Configuration of Virtual Organizations in gLite 3.0

ADMINISTRATOR'S GUIDE

Robert Harakaly
EGEE – JRA1

20 January 2006

The information contained in this document represents the views of EGEE as of the date they are published. EGEE does not guarantee that any information contained herein is errorfree, or up to date.

EGEE MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, BY PUBLISHING THIS DOCUMENT.

CONTENT

1 INTRODUCTION

1.1 PURPOSE

This document provides a detailed description of the so called VO management feature implemented in the gLite configuration system. It provides the implementation description, use-cases and extensive examples for any gLite middleware administrator doing modifications in the default VO management configuration provided in the release.

1.2 APPLICATION AREA

The document contains VO management implementation details useful for understanding of the VO management functionality and also hints and examples for advanced and expert configuration.

Document assumes good knowledge of the gLite configuration model (configuration procedure, schema, etc.) and basic knowledge of XML.

1.3 REFERENCES

[This subsection provides a complete list of all documents referenced elsewhere in the document.]

[R1]	
[R2]	

1.4 TERMINOLOGY

Glossary

VO	Virtual organization
XML	eXtensible Markup Language

Definitions

2 INTRODUCTION

Starting from gLite 1.5.0, gLite provides a new functionality in the gLite middleware configuration to perform VO configuration management operations. The new functionality enables to group any virtual organization related information, previously dispersed throughout the gLite configuration files, in one place and later use this information from different locations in the gLite configuration.

This approach enables to simplify the management of this type of information and facilitates automatic reconfiguration of the services when VOs are added, modified or removed. It also prevents duplication in the parameter definition. As an example, the VOMS related parameters voms.host and voms.port for given VO can be listed. Without use of the VO management approach these parameters would need to be defined in several configuration files. As a consequence after the change of the VOMS server name or TCP port, these parameters should be found and changed in several places/files, which considerably increases the risk of misconfiguration.

This feature also enables to centralize and simplify the standard security setup like gridmap file creation, pool user account creation, etc. and enforce common settings for all gLite nodes and services.

The document is split in two main parts. The first one describes the implementation of standard features and how to perform configuration of VO related information. The second part is dedicated to expert administrators. Both parts provide extensive 'Example' sections with configuration examples for common configuration scenarios.

3 IMPLEMENTATION

The implementation is based on the extension of the current gLite XML configuration schema and on additional background processing of the configuration files in order to keep the configuration file complexity as low as possible with the goal of minimizing changes in the existing configuration procedure. At the same time, as usual, great freedom is left in the configuration for expert administrators.

To implement the VO management in the gLite configuration, following features were added:

1. Centralized definition of VO related information
2. Node scoped inclusion and exclusion of VOs
3. Automatic VO based service instance generation

3.1 VO DEFINITION

VO related information can be defined by using of the `<vo/>` tag in the global or site configuration file or in a separate file, for example vo-list.cfg.xml. There must be one `<vo/>` tag per virtual organization.

The `<vo/>` tag is a child element of the root element of the XML configuration file. It has one attribute called 'name' defining the VO name. Parameters are grouped in the `<parameters/>` container, following the standard gLite schema. For example:

```
<siteconfig>
      <vo name="vo_name">
            <parameters>
                  ...
            </parameters>
      </vo>
      <vo name="vo2_name">
            <parameters>
                  ...
            </parameters>
      </vo>
      ...
      <node name="...">
            ....
      </node>
</siteconfig>
```

In case the local configuration is used the recommended location for VO definition is the glite-global.cfg.xml file:

```
<global>
      <vo name="vo_name">
            <parameters>
                  ...
            </parameters>
      </vo>
      <vo name="vo2_name">
            <parameters>
                  ...
            </parameters>
      </vo>
      ...
      <parameters>
            ...
      </parameters>
</global>
```

Note: any parameter grouping, usage of the xinclude mechanism, etc. is allowed until the resulting XML document will follow the standard gLite configuration file schema. See example in the Examples section.

3.1.1 VO list restriction

During the reading of the configuration file, every node is automatically automatically passed an element called volist containing by default the list of all defined VOs. This list can be modified by using the <volist/> tag inside a <node/> section in case of site configuration or in the local configuration file.

Example:

```
<siteconfig>
      <node name="node1">
            <volist>
                  <include name="<vo1Name>"/>
                  <include name="<vo2Name>"/>
            </volist>
      </node>
      <node name="node2">
            <volist>
                  <exclude name="<vo1Name>"/>
                  <exclude name="<vo2Name>"/>
            </volist>
      </node>
</siteconfig>
```

VO list restriction:

node1: *all VO related tasks and services will be configured only for the <vo1Name> and <vo2Name> virtual organizations.*

node2: *all VO related tasks and services will be configured for all defined VOs except the <vo1Name> and <vo2Name>.*

As a consequence, after adding a new VO and reconfiguring the node1 and node2 nodes, the configuration of node1 will not change while configuration of node2 will add additional (newly added) VO to its configuration.

Note: the tags <include> and <exclude> cannot be used in one <volist> tag at the same time.

3.2 VO RELATED TASKS

gLite release 1.5 and higher contains a reorganization of some of the VO related tasks operations as for example general security settings, gridmap file creation, pool accounts creation, etc. All these tasks fully depend on the VO definition described in previous section.

All these operations are defined in a dedicated file called gridSecurity.py installed with the glite-config package.

3.3 VO RELATED SERVICE INSTANCES

VO management allows simplified configuration for services starting one instance per virtual organization. Virtual organization definition described in Chapter 1 can be used for automatic declaration of VO based server instances. Definition of VO instances of service 'xyz' by using of VO related and global information can be done by using:

```
<instance service=”xyz” iterate=”volist”/>
```

This configuration line internally creates one instance per each VO referenced in the volist. Name of each instance created by using the 'iterate' attribute is equal to the corresponding VO name. All possibilities of volist restriction described in the chapter 1.1 are valid also for instances. All vo-related parameters needed by the service instances created in this way are automatically replaced with the correct values for each VO.

3.4 PARAMETER PRECEDENCE AND ADDITION

VO management implementation in gLite R3.0 has built-in extensive support for parameter modification and tuning ... It extends the standard gLite precedence schema where:

- local glite-global.cfg.xml has the **lowest precedence**
- site-config file
- local service configuration files have the **highest precedence**

At the same time in each of these files, internal precedence is defined:

1. vo definition and <parameters> section have the lowest precedence
2. parameters defined in the instance section have the highest precedence

For more details on advanced usage of parameter overwriting see chapter “Advanced parameter precedence”

3.4.1 VO parameter modification and addition

In some cases there is a need to store some information locally on the node even when yusing site configuration. As an example the VOMS node can be taken, where the MySQL root password is stored for the security reasons on the local node.

The approach used is to define locally new <vo> tags with the same name as the global ones, but providing only the set of modified and additional parameters. Any parameter defined in both global and local vo definitions will be overwritten by the value provided by the local <vo> definition.

3.5 EXAMPLES

This chapter provides an example configuration file layouts for a set of use-cases.

NOTE: Examples are supposed to demonstrate the structure of the configuration files. The parameters and values in these files are purely random and doesn't provide any usable information. To improve the readability of the examples some of the parameter attributes (description, etc.) are omitted. For exact parameter naming and values please refer to the “gLite installation guide” corresponding to your release.

3.5.1 Site configuration file with the vo definition.

glite-siteconfig.xml:

```
<siteconfig>
      <vo name=”EGEE”>
            <parameters>
                  <vo.name value=”EGEE”/>
```

```
                        <voms.host value=”kuiken.nikhef.nl”/>
                        .....
                </parameters>
        </vo>
        <vo name=”iteam”>
                <parameters>
                        <vo.name value=”iteam”/>
                        <voms.host value=”iteam-voms.cern.ch”/>
                        ....
                </parameters>
        </vo>
        ...
        <node name=”node1”>
                ......
        </node>
        ....
        <parameters>
                .....
        </parameters>
</siteconfig>
```

3.5.2 Site configuration file with the included vo definition.

glite-siteconfig.xml:

```
<siteconfig xmlns:xi=”http://www.w3.org/2001/XInclude”>
        <xi:include href=”vo-def.xml” xpointer=”vos”/>
        <node name=”node1”>
                ......
        </node>
        <parameters>
                .....
        </parameters>
        ....
</siteconfig>
```

vo-def.xml:

```
<vos>
        <vo name=”EGEE”>
                <parameters>
                        <vo.name value=”EGEE”/>
                        <voms.host value=”kuiken.nikhef.nl”/>
                        .....
                </parameters>
        </vo>
        <vo name=”iteam”>
                <parameters>
                        <vo.name value=”iteam”/>
                        <voms.host value=”iteam-voms.cern.ch”/>
                        ....
                </parameters>
        </vo>
        <vo name=”dteam”>
                <parameters>
                        <vo.name value=”iteam”/>
                        <voms.host value=”iteam-voms.cern.ch”/>
                        ....
                </parameters>
        </vo>
        ....
</vos>
```

3.5.3 Site configuration with restricted vo list

```
<siteconfig xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:include href="vo-def.xml" xpointer="vos"/>
      ...
      <node name="node1">
            <volist>
                  <include name="iteam"/>
                  <include name="dteam"/>
            </volist>
            <parameters>
                  ....
            </parameters>
      </node>
      <node name="node2">
            <volist>
                  <exclude name="iteam"/>
                  <exclude name="dteam"/>
            </volist>
            <parameters>
                  ....
            </parameters>
      </node>

      ....
      <parameters>
            .....
      </parameters>
</siteconfig>
```

As a consequence of this configuration node 'node1' will be configured for two virtual organizations: 'iteam' and 'dteam' while node 'node2' will be configured for all virtual organisations, except the 'iteam' and 'dteam'. In the case of this example this node will serve VO 'EGEE'. In case of later adding of new VO to the configuration, this will automatically be configured on the 'node2'.

3.5.4 Service instance creation

```
<siteconfig xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:include href="vo-def.xml" xpointer="vos"/>
      <node name='node1'>
            <instance service='xyz' iterate='volist'/>
            <parameters>
                  .....
            </parameters>
      </node>
      <node name='node2'>
            <volist>
                  <include name='EGEE'/>
            </volist>
            <instance name='zzz' iterate='volist'>
                  <parameters>
                        ....
                  </parameters>
            </instance>
            <parameters>
                  ....
            </parameters>
      </node>
      <node name='node3'>
            <volist>
                  <exclude name='EGEE'/>
            </volist>
            <instance name='zzz' iterate='volist'>
                  <parameters>
                        ....
                  </parameters>
            </instance>
            <parameters>
                  ....
            </parameters>
      </node>
</siteconfig>
```

node1: is configured for the service 'xyz' one instance for each virtual organisation defined in the configuration.

Node2: is configured for the service 'zzz' for virtual organization 'EGEE'. The configuration is fully equivalent to:

```
      ...
      <node name='node2'>
            <instance name='EGEE' name='zzz'>
                  <parameters>
                        ....
                  </parameters>
            </instance>
            <parameters>
                  ....
            </parameters>
      </node>
      ...
```

Node3: is configured for the service 'zzz' for all virtual organizations defined except 'EGEE'. In case of the example it will be for 'iteam' and 'dteam', but also any newly defined VO will land on this node.

3.5.5 VO parameter modification and addition

siteconfig.cfg.xml:

```
<siteconfig xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:include href="vo-def.xml" xpointer="vos"/>
      ....
</siteconfig>
```

glite-voms-server.cfg.xml:

```
<config>
      <vo name="EGEE">
            <parameters>
                  <mysql.root.password value="xxxxxx"/>
            </parameters>
      </vo>
      .....
</config>
```

The resulting configuration propagated to the services is equivalent to following vo definition:

```
      <vo name="EGEE">
            <parameters>
                  <vo.name value="EGEE"/>
                  <voms.host value="kuiken.nikhef.nl"/>
                  <mysql.root.password value="xxxxxx"/>
                  .....
            </parameters>
      </vo>
```

3.5.6 Incorrect configuration

```
            ...
```

```
<volist>
      <include name="iteam"/>
      <exclude name="dteam"/>
</volist>
...
```

Error: 'include' and 'exclude' tag cannot be used at the same time in one volist

4 EXPERT USAGE

4.1 ADVANCED INSTANCE ITERATE

VO management feature is just a special case of the much more generic approach implemented. The used approach introduces a concept of variable, which can be used to automatically create service instances. As an example can be taken service “xyz” which needs to create an instances as a function of some other entity like database or mass storage backend. Consequently all the entity related parameters can be grouped to entity definition (“vo” is the special case of such an entity) in the form:

```
<config>
      <entity name='entityName1'>
            <parameters>
                  ....
            </parameters>
      </entity>
      <entity name='entityName2'>
            <parameters>
                  ....
            </parameters>
      </entity>
      ....
</config>
```

where the tag name (<entity ...) can be chosen freely like:

```
      <storage name=”castor”>
      </storage>
      <storage name=”dpm”>
      </storage>
      ...
```

```
or
      <DBBackend name=”oracle”>
      </DBBackend>
      <DBBackend name=”MySQL”>
      </DBBackend>
      ...
```

to support it we note that <vo/> tag is only a special name chosen for the VO management purposes and fully fits into the presented schema.

Consequently, to create the needed service instances

```
      <instance service=”xyz” iterate=”entity”/>
```

tag should be used. The iterate attribute value (entity) should correspond with the real tag name of the variable, thus the corresponding examples are:

```
      <instance service=”xyz” iterate=”storage”/>
or
      <instance service=”xyz” iterate=”DBBackend”/>
```

All generated instances will be named corresponding to the name attribute of the given parameter:

```
      <instance name="castor" service="xyz"/>
      <instance name="DPM" service="xyz"/>
      ...
or
      <instance name="oracle" service="xyz"/>
      <instance name="MySQL" service="xyz"/>
      ...
```

Note: Support for expert usage described in this section is very limited and it's usage in production environment is recommended only after detailed testing.

4.2 ITERATION RESTRICTIONS ON THE INSTANCE LEVEL

Similarly to volist for vo definition, instance tag can also contain include or exclude tags to restrict iteration on the chosen parameter. The behaviour is equivalent to one described for volist.

4.3 ADVANCED PARAMETER PRECEDENCE

VO management in addition enables also to redefine/add parameter of the automatically generated instance by defining an instance with the same name and service as the generated one and by redefinition of the needed parameter. For examples see chapter 'Examples'.

4.4 EXAMPLES

4.4.1 Parameter replacement and addition

siteconf.cfg.xml:

```
<siteconfig xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:include href="vo-def.xml" xpointer="vos"/>
      <node name='node1'>
            <instance service='xyz' iterate='volist'/>
            <parameters>
                  .....
            </parameters>
      </node>
      ...
</siteconfig>
```

glite-<service>.cfg.xml

```
<config>
      <instance name='EGEE' service='xyz'>
            <parameters>
                  <voms.host value="EGEE-vo.cern.ch"/>
                  <new.value value="new"/>
            </parameters>
      </instance>
</config>
```

the voms.host parameter for instance name 'EGEE' service 'xyz' will be overwritten to the "EGEE-vo.cern.ch". Parameter new.value will be added to the given instance parameter list. The resulting configuration for EGEE instance of the service 'xyz' can be expressed:

```
      <instance name="EGEE" service='xyz'>
            <parameters>
                  <vo.name value="EGEE"/>
                  <voms.host value="EGEE-vo.cern.ch"/>
                  <new.value value=”new”/>
                  .....
            </parameters>
      </vo>
```

4.4.2 Iteration restrictions on the instance level

```
<siteconfig>
      <storage name=”castor”>
      </storage>
      <storage name=”dpm”>
      </storage>
      <storage name=”dcache”>
      </storage>

      <node name=”storagenode1”>
            <instance service=”storageservice” iterate=”storage”>
                  <include name=”castor”/>
                  <include name=”dpm”/>
            </instance>
      </node>
      <node name=”storagenode2”>
            <instance service=”storageservice: iterate=”storage”>
                  <exclude name=”castor”/>
                  <exclude name=”dpm”/>
            </instance>
      </node>
</siteconfig>
```

Following the rules described for volist it is evident that the storageservice on “storagenode1” will be configured to serve “castor” and “dpm” mass storage systems and storageservice on storagenode2 will serve the “dcache” and any newly added mass storage system.

4.4.3 Confusing configuration

Following previously explained approach the configuration:

```
            <volist>
                  <include name='EGEE'/>
            </volist>
            <instance name='zzz' iterate='vo'/>
```

is fully correct and will not cause any error message. Nevertheless, with the very high probability the resulting functionality will not fit with the requirements. This configuration will configure all security services following the volist but the service instances will be configured allways against ALL virtual organizations defined.

NOTE: The usage of the 'vo' for iteration is discouraged since it can cause inconsistency with the security settings.