IST-2001-32133
GridLab - A Grid Application Toolkit and Testbed

GridLab MDS Release 2.0

Author(s):	Giovanni Aloisio, Massimo Cafaro, Italo Epicoco, Daniele Lezzi, Maria Mirto, Silvia Mocavero
Document Filename:	GridLab-10-5-0001-MDS_2.0
Work package:	WP10 Information Services
Partner(s):	MPG, Germany MU, Czech Republic, SZTAKI, Hungary ISI, USA
Lead Partner:	University of Lecce
Config ID:	GridLab-10-5-0001-MDS_2.0
Document classification:	INTERNAL

Abstract: This document describes the release 2.0 of GridLab MDS. The documentation covers the GridLab MDS schema extension, related information providers and a web service to publish and retrieve information.

Last amendment date: 2004/01/19 & time: 09:06:42

Contents

1 Introduction

This document is associated to the prototype release of the GridLab MDS version 2.0. The GridLab MDS has been designed as an extension of Globus Project MDS, taking into account the requirements for grid computing suggested by others GridLab WPs . The prototype release includes:

- GridLab MDS schema extension;
- information providers related to the new schema;
- a set of functionalities, developed as a web service, to publish and retrieve information stored in the MDS;

The document is organized as follows: in section 2 we give the instructions needed to configure, build and install the GridLab MDS and the MDS web service; in section 3 we describe the additional information introduced in this release of the GridLab MDS and finally, in section 4 we present the web service methods to be used to publish and retrieve information.

2 Installation

This package includes the GridLab MDS Schema and related information providers; also included is a web service and related sample clients. To build and install the package please follow these installation instructions.

2.1 Installation of GridLab MDS

Required software:

perl, at least v5.x
Globus Toolkit v2.2.4

1) Modify the /etc/gridlab.conf file according to your environment

The GridLab MDS package requires the following environmental variables:

HOSTNAME: the FQDN name of the host
GRIDLAB_LOCATION: the gridlab installation directory
GLOBUS_FLAVOR_THREADS: the name of the threaded flavor of globus SDK
GLOBUS_FLAVOR: the name of the non threaded flavor of globus SDK
GLOBUS_LOCATION: the globus toolkit installation directory
GLOBUS_INCLUDE_THREADS: the directory containing the header files for the threaded flavor of the globus toolkit
GLOBUS_INCLUDE: the directory containing the header files for the non threaded flavor of the globus toolkit
GRID_MAPFILE: pathname of globus grid-mapfile
CERTIFICATION_AUTHORITY_DIR: pathname of directory containing recognized CAs certificates, usually /etc/grid-security/certificates

2) configure the distribution

```
configure --help
```

This will provide you with configure options; please apply to the following the options you need. Options relevant to GridLab users are the following ones:

```
--with-mds-hostname
```

This option sets the hostname of the GridLab MDS GIIS server; defaults to mds.gridlab.org. The mds hostname should not be changed when using this software on the GridLab testbed.

```
--with-spool-dir
```

This option sets the spool directory used by the web service; defaults to $GLOBUS_LOCATION/spool. Note that the web service must be able to read/write this directory at runtime.

3) configure

```
configure
```

This will configure the package using default options.

4) Build

```
make
```

5) Install the information providers as the user owning the Globus installation (typically the user "globus")

```
 make install
```

6) Modify the ldap server configuration file <GLOBUS_LOCATION>/etc/grid-info-slapd.conf as follows:
insert the following line after all of the include directives
include <GLOBUS_LOCATION>/etc/gridlab.schema

modify suffix related to GIIS database as "Mds-Vo-name=gridlab, o=Grid"

7) append the content of the file gridlab-ldif.conf to <GLOBUS_LOCATION>/etc/grid-info-resource-ldif.conf

```
cd <GLOBUS_LOCATION>/etc
cat gridlab-ldif.conf >> grid-info-resource-ldif.conf
```

8) Configure the <GLOBUS_LOCATION>/etc/grid-info-resource-register.conf setting the timing parameters as follows:

regperiod: time in seconds between outgoing registration messages. A time period too short will generate a lot of useless registration messages; a time period too long is such that the GIIS server will not receive updated registration messages. A tradeoff value for this parameter can be 300.

ttl: time in seconds to keep registration data in the GIIS. Normally it is twice the value of regperiod (600).

timeout: after how long should a client give up when querying the service. The value of this parameter is tightly coupled to the execution time of all of the information providers. To establish the right value on your platform, run the evaluation script:
sbin/eval_providers_execution_time

cachettl: time in seconds for client to cache data. This parameter strongly affects performances.

9) Configure the <GLOBUS_LOCATION>/etc/grid-info-resource-register.conf updating the following fields as shown:

```
dn: Mds-Vo-Op-name=register, Mds-Vo-name=gridlab, o=grid
bindmethod: ANONYM-ONLY
```

and adding:

```
regname: gridlab
```

10) Configure the etc/cluster-info.conf setting the parameters of the local resource management system (if there is one on the machine, otherwise do nothing)

11) as root user, restart gris daemon:

service SXXgris stop
service SXXgris start

2.2 Installation of GridLab MDS web service

This package includes a web service and related sample clients. To build and install the web service please follow the instructions below.

Required software:

Globus Toolkit v2.2.4 (http://www.globus.org)
gSOAP Toolkit v2.4.1
GSI plugin for gSOAP v1.5.5

In particular, we require a Globus installation with both threaded and non threaded flavors for the Information Services SDK bundle; another requirement is that the HOSTNAME variable in the gridlab.conf file must be set to the fully qualified domain name. To install on your platform please download gSOAP and GSI plugin for gSOAP, and copy the files according to the following:
gSOAP

- gSOAP compiler file (soapcpp2) -> bin
- gSOAP library file (stdsoap2.c) -> src/common
- gSOAP header file (stdsoap2.h) -> include

GSI plugin for gSOAP

- gSOAP GSI plugin library file (gsi.c) -> src/common
- gSOAP GSI plugin library header file (gsi.h) -> include

We assume here that the gridlab.conf file has been already properly configured when installing the GridLab MDS.

1) type

```
configure --help
```

This will provide you with configure options; please apply to the following the options you need. Options relevant to GridLab users are the following ones:

```
--with-mds-hostname
```

This option sets the hostname of the GridLab MDS GIIS server; defaults to mds.gridlab.org. The mds hostname should not be changed when using this software on the GridLab testbed.

`--with-mds-default-port`

This option sets the port where the Globus GRIS server is listening on; defaults to 2135. The port number should not be changed unless absolutely necessary. We strongly advise the use of default, IANA registered, port 2135.

`--with-base-dn`

This option sets the base dn to be used for searching the MDS; defaults to Mds-Vo-name=gridlab, o=grid. This value should not be changed, as it allows searching the entire MDS DIT (Directory Information Tree).

`--with-listening-port`

This option sets the listening port for the MDS web service to be installed; defaults to 21000. This value should not be changed when using this software on the GridLab testbed.

`--with-authorized-dn`

This option sets the pathname of the file authorized_dn used to authorize incoming connections through distinguished names; defaults to etc/authorized_dn. Please note that this pathname is not an absolute one, it is relative to the MDS package installation on $GRIDLAB_LOCATION/mds.

`--with-spool-dir`

This option sets the spool directory used by the web service; defaults to $GLOBUS_LOCATION/spool. Note that the web service must be able to read/write this directory at runtime.

2) configure

```
configure
```

This will configure the package using default options.

3) Build the server and client applications.

```
make
```

4) install everything as user with read/write permissions to the $PREFIX directory

```
make install
```

5) Run the server using a non privileged user account, for instance the globus account as follows. In order to succeed, you will need to create in the home directory of the user that will be used for running the web service a .globus directory containing the host machine credential. Please, be sure to set appropriate permissions (.globus directory: 755; host certificate: 444; host key: 400).

```
a) grid-proxy-init -valid 720:00 (acquire host machine credential valid one month)

In the next release we will make the server acquire automatically the credential.

b) cd $GRIDLAB_LOCATION/mds/server
c) nohup ./gridlab-server-threaded &
```

6) Be aware that if you are updating the gridlab-MDS-service from a previous version, you must delete all of the registered services and webservices in the spool directories

3 GridLab MDS schema extension

This section describes the information schema developed. The schema closely takes into account the requirements for grid computing provided by others GridLab WPs as described initially in the deliverable D10.2 and as required during the course of the project. Of course, this is not meant to be static, the schema will continue to evolve and will be extended to support additional information that will be required by the GridLab project. The current security policy adopted for this prototype release of GridLab MDS is to allow anonymous binding to the MDS for reading; however, only authorized users are actually allowed to write data through our MDS web service, and authentication/authorization is based on globus GSI. It is worth noting here that, indeed, part of the information published inside the MDS is sensitive, so that an even better security policy will be in place. As a matter of fact, we will enforce a GridLab security policy by using the authorization service that will be provided by the security WP6, to authorize trusted users to access the MDS for reading/writing. The current prototype release version 2.0 provides information related to:

- Services;
- Web Services;
- Software;
- Users;
- Firewalls;
- Virtual Organizations;
- Certification Authorities;
- Clusters.

The information providers for users, CA certificates and clusters get information directly. Information providers for services, web services, software, firewalls, and VOs get information from an internally managed database, which is initially empty and must be populated through the GridLab MDS web service. This can be done, for instance, using the provided gridlab-*-register-client programs. It is worth noting here that our MDS is able to handle all of the characters set specified in rfc1738. Since many characters are not admitted in the LDAP server (rfc2252), an escape sequence is generated for each character included in rfc1738 but not admitted in rfc2252. Using the gridlab_search or gridlab_*_lookup methods, each escape sequence is properly convertend into the corresponding character upon return. The encoded characters are:

```
~ _ & ? % { } | \ \ ^ [ ] ‘ ; @ = $ + ! \ ’
```

We now briefly review the attributes to be inserted inside the GridLab MDS. These attributes comes from a detailed comparison of attributes already available inside the Globus toolkit MDS and attributes needed for GridLab grid computing scenarios.

3.1 Information related to Services and Web Services

During the course of the GridLab project, a number of services and Web Services will be developed by GridLab Work Packages. One of the most important requirements for GridLab grid computing scenarios is the ability to discover services dynamically. The MDS will provide GridLab developers with the following functionalities: registration, unregistration and lookup. More than one instance for each service or Web Service can be registered. The following attributes belong to these categories:

- GridLab-Mds-Service-name: service logical name;
- GridLab-Mds-Service-description: service description;
- GridLab-Mds-Service-default-port: service default port;
- GridLab-Mds-Service-keywords: set of key words of a service;
- GridLab-Mds-Service-accessurl: service access URL;
- GridLab-Mds-Service-publisher: service publisher (X509v3 certificate distinguished name).
- GridLab-Mds-WebService-name: Web Service name;
- GridLab-Mds-WebService-WSDLlocationurl: (multi-valued) URL where the WSDL document for the Web Service can be found;
- GridLab-Mds-WebService-description: Web Service description;
- GridLab-Mds-WebService-keywords: set of key words of a web service;
- GridLab-Mds-WebService-publisher: Web Service publisher (X509v3 certificate distinguished name);
- GridLab-Mds-WebService-accessurl: URL where the web service is listening on.

3.2 Information related to installed software

In order to submit jobs on grid resources, it is of crucial importance to know details related to the software packages that will be used in a run, may be a complex parallel simulation or a simple batch job. The following attributes belong to this category:

- GridLab-Mds-Software-name: name;
- GridLab-Mds-Software-version: version;
- GridLab-Mds-Software-path: pathname;
- GridLab-Mds-Software-totalLicences: number of total software licences;
- GridLab-Mds-Software-licenceInfo: information about software licence;
- GridLab-Mds-Software-startupEnvironment: (multi-valued) software startup environmental variables;
- GridLab-Mds-Software-executable: software executable;

- GridLab-Mds-Software-arguments: (multi-valued) software arguments;
- GridLab-Mds-Software-description: software description;
- GridLab-Mds-Software-type: software type;
- GridLab-Mds-Software-helpURL: software help URL;
- GridLab-Mds-Software-usage: software usage.

3.3 Information related to users

Information related to users allow complex brokering strategies: for instance, once the set of computing resources available to a user is known to a broker, it is then possible to choose carefully where to submit a user's job. The broker's decision will be based on the information gathered from the MDS and on the job's requirements. The following attributes belong to this category:

- GridLab-Mds-User-ID: user's login name on local resource;
- GridLab-Mds-User-Mapped-DN: (multi-valued) attribute representing the distinguished name mapped on the user;
- GridLab-Mds-User-homedir: user's home directory;
- GridLab-Mds-User-shell: user's shell;
- GridLab-Mds-User-UID: user's UID;
- GridLab-Mds-User-GID: user's GID;
- GridLab-Mds-User-comment: a short comment about the user.

3.4 Information related to firewalls

This kind of information is strictly related to service information. As a matter of fact, before registering a service, GridLab developers will query the MDS to know dynamically the range of open ports available on a specified computational resource. This is required to allow other people to connect to a service. The following attributes belong to this category:

- GridLab-Mds-Firewall-hostname: firewall hostname;
- GridLab-Mds-Firewall-ports: (multi-valued) attribute representing open ports (range);
- GridLab-Mds-Firewall-validityTime: time frame during which open ports ranges are valid;
- GridLab-Mds-Firewall-adminDN: distinguished name of the firewall administrator.

3.5 Information related to Virtual Organizations

The GridLab project will span multiple Virtual Organizations. Corresponding information will allow people to know, for instance, how to request an account on a machine belonging to a particular Virtual Organization, or the people to contact in case of trouble. The following attributes belong to this category:

- GridLab-Mds-Vo-name: Virtual Organization to which a specified computational resource belongs to;
- GridLab-Mds-Vo-resourceType: Virtual Organization resource type;
- GridLab-Mds-Vo-helpDeskPhoneNumber: (multi-valued) help desk phone number;
- GridLab-Mds-Vo-helpDeskURL: URL pointing to a Virtual Organization's web page;
- GridLab-Mds-Vo-adminName: administrator name of the VO.

3.6 Information related to recognized Certification Authorities

This is a set of information about the certification authorities allowed to sign the user's certificates that can be used to access and use the resource; it is important to know which CAs must be contacted to obtain a valid certificate and which CAs are recognized on a given computational resource. The following attributes belong to this category:

- GridLab-Mds-Certificate-Subj: distinguished name of the recognized certification authority;
- GridLab-Mds-Certificate-version: CA's certificate version;
- GridLab-Mds-Certificate-serialNumber: CA's certificate serial number;
- GridLab-Mds-Certificate-signatureAlgorithm: CA's certificate signature algorithm;
- GridLab-Mds-Certificate-issuer: CA's certificate issuer;
- GridLab-Mds-Certificate-validity-from: beginning date of the CA's certificate validity;
- GridLab-Mds-Certificate-validity-to: end date of the CA's certificate validity;
- GridLab-Mds-Certificate-publicKeyAlgorithm: CA's certificate public key algorithm;
- GridLab-Mds-Certificate-RSAPublicKey: CA's certificate RSA public key;
- GridLab-Mds-Certificate-crlURL: url where the CA CRL can be found.

3.7 Information related to Clusters

The information about Clusters include general information about nodes (number of CPUs, available memory etc), and detailed information about queues and jobs running. Accessing such information is crucial for resource management, brokering strategies etc. The current release provides support for the PBS resource management system; the final GridLab MDS release 2.0 will provide additional support for LSF. The following attributes belong to this category:

- GridLab-Mds-Cluster-aliasname: the alias name of the cluster;

- GridLab-Mds-Cluster-cpudistribution: the cpu distribution of the nodes given in the form of ”m cpu:n nodes”;
- GridLab-Mds-Cluster-homogeneity: a boolean flag indicating the homogeneity of the cluster nodes;
- GridLab-Mds-Cluster-Lrms-type: the type of Local Resource Management System;
- GridLab-Mds-Cluster-Lrms-version: the version of Local Resource Management System;
- GridLab-Mds-Cluster-Lrms-config: additional remarks on the LRMS configuration;
- GridLab-Mds-Cluster-nodecpu: the cpu type of the nodes (model name + MHz);
- GridLab-Mds-Cluster-nodememory: the memory installed on the node in MB;
- GridLab-Mds-Cluster-runtimeenvironment: (multi-valued) pre-installed software packages of the cluster;
- GridLab-Mds-Cluster-support: (multi-valued) RFC822 email address of support;
- GridLab-Mds-Cluster-total-jobs: the total number of batch jobs in the cluster;
- GridLab-Mds-Cluster-freecpus: the number of free cpus in the cluster;
- GridLab-Mds-Cluster-totalcpus: the total number of cpus in the cluster;
- GridLab-Mds-Cluster-queue-total-count: total number of cluster queues;
- GridLab-Mds-Cluster-queues: (multi-valued) cluster queue name;
- GridLab-Mds-Queue-name: the name of the queue;
- GridLab-Mds-Queue-assignedcpunumber: the number of cpus assigned to the queue;
- GridLab-Mds-Queue-status: the queue status;
- GridLab-Mds-Queue-maxqueueable: the max number of jobs allowed to reside in the queue;
- GridLab-Mds-Queue-maxrunning: the max number of jobs allowed to run from this queue;
- GridLab-Mds-Queue-queued: the number of jobs waiting in the queue;
- GridLab-Mds-Queue-running: the number of running jobs in the cluster belonging to this queue;
- GridLab-Mds-Queue-maxwallclocktime: the max wall clock time allowed for jobs submitted to the queue in mins;
- GridLab-Mds-Queue-maxcputime: the max CPU time allowed for jobs submitted to the queue in mins;
- GridLab-Mds-Queue-job-total-count: the number of queue total jobs;
- GridLab-Mds-Queue-jobs: (multi-valued) queue total jobs name;
- GridLab-Mds-Job-name: the name of the job;

- GridLab-Mds-Job-execcluster: the name of the cluster where the job is managed;
- GridLab-Mds-Job-execqueue: the name of the execution queue of the job;
- GridLab-Mds-Job-id: the job identifier string;
- GridLab-Mds-Job-owner: the SubjectName of the job owner;
- GridLab-Mds-Job-reqcput: the cputime request of the job in minutes;
- GridLab-Mds-Job-status: the status of the job;
- GridLab-Mds-Job-submissiontime: the submission time of the job;
- GridLab-Mds-Job-usedcputime: the consumed cputime of the job in minutes;
- GridLab-Mds-Job-usedmem: the memory usage of the job (in KB);
- GridLab-Mds-Job-usedwalltime: the consumed walltime of the job in minutes;
- GridLab-Mds-Job-estimatedresponsetime: estimated time between job submission and execution in sec.

3.8 Information model

In this section we provide a detailed explanation of the object classes and attributes belonging to the GridLab information schema.
Object class GridLabServiceName (OID 1.3.6.1.4.1.3536.2.6.3536.9.1.1) is a structural class for information related to services. The class inherits from the Globus schema class Mds the following attributes:

- Mds-validfrom (global time at which the object and its information is first valid)
- Mds-validto (global time at which the object and its information is no longer valid)
- Mds-keepto (global time at which the object and its information should be deleted; the purge time should always be later than the expiration time, if it exists; the existence of a purge time hints that some information in the object may be useful even when the object is invalid, i.e. that the invalid object is better than no information at all)

The class contain the following attributes:

- GridLab-Mds-Service-name (OID 1.3.6.1.4.1.3536.2.6.3536.9.1.1.0.1) is a required single-valued attribute representing the service's name.

Object class GridLabService (OID 1.3.6.1.4.1.3536.2.6.3536.9.1) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Service-accessurl (OID 1.3.6.1.4.1.3536.2.6.3536.9.1.0.1) is a required single-valued attribute representing the URL where the service is listening on.
- GridLab-Mds-Service-publisher (OID 1.3.6.1.4.1.3536.2.6.3536.9.1.0.2) is a required single-valued attribute representing the distinguished name of the service's publisher.

Object class GridLabServiceInstanceGroup (OID 1.3.6.1.4.1.3536.2.6.3536.9.2.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Service-Instances-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.2.1.0.1) is a required single-valued attribute representing service instances group name.
- GridLab-Mds-Service-description (OID 1.3.6.1.4.1.3536.2.6.3536.9.2.1.0.2) is a required single-valued attribute representing the service description.
- GridLab-Mds-Service-Instance-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.2.1.0.3) is a required single-valued attribute representing the total number of service instances.
- GridLab-Mds-Service-Instance-accessurl (OID 1.3.6.1.4.1.3536.2.6.3536.9.2.1.0.4) is a required multi-valued attribute representing the service instances URL.
- GridLab-Mds-Service-keywords (OID 1.3.6.1.4.1.3536.2.6.3536.9.2.1.0.5) is a required single-valued attribute representing the service key words.
- GridLab-Mds-Service-default-port (OID 1.3.6.1.4.1.3536.2.6.3536.9.2.1.0.6) is a required single-valued attribute representing the service default port.

Object class GridLabServiceGroup (OID 1.3.6.1.4.1.3536.2.6.3536.9.3.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Service-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.3.1.0.1) is a required single-valued attribute representing the total number of services registered on the local resource.
- GridLab-Mds-Services (OID 1.3.6.1.4.1.3536.2.6.3536.9.3.1.0.2) is a required multi-valued attribute representing the names of all of the services registered on the local resource.
- GridLab-Mds-Services-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.3.1.0.3) is a required single-valued attribute representing the name of the group

Object class GridLabSoftwareName (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Software-name (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.1.0.1) is a required single-valued attribute representing a software package name.

Object class GridLabSoftware (OID 1.3.6.1.4.1.3536.2.6.3536.9.4) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Software-version (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.1) is a required single-valued attribute representing a software package version.
- GridLab-Mds-Software-path (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.2) is a required single-valued attribute representing the pathname (directory) where software is installed.
- GridLab-Mds-Software-totalLicences (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.3) is a required single-valued attribute representing the total number of software package licences.
- GridLab-Mds-Software-licenceInfo (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.4) is a required single-valued attribute describing information related to the software licence.
- GridLab-Mds-Software-startupEnvironment (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.5) is a required multi-valued attribute representing a software package startup environment.
- GridLab-Mds-Software-executable (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.6) is a required single-valued attribute representing a software package executable pathname.
- GridLab-Mds-Software-arguments (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.7) is a required multi-valued attribute describing the software package command line arguments.

- GridLab-Mds-Software-description (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.8) is a required single-valued attribute representing a software package description.
- GridLab-Mds-Software-type (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.9) is a required single-valued attribute representing the software type.
- GridLab-Mds-Software-helpURL (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.10) is a required single-valued attribute representing a software package URL pointing to a web page describing the software.
- GridLab-Mds-Software-usage (OID 1.3.6.1.4.1.3536.2.6.3536.9.4.0.11) is a required single-valued attribute representing the software usage.

Object class GridLabSoftwareGroup (OID 1.3.6.1.4.1.3536.2.6.3536.9.5.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Software-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.5.1.0.1) is a required single-valued attribute representing the total number of software packages registered.
- GridLab-Mds-Software (OID 1.3.6.1.4.1.3536.2.6.3536.9.5.1.0.2) is a required multi-valued attribute representing the software packages name registered.
- GridLab-Mds-Software-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.5.1.0.3) is a required single-valued attribute representing the name of the group

Object class GridLabUserUniqueID (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.1) is a structural class that inherits from Mds and contains the following attributes:

-GridLab-Mds-User-ID (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.1.0.1) is a required single-valued attribute representing the user login name.

Object class GridLabUser (OID 1.3.6.1.4.1.3536.2.6.3536.9.6) is an auxiliary class that inherits from Mds and contains the following attributes:

-GridLab-Mds-User-Mapped-DN (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.0.1) is a required multi-valued attribute representing the Distinguished Name mapped on the user.
-GridLab-Mds-User-homedir (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.0.2) is a required single-valued representing the user's home directory
-GridLab-Mds-User-shell (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.0.3) is a required single-valued attribute representing the user shell.
-GridLab-Mds-User-UID (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.0.4) is a required single-valued attribute representing the user UID.
-GridLab-Mds-User-GID (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.0.5) is a required single-valued attribute representing the user GID.
-GridLab-Mds-User-comment (OID 1.3.6.1.4.1.3536.2.6.3536.9.6.0.6) is a required single-valued attribute representing the user comment.

Object class GridLabUserGroup (OID 1.3.6.1.4.1.3536.2.6.3536.9.7.1) is a structural class that inherits from Mds and contains the following attributes:

-GridLab-Mds-User-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.7.1.0.1) is a required single-valued attribute representing the number of users.
-GridLab-Mds-User (OID 1.3.6.1.4.1.3536.2.6.3536.9.7.1.0.2)is a required multi-valued attribute representing the list of user unique ID.

-GridLab-Mds-User-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.7.1.0.3) is a required single-valued attribute representing the name of the group

Object class GridLabFirewallHostname (OID 1.3.6.1.4.1.3536.2.6.3536.9.8.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Firewall-hostname (OID 1.3.6.1.4.1.3536.2.6.3536.9.8.1.0.1) is a required single-valued attribute representing the firewall hostname.

Object class GridLabFirewall (OID 1.3.6.1.4.1.3536.2.6.3536.9.8) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Firewall (OID 1.3.6.1.4.1.3536.2.6.3536.9.8.0.1) is a required single-valued attribute representing the firewall.
- GridLab-Mds-Firewall-ports (OID 1.3.6.1.4.1.3536.2.6.3536.9.8.0.2) is a required multi-valued attribute representing open ports ranges.
- GridLab-Mds-Firewall-adminDN (OID 1.3.6.1.4.1.3536.2.6.3536.9.8.0.3) is a required single-valued attribute representing a firewall administrator Distinguished Name.
- GridLab-Mds-Firewall-validityTime (OID 1.3.6.1.4.1.3536.2.6.3536.9.8.0.4) is a required single-valued attribute representing the time frame during which open port ranges are valid.

Object class GridLabVoName (OID 1.3.6.1.4.1.3536.2.6.3536.9.9.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Vo-name (OID 1.3.6.1.4.1.3536.2.6.3536.9.9.1.0.1) is a required single-valued attribute representing the name of virtual organization to which the computing resource belongs to.

Object class GridLabVo (OID 1.3.6.1.4.1.3536.2.6.3536.9.9) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Vo-resourceType (OID 1.3.6.1.4.1.3536.2.6.3536.9.9.0.1) is a required single-valued attribute representing the resource type related to the virtual organization.
- GridLab-Mds-Vo-helpDeskPhoneNumber (OID 1.3.6.1.4.1.3536.2.6.3536.9.9.0.2) is a required multi-valued attribute representing the help desk phone number related to the virtual organization.
- GridLab-Mds-Vo-helpDeskURL (OID 1.3.6.1.4.1.3536.2.6.3536.9.9.0.3) is a required single-valued attribute representing the URL pointing to a web page describing the Virtual Organization.
- GridLab-Mds-Vo-adminName (OID 1.3.6.1.4.1.3536.2.6.3536.9.9.0.4) is a required single-valued attribute representing the name of the virtual organization administrator.

Object class GridLabVoGroup (OID 1.3.6.1.4.1.3536.2.6.3536.9.10.1) is a structural class that inherits from Mds and contains the following attributes:

-GridLab-Mds-Vo-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.10.1.0.1) is a required single-valued attribute that represents the total number of Virtual Organization which the resource belongs to
-GridLab-Mds-Vo (OID 1.3.6.1.4.1.3536.2.6.3536.9.10.1.0.2) is a required multi-valued attribute representing the list of all the Virtual Organization which the resource belongs to

-GridLab-Mds-Vo-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.10.1.0.3) is a required single-valued attribute representing the name of the group

Object class GridLabCertificateSubject (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.1) is a structural class that inherits from Mds and contains the following attributes:

-GridLab-Mds-Certificate-subject (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.1.0.1) is a required multi-valued attribute representing the subject name of all of the certification authority accepted by the computational resource.
-GridLab-Mds-Certificates-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.1.0.2) is a required single-valued attribute representing the name of the group

Object class GridLabCertificate (OID 1.3.6.1.4.1.3536.2.6.3536.9.11) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Certificate-version (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.1) is a required single-valued attribute representing the version of the certificate.
- GridLab-Mds-Certificate-serialNumber (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.2) is a required single-valued attribute representing the serial number of the certificate.
- GridLab-Mds-Certificate-signatureAlgorithm (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.3) is a required single-valued attribute representing the signature algorithm of the certificate.
- GridLab-Mds-Certificate-issuer (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.4) is a required single-valued attribute representing the issuer of the certificate.
- GridLab-Mds-Certificate-validity-from (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.5) is a required single-valued attribute representing the beginning date of the certificate validity.
- GridLab-Mds-Certificate-validity-to (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.6) is a required single-valued attribute representing the end date of the certificate validity.
- GridLab-Mds-Certificate-publicKeyAlgorithm (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.7) is a required single-valued attribute representing the public key algorithm of the certificate.
- GridLab-Mds-Certificate-RSAPublicKey (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.8) is a required single-valued attribute representing the RSA public key of the certification authority.
-GridLab-Mds-Certificate-Subj (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.9) is a required single-valued attribute representing the subject name of the CA's certificate.
-GridLab-Mds-Certificate-crlURL (OID 1.3.6.1.4.1.3536.2.6.3536.9.11.0.10) is an optional single-valued attribute representing the url where the CA CRL can be found.

Object class GridLabWebServiceName (OID 1.3.6.1.4.1.3536.2.6.3536.9.12.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-WebService-name (OID 1.3.6.1.4.1.3536.2.6.3536.9.12.1.0.1) is a required single-valued attribute representing the web service name.

Object class GridLabWebService (OID 1.3.6.1.4.1.3536.2.6.3536.9.12) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-WebService-accessurl (OID 1.3.6.1.4.1.3536.2.6.3536.9.12.0.1) is a required single-valued attribute representing the URL where the web service is listening on (constrained to be local).
- GridLab-Mds-WebService-publisher (OID 1.3.6.1.4.1.3536.2.6.3536.9.12.0.2) is a required single-

valued attribute representing the distinguished name of the web service publisher.

Object class GridLabWebServiceInstanceGroup (OID 1.3.6.1.4.1.3536.2.6.3536.9.13.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-WebService-Instances-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.13.1.0.1) is a required single-valued attribute representing the web service instances group name.
- GridLab-Mds-WebService-WSDLlocationurl (OID 1.3.6.1.4.1.3536.2.6.3536.9.13.1.0.2) is a required multi-valued attribute representing the location of the web service WSDL file (URL).
- GridLab-Mds-WebService-description (OID 1.3.6.1.4.1.3536.2.6.3536.9.13.1.0.3) is a required single-valued attribute representing the web service description.
- GridLab-Mds-WebService-keywords (OID 1.3.6.1.4.1.3536.2.6.3536.9.13.1.0.4) is a required single-valued attribute representing the set of web service key words.
- GridLab-Mds-WebService-Instance-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.13.1.0.5) is a required single-valued attribute representing the total number of web service instances.
- GridLab-Mds-WebService-Instance-accessurl (OID 1.3.6.1.4.1.3536.2.6.3536.9.13.1.0.6) is a required multi-valued attribute representing the list of the URLs where the web service instances are listening on.

Object class GridLabWebServiceGroup (OID 1.3.6.1.4.1.3536.2.6.3536.9.14.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-WebService-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.14.1.0.1) is a required single-valued attribute representing the total number of Web Services.
- GridLab-Mds-WebServices (OID 1.3.6.1.4.1.3536.2.6.3536.9.14.1.0.2) is a required multi-valued attribute representing the web service name.
- GridLab-Mds-WebServices-Group (OID 1.3.6.1.4.1.3536.2.6.3536.9.14.1.0.3) is a required single-valued attribute representing the web services group name.

Object class GridLabJobName (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Job-name (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.1.0.1) is a required single-valued attribute representing the name of the job.

Object class GridLabJob (OID 1.3.6.1.4.1.3536.2.6.3536.9.15) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Job-execcluster (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.1) is an optional single-valued attribute representing the name of the cluster where the job is managed.
- GridLab-Mds-Job-execqueue (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.2) is an optional single-valued attribute representing the name of the execution queue of the job.
- GridLab-Mds-Job-id (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.3) is a required single-valued attribute representing the job identifier string.
- GridLab-Mds-Job-owner (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.4) is a required single-valued attribute representing the subject name of the job owner.
- GridLab-Mds-Job-reqcput (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.5) is an optional single-valued attribute representing the cputime request of the job in minutes.
- GridLab-Mds-Job-status (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.6) is a required single-valued at-

tribute representing the status of the job.
- GridLab-Mds-Job-submissiontime (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.7) is an optional single-valued attribute representing the submission time of the job.
- GridLab-Mds-Job-usedcputime (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.8) is an optional single-valued attribute representing the consumed cputime of the job in minutes.
- GridLab-Mds-Job-usedmem (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.9) is an optional single-valued attribute representing the memory usage of the job (in KB).
- GridLab-Mds-Job-usedwalltime (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.10) is an optional single-valued attribute representing the consumed walltime of the job in minutes.
- GridLab-Mds-Job-estimatedresponsetime (OID 1.3.6.1.4.1.3536.2.6.3536.9.15.0.11) is an optional single-valued attribute representing the estimated time between job submission and execution in sec.

Object class GridLabQueueName (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.1) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Queue-name (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.1.0.1) is a required single-valued attribute representing the name of the queue.

Object class GridLabQueue (OID 1.3.6.1.4.1.3536.2.6.3536.9.16) is an auxiliary class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Queue-assignedcpunumber (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.1) is an optional single-valued attribute representing the number of cpus assigned to the queue.
- GridLab-Mds-Queue-status (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.2) is a required single-valued attribute representing the queue status.
- GridLab-Mds-Queue-maxqueueable (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.3) is an optional single-valued attribute representing the max number of jobs allowed to reside in the queue.
- GridLab-Mds-Queue-maxrunning (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.4) is an optional single-valued attribute representing the max number of jobs allowed to run from this queue.
- GridLab-Mds-Queue-queued (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.5) is an optional single-valued attribute representing the number of jobs waiting in the queue.
- GridLab-Mds-Queue-running (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.6) is an optional single-valued attribute representing the number of running jobs in the cluster belonging to this queue.
- GridLab-Mds-Queue-maxwallclocktime (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.7) is an optional single-valued attribute representing the max wall clock time allowed for jobs submitted to the queue.
- GridLab-Mds-Queue-maxcputime (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.8) is an optional single-valued attribute representing the max CPU time allowed for jobs submitted to the queue in mins.
- GridLab-Mds-Queue-job-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.9) is an optional single-valued attribute representing the number of queue total jobs.
- GridLab-Mds-Queue-jobs (OID 1.3.6.1.4.1.3536.2.6.3536.9.16.0.10) is an optional multi-valued attribute representing the queue jobs name.

Object class GridLabCluster (OID 1.3.6.1.4.1.3536.2.6.3536.9.17) is a structural class that inherits from Mds and contains the following attributes:

- GridLab-Mds-Cluster-aliasname (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.1) is an optional single-valued attribute representing the alias name of the cluster.

- GridLab-Mds-Cluster-cpudistribution (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.2) is an optional single-valued attribute representing the cpu distribution of the nodes given in the form of m cpu:n machines.
- GridLab-Mds-Cluster-homogeneity (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.3) is an optional single-valued attribute representing a boolean flag indicating the homogeneity of the cluster nodes.
- GridLab-Mds-Cluster-Lrms-type (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.4) is a required single-valued attribute representing the type of the Local Resource Management System.
- GridLab-Mds-Cluster-Lrms-version (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.5) is a required single-valued attribute representing the version of the Local Resource Management System.
- GridLab-Mds-Cluster-Lrms-config (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.6) is an optional single-valued attribute representing additional remarks on the lrms config.
- GridLab-Mds-Cluster-nodecpu (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.7) is an optional single-valued attribute representing the cpu type of the nodes (model name + MHz).
- GridLab-Mds-Cluster-nodememory (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.8) is an optional single-valued attribute representing the installed memory of a node in MB.
- GridLab-Mds-Cluster-runtimeenvironment (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.9) is an optional multi-valued attribute representing the pre-installed software packages of the cluster.
- GridLab-Mds-Cluster-support (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.10) is an optional multi-valued attribute representing the RFC822 email address of support.
- GridLab-Mds-Cluster-total-jobs (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.11) is an optional single-valued attribute representing the total number of batch jobs in the cluster.
- GridLab-Mds-Cluster-freecpus (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.12) is an optional single-valued attribute representing the total number of free cpus in the cluster.
- GridLab-Mds-Cluster-totalcpus (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.13) is an optional single-valued attribute representing the total number of cpus in the cluster.
- GridLab-Mds-Cluster-queue-total-count (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.14) is an optional single-valued attribute representing the total number of cluster queues.
- GridLab-Mds-Cluster-queues (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.15) is an optional multi-valued attribute representing the cluster queue name.
- GridLab-Mds-Cluster-info (OID 1.3.6.1.4.1.3536.2.6.3536.9.17.0.16) is an optional single-valued attribute representing the cluster info name.

4 GridLab MDS web service

The web service has been developed as a threaded server in order to improve the performances; it provides the user with several methods that can be invoked to search, register, unregister and lookup information.
The namespace used by the web service is urn:mds. The web service's methods are described using Web Service Description Language in Appendix; here, we describe (using the C language, gSOAP return values for remote methods and classifying the parameters as IN, OUT, INOUT) the data structures and methods:

```
typedef int xsd__int;
typedef char * xsd__string;

struct string_array{
xsd__string *__ptr;
xsd__int __size;
```

```
};

struct Attribute {
xsd__string *__ptr;       //pointer to the values array
xsd__int     __size;      //size of the values array
};

struct Entry {
struct Attribute *__ptr;  //pointer to the attribute array
xsd__int __size;          //size of the attribute array
};

struct mdsInfo {
struct Entry  *__ptr;
xsd__int  __size;
};

enum gridlab__scopes {LDAP_SCOPE_BASE, LDAP_SCOPE_ONE, LDAP_SCOPE_SUB};
```

4.1 Search and description

-int gridlab__search(xsd__string mds_hostname, xsd__int port, xsd__string base_dn, enum gridlab__scopes scope, xsd__string attributes, xsd__string filter, struct mdsInfo**result)
This method can be used to query an arbitrary MDS server for specified information.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: hostname of the LDAP server to contact
port: port number where the LDAP server is listening on
base_dn: base distinguished name to be used as starting point for searching
scope: the scope of the search. It can be: LDAP_SCOPE_BASE, LDAP_SCOPE_ONE, LDAP_SCOPE_SUB
attributes: space-separated list of attributes
filter: search filter criteria expressed according to RFC 2252.

OUT parameters:

result: mdsInfo structure containing all the information provided by the specified mds server.

-int gridlab__getServiceDescription(void *_, xsd__string *description)
This method can be used to get the description of the service.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

_: dummy input parameter to allow compatibility across different unix platforms

OUT parameters:

description: a description of the functionalities provided by this server.

4.2 Services

-int gridlab__register_service(xsd__string name, xsd__string hostname, xsd__int port, xsd__string protocol, xsd__int dport, xsd__string description, xsd__string keywords, xsd__int *result)
This method allows the user to register a new service or a an instance of an existing service. Please note that this method allows registering services runnning on machines lacking a globus toolkit installation. In this case, the GRIS server where such services will be registered will act as a front-end machine for those non globus machines. It is worth noting here that the following restriction applies: the user is allowed to register services running on non globus machines only if these machines belong to the same domain name as the GRIS machine where the service is going to be registered.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the service to be registered
hostname: the FQDN of the server that provides the service (must be a symbolic address)
port: port number where the service will be listening on
protocol: communication protocol to be used to contact the service
dport: default port number where the service will be listening on
description: human readable description of the service
keywords: service key words

OUT parameters:

result: set to a value < 0 if an error occurs during the registration, 0 on success for the service first instance registration; set to 1 for successful registration of additional instances of the service.

If the service being registered already exists, the registration will fail. If you want to modify an existing service, you must unregister it and register it again.

-int gridlab__unregister_service(xsd__string name, xsd__string hostname, xsd__int port, xsd__int allinst, xsd__int *result)
This method removes a service, a single instance of a service or all of the instances of a service.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the service to be removed
hostname: the FQDN of the server that provides the service
port: port number where the service will be listening on
allinst: a flag set to 1 to remove all of the instances of the specified service, set to 0 to remove a single instance.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

If the service does not exist the method returns 0 anyway.

-int gridlab__lookup_service(xsd__string mds_hostname, xsd__string name, xsd__int dport, xsd__int sport, xsd__string prot, struct mdsInfo**result)
This method returns all the information about the services. It contacts the supplied MDS server on the default port. If any input parameters has been specified, the method returns the information about all the services and the related instances registered on the MDS server. It is possible to search the information about one or more services specifying the service name, port, default port and protocol parameters or a combination of these.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the hostname of the MDS server to be contacted
name: the logical name of the service to be searched
dport: the default port of the service to be searched
sport: the port number where the service to be searched is listening on
prot: the protocol of the service to be searched.

OUT parameters:

result: mdsInfo structure containing services information.

4.3 Web Services

-int gridlab__register_webservice(xsd__string name, xsd__string wsdllocation, xsd__string description, xsd__string url, xsd__string keywords, xsd__int *result)
This method allows the user to register a new web service. The wsdllocation parameter is a list, space-separated, of web service WSDL file location URL variables; each variable must not contain spaces. Please note that this method allows registering web services runnning on machines lacking a globus toolkit installation. In this case, the GRIS server where such web services will be registered will act as a front-end machine for those non globus machines. It is worth noting here that the following restriction applies: the user is allowed to register web services running on non globus machines only if these machines belong to the same domain name as the GRIS machine where the web service is going to be registered.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the new web service to be registered
wsdllocation: a list, space-separated, of web service WSDL file location URL variables
description: human readable description of the web service
url: the web service access URL (note: the URL must contain a symbolic address)
keywords: web service key words

OUT parameters:

result: set to a value < 0 if an error occurs during the registration, 0 on success when registering the first instance of a web service; 1 on success when registering additional instances of the web service.

If the new web service being registered already exists, the registration will fail. If you want to modify an existing web service, you must unregister it and register it again.

-int gridlab__unregister_webservice(xsd__string name, xsd__int instnum, xsd__int allinst, xsd__int *result)
This method removes an existing web service, a single instance of a web service or all of the instances of a web service.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the web service to be removed
url: the access URL of the web service instance to be removed
allinst: a flag set to 1 to remove all of the instances of the specified service, set to 0 to remove a single instance.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

If the web service does not exist the method returns 0 anyway.

-int gridlab__lookup_webservice(xsd__string mds_hostname, xsd__string name, struct mdsInfo **result)
This method returns all the information about the web services. It contacts the supplied MDS server on the default port. If any input parameters has been specified, the method returns the information about all the web services and the related instances registered on the MDS server.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the hostname of the MDS server to be contacted
name: the logical name of the web service to be searched.

OUT parameters:

result: mdsInfo structure containing web services information.

4.4 Software

-int gridlab__register_software(xsd__string name, xsd__string version, xsd__string path, xsd__int totallicences, xsd__int freelicence, xsd__string licenceinfo, xsd__string startupenvironment, xsd__string executable, xsd__string arguments, xsd__string description, xsd__string helpURL, xsd__string usage, xsd__int *result)
This method allows the user to register a new software package; all of the input parameters are mandatory to describe and register the software package.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: software package name
version: package version number
path: pathname of the executable
totallicences: total number of available software licences
licenceinfo: human readable information about the licence
startupenvironment: a list, space-separated, of startup environment variables
executable: executable file name
arguments: list of arguments
description: human readable description of the software package
type: software type
helpURL: link to a help-desk URL
usage: a short description about executable's usage.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

If software is already registered the method will fail.

-int gridlab__unregister_software(xsd__string name, xsd__int *result)
This method removes an existing software package.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the name of the software package to be removed.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

If the software information does not exist the method sets result to 0 anyway.

-int gridlab__lookup_software(xsd__string mds_hostname, xsd__string name, struct mdsInfo **result)
This method returns all the information about the software packages. It contacts the supplied

MDS server on the default port. If any input parameters has been specified, the method returns the information about all the software registered on the MDS server.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the hostname of the MDS server to be contacted
name: the name of the software to be searched.

OUT parameters:

result: mdsInfo structure containing software packages information.

4.5 Firewall

-int gridlab__register_firewall(xsd__string hostname, xsd__string ports, xsd__string validitytime, xsd__int *result)
This method allows the registration of information about a firewall installed on the grid resource.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

hostname: host name of the machine where the firewall is installed
ports: space-separated list of open ports
validitytime: time frame during which the open ports are valid.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

-int gridlab__unregister_firewall(xsd__string hostname, xsd__int *result)
This method removes information about a registered firewall.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

hostname: host name of the machine where the firewall is installed; the corresponding information will be removed.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

If the firewall information does not exist the method sets result to 0 anyway.

-int gridlab__lookup_firewall(xsd__string mds_hostname, xsd__string name, struct mdsInfo **result)

This method returns all the information about the registered firewall. It contacts the supplied MDS server on the default port. If any input parameters has been specified, the method returns the information about all the firewall registered on the MDS server.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the host name of the MDS server to be contacted
name: the hostname of the firewall to be searched.

OUT parameters:

result: mdsInfo structure containing firewalls information.

4.6 Virtual Organization

-int gridlab__register_vo(xsd__string name, xsd__string helpDeskPN, xsd__string restype, xsd__string helpDeskURL, xsd__string adminname, xsd__int *result) This method allows the user to register a Virtual Organization ”owning” the grid resource.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: name of the VO
helpDeskPN: the Help-Desk Phone number of the Virtual Organization
restype: resource type of the Virtual Organization
helpDeskURL: the URL of the help-Desk
adminname: Distinguished name of the administrator.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

-int gridlab__unregister_vo(xsd__string name, xsd__int *result)
This method removes information about a Virtual Organization.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: name of the virtual organization whose information must be removed.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

If the VO information does not exist the method sets result to 0 anyway.

-int gridlab__lookup_vo(xsd__string mds_hostname, xsd__string name, struct mdsInfo **result)
This method returns the information about the Virtual Organization. It contacts the supplied MDS server on the default port. If any input parameters has been specified, the method returns the information about all the Virtual Organizations registered on the MDS server.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the host name of the MDS server to be contacted
name: the name of the Virtual Organization to be searched.

OUT parameters:

result: mdsInfo structure containing virtual organizations information.

4.7 User

-int gridlab__lookup_user(xsd__string mds_hostname, xsd__string name, struct mdsInfo **result)
This method returns all the information about the users. It contacts the supplied MDS server on the default port. If any input parameters has been specified, the method returns the information about all the users registered on the MDS server.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the host name of the MDS server to be contacted
name: the DN mapped on the user to be searched.

OUT parameters:

result: mdsInfo structure containing users information.

4.8 Certification Authority

-int gridlab__lookup_cert(xsd__string mds_hostname, xsd__string name, struct mdsInfo **result)
This method returns the information for an accepted Certification Authority. It contacts the supplied MDS server on the default port. If any input parameters has been specified, the method returns the information about all the accepted Certification Authority.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the host name of the MDS server to be contacted
name: the subject of the Certification Authority to be searched.

OUT parameters:

result: mdsInfo structure containing certificates information.

4.9 Local Resource Management System

-int gridlab__lrms_search(xsd__string mds_hostname, struct mdsInfo **result)
This method queries a specified GRIS hostname about some important lrms information.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mds_hostname: the host name of the MDS server to be contacted.

OUT parameters:

result: mdsInfo structure containing information about the lrms.

4.10 Genfunctions Library

The GridLab MDS package also includes a library with some functions useful to handle the data structure mdsInfo returned invoking a lookup web service. The library includes functions that filter information and format it in a given data structure.

This structure stores information about services.
struct mdsService

```
{
  char *name;    /* logical name of the service */
  char *host;    /* name of the machine that hosts the service */
  char *descr;   /* service description */
  char *keyw;    /* service keywords */
  char *dport;   /* service default port */
  char **accurl; /* service access URL */
  char **publ;   /* array of publisher's Distinguished Name */
};
```

This structure stores information about web services.
struct mdsWebService

```
{
  char *name;    /* logical name of the web service */
  char *host;    /* name of the machine that hosts the web service */
  char *descr;   /* web service description */
  char *keyw;    /* web service keywords */
  char **wsdl;   /* wsdl location URLs */
  char **accurl; /* web service access URLs */
  char **publ;   /* publisher's Distinguished Names */
  };
```

This structure stores information about software.
struct mdsSoftware

```
 {
  char *name;    /* name of application or software package */
  char *host;    /* hostname of machine where the application is installed on */
  char *vers;    /* version number of the software application */
  char *path;    /* pathname of the directory */
  char *totl;    /* total number of licence nstalled for the application software */
  char *linfo;   /* licence policy description */
  char **sten;   /* list of startup environment variables needed by the application */
  char *exec;    /* pathname of the executable file */
  char **argum;  /* list of arguments */
  char *descr;   /* application description */
  char *type;    /* the type of the software */
  char *hurl;    /* Help Desk URL */
  char *usage;   /* application input parameters usage */
};
```

This structure stores information about firewalls.
struct mdsFirewall

```
  {
  char *name;    /* FQDN of the machine where the firewall is installed */
  char *host;    /* hostname of the machine under the firewall */
  char **port;   /* list of open ports number */
  char *valt;    /* time frame during which open ports ranges are valid */
  char *admdn;   /* firewall administrator Distinguished Name */
};
```

This structure stores information about Virtual Organizations.
struct mdsVO

```
 {
  char *name;    /* name of the VO */
  char *host;    /* name of the machine that hosts the VO */
  char *rtype;   /* the type of resources */
  char **hdpn;   /* help-Desk Phone number of the VO */
  char *hdurl;   /* URL of the help-Desk */
  char *admdn;   /* distinguished name of the administrator */
};
```

This structure stores information about certificates.
struct mdsCert

```
{
  char *sub;     /* subject of the certificate */
  char *host;    /* name of the machine that hosts the VO */
  char *ver;     /* version of the certificate */
  char *snum;    /* serial number of the certificate */
  char *sign;    /* signature algorithm of the certificate */
  char *iss;     /* issuer of the certificate */
  char *valf;    /* beginning date of the certificate validity */
```

```
  char *valt;    /* end date of the certificate validity */
  char *pka;     /* public key algorithm of the certificate */
  char *pk;      /* RSA public key of the certification authority */
  char *crl;     /* URL where the Certificate Revocation List can be found */
};
```

This structure stores information about users.
struct mdsUser

```
 {
  char *usid;    /* user ID */
  char *host;    /* name of the machine that hosts the user */
  char **mdn;    /* Distinguished Name mapped on the user */
  char *hd;      /* user home directory */
  char *ush;     /* user shell */
  char *uid;     /* user UID */
  char *gid;     /* user GID */
  char *comm;    /* user comment */
};
```

This structure stores information about the local management system.
struct mdsLrms

```
  {
  char *host;    /* name of the machine that hosts the LRMS */
  char *cpud;    /* cpu distribution of the nodes given in the form of mcpu:nmachines */
  char *type;    /* type of the Local Resource Management System */
  char *vers;    /* version of the Local Resource Management System */
  char *ncpu;    /* cpu type of the nodes (model name + MHz) */
  char *nmem;    /* installed memory of a node in MB */
  char *totj;    /* total number of batch jobs in the cluster */
  char *fcpu;    /* number of free cpus in the cluster */
  char *totq;    /* total number of cluster queues */
  char **qname;  /* array of the queue's names */
  char **qstat;  /* array of queue status */
  char ***job;   /* list of jobs associated to each queue */
  char ***jobid; /* job identifier */
  char ***jobo;  /* subjectName of the job owner */
  char ***jobr;  /* cputime request of the job in minutes */
  char ***jobs;  /* status of the job */
  char ***jobst; /* submission time of the job */
};
```

The gridlab_service_translate function translates data about a service and its instances from the generic mdsInfo returned by gridlab__lookup_service to a specific mdsService structure.

int gridlab_service_translate(struct mdsInfo *result, struct mdsService ***ser_result);

Returns: the number of the GRIS server where the specified service has been found on success, -1 on error.

IN parameters:

result: gridlab__lookup_service result.

OUT parameters:

ser_result: an mdsService structure containing information about the service and its instances.

The gridlab_webservice_translate function translates data about a web service from the generic mdsInfo structure returned by gridlab__lookup_webservice to a specific mdsWebService structure.

int gridlab_webservice_translate(struct mdsInfo *result, struct mdsWebService ***ser_result);

Returns: the number of the GRIS server where the specified web service has been found on success, -1 on error.

IN parameters:

result: gridlab__lookup_webservice result.

OUT parameters:

ser_result: an mdsWebService structure containing information about the web service.

The gridlab_software_translate function translates data about a software package from the generic mdsInfo returned by gridlab__lookup_software to a specific mdsSoftware structure.

int gridlab_software_translate(struct mdsInfo *result, struct mdsSoftware ***ser_result);

Returns: the number of the GRIS server where the specified software has been found on success, -1 on error.

IN parameters:

result: gridlab__lookup_software result.

OUT parameters:

ser_result: an mdsSoftware structure containing information about the software package.

The gridlab_firewall_translate function translates data about the firewall from the generic mdsInfo structure returned by gridlab__lookup_firewall to a specific mdsFirewall structure.

```
int gridlab_firewall_translate(struct mdsInfo *result, struct mdsFirewall ***ser_result);
```

Returns: the number of the GRIS server where the specified firewall has been found on success, -1 on error.

IN parameters:

result: gridlab__lookup_firewall result.

OUT parameters:

ser_result: an mdsFirewall structure containing information about the registered firewall.

The gridlab_vo_translate function translates data about a virtual organization from the generic mdsInfo structure returned by gridlab__lookup_vo to a specific mdsVO structure.

```
int gridlab_vo_translate(struct mdsInfo *result, struct mdsVO ***ser_result);
```

Returns: the number of the GRIS server where the specified VO has been found on success, -1 on error.

IN parameters:

result: gridlab__lookup_vo result.

OUT parameters:

ser_result: an mdsVO structure containing information about the virtual organization.

The gridlab_user_translate function translates data about a user from the generic mdsInfo structure returned by gridlab__lookup_user to a specific mdsUser structure.

```
int gridlab_user_translate(struct mdsInfo *result, struct mdsUser ***ser_result);
```

Returns: the number of the GRIS server where the specified user has been found on success, -1 on error.

IN parameters:

result: gridlab__lookup_user result.

OUT parameters:

ser_result: an mdsUser structure containing information about the user.

The gridlab_cert_translate function translates data about a certificate from the generic mdsInfo structure returned by gridlab__lookup_cert to a specific mdsCert structure.

int gridlab_cert_translate(struct mdsInfo *result, struct mdsCert ***ser_result);

Returns: the number of the GRIS server where the specified certificate has been found on success, -1 on error.

IN parameters:

result: gridlab__lookup_cert result.

OUT parameters:

ser_result: an mdsCert structure containing information about the user.

The gridlab_lrms_translate function translates data about the lrms from the generic mdsInfo structure returned by gridlab__lrms_search to a specific mdsLrms structure.

int gridlab_lrms_translate(struct mdsInfo *result, struct mdsLrms **ser_result);

Returns: 1 on success, -1 on error.

IN parameters:

result: gridlab__lrms_search result.

OUT parameters:

ser_result: an mdsLrms structure containing information about the lrms.

Appendix

We show here the WSDL document related to the GridLab MDS web service. Please note that this WDSL document is just a reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="mds"
 xmlns="http://schemas.xmlsoap.org/wsdl/"
 targetNamespace="urn:mds"
 xmlns:tns="urn:mds"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:gridlab="urn:mds"
 xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/">

<types>
 <schema targetNamespace="urn:mds"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:gridlab="urn:mds"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified">
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
  <simpleType name="scopes">
   <restriction base="xsd:string">
    <enumeration value="LDAP-SCOPE-BASE"/>
    <enumeration value="LDAP-SCOPE-ONE"/>
    <enumeration value="LDAP-SCOPE-SUB"/>
   </restriction>
  </simpleType>

  <complexType name="ArrayOfstring">
   <complexContent>
    <restriction base="SOAP-ENC:Array">
     <sequence>
      <element name="item" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
     </sequence>
     <attribute ref="SOAP-ENC:arrayType" WSDL:arrayType="xsd:string[]"/>
    </restriction>
   </complexContent>
  </complexType>

  <complexType name="ArrayOfstring">
```

```
   <complexContent>
    <restriction base="SOAP-ENC:Array">
     <sequence>
      <element name="item" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
     </sequence>
     <attribute ref="SOAP-ENC:arrayType" WSDL:arrayType="xsd:string[]"/>
    </restriction>
   </complexContent>
  </complexType>

  <complexType name="ArrayOfArrayOfstring">
   <complexContent>
    <restriction base="SOAP-ENC:Array">
     <sequence>
      <element name="item" type="gridlab:ArrayOfstring" minOccurs="0" maxOccurs="unbounded"/
     </sequence>
     <attribute ref="SOAP-ENC:arrayType" WSDL:arrayType="gridlab:ArrayOfstring[]"/>
    </restriction>
   </complexContent>
  </complexType>

  <complexType name="ArrayOfArrayOfArrayOfstring">
   <complexContent>
    <restriction base="SOAP-ENC:Array">
     <sequence>
      <element name="item" type="gridlab:ArrayOfArrayOfstring" minOccurs="0" maxOccurs="unbo
     </sequence>
     <attribute ref="SOAP-ENC:arrayType" WSDL:arrayType="gridlab:ArrayOfArrayOfstring[]"/>
    </restriction>
   </complexContent>
  </complexType>

 </schema>
</types>

<message name="register-serviceRequest">
 <part name="name" type="xsd:string"/>
 <part name="hostname" type="xsd:string"/>
 <part name="port" type="xsd:int"/>
 <part name="protocol" type="xsd:string"/>
 <part name="dport" type="xsd:int"/>
 <part name="description" type="xsd:string"/>
 <part name="keywords" type="xsd:string"/>
</message>

<message name="register-serviceResponse">
 <part name="result" type="xsd:int"/>
</message>
```

```
<message name="unregister-serviceRequest">
 <part name="name" type="xsd:string"/>
 <part name="hostname" type="xsd:string"/>
 <part name="port" type="xsd:int"/>
 <part name="allinst" type="xsd:int"/>
</message>

<message name="unregister-serviceResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="lookup-serviceRequest">
 <part name="mds-hostname" type="xsd:string"/>
 <part name="name" type="xsd:string"/>
 <part name="dport" type="xsd:int"/>
 <part name="sport" type="xsd:int"/>
 <part name="prot" type="xsd:string"/>
</message>

<message name="lookup-serviceResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="register-webserviceRequest">
 <part name="name" type="xsd:string"/>
 <part name="wsdllocation" type="xsd:string"/>
 <part name="description" type="xsd:string"/>
 <part name="url" type="xsd:string"/>
 <part name="keywords" type="xsd:string"/>
</message>

<message name="register-webserviceResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="unregister-webserviceRequest">
 <part name="name" type="xsd:string"/>
 <part name="url" type="xsd:string"/>
 <part name="allinst" type="xsd:int"/>
</message>

<message name="unregister-webserviceResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="lookup-webserviceRequest">
 <part name="mds-hostname" type="xsd:string"/>
 <part name="name" type="xsd:string"/>
</message>
```

```
<message name="lookup-webserviceResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="searchRequest">
 <part name="mds-hostname" type="xsd:string"/>
 <part name="port" type="xsd:int"/>
 <part name="base-dn" type="xsd:string"/>
 <part name="scope" type="gridlab:scopes"/>
 <part name="attributes" type="xsd:string"/>
 <part name="filter" type="xsd:string"/>
</message>

<message name="searchResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="register-softwareRequest">
 <part name="name" type="xsd:string"/>
 <part name="version" type="xsd:string"/>
 <part name="path" type="xsd:string"/>
 <part name="totallicence" type="xsd:int"/>
 <part name="licenceinfo" type="xsd:string"/>
 <part name="startupenvironment" type="xsd:string"/>
 <part name="executable" type="xsd:string"/>
 <part name="arguments" type="xsd:string"/>
 <part name="description" type="xsd:string"/>
 <part name="type" type="xsd:string"/>
 <part name="helpURL" type="xsd:string"/>
 <part name="usage" type="xsd:string"/>
</message>

<message name="register-softwareResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="unregister-softwareRequest">
 <part name="name" type="xsd:string"/>
</message>

<message name="unregister-softwareResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="register-firewallRequest">
 <part name="hostname" type="xsd:string"/>
 <part name="ports" type="xsd:string"/>
 <part name="validity-time" type="xsd:string"/>
```

```
</message>

<message name="register-firewallResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="lookup-softwareRequest">
 <part name="mds-hostname" type="xsd:string"/>
 <part name="name" type="xsd:string"/>
</message>

<message name="lookup-softwareResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="unregister-firewallRequest">
 <part name="hostname" type="xsd:string"/>
</message>

<message name="unregister-firewallResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="lookup-firewallRequest">
 <part name="mds-hostname" type="xsd:string"/>
 <part name="name" type="xsd:string"/>
</message>

<message name="lookup-firewallResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="register-voRequest">
 <part name="name" type="xsd:string"/>
 <part name="helpDeskPN" type="xsd:string"/>
 <part name="restype" type="xsd:string"/>
 <part name="helpDeskURL" type="xsd:string"/>
 <part name="adminname" type="xsd:string"/>
</message>

<message name="register-voResponse">
 <part name="result" type="xsd:int"/>
</message>

<message name="unregister-voRequest">
 <part name="name" type="xsd:string"/>
</message>

<message name="unregister-voResponse">
```

```
 <part name="result" type="xsd:int"/>
</message>

<message name="lookup-voRequest">
 <part name="mds-name" type="xsd:string"/>
 <part name="name" type="xsd:string"/>
</message>

<message name="lookup-voResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="lookup-userRequest">
 <part name="mds-name" type="xsd:string"/>
 <part name="name" type="xsd:string"/>
</message>

<message name="lookup-userResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="lookup-certRequest">
 <part name="mds-name" type="xsd:string"/>
 <part name="name" type="xsd:string"/>
</message>

<message name="lookup-certResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<message name="getServiceDescriptionRequest">
</message>

<message name="getServiceDescriptionResponse">
 <part name="result" type="xsd:string"/>
</message>

<message name="lrms-searchRequest">
 <part name="mds-name" type="xsd:string"/>
</message>

<message name="lrms-searchResponse">
 <part name="result" type="gridlab:ArrayOfArrayOfArrayOfstring"/>
</message>

<portType name="mdsPortType">
 <operation name="register-service">
  <documentation>Service definition of function gridlab__register_service</documentation>
  <input message="tns:register-serviceRequest"/>
```

```
 <output message="tns:register-serviceResponse"/>
</operation>
<operation name="unregister-service">
 <documentation>Service definition of function gridlab__unregister_service</documentation>
 <input message="tns:unregister-serviceRequest"/>
 <output message="tns:unregister-serviceResponse"/>
</operation>
<operation name="lookup-service">
 <documentation>Service definition of function gridlab__lookup_service</documentation>
 <input message="tns:lookup-serviceRequest"/>
 <output message="tns:lookup-serviceResponse"/>
</operation>
<operation name="register-webservice">
 <documentation>Service definition of function gridlab__register_webservice</documentation>
 <input message="tns:register-webserviceRequest"/>
 <output message="tns:register-webserviceResponse"/>
</operation>
<operation name="unregister-webservice">
 <documentation>Service definition of function gridlab__unregister_webservice</documentatio
 <input message="tns:unregister-webserviceRequest"/>
 <output message="tns:unregister-webserviceResponse"/>
</operation>
<operation name="lookup-webservice">
 <documentation>Service definition of function gridlab__lookup_webservice</documentation>
 <input message="tns:lookup-webserviceRequest"/>
 <output message="tns:lookup-webserviceResponse"/>
</operation>
<operation name="search">
 <documentation>Service definition of function gridlab__search</documentation>
 <input message="tns:searchRequest"/>
 <output message="tns:searchResponse"/>
</operation>
<operation name="register-software">
 <documentation>Service definition of function gridlab__register_software</documentation>
 <input message="tns:register-softwareRequest"/>
 <output message="tns:register-softwareResponse"/>
</operation>
<operation name="unregister-software">
 <documentation>Service definition of function gridlab__unregister_software</documentation>
 <input message="tns:unregister-softwareRequest"/>
 <output message="tns:unregister-softwareResponse"/>
</operation>
<operation name="register-firewall">
 <documentation>Service definition of function gridlab__register_firewall</documentation>
 <input message="tns:register-firewallRequest"/>
 <output message="tns:register-firewallResponse"/>
</operation>
<operation name="lookup-software">
 <documentation>Service definition of function gridlab__lookup_software</documentation>
```

```
  <input message="tns:lookup-softwareRequest"/>
  <output message="tns:lookup-softwareResponse"/>
 </operation>
 <operation name="unregister-firewall">
  <documentation>Service definition of function gridlab__unregister_firewall</documentation>
  <input message="tns:unregister-firewallRequest"/>
  <output message="tns:unregister-firewallResponse"/>
 </operation>
 <operation name="lookup-firewall">
  <documentation>Service definition of function gridlab__lookup_firewall</documentation>
  <input message="tns:lookup-firewallRequest"/>
  <output message="tns:lookup-firewallResponse"/>
 </operation>
 <operation name="register-vo">
  <documentation>Service definition of function gridlab__register_vo</documentation>
  <input message="tns:register-voRequest"/>
  <output message="tns:register-voResponse"/>
 </operation>
 <operation name="unregister-vo">
  <documentation>Service definition of function gridlab__unregister_vo</documentation>
  <input message="tns:unregister-voRequest"/>
  <output message="tns:unregister-voResponse"/>
 </operation>
 <operation name="lookup-vo">
  <documentation>Service definition of function gridlab__lookup_vo</documentation>
  <input message="tns:lookup-voRequest"/>
  <output message="tns:lookup-voResponse"/>
 </operation>
 <operation name="lookup-user">
  <documentation>Service definition of function gridlab__lookup_user</documentation>
  <input message="tns:lookup-userRequest"/>
  <output message="tns:lookup-userResponse"/>
 </operation>
 <operation name="lookup-cert">
  <documentation>Service definition of function gridlab__lookup_cert</documentation>
  <input message="tns:lookup-certRequest"/>
  <output message="tns:lookup-certResponse"/>
 </operation>
 <operation name="getServiceDescription">
  <documentation>Service definition of function gridlab__getServiceDescription</documentatio
  <input message="tns:getServiceDescriptionRequest"/>
  <output message="tns:getServiceDescriptionResponse"/>
 </operation>
 <operation name="lrms-search">
  <documentation>Service definition of function gridlab__lrms_search</documentation>
  <input message="tns:lrms-searchRequest"/>
  <output message="tns:lrms-searchResponse"/>
 </operation>
</portType>
```

```
<binding name="mds" type="tns:mdsPortType">
 <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
 <operation name="register-service">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="unregister-service">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="lookup-service">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="register-webservice">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="unregister-webservice">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="lookup-webservice">
```

```
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="search">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="register-software">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="unregister-software">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="register-firewall">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="lookup-software">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
```

```
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="unregister-firewall">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="lookup-firewall">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="register-vo">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="unregister-vo">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
<operation name="lookup-vo">
 <SOAP:operation soapAction=""/>
 <input>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </input>
 <output>
  <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
 </output>
</operation>
```

```
 <operation name="lookup-user">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="lookup-cert">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="getServiceDescription">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
 <operation name="lrms-search">
  <SOAP:operation soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:mds" encodingStyle="http://schemas.xmlsoap.org/so
  </output>
 </operation>
</binding>

<service name="mds">
 <documentation>gSOAP 2.4.1 generated service definition</documentation>
 <port name="mds" binding="tns:mds">
  <SOAP:address location="httpg://gridsurfer.unile.it:21000"/>
 </port>
</service>

</definitions>
```