



IST-2001-32133

GridLab - A Grid Application Toolkit and Testbed

GridLab iGrid Release 1.0

Author(s):	Giovanni Aloisio, Massimo Cafaro, Italo Epicoco, Daniele Lezzi, Maria Mirto, Silvia Mocavero
Document Filename:	GridLab-10-D.5-0001-IGRID
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-D.5-0001-IGRID
Document classification:	INTERNAL

Abstract: This document describes the release 1.0 of GridLab iGrid, a relational information service. The documentation covers the GridLab iGrid relational schema, related information providers, in C language, to retrieve information about the system and a web service to publish information provided by user and to retrieve all of the information stored in iGrid DBMS.





Contents

1	Introduction	2
2	Installation	3
2.1	Installation of GridLab iGrid	3
3	GridLab iGrid relational schema	7
3.1	Information provided by the user (user info)	7
3.2	Information provided by the information providers (system info)	9
3.3	Information model	13
4	GridLab iGrid web service	29
4.1	Search	30
4.2	Services	30
4.3	Web Services	32
4.4	Firewall	33
4.5	Virtual Organization	35
4.6	User	37
4.7	Certification Authority	37
4.8	Local Resource Management System	37
4.9	CPU	38
4.10	Device	38
4.11	Memory	39
4.12	Network	39
4.13	Operative System	40

1 Introduction

This document is associated to the prototype release of the GridLab iGrid version 1.0. The GridLab iGrid has been designed as a relational Information Service for discovering and monitoring static and dynamic information related to resources belonging to a computational grid, taking into account the requirements suggested by others GridLab WPs. iGrid architecture is based on two servers fulfilling different roles: the iServe server gathers local information (like the GRIS server in the Globus Toolkit MDS), while the iStore server collects information related to all of registered iServes (like the GIIS server in the Globus Toolkit MDS). The iStore servers can be organized on different layers, according to a hierarchical structure.

The GRelC technology (Grid Relational Catalog) provides a set of primitives to directly establish and manage a connection with data sources. Moreover, its MultiQuery library is used to insert (in one single shot) a huge amount of data (generated by the Information Providers) in the iGrid Relational Information System. Using the MultiQuery it is possible to:

1. compute in one single shot many insert/update/delete queries (indeed in the same MultiQuery XML file it is possible to specify information related to several records of multiple tables);
2. reduce the interactions between Information Providers and the iGrid Server (indeed for a MultiQuery submission there is only one feedback, whereas for N queries there are N feedbacks);
3. transfer on the iGrid Server side the entire computation (reducing the connection time between Information Providers and iGrid Server and speeding up the entire submission process).

The system is able to store two kind of information:

1. user supplied information which requires user/resource interaction to publish information;
2. system information which are directly provided by the system.

The security mechanism is based on the Globus GSI infrastructure; the iGrid Web service has been secured using our GSI plug-in for gSOAP technology. The prototype release includes:

- GridLab iGrid relational schema dump;
- information providers to retrieve static and dynamic information from the system;
- a set of functionalities, developed as a web service, to publish and retrieve information that will be stored in the relational iGrid DBMS.

The document is organized as follows: in section 2 we give the instructions needed to configure, build and install the GridLab iGrid system (iGrid db, web service and providers); in section 3 we describe the information stored in the iGrid DBMS 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 iGrid relational Schema and related system information providers; a web service and related sample clients are also included. To build and install the package please follow these installation instructions.

2.1 Installation of GridLab iGrid

Required software:

Globus Toolkit v3.2.1 pre-ogsi
gSOAP Toolkit v2.6.2
GSI plugin for gSOAP v2.3
Libxml2
postgresql v.7.4.3

Moreover, the following information providers require the library libgtop v2.6.0

- grid-info-cpu_d
- grid-info-mem_d
- grid-info-mem_s
- grid-info-dev_s
- grid-info-dev_d

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

a) Installation of iGrid DB on postgresSQL as postgres user.

a.1) Create the "iGrid" DB user (with permission to create database only)

```
createuser iGrid
```

a.2) Create the "iGrid" DB user

```
createdb <dbname> -O iGrid
```

a.3) Append a new entry in the pg_hba.conf in the directory data located in the postgresQL install dir

#	TYPE	DATABASE	USER	IP-ADDRESS	IP-MASK	METHOD
	local	<dbname>	iGrid			trust

a.4) Restart the postgresQL server

a.5) Set the "iGrid" DB user password

```
psql <dbname> -U iGrid  
ALTER USER iGrid WITH PASSWORD '<password>';
```

a.6) Modify previously inserted entry in the pg_hba.conf

#	TYPE	DATABASE	USER	IP-ADDRESS	IP-MASK	METHOD
	local	<dbname>	iGrid			password

a.7) Restart the postgresQL server

a.8) Import as postgres user the iGrid relational schema ($\{\text{GRIDLAB_LOCATION}\}/\text{schema}/\text{iGrid.sql}$) and write the result of the dump in a log file

```
psql <dbname> -e < <iGrid.sql full path> 2>&1 | tee iGrid-importing-log
```

b) Installation of iGrid package

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

The GridLab iGrid package requires the following environmental variables:

HOSTNAME: the FQDN name of the host

GRIDLAB_LOCATION: the gridlab iGrid 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

IGRID_CONF_FILENAME: pathname of the Gridlab iGrid configuration file; usually should be `/${GRIDLAB_LOCATION}/etc/igrid.conf.xml`

The HOSTNAME variable in the gridlab.conf file must be set to the fully qualified domain name.

Modify the iGrid configuration file according to your environment

The GridLab iGrid package requires the following variables:

SystemInfo_SD: pathname (relative to the iGrid server running directory or absolute) of the system information spool directory

UserInfo_SD: pathname (relative to the iGrid server running directory or absolute) of the user information spool directory

iNodeInfo_SD: pathname (relative to the iGrid server running directory or absolute) of the iStore information spool directory

RegistrationPeriod: how much frequently (in seconds) new iServe host information is transferred to the iStore hosts the iServe host is registered to.

PurgePeriod: time frame (in seconds) during which an iStore the iServe is registered to, is temporarily removed from the local iStore registration list. This happens automatically when a registered iStore can not be contacted. After this time frame has expired, an iStore previously deleted is automatically reinserted in the registration list.

GrelC_Db_Name: name of iGrid DB (specified in a.2).

GrelC_Db_Port: TCP port of postgresSQL DBMS as specified in the postgresql.conf in the directory data located in the postgresSQL install dir.

GrelC_Db_Password: password (clear-text, this release only) of the iGrid DB user (specified in a.5).

The configuration file contains a list of local information providers characterized by:

Provider name: name of the provider.

Provider type: type of the provider.

Module: pathname (relative to the iGrid server running directory or absolute) of the information provider.

Arguments: arguments for the execution of the information provider.

InfoTTL: validity time of information generated by the information provider.

Moreover, the configuration file contains a list of the hosts allowed to contact the local iGrid web service.

iNode_DN: Distinguished Name of the host allowed to contact the local iGrid web service.

At the end of the file there is a list of the iStore hosts the iServe is registered to.

iStore_WS: access url of the igrid web service running on the iStore.

b.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-listening-port
```

This option sets the listening port for the 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 iGrid package installation on `$GRIDLAB_LOCATION/igrid`.

```
--with-conf-filename
```

This option sets the pathname of the iGrid configuration file. This option override the value specified in the `gridlab.conf` file; defaults to `etc/igrid.conf.xml`. Please note that this pathname is not an absolute one, it is relative to the iGrid package installation on `$GRIDLAB_LOCATION/igrid`.

b.3) configure

```
configure
```

This will configure the package using default options.

b.4) Build

```
make
```

b.5) Install everything as user with read/write permissions to the `$PREFIX` directory

```
make install
```

b.6) Run the server using a standard user account. 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) cd $GRIDLAB_LOCATION/igrid/server b) nohup ./igrid &
```

3 GridLab iGrid relational schema

This section describes the developed information schema. 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 prototype release version 1.0 provides information related to:

- User supplied information.
 - Services;
 - Web Services;
 - Firewalls;
 - Virtual Organizations;
- System information.
 - Network interfaces;
 - Devices;
 - Operative System;
 - Memories;
 - Cpus;
 - Users;
 - Trusted Certification Authorities;
 - Local Resource Management Systems.

System information is directly provided by related information providers. User information is provided by the user through the GridLab iGrid web service. This can be done, for instance, using the `gridlab-*-register-client` programs. It is worth noting here that our system is able to handle all of the character set specified in rfc1738. Since many characters are not supported by the PostgreSQL DBMS, an escape sequence is generated for each character included in rfc1738 but not admitted in PostgreSQL. Using the `gridlab_search` or `gridlab_*_lookup` methods, each escape sequence is properly converted into the corresponding character upon return.

We now briefly review the attributes of the GridLab iGrid schema.

3.1 Information provided by the user (user info)

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 and web services dynamically. The iGrid system provides GridLab developers with the following functionalities: registration, unregistration, update and lookup. More than one instance for each Service or Web Service can be registered. The following information belongs to these categories:

- Service information.

- GridLab-iGrid-Service-id: service identification number;
- GridLab-iGrid-Service-name: service logical name;
- GridLab-iGrid-Service-description: service description;
- GridLab-iGrid-Service-keywords: set of key words of a service;
- GridLab-iGrid-Service-defaultport: service default port.
- Service instance information.
 - GridLab-iGrid-Service-accessurl: service access URL;
 - GridLab-iGrid-Service-publisher: service publisher (X509v3 certificate distinguished name);
 - GridLab-iGrid-Service-creationdate: creation date of the service instance information;
 - GridLab-iGrid-Service-validitytime: validity time of the service instance information.
- Web Service information.
 - GridLab-iGrid-WebService-id: web service identification number;
 - GridLab-iGrid-WebService-name: web service logical name;
 - GridLab-iGrid-WebService-description: web service description;
 - GridLab-iGrid-WebService-keywords: set of key words of a web service;
 - GridLab-iGrid-WebService-wsdlloc: (multivalue) URL where the WSDL document for the Web Service can be found.
- Web Service instance information.
 - GridLab-iGrid-WebService-accessurl: web service access URL;
 - GridLab-iGrid-WebService-publisher: web service publisher (X509v3 certificate distinguished name);
 - GridLab-iGrid-WebService-creationdate: creation date of the web service instance information;
 - GridLab-iGrid-WebService-validitytime: validity time of the web service instance information.

Information related to firewalls is strictly related to service information. As a matter of fact, before registering a service, GridLab developers will query iGrid 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 information belongs to this category:

- Firewall information.
 - GridLab-iGrid-Firewall-id: firewall identification number;
 - GridLab-iGrid-Firewall-hostname: firewall hostname;
 - GridLab-iGrid-Firewall-ports: (multivalue) attribute composed by the open ports (GridLab-iGrid-Firewall-port), the time frame during which each port (or a range of ports) is open (GridLab-iGrid-Firewall-firvaliditytime) and the protocol (TCP/UDP) used to connect to firewall ports (GridLab-iGrid-Firewall-prot);
 - GridLab-iGrid-Firewall-adminDN: distinguished name of the firewall administrator;

- GridLab-iGrid-Firewall-publisher: firewall publisher (X509v3 certificate distinguished name);
- GridLab-iGrid-Firewall-creationdate: creation date of the firewall information;
- GridLab-iGrid-Firewall-validityTime: validity time of the firewall information.

The GridLab project spans 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 information belongs to this category:

- Virtual Organization information.
 - GridLab-iGrid-Vo-id: Virtual Organization identification number;
 - GridLab-iGrid-Vo-name: Virtual Organization to which a specified computational resource belongs to;
 - GridLab-iGrid-Vo-resourceType: Virtual Organization resource type;
 - GridLab-iGrid-Vo-helpDeskPhoneNumber: (multivalue) help desk phone number;
 - GridLab-iGrid-Vo-helpDeskURL: URL pointing to a Virtual Organization's web page;
 - GridLab-iGrid-Vo-adminName: administrator name of the VO;
 - GridLab-iGrid-Vo-jobmanager: jobmanager used by the Virtual Organization;
 - GridLab-iGrid-Vo-queue: job queue of the Virtual Organization;
 - GridLab-iGrid-Vo-fileSystem: (multivalue) available file systems composed by the type (GridLab-iGrid-Vo-type) and the pathname (GridLab-iGrid-Vo-pathname);
 - GridLab-iGrid-Vo-publisher: publisher (X509v3 certificate distinguished name) of the Virtual Organization;
 - GridLab-iGrid-Vo-creationdate: creation date of the Virtual Organization information;
 - GridLab-iGrid-Vo-validityTime: validity time of the Virtual Organization information.

3.2 Information provided by the information providers (system info)

System information allows complex brokering strategies: for instance, once the set of computing resources available to a user and their features are known to a broker, it is possible to choose carefully where to submit a user's job. The broker's decision will be based on the information gathered from the iGrid system and on the job's requirements. Moreover, accessing local resource management information is crucial for resource management, brokering strategies etc. The current release provides support for the PBS local resource management system. The following information belongs to system info category:

- Host information.
 - GridLab-iGrid-Host-id: Host identification number;
 - GridLab-iGrid-Host-hostname: the FQDN of the Host;
 - GridLab-iGrid-Host-domainname: the domain name of the Host;
- Network information.

- GridLab-iGrid-Net-name: Network name;
- GridLab-iGrid-Net-address: Network address;
- GridLab-iGrid-Net-mask: Network mask;
- GridLab-iGrid-Net-creationdate: creation date of the Network information;
- GridLab-iGrid-Net-validityTime: validity time of the Network information.
- Device dynamic information.
 - GridLab-iGrid-Devdyn-name: Device name;
 - GridLab-iGrid-Devdyn-freespace: Device free space;
 - GridLab-iGrid-Devdyn-creationdate: creation date of the dynamic Device information;
 - GridLab-iGrid-Devdyn-validityTime: validity time of the dynamic Device information.
- Device static information.
 - GridLab-iGrid-Devsta-name: Device name;
 - GridLab-iGrid-Devsta-mountpoint: Device mount point;
 - GridLab-iGrid-Devsta-size: Device size;
 - GridLab-iGrid-Devsta-accessright: Device access right;
 - GridLab-iGrid-Devsta-fstype: filesystem type of the Device;
 - GridLab-iGrid-Devsta-creationdate: creation date of the static Device information;
 - GridLab-iGrid-Devsta-validityTime: validity time of the static Device information.
- Operative system information.
 - GridLab-iGrid-System-id: Operative System identification number;
 - GridLab-iGrid-System-name: Operative System name;
 - GridLab-iGrid-System-version: Operative System version;
 - GridLab-iGrid-System-release: Operative System release;
 - GridLab-iGrid-System-machine: host platform;
 - GridLab-iGrid-System-creationdate: creation date of the Operative System information;
 - GridLab-iGrid-System-validityTime: validity time of the Operative System information.
- Memory dynamic information.
 - GridLab-iGrid-Memdyn-freeram: free ram space;
 - GridLab-iGrid-Memdyn-freeswap: free swap space;
 - GridLab-iGrid-Memdyn-creationdate: creation date of the dynamic Memory information;
 - GridLab-iGrid-Memdyn-validityTime: validity time of the dynamic Memory information.

- Memory static information.
 - GridLab-iGrid-Memsta-totalram: total ram space;
 - GridLab-iGrid-Memsta-totalswap: total swap space;
 - GridLab-iGrid-Memsta-creationdate: creation date of the static Memory information;
 - GridLab-iGrid-Memsta-validityTime: validity time of the static Memory information.
- Cpu dynamic information.
 - GridLab-iGrid-Cpudyn-number: Cpu internal identification number;
 - GridLab-iGrid-Cpudyn-load: Cpu load;
 - GridLab-iGrid-Cpudyn-uptime: Cpu user time;
 - GridLab-iGrid-Cpudyn-nicetime: Cpu nice time;
 - GridLab-iGrid-Cpudyn-systemtime: Cpu system time;
 - GridLab-iGrid-Cpudyn-idletime: Cpu idle time;
 - GridLab-iGrid-Cpudyn-creationdate: creation date of the dynamic Cpu information;
 - GridLab-iGrid-Cpudyn-validityTime: validity time of the dynamic Cpu information.
- Cpu static information.
 - GridLab-iGrid-Cpusta-id: Cpu identification number;
 - GridLab-iGrid-Cpusta-vendor: Cpu vendor;
 - GridLab-iGrid-Cpusta-version: Cpu version;
 - GridLab-iGrid-Cpusta-model: Cpu model;
 - GridLab-iGrid-Cpusta-cpumhz: Cpu Mhz;
 - GridLab-iGrid-Cpusta-cachesize: cache size;
 - GridLab-iGrid-Cpusta-features: Cpu features;
 - GridLab-iGrid-Cpusta-number: number of Cpu belonging to the host;
 - GridLab-iGrid-Cpusta-creationdate: creation date of the static Cpu information;
 - GridLab-iGrid-Cpusta-validityTime: validity time of the static Cpu information.
- User information.
 - GridLab-iGrid-User-id: User identification number;
 - GridLab-iGrid-User-login: User login;
- Distinguished Name information.
 - GridLab-iGrid-Dn-id: Distinguished Name identification number;
 - GridLab-iGrid-Dn-dn: Distinguished Name subject;

Each Host-User pair is characterized by the following information:

- Host-User information.
 - GridLab-iGrid-DnUsHo-path: user's home directory path on the host;
 - GridLab-iGrid-DnUsHo-shell: user's shell on the host;

- GridLab-iGrid-DnUsHo-uid: user’s UID on the host;
 - GridLab-iGrid-DnUsHo-creationdate: creation date of the tern Host-DN-User information;
 - GridLab-iGrid-DnUsHo-validityTime: validity time of the tern Host-DN-User information.
- Certification Authority information.
 - GridLab-iGrid-Cert-id: recognized CA identification number;
 - GridLab-iGrid-Cert-Subj: distinguished name of the recognized CA;
 - GridLab-iGrid-Cert-version: CA’s certificate version;
 - GridLab-iGrid-Cert-serialNumber: CA’s certificate serial number;
 - GridLab-iGrid-Cert-signatureAlgorithm: CA’s certificate signature algorithm;
 - GridLab-iGrid-Cert-issuer: CA’s certificate issuer;
 - GridLab-iGrid-Cert-validity-from: beginning date of the CA’s certificate validity;
 - GridLab-iGrid-Cert-validity-to: end date of the CA’s certificate validity;
 - GridLab-iGrid-Cert-publicKeyAlgorithm: CA’s certificate public key algorithm;
 - GridLab-iGrid-Cert-RSAPublicKey: CA’s certificate RSA public key;
 - GridLab-iGrid-Cert-crlURL: url where the CA CRL can be found.
 - GridLab-iGrid-Cert-creationdate: creation date of the CA information;
 - GridLab-iGrid-Cert-validityTime: validity time of the CA information.
 - Local Resource Management System information.
 - GridLab-iGrid-Lrms-id: Local Resource Management System identification number;
 - GridLab-iGrid-Lrms-type: the type of Local Resource Management System;
 - GridLab-iGrid-Lrms-version: the version of Local Resource Management System;
 - GridLab-iGrid-Lrms-creationdate: creation date of the Lrms information;
 - GridLab-iGrid-Lrms-validityTime: validity time of the Lrms information.

Each Local Resource Management System can manage one or more queues. Related information includes:

- Queue information.
 - GridLab-iGrid-Queue-id: the identification number of the Queue;
 - GridLab-iGrid-Queue-name: the name of the Queue;
 - GridLab-iGrid-Queue-assignedcpunumber: the number of cpus assigned to the Queue;
 - GridLab-iGrid-Queue-status: the Queue status;
 - GridLab-iGrid-Queue-maxqueueable: the max number of jobs allowed to reside in the Queue;
 - GridLab-iGrid-Queue-maxrunning: the max number of jobs allowed to run from this Queue;
 - GridLab-iGrid-Queue-queued: the number of jobs waiting in the Queue;

- GridLab-iGrid-Queue-running: the number of running jobs in the cluster belonging to this Queue;
- GridLab-iGrid-Queue-maxwallclocktime: the max wall clock time allowed for jobs submitted to the Queue in mins;
- GridLab-iGrid-Queue-maxcputime: the max CPU time allowed for jobs submitted to the Queue in mins;

One or more job can be assigned to each queue. Related information includes:

- Job information.
 - GridLab-iGrid-Job-id: the identification number of the Job;
 - GridLab-iGrid-Job-name: the name of the Job;
 - GridLab-iGrid-Job-jobid: the Job identifier string;
 - GridLab-iGrid-Job-owner: the SubjectName of the Job owner;
 - GridLab-iGrid-Job-reqcput: the cputime request of the Job in minutes;
 - GridLab-iGrid-Job-status: the status of the Job;
 - GridLab-iGrid-Job-submissiontime: the submission time of the Job;
 - GridLab-iGrid-Job-usedcputime: the consumed cputime of the Job in minutes;
 - GridLab-iGrid-Job-usedmem: the memory usage of the Job (in KB);
 - GridLab-iGrid-Job-usedwalltime: the consumed walltime of the Job in minutes;
 - GridLab-iGrid-Job-estimatedresponsetime: estimated time between Job submission and execution in sec.

3.3 Information model

In this section we detail the GridLab iGrid information schema using Entity Relationship diagrams. The following figures describe the entities belonging to the current iGrid schema.

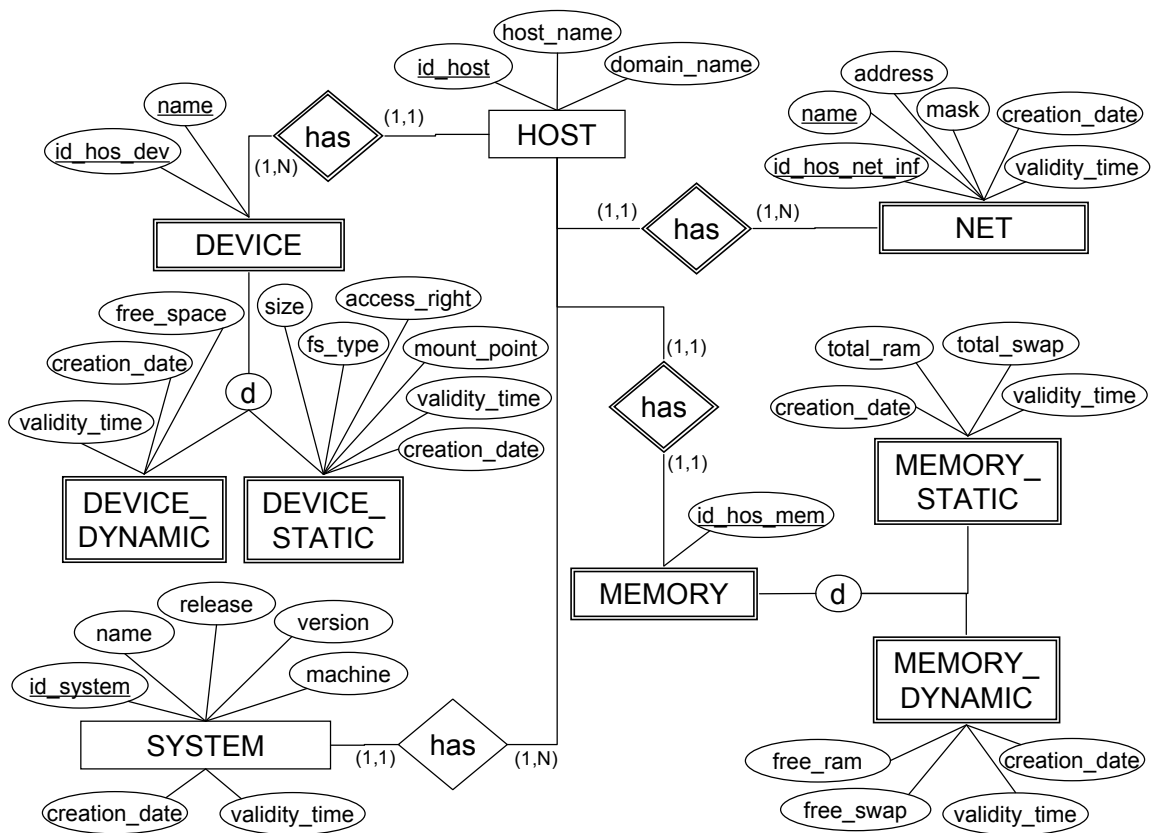


Figure 1: iGrid Entity Relationship Diagram

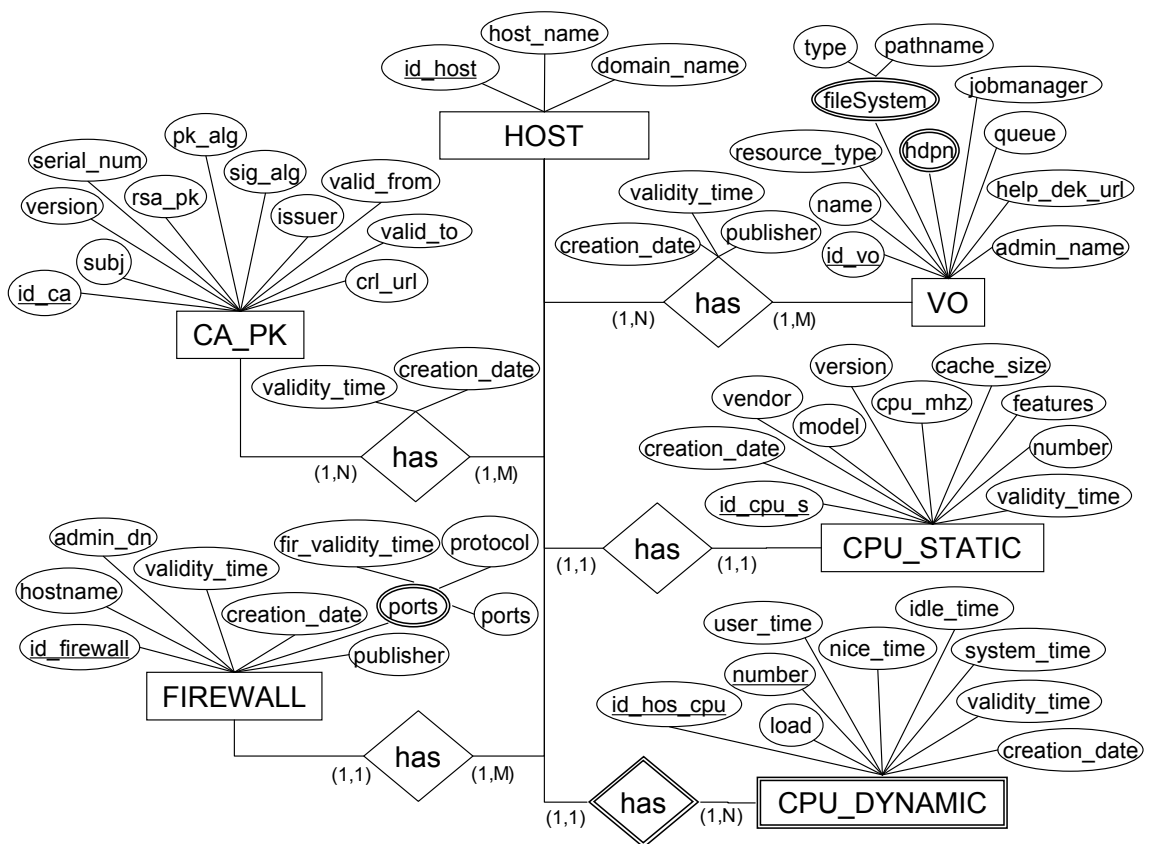


Figure 2: iGrid Entity Relationship Diagram (continued)

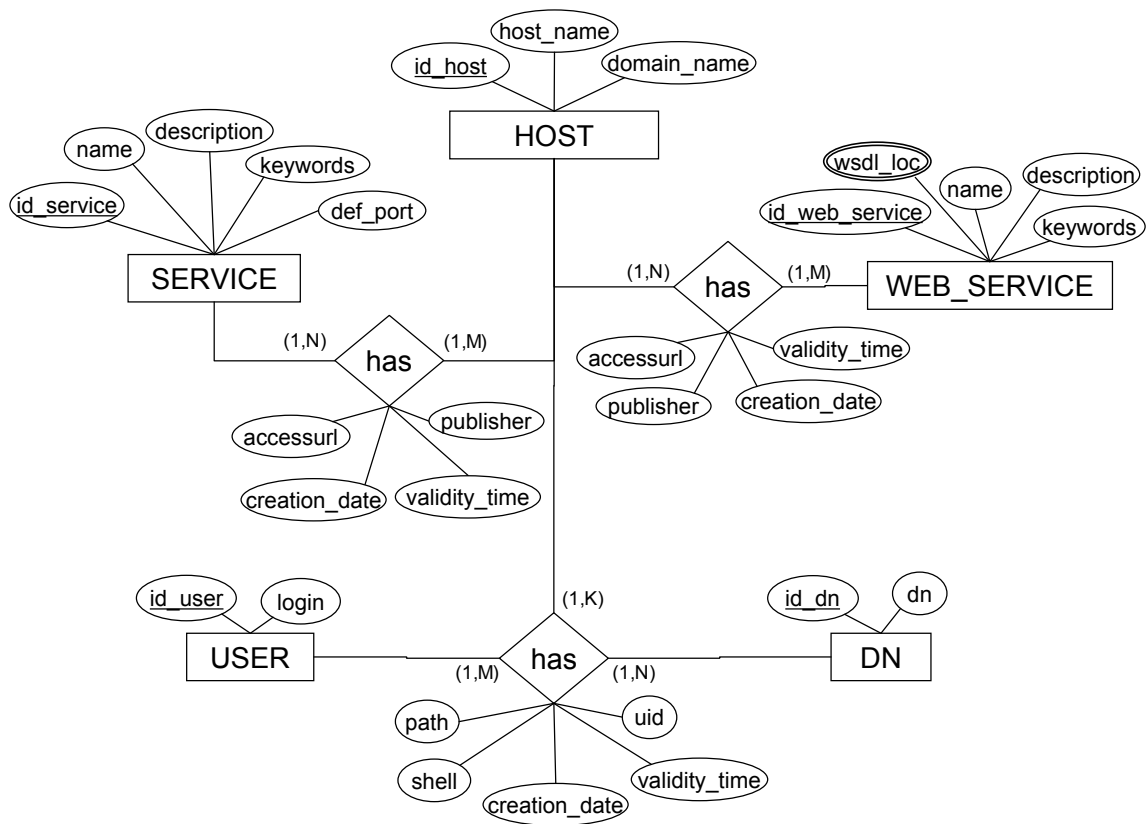


Figure 3: iGrid Entity Relationship Diagram (continued)

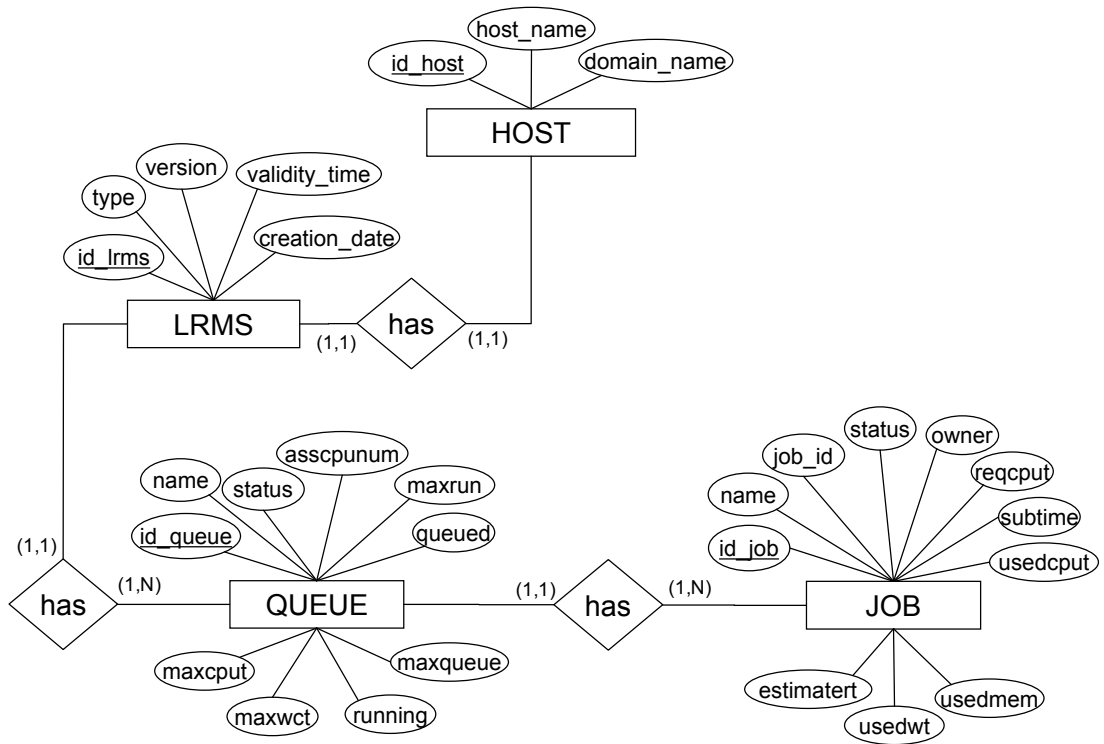


Figure 4: iGrid Entity Relationship Diagram (continued)

We now present the iGrid data dictionary.

Table name	Definition	Obligation	Maximum size	Data type
TB_HOST	Contains the information about host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_host	Host machine table id (primary key)	Mandatory	4	Serial
host_name	FQDN of the host	Mandatory	256	Character
domain_name	Domain name of the host	Mandatory	256	Character
id_hos_sys	Primary key of the TB_SYSTEM table (foreign key)	Optional	4	Integer
id_hos_fir	Primary key of the TB_FIREWALL table (foreign key)	Optional	4	Integer
TB_NET	Contains the information about the network	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_net_inf	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
name	Interface name (primary key)	Mandatory	64	Character
address	Network interface address	Mandatory	15	Character
mask	Network interface mask	Optional	15	Character
creation_date	Creation date (timestamp UTC) of net information	Mandatory	8	bigint
validity_time	Validity time (sec) of net information	Mandatory	4	Integer
TB_CPU_STATIC	Contains static information about cpu	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_cpu_s	Cpu static table id (primary key)	Mandatory	4	Serial
number	Processor number	Mandatory	4	Integer
vendor	Processor vendor	Optional	128	Character
version	Processor version	Optional	32	Character

model	Processor model	Optional	64	Character
cpu_mhz	Processor clock speed (MHz)	Mandatory	4	Real
cache_size	Cache size (KB)	Optional	4	Integer
features	Processor feature and option flags	Optional	256	Character
creation_date	Creation date (timestamp UTC) of net information	Mandatory	8	bigint
validity_time	Validity time (sec) of cpu static information	Mandatory	4	Integer
id_cpu_s_hos	HOST machine table id (foreign key)	Mandatory	4	Integer
TB_SYSTEM	Contains the information about the operating system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_sys	System table id (primary key)	Mandatory	4	Serial
name	Operating system name	Mandatory	50	Character
release	Operating system release	Mandatory	50	Character
version	Operating system version	Optional	50	Character
machine	Operating system platform type	Optional	30	Character
creation_date	Creation date (timestamp UTC) of system information	Mandatory	8	bigint
validity_time	Validity time (sec) of system information	Mandatory	4	Integer
TB_DEVICE_STATIC	Contains the information about the static file system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_dev_s	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
name	Device static name (primary key)	Mandatory	256	Character
mount_point	File system mount point	Mandatory	256	Character
fs_type	Type of system	Mandatory	32	Character

size	Total space of the file system (KB)	Mandatory	4	Integer
creation_date	Creation date (timestamp UTC) of device static information	Mandatory	8	bigint
validity_time	Validity time (sec) of device static information	Mandatory	4	Integer
TB_DEVICE_DYNAMIC	Contains the information about the dynamic file system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_dev_d	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
name	Device dynamic name (primary key)	Mandatory	256	Character
free_space	File system free space (KB)	Mandatory	4	Integer
creation_date	Creation date (timestamp UTC) of device dynamic information	Mandatory	8	bigint
validity_time	Validity time (sec) of device dynamic information	Mandatory	4	Integer
TB_MEMORY_STATIC	Contains the information about the static memory	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_mem_s	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
total_ram	Total RAM (MB)	Mandatory	4	Integer
total_swap	Total Swap (MB)	Optional	4	Integer
creation_date	Creation date (timestamp UTC) of memory static information	Mandatory	8	bigint
validity_time	Validity time (sec) of memory static information	Mandatory	4	Integer

TB_MEMORY_DYNAMIC	Contains the information about the dynamic memory	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_mem_d	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
free_ram	Free RAM (MB)	Mandatory	4	Integer
free_swap	Free Swap (MB)	Optional	4	Integer
creation_date	Creation date (timestamp UTC) of memory dynamic information	Mandatory	8	bigint
validity_time	Validity time (sec) of memory dynamic information	Mandatory	4	Integer
TB_CPU_DYNAMIC	Contains dynamic information about cpu	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_cpu_d	Host machine table id (primary key and foreign key)	Mandatory	4	Serial
number	Cpu number	Mandatory	4	Integer
load	Cpu load (%)	Mandatory	4	Real
user_time	User time (%)	Optional	4	Real
nice_time	Nice time (%)	Optional	4	Real
system_time	System time (%)	Optional	4	Real
idle_time	Idle time (%)	Optional	4	Real
creation_date	Creation date (timestamp UTC) of cpu dynamic information	Mandatory	8	bigint
validity_time	Validity time (sec) of cpu dynamic information	Mandatory	4	Integer
TB_FIREWALL	Contains information about the firewall hostname	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_firewall	Firewall table id (primary key)	Mandatory	4	Serial
hostname	Firewall hostname	Mandatory	256	Character
admin_dn	Administrator Distinguished Name	Optional	256	Character

publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	Creation date (timestamp UTC) of firewall information	Mandatory	8	bigint
validity_time	Validity time (sec) of firewall information	Mandatory	4	Integer
TB_FIR_PORTS	Contains information about ports range of the firewall hostname	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_fir_por	Firewall table id (primary key and foreign key)	Mandatory	4	Integer
ports	Open ports range (primary key) in the following format: <dddddd>:<dddddd>	Mandatory	13	Character
protocol	Protocol UDP/TCP	Mandatory	16	Character
creation_date	Creation date (timestamp UTC) of firewall open ports range	Mandatory	8	bigint
validity_time	Validity time (sec) of firewall open ports range	Mandatory	4	Integer
TB_VO	Contains information about the Virtual Organization	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_vo	Virtual Organization table id (primary key)	Mandatory	4	Serial
name	VO name	Mandatory	256	Character
admin_name	VO Administrator name	Optional	256	Character
resource_type	VO resource type	Optional	32	Character
job_manager	VO job manager	Optional	50	Character
queue	VO queue	Optional	50	Character
help_desk_url	URL pointing to a web page describing the VO	Optional	256	Character

RL_HOS_VO	Contains information about the relation between Virtual Organization and Host table	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_vo	VO table id (primary key and foreign key)	Mandatory	4	Integer
publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	Creation date (timestamp UTC) of VO information relative to a host machine	Mandatory	8	bigint
validity_time	Validity time (sec) of VO information relative to a host machine	Mandatory	4	Integer
TB_VO_HDPN	Contains information about the help desk phone number of a Virtual Organization	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_vo_hdp	VO table id (primary key and foreign key)	Mandatory	4	Integer
help_desk_ph_num	Phone number related to the VO (primary key)	Mandatory	32	Character
TB_VO_FS	Contains information about the file system of a Virtual Organization	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_vo_fs	VO table id (primary key and foreign key)	Mandatory	4	Integer
path	Path of the VO file system (primary key)	Mandatory	256	Character

type	Type of the VO file system	Optional	256	Character
TB_LRMS	Contains information about the local resource management system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_lrms	Local Resource Management System table id (primary key)	Mandatory	4	Serial
type	type of the LRMS	Mandatory	32	Character
version	version of the LRMS	Optional	16	Character
creation_date	Creation date (timestamp UTC) of LRMS information	Mandatory	8	bigint
validity_time	Validity time (sec) of LRMS information	Mandatory	4	Integer
id_hos_lrm	Host machine table id (foreign key)	Mandatory	4	Integer
TB_QUEUE	Contains information about the queues of a local resource management system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_queue	Queue table id (primary key)	Mandatory	4	Integer
name	Queue name	Mandatory	256	Character
asscpunum	Number of the cpus assigned to the queue	Optional	4	Integer
status	Queue status	Optional	32	Character
maxqueue	Max number of jobs allowed to reside in the queue	Optional	4	Integer
maxrun	Max number of jobs allowed to run from this queue	Optional	4	Integer
queued	Number of jobs waiting in the queue	Optional	4	Integer
running	Number of running jobs in the cluster belonging to this queue	Optional	4	Integer

maxwct	Max wall clock time allowed for jobs submitted to the queue in mins	Optional	4	Integer
maxcput	Max CPU time allowed for jobs submitted to the queue in mins	Optional	4	Integer
id_lrm_que	LRMS table id (foreign key)	Optional	4	Integer
TB_JOB	Contains information about the jobs of a local resource management system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_job	Job table id (primary key)	Mandatory	4	Integer
name	Name of the job	Mandatory	256	Character
job_id	Job identifier assigned by lrms	Mandatory	64	Character
owner	SubjectName of the job owner	Optional	256	Character
status	Status of the job	Optional	32	Character
reqcput	Cputime request of the job in minutes	Optional	4	Integer
subtime	Submission time of the job	Optional	8	Character
usedcput	Consumed cputime of the job in minutes	Optional	4	Integer
usedmem	Memory usage of the job	Optional	4	Integer
usedwt	Consumed wall-time of the job in minutes	Optional	4	Integer
estimatert	Estimated time between job submission till when job starts its execution	Optional	4	Integer
id_que_job	Queue table id (foreign key)	Optional	4	Integer
TB_SERVICE	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object

id_service	Service table id (primary key)	Mandatory	4	Serial
name	Service name	Mandatory	256	Character
description	Service description	Optional	1024	VarChar
keywords	Service keywords	Optional	256	Character
def_port	Service default port	Mandatory	4	Integer
RL_HOS_SERVICE	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_service	Service table id (primary key and foreign key)	Mandatory	4	Integer
accessurl	Service access URL (primary key)	Mandatory	256	Character
publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	Creation date (timestamp UTC) of service information	Mandatory	8	bigint
validity_time	Validity time (sec) of service information	Mandatory	4	Integer
TB_WEB_SERVICE	Contains information about the web service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_web_service	Web Service table id (primary key)	Mandatory	4	Integer
name	Web Service name	Mandatory	256	Character
description	Service description	Optional	1024	VarChar
keywords	Service keywords	Optional	256	Character
TB_WS_WSDL_LOC	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_ws_wsdl	Web Service table id (primary key and foreign key)	Mandatory	4	Integer

wSDL_locationurl	WSDL location file URL (primary key)	Mandatory	256	Character
RL_HOS_WS	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_ws	Web Service table id (primary key and foreign key)	Mandatory	4	Integer
accessurl	Web Service access URL (primary key)	Mandatory	256	Character
publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	Creation date (timestamp UTC) of web service information	Mandatory	8	bigint
validity_time	Validity time (sec) of web service information	Mandatory	4	Integer
TB_USER	Contains information about the user	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_user	User table id (primary key)	Mandatory	4	Serial
login	User login	Mandatory	32	Character
TB_DN	Contains information about the Distinguished Name mapped on the user	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_dn	DN table id (primary key)	Mandatory	4	Serial
dn	DN mapped on the user	Mandatory	256	Character

RL_HOS_DN_USE	Contains information about the relation among Distinguished Name, user and host tables	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_use	User table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_dn	DN table id (primary key and foreign key)	Mandatory	4	Integer
uid	User ID	Optional	4	Integer
shell	User shell	Optional	256	Character
path	User home dir	Optional	256	Character
creation_date	Creation date (timestamp UTC) of information related to host, user and Distinguished Name tables	Mandatory	8	bigint
validity_time	Validity time (sec) of information related to host, user and Distinguished Name tables	Mandatory	4	Integer
TB_CA	Contains information about the Certification Authority	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_ca	CA table id (primary key)	Mandatory	4	Integer
version	Version of the certificate	Optional	16	Character
serial_num	Serial Number of the certificate	Optional	16	Character
sig_alg	Signature algorithm of the certificate	Optional	64	Character
issuer	Issuer of the certificate	Mandatory	256	Character

valid_from	Beginning date of the certificate validity (timestamp UTC)	Mandatory	8	bigint
valid_to	End date of the certificate validity (timestamp UTC)	Mandatory	8	bigint
pk_alg	Public key algorithm of the certificate	Optional	64	Character
rsa_pk	RSA public key of the certification authority	Optional	64	VarChar
subj	Subject name of the CA certificate	Mandatory	256	Character
crL_url	Crl URL	Optional	256	Character
RL_HOS_CA	Contains information about the relation between Certification Authority and host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host table id (primary key)	Mandatory	4	Integer
id_rl_ca	CA table id (primary key)	Mandatory	4	Integer
creation_date	Creation date (timestamp UTC) of information related to host and CA tables	Mandatory	8	bigint
validity_time	Validity time (sec) of information related to host and CA tables	Mandatory	4	Integer

4 GridLab iGrid 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, update and lookup information.

The namespace used by the web service is urn:igrid. 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 methods:

4.1 Search

-int gridlab_search(xsd_string filter, struct dimeData **dime_data)

This method can be used to query the local iStore server for specified information.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

filter: search filter criteria.

OUT parameters:

dime_data: dimeData structure containing all the information stored in the local iStore db.

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 validity_time, xsd_int *result)

This method allows the user to register a new service or a an instance of an existing service.

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
validity_time: validity time of the service instance information

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.

-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_update_service (xsd_string name, xsd_string keywords, xsd_string description, xsd_string url, xsd_int dport, xsd_int * result)

This method allows updating name, description, keywords and default port of a service or the access url of an instance of a service.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the service to be updated
keywords: the key words of the service to be updated
description: the description of the service to be updated
dport: the default port where the service to be updated is listening on
url: the access url of the service instance to be updated

OUT parameters:

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

-int gridlab_lookup_service (xsd_string groupby, xsd_string name, xsd_int dport, xsd_int sport, xsd_string prot, struct dimeData **dime_data)

This method returns all the information about the services stored on the local iStore server grouped by hostname or service name. If any input parameters have been specified, the method returns the information about all the services and the related instances matching parameters values. 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 and grouping them by hostname or service name.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

groupby: the parameter used to group searched information.
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:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.3 Web Services

-int gridlab_register_webservice (xsd_string name, xsd_string wsdllocation, xsd_string keywords, xsd_string description, xsd_string url, xsd_int validity_time, 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.

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.

validity_time: validity time of the web service instance information.

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.

-int gridlab_unregister_webservice (xsd_string name, xsd_string url, 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__update_webservice (xsd__string name, xsd__string keywords, xsd__string description, xsd__string url, xsd__int * result)

This method allows updating name, description and keywords of a web service or the access url of an instance of a web service.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

- name: the logical name of the service to be updated.
- keywords: the key words of the service to be updated.
- description: the description of the service to be updated.
- url: the access url of the web service instance to be updated.

OUT parameters:

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

-int gridlab__lookup_webservice (xsd__string groupby, xsd__string name, struct dimeData **dime_data)

This method returns all the information about the web services stored on the local iStore server grouped by hostname or web service name. If the logical name has been specified, the method returns the information about specified web services and related instances grouping them by hostname or web service name.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

- groupby: the parameter used to group searched information.
- name: the logical name of the web service to be searched.

OUT parameters:

- dime_data: dimeData structure containing searched information stored in the local iStore db.

4.4 Firewall

-int gridlab__register_firewall(xsd__string hostname, xsd__string ports, xsd__string admin, xsd__string host, xsd__int validity_time, 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: name of the firewall.
- ports: space-separated list of open ports (or range of ports separated by - symbol), the protocol used to contact them and related validity time with format port-protocol:validity_time.

admindn: Distinguished Name of the firewall administrator.
hostname: host name of the machine where the firewall is installed.
validity_time: validity time of the firewall information.

OUT parameters:

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

`-int gridlab_unregister_firewall(xsd_string hostname, xsd_string host, xsd_int allinst, xsd_int * result)`

This method removes information about a registered firewall.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

hostname: name of the firewall.
host: host name of the machine where the firewall is installed; the corresponding information will be removed.
allinst: a flag set to 1 to remove specified firewall information and the relations between it and all of the related hosts, set to 0 to remove a single relation.

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_update_firewall(xsd_string hostname, xsd_string admindn, xsd_string protocol, xsd_int * result)`

This method allows updating name, protocol and admin DN of a firewall.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

hostname: the name of the firewall to be updated.
admindn: the Distinguished Name of the administrator of the firewall to be updated.
protocol: the protocol (TCP/UDP) used to contact the ports of the firewall to be updated.

OUT parameters:

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

`-int gridlab_lookup_firewall(xsd_string groupby, xsd_string name, struct dimeData **dime_data)`

This method returns all the information about the registered firewalls grouped by hostname or firewall name. If a name has been specified, the method returns the information about specified firewall grouping them by hostname or firewall name.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

groupby: the parameter used to group searched information.
name: the hostname of the firewall to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.5 Virtual Organization

```
-int gridlab__register_vo(xsd_string name, xsd_string helpDeskPN, xsd_string restype, xsd_string jobm, xsd_string queue, xsd_string fs_path, xsd_string helpDeskURL, xsd_string adminname, xsd_string host, xsd_int validity_time, 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 Virtual Organization.
helpDeskPN: the Help-Desk Phone number of the Virtual Organization.
restype: resource type of the Virtual Organization.
jobm: jobmanager used by the Virtual Organization.
queue: job queue of the Virtual Organization.
fs_path: space separated available file systems (type:pathname) for Virtual Organization.
helpDeskURL: the URL of the help-Desk.
adminname: Distinguished Name of the administrator.
host: the name of the host to be registered to the Virtual Organization.
validity_time: validity time of the firewall information.

OUT parameters:

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

```
-int gridlab__unregister_vo(xsd_string name, xsd_string hostname, xsd_int allinst, 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.
hostname: host name of the machine belonging to the Virtual Organization.
allinst: a flag set to 1 to remove specified Virtual Organization information and the relations between it and all of the registered hosts, set to 0 to remove a single relation.

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_update_vo(xsd_string name, xsd_string restype, xsd_string jobm, xsd_string queue, xsd_string helpDeskURL, xsd_string adminname, xsd_int * result)
```

This method allows updating name, resource type, job manager, queue, help desk url and admin DN of a VO.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: name of the Virtual Organization to be updated.
restype: resource type of the Virtual Organization to be updated.
jobm: jobmanager used by the Virtual Organization to be updated.
queue: job queue of the Virtual Organization to be updated.
helpDeskURL: the URL of the help-Desk of the Virtual Organization to be updated.
adminname: Distinguished Name of the administrator of the Virtual Organization to be updated.

OUT parameters:

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

```
-int gridlab_lookup_vo(xsd_string groupby, xsd_string name, struct dimeData **dime_data)
```

This method returns all the information about the registered virtual organizations grouped by hostname or virtual organization name. If a name has been specified, the method returns the information about specified virtual organization grouping them by hostname or virtual organization name.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

groupby: the parameter used to group searched information.
name: the name of the Virtual Organization to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

```
-int gridlab_register_wsdlloc (xsd_string name, xsd_string wsdlloc, xsd_int * result)
```

This method allows the user to register one or more new wsdl locations url associated to a web service. The wsdlloc parameter is a list, space-separated, of web service WSDL file location URL variables; each variable must not contain spaces.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the new web service to be registered.
wsdlloc: a list, space-separated, of web service WSDL file location URL variables.

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

4.6 User

-int gridlab_lookup_user(xsd_string groupby, xsd_string name, struct dimeData **dime_data)
This method returns all the information about registered users grouped by hostname or distinguished name. If a distinguished name has been specified, the method returns the information about accounts mapped on it grouped by hostname or distinguished name.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

groupby: the parameter used to group searched information.
name: the DN mapped on the user to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.7 Certification Authority

-int gridlab_lookup_ca(xsd_string groupby, xsd_string name, struct dimeData **dime_data)
This method returns the information for an accepted Certification Authority grouped by hostname or subject. If a subject has been specified, the method returns the information about specified Certification Authority grouped by hostname or Certification Authority.
Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

groupby: the parameter used to group searched information.
name: the subject of the Certification Authority to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.8 Local Resource Management System

-int gridlab_lookup_lrms(xsd_string name, struct dimeData **dime_data)
This method returns the information about the registered local resource management systems (lrms) grouped by hostname. If a type of lrms has been specified, the method returns the information about specified lrms grouped by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the type of lrms to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.9 CPU

-int gridlab_lookup_cpu(xsd_int mhz, xsd_int cache, xsd_int number, xsd_int load, struct dimeData **dime_data)

This method returns static and dynamic information about CPUs stored on the local iStore server grouped by hostname. If any input parameters have been specified, the method returns the information about all of the CPUs matching parameters values. It is possible to search the information about one or more CPUs specifying the CPU mhz, cache size, number and load parameters or a combination of these and grouping retrieved information by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mhz: mhz of the CPU to be searched.

cache: cache size of the CPU to be searched.

number: cardinal number of the CPU to be searched.

load: load of the CPU to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.10 Device

-int gridlab_lookup_device (xsd_string space, struct dimeData **dime_data)

This method returns static and dynamic information about devices stored on the local iStore server grouped by hostname. If space input parameter has been specified, the method returns the information matching its value.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

space: device space to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.11 Memory

-int gridlab_lookup_memory(xsd_int tot_ram, xsd_int tot_swap, xsd_int free_ram, xsd_int free_swap, struct dimeData **dime_data)

This method returns static and dynamic information about memory stored on the local iStore server grouped by hostname. If any input parameters have been specified, the method returns the information matching parameters values. It is possible to search the information about one or more memories specifying the total ram, total swap, free ram and free swap or a combination of these and grouping retrieved information by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

tot_ram: total ram to be searched.
tot_swap: tot swap to be searched.
free_ram: free ram to be searched.
free_swap: free swap to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.12 Network

-int gridlab_lookup_network(xsd_string address, struct dimeData **dime_data)

This method returns information about network stored on the local iStore server grouped by hostname. If address parameter has been specified, the method returns the information matching its value.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

address: IP address of the host to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

4.13 Operative System

-int gridlab_lookup_system(xsd_string groupby, xsd_string name, xsd_string release, struct dimeData **dime_data)

This method returns operative system information stored on the local iStore server. If any input parameters have been specified, the method returns the information matching parameters values. It is possible to search the information about operative system specifying its name and/or the release and grouping retrieved information by hostname or operative system.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

groupby: the parameter used to group searched information.

name: name of the operative system to be searched.

release: release of the operative system to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore db.

Appendix

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

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="igrid"
  targetNamespace="urn:igrid"
  xmlns:tns="urn:igrid"
  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:igrid"
  xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

<types>

<schema targetNamespace="urn:igrid"
  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:igrid"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <simpleType name="dimeData">
    <restriction base="xsd:base64Binary">
    </restriction>
  </simpleType>
  <element name="register-webservice">
    <complexType>
      <sequence>
        <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
        <element name="wsdllocation" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
        <element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
        <element name="url" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
        <element name="keywords" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
        <element name="validity-time" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      </sequence>
    </complexType>
  </element>
  <element name="register-webserviceResponse">
    <complexType>
```

```
<sequence>
  <element name="result" type="xsd:int"/>
</sequence>
</complexType>
</element>
<element name="unregister-webservice">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="url" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="allinst" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="unregister-webserviceResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="istore">
  <complexType>
    <sequence>
      <element name="data" type="gridlab:dimeData" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="istoreResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="search">
  <complexType>
    <sequence>
      <element name="filter" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="searchResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
```

```
<element name="update-webservice">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="keywords" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="url" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="update-webserviceResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="register-service">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="hostname" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="port" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="protocol" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="dport" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="keywords" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="validity-time" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="register-serviceResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="unregister-service">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="hostname" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="port" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="allinst" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="unregister-serviceResponse">
```

```
<complexType>
  <sequence>
    <element name="result" type="xsd:int"/>
  </sequence>
</complexType>
</element>
<element name="update-service">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="keywords" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="url" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="dport" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="update-serviceResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-webservice">
  <complexType>
    <sequence>
      <element name="groupby" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-webserviceResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-service">
  <complexType>
    <sequence>
      <element name="groupby" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="dport" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="sport" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="prot" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
```

```
</element>
<element name="lookup-serviceResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-firewall">
  <complexType>
    <sequence>
      <element name="groupby" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-firewallResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-system">
  <complexType>
    <sequence>
      <element name="groupby" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="system" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-systemResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-user">
  <complexType>
    <sequence>
      <element name="groupby" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-userResponse">
  <complexType>
```

```
<sequence>
  <element name="dime-data" type="gridlab:dimeData"/>
</sequence>
</complexType>
</element>
<element name="lookup-vo">
  <complexType>
    <sequence>
      <element name="groupby" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-voResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-ca">
  <complexType>
    <sequence>
      <element name="groupby" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-caResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-lrms">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-lrmsResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
```

```
<element name="lookup-cpu">
  <complexType>
    <sequence>
      <element name="mhz" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="cache" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="number" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="load" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-cpuResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-memory">
  <complexType>
    <sequence>
      <element name="total-ram" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="total-swap" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="free-ram" type="xsd:int" minOccurs="1" maxOccurs="1"/>
      <element name="free-swap" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-memoryResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-network">
  <complexType>
    <sequence>
      <element name="address" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-networkResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="lookup-device">
```

```
<complexType>
  <sequence>
    <element name="device" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
  </sequence>
</complexType>
</element>
<element name="lookup-deviceResponse">
  <complexType>
    <sequence>
      <element name="dime-data" type="gridlab:dimeData"/>
    </sequence>
  </complexType>
</element>
<element name="register-vo">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="helpDeskPN" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="restype" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="jobm" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="queue" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="fs-path" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="helpDeskURL" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="adminname" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="host" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="validity-time" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="register-voResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="unregister-vo">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="host" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="allinst" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="unregister-voResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
```

```
</sequence>
</complexType>
</element>
<element name="update-vo">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="restype" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="jobm" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="queue" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="helpDeskURL" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="adminname" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="update-voResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="register-firewall">
  <complexType>
    <sequence>
      <element name="hostname" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="ports" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="admindn" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="host" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="validity-time" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<element name="register-firewallResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="unregister-firewall">
  <complexType>
    <sequence>
      <element name="hostname" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="host" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="allinst" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

```
<element name="unregister-firewallResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="update-firewall">
  <complexType>
    <sequence>
      <element name="hostname" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="adminIn" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="protocol" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="update-firewallResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
<element name="register-wsdlloc">
  <complexType>
    <sequence>
      <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
      <element name="wsdlloc" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    </sequence>
  </complexType>
</element>
<element name="register-wsdllocResponse">
  <complexType>
    <sequence>
      <element name="result" type="xsd:int"/>
    </sequence>
  </complexType>
</element>
</schema>

</types>

<message name="register-webserviceRequest">
  <part name="parameters" element="gridlab:register-webservice"/>
</message>

<message name="register-webserviceResponse">
  <part name="parameters" element="gridlab:register-webserviceResponse"/>
</message>
```

```
<message name="unregister-webserviceRequest">
  <part name="parameters" element="gridlab:unregister-webservice"/>
</message>

<message name="unregister-webserviceResponse">
  <part name="parameters" element="gridlab:unregister-webserviceResponse"/>
</message>

<message name="istoreRequest">
  <part name="parameters" element="gridlab:istore"/>
</message>

<message name="istoreResponse">
  <part name="parameters" element="gridlab:istoreResponse"/>
</message>

<message name="searchRequest">
  <part name="parameters" element="gridlab:search"/>
</message>

<message name="searchResponse">
  <part name="parameters" element="gridlab:searchResponse"/>
</message>

<message name="update-webserviceRequest">
  <part name="parameters" element="gridlab:update-webservice"/>
</message>

<message name="update-webserviceResponse">
  <part name="parameters" element="gridlab:update-webserviceResponse"/>
</message>

<message name="register-serviceRequest">
  <part name="parameters" element="gridlab:register-service"/>
</message>

<message name="register-serviceResponse">
  <part name="parameters" element="gridlab:register-serviceResponse"/>
</message>

<message name="unregister-serviceRequest">
  <part name="parameters" element="gridlab:unregister-service"/>
</message>

<message name="unregister-serviceResponse">
  <part name="parameters" element="gridlab:unregister-serviceResponse"/>
</message>
```

```
<message name="update-serviceRequest">
  <part name="parameters" element="gridlab:update-service"/>
</message>

<message name="update-serviceResponse">
  <part name="parameters" element="gridlab:update-serviceResponse"/>
</message>

<message name="lookup-webserviceRequest">
  <part name="parameters" element="gridlab:lookup-webservice"/>
</message>

<message name="lookup-webserviceResponse">
  <part name="parameters" element="gridlab:lookup-webserviceResponse"/>
</message>

<message name="lookup-serviceRequest">
  <part name="parameters" element="gridlab:lookup-service"/>
</message>

<message name="lookup-serviceResponse">
  <part name="parameters" element="gridlab:lookup-serviceResponse"/>
</message>

<message name="lookup-firewallRequest">
  <part name="parameters" element="gridlab:lookup-firewall"/>
</message>

<message name="lookup-firewallResponse">
  <part name="parameters" element="gridlab:lookup-firewallResponse"/>
</message>

<message name="lookup-systemRequest">
  <part name="parameters" element="gridlab:lookup-system"/>
</message>

<message name="lookup-systemResponse">
  <part name="parameters" element="gridlab:lookup-systemResponse"/>
</message>

<message name="lookup-userRequest">
  <part name="parameters" element="gridlab:lookup-user"/>
</message>

<message name="lookup-userResponse">
  <part name="parameters" element="gridlab:lookup-userResponse"/>
</message>

<message name="lookup-voRequest">
```

```
<part name="parameters" element="gridlab:lookup-vo"/>
</message>

<message name="lookup-voResponse">
  <part name="parameters" element="gridlab:lookup-voResponse"/>
</message>

<message name="lookup-caRequest">
  <part name="parameters" element="gridlab:lookup-ca"/>
</message>

<message name="lookup-caResponse">
  <part name="parameters" element="gridlab:lookup-caResponse"/>
</message>

<message name="lookup-lrmsRequest">
  <part name="parameters" element="gridlab:lookup-lrms"/>
</message>

<message name="lookup-lrmsResponse">
  <part name="parameters" element="gridlab:lookup-lrmsResponse"/>
</message>

<message name="lookup-cpuRequest">
  <part name="parameters" element="gridlab:lookup-cpu"/>
</message>

<message name="lookup-cpuResponse">
  <part name="parameters" element="gridlab:lookup-cpuResponse"/>
</message>

<message name="lookup-memoryRequest">
  <part name="parameters" element="gridlab:lookup-memory"/>
</message>

<message name="lookup-memoryResponse">
  <part name="parameters" element="gridlab:lookup-memoryResponse"/>
</message>

<message name="lookup-networkRequest">
  <part name="parameters" element="gridlab:lookup-network"/>
</message>

<message name="lookup-networkResponse">
  <part name="parameters" element="gridlab:lookup-networkResponse"/>
</message>

<message name="lookup-deviceRequest">
  <part name="parameters" element="gridlab:lookup-device"/>
</message>
```

```
</message>

<message name="lookup-deviceResponse">
  <part name="parameters" element="gridlab:lookup-deviceResponse"/>
</message>

<message name="register-voRequest">
  <part name="parameters" element="gridlab:register-vo"/>
</message>

<message name="register-voResponse">
  <part name="parameters" element="gridlab:register-voResponse"/>
</message>

<message name="unregister-voRequest">
  <part name="parameters" element="gridlab:unregister-vo"/>
</message>

<message name="unregister-voResponse">
  <part name="parameters" element="gridlab:unregister-voResponse"/>
</message>

<message name="update-voRequest">
  <part name="parameters" element="gridlab:update-vo"/>
</message>

<message name="update-voResponse">
  <part name="parameters" element="gridlab:update-voResponse"/>
</message>

<message name="register-firewallRequest">
  <part name="parameters" element="gridlab:register-firewall"/>
</message>

<message name="register-firewallResponse">
  <part name="parameters" element="gridlab:register-firewallResponse"/>
</message>

<message name="unregister-firewallRequest">
  <part name="parameters" element="gridlab:unregister-firewall"/>
</message>

<message name="unregister-firewallResponse">
  <part name="parameters" element="gridlab:unregister-firewallResponse"/>
</message>

<message name="update-firewallRequest">
  <part name="parameters" element="gridlab:update-firewall"/>
</message>
```

```
<message name="update-firewallResponse">
  <part name="parameters" element="gridlab:update-firewallResponse"/>
</message>

<message name="register-wsdllocRequest">
  <part name="parameters" element="gridlab:register-wsdlloc"/>
</message>

<message name="register-wsdllocResponse">
  <part name="parameters" element="gridlab:register-wsdllocResponse"/>
</message>

<portType name="igridPortType">
  <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</documentation>
    <input message="tns:unregister-webserviceRequest"/>
    <output message="tns:unregister-webserviceResponse"/>
  </operation>
  <operation name="istore">
    <documentation>Service definition of function gridlab__istore</documentation>
    <input message="tns:istoreRequest"/>
    <output message="tns:istoreResponse"/>
  </operation>
  <operation name="search">
    <documentation>Service definition of function gridlab__search</documentation>
    <input message="tns:searchRequest"/>
    <output message="tns:searchResponse"/>
  </operation>
  <operation name="update-webservice">
    <documentation>Service definition of function gridlab__update_webservice</documentation>
    <input message="tns:update-webserviceRequest"/>
    <output message="tns:update-webserviceResponse"/>
  </operation>
  <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>
</portType>
```

```
<operation name="update-service">
  <documentation>Service definition of function gridlab__update_service</documentation>
  <input message="tns:update-serviceRequest"/>
  <output message="tns:update-serviceResponse"/>
</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="lookup-service">
  <documentation>Service definition of function gridlab__lookup_service</documentation>
  <input message="tns:lookup-serviceRequest"/>
  <output message="tns:lookup-serviceResponse"/>
</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="lookup-system">
  <documentation>Service definition of function gridlab__lookup_system</documentation>
  <input message="tns:lookup-systemRequest"/>
  <output message="tns:lookup-systemResponse"/>
</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-vo">
  <documentation>Service definition of function gridlab__lookup_vo</documentation>
  <input message="tns:lookup-voRequest"/>
  <output message="tns:lookup-voResponse"/>
</operation>
<operation name="lookup-ca">
  <documentation>Service definition of function gridlab__lookup_ca</documentation>
  <input message="tns:lookup-caRequest"/>
  <output message="tns:lookup-caResponse"/>
</operation>
<operation name="lookup-lrms">
  <documentation>Service definition of function gridlab__lookup_lrms</documentation>
  <input message="tns:lookup-lrmsRequest"/>
  <output message="tns:lookup-lrmsResponse"/>
</operation>
<operation name="lookup-cpu">
  <documentation>Service definition of function gridlab__lookup_cpu</documentation>
  <input message="tns:lookup-cpuRequest"/>
  <output message="tns:lookup-cpuResponse"/>
</operation>
```

```
</operation>
<operation name="lookup-memory">
  <documentation>Service definition of function gridlab__lookup_memory</documentation>
  <input message="tns:lookup-memoryRequest"/>
  <output message="tns:lookup-memoryResponse"/>
</operation>
<operation name="lookup-network">
  <documentation>Service definition of function gridlab__lookup_network</documentation>
  <input message="tns:lookup-networkRequest"/>
  <output message="tns:lookup-networkResponse"/>
</operation>
<operation name="lookup-device">
  <documentation>Service definition of function gridlab__lookup_device</documentation>
  <input message="tns:lookup-deviceRequest"/>
  <output message="tns:lookup-deviceResponse"/>
</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="update-vo">
  <documentation>Service definition of function gridlab__update_vo</documentation>
  <input message="tns:update-voRequest"/>
  <output message="tns:update-voResponse"/>
</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="unregister-firewall">
  <documentation>Service definition of function gridlab__unregister_firewall</documentation>
  <input message="tns:unregister-firewallRequest"/>
  <output message="tns:unregister-firewallResponse"/>
</operation>
<operation name="update-firewall">
  <documentation>Service definition of function gridlab__update_firewall</documentation>
  <input message="tns:update-firewallRequest"/>
  <output message="tns:update-firewallResponse"/>
</operation>
<operation name="register-wsdlloc">
  <documentation>Service definition of function gridlab__register_wsdlloc</documentation>
  <input message="tns:register-wsdllocRequest"/>
```

```
<output message="tns:register-wsdllocResponse"/>
</operation>
</portType>

<binding name="igrid" type="tns:igridPortType">
  <SOAP:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="register-webservice">
    <SOAP:operation style="document" soapAction=""/>
    <input>
      <SOAP:body use="literal"/>
    </input>
    <output>
      <SOAP:body use="literal"/>
    </output>
  </operation>
  <operation name="unregister-webservice">
    <SOAP:operation style="document" soapAction=""/>
    <input>
      <SOAP:body use="literal"/>
    </input>
    <output>
      <SOAP:body use="literal"/>
    </output>
  </operation>
  <operation name="istore">
    <SOAP:operation style="document" soapAction=""/>
    <input>
      <SOAP:body use="literal"/>
    </input>
    <output>
      <SOAP:body use="literal"/>
    </output>
  </operation>
  <operation name="search">
    <SOAP:operation style="document" soapAction=""/>
    <input>
      <SOAP:body use="literal"/>
    </input>
    <output>
      <SOAP:body use="literal"/>
    </output>
  </operation>
  <operation name="update-webservice">
    <SOAP:operation style="document" soapAction=""/>
    <input>
      <SOAP:body use="literal"/>
    </input>
    <output>
      <SOAP:body use="literal"/>
    </output>
  </operation>

```

```
</output>
</operation>
<operation name="register-service">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="unregister-service">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="update-service">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-webservice">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-service">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-firewall">
  <SOAP:operation style="document" soapAction=""/>
```

```
<input>
  <SOAP:body use="literal"/>
</input>
<output>
  <SOAP:body use="literal"/>
</output>
</operation>
<operation name="lookup-system">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-user">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-vo">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-ca">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-lrms">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
```

```
<SOAP:body use="literal"/>
</output>
</operation>
<operation name="lookup-cpu">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-memory">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-network">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="lookup-device">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="register-vo">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="unregister-vo">
```

```
<SOAP:operation style="document" soapAction=""/>
<input>
  <SOAP:body use="literal"/>
</input>
<output>
  <SOAP:body use="literal"/>
</output>
</operation>
<operation name="update-vo">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="register-firewall">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="unregister-firewall">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="update-firewall">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
  <output>
    <SOAP:body use="literal"/>
  </output>
</operation>
<operation name="register-wsdlloc">
  <SOAP:operation style="document" soapAction=""/>
  <input>
    <SOAP:body use="literal"/>
  </input>
```

```
<output>
  <SOAP:body use="literal"/>
</output>
</operation>
</binding>

<service name="igrid">
  <documentation>gSOAP 2.6.2 generated service definition</documentation>
  <port name="igrid" binding="tns:igrid">
    <SOAP:address location="httpg://gridsurfer.unile.it:21000"/>
  </port>
</service>

</definitions>
```