



# Grid Advanced Information System

## System Administrator & User's Guide

KMI-R1 Developer Team <[kmi@moredream.org](mailto:kmi@moredream.org)>

Document Version: 0.9    Date: June 13, 2005



Copyright 2002 - 2005 Korea Institute of Scientist and Technology Information.  
All rights reserved.

This document is licensed under the terms of the K\*Grid Public License.  
The details of K\*Grid Public License is found at  
<http://kmi.moredream.org/downloads/license.html>.

# Contents

1. Key Concept.....	4
1.1. Introduction .....	4
1.2. Components.....	4
1.2.1. Datacan Factory Service (DFS).....	4
1.2.2. VO Roster Service (VRS) .....	5
1.2.3. VO Crawler Factory Service (VCFS) .....	6
1.2.4. MoreDream Providers.....	7
1.3. Schema .....	8
1.4. User Interface.....	11
1.5. Features .....	11
1.6. GAIS in actions .....	12
2. Installation and Configuration.....	13
2.1. Requirements.....	13
2.2. Installing required software .....	13
2.2.1. Installing Globus Toolkit .....	13
2.2.2. Installing Gridsphere .....	13
2.2.3. Installing OpenPBS.....	13
2.3. Installing GAIS .....	13
2.3.1. Download and extract.....	13
2.3.2. Compile & Installation .....	14
2.3.3. How to set up GAIS services .....	15
2.3.4. Starting.....	24
3. Using GAIS.....	25
4. References .....	27

# 1. Key Concept

## 1.1. Introduction

Grid information system is a critical component for Grid computing, by which all types of Grid resources are virtually integrated and their information can be effectually managed and accessed. Furthermore, the efficiency of Grid computing is dependent on the functionalities supported by Grid information system. But the existing information system such as MDS (Monitoring and Discovery System) of GT (Globus Toolkit), which is currently received wide publicity in Grid community, is not appropriate for a production-mode service or a long-lived service because it supports only basic functions. That is why we developed a new Grid information system named Grid Advanced Information System (GAIS), which is a versatile information system that provides information about the available resources on Grids and their status.

GAIS is the information services component of the MoreDream and is composed of a collection of OGSI-compliant services which add and extend the functionalities of GT3 MDS3. It is differentiated from the dynamic management and the flat network of directory servers mentioned below.

## 1.2. Components

GAIS is composed of three grid services and two information providers. Each service is related to manage and search information in a Grid, whereas two providers play information sources for the GAIS.

### 1.2.1. Datacan Factory Service (DFS)

*Datacan* (a compound word of “data” and “can”) is an enhanced version of GT3.x index service. Like an index service, it aggregates *Service Data* from Resource Services such as RIPS (Resource Information Provider Service) or other Grid service instances by means of the *Aggregator* mechanism. It also registers Grid service instances using the *ServiceGroup* mechanism. For these aggregation and registration, it uses the *RegistryPublishPrivder* mechanism. Additionally, it provides the following functionalities:

- 1) It removes stale data to assure data accuracy using the *Data Sweep* mechanism. When a datacan is created, the mechanism in the datacan calls a *ServiceDataSweeper*, which periodically checks the available time of registered service data and deletes old service data.

- 2) It has two types. The one is a public datacan (*pubcan*) to announce its information to a VO, and the other is a private datacan (*prican*) to share it only in a domain. A DFS administrator can configure a pubcan suitable for a VO according to her policy and she can also set up a prican to serve some users' purpose in her domain. It operates together with CAS (Community Authority Service) to control access to a pubcan or prican (not implemented).

DFS manages the lifecycle of a datacan using the *Factory* mechanism and maintains the snapshot of DFS status (the list of published datacans) through the *Configuration* mechanism. The snapshot is stored in a configuration file. Figure 1 shows the structure of DFS.

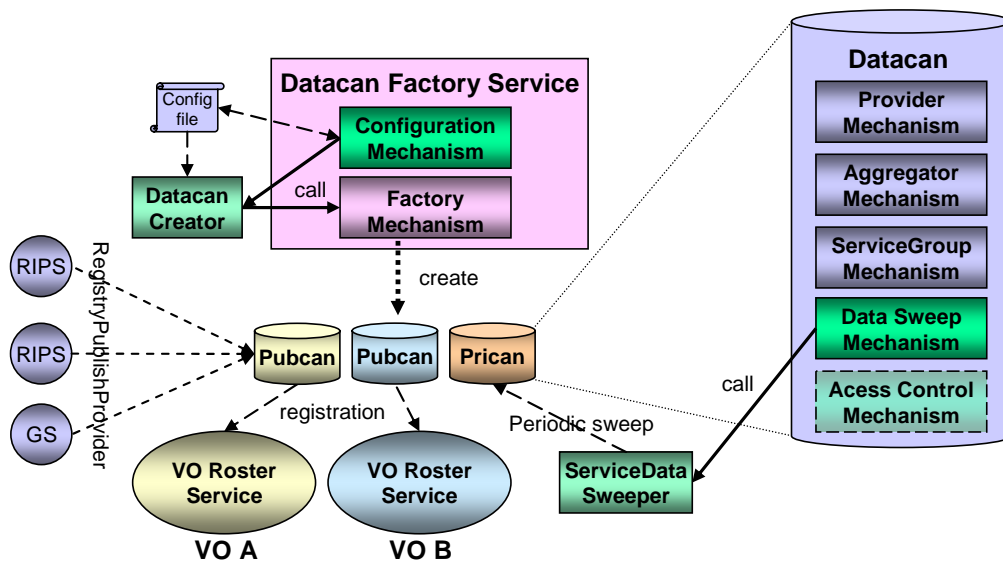
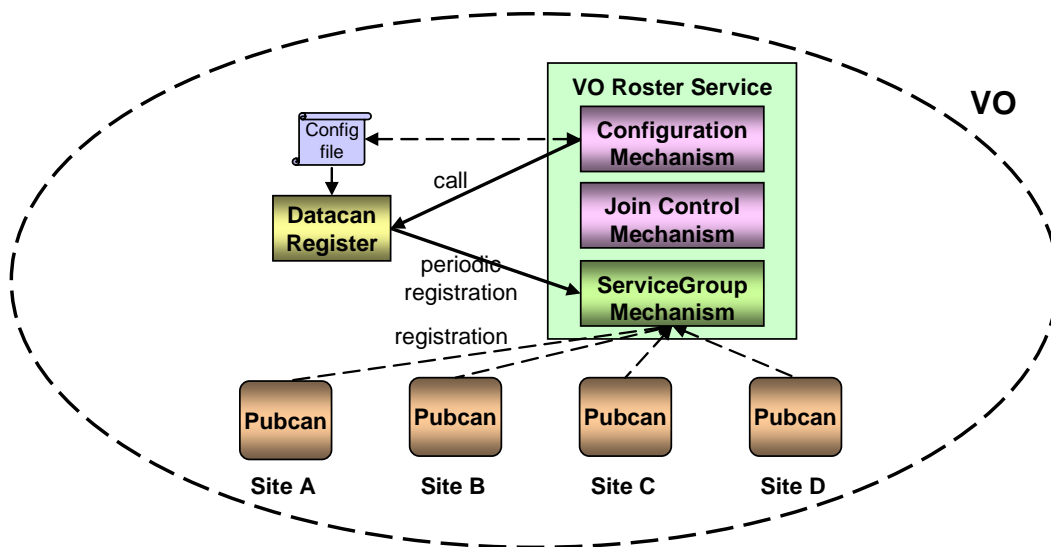


Figure 1. The structure of Datacan Factory Service

### 1.2.2. VO Roster Service (VRS)

Only one VRS exists in a VO because it typifies a VO. It manages the participants of VO and provides a registration interface to the VO. The structure of VRS is as Figure 2.



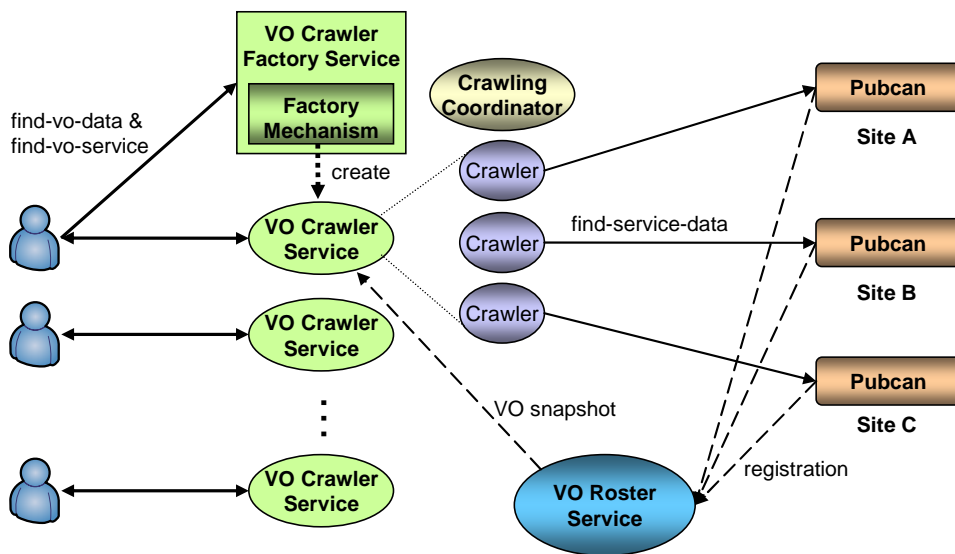
**Figure 2. The structure of VO Roster Service**

It provides the following functionalities:

- 1) It uses the *ServiceGroup* mechanism to register/unregister a datacan to its own VO.
- 2) It admits only a pubcan. The registration of a prican is rejected. The *Join Control* mechanism does this.
- 3) It makes use of the *Configuration* mechanism to store the snapshot of VO status (the list of registered datacan). But the status of resources in a VO changes dynamically. This may make the maintenance of the VO snapshot difficult. A *DatacanRegister* executes periodic registrations to preserve this.

### 1.2.3. VO Crawler Factory Service (VCFS)

VCFS provides a user with VO information. Only one VCFS exists in a VO because it is basically a service for a VO like VRS, but we recommend a site-based deployment of this service to avoid heavy load from plenty of users in a VO. This enables the load to be decentralized to each site in the VO. Figure 3 illustrates the structure of DFS.



**Figure 3. The structure of VO Crawler Factory Service**

It provides the following functionalities:

- 1) This service has two query options. The one of two options is the *find-vo-data*, which crawls on VO information for a user. The other is the *find-vo-service*, which provides the location (GSH: Grid Service Handle) of a Grid service in a VO for a user. Actually, the *find-vo-service* is a special form of the *find-vo-data* to serve the convenience of a user.
- 2) It also creates the VO Crawler Service (VCS) using the Factory mechanism to protect a user session. VCS gains a VO snapshot from VRS. To achieve the efficiency of query, VCS creates the *Crawlers* corresponding to each participant (pubcan) using the Thread mechanism. Each crawler uses the OGSi *find-service-data* to query its own pubcan. The *CrawlingCoordinator* orchestrates each crawler's behavior.

### 1.2.4. MoreDream Providers

MoreDream Information Providers provide lots of information about data replication as well as computing resource. It is based on the use of provider execution mechanism. They are composed of MceScriptProvider and MseScriptProvider. MceScriptProvider provides lots of resource information used in K\*Grid. It conforms to Glue schema and extends it. MseScriptProvider provides information about data replications. It defines a new information schema related to storage elements. You can easily obtain information that is produced in MCAT-enabled SRB (Storage Resource Broker) Server. If you'd like to add new information, you have only to create your own provider and register it to RIPS.

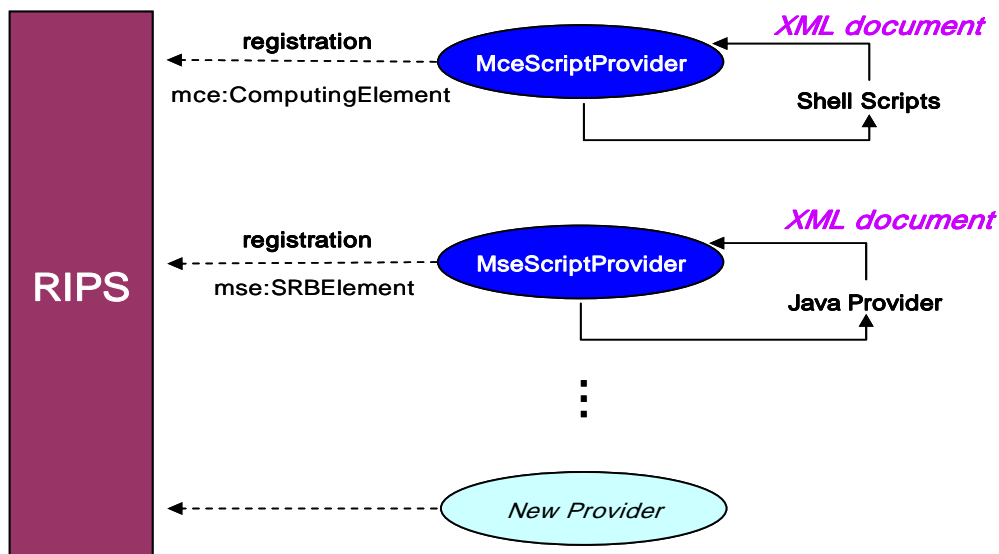


Figure 4. The registration of MoreDream Information Providers

### 1.3. Schema

GAIS uses the MoreDream schema, which adds and extends GLUE (Grid Laboratory Uniform Environment) schema. The schema is categorized into two element; MoreDream Computing Element (MCE) and MoreDream Storage Element (MSE). MCE enhances the computing element of GLUE schema 1.1 for supporting GRASP (Grid Resource Allocation Services Package), a resource management component of MoreDream. MSE is formed by processing the data replication information of SRB and it will be changed to agree to the storage element of GLUE schema later.

Table 1 shows the content of MCE.

Table 1 MoreDream Computing Element

Category	Object	Description	Unit
ComputingElement	Name	ComputingElement name	
	UniqueID	ComputingElement ID	
Info	LRMSType	Local Resource Manager type	
	LRMSVersion	Local Resource Manager version	
	GRAMVersion	GRAM version	
	HostName	Host name	
	GateKeeperPort	GateKeeper port	
	TotalCPUs	Total CPUs	
State	Staus	Queue status	
	TotalJobs	Total Jobs	

	RunningJobs	Running Jobs	
	WaitingJobs	Waiting Jobs	
	FreeCPUs	Free CPUs	
Policy	HostName	Host name	
UserStorage	LocalID	Local user ID	
	DN	User DN	
	Quota	Local user Quota	MB
	DefaultCapacity	Local user default Quota	MB
Job	GlobalID	Global Job ID	
	LocalID	Local Job ID	
	LocalOwner	Local Owner ID	
	GlobalOwner	Global Owner ID	
	Status	Job status	
Cluster	Name	Cluster name	
	UniqueID	Cluster ID	
SubCluster	Name	SubCluster name	
	UniqueID	SubCluster ID	
Filesystem	Name	File system name	
	Root	File system root	Path
	Size	Total size	MB
	AvailableSpace	Available space	MB
	ReadOnly	Read only or not	T/F
	Type	File system type	eg. NFS
Processor	Vendor	CPU vendor name	
	Model	Model name	
	Version	CPU version	
	Clockspeed	CPU Clock speed	MHz
	OtherProcessorDescription	Other description	
MainMemory	RAMSize	RAM size	MB
	RAMAvailable	Available RAM size	MB
	VirtualSize	Virtual RAM size	MB
	VirtualAvailable	Available virtual RAM size	MB
ProcessorLoad	Last1Min	1-minute average processor availability	%
	Last5Min	5-minute average processor availability	%
	Last15Min	15-minute average processor availability	%

OperatingSystem	Name	OS name	
	Release	OS Release #	
	Version	OS version	
NetworkAdapter	Name	Interface name	
	IPAddress	IP address	IP addr
	MTU	MTU size	Byte
	OutboundIP	OutboundIP or not	T/F
	InboundIP	InboundIP or not	T/F
Host	Name	Host name (Computation Element)	
	UniqueID	Host ID	
ProcessorLoad	Last1Min	1-minute average processor availability	%
	Last5Min	5-minute average processor availability	%
	Last15Min	15-minute average processor availability	%
MainMemory	RAMSize	RAM size	MB
	RAMAvailable	Available RAM size	MB
	VirtualSize	Virtual RAM size	MB
	VirtualAvailable	Available virtual RAM size	MB

Table 2 describes the content of MSE.

**Table 2 MoreDream Storage Element**

Category	Object	Description	Unit
SRBElement	CollectionName	Collection name	
	UserName	User name	
	ServerLocation	SRB server location	IP addr
SRBResources	CollectionName	Collection name	
	UserName	User name	
	ServerLocation	SRB server location	IP addr
SRBResource	ResourceName	Resource name	
	ResourceLocation	Resource location	IP addr
	ResourceType	Resource type	
	ResourceClassName	Resource class name	
	AdminName	Admin name	
	DomainDesc	Domain description	
	ZoneID	MCAT Zone ID	
SRBReplicas	CollectionName	Collection name	

	UserName	User name	
	ServerLocation	SRB server location	IP addr
SRBReplica	CollectionName	Collection name	
	UserName	User name	
	ServerLocation	SRB server location	IP addr
	FileName	File name	
	FileSize	File size	Byte
	FileType	File type	
ReplicaDetail	FileReplicationID	replica ID	
	ResourceLocation	Resource location	IP addr
	ResourceName	Resource name	

## 1.4. User Interface

There are now ways in which you can view VO information collected by GAIS or manage GAIS itself: the GAIS portlets and the GAIS PortType panels.

- 1) GAIS portlets: They offer resource information, service information or data replication information of a VO to users.
- 2) GAIS PortType panels: They enable a system administrator to control GAIS services.

The user interfaces will be added continuously as the version of GAIS is up.

## 1.5. Features

GAIS whose aim is to facilitate the management of information in Grid has the following features.

- 1) Dynamic management of directory server (datacan): GAIS can create a datacan easily whenever needed and can also destroy it freely. This enables a resource owner to share his resource according to his policies. For example, he can publish a datacan, which contains the entire information of his resource, for VO A, whereas the other datacan, which holds half of the information, for VO B.
- 2) Flat network of directory servers: The network of directory servers in GAIS is not configured hierarchically. Instead, it is flat. This has some merit. First, information is not duplicated. In hierarchy, higher level directory overlaps the information of lower level. Second, consistent synchronization of information is guaranteed. Hierarchical structure may pollute the



## 2. Installation and Configuration

### 2.1. Requirements

To install and use GAIS, you need a Linux system and the following softwares.

#### 1) Hardware

- \* Linux System

#### 2) Software

- \* OS: Linux (Redhat 7.3 or more except Redhat 8.x are recommended)

- \* Globus Toolkit 3.2.x

- \* Gridsphere 2.0.x

: If you want to use GAIS portlets, Gridsphere should be installed.

- \* OpenPBS (Portable Batch System)

: If you'd like to obtain the information about cluster, OpenPBS should be installed

- \* SRB Account

: If you'd like to obtain the information that is produced in MCAT-enabled SRB Server, contact the admin of MCAT-enabled SRB Server and obtain it.

### 2.2. Installing required software

#### 2.2.1. Installing Globus Toolkit

The information about Globus Toolkit can be found at: <http://www.globus.org>

#### 2.2.2. Installing Gridsphere

The information about Gridsphere can be found at: <http://www.gridsphere.org>

#### 2.2.3. Installing OpenPBS

The information about OpenPBS can be found at: <http://www.openpbs.org>

### 2.3. Installing GAIS

#### 2.3.1. Download and extract

Download gais\_v1.0.tar.gz from GAIS web site <http://kmi.moredream.org/MoreDream/GAIS/>

Untar the distribution file and move to gais\_v1.0.

```
$ tar xvfz gais_v1.0.tar.gz
```

```
$ cd gais_v1.0
```

You can find some files & directories as follows:

```
gais_v1.0
```

```
|-- INSTALL.txt
```

```
|-- README.txt
```

```
|-- build.xml                Ant build script
```

```
|-- gais-datacanFactory      Datacan Factory Service
```

```
|-- gais-voRoster            VO Roster Service
```

```
|-- gais-voCrawlerFactory    VO Crawler Factory Service
```

```
|-- gais-providers
```

```
    |-- mceScriptProvider    MCE Information Provider
```

```
    |-- mseScriptProvider    MSE Information Provider
```

```
|-- gais-portlets            Portlets
```

## 2.3.2. Compile & Installation

Basic GAIS package includes source files. If you want to obtain binary files (coming soon), visit GAIS web site <http://kmi.moredream.org/MoreDream/GAIS/>.

It is very simple to install. As we mentioned above, GAIS is composed of 3 Grid Services, 2 Information Providers and 1 Portlets. It is your choice whether whole components are installed or not.

Note 1: We assume that GT3.x is installed and GLOBUS\_LOCATION is set

Note 2: Current directory is gais\_v1.0

(1) Installation of GAIS Services & Providers

: First, must be the administrator of GT (ex. globus).

: Second, check \$GLOBUS\_LOCATION environment variable.

: Third, select the proper ant <target> according to your purpose and run it.

```
$ ant { deployAll | deployVoServices | deployDatacanWithVoCrawler |  
        deployDatacanWithProviders | deployProviders }
```

: Next, edit the config files of each services and providers (See 2.3.3 Configuration section).

: Finally, run GT container.

```
$ cd $GLOBUS_LOCATION
```

```
$ bin/globus-start-container
```

: You should see the following line in the output.

```
http://<hostname>:<port>/ogsa/services/base/gais/DatacanFactoryService
```

```
http://<hostname>:<port>/ogsa/services/base/gais/VoCrawlerFactoryService
```

```
http://<hostname>:<port>/ogsa/services/base/gais/VoRosterService
```

## (2) Installation of GAIS Portlets

: First, must be the administrator of Jakarta Tomcat (ex. root)

: Second, check \$CATALINA\_HOME environment variable

: Third, edit build.properties to modify location of Gridsphere source and build directory

```
$ edit build.properties
```

: Next, just run the following ant build script

```
$ ant install
```

: Finally, restart Jakarta Tomcat container.

```
$ cd $CATALINA_HOME
```

```
$ bin/shutdown.sh
```

```
$ bin/startup.sh
```

## 2.3.3. How to set up GAIS services

After 3 GAIS grid services and GAIS providers are installed, you can test your installation.

Before test, you should edit some files. After installation of GAIS, you can see files that start with gais-\* in \$GLOBUS\_LOCATION/etc. Now, we will explain about the configuration files.

### 1) How to create a datacan

You can create a datacan which participates in a VO by editing gais-data-factory-config.xml and adding a **datacan** element in the **publishedDatacans** element. At below example, the "DomainIndexService" and "Sample-VoMember" datacan will be created. The type of datacan will be "PUBLIC" or "PRIVATE." PUBLIC datacan is open to a VO. On the contrary, PRIVATE datacan is used in only local domain.

...

```
xmlns="http://www.moredream.org/namespaces/2004/12/datacan_factory"
```

...

```
<publishedDatacans>
```

```
    <datacan    type="PRIVATE"    name="DomainIndexService"    desc="Index
```

```

Service for my domain"/>
    <datacan type="PUBLIC" name="Sample-VoMember" desc="Index Service
for sample VO"/>
</publishedDatacons>
...

```

## 2) Datacan setup

gais-datacan-config.xml contains **serviceDataSweeper**, **installedProviders** and **executedProviders** elements. The serviceDataSweeper element assigns the execution period (the unit is second) of MoreDream Service Data Sweeper. The **targetData** element, a child of serviceDataSweeper element, contains the name and namespace of a Service Data to be checked periodically. The installedProviders element specifies the core Service Data Providers. One or more piece(s) of service data is produced by each execution of each Service Data Provider specified in the executedProviders.

Note: For MoreDream Service Data Sweeper to operate correctly, you should synchronize the time clock of resource in a VO.

```

...
xmlns="http://www.moredream.org/namespaces/2005/01/datacan"
...
<serviceDataSweeper period="30">
    <targetData namespace="http://www.moredream.org/ce/1.0"
name="ComputingElement"/>
    <targetData namespace="http://www.moredream.org/se/1.0"
name="SRBElement"/>
</serviceDataSweeper>
...

<installedProviders>
    <providerEntry
class="org.moredream.ogsa.impl.base.providers.servicedata.impl.MceScri
ptProvider" />
...
</installedProviders>
...

```

```

<executedProviders>
  <provider-exec:ServiceDataProviderExecution>
    <provider-exec:serviceDataProviderName>
      MceScriptProvider
    </provider-exec:serviceDataProviderName>
    <provider-exec:serviceDataProviderImpl>
      org.moredream.ogsa.impl.base.providers.servicedata.impl.MceScript
Provider
    </provider-exec:serviceDataProviderImpl>
    <provider-exec:serviceDataProviderArgs>
    </provider-exec:serviceDataProvider Args>
    <provider-exec:serviceName>
xmlns:mce="http://www.moredream.org/ce/1.0">
      mce:ComputingElement
    </provider-exec:serviceName>
    <provider-exec:refreshFrequency>30</provider-exec:refreshFrequency>
    <provider-exec:async>true</provider-exec:async>
    </provider-exec:ServiceDataProviderExecution>
  ...
</executedProviders>
  ...

```

### 3) How to register datacans to GAIS-Vo-Roster Service

You can register some datacans by editing gais-vo-roster-config.xml. **datacan** element, a child of **registeredDatacans** element, specifies the domain, handle and description of a registered datacan. At below example, the Sample-VoMember datacan will be registered in the VO managed by this VO roster service. The **period** attribute in **datacanRegister** element means how often this VO roster service tries to register datacan entries to maintain registration information in spite of the fault and error of resource in the VO.

```

...
xmlns="http://www.moredream.org/namespaces/2005/01/vo_roster"
...

<datacanRegister period="10"/>

```

```

<registeredDatacans>
  <datacan domain="localhost.localdomain"

    handle=http://127.0.0.1:8080/ogsa/services/base/gais/DatacanFactoryService/Sample-VoMember
    desc="VoMember for Sample VO in localhost.localdomain"/>
</registeredDatacans>

```

#### 4) VO Crawler Factory Service setup

You first should configure the location of VO roster service to query service data and find service location in a VO by editing `gais-vo-crawler-config.xml`. Register it by adding `voRoster` element. At below example, a local `VoRosterService` will be used.

```

...
xmlns="http://www.moredream.org/namespaces/2005/01/vo_crawler"
...
<voRoster
  handle="http://127.0.0.1:8080/ogsa/services/base/gais/VoRosterService"
/>
...

```

#### 5) MCE provider setup (lrms-info.conf)

Configure the type of your local scheduler by editing `lrms-info.conf`.

```

$ vi lrms-info.conf
...
LRMSType=PBS           # PBS or FORK
...

```

#### 6) MSE provider setup.

When MSE information provider initiates, it need some information about GSI. They are the file path of user proxy and CA. This information is in `gais-mse-proxy.properties` file.

```

$ vi gais-mse-proxy.properties
proxy=/tmp/x509up_<uid>
gridcas=/etc/grid-security/certificates/<hash#>.0

```

## 7) Registering a Resource Service to an Datacan Factory Service

The Service Data of any Grid service can be registered to GAIS Datacan Factory Service using the core RegistryPublishProvider. The RegistryPublishProvider should be configured in the WSDD of each resource service (for example, RIPS or MMJFS) by the administrators of those services.

Note: The administrator of the Datacan Factory Service does not need to take action in order for a new resource service to register.

(1) Create an XML configuration file for the registration in \$GLOBUS\_LOCATION/etc/data\_registration\_config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<serviceConfiguration
  xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"
  xmlns:aggr="http://www.globus.org/namespaces/2003/09/data_aggregator"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <registrations>
    <registration
      registry="http://127.0.0.1:8080/ogsa/services/base/gais/DatacanFactory
      Service/DomainIndexService"
      keepalive="true"
      lifetime="120"
      remove="true">
      <aggr:DataAggregation>
        <ogsi:params>
          <aggr:AggregationSubscription>
            <ogsi:serviceDataNames>
              <ogsi:name
xmlns:mce="http://www.moredream.org/ce/1.0">mce:ComputingElement</ogsi
:name>
              </ogsi:serviceDataNames>
            <aggr:lifetime>60000</aggr:lifetime>
          </aggr:AggregationSubscription>
```

```

    </ogsi:params>
    </aggr:DataAggregation>
  </registration>
</registrations>
</serviceConfiguration>

```

(2) Edit the fields as appropriate. Most importantly, edit the **registry** attribute to refer to the Datacan Factory Service instance (datacan) you want to register with. **DataAggregation** element contains service data name which subscribes to the instance.

(3) You must add the RegistryPublishProvider operation provider to the resource service's deployment descriptor in the Server Configuration file (server-config.wsdd).

(4) Find **service** element (for example, <service name="base/gram /ResourceInformationProviderService" ...) definition for a resource service.

(5) Then, add the following parameters in the **service** element:

```

    <parameter name="operationProviders" value="org.globus.ogsa.impl.core.registry.
RegistryPublishProvider"/>
    <parameter name="registrationConfig" value="etc/data_registration_config.xml"/>

```

## 8) Registering a Grid Service to an Datacan Factory Service

Any Grid Service Handle (GSH) can be registered to GAIS Datacan Factory Service using the core RegistryPublishProvider. The RegistryPublishProvider should be configured in the WSDD of each Grid service (for example, PingService) by the administrators of those services.

Note: The administrator of the Datacan Factory Service does not need to take action in order for a grid service to register.

(1) Create an XML configuration file for the registration in \$GLOBUS\_LOCATION/etc/service\_registration\_config.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<serviceConfiguration
  xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"
  xmlns:aggr="http://www.globus.org/namespaces/2003/09/data_aggregator"

```

```

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<registrations>
  <registration
registry="http://127.0.0.1:8080/ogsa/services/base/gais/DatacanFactory
Service/ DomainIndexService "
    keepalive="true"
    lifetime="120"
    remove="true">
  </registration>
</registrations>
</serviceConfiguration>

```

(2) Edit the fields as appropriate. Most importantly, edit the **registry** attribute to refer to the Datacan Factory Service instance (datacan) you want to register with.

(3) You must add the RegistryPublishProvider operation provider to the grid service's deployment descriptor in the Server Configuration file (server-config.wsdd).

(4) Find **service** element (for example, <service name="core/ping/PingService"...>) definition for a grid service.

(5) Then, add the following parameters in the **service** element:

```

<parameter name="operationProviders"
value="org.globus.ogsa.impl.core.registry. RegistryPublishProvider "/>
<parameter name="registrationConfig"
value="etc/service_registration_config.xml"/>

```

## 9) Registering GAIS information providers to RIPS

GAIS information providers should be registered to RIPS as follows (\$GLOBUS\_LOCATION/etc/rips-service-config.xml). They produce {http://www.moredream.org/ce/1.0}ComputingElement and {http://www.moredream.org/se/1.0} SRBE lement as Service Data.

(1) Register MCE Information Provider.

```
$ vi $GLOBUS_LOCATION/etc/rips-service-config.xml
```

```

...
<installedProviders>
  <providerEntry
    class="org.globus.ogsa.impl.base.providers.servicedata.impl.S
criptExecutionProvider" handler="jobDataHandler"/>
  <providerEntry
class="org.globus.ogsa.impl.base.providers.servicedata.impl.HostScript
Provider" />
    <providerEntry
class="org.moredream.ogsa.impl.base.providers.servicedata.impl.MceScri
ptProvider" />
  </installedProviders>
...
<executedProviders>
...
<provider-exec:ServiceDataProviderExecution>
  <provider-exec:serviceDataProviderName>
MceScriptProvider
  </provider-exec:serviceDataProviderName>
  <provider-exec:serviceDataProviderImpl>
org.moredream.ogsa.impl.base.providers.servicedata.impl.MceScrip
tProvider</provider-exec:serviceDataProviderImpl>
  <provider-exec:serviceDataProviderArgs>
</provider-exec:serviceDataProviderArgs>
  <provider-exec:serviceName
xmlns:mce="http://www.moredream.org/ce/1.0">mce:ComputingElement
  </provider-exec:serviceName>
  <provider-exec:refreshFrequency>30
  </provider-exec:refreshFrequency>
  <provider-exec:async>true</provider-exec:async>
</provider-exec:ServiceDataProviderExecution>
...
</executedProviders>
...

```

(2) Register MSE Information Provider.

```

$ vi $GLOBUS_LOCATION/etc/rips-service-config.xml
...
<installedProviders>
  <providerEntry
    class="org.globus.ogsa.impl.base.providers.servicedata.impl.S
criptExecutionProvider" handler="jobDataHandler"/>
  <providerEntry
class="org.globus.ogsa.impl.base.providers.servicedata.impl.HostScript
Provider" />
  <providerEntry
class="org.moredream.ogsa.impl.base.providers.servicedata.impl.SRBScri
ptProvider" />
  </installedProviders>
...
<executedProviders>
...
<provider-exec:ServiceDataProviderExecution>
  <provider-exec:serviceName>
SRBScriptProvider</provider-exec:serviceName>
  <provider-exec:serviceDataProviderImpl>
org.moredream.ogsa.impl.base.providers.servicedata.impl.SRBScrip
tProvider
  </provider-exec:serviceDataProviderImpl>
  <provider-exec:serviceDataProviderArgs>
  </provider-exec:serviceDataProviderArgs>
  <provider-exec:serviceName>
xmlns:mse="http://www.moredream.org/mse/1.0">mse:SRB
  </provider-exec:serviceName>
  <provider-exec:refreshFrequency>30</provider-
exec:refreshFrequency>
  <provider-exec:async>true</provider-exec:async>
  </provider-exec:ServiceDataProviderExecution>
...
</executedProviders>
...

```

## 10) Enabling OGSA Service Browser GUI

If you want to use the OGSA Service Browser GUI, you will need the GUI control panels for the GAIS Service, which is configured in a different file. Add the following lines to the client-gui-config.xml file in \$GLOBUS\_LOCATION

```
...  
<panel portType="DatacanFactoryPortType"  
    class="org.moredream.ogsa.gui.DatacanFactoryPortTypePanel" />  
<panel portType="VoRosterPortType"  
    class="org.moredream.ogsa.gui.VoRosterPortTypePanel" />  
<panel portType="VoCrawlerFactoryPortType"  
    class="org.moredream.ogsa.gui.VoCrawlerFactoryPortTypePanel" />  
...
```

## 2.3.4. Starting

### 2.3.4.1. Starting GAIS Service

Now, let's test your GAIS services installation. Verify whether the installation is okay or not as follows.

1) To start GT3.2.x container, run as GT administrator:

```
$ cd $GLOBUS_LOCATION  
$ bin/globus-start-container
```

2) To start GT3.2.x service browser, run:

```
$ bin/globus-service-browser
```

3) Find a VO Crawler Factory Service at Service Group Entry Inspection of ContainerRegistry Service. By double clicking of that, the VO Crawler Factory Service browser will be displayed.

4) Query "ComputingElement" Service Data with "http://www.moredream.org/ce/1.0" namespace at *find-vo-data* tab of *VO Query*. If "mce:ComputingElement" is shown, your installation is successful.

5) Query with "service" pattern at *find-vo-service* tab of *VO Query*. In case of successful installation, registered GSH will be shown.

### 2.3.4.2 Starting GAIS Portlets

Next, let's test your GAIS portlets installation. Verify whether the installation is okay or not as follows.

1) To start Jakarta Tomcat container, run as Tomcat administrator:

```
$ cd $CATALINA_HOME
```

```
$ bin/startup.sh
```

2) Browse the following URL

```
http://<hostname>:<port>/gridsphere/gridsphere
```

3) Login with the appropriate ID

4) Click the **Configuration** link of the **Information Service** tab

5) Configure the GSH of your VO Crawler Factory Service.

## 3. Using GAIS

How about your installation, successful or failed? We expect your successful installation & configuration. Now, let's use GAIS in real Grid environment. We assume that there is a KMI (Korea Middleware Initiative) VO at figure 6. GAIS services can be deployed as like following table 3.

**Table 3. KMI VO configuration**

Service/Provider	Machine name with the service/provider
DFS	sdd107, eros01, vega01
VRS	gais
VCFS	sdd107, eros01, vega01, gais
MCE provider	orin01,nova01, eros01, nova01 → Front node of cluster
MSE provider	sdd107

Note. To obtain the information that is produced in MCAT-enabled SRB Server, contact the admin of MCAT-enabled SRB Server and obtain a SRB account.

# KMI Testbed

- DFS: Datacan Factory Service
- VRS: VO Roster Service
- VCFS: VO Crawler Factory Service
- JSS: Job Submission Service
- GSS: Grid Scheduling Service
- RMS: Resource Mgmt Service
- MCEP: Moredream CE Provider
- MSEP: Moredream SE Provider

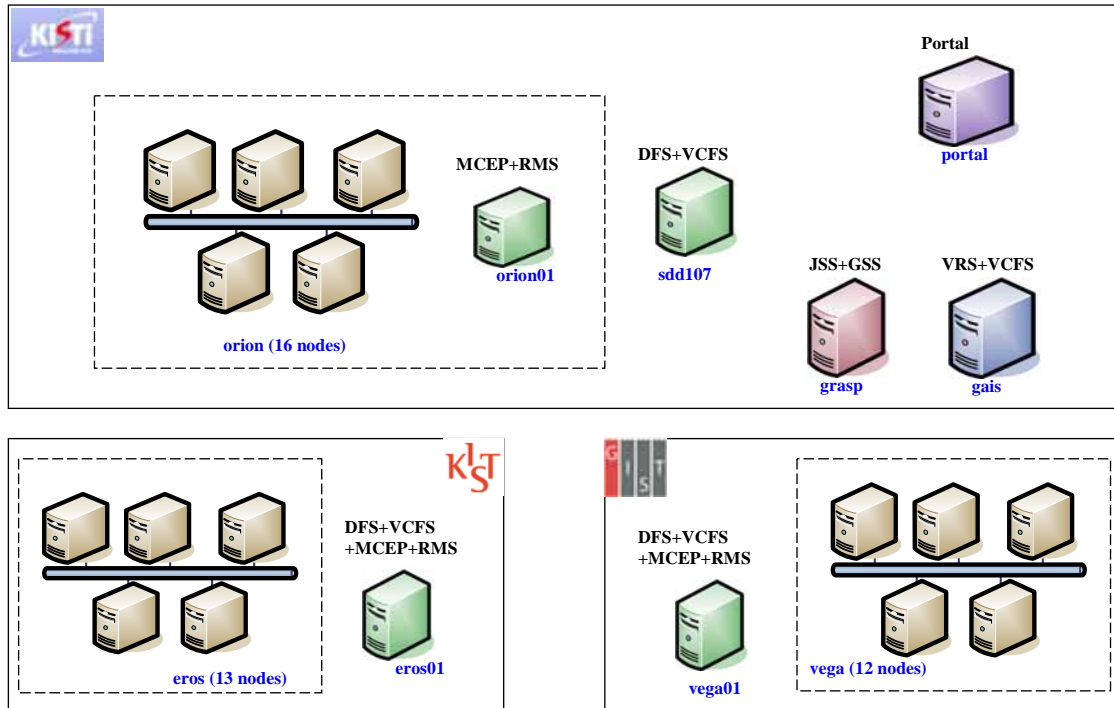


Figure 6 KMI VO

## 4. References

[1] <http://www.globus.org/>

[2] <http://www.moredream.org/gais.htm>

[3] <http://www.npaci.edu/DICE/SRB/>

[4] <http://ant.apache.org/manual/index.html>

[5] <http://www.gridisphere.org/gridsphere/docs/index.html>

[6] <http://java.sun.com/>

[7] [http://testbed.gridcenter.or.kr/software/OpenPBS/doc/v2.3\\_admin.pdf](http://testbed.gridcenter.or.kr/software/OpenPBS/doc/v2.3_admin.pdf)