# VOMS Issues

SCG Meeting

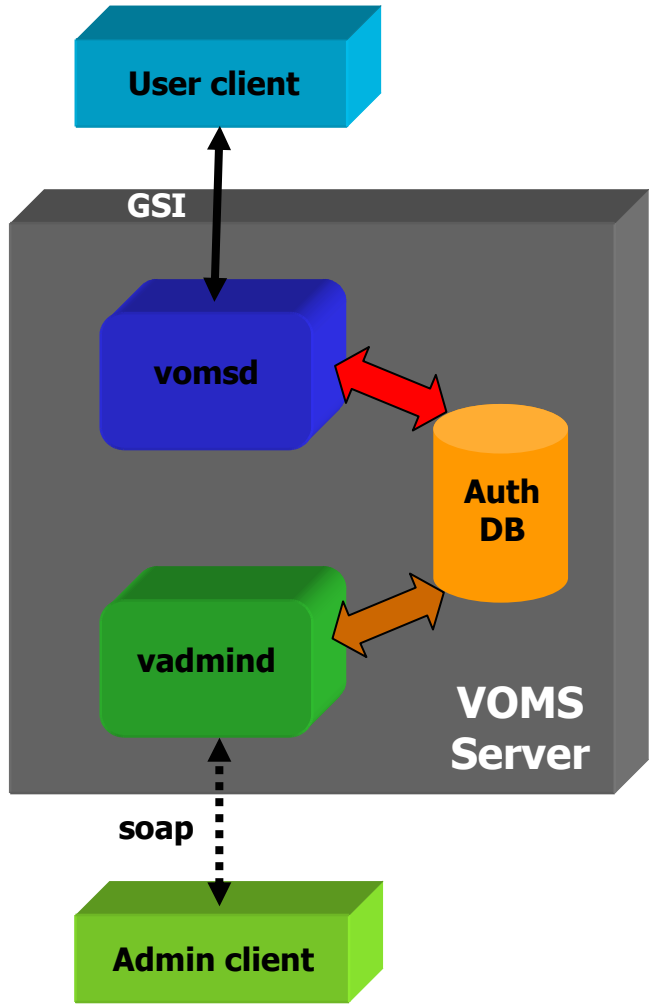
CERN, November 18-19 2002

# VOMS Entities

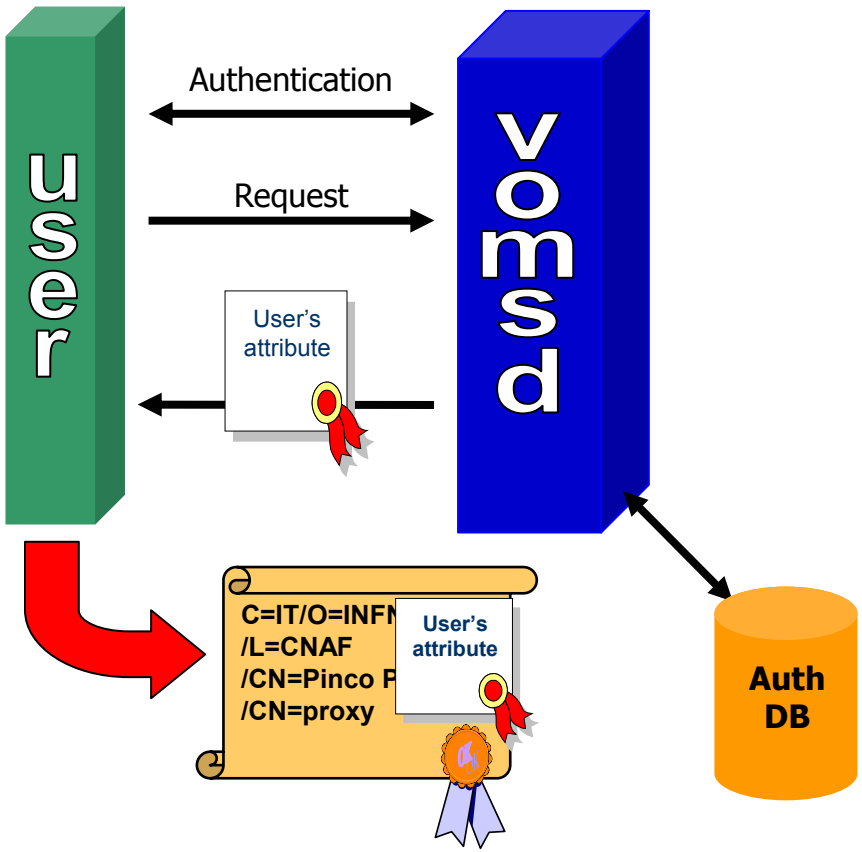
- User
- Group
  - collection of users
- Role
- Capability
  - credential that explicitly grants access rights (free-form string)
- Query
  - for personalization by the VO
- Administrator
- ACL
  - controls the operations of the Administrators
- Certification Authority

# VOMS Components

- User Client
  - obtains user's attributes
    - voms-proxy-init command
- VOMS Server
  - Client Server
    - C
  - Admin Server
    - Java
  - Database
    - MySQL
- Admin Client
  - manages VOMS entities
  - GUI & CLI
    - Java



# User Client Operations



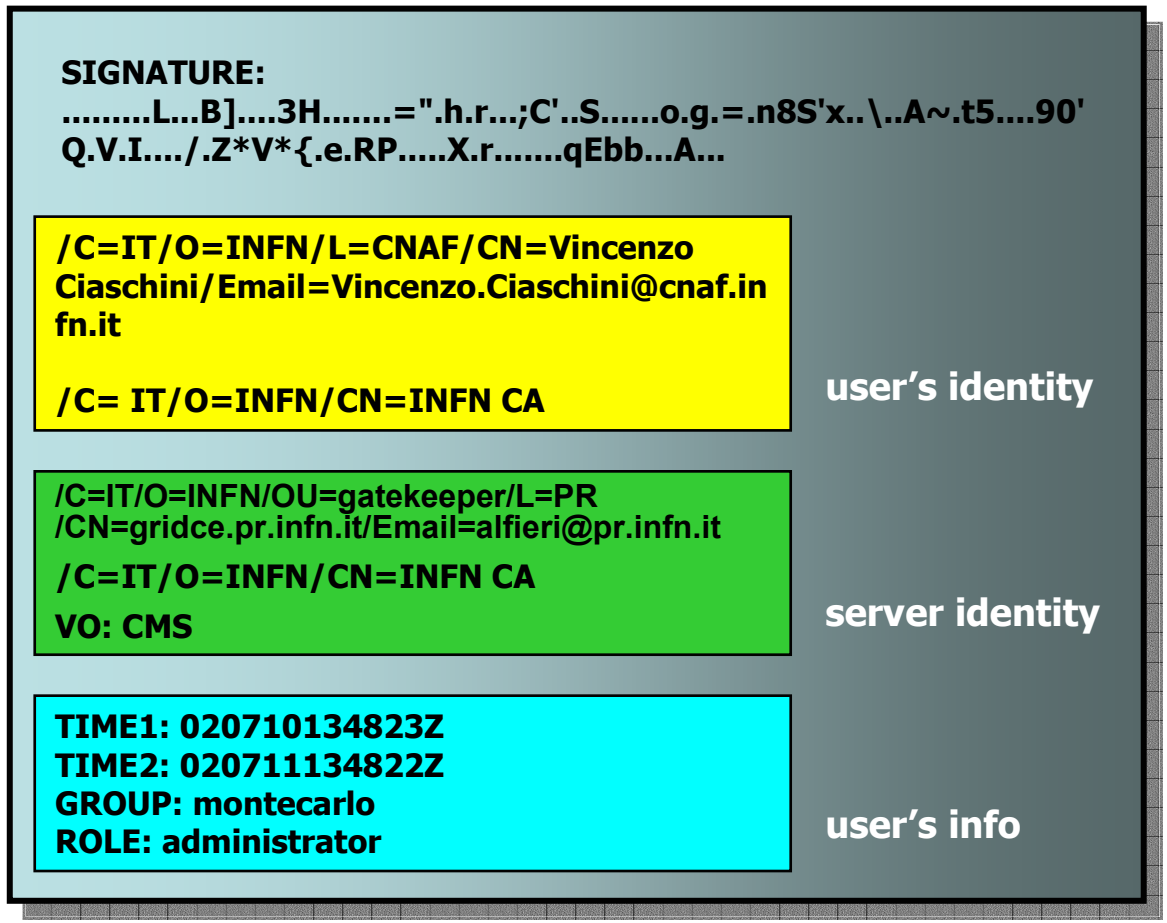

1. Mutual authentication Client-Server
   - Secure communication channel via standard Globus API
2. Client sends request to Server
3. Server checks correctness of request
4. Server sends back the required info, signed by itself
5. Client repeats process for other VOMS's
6. Client creates proxy certificates containing all the info received into a (non critical) extension

# `voms-proxy-init` Options

- All the queries have an implicit <*userid*> field, derived from the user's certificate.
  - **A**: all info regarding the user;
  - **G <*group*>**: user is member of **<*group*>;**
  - **R <*role*>**: user has role **<*role*>**;
  - **B <*group*>:<*role*>**: user is member of **<*group*>** with role **<*role*>**;
  - **L**: lists all available queries;
  - **S <*qid*>**: executes the query **<*qid*>**.

# User's Attributes Info

- Inserted in a non-critical extension of the user's proxy
  - OID: 1.3.6.1.4.1.8005.100.100.1
- One for each VOMS Server contacted.



# VOMS Administration

- Admin Server routines:
  - **Core services**: basic functionality;
  - **Admin interface**: methods to administrate the database;
  - **History**: "going back in time" functionality.
- Traceability
  - every table has a corresponding "archive" table;
  - every table has a pair of columns:
    - *createdBy:* the id of the requester of the operation that created this record;
    - *createdSerial*: a database-wide unique, ordered serial number that identifies this exact operation (it is a transaction id);
  - rows are never deleted or modified: they are moved to the corresponding archive table.
  - archive tables have the same scheme as data tables, plus:
    - *deletedBy:* the requester of the operation that expired the row;
    - *deletedSerial:* the transaction number of the operation.
  - The server can query the state of the database at any given time or transaction number.

# "Strong Authentication" Querelle

- **`voms-proxy-init -include`**
  - inserts a specific file into the user's proxy (OID: 1.3.6.1.4.1.8005.100.100.2)
  - the user inserts into his proxy (signed by his "regular" certificate) the KCA certificate
  - the KCA certificate is checked by the LCAS plugin at the site which requires the additional authorization
- **aVOMS** (Authenticating VOMS) proposal (by Andrew Hanushevsky, Stanford)
  1. The user obtains a site-specific authenticator for their domain account (e.g. Kerberos ticket via afs klog, dce_login, etc.).
  2. Using the **`avoms-proxy-init`** command, the user requests a signed certificate from the aVOMS (the request includes the authenticator).
  3. The aVOMS, once the user is authenticated, retrieves the user's requested certificate, *signs it with the user's stored private key*, appends a *signed list of valid virtual organization* [sic], and sends it back to the user.
  4. The user generates a public/private key pair in the standard way.
  5. The user generates a proxy certificate to be used on the grid.

# Intermezzo

- Support for multiple VO's
  - the Subject of the user's proxy contains the VO name
    - **`mkgridmap --vo`**
    - **`grid-proxy-init --vo`**
    - **`voms-proxy-init --vo`**
  - a patched **libglobus_ssl_utils** must be installed on every farm that wants to accept the new proxies, on the RB and II (?)
  - waiting for support from the Condor Team
- Coexistence of VOMS and VO LDAP servers.
  - **`mkgridmap++`**
    - produces *grid-mapfiles* from both LDAP and VOMS servers;
    - new directive in the config file
    - authenticated access to VOMS (*not LDAP*) servers
      - restricts the clients allowed to download the list of the VO members

# Future Developments

- US Grid Projects "strongly considering" VOMS & LCAS
  - a VOMS server installed at Fermilab.
- VOMS entities revised according to "true life" VO management experience.
- VOMS certificates will be Attribute Certificates (RFC3281).
- User Client & VOMS Client Server rewritten in C++.
- Subgroups.
- Replication.
- VOMS certificates with more sofisticated time validity.