Workload Management with Pilot Agents in DIRAC

LHCb Technical Note

Issue: Draft
Revision: 8

Reference: LHCB Comp 08-006
Created: 11th Dec 2007
Last modified: 11th June 2008

Prepared By: LHCb Computing Group
A. Tsaregorodtsev/Editor

Abstract

This note describes the use of the Pilot Agents paradigm in the DIRAC Workload Management System. The details about the job scheduling with generic Pilot Agents are given and security aspects are discussed.

Document Status Sheet

Table 1 Document Status Sheet

1. Document Title: [Project Name Qualification] User Requirements Document			
2. Document Reference Number: [Document Reference Number]			
3. Issue	**4. Revision**	**5. Date**	**6. Reason for change**
Draft	1	11 Dec 07	First version
Draft	2	14 Jan 08	LHCb note formatted
Draft	3	31 Jan 08	Abstract added
Draft	4	26 Mar 08	Figure added, updates after discussion in the Pilot WG
Draft	5	10 May 08	Additions following remarks of Igor
Draft	6	18 May 08	More modifications after internal DIRAC project discussion, and Pilot WG discussion
Draft	7	28 May 08	Comments by Maarten
Draft	8	11 June 08	More comments from Maarten and Mine

1. Overview

DIRAC is the LHCb Workload and Data Management System (WMS and DMS), which is developed to support production activities (simulation, reconstruction, re-processing, group analysis…) as well as user data analysis. Integrating into the same system the production and analysis activities allows optimization of the overall workload of the LHCb Virtual Organization (VO), applying efficiently the VO policies with respect to the prioritization of the tasks of different users and groups.

2. Workflow in the DIRAC WMS

Production or analysis activities consist of executing a number of applications on the distributed computing infrastructure. Hereafter we call these applications *"workload"*. The general life cycle of a workload in the DIRAC WMS consists of the following steps:

- Workload preparation: bundling together the executable and necessary files, specifying all the job parameters in a JDL description.
- Workload submission to the DIRAC WMS: sending the JDL, the files in the Input Sandbox and the user proxy to the DIRAC *Job Manager* and *Input Sandbox* services.
- Workload analysis by the WMS to evaluate its requirements in terms of resources; estimation of the priority applying the LHCb policies; insertion of the workload into the DIRAC *Central Task Queue*.
- Submission by the WMS of a generic script as a Grid job (called hereafter Pilot Job or PJ) with the same resource requirements as the original workload; submission of additional Pilot Jobs if the previous ones fail for some reason (e.g. aborted). Pilot Jobs are submitted with special credentials (role) only used for running this kind of jobs.
- Start of the Pilot Job on one of the Grid Worker Nodes (WN): installation of the DIRAC *Job Agent* software and check of the WN for its capacity (CPU time limit, disk space…) and software environment.
- Request for a Workload matching the WN capacities to the WMS *Task Queue.* Even if a Pilot Job is submitted with the resources requirements of a particular job, the WN capacity can allow taking other jobs in the *Task Queue* which have higher priority. So, other job than the one that triggered the Pilot Job submission can be picked up.
    - download of the Workload description;
    - download of the grid proxy of the owner of the Workload;
    - preparation of a *Job Wrapper* script for the execution of the Workload;
    - execution of the *Job Wrapper*. The execution can be performed either directly by the Pilot Job or by means of the ***glexec*** site policy enforcement tool, which eventually uses a site authorization service to resolve user's rights to run his/her workload on the site.
        - resolution of the workload Input Data and preparation of an XML file catalog with local replicas of the input files;
        - download of the Input Sandbox;
        - installation of the missing elements of the LHCb application software if needed;

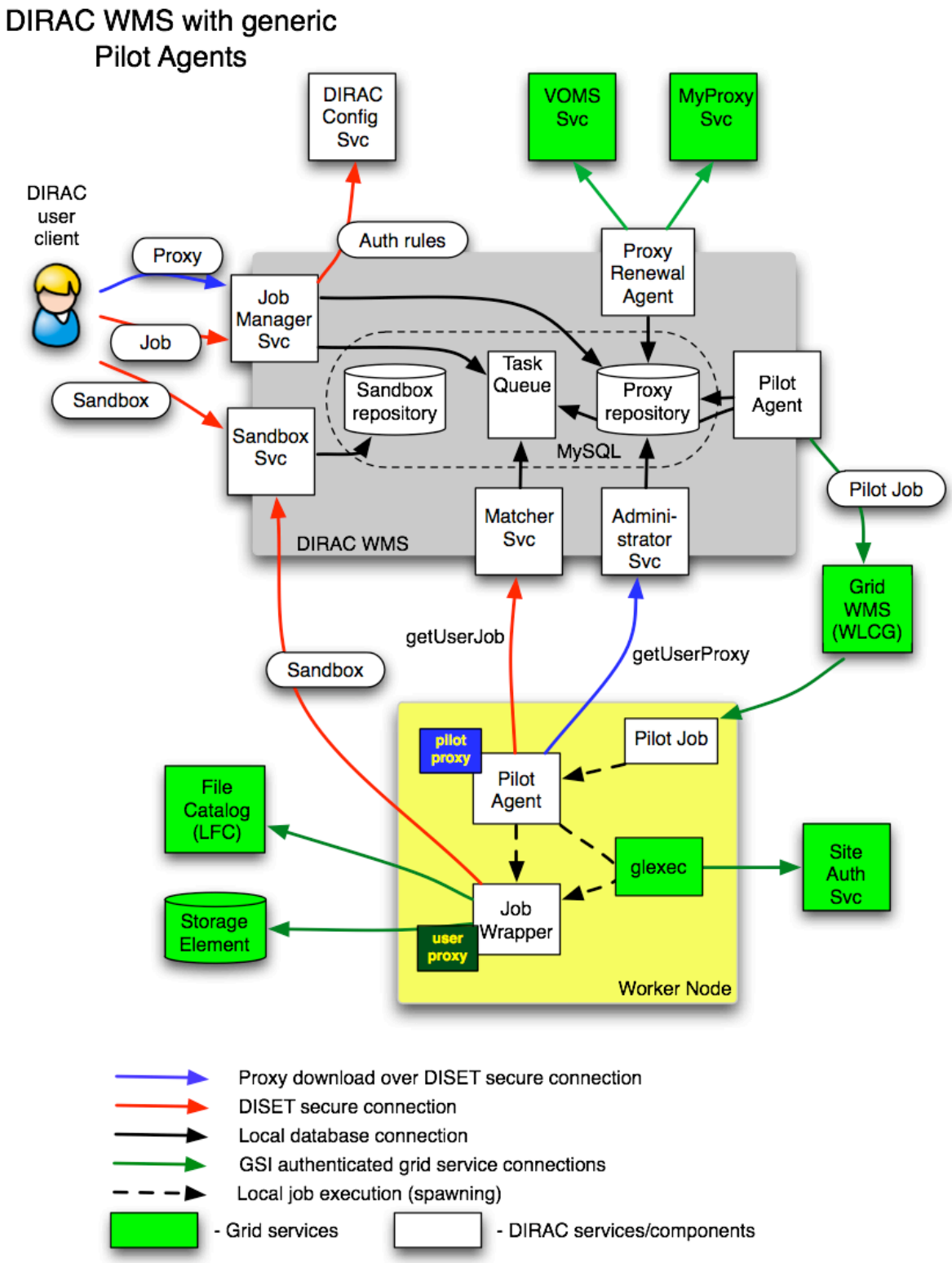

Figure 1 DIRAC WMS with generic Pilot Jobs.

The arrows are pointing from a component that initiates connection to a service which authenticates the client and possibly sends back the requested data.

- execution of the application in a separate thread with respect to the application watchdog process;
- after the application's end, upload of the results of the user Workload: Output Sandbox and Output Data.

- report of the resource consumption parameters to the DIRAC WMS (to be passed to the Accounting System).

In the following we describe in details all the steps in the user workload life cycle, paying special attention to the security aspects of the process.

3. User authentication and authorization

Various operations in the DIRAC WMS need authentication of the service requestors as well as evaluation of their access rights for various service functionalities. The application of both authentication and authorization is based on the description of the LHCb users in terms of their identities and roles within the LHCb Virtual Organization. This description is done by means of the VOMS database where the following information is stored for each user:

- User Distinguished Name (DN) – a unique user Grid Identifier;
- roles within the LHCb Virtual Organization that the user is eligible to exercise;
- short user name similar to the user login in the CERN AFS cell (stored as a Generic Attribute);
- possibly other user-related data stored as Generic Attributes.

The VO manager for each user provides this information at the time of his/her registration with the LHCb VO. The VO manager is responsible to maintain the user data up to date in case of any changes in the user identity or affiliation.

Users in LHCb can participate in more than one activity within the VO (e.g. various physics groups, production managers, software managers…). All the LHCb users belong to a single general *lhcb* group. The activities are expressed in the VOMS database as a number of VOMS Roles. The VOMS groups are not foreseen to be used in LHCb as considered being an unnecessary complication.

The DIRAC Services Framework implements its proprietary mechanism (within the DISET security layer of DIRAC) for user authentication and authorization that uses a description of the LHCb users and groups in the DIRAC Configuration Service. This description is derived from the VOMS information as a primary source by regularly executing scripts (cron jobs) that extract it. The DIRAC Users and Groups are thus in one to one correspondence with the VOMS Users and Roles. The User and Group information is available to all the distributed DIRAC components via the *Configuration Service*.

When an LHCb user is acquiring or renewing his/her Grid Credentials via his/her respective Certification Authority (CA), he/she uploads a long-lived proxy with duration equal to the lifetime of his/her certificate to the MyProxy LCG service. This is ensured by the LHCb user level grid utilities and made transparent for the users. The users are retaining their original certificates, which are used to generate proxies for the user working sessions. Later on, the MyProxy service is used by the DIRAC WMS to renew, as needed, the user proxy stored in the DIRAC Proxy Repository. Access to the MyProxy service is allowed only to the DIRAC WMS services and to the Grid WMS services, which use standard grid security enabled mechanisms for this purpose. A MyProxy server is configured with a list of trusted hosts authorized to renew proxies that must still be valid at the time of renewal. Users cannot obtain a renewed proxy by this mechanism; they are always obliged to use password-protected end-user certificates.

4. Brief DISET overview

The DIRAC secure Services Framework is called DISET and described in details in [1]. The DISET main features include:

- A special client-service secure protocol that allows for mutual client and server authentication using a mechanism compliant with the Grid Security Infrastructure (GSI) framework. It is based on X509 type user credentials, and allows for limited life time credential proxies;
- the possibility to define versatile authorization rules for Remote Procedure Calls (RPC) with the granularity level of per service method and per user or group. These authorization rules are specified in the Service Configuration data. The finer granularity rules can be coded in particular methods needing more precise policy definition, e.g. job data access only by the owners of the jobs.
- the possibility to make RPC requests from clients to services and passing the results over secure encrypted channels, ensuring data protection while transmission over the public network.
- the transmission of large data files over the network with or without encryption, with GSI compliant authentication and application of defined access rights.

The DISET framework is written in Python language as the whole of DIRAC. It is built on top of the standard OpenSSL library and provids a pyGSI module derived from the Open Source pyOpenSSL module with modifications that allow, in particular, GSI client-service authentication. The secure encrypted channels are provided by the standard OpenSSL software.

The DISET framework makes it possible to the developers of the services to apply specific authorization rules per user or group. It can use standard Grid “full legacy” proxies and VOMS proxies. The user group information is defined independently of the VOMS FQANs and passed as an additional parameter. The eligibility of a given user to perform actions in the context of the specified group is verified in each RPC call. DISET is capable to use also DIRAC proxies that are constructed out of the standard grid user credentials using the OpenSSL tools only. These proxies are completely analogous to the Grid “full legacy” proxies and are used when accessing DIRAC services from non-standard grid platforms such as MAC OS, MS Windows or non-SLC Linux flavors.

5. Job submission to the DIRAC WMS

The job submission to the DIRAC WMS is done by performing several authenticated and authorized DISET RPC calls. First, the job JDL description is uploaded to the Job Manager service. The user proxy is uploaded at the same time as an RPC call string type argument. The data upload is performed over the encrypted channel making it ssl-grade difficult to intercept the proxies on the network. Once uploaded the job JDL parameters are stored in the Job Database and proxies are stored in the Proxy Repository. Only one proxy per user/group combination having active jobs in the DIRAC WMS is stored. Several proxies for the same user can be stored only in the case when there are jobs submitted with distinct user groups.

All the databases are implemented as MySQL databases with no external access via the MySQL protocol allowed. All the databases are protected by passwords that are stored in the local configuration accessible for read access only for the local user account under which the DIRAC services are running.

6. Proxy handling in the DIRAC WMS

Once a user’s workload is successfully submitted to the WMS the accompanying user proxy is analyzed. If it is of the old “full legacy” type, it is converted into a VOMS proxy with the user authorization information (FQAN) embedded corresponding to the user group as specified in the DISET RPC request. The one-to-one

correspondence of the DIRAC user group and the LHCb VOMS roles is defined in the DIRAC Configuration Service and is used to determine the VOMS FQAN for this conversion. Standard gLite command *voms-proxy-init* is used in this case invoked with the user proxy to be converted. The new proxy has the same lifetime as the original proxy.

As long as there are active jobs of a given user in the DIRAC WMS, his/her proxy is maintained in the Proxy Repository database. It may need periodical renewal to keep it in a valid state. To achieve that, the DIRAC WMS host is declared to be a trusted host for the standard LCG MyProxy service where the long proxies for all the LHCb users are stored. This allows renewal of the user proxies if they are close to expiration. Typically, the proxies that are kept inside the Proxy Repository database are relatively short (no more than 12 hours) to minimise the risk of damage in case of unfriendly intrusion to the WMS host. This proxy lifetime can be insufficient for a particular user job which can have larger wall-clock time requirements. In this case an individual longer proxy is generated using the LCG MyProxy server in order to submit this job to the Grid. This proxy is then destroyed immediately after the LCG job submission. It is important to mention that it is impossible to get a new user proxy without having a still valid one. A user proxy can be only extended but not acquired anew.

The user proxies are required by the generic Pilot Job Agent that can execute jobs of any LHCb user. A special WMSAdministrator service can serve user proxies to the requests of Pilot Jobs with a specified lifetime. The Pilot Job Agents are executed with credentials having a specific VOMS role (e.g. Role=Pilot) enabling them to execute arbitrary user workload as described in the first section through the ***glexec*** mechanism. The WMSAdministrator service authorizes the RPC requests with the Pilot Job Agent credentials to retrieve the proxies of any arbitrary LHCb user. It is important to mention that Pilot Job Agents are not allowed to retrieve their own proxies, i.e. proxies with the Role=Pilot. This limits the potential damage in case the Pilot proxy is stolen. Since Pilot Jobs are regular Grid jobs, their proxies can be renewed as necessary by the standard Grid WMS mechanism.

The user proxies retrieved by the Pilot Job Agent can be made of *limited* type to forbid their usage for submission of new jobs to the system on behalf of the user.

7. Pilot Job

The Pilot Jobs are regular Grid jobs that are submitted in the case of LHCb on the EGEE infrastructure by means of the standard gLite Workload Management System. Their purpose is to reserve a CPU time slot on a Grid Worker Node and to retrieve the actual user workload for its execution.

7.1. Submission

The Pilot Job submission is done using user credentials with a special VOMS role that has the capacity to execute the workload of any LHCb user on his/her behalf (hereafter called Pilot Proxy). This role is only used for the operations performed by the Pilot Jobs. A small number of users within the Virtual Organization are enabled to have this special role.

The Pilot Jobs submission is triggered by the presence of user jobs in the DIRAC Task Queue. The Pilot Jobs are submitted with the resources requirements of the workload, including the CPU time limits. The Pilot Proxy is kept in the same Proxy Repository database as other user's proxies. They are renewed also with the same mechanism. If the user workload requirements for CPU time exceeds the life time of the Pilot Proxy, the latter is renewed with a sufficient life time. The validity length of the Pilot Job proxy is included in the Pilot Job Agent request to the DIRAC WMS in order to match the CPU requirements of the user workload.

In DIRAC all the potential grid CEs are described in the Configuration Service. Only CE's that are tested and hence approved for usage by the LHCb managers are added to this list. The jobs can only run on the approved CE's because a list of allowed CE's is always passed to the gLite RBs as part of the pilot job

requirements. In addition, the list can be limited by the DIRAC site mask where certain sites can be forbidden to execute jobs, for example because of some local technical problems. The mask is translated into the CE names and also applied in the pilot job requirements. The same mask is applied when a Pilot Job Agent requests a user job presenting the site name as part of the CE capacity description. This allows to refuse dispatching user jobs to unwanted sites if even the site in question was banned after the Pilot Job was submitted. This reduces the chance that untested, unknown or otherwise unwanted sites can get an LHCb user job.

7.2. User Job retrieval

Once the Pilot Job starts running on the Grid Worker Node, it retrieves the DIRAC software from one of the Grid storages using the Pilot Proxy credentials. The software can be also preinstalled on sites together with the LHCb application software. Once the DIRAC software is installed, the Pilot Job Agent starts running presenting the capacity of the reserved Worker Node to the DIRAC Matcher Service and requesting a user workload. As a result of this request, if the Pilot Job uses the generic Pilot Credentials, it will receive a workload belonging to another user than the Pilot User. This mechanism allows efficient application of the VO job prioritization policy and justifies the need of the Generic Pilot Jobs as discussed in [2].

Many of the user workload operations must be performed using the workload owner's credentials. Therefore the Pilot Job Agent downloads the user proxy from the DIRAC WMSAdministrator service with the lifetime corresponding to the user job requirements. This is done with a secure DISET RPC request over an encrypted SSL connection. Once the proxy is downloaded, it is stored in a local file that is only accessible by the local UNIX user account on which the Pilot Job is running.

Once the job description and the user proxy are available on the site, the Pilot Job Agent generates a Job Wrapper scripts that can be passed for actual execution.

7.3. Application of site policies

Before the actual user workload can start execution, it should be checked against the site policies to verify if the user is eligible to use the hosting site. This is typically necessary if a site is maintaining a list of users banned from working locally even if it is providing access to their Virtual Organization. In this case the application of the site policies can be done by using the system provided ***glexec*** utility that imposes the site authorization rules onto the user identity, possibly denying access to the site resources. In the case of the access denial, the Pilot Job Agent cleans up all the user workload-related data including the user proxy and reschedules the job in the DIRAC Task Queue. It is now ready to pick up a workload of another user if there are any matching the Worker Node capacity. If there are no more user workloads available, the Pilot Job Agent exits immediately, releasing the resource.

If the user is eligible to use the site, the Job Wrapper script is passed to the ***glexec*** utility for execution together with the grid proxy of the owner of the workload. It is up to the site authorities to define the exact behaviour of this tool, either limiting it to just logging the LHCb user activity or changing the UNIX user identity to allow for strict sandboxing and traceability of the job. The DIRAC Pilot Job Agent however logs in a well-defined format the user identity of the owner of the job, the time stamps of the job start and end, etc. The same information is also reported to the central DIRAC WMS *Job Monitoring* service and is available through this service interface to the request of interested clients.

It is important that the proxy of the Pilot Job is not exposed to the process of the user workload which will be started by execution of the Job Wrapper script in the ***glexec*** "logging mode". Otherwise, an LHCb user can potentially obtain the Pilot Proxy which can be considered as a security risk. To prevent that, the Pilot Proxy is made hardly accessible by the Pilot Job process by keeping it in the process memory while removing it from the local disk and restoring it after the control is returned back to the Pilot Job process after the user job execution is over. This mechanism makes it difficult to access the Pilot Proxy from the

user job, although a determined malicious user still can manage to do that. This is, however, not possible if the user job is executed by a different user id from the one of the Pilot Job process which is the case using the ***glexec*** facility in the mode with changing the user identity.

7.4. Multiple user job retrieval

A single Pilot Job can in principle retrieve more than one workload from the same user or from different users if the remaining CPU time is sufficient. In this case, for each subsequent user job, the same procedure as described above is applied.

7.5. Multiple concurrent jobs on the same Worker Node

Some grid sites have their batch systems configured to allow more than one concurrent jobs running on the same Worker Node. If such a site uses the version of the *glexec* utility without changing the UNIX identity of the executed user workload, there is a possibility of making the grid proxy of one LHCb user to be read accessible to the process of another LHCb user. This becomes possible because both jobs are started by the LHCb Pilot Job that runs under the same user credentials and will be typically mapped on the same local UNIX account. This can be regarded as a potential security risk by some Virtual Organizations or sites. In LHCb we do not consider it to be a risk if one of the LHCb users can *intentionally* gain access to the proxy of another LHCb user. If a site considers this to be a security risk, it will have to use the ***glexec*** version with changing the UNIX identity of the process to that of the owner of the workload.

7.6. Termination of the Pilot Job

When the User Jobs are completed and there is no more workload available in the DIRAC Central Task Queue, the Pilot Job terminates. Since this is a standard grid job, the grid middleware software performs all the clean-up operations including destroying of the Pilot Proxy. In case of the Worker Node crashes, it is the responsibility of the site managers to clean-up the possible remnants of the previous jobs while bringing the node back into operation.

8. Running user job

8.1. User job initialization

Once the user workload is permitted to run on the Worker Node, its Job Wrapper script starts execution. It is either invoked by Pilot Job Agent or executed in a process spawned by the ***glexec*** (or similar) utility. A special directory is created to be the current directory of the user job in order to facilitate the complete clean-up of the local disk at the end of the job.

Before starting the user application, it has to perform several operations, using the user credentials. Therefore, the environment is configured to use the job owner proxy: the proxy is stored in a local file and the X509_USER_PROXY environment variable is defined accordingly. These settings can be kept during the whole time of the user workload preparation, execution and finalization phases as necessary. In case of Job Wrapper execution by the ***glexec***, the user proxy is set up as part of the ***glexec*** execution. The operations needing user credentials are the following (the list is not exhaustive):

- Resolution of the input data. This needs access to the LHCb instance of the LFC replica catalog as well as to the SRM interface of the local storage system;
- download of the job Input Sandbox needing secure access to the DIRAC Sandbox service;
- report on the job status and parameters, e.g. CPU and disk space consumption, to the Job Monitoring service;
- access to the Conditions database;
- upload of the results of the job execution. This needs secure access to both DIRAC (Output Sandbox) and LCG (LFC, Grid storage) services.

The user job application software is usually preinstalled on the sites as part of the LHCb standard site verification procedure. However, user application specific libraries containing pluggable algorithms can be shipped in the Input Sandbox of the job. The user can also choose to run his own executable which is shipped in the job Input Sandbox.

8.2. User job execution

The user job application is executed in a process spawned by the Job Wrapper in the current directory created for the job. The Job Wrapper process itself runs in parallel as a watchdog monitoring the application CPU consumption and other parameters. In particular, it can kill the user job if it exceeds considerably the requested execution time.

Since the JobWrapper runs with the user job credentials, a malicious user can in principle, compromise the accounting information sent by the JobWrapper. The DIRAC *Accounting Service* relies on this information for the user activity accounting. A user can modify this information only deliberately and not by some mistake. We do not consider this to be a security problem for LHCb and it is certainly not a problem for the sites that are maintaining their own LHCb activity accounting.

While the user job execution it can turn out that the user proxy is expired before the end of the job. This can happen in case, for example, of a low CPU/WallClock time ratio for I/O intensive jobs. In this case there is a possibility to renew the user proxy by the Pilot Job Agent using the mechanism described earlier in this paper. The renewed proxy is placed in the same location as the initial one. If the Job Wrapper is submitted to the ***glexec*** utility, the Pilot Job Agent can extend the user proxy for the running job using the means of the ***glexec***.

8.3. User job completion

After the user application finishes and all the results are uploaded, the current directory in which the user workload is executed is completely wiped out removing all the job artefacts including security sensitive ones (e.g. the user's proxy). If the user Job Wrapper is executed by the ***glexec*** utility, The Pilot Agent will use the means provided by the ***glexec*** to remove all the user directories and files, including the user proxy, to perform the necessary clean-up.

9. Job monitoring, user interaction with a job

During the execution the user workload there is a watchdog process running in parallel with the user application which is sending information about the progress of the user workload to the central *Job Monitoring* service. In particular, it sends the so called "heart beat" messages at regular time intervals to indicate that the user job is running and to report some numbers about the current resources consumption.

The user workload can also be configured to send more information using this mechanism, e.g. the standard output lines of the user application. The watchdog process in turn can receive instructions from the central WMS service, for example to stop gracefully the user job and release the resource.

In all these communications, the outbound connections are established from the Worker Node to the central WMS services using GSI authentication with the user proxy and sending encrypted data over the network. Therefore, these communications are not creating any security risks.

10. User job accounting

LHCb does not require from the sites a per user accounting of the resources. Only the consumption of the resources for the LHCb VO as a whole is requested. The user level accounting will be performed by the DIRAC Accounting Service that is accumulating reports from all the production and user analysis jobs. If sites are willing to have a per user accounting of the site resources consumption, then this information can be requested from the LHCb Accounting service. In case a site wants to have a reliable independent per user accounting, the ***glexec*** utility in the mode with changing user identity should be used.

11. Pilot Job Questionnaire

0. *Describe in a schematic way all the components of the system. If a component needs to use IPC to talk to another component for any reason, describe what kind of authentication, authorization, integrity and/or privacy mechanisms are in place. If configurable, specify the typical, minimum and maximum protection you can get.*

Figure 1 is doing that. Most of the document describes various operations in the system with security implications.

1. *Describe how user proxies are handled from the moment a user submits a task to the central task queue to the moment that the user task runs on a WN, through any intermediate storage.*

Section 5 describes the user proxy passing while job submission. Section 6 presents the proxy storage and renewal mechanism. Section 7.2 deals with the user proxy passing to a Worker Node.

2. *What happens around the identity change on the WN, e.g. how is each task sandboxed and to what extent?*

See Section 7.3, 8.1

3. *How can running processes be accounted to the correct user?*

See Section 7.3

In addition, all the job execution environment information is collected in the Job Monitoring Service. In particular, it allows to see for each active Pilot Job which User Jobs they are running on which worker node, etc.

4. *How is a task spawned on the WN and how is it destroyed?*

See Section 7.3

5. *How can a site be blocked?*

See Section 7.1

6. *What site security processes are applied to the machine(s) running the WMS? [Here WMS means the VO WMS, not the gLite WMS.]*
Who is allowed access to the machine(s) on which the service(s) run, and how do they obtain access?
How are authorized individuals authenticated on the machine(s)?

What is the process for keeping the service(s) and OS patched and up-to-date, especially with respect to security patches?
Do you have an identified security contact?
Describe the incident response plan to deal with security incidents and reports of unauthorized use?
What services (in general) run on the machine(s) that offer the WMS service?
What processes exist to maintain audit logs (e.g. for use during an incident)?
What monitoring exists on the machine(s) to aid detection of security incidents or unauthorized use?

The DIRAC services are running on machines hosted by the IT department at CERN. The hosts are managed by the IT personnel who install the basic OS software as well as the standard grid middleware. They are also responsible to apply software patches and security updates.

The machines running DIRAC services at CERN are subject to the VO-box agreement [3] between the LHCb Collaboration and the CERN grid site. This agreement defines requirements, procedures and contact information necessary to run LHCb specific services respecting the site security policies and rules.

The machines are protected by the CERN site firewall. Before opening access to the machines in the firewall for the ports used by the DIRAC services, the port scanning procedure is applied to avoid any unwanted access points to the hosts. All the standard server security rules are applied.

The LHCb managers do have root access to these machines with a sudo mechanism (without knowing root password) although this is only required to configure the machines to automatically start the DIRAC services at boot time. CERN AFS tokens are used to authorize individual's access to the machines. The DIRAC software is installed by the LHCb managers in the user space not requiring root access. The services are running also only in the user space, no service is running with the root privileges.

The WMS service is composed of several DISET services as described in the document. No service without GSI authentication is used. All the services are logging their activities, which can be used for incident analysis.

7. *Can you limit the users that can submit jobs to the VO WMS? How?*

The user submitting jobs to the DIRAC WMS are authenticated and LHCb defined authorization rules are applied. The rules can be defined per user, per operation effectively giving full flexibility to limit user access if necessary.

12. References

[1] R.Graciani Diaz, A.Casajus Ramo, DIRAC Framework for Distributed Computing. Proceedings of the CHEP 2007 Conference, Sep 2007, Victoria.

[2] A.Tsaregorodtsev et al, DIRAC: Community Grid Solution. Proceedings of the CHEP 2007 Conference, Sep 2007, Victoria.

[3] LHCb-CERN VO-box agreement document