing will be done by Security group.**
 - **Some crucial features still need to be implemented**

VOMS Core And Admin Services (DATA Grid and Data Tag)

- VOMS core services - lead developer Vincenzo Ciaschini
 - Installed, built, tested , part of upcoming UCCMS DPE
 - Modifications have been made to accept KCA and DOESG certificate
 - Will be used by VOX project to generate extended proxy
- VOMS admin service - lead developer Frohner Akos
 - Installed, built , tested part of upcoming USCMS DPE
 - Will be used as interim solution while VOX is not ready
 - The discussion is underway to find the way to join VOX and VOMS admin service

What needs to be done

- VOMRS
 - Understand the requirements related to “deletion/modification” of any of VO objects:
 - member transfer from one institution to another
 - spokesperson resignation
 - side drops out from VO resource provider
 - Database schema evaluation
 - Admin scripts that initially create and populate database and back it up
 - Database API (for web services and VOM registration server) should be reviewed
 - Security issues related to database
 - Collect requirements for WEB GUI
 - Implement Web GUI
- LRAS
 - Integrate LRAS and VOMRS
 - Handle group accounts
- SAZ
 - Code needs to be cleaned up, appropriate exception handling and logging should be implemented.
 - There are several remaining features that should be implemented and fully tested (e.g. certificate authorities chain verification, database replication...)

Summary

- It is a real challenge:
 - Requirements are not fully defined and it is unclear who is responsible for defining them
 - The scope of the project is constantly increasing
 - There are a lot of collaborators with a strong opinion, it is a complicated task to find the compromise
- New technology (at least for some members of the group):
 - GRID
 - GLOBUS
 - JAVA
 - Security
 - Web services
- The resources allocated to carry out this project are not sufficient