DataGRID WP4 Gridification Task Definition

DataGRID Fabric Gridification Task

DRAFT version 20010406.01

Id

Abstract

1 Definition of the Gridification Task

The gridification task (GT) provides the mechanisms for grid-wide services to access the local fabric services and vice versa. It will hide protect the local fabric from the intrusion from other grid middleware into the fabric and enable site management and local control for compute centres.

At least logically, all interaction from the 'outside' grid with a compute centre, as well as all interaction of a compute centre with the outside world should be mediated by services provided by the Gridification task. The GT will not mediate between services that are local to the fabric.

2 Basic Functionality

Gridification of a local fabric concerns five basic subsystems. The ensemble of these subsystems constitutes the deliverable for the Gridification Task.

- local authorization for jobs posed to the fabric by grid services (LCAS)
- provide all local credentials needed for jobs allowed into the fabric (LCMAPS)
- mediate the job received from any grid entity to the local job management system (GjMS)
- supply aggregate or abstracted information content about the local fabric to external entities on the grid (GriFIS)
- provide a mechanism to support connections from individual farm nodes to locations outside the fabric, for those types of communication that cannot be supported by transferring predefined integral data elements (GridGATE)

For some of its functionality, the gridification subsystems rely on components provided by other WP4 tasks and on functionality provided by other DataGRID Work Packages. These will be indicated per subsystem.

One general remark concerns fabric-wide repositories. Many of the GT subsystem have a need to job publish information in a repository. The data in such repositories is not per-se classifiable as monitoring information. It does also not seem to fit with the configuration data, since the job data is relatively dynamic as opposed to machine and fabric configuration.

The GT task will user the monitoring subsystem for all auditing information generated by the subsystems. This auditing information should be logged and retained.

3 Local authorization or LCAS

Specification of functionality

- The user credential or authorization token received from a Grid authorization service will be received by the LCAS subsystem. This LCAS will verify the authorization in an iterative and extendible way.

- A series of basic authorization modules will be provided by the gridification task. These are: static user checking, static user banning, and the application of resource-independent policies. It will provide hooks to insert external authorization modules, *e.g.*, to apply resource-dependent and availability policies. These external modules are to be provided by the other WP4 sub tasks.

- The end result of the authorization chain will be a user certificate signed by the LCAS. It will include an authorization audit trail.

- Authorization modules provided by this subsystem itself will be: *(i)* static user checking against a ban list, *(ii)* application of high-level policy decisions that are dependent only on static sources like wall time, *(iii)* application of rules regarding external connectivity, based on a fixed list of allowed remote networks.

Dependence on other systems and subsystems

Grid-wide authorization by WP1 or Security group

The LCAS assumed a grid-wide Authorization service (CAS) exists, that will classify users or roles as being part of a group. Off-line arrangements between the local centres and the grid-wide CAS define high-level authorization for classes of users (example: NIKHEF will accept ATLAS, LHCb and ALICE jobs, but no CMS). The grid scheduler should take this authorization into account before posing jobs to a fabric.

The LCAS primary focus is on individual or role authorization. For this to work, the job credentials provided to the LCAS must[*] include a unique identification of the user or role who submitted the job (example: the DN as stated in the users personal certificate).

WP4 resource management and other tasks

For any internal WP4 subsystem that needs to influence the user authorization at job submission time, should provide plug-able modules that can be called by the LCAS. These modules will be supplied with the users credentials and the job description as supplied by the ComputeElement. The modules should return a boolean value whether or not to grant authorization.

In particular, the Resource Management task is invited to provide modules to implement accounting and quota-based authorization plug ins.

4 Local credential generation

Specification of functionality

- The credential mapping service (LCMAPS) will provide all credentials necessary to access services within the centre. It will only accept requests that can present a credential properly signed by the LCAS.

[*] in the sense of RFC 2119

- If the identity of the user exists within an administrative domain addressed by the job, the LCMAPS will issue credentials corresponding to this pre-existing identity.
- For those users who have no pre-existing identity within the administrative domain addressed, the LCMAPS will generate a new identity.
- The LCMAPS will maintain a repository of issued local identities. This repository will have access control rights associated with each individual credential.
- The LCMAPS will provide auditing logs (but is not prepared to store them).
- The LCMAPS will at least provide for generation of UNIX user IDs and group IDs.
- If a local fabric supports other authentication methods, like Kerberos, the LCMAPS may provide mappings for those systems. The availability of these methods and the authentication and authorization types will be dependent on the underlying mechanism. *Details are to be specified.*
- The issues local credentials may have a limited lifetime. For UNIX uids and gids, the LCMAPS service will have the possibility to make the mapping persistent and re-usable.
- The LCMAPS must create and issue local credentials for every authorised user. No additional authorization is allowed at this level. Sole reason for refusing the mapping is lack of resources at this level, *e.g.*, no more uids available.

Dependence on other systems and subsystems

Local operating system

The LCMAPS will use the underlying operating systems support for creating the user mapping: e.g. for generating user leases on a single workstation it will use the passwd and group files. If the underlying operating system does not support more advanced user authorization mechanisms than plain uid/gid, the LCMAPS will not provide more.

Other WP4 subsystems

The LCMAPS will need to be called by either the GjMS or the Resource Manager on startup and termination of a job or request. The information provided should at least be the unique job ID and the users credential. In case the LCMAPS was used to allocate a local credential for a non-job request like storage, it will be called when all entities related to that single request have been removed. The Resource Manager will need to contact the LCMAPS repository to obtains the local credentials to use. This will be indexed using the users original individual subject name (cert DN of the individual's credential).

5 Grid job mediating service

Specification of functionality

- The Grid job mediating service (GjMS) will receive the job description from the ComputeElement.
- The GjMS will assign a per-fabric unique local job ID to every incoming job and maintain a repository of current local jobs. [*discuss please*]

- The GjMS will call the LCAS and the LCMAPS service and act according to the output of those subsystems. In case of failure, it will notify the ComputeElement and not call any further fabric-internal subsystems.
- The GjMS will contact the Resource Management subsystem and present the original and entire job there. The syntax and semantics of the job submission will be unchanged. The Resource Management subsystem will not be contact if authorization or credential mapping failed.
- The GjMS will notify the LCMAPS subsystem after a job has been 'finished' by the Resource Manager subsystem.

Dependence on other systems and subsystems

Other Grid systems and WPs

This subsystem needs the job description and the user's or role's credential from the ComputeElement.

Other WP4 subsystems

The GjMS will no nothing with the job, except for calling the different WP4 subsystems. It needs a ResourceMngt subsystem capable of accepting any job with sufficient authorization.

It will provide to the other components a repository in which to look up references to the user grid credential, local credentials and job description, based on the local unique job ID.

6 Grid Fabric Information Service

The GriFIS is still a very immature concept. It might not be needed at all, if the monitoring task is prepared to provide aggregate information to the GIS that is compatible with the requirements from WP1 and WP3.

It is also heavily dependent on the architecture of the WP4 monitoring task, since even a full-scale GriFIS will only be prepared to provide (the implementation of) the algorithms needed to aggregate and abstract the Grid-relevant information. It will not provide the framework in which to extract and publish this information.

Specification of functionality

- The subsystem will provide routines that can be called to calculate monitoring metrics that are needed by other Grid WPs.
- The subsystem will publish this information in the fabric's GIS.

Dependence on other systems and subsystems

Other Grid systems and WPs

It needs the monitoring framework provided by WP3.

Other WP4 subsystems

It needs the subsystems of the monitoring task. In particular, it will just provide an additional correlation engine for aggregate fabric information. It will use the event subscription options of the monitoring task.

It will also use the high-level data provided by the configuration database, but the GriFIS is not prepared to abstract information based on individual software packages. Example: it will not be able to induce from the fact the gcc, CMT, CERNlib, Gaudi, Geant, ROOT and SicbMC are there, that there is a functional LHCb Monte Carlo generation service available. This information has to be provided in the Configuration database, using the information provided by the respective VO's.

7 Grid Gateway interface for worker nodes

The Grid Gateway (GridGATE) will provide a method for streaming connections (data pipes for visualisation, interactive sessions, MPI, etc) to be channelled out of the local fabric onto the wide-area Grid environment.

Specification of functionality

- Provide a gateway system to create and destroy streaming connections between individual worker nodes within a fabric to the external fabric boundary. This fabric boundary is defined as being on the same connectivity level as the ComputeElement.
- **If** the internal connectivity of the nodes uses a network protocol or network addressing space different from the one used on the external side of the ComputeElement **and** it is required that connectivity is provided to a final destination that cannot be tunnelled transparently, **then** the GridGATE subsystem will maintain a repository where the mapping between intra-fabric connectivity and the externally visible connectivity is stored. This repository will be an integral part of the fabric's GIS.
- The GridGATE subsystem will not guarantee that a persistent communications channel can be established in a fabric that allows for job migration between nodes.
- The GridGATE subsystem will assign the ports and port-ranges to be assigned to a specific job or a specific machine.

Dependence on other systems and subsystems

Other Grid systems and WPs

The jobs supplied to the fabric by WP1 should contain explicitly in their description that an outgoing communications channel is needed and what its final destination will be.

Other WP4 subsystems

It should be told by the Resource Manager that there is a need for a communications channel by a particular job. The availability of communications channels will be announces to the Resource Manager on request.

The subsystem will provide an authorization module to check the validity of the communications destination requested. This is a static check only; other checks should be performed by the module provided by the Resource Management Task.

8 Overview

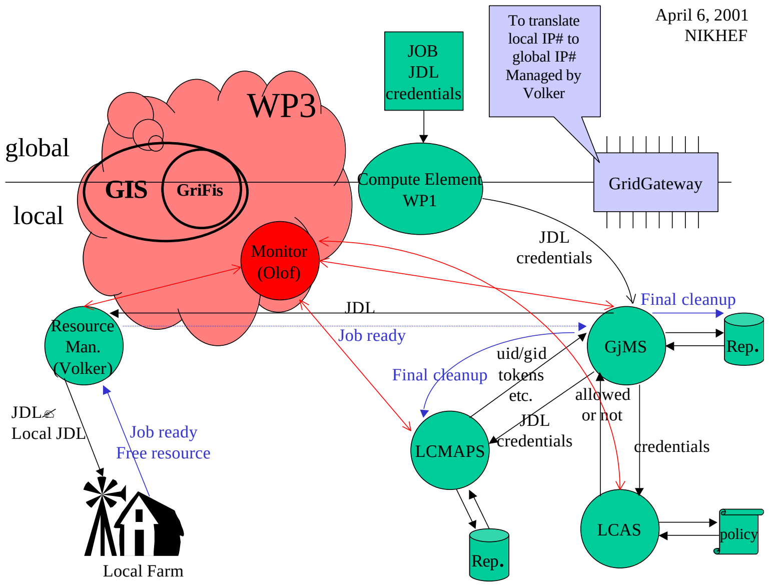

Figure 1: *A arrowy overview of the WP4 gridification task. The figure represents a use case of job submission, augmented with the GriFIS and GridGATE functionality.*

Part I
Use Cases

9 Case: Submit a job to a fabric

This job will go through the following steps:

1. ComputeElement will have received a job, consisting of a JDL description and a user grid-authorization token (G-UAT) from the users favourite CAS. Some way or another, the executable and the associated data files are already in plase (for the local credential and UID problem see lateron).

2. The ComputeElement will contact the GjMS and present the JDL and the G-UAT.

3. The GjMS assigns the unique job ID and stores it in the Job Repository. The attributes are the intact JDL and the G-UAT.

4. The GjMS calls LCAS:

   (a) LCAS ontains the basic policy (*e.g.*, `order=deny,allow`).

   (b) LCAS calls the registered authroization modules in sequence, with the JDL and G-UAT as input. The modules will say "yes" or "no". Modules are only called until a definite authorization decision has been reached. The modules called are: *i)* the local user ban list *ii)* check wallclock time *iii)* the quota check, *module supplied by the RM task iv)* gridgate destination network check

   (c) the the final decision is "rejected", the GjMS returns the job to the compute element, stating that local authorization failed.

5. if the decision is "allowed", the GjMS passes the JDL and G-UAT to the LCMAPS system. This system checks, based on the users individual DN contained in the G-UAT, whether a local credential already exists. If so, this cred is assign to the G-UAT adn stored in the repository. The repository's key is the user DN, attributes are the UID, GID, Kerberos token, etc.

   If the identity' uid did not exist, a new one is allocated and stored in the repository.

6. The JDL and the pointer to the local credential mapping repository entry is passed to the local resource manager.

7. when the RM finished with the job, it calls the GjMS again.

8. The GjMS will inform LCMAPS that the job has finished. Depending on local policy, the user account is retained, temporaroty disabled for retained, or permanently erased. The erasure can only happen after the last subsystem holding a lock on the credentials has finished (i.e. a UID still held by the Data Manager).

The Data Manager may have a need to call the LCAS and LCMAPS services. In those cases, these subsystems shall have a access count associated with every entry.