# Table of Contents

# gLite Virtual Organisation Membership system

## Functional description

VOMS is a system to classify users that are part of a Virtual Organization (VO) on the base of a set of attributes that will be granted to them upon request and to include that information inside Globus-compatible proxy certificates.

VOMS consists of two main components:

- VOMS - includes the VOMS server and the VOMS client tools and APIs (e.g. voms-proxy-init)
- VOMS Admin - a Java server application (and UI servlet) used to manage users and their privileges for a VO

**Released version**

Official VOMS stable version :1.9 Official VOMS Admin stable version : 2.5

## Daemons running

The following daemons need to be running:

- tomcat5
- edg-voms
- mysql (in case of MySQL is running directly on the VOMS server)

## Init scripts and options (start|stop|restart|...)

- /etc/init.d/gLite (start|stop|restart)

## Configuration files location with example or template

The configuration files for the VOMS service are located in:

- /opt/glite/etc/voms/

## Logfile locations (and management) and other useful audit information

The gLite log files can be found under

- /var/log/glite/
- /var/log/tomcat5/

## Open ports

Open ports are 8443 for voms-admin and a series of configurable ports (typically starting with the default 15000) for the voms server instances.

## Possible unit test of the service

voms comes with a testsuite that can be run through ETICS

## Where is service state held (and can it be rebuilt)

There is no significant service state associated with VOMS. voms-admin only has state when a registration request is being processed, and such state is kept in the DB.

## Cron jobs

The cron jobs can be found in:

- /etc/cron.d/

and are:

- /cron.d/glite-fetch-crl.cron
- /cron.d/ccm-purge.cron
- /cron.d/ccm-fetch.cron

## Security information

### Access control Mechanism description (authentication & authorization)

This node type has two interfaces. One for the administration where VO admins can add/remove users and assign VO Roles and a second one where the middleware applications ask for proxy signature. On both interfaces the authentication part is done via x509 authentication against the trusted CAs that are installed at the node. The authorization part is done via the VO roles that are assigned to the uses's DN.

### How to block/ban a user

There is no way to block users from accessing the administration interface for view only (anonymous read access is actually required by some middleware components). Write access at the administration interface is limited to the VO Admins per VO. Removing the VO admin role will block user from future write accesses. The access to proxy signature interface is limited to the users that are listed as members to the VO. Removing a user from the VO will block his/her access.

### Network Usage

Three services are running that need network access on this node-type.

- the MySQL server service. The server binds to the 3306/tcp port. Alternatively, Oracle may be used, which is usually run on a different node. Access to this node should be allowed.
- the VOMS-Admin webapp on a TomCat server at the 8443/tcp port.
- the edg-voms server which binds to one tcp port per VO (usually something like 15010/tcp)

### Firewall configuration

The proposed firewall configuration is to deny access to anyhost/anyport and allow:

- 8443/tcp from everywhere (this is used for VO management (via x509 authentication) and gridmapfile creation)
- Any edg-voms server configured port (i.e. 15010/tcp) from everywhere (this is used by users directly (from UIs or WNs) or indirectly (from WMSes))
- Any other administration required port for the administration subnet/interface (i.e. 22/tcp (ssh))

### Security recommendations

No stipulation.

### Security incompatibilities

No stipulation.

### List of externals (packages are NOT maintained by Red Hat or by gLite)

#### Maintained by JPackage repository maintainers

- bcel
- bea-stax
- bea-stax-api
- bouncycastle
- dom4j
- ecj
- geronimo-j2ee-1.4-apis
- geronimo-jaf-1.0.2-api
- geronimo-javamail-1.4-api
- geronimo-specs-poms
- geronimo-stax-1.0-api
- glassfish-jaf
- glassfish-jaxb
- icu4j
- isorelax
- jakarta-commons-beanutils
- jakarta-commons-collections
- jakarta-commons-collections-tomcat5
- jakarta-commons-daemon
- jakarta-commons-dbcp-tomcat5
- jakarta-commons-digester
- jakarta-commons-el
- jakarta-commons-launcher
- jakarta-commons-logging
- jakarta-commons-modeler
- jakarta-commons-pool-tomcat5
- jaxen
- jdom
- log4j
- msv
- msv-xsdlib
- mx4j
- regexp
- relaxngDatatype
- saxon
- tomcat5
- tomcat5-common-lib
- tomcat5-jasper
- tomcat5-jsp-2.0-api
- tomcat5-server-lib
- tomcat5-servlet-2.4-api
- ws-jaxme
- xalan-j2

- xerces-j2
- xml-commons
- xml-commons-jaxp-1.2-apis
- xml-commons-jaxp-1.3-apis
- xml-commons-resolver11
- xom
- xpp2
- xpp3

#### Maintained by DAG repository maintainer

- perl-Crypt-SSLeay
- perl-DBI
- perl-HTML-Parser
- perl-HTML-Tagset
- perl-IO-Socket-SSL
- perl-LDAP
- perl-Net-Daemon
- perl-Net-SSLeay
- perl-PlRPC
- perl-TermReadKey
- perl-XML-DOM
- perl-XML-NamespaceSupport
- perl-XML-RegExp
- perl-XML-SAX

### Other security relevant comments

This node-type should not be collocated with any other node-type and should not allow shell access to users. Any connection other than the ones described above should be treated as suspicious.

## Utility scripts

- voms-admin (server side)
- voms-proxy-* (client side)

## Location of reference documentation for users

- VOMS FAQ for service managers
- FAQ for management
- VOMS Install Guide
- VOMS dicumentation directoy from VDT
- The VOMS official user guide
- The VOMS-Admin official user guide

## Location of reference documentation for administrators

See user documentation.

---

This topic: EGEE > GLiteVOMS
Topic revision: r15 - 2011-01-14 - unknown