

Review of Accounting and Usage Monitoring

Final Report

M. A. Pettipher, A. Khan, T.W. Robinson, X. Chan,

July 2007

NB:

This document contains information sensitive to The University of Manchester and should not be further distributed without specific permission from the authors.

Contact details:

M. A. Pettipher
m.pettipher@manchester.ac.uk
+44 (0) 161 275 6063

Copyright (2007) The University of Manchester
Manchester Computing
The University of Manchester
Manchester M13 9PL
United Kingdom

<http://www.manchester.ac.uk/manchestercomputing>

Executive Summary

This is the final report of a three-month project funded by JISC to perform a Review of Accounting and Usage Monitoring. The objectives are to address the following topics:

- An analysis of stakeholders and their requirements;
- The identification of key metrics and a framework within which they could fit;
- An assessment of current tools and how they could work or be re-tasked to work within the framework;
- Recommendations for further work to create an eventual solution.

The stakeholder review has been conducted through visits, teleconferences, email, and via a questionnaire. Feedback has been received from over forty people, from various stakeholder groups including national, regional and campus grid services, EDINA, MIMAS, NeSC, OMII-UK, OMII-Europe, and grid accounting software developers. From these discussions, the major requirements are deemed to be:

- Standards should be used where possible to maximise interoperability. However deficiencies identified by the stakeholders must be addressed both through extensions to the standards and by other means if necessary.
- The solution should be lightweight and not prescriptive, and should not compromise performance.
- Further development is required to address the longer term requirements for data & service accounting.

A wide range of tools has been reviewed. Most of these tools require specific software for deployment, and have been developed in conjunction with a particular community. The deployment of a specific tool may subsequently limit interoperability, although progress in adopting the standards (an objective of OMII-Europe) will assist in this. Similarly, because of the investment in a tool, once deployed there may be reluctance to subsequently consider using an alternative. Thus the reviewers believe it is not appropriate to recommend the adoption of a single tool for the UK communities.

The motivation behind the recommendations is to provide a solution that adopts the standards and assists the services in their deployment, without being prescriptive in the use of specific software. The key recommendations are:

- Standards - JISC should provide support, including funding for the involvement of UK community in the OGF UR and RUS working groups, to ensure that their general needs and key outstanding issues are addressed.
- Accounting Framework - JISC should fund the development of an accounting framework, that will assist deployments based on the standards specifications.
- Client/Usage Monitoring Framework - JISC should fund the development of a client based usage monitoring/management framework to provide a common basis for monitoring related activities.

In addition we recommend:

- A steering group comprising members from different stakeholder groups such as NGS, regional and campus grid, the HPC-SIG and others should be set up to oversee future developments.
- JISC should consider establishing an advisory or support service, which provides a focus of expertise and assistance to those wishing to deploy such software, to encourage greater co-operation.

With respect to the funding of further work on behalf of the UK communities, the reviewers believe that without a co-ordinated, funded approach, there is a strong likelihood that different services will adopt their own solutions, resulting in many different developments and much duplication of effort. Thus there is a strong case to support such developments.

Contents

Review of Accounting and Usage Monitoring.....	1
Executive Summary.....	4
1 Introduction.....	7
1.1 Objectives.....	7
1.2 Terminology, Context and Scope.....	7
2 Stakeholder review and requirement analysis.....	8
2.1 Stakeholder Requirements Summary.....	8
3 Standards.....	10
3.1 Usage Record.....	10
3.2 Resource Usage Service.....	11
3.3 OGF Roadmap.....	11
3.4 Summary Usage Representation.....	11
3.5 Storage Usage Representation.....	11
3.6 Further Details.....	12
3.7 Other Standards.....	12
4 Accounting Tools.....	12
4.1 Overview.....	12
4.2 Usage Metrics.....	13
4.3 APEL.....	14
4.4 DGAS.....	14
4.5 SGAS.....	14
4.6 Gratia.....	15
4.7 NGS RUS Accounting System.....	15
4.8 UNICORE Accounting System.....	15
4.9 SGE ARCO.....	15
4.10 Gold.....	15
4.11 SAFE.....	15
4.12 Functional Comparison.....	16
5 Usage Monitoring.....	16
5.1 Overview.....	16
5.2 EGEE Accounting Portal/APEL.....	17
5.3 Home Location Registry/DGAS.....	17
5.4 Reporting Framework/SGAS.....	17
5.5 Reporting System/Gratia.....	17
5.6 LLview/UNICORE.....	18
5.7 UAS/NGS.....	18
5.8 Web Console/ARCO.....	18
5.9 Gold.....	18
5.10 SAFE.....	18
5.11 CSAR Accounting System.....	18
5.12 Functional Comparison.....	19
6 Discussion and Recommendations.....	19
6.1 Requirements on Standards.....	19
6.2 Requirement for an Accounting Framework.....	21

6.3	Requirement for a Usage Monitoring Framework.....	22
6.4	Recommendations	24
7	Business Case for Suggested Implementation.....	25
7.1	fEC	25
7.2	Common Approach for UK Grid Accounting	26
7.3	Implementation of Recommendations and Cost Estimates	27
8	Risk Analysis.....	28
8.1	Stakeholders do not accept the recommendations	28
8.2	Grid communities do not adopt the proposed frameworks	28
8.3	Insufficient take-up by service providers	29
9	References	29
10	Acknowledgements.....	30
A	Stakeholder Reviews	31
A.1	National Grid Service	31
	Key requirements	31
A.2	GridPP.....	31
A.3	Campus Grids	32
A.4	Regional Grids	33
A.5	Other Compute Services.....	33
A.6	End user	34
A.7	Standards Bodies.....	34
A.8	Data service providers.....	34
B	Standards in Grid Accounting.....	36
B.1	OGF Usage Record Format	36
B.2	OGF Resource Usage Service	38
B.3	EGEE/WLCG Summary Usage Schema.....	40
B.4	EGEE/WLCG Storage Usage Schema	41
C	Accounting Tool Assessment.....	42
C.1	Schema Mapping.....	42
C.2	Custom OGF-UR Extensions	44
C.3	APEL Accounting System	45
C.4	DGAS Accounting System.....	45
C.5	SGAS Accounting System	46
C.6	Gratia Accounting System	47
C.7	NGS RUS Accounting	47
C.8	UNICORE RUS Accounting.....	48
D	Assessment: Usage Monitoring Deployment and Client View.....	49
D.1	EGEE Accounting Portal	49
D.2	HLR	50
D.3	UAS.....	54
D.4	LLview.....	56

1 Introduction

1.1 Objectives

The primary output of this review of accounting and usage monitoring is the present report, which includes the following information:

- An analysis of stakeholders and their requirements;
- The identification of key metrics and a framework within which they could fit;
- An assessment of current tools and how they could work or be re-tasked to work within the framework;
- Recommendations for further work to create an eventual solution;

1.2 Terminology, Context and Scope

This document uses terms based on the “grid accountable map” (figure 1.1) developed from discussions with stakeholders during the review process. The map is intended to illustrate the wide range of activities to which the term grid accounting may be applied.

In particular, it is convenient to distinguish between the concepts of resource accounting and service accounting.

Resource accounting includes ‘physical’ resources such as compute, storage and network, as well as ‘application resources’, which could be an image, dataset, video or journal paper, for example. The accounting of physical resources is well established, although primarily focussed on compute resources. The accounting of application resources is limited at present, but there may well be requirements in the future for such accounting – to assess the type of resources used rather than the quantity.

Service accounting refers to core grid services, such as the use of grid middleware, grid resource management, for which system administrators may require accounting details, and also to application services. The accounting of application services, of which there are many current examples outside the grid, such as data based services from JISC, typically involves the collection of statistics associated with frequency of use, service availability and other characteristics related to quality of service. In this document the term service accounting is generally applied to these applications services.

This review is focussed predominantly on the accounting of physical resources as this is the area of most immediate concern in the grid communities and that most clearly identified in the call, but consideration is given to other areas, in order to assess how a framework for grid accounting can encompass future requirements in related areas.

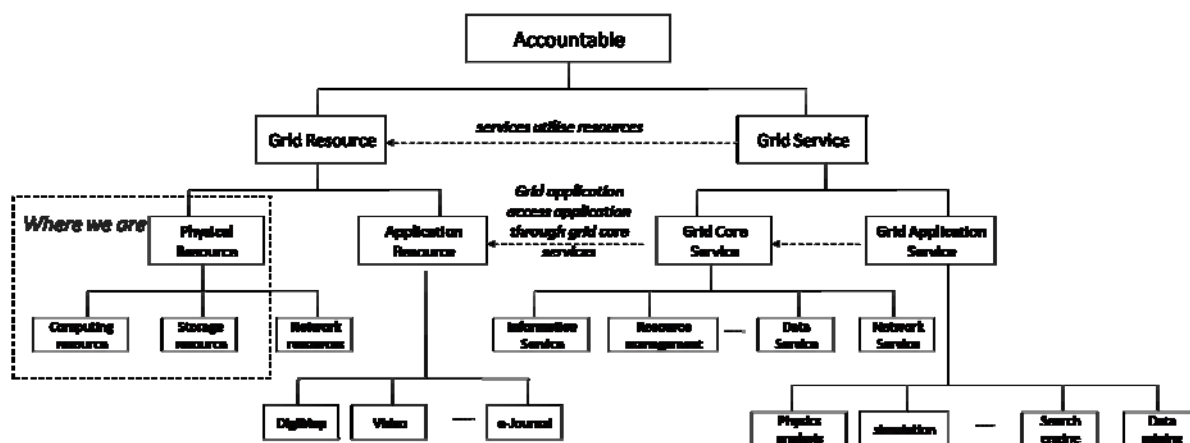


Figure 1.1: Grid Accountable Map

The term Usage Monitoring refers to the processing, reporting and to some extent the management of the usage of the resources (and potentially services). For example, ensuring that the data is reported in a secure manner is crucial for grid based services. Also it is commonly the case that usage monitoring is closely integrated with other activities such as policing resource usage, so some consideration is given to such topics. However, other topics such as economic models are considered beyond the scope of this review, although again some mechanisms for managing aspects of charging are integrated with some of the products mentioned, so this is mentioned when appropriate.

2 Stakeholder review and requirement analysis

A major part of this review has been to identify the requirements of the stakeholders in grid accounting in the UK. This has focussed particularly on those providing grid services, and thus having to perform grid accounting. However it has also included those involved in defining standards and in developing accounting software and tools. The reviewers are very grateful for all the assistance received from numerous people.

The following is a summary of the responses received. Further details categorised into stakeholder groups is provided in Appendix A (section 11.1).

2.1 Stakeholder Requirements Summary

2.1.1 Interoperability

Most stakeholders would consider accounting and usage monitoring best served by a grid service interoperable with other grid services. The consensus is that a standards-based method of sharing accounting data is desirable, but that standardization should be limited to the interfaces, with freedom for site-specific implementations.

2.1.2 Performance

Performance is considered by most stakeholders to be principal requirement. Real time usage monitoring is not considered necessary if it were to incur too heavy a load on back end compute nodes or the network infrastructure.

2.1.3 Accounting Metrics

For those stakeholders who have considered the OGF-UR standard[1], it is generally agreed that the format is extremely useful, at the least as an exchange format. Some shortcomings of the current UR standard have been noted, including ambiguity in field definitions, and the lack of specific fields necessary for grid interoperability such as “*SiteName*” and “*VOName*”. Some site-specific requirements, such as full command line details, could be covered by the UR extension framework; however it is considered by many that the extension framework, designed to provide customization and flexibility, does so at the expense of interoperability.

In terms of accountable resources, compute usage is considered to be of highest importance, and it is also the most well-defined in terms of metrics, at least for traditional batch-job-like environments. Persistent storage and data access accounting are less well defined and are of immediate importance, while network usage and memory usage are not considered of immediate concern. Service level accounting will need to be addressed in the future as grid usage embraces the concept of workflow, where accounting for data access, data movement, and computation should be considered in a coherent and consistent manner. Specific issues relating to each resource accountable have been raised by the stakeholders, and these will be considered in turn.

- Compute usage: In the situation where jobs are given exclusive rights to processors or nodes, compute usage is generally accounted for in terms of wall duration, although raw CPU time is often recorded to assess performance of the code. This is the case for most batch system-like grid environments, but in the case of GridPP[2], and some shared memory resources, the CPU duration

itself is the metric of interest. If one is to consider the actual contribution of resources to and usage of resources within a computational grid then duration (CPU or wall) is not necessarily the best metric, as CPU performance varies from one resource to another, and even within a single resource in the case of a heterogeneous cluster. In situations where there is a need to quantify the real amount of resource a site or VO is providing compared to its consumption, relative CPU performance is clearly of significant importance. Some stakeholders such as GridPP and HPCx[3] address this issue by scaling CPU usage according to standard or non-standard benchmarks (for example specint2K performance) however there is no one benchmark that is universally applicable, as relative performance between two processors is application specific and overall performance is governed by many factors such as the quality of network and interconnect, compilers and run-time environment. To allow for sharing of usage records across grids, we advise that normalization or scaling should only ever be performed after generation of the UR, as any future changes in normalization policies or scaling factors would be difficult to implement without the raw CPU usage data being available. If normalization is implemented, which might be required for charging or resource brokering, for example, then that process should be independent of the resource usage records.

- Persistent storage: is not a high priority for all stakeholders, for example some campus grid infrastructures do not provide any centralized storage facility, and temporary disk is automatically cleaned after job termination. For other stakeholders there is an urgent need for a standard means to implement persistent storage based accounting, as there is a general lack of client-side tools available. It is generally considered storage accounting in the present usage record format is not well defined and insufficient for the needs of the community, and we endorse the development of a storage usage record format.
- Database and file accounting: Stakeholders involved in the provision of data services need to determine whether they are providing value for money. It is generally agreed that there is currently a gap in understanding of how to account for database and file access. The types of metrics required by stakeholders range from simple statistics like the number of database transactions or queries and number of individuals accessing data, right through to information on the actual nature of the data that is being accessed. A significant amount of work is needed to develop a generally applicable solution for data accounting, ideally one that provides a consistent approach to accounting of SRB[4] usage, database usage, and OGSA-DAI [5] usage.
- Memory and Network usage: There is little need for accounting memory usage as sites typically simply allow jobs to use all available memory. Network usage is not generally being reported at the job level, but rather the overall network performance is monitored site- or grid-wide in the same manner that CPU utilization is monitored for capacity planning and justification for service upgrades.
- Service accounting: The general consensus of the stakeholders is that accounting requirements for services cannot be covered by standard usage record formats as currently specified. The metrics of interest are more akin to those corresponding to web server activity: hits/sessions/unique visitors. The question of what a "service instance" means needs to be defined and it should be considered as a long-term goal to develop a service accounting framework. (Further discussion about this topic is given in section 11.1.8 in Appendix A.)
- Custom charge metrics: Stakeholders stressed the importance of custom charge measures to provide for quality of service such as guaranteed processor reservation.

2.1.4 Allocation management

Most stakeholders require methods for allocating resources and automated policing of these allocations. Ideally, there should be the ability to devolve the management of resource allocations to individual project managers or VO managers. Most sites are using home built software or facilities within a specific batch system such as SGE[6] for allocation management.

2.1.5 Fault tolerance

Stakeholders stressed the need for tools to guarantee the sanity of accounting data and the reliability of transfer of usage records from local resources to local or centralized storage repositories. The accounting system must be resilient to the failure of any one of its components and also resilient to communication failures.

2.1.6 Security and data protection

The accounting system must protect accounting data by requiring that publishing and extraction is done via secure communication channels. If a site or grid has a specific requirement for anonymity of accounting data then there should be sufficient flexibility to ensure that the site is able to share depersonalized data.

3 Standards

There are two Open Grid Forum working groups directly relating to grid accounting, the OGF Usage Record working group and the Resource Usage Service (RUS)[7] working group. The current UR-WG proposed specification is Usage Record 1.0, published as GFD.98 Usage Record – Format Recommendation. The Resource Usage Service specification has not yet been published as a proposed recommendation, although stable drafts have undergone public comment. In this section we make reference to draft specifications 1.7 and 1.9. Other standardization efforts are being put into the development of a summary usage record schema (summarised resource usage was outside of the scope of the Usage Record 1.0 specification) and into the development of a storage usage schema. In this section, we introduce and review the OGF draft specifications as well as other relevant standards. Further details of the draft specifications are listed in Appendix B.

3.1 Usage Record

When resources are shared, sites must be able to exchange accounting data in a common format to provide a grid-wide view of resource usage. A definition of common usage representation should address the following factors:

- Target Stakeholder

According to the accountable map (figure 1.1), there are two main accounting categories, resource accounting and service accounting, targeted at the resource provider and service provider, respectively. The resource provider is interested in how shared resources are being utilized in a quantitative sense (i.e. how many CPU cycles and how much disk space is consumed), while service providers might be more interested in how the service/application resources are being used (i.e. access to application data or files).

- Representation Granularity

A usage record could report accounted data at various granularities, from a fine granularity such as job level, to more coarse granularities corresponding to user, site, VO or Grid, depending on requirements.

- Usage Metrics/Properties

A usage record is composed of a list of usage metrics, or usage properties. These metrics should be defined using unambiguous semantics. The collection of usage metrics should be general enough

to accommodate common usage scenarios as well as providing flexibility to allow for custom metric extensions.

- Format

The usage metrics should be organised in a well-defined structure with a consistent typing system and syntax format.

The OGF-UR working group defines an XML-formatted representation for resource usage on a per-job, or “atomic” basis: the Usage Record schema. Version 1.0 of the UR specification has been published and has the status of a Proposed Recommendation. The usage metrics defined in the OGF-UR encapsulate common CPU, storage, network and service usage properties as well as extensions for definitions of custom resource usage metrics (see Appendix B for more details of the schema). Although a service property is present in the OGF-UR schema it is clear from stakeholder analysis that the schema as it stands is insufficient to provide for service-level accounting.

3.2 Resource Usage Service

The Resource Usage Service (RUS) defined by the OGF RUS working group specifies a set of normative Service Interface Definitions (SIDs) facilitating common interfaces for high-level treatment of usage records in OGF-UR format. These SIDs are divided into two classes: mandatory core features and optional advanced features. There are currently two working group draft RUS core specifications: the current draft specification (version 1.7) and a further proposed specification (version 1.9). In version 1.7, a RUS usage record schema is defined based on the OGF-UR schema with additional RUS auditing information included on a per usage record basis. The RUS version 1.9 proposes to use the OGF-UR schema for usage record representation while maintaining RUS-specific information separately.

3.3 OGF Roadmap

- OGF-UR: version 1.0 is a final specification for the job usage record representation. OGF-UR 2.0 is under development and discussion.
- OGF-RUS: WS-I versions[8] of the core and advanced specifications are targeted for the end of summer 2007; WSRF versions for end 2007.

3.4 Summary Usage Representation

The purpose of a summary accounting schema is to provide high-level accounting information to potential accounting users while shielding job-specific information, a requirement that might, for instance, arise as a consequence of data protection rights. The development of a summary usage record format is currently a recognised work item within the OGF UR working group.

A typical usage scenario for summary usage accounting is the EGEE/WLCG accounting system, which allows for usage information to be represented in aggregated (summary) format where multiple jobs are grouped and fields summed according to certain criteria. The aggregate schema of the EGEE/WLCG[9][10] project describes summary usage information on a per VO, per site basis, i.e. the total (summed) usage information of jobs belonging to a particular VO running on a particular site resource during a specified period of time. The detailed schema being used in WLCG project can be found in Appendix B (table B-8).

3.5 Storage Usage Representation

There is an increasing requirement for storage usage accounting, generally in terms of persistent storage. The OGF-UR format as it currently stands is limited with respect to storage accounting. Metrics supported for disk usage (Disk element) are total, min, max or average, and the type of disk can be reported as, for example, temp or scratch, but there is no identifier for files. Therefore, storage usage accounting requires significant further research and development efforts beyond the current OGF-UR standards. A schema for summary storage accounting has been developed by the EGEE/WLCG project (see table B-9).

3.6 Further Details

Further details of these standards are given in Appendix B of this document.

3.7 Other Standards

There are other accounting standards that provide schemas for service/application-specific resource usage representations. The metrics defined in those standards are targeted at service providers who would like to account the usage of application resources, such as “media files” and “digital maps”, for auditing and/or charging purposes. Therefore, the resource metrics or properties are various, and application dependent. The SUSHI/NISO³ standard[11], for example, defines a normative usage statistics schema and a standard message format for communication to the COUNTER[12] application as a basis for counting and reporting the access of electronic library resources. Recommendations regarding standards for service level accounting are outside the scope of this report.

4 Accounting Tools

In this section, we introduce existing accounting tools for the metering of resource usage and the storage of generated usage records in standard or custom defined schemas. Further details of the underlying accounting mechanisms and design plans are given in Appendix III.

4.1 Overview

The review content of this chapter is based on a list of accounting tools, some of which are targeted at global grid projects while others are designed for specific batch systems.

Table 4-1: Current Grid/Cluster Accounting Tools

Name	Project	Description	Affiliation
APEL	EGEE/WLCG	APEL (Account Processor for Event Logs) is an accounting tool used in the LCG project, and is a part of the gLite middleware.	STFC RAL
DGAS	EGEE	DGAS (Distributed Grid Accounting System) previously known as the DataGrid accounting system was developed within the EU Datagrid project and is currently being re-engineering by EGEE and OMII-Europe.	Istituto Nazionale di Fisica Nucleare (INFN)
SGAS	NorduGrid	SGAS (SweGrid Accounting System), developed for SweGrid, is a Java implementation based on OGSA architecture that is now integrated as a Grid service in Globus Toolkit 4. SGAS has been used in NorduGrid as a standard accounting service.	The Royal Institute of Technology - KTH
UNICORE Accounting Service	UNICORE	The UNICORE accounting system is an OMII-Europe component that provides a WS-RF compatible RUS implementation for real-time usage monitoring.	Forschungszentrum Juelich - FZJ
Gratia	OSG	Gratia is the grid accounting system being developed for Open Science Grid. A project is currently underway to develop a RUS	SLAC and FNAL

³ The standardized Usage Statistics Harvesting Initiative Protocol/National Information Standards Organisation

		compatible implementation.	
Gold	SSS	GOLD is an open source accounting and allocation management system developed at PNNL as part of the DOE Scalable Systems Software (SSS) project and now provided by Cluster Resources Inc. It is based on the QBank resource allocation manager.	Cluster Resources
SAFE	HPCx	EPCC's service administration software, SAFE, provides accounting information for the UK's national HPC services HPCx and HECToR as well as for local EPCC machines.	EPCC
NGS-RUS Accounting	NGS	The NGS accounting system is based on the parsing of batch logs and upload of the generated usage records to RUS via the RUS interfaces.	The University of Manchester
ARCO	SGE	The Accounting and Reporting Console (ARCO) is a module specific to the N1 Grid Engine from Sun Microsystems, which provides collection and analysis of grid usage statistics. The usage of shared resources can be tracked on multiple levels, such as per user or project.	Sun Microsystems

4.2 Usage Metrics

The usage representations can be divided roughly into two categories according to compliance or non-compliance to the OGF-UR standard. Some of tool candidates, such as APEL, DGAS, and NGS RUS are OGF-UR *compatible*, in the sense that a clear mapping exists between the OGF-UR standard data fields (XML element tags) and the local schema fields; however for the purposes of this discussion classification as OGF-UR compliant requires that:

- data is represented in XML format and
- is validated by OGF-UR schema;

Other accounting tools, such as Gold (Cluster Resources), SAFE (EPCC), and ARCO (SGE), have custom representations of usage metrics. For some of these tools we establish mappings to OGF-UR metrics where possible.

4.2.1 OGF-UR Non-compliant Usage Representation

Among the accounting tool candidates, APEL and DGAS share a common usage schema (the LCG schema) for the storage of usage records in relational databases. The NGS-RUS accounting system is an implementation of a WS-I RUS service in which usage records are stored in an Oracle database and queried via SQL statements, while DGAS is also developing a RUS implementation (DGAS-RUS), as an equivalent service to its Home Location Registry (HLR) component to store and expose the usage records in OGF-UR format. Currently, these Grid accounting systems all store usage records in relational databases, with customized schema definitions. The usage representations are, however, OGF-UR compatible in that their schemas have clearly defined mappings to the metrics define in OGF-UR as can be seen in table C-1 in the appendices.

Some cluster/batch systems have accounting systems targeted primarily at compute clusters, and these tools tend to fall into the OGF-UR incompatible category. Potential mappings to OGF-UR metrics are summarised in table C-2 in appendix C.

4.2.2 Metric Extensions to OGF-UR

The usage metrics and representations of SGAS, Gratia and the UNICORE accounting service use the syntax and semantics defined in the OGF-UR specification and are thus designated as OGF-UR compliant. It is clear from discussions with stakeholder groups that nearly every service provider has specific accounting requirements beyond the scope of base or differentiated properties defined in OGF-UR. Common extensions are listed in table C-3 in appendix C.

4.3 APEL

Accounting Processor for Event Logs (APEL)[13] is an accounting tool deployed within the WLCG and EGEE projects. APEL comprises a set of log parsers running as “cron” jobs that parse batch system and gatekeeper logs generated by a site. Job based resource usage metrics such as CPU time, Wall Clock time, Memory and grid user DN, are formed into a single usage record in the WLCG accounting schema (see section 4.2.1). These usage records parsed at sites are published into the R-GMA (Relational Grid Monitoring Architecture)[14] and archived in a centralized relational database at the Grid Operations Centre at RAL for processing by a graphical front-end utilized by the accounting web tool (see 5.2).

The accounting system used by EGEE/WLCG is in the process of being re-engineered to provide an additional RUS layer between the web portal and GOC database. The idea behind WLCG-RUS[15] is to allow usage information produced by other grid projects (i.e. Open Science Grid) to be shared to GOC through standard service interfaces, and to offer enhanced security to provide users with access to user-level accounting information via the web tool. The WLCG-RUS project will facilitate job level and aggregate/summary usage accounting by mapping the LCG schema to the OGF standard usage schema.

4.4 DGAS

The Distributed Grid Accounting System (DGAS)[16], previously known as the DataGrid Accounting System, is another grid accounting tool deployed within the EGEE project. DGAS is targeted at providing job-level resource usage metering in a client/server infrastructure. Although out of the immediate scope of the present study, it should be noted that DGAS also introduces a pricing authority that can integrate with workload management to facilitate economic model-based resource allocation.

The DGAS accounting functionality is realised by two main components: the lightweight usage sensor, Giandua, and the distributed Home Location Registry (HLR), which acts as a repository for usage information related to registered users or resources. Usage information is collected from usage sensors deployed at each site and published to the HLRs, from which usage information can be retrieved for both individual jobs and in aggregate/summary form. DGAS-APEL integration enables usage data produced by DGAS to be published to the GOC centre.

A DGAS-RUS module is being developed for the DGAS system that will co-exist with the HLR component. The RUS interface will enable “insertion” of usage records metered at sites and “extraction” for user queries. Each HLR will have a corresponding RUS instance so that the resource provider can contribute usage data either through HRL legacy interfaces or through standard RUS interfaces. In this context, there will be multiple RUS instances for job and aggregate usage queries. Synchronisation will be implemented to guarantee consistency between the HLR and RUS backends.

4.5 SGAS

The SweGrid Accounting System (SGAS)[17] is an accounting system designed for SweGrid,[18] the national Grid test-bed in Sweden, and has been integrated with Globus Toolkit 4 as a standard grid accounting service. Within SGAS, usage metering is realised through the Job Account Resource Management (JARM) component, which is responsible for providing the accounting system with information from the local batch systems. Each user requires a valid account with credits in a banking service. When submitting a job, the JARM computes a maximum cost and reserves that amount of credit on the user’s account through the banking service. On completion, the JARM reports the actual resource consumption in the form of a usage record and the associated charge is made to the user’s account. The usage record is then populated into the

RUS instance, the Logging and Usage Tracing Service, for centralised storage. Any query on job usage information is directly sent to LUTS via an authorisation service that protects usage data from invalid access. .

4.6 Gratia

Gratia[19] is the grid accounting system being developed for Open Science Grid (OSG)[20]. The Gratia accounting system is composed of four functional components: the probe, collector, accountant and publisher. Usage information from worker nodes is generated with a set of probes and stored at the local file system. A collection process embedded in the collector component is scheduled to run daily and pull the usage information from each registered node/probe (one probe for each node). These usage records are stored centrally in the file system of a collector machine. Within the collector machine, two configuration files are also initialised for CPU information and User information respectively to provide user-usage and resource-usage mappings.

4.7 NGS RUS Accounting System

The RUS accounting system deployed within NGS[21] project was originally designed for the Market for Computational Services (MCS) project and has been re-engineered for metering and collection of usage information from computing centres around the UK. Usage metering is performed by two modules that parse globus logs and PBS/LSF batch logs. The generated usage records are streamed into a centralised RUS instance with an Oracle database backend via the RUS upload interface. Querying is done via SQL statements rather than the RUS querying interfaces.

4.8 UNICORE Accounting System

The accounting system in the UNICORE[22] project provides a Web Service Resource Framework (WS-RF)[23] compatible implementation of OGF-RUS. The RUS service in UNICORE is intended to expose usage records generated at the batch system level in real time. A batch system-specific usage record generator is invoked whenever a usage query is submitted through RUS. The RUS implementation uses an Apache Derby database backend to provide sufficient performance, but usage records are returned to the client after conversion to OGF-UR format. The current RUS implementation does not provide for persistent storage of usage records.

4.9 SGE ARCO

The Accounting and Reporting Console (ARCO) is the accounting tool for Sun Grid Engine (SGE) provided by Sun Microsystems. The ARCO tool is designed to provide usage statistics for cluster resource usage and auditing. There are two main components defined within ARCO, a data-aggregation daemon and a web-based tool. The data-aggregation daemon, DBwriter, is the component that ingests and collects job usage information, which is then stored centrally in a relational database. The system provides job-level usage information, as well as aggregation over users, departments, or projects.

4.10 Gold

The Gold Allocation Manager[24] is an open source accounting system designed to interact dynamically with the local resource management system: to provide job quotations at submission time, holds on accounts at job start time, and charging at the point of job completion. Gold can also function in a non-dynamical way through, for example, the regular parsing of batch system logs. Gold consists of a client tools and a daemon that interact via the SSSRMAP protocol. Various interfaces are available including command-line interfaces, Perl and Java APIs and communication protocols.

4.11 SAFE

EPCC's service administration software, SAFE, provides accounting information for the national HPC services HPCx and HECToR as well as local EPCC machines. Accounting information is provided both through the parsing of batch-system logs and from process-level system accounting, typically posted to the SAFE via a

servlet once a day. Resource usage is compared to allocations at job submission. At the server side, accounting data is stored in a MySQL database but accounting records are manipulated as Java classes. The system can support multiple SQL accounting schemas provided each schema has a supporting class derived from the AccountingRecord superclass. A RUS instance is in the process of being deployed.

4.12 Functional Comparison

Table 4.2: Functional Comparison

System Name	Schema Type	Storage Type	Meter Type	Publish Protocol
APEL	SQL	Relational database	After events	R-GMA
DGAS	SQL	Relational database	Real time	SOAP/R-GMA
Gratia	XML	File System	After events	SOAP
SGAS	XML	XML:DB (eXist)	Realtime	SOAP
NGS RUS Accounting System	SQL	Relational database	After events	SOAP
UNICORE RUS Accounting System	XML	Relational (Apache Derby)	Realtime	SOAP
Gold	SQL	Relational database	Realtime	SSSRMAP Wire Protocol
SAFE	SQL	Relational database	After events	REST

5 Usage Monitoring

5.1 Overview

In this section, we briefly describe the usage monitoring capabilities of the accounting tools introduced previously. In some cases the usage monitoring tool interrogates the usage records produced by the underlying accounting tools, while in other cases the monitoring tool connects to the accounting tool or service directly. Further details regarding functionality and client-side views are presented in Appendix D.

Table 5-1: Usage Monitoring Component List

Usage Monitoring Tool	Associated Accounting Tool	Project Name	Description
EGEE Accounting Portal	APEL	EGEE/WLCG	The Web portal acts as a standalone Web application that provides graphical usage monitoring interfaces on APEL usage records.
HLR	DGAS	INFN	The Home Location Registry of DGAS provides usage monitoring facilities for job and aggregate usage tracing.
LLview	UNICORE	UNICORE	LLview is a usage monitoring tool based on a UR/RUS implementation designed to provide real-time usage monitoring for running jobs on UNICORE middleware.
Reporting Framework	SGAS	SweGrid	The Logging and Usage Tracking Service (LUTS) exposes client-side web service APIs on a RUS service implementation of SGAS accounting tools.

Reporter	Gratia	OSG	The Gratia accounting system reporter provides a graphical user interface for usage monitoring.
UAS	NGS RUS Accounting	NGS	Usage Accounting System provides user-level usage monitoring facilities via a Web portal and also as usage policing of allocated resource quotas.
Web Console	ARCO	SGE	The Accounting Reporting Console is a module available with the N1 Grid Engine product from Sun Microsystems.
Web based GUI	Gold	SSS	The web gui provides remote access for users, managers and administrators via the Perl interface to Gold
SAFE	SAFE	EPCC/HPCx	A java based web interface to SAFE provides graphical usage monitoring and allows Principal Investigators to administer their projects' users and resources.

5.2 EGEE Accounting Portal/APEL

The EGEE accounting portal is targeted at reporting resource usage information of jobs running within the EGEE/WLCG project. The portal is deployed as a standalone Web application managed by the Grid Operations Centre at Rutherford Appleton Laboratory. The portal presents summary usage information, aggregated from job usage information published from APEL and APEL-DGAS integration as well as summary usage information from OSG and NorduGrid. The Portal presents usage data rather than directly communicating with any underlying accounting system modules. The portal currently reports summary CPU usage information on per site, per VO and per region basis. The portal is designed to provide client-side access control over five roles: user, VO member, VO manager, site manager and GOC developer.

5.3 Home Location Registry/DGAS

The Home Location Registry (HLR), a component of DGAS, enables distributed storage of usage records and provides secure usage monitoring interfaces for end users. A HLR server receives and keeps accounting information for registered resources and registered users. Implementation of the HLR module is based on client/server infrastructure. Two server-side modules enable user queries for interrogation of usage records stored within the HLR server: *advancedQueryEngine* and *uiEngine* for flexible SQL queries and queries with predefined SQL parameters.

5.4 Reporting Framework/SGAS

The Logging and Usage Tracing Service (LUTS) in the SGAS accounting system provides an OGF-RUS implementation that allows population and retrieval of OGF URs. The design of LUTS exhibits a WSRF compatible implementation of OGF-RUS specification. An API is defined to allow the querying of usage records through the LUTS service. The SGAS system also provides a reporting framework that is used to construct periodic usage reports from LUTS and other services. These reports can be published to external sources such as Web servers.

5.5 Reporting System/Gratia

The reporting system of Gratia provides a Web interface for a variety of stakeholders to import and export accounting data in a secure manner. The reporting system is implemented as a standalone Web portal that interrogates the accounting data collected from probes and centrally collected within the Gratia database. The reporting system is configurable to specify certain query logics for various stakeholders. These configuration query logics are being used by a third-party component, called BIRT, for presenting the results of queries in graphical form.

5.6 LLview/UNICORE

The LLview[25] monitoring application is a well used tool in the areas of system management and resource reservation. LLview is able to run in various modes, for machine monitoring (when running on the same machine being monitored); for accessing data from other resources like Web server; and for persistent local storage of usage data. Within the UNICORE project, LLview is deployed and configured to talk to a RUS service through a RUS client module, which enables the LLview application to capture up-to-date usage information from underlying batch systems and report usage statistics in near real time.

5.7 UAS/NGS

The User Accounting System (UAS) developed by the NGS at Manchester and RAL is used for reporting and policing usage statistics on per user basis. There are two main data repositories at RAL: a job level usage database, which is synchronised to the RUS database at Manchester, and a summary usage database for presentation of accounting statistics at a per user level. The summary usage statistics are updated daily by via cron jobs and presented to the users in static graphical forms by comparison to resource allocations, with limited user-level interaction and no querying permitted. Usage policing is a backend process, run daily, that compares the actual resource usage and allocation quotas for each user. The policing process suspends the user account if usage exceeds the allocated quota by removing the user's DN from the gridmapfiles.

5.8 Web Console/ARCO

The Web Console is a module of ARCO accounting system designed exclusively for SGE. The Web console is configurable to report usage statistics on per job, per project, per user and other user-defined bases.

5.9 Gold

Gold provides a variety of means of interaction including command-line interfaces, APIs and a graphical user interface written in Perl. The full Gold functionality can be accessed via the Perl API. The GUI provides balance and usage information to users, managers and system administrators, and role-based authentication determines which actions can be performed on each object.

5.10 SAFE

SAFE's functionality includes user registration and management, Principal Investigator-level project administration, including resource suballocation, report generation and helpdesk facilities. The java-based web interface to SAFE is a single point of entry to view resource usage statistics (job-level and aggregated) and manage projects and users. There are multiple levels of view – service, project, and user – with authentication implemented via access control lists.

5.11 CSAR Accounting System

The CSAR[26] HPC service started in 1998 and introduced a sophisticated allocation management features beyond those used in many other production services today. Resources were priced in terms of generic charge units (tokens), which enabled project managers to exchange one resource type for another, from a diverse pool of resources including a range of compute resources, storage devices of various performances, and even user training and support. Service administrators allocated resources at a project level, according to the awards from the research councils, and project managers could sub-allocate resources to their users within their overall allocation. All activities were managed through an in-house Perl-based web interface.

5.12 Functional Comparison

Table 6-2: Usage Monitoring Functional Comparison

Usage Monitoring Tool	Main Usage	Reporting Level	Access Control	Target Contact Point
EGEE Accounting Portal	Usage reporting	Summary	No (as yet)	Usage database (summary)
HLR	Usage reporting	Job/Summary	Yes	HLR module
LLview	Real-time usage monitoring	Job	Yes	UNICORE RUS module
SGAS Reporting Framework	Usage reporting	Job/Summary	Yes	LUTS module
Gratia Reporter	Usage reporting	Job/Summary	Yes	Gratia database
UAS	Usage reporting and Policing	Summary	Yes	Usage database (summary)
ARCO	Usage reporting	Job/Summary	Yes	Usage database (summary)
Gold GUI	User and project management, usage reporting and policing	Job/Summary	Yes	Usage database (Job)
SAFE GUI	User and project management, usage reporting and policing	Job/Summary	Yes	Usage database (Job)
CSAR GUI	User and project management, usage reporting and policing	Summary	Yes	Usage database (summary)

6 Discussion and Recommendations

In order to provide an integrated and widely adopted approach to grid accounting and usage monitoring in real production Grids, there must be continued efforts both on standards and the implementation, and also continued research and development activities in these areas. At present, there is significant diversity in the solutions adopted by different service providers, many adopting standards where they can, but also taking a pragmatic approach to provide the best solution for their own requirements. Once effort has been devoted to produce a customised solution, there will be reluctance to re-engineer even if new developments provide an improved framework, unless there is a clear benefit in doing so. Thus it is of great importance to prioritise and modularise the tasks to be performed, so that progress can be achieved incrementally, service providers can decide when it is appropriate to do their own customisation, and there is the flexibility to exploit new developments as they arise.

6.1 Requirements on Standards

The development of grid accounting systems should be conformable to well-defined standards, OGF-UR and OGF-RUS in particular, to enable data and system interoperability. However, most of the accounting tools reviewed are not yet fully compatible with these (draft) specifications, indicating additional requirements on those two standards. It is worth noting that OMII-Europe are also adopting these standards - the goal of the Accounting Task within OMII-Europe is to provide a RUS interface to the accounting systems of the three Grid middleware stacks gLite, Globus toolkit and UNICORE to allow the exchange of accounting information. This process is progressing well and much of the required functionality is already available in UNICORE.

6.1.1 Standards – Common usage properties

The most common extensions to the OGF-UR schema are the virtual organization (VO) property and the grid-wide property, site name. Extensions can undermine interoperability – for example the VO property has been named “VOName” in Gratia and “LCGUserVO” in APEL. Further, we believe the OGF-UR is too batch system specific, lacking certain grid-wide properties. Although it is possible to define these grid-wide properties from OGF-UR extension framework, the semantics of the grid-wide properties are quite different. Thus the “resource” property has been used to define customized properties outside the scope of base resource properties, CPU, disc, memory etc. An alternative mechanism is needed to define other grid-wide properties.

6.1.2 Standards - Resource usage properties

Normalization is another property suggested as a standard attribute for CPU usage. The CPU performance varies from one resource to another, and even within a single resource in the case of heterogeneous clusters. In situations where there is a need to quantify the amount of resource a site is providing compared to the amount of resource it is consuming the relative CPU performance is clearly of significant importance. However, it is difficult to decide on what are reasonable scaling factors in comparing resources because relative performance is application specific (specint or linpack benchmarks may be useful) and is also governed by factors such as the quality of network and/or interconnect, and compilers etc. We advise that normalization should only ever be performed after generation of the UR, as any future changes in normalization policies or scaling factors would be difficult to implement/qualify without the raw CPU usage data being available.

There is an increasing demand for storage accounting which is currently out of scope of the OGF-UR. It is difficult to use the current OGF-UR as an integrated usage schema for both CPU and storage usage representation - there is only one “disk” resource property defined in OGF-UR, and this is not enough to accommodate the other storage accounting requirements, like tape. Further, additional usage properties are required to represent the storage usage record in more meaningful manner. These might include storage architecture name (OGSA-DAI, SRB and SRM), storage element name used in EGEE and other relevant properties.

Network accounting is also of concern (although with a lower priority than storage accounting) for grid accounting. It is not yet clear whether the network resource should be considered as a standalone accountable. However there is no question of the importance of network accounting for service-level accounting on per data access basis. Some data services have application-specific data stored within distributed databases accessible through the OGSA-DAI service. A particular service invocation might return a subset of application data based on a user query. The transferred data bytes then become one of the important service usage properties for service invocation can be charging.

6.1.3 Standards – Service Usage Properties

Service-level accounting is a relatively new topic for grid accounting and requires further research effort. There are two aspects of service-level accounting that differentiate it from resource usage accounting. Service-level accounting aims to provide the service provider with high-level usage information about application services and the service usage properties are more service-specific (although there are certain common requirements). Similar to resource usage accounting, service-level accounting requires usage information to be logged on a per invocation or per session basis, and it is essential to identify the user. In this perspective, a globally unique session identity is useful for identifying a particular service usage. In addition, it should be possible to define common usage properties for a particular category of services (as in the SUSHI specification mentioned in section 3.7).

6.1.4 Standards – Summary Usage Format

Another common requirement for usage representation is a summary usage record format applied to both resource and service usage representations. From the resource usage accounting perspective, a summary

usage record groups a set of job usage records according to certain grouping criteria – such as per user, per virtual organisation, per site, per host or combination of multiple groups. Accordingly, summary usage records can be grouped with multiple service sessions. Thus the definition of the summary usage schema is required to be flexible enough to accommodate various grouping criteria. The summary usage record may provide a more meaningful representation for usage statistics compared to atomic level usage records.

6.1.5 Standards – Resource Usage Service (RUS)

The OGF-RUS specification is defined to provide service-interoperability among accounting tools developed independently. It is extremely important for multi-Grid accounting purposes. For example, In the WLCG project, grid jobs can be submitted to resources across three grid infrastructures and the resource usage information is logged by grid-specific accounting tools. It is relatively easy to share this usage information via a common service interface. The OGF-RUS specification is a sound candidate but requires further development partly to address deficiencies identified by stakeholders.

There are two major concerns – performance, and the return of results from a RUS query. The current OGF-RUS mechanism of bundling all matches into a single XML document is unusable for the volumes of usage data collected on operational grids. This is being addressed in the forthcoming RUS specification, which allows subsets of the full usage record to be returned, thus reducing the amount of data returned. However it will not resolve the problem in extreme cases. A possible solution could involve streaming or chunking of data. It is possible currently to circumvent this problem by non-standard means (using SQL), but querying via the RUS interfaces must be workable when sites adopt hierarchical or P2P implementations of RUS.

In some grid projects, grid-specific security policies prevent the sharing of usage information at the job level. The summary usage record could be useful as a method of exchanging usage information in these circumstances. In addition, and in tandem with the corresponding OGF-UR developments, the OGF-RUS will be required to provide constructive solutions for operations on summary usage records.

6.2 Requirement for an Accounting Framework

Based on the reviews of existing accounting tools, both standards compatible and incompatible, the tools share common requirements while being implemented in various manners with dependencies on custom deployment environments. We describe here a proposed accounting framework that abstracts the basic functionalities an accounting tool should provide based on client/server infrastructure with the usage data provider at the client side and the RUS service at the server side. The advantages to the UK communities of developing such a framework are:

- Access to Usage Record generators for any desired batch service.
 - Currently it is necessary for services to produce their own generators, or acquire, and possibly adapt from another service, which may have different requirements.
- Access to a RUS implementation, which provides basic functionality, initially implementing simple access control list, but later incorporating more sophisticated security systems such as VOMS, configurable for specific service requirements.
 - This should significantly simplify deployment for those without an implementation of RUS, and eventually provide a fully functional implementation that could usefully replace existing implementations such as that used by the NGS.
 - It could also enable potential deployers to better assess the applicability to their own environment, before investing the effort in deployment.

Thus the framework is intended to assist services to provide their own customisable solution, not to prescribe a single solution for all services. The reviewers believe this approach is more appropriate for the diverse nature of the UK grid communities, than proposing a single ‘unified’ solution such as DGAS or UNICORE, which may require for example specific grid middleware components to be implemented.

This proposed framework is represented in the following diagram:

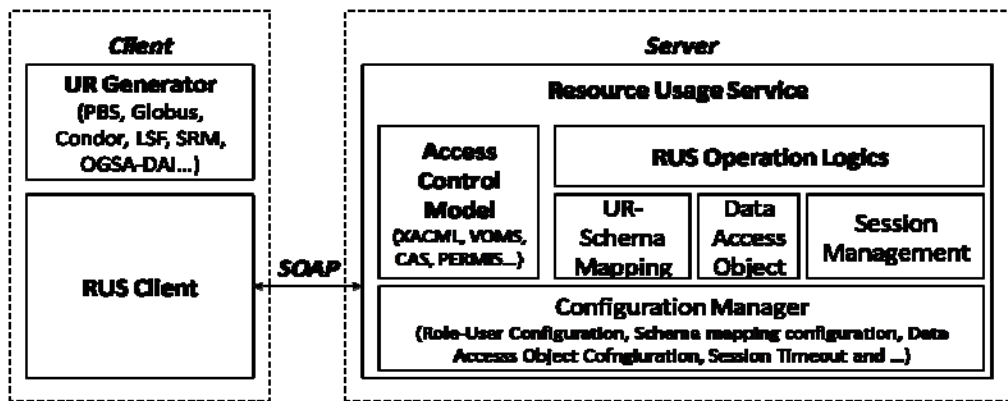


Figure 7.1: Proposed Framework (Component Architecture)

- Service Side
 - Access control module: an accounting tool should provide data privacy for usage records being shared. The module is also required to be delegated into other local security services (e.g. VOMS, CAS, PERMIS and etc.);
 - RUS operation logics: implementations of service interface definitions proposed within the OGF-RUS standard;
 - UR-schema mapping: an accounting tool is also required to be able to establish usage mapping to OGF-UR schema if custom usage representation schema is being used. (e.g. EGEE/WLCG project would transform OGF-UR records into WLCG schema and store them into a relational database.)
 - Session management: The session object can be used as temporary usage records being requested from or populated into RUS. The session object can also be used to maintain service transaction status.
 - Data access object: if usage records are stored persistently within a RUS service endpoint, the accounting tool should provide access facilities on underlying storage mechanisms (e.g. relational database or XML:DB). The data access object can be extended to enable hierarchical RUS deployment by exposing another RUS instance as a data access object.
 - Configuration manager: providing configuration facilities for the above RUS components;
- Client-side
 - UR generator: used to meter usage records in OGF-UR format;
 - RUS Client: the client-side APIs of the RUS service.

6.3 Requirement for a Usage Monitoring Framework

Most of the grid service providers have their own specific objectives and Service Level Definitions (SLDs) within their own organisations, so they need control over the ways in which data is presented and how the usage is managed, according to their own requirements. However, most of these services do have certain requirements in common:

- Usage Reporting - to account and report usage at a variety of levels for a variety of services, with access control to protect data privacy and malicious usage of accounting data
- Usage Policing and Management – to manage use of the resources and services through allocation and policing

- Authority devolution – to enable a role-based hierarchical management structure in management, allowing principal investigators or project managers to allocate resources for all in their virtual organisation.
- Charging - Most sites expect ultimately to charge for use of at least some of their resources and services. It would be expected that at least some aspects of charging could be determined within a user management system using service specific details, although it is recognised that this is only one component of the far more complex issue of economic models, which is well outside the scope of this review.

As discussed previously, there has been significant effort devoted to standardise the processes for generating, storing and sharing accounting data, but the same cannot be said of the mechanisms of reporting and managing the resources, despite the similarity of requirements amongst different service providers. In some cases, software provided by batch system developers, such as LSF, PBS, SGE, is being used to report on usage and manage use of the resources, but this is typically in homogeneous environments and is not easily extensible to grid services using different software, nor when the software is changed. Cluster management software such as Moab[27] Grid Suite provides greater flexibility across heterogeneous services. Such tools may be appropriate in some environments, but the desire for lightweight, customisable solutions may deter many from considering such a solution, particularly when significant costs may be involved.

6.3.1 Integrated User Management System

The reviewers believe there is scope for, and value in, some co-ordination of effort for the provision of customisable software or tools that provide a framework for such usage reporting and management activities. It is noted particularly that the national HPC services (CSAR, HPCx and HECToR), and the NGS, all have integrated user management systems, which incorporate job usage details within a framework for managing the services, and discussions with other sites have indicated the desire for such a system. Most such development is being performed independently, and as a consequence the systems are tied closely to the services for which they are designed. The Gold Allocation Manager, from Pacific North West Laboratories (PNNL), provides an extensible system for managing users and resources. This product is not linked to a specific service, and is open source. It is recognised that, with the exception of the NGS User Account Lifecycle, the systems mentioned are not designed specifically for grid based services, so work would be required to extend any of these tools for more generic use in a grid environment.

6.3.2 Usage Monitoring Framework

It is envisaged that a usage monitoring framework would provide the basis for an integrated user management system. The core of the system would be a user database comprising all user and project details, with a web interface or portal providing the service and user interface. This would interface to the RUS to obtain usage data as required. The modular structure would enable specific functions to be added as required. The basic service should provide allocation and policing, allowing site customisable actions when resources are exhausted. In the long term, additional modules for other functions such as charging could be incorporated.

The intention is that services use the customisable framework as a basis for their own requirements. As with the accounting framework, it should provide the flexibility to meet individual requirements, without prescribing the same solution to all.

Additional flexibility will be achieved by enabling the export of accounting data from an existing usage monitoring software to the RUS. This would for example assist sites already using software (not necessarily based on Usage Record format and RUS standards) to provide data to others (such as the NGS) through standard mechanisms.

6.4 Recommendations

There are a large number of tasks that need to be performed for the long-term deployment of grid accounting and many of these are listed in more detail below, but the following three recommendations summarise how the reviewers think JISC should focus its activities for both short and long-term developments, given the current status of grid based services in the UK:

- Standards - JISC should provide support, including funding for the involvement of UK community in the OGF UR and RUS working groups, to ensure that their general needs and key outstanding issues are addressed.
- Accounting Framework - JISC should fund the development of an accounting framework, that will assist deployments based on the standards specifications.
- Client/Usage Monitoring Framework - JISC should fund the development of a client based usage monitoring/management framework to provide a common basis for monitoring related activities.

In addition the reviewers recommend:

- A steering group comprising members from different stakeholder groups such as NGS, regional and campus grid, the HPC-SIG and others should be set up to oversee future developments.
- JISC should consider providing an advisory or support service, which provides a focus of expertise and assistance to those wishing to deploy such software, to encourage greater co-operation.

With respect to the three main recommendations, we now present additional details.

6.4.1 Recommendations for standards

Standards: JISC should provide support, including funding for the involvement of UK community in the OGF UR and RUS working groups, to ensure that their general needs and key outstanding issues are addressed.

Highest Priorities:

1. The OGF-UR specification:
 - a. The OGF-UR should be expanded with extra base properties (as discussed above)
 - b. A summary usage schema must be defined
2. The OGF-RUS specification:
 - a. The RUS specification needs to be addressed for performance and fault tolerance
 - b. The RUS specification must also support operations on summary usage records.

High Priorities:

3. Standardisation on storage usage record format
4. Usage record format for service level accounting should be addressed in terms of
 - a. Common service usage properties
 - b. Service-specific properties

6.4.2 Recommendations for an Accounting Framework

Accounting Framework: JISC should fund the development of an accounting framework, that will assist deployments based on the standards specifications.

Highest Priorities

5. Establish a general accounting framework conforming to OGF-RUS specification while providing extensibility on evolving accounting requirements;
 - a. Based on a well-defined accounting framework, further providing UR generator implementations on heterogeneous batch systems according to available OGF-UR usage record format;
 - b. Implementation on RUS operation logics (including summarization operations)
 - c. Access control mechanism

High Priorities

6. Development of
 - a. storage UR generator based on storage usage record schema;
 - b. service UR generator based on service usage record schema;
7. Hierarchical RUS deployment allowing flexible data sharing among sites, each of which has a RUS instance deployed;

6.4.3 Recommendations for a Client/Usage Monitoring Framework

Client/Usage Monitoring Framework: JISC should fund the development of a client based user monitoring/management framework to provide a common basis for monitoring related activities.

Highest Priorities

8. Determine the structure for a framework to provide integrated usage monitoring and user management functionality.
 - a. This should if possible use one of the existing systems (such as CSAR, HPCx, Gold, UAS) as a basis, but extended as necessary for grid services. It is expected that a local database will be accessed for user and project details, and the RUS for usage information. It is important that this development is given high priority, before many different services expend significant effort in developing their own facilities.
9. Develop the basic framework including most important functionality, such as usage policing, and reporting at different levels, and a portal or web interface to provide simple access.

High Priorities

10. Add additional functionality for longer term requirements such as a billing service.

7 Business Case for Suggested Implementation

The need for accounting in the wide range of computing and data based services is clear – the expansion of grid-based services will be hampered without satisfactory mechanisms to provide details of all aspects of usage. The recommendations aim to encourage a co-ordinated approach to the generation and use of accounting data within and across UK and international communities, standardising and simplifying the process and thus assisting in the development, deployment and management of many new grid-based services. The urgency of such developments is emphasised by the number of new and expanding services, all of which have common requirements, and need to provide the information as soon as the services start.

7.1 fEC

One of the implications of fEC is that research projects which need computing resources must identify the costs and obtain the funding to use the resources, and services providing these resources must recoup their costs. In some cases, this could be achieved by levying a fee for unlimited access to a service, as with some

existing data based services, but this occurs and is generally acceptable only when actual resource usage is relatively low and equitably distributed. This may continue to be viable for such services, but services that are resource intensive, which is typically the case for all compute based services provided by the NGS, regional and campus grids, local non-grid services, and some data based services, will be expected to provide detailed accounting details and ultimately to charge for the services. For 'standalone' services such as those based on single large systems, it is possible to produce such information using existing software, provided by (purchased from) the vendor or another software supplier. However once multiple systems and services are used, as will be the case in most grid services, this simple approach becomes impractical. The heterogeneity and complexity of grid services means that a co-ordinated and sophisticated approach is required to address the accounting requirements in such environments.

Even if a service is not currently driven by fEC, it is highly likely that the service provider will be expected to report on usage of the service, identifying the major users, whether at a site, school/department, project or individual level. In addition many services will wish to control usage of the limited resources to ensure all users have access to a 'fair' share of the resources. Even the simplest services need to uniquely identify all users, if only to ensure they are eligible to use the service. All of these activities are far more complex in grid-based environments.

7.2 Common Approach for UK Grid Accounting

At present there is no co-ordinated approach to the provision and development of grid accounting and usage monitoring in the UK communities. The major grid service providers are currently GridPP and the NGS. GridPP has a framework for its own use, and software developed for this purpose, particularly APEL and R-GMA. The NGS has used the OGF standards UR and RUS, but encountered some problems in doing so, and has therefore adapted this approach to meet their own requirements, and developed an integrated user management system for their service. The national HPC services from CSAR and HPCx to HECToR have developed their own software/tools to meet their specific requirements, although most of their requirements in accounting (not specifically grid accounting) are becoming much more widespread across the computing communities. Campus grid are providing very varied solutions to similar problems and requirements – from those who wish to be affiliates or partners of NGS, to those who have a narrower focus within their institution.

Some of the tools in existence have been adopted by a number of services linked through a common project. For example SGAS by SWeGrid/NorduGrid, APEL by EGEE/WLCG UNICORE by DEISA, DGAS by EGEE. This provides a focus for developments within these communities. There is as yet no comparable focus in software/tool development and provision across the UK. This has led to the diverse approach seen today in the UK.

It would be of great value for the UK communities to adopt a common approach, both to channel the expertise in a similar direction and to minimise duplication of effort, and thus make more cost effective use of available resources. If funding is provided for a focussed approach with the UK communities supporting the objectives, this should result in close co-operation between those involved in the developments and the current and emerging service providers. In addition the use of standards compatible software will be beneficial for interoperability with other existing standards compatible tools.

National and international HPC services have led the way in developing integrated systems to account for and manage a variety of different resources in a tightly coupled service environment. Similar developments are required in the more diverse, loosely coupled world of grid computing. However the distributed and heterogeneous nature of this environment means that an approach based on standards, modularity and customisability are crucial for widespread adoption.

7.3 Implementation of Recommendations and Cost Estimates

Estimates are given below for the time to develop the proposed frameworks. The times are given in person months for an employee (or employees) already familiar with the topics. The reviewers believe that the times should be sufficient to provide working software, but will not be sufficient to cover all eventualities and functionalities that may be deemed desirable. However the frameworks are intended to be extensible for additional functionality as and when required. With sufficient involvement from others, such as batch or accounting system providers, some of the effort may be reduced. The details given below refer directly to the recommendations listed in section 6.4.

It would be expected that full economic costs would apply, meaning about £100,000 per annum, or about £8000 per month.

Modest additional funds may be required for:

- a PI to oversee the project(s)
- expenses for steering group members and others who wish to play active roles in the development of the standards
- travel to ensure close contact with grid service providers and possibly other relevant software developers (such as batch system providers who may collaborate in the development of UR generators) during the project and to some meetings/conferences such as the Open Grid Forum.
- for access to relevant grid services (although it would be hoped that such services would co-operate closely in the developments)
- basic equipment such as a PC

7.3.1 Recommendations for Standards

These recommendations are to ensure as far as is possible that the OGF standards are developed to meet the needs of the UK communities.

The value of the standards is to assist in providing a common framework for accounting. Generating and using accounting data in the same format will simplify the exchange of such data between services. (The NGS requires accounting data from partners, and it may encourage partnerships if potential sites already generate accounting data in an appropriate format. Similarly campus grids may have a variety of systems and software, and the use of a common approach for accounting should be beneficial.) Standardisation should also enable software to be more modular and interoperable, thus minimising the development of multiple solutions. Adopting a standards based approach for current, relatively simple, accounting requirements will also help in providing a sound basis for accounting and charging for future services which make complex use of diverse and distributed resources.

The effort and cost for this recommendation would primarily focus on expenses for a steering group and others who participate in developing the standards on behalf of the UK communities. The project investigators would be expected to contribute in this area as part of their normal activities.

7.3.2 Recommendations for Accounting Framework

The framework is intended to assist services in deploying standardised accounting software thus enabling and simplifying the exchange of accounting data between services.

The development of a UR generator will enable those who wish to use accounting data from batch system software, to 'plug-in' the appropriate generator(s) rather than writing their own. Additionally, UR generators can be developed to enable interoperability from an existing software accounting package (such as Gold, ARCO or MOAB) to the RUS, thus assisting sites who already have accounting in place to export required accounting data to other sites such as the NGS.

The RUS implementation is intended to simplify the deployment for those without an implementation of RUS, and eventually provide a fully functional implementation that could usefully replace existing implementations such as that used by the NGS. It could also enable potential deployers to better assess the applicability to their own environment, before investing the effort in deployment. The UR-schema mapping will enable the establishment of usage mapping to OGF-UR schema if a custom usage representation schema is being used, as is currently the case for the NGS.

Highest priorities

- Usage Record Generators – 6 months
- RUS implementation – 12 months
- Access control mechanism – 3 months

High Priorities

- Storage UR generator – 1 month
- Service UR generator – 1 month
- Hierarchical RUS deployment – 3 months

7.3.3 Recommendations for a Client/Usage Monitoring Framework

The objective is to develop an extensible and customisable framework, providing the basic functionalities required by most services, particularly including extensive reporting capabilities and allocation management. It should be applicable in both grid and non grid-based services, thus increasing the applicability and value of the software. Extensions for example in charging could be incorporated after the basic framework is in place. The time required will depend on the functionalities required, the approach adopted (particularly whether based on existing software) and the level of testing/documentation needed. A development system (developed in conjunction with the NGS for example), could be produced relatively quickly, enabling progress to be made, but quality assurance for a range of environments will require significant effort.

Highest priorities

- Determine structure and develop prototype system – 6 months
- Generate distributable version – 12 months

High Priorities

- Provide additional functionality – 6 months

8 Risk Analysis

8.1 Stakeholders do not accept the recommendations

This could result in service providers continuing or starting to develop their own solutions.

- *After discussions with some of the stakeholders about the recommendations, this is thought to be unlikely*

8.2 Grid communities do not adopt the proposed frameworks

This could be for a number of reasons:

- The timescale is too long, so the service providers cannot wait for the work to be completed. An alternative approach is adopted and there is likely to be reluctance to change once the work is completed.

- *It is recognised that timescales are short for all concerned, both the established services like NGS, who need to enhance existing mechanisms, and newly developing services which need accounting in place as quickly as possible. The approach suggested is to separate the work into almost independent components, which can be developed reasonably independently, with priorities determined in conjunction with the service developers. Thus the service developers will be able to co-ordinate their work with the planned development of specific components. The proposed work should be performed in close co-operation with the service providers.*
- Expected standards do not materialise
 - *Some of the proposals are based on expected changes in the UR and RUS standards, and it may not be possible to wait until these have been agreed before proceeding. The extensions mentioned in the proposals have received much attention, and it is expected they will be agreed. .Provided the concerns raised by the stakeholders are addressed, and the deployments are successful, adopting this approach would still provide a common focus, and the fact that OMII-Europe is also promoting such deployments adds further support*
- The performance, particularly of RUS, is deemed inadequate
 - *This is one of the concerns already raised, and it should be addressed. As noted previously, some of the enhancements planned should assist with performance. It is also understood that GridPP is planning to evaluate the RUS in this context independently of further developments, so this should provide valuable feedback..*

8.3 Insufficient take-up by service providers

- *This issue is one of the reasons why the reviewers have suggested a framework with components that can be deployed and extended as required at different sites – it is believed that many sites are currently facing the same issues, and potentially duplicating effort in developing their own software/tools. The objectives of this framework are to provide a basis from which all can customise as required. Providing there is sufficient support from the community for this approach, and close co-operation between the developers and the service providers, the risk of this should be minimised.*

9 References

- [1] Open Grid Forum Usage Record Format (version 1.0), <https://forge.gridforum.org/projects/ur-wg/>
- [2] GridPP, <http://www.gridpp.ac.uk/>
- [3] HPCx, <http://www.hpcx.ac.uk/>
- [4] Storage Resource Broker, http://www.sdsc.edu/srb/index.php/Main_Page
- [5] Open Grid Service Architecture-Data Access and Integration, <http://www.ogsadai.org.uk>
- [6] Sun Grid Engine, <http://www.sun.com/software/gridware/>
- [7] Open Grid Forum Usage Record Format (version 1.0), <https://forge.gridforum.org/projects/ur-wg/>
- [8] J. Ainsworth, S. Newhouse and J. MacLaren, Resource Usage Service (RUS) based on WS-I Basic Profile 1.0, 2007, https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.ruswg/docman.root.documents.version_1_o.draft_wsi_rus_17/doc7965
- [9] EGEE (Enabling Grids for E-science), <http://www.eu-egee.org>
- [10] WLCG (LHC Grid), <http://lcg.web.cern.ch/LCG/>
- [11] NISO Standardized Usage Statistics Harvesting Initiative (SUSHI), http://www.niso.org/committees/SUSHI/SUSHI_comm.html
- [12] Counting Online Usage of NeTworked Electronic Resources (COUNTER), <http://www.projectcounter.org/>

- [13] R. Byrom et al. "APEL: An implementation of Grid accounting using R-GMA", *UK e-Science All Hands Conference*, Nottingham, 2005.
- [14] Relational-Grid Monitoring Architecture (R-GMA), <http://www.r-gma.org/>
- [15] X. Chen and A. Khan, Design and Performance Analysis of Resource Usage Service for LCG, *Nuclear Science Symposium Conference*, IEEE, Oct, 2006, vol. 1 pp. 603-606.
- [16] Distributed Grid Accounting System (DGAS), <http://www.to.infn.it/grid/accounting/main.html>
- [17] SweGrid Accounting System (SGAS), <http://www.sgas.se>
- [18] SweGrid, <http://www.swegrid.se>
- [19] Gratia, <https://twiki.grid.iu.edu/twiki/bin/view/Accounting/WebHome>
- [20] Open Science Grid (OSG), <http://www.opensciencegrid.org/>
- [21] National Grid Service (NGS), <http://www.grid-support.ac.uk/>
- [22] Uniform Interface to Computing Resources (UNICORE), <http://www.unicore.eu/>
- [23] Web Service Resource Framework (WS-RF), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
- [24] Gold Allocation Manager, <http://www.clusterresources.com/pages/products/gold-allocation-manager.php>
- [25] W. Frings, M. Riedel A. Streit and et al, LLview: User-level Monitoring in Computational Grids and e-Science Infrastructures, *German e-Science Conference 2007*.
- [26] Computer Services for Academic Research (CSAR), <http://www.csar.cfs.ac.uk>
- [27] Moab Grid Suite, <http://www.clusterresources.com/pages/products/moab-grid-suite.php>

10 Acknowledgements

The authors thank in particular the following individuals and organisations for their time, support and input during the course of this review. We apologize to others who have also contributed, but whose names have been omitted.

Cliff Addison, Mike Baker, Stephen Booth, John Brooke, Mark Calleja, Neil Chue-Hong, Keith Cole, Tony Doyle, Alistair Dunlop, James Farnhill, Donal Fellows, Alan Ferguson, Matt Ford, Robert Frank, Neil Geddes, John Gordon, Clare Gryce, Andrea Guarise, Kevin Haines, Robert Haines, Paul Hatton, Mark Hewitt, Terry Hewitt, Chris Higgins, Andrew Jones, David Kant, Kami Kitmitto, Laura McGinnis, Ross McIntyre, Gilbert Netzer, Jeremy Nowell, Stephen Pickles, Rosario Piro, Andrew Richards, Joanna Schmidt, Dave Snelling, Steve Thorn, Sven van den Berghe Jon Wakelin, David Wallom, and Katie Weeks.

DEISA, EDINA, EGEE, EPCC, Fujitsu Labs UK, GridPP, HPC-SIG, INFN, LCG, MIMAS, NGS, NeSC, NW-Grid, OGF, OMII-Europe, OMII-UK, PSC, ScotGrid, University of Bristol, University of Cambridge, University of Oxford, UCL, University of Manchester, University of Reading, and White Rose Grid.

APPENDICES

A Stakeholder Reviews

A.1 National Grid Service

The NGS aims to provide computational and data based resources and facilities to UK researchers, independent of resource or researcher location. This is currently achieved using resources (both compute and data) at four core sites (RAL, Oxford, Leeds and Manchester), and a growing number of partner and affiliate sites, together with the provision of software and services, to enable a consistent method of access to any resource from any location. As resources may have different 'owners', each of whom may have different charging policies, it is essential there is a reliable mechanism to account for all aspects of use, in an environment with dynamically varying resources and services.

The NGS already has a sophisticated accounting system in operation and needs to extend the functionality and scope to meet its objectives and address future service requirements. There is a strong desire to use a standard approach maximizing interoperability with other services, and enabling straightforward deployment on sites wishing to partner with the NGS. Major stakeholders to the NGS in the context of accounting and usage monitoring are the grid operations support centre, software developers and standards bodies, current and potential partner and affiliate sites (including campus grids and SRIF funded clusters), funding bodies and end users.

Key requirements

- Performance
- Interoperability – clearly defined APIs / protocols to enable exchange of information with:
 - partner/affiliate sites and dataset providers
 - large scale grid projects such as GridPP/WLCG
- Ease of deployment
- Ability to trace individual jobs; legal requirement for auditability to an individual
- Ability to view historical usage data at user, VO and resource levels
- Metrics:
 - Required – CPU time, Wall time, permanent storage, data services
 - Desirable – executable, memory usage, network usage, QoS
 - Not generally of concern – temporary storage
- Resource allocation and policing
- Custom charge rates for QoS, eg. advanced reservation
- Integration with user/project management system
- Integrity of accounting data through automated monitoring/notification systems

Key concerns/issues

- RUS querying currently not functional
- Current accounting methodologies and practices are batch job centric
- Interfaces alone should be standardized, allowing site-specific implementation
- Significant investment in current system – would need to see clear benefit in change
- Partner/affiliates not wishing to entrust their data to a centralized site

A.2 GridPP

GridPP is a collaboration of particle physicists and computer scientists from the UK and CERN, with distributed compute resources spanning 17 UK institutions. GridPP has a number of key stakeholders – it is the UK's contribution to worldwide Large Hadron Collider (LHC) Grid (WLCG), overseeing the Tier 1 facility at RAL and the Tier 2 organisations of ScotGrid, NorthGrid, London and SouthGrid, and also contributes to the interdisciplinary project EGEE - Enabling Grids for E-science.

LCG is a production-level grid and GridPP has a contractual obligation to provide accounting data as part of the LCG project. At present over 150 sites worldwide are publishing accounting data to the Grid Operations Centre (GOC) at RAL making aggregation, scalability and validation of accounting data critical concerns.

Key requirements

- Performance and scalability
- Ability to view historical usage data at VO, resource, country and EGEE region levels
- Metrics:
 - Required – CPU (normalized to reflect “work done”)/Wall time
 - Desirable – permanent storage
 - Not generally of concern – memory usage, network usage
- Interoperability across international production grids
- Integrity of accounting information through automated monitoring/notification systems
- Ability to modify records eg. siteName change does not break historical querying

Key concerns/issues

- Significant investment in current system – 150 sites publishing via APEL/R-GMA
- Scalability of RUS – XML only useful as an exchange format
- CPU normalization and benchmarking needs to be addressed
- Sharing of accounting data across different grids poses difficulties in terms of data protection
- Charging mechanism should be separate and require digital signatures and auditability

A.3 Campus Grids

The accounting requirements of campus grids across the UK academic sector range from simple “best effort” usage statistics from condor pools to sophisticated job-level accounting across a range of disparate resources. In cases where departmental resources or SRIF-funded hardware are available to the grid there is a more urgent requirement for accounting as a direct consequence of the fEC model (see Other Compute Services, to follow). Less mature campus grids can see immediate benefit from the development of a clearly defined accounting framework and tools to prevent further duplicity of effort.

Key Requirements

- Performance
- Interoperability with NGS / other grids but flexibility to allow site-specific access control policies
- Ability to trace individual jobs
- Ability to view historical usage data at user, project, School, and resource levels
- Resource allocation and policing
- Charging mechanisms for fEC (especially HPC component)
- Metrics:
 - Required – CPU time, Wall time, permanent storage
 - Desirable – memory usage, full job command line
 - Not generally of concern – temporary storage, network usage

Key concerns

- Performance of XML database
- Interfaces alone should be standardized, allowing site-specific implementations
- RUS aggregation needs attention
- Wide range of job managers: Linux/Windows Condor, Windows Compute Cluster, PBS, TORQUE, LSF
- Solution should be lightweight and not be tied to a specific project

A.4 Regional Grids

Most regional grids currently operate fairly homogeneous systems at different sites and thus can provide the service with a limited range of software such as a single batch system, and therefore do not, as yet, require the same degree of flexibility as NGS or some campus grids.

Key Requirements

- Contractual obligations to provide accounting data to specific large scale projects, eg. GridPP
- Interoperability with campus grids
- Ability to trace individual jobs
- Ability to view historical usage data at user, project/VO, University, and resource levels
- Resource allocation and policing
- Devolution of allocation management to PIs
- Charging mechanisms required in the future
- Required metrics:
 - Required – CPU time, Wall time
 - Desirable – permanent storage
 - Not generally of concern – temporary storage, network usage

Key Concerns

- Data protection

A.5 Other Compute Services

There is an increasing number of universities providing or starting to provide large scale local compute services, particularly after the recent SRIF funding programmes. In many cases this has resulted in a 'standalone' service, typically for local high performance computing (HPC), even at sites where there is or has been campus grid activity, such as Oxford, Cambridge and UCL. Many such services are influenced by fEC and thus need to manage and report on usage. While it may be relatively simple for such services to use resource management or batch engine software to address the accounting requirements, it may be at the cost of interoperability or extensibility for future services. Nevertheless some such services are developing their own accounting and user management systems not tied to a specific supplier, thus providing greater long-term flexibility, but also requiring significant development effort. Thus the objectives for the grid communities, in providing a standard approach for usage data metering, storage and sharing, could be of great value to these other specialist services.

It is recognised that where significant effort has already been invested and the service requirements fully met, such as the national HPC services, there is unlikely to be a good reason for changing existing practices in the short-term. However it would be hoped that such services would see the long-term benefit of a co-ordinated approach, ideally resulting in convergence in development. It is known for example that the developers of the 'SAFE' system used by the national HPC services, are developing a generator for converting SAFE-specific usage information into OGF-UR records, and are in the process of implementing a RUS service. It should be made clear that the UR format is not useful only for grid environments – it is a standard format for storing job usage information, which may be used for accounting on any system.

Key Requirements

- Job tracing
- Historical usage monitoring at project and user levels
- Management of project resources (sub-allocation)
- Automated policing
- Integration with user management system
- Accuracy of accounting data critical – charging
- Auditability

Key Concerns

- Independent contractual arrangements regarding data protection
- Significant investment in current accounting system(s)

A.6 End user

Key Requirements

- Intuitive interface, preferably integrated with user management interface
- Job tracing
- Historical usage monitoring at VO and user levels
- Management of project/VO resources (sub-allocation)
- Confidence in the accuracy of accounting data – critical if being charged

A.7 Standards Bodies

There appears to be general support in the grid communities for the OGF-UR and RUS specifications as standards for storing and sharing usage information. OMI-Europe, who are concerned with interoperability between different Grid systems through the implementation of common standard interfaces, are evaluating the implementation of RUS interfaces for the gLite, Globus and UNICORE middleware stacks. To this end, preliminary design documents have been prepared for the SGAS, DGAS and UNICORE accounting systems.

Key Requirements

- Acceptance and rollout of OGF usage record format
- Support for development and adoption of aggregated usage record format
- Support for development and adoption of storage usage record format
- Support for development and adoption of network usage record format
- Understanding of more complex use cases
- Hierarchical and P2P RUS deployments

Key Concerns

- Site implementations not strictly standards compliant
- Standards not flexible enough to cater for individual accounting requirements
- Standards too bloated for individual requirements
- Issues regarding RUS specification querying interface
- Is Xpath querying expressive enough?

A.8 Data service providers

There are a large number of data based services funded by JISC, including the MIMAS and EDINA services. There is an increasing interest in the 'grid enablement' of these services, which includes the management of security through grid mechanisms; the ability to combine and analyse data in distributed datasets; and the ability to access grid based (compute) resources dynamically at periods of high loads. There have been a number of grid enablement pilot projects including Gameda, GEMS (1 and 2), GESSE and SEE-GEO but there are few if any production grid based data services. Authentication and authorisation are key issues in this context – the services currently use ATHENS or Shibboleth, rather than grid certificate authentication.

Most of the data services are required to provide accounting details to JISC on a regular basis as defined by SLAs. The statistics reported are primarily concerned with the number of accesses and searches, on a per site basis, as well as service availability. In addition the service providers need to ensure that accesses are restricted to licensed users (whether individual or site based), so the ability to identify the user of the service is crucial.

Thus most of the data based services are required to provide service usage accounting, rather than resource usage accounting typically required by grid (and other compute based) services; However there are some specialist services, such as the satellite image service, which do have significant resource usage requirements.

While the NGS, for example, does see a long term need for service usage accounting, recognising that such services may be provided through NGS itself, even though the data is hosted elsewhere, there is little in the present standards framework to address this type of accounting. It is not clear to the reviewers how best this should be addressed. It should in principle be possible to define such metrics, but whether it is appropriate or desirable to extend the UR specification, for example, for this purpose is certainly questionable: the UR has been designed with resource based accounting in mind, not service accounting.

In addition, it is clear that many of the current services are well established, and the mechanisms used for collecting the statistics frequently closely integrated with the service itself. The adoption of a new approach for the collection of the statistics across a range of services would probably not be considered favourably. Thus, the reviewers believe it is outside the scope of this review to provide tangible recommendations in this context, although it is felt that such issues should be addressed through further exploratory projects in setting up 'grid enabled' services, and subsequently establishing new grid based services as required, rather than adapting accounting mechanisms in existing services.

With respect to some services such as the satellite image service, very large amounts of data must be stored, analysed, and possibly downloaded, and JISC may request information on resource usage to demonstrate a requirement of the service, in order to justify funding streams. The focus is on service access to justify the provision of the service. It is likely that there will be an increase in resource usage associated with these and other data based services, particularly when utilising multiple distributed datasets – something that has not easily been possible previously. This is likely to result in additional accounting requirements, although it does depend (at least partly) on the funding bodies - for example on whether JISC continues to focus on service usage accounting, with little direct interest in details of compute, storage and network usage. However if the service is grid based, with significant storage, network traffic, and high compute requirements possibly at hosts determined dynamically, the owner of the resources will need to be able to charge for use of these resources. Thus it seems essential in the long-term that a mechanism is developed to account for all of these activities. The approaches adopted in the grid accounting context should be applicable to these types of services, bearing in mind the work and time still required to address usage of resources involving storage and network activities.

Key requirements

- Data security, authentication/authorization
- Accounting in workflows: single access/instance may involve multiple services
- Metrics:
 - Required: number of logins, searches, amount of data downloaded, nature of data downloaded
 - Desirable: permanent storage (resource provider end) and network usage if significant downloads are performed
 - Not generally of concern : temporary storage

Key Concerns

- Charging model does not fit easily in job-level accounting schema
- Distribution of datasets presents difficulties with respect to licensing
- Grid enablement still in its infancy

B Standards in Grid Accounting

B.1 OGF Usage Record Format

The OGF-UR working group defines an XML-format usage representation on a per job basis. The metrics listed in OGF-UR are divided into three categories: base properties, differentiated properties, and extension framework.

Base Properties

The base properties, listed in the following table, define common usage metric requirements for a usage record, such as user and job identification. Compulsory fields are marked with (*)

Table B-1: OGF-UR Base Properties

Base Metric Name	Base Data Type ⁴	Semantic Clarification
RecordIdentity (*)	xsd:string	Uniquely defines a record in a set of all usage records among sharing communities
GlobalJobId	xsd:string	Global job identifier as assigned by a metascheduler or federation scheduler
LocalJobId	xsd:string	Local job identifier assigned by the batch queue
ProcessId	xsd:integer	The process id of the job
LocalUserId	xsd:string	The local user identity associated with the resource consumption
GlobalUsername	xsd:string	The global user identity associated with the resource consumption
JobName	xsd:string	The job name or application/executable name
Charge	xsd:float	Total charge of the job calculated by a (site-dependent) pricing system
Status	xsd:string	The completion status of the job
WallDuration	xsd:duration	Wall clock time elapsed for the job
CpuDuration	xsd:duration	Cpu duration summed over all processes of the job
EndTime	xsd:timestamp	The time at which the job completed
StartTime	xsd:timestamp	The time at which the job started
MachineName	xsd:string	A descriptive name of the machine on which the job ran
Host	xsd:string	The system host name on which the job run
SubmitHost	xsd:string	The system host name from which the job was submitted
Queue	xsd:string	The name of the queue from which the job was executed or submitted
ProjectName	xsd:string	The project associated with resource usage

⁴ The base data type refers to those of XML primitive data type. Most usage properties are defined as instance of base data types, while some of them are simple types derived from XML based data types. The table only lists base types from which simple types derived.

Differentiated Properties

Differentiated properties are resource consumption measurements that correspond to a grid organisation's particular requirements. The differentiated properties are listed in the following table.

Table B-2: OGF-UR Differentiated Properties

Differentiated Metric Name	Base Data Type	Semantic Clarification
Network	xsd:positiveInteger	The amount of network resource used by the job.
Disk	xsd:positiveInteger	The storage used by the job
Memory	xsd:positiveInteger	The amount of memory used by all processes in the job
Swap	xsd:positiveInteger	The swap usage of the job
NodeCount	xsd:postiveInteger	Number of nodes used by the job. The definition of "node" is site dependent
Processors	xsd:positiveInteger	The number of processors the job requested or used
TimeDuration	xsd:duration	Additional measure of time duration associated with the job
TimeInstant	xsd:dateTime	Additional timestamp property associated with the job e.g., time when queued
ServiceLevel	xsd:string	Quality of service associated with the resource consumption

Extension Framework

The extension properties are intended to be used for custom usage metric representations that are outside the scope defined by the base and differentiated properties. The extension properties introduced within the OGF-UR are mainly used for encoding custom resource properties. The extension element definitions are listed in the following table.

Table B-3: OGF-UR Extension Property List

Extension Name	Metric	Base Data Type	Semantic Description
Resource		xsd:string	Extension property regarding the resource associated with the usage
ConsumableResource		xsd:string	Extension property regarding a measured resource associated with the usage
PhaseResource		xsd:string	Extension property regarding certain measured resource usage with phase unit attribute
VolumeResource		xsd:string	Extension property regarding certain measured resource usage with a storage unit attribute

B.2 OGF Resource Usage Service

Core Features

The core features defined in OGF-RUS specification provide common operations, mainly in uploading and retrieval of usage records through RUS.

Usage Population

The RUS service allows usage records to be uploaded through the “storage” service interface. A comparison between the storage service interface definition in version 1.7 and version 1.9 are illustrated in the following table.

Table B-4: RUS Storage SID

SID (version 1.7)			SID (version 1.9)			Description
Interface Name	Input	Output	Interface Name	Input	Output	
InsertUsageRecords	A list of single usage record elements	Operational results and a list of inserted record identities	InsertUsageRecords	A list of OGF-UR usage record files, each of which may contain multiple single usage records	Operational Result containing inserted record identity	Population of usage records through RUS

Usage Query

The RUS service defines a set of extraction interfaces for usage monitoring and tracing through RUS. The comparison between extraction SIDs in version 1.7 and version 1.9 are illustrated in the following table.

Table B-5: RUS Extraction SIDs

SID (version 1.7)			SID (version 1.9)			Description
Interface Name	Input	Output	Interface Name	Input	Output	
extractRUSRecordIds	Search Term to identify target RUS usage records	Operational results and a list of RUS usage record identities	extractRecordIds	Search term expressed as XPath statement	Operational Result containing target OGF-UR record identities	Extraction of usage record identities of matched usage records
extractRUSUsageRecords	Search Term to identify target RUS usage records	Operation Result and zero or more RUS usage records	extractUsageRecords	Search term expressed as XPath statement	Operation result and a single usage record file containing zero or more matched OGF usage	Extraction of usage records according to search terms

					records.	
			extractSpecUsageRecords	A list of usage record identities	Operation result and a single usage record file containing zero or more matched OGF usage records.	Extraction of usage records according to specific record identities.

Administration

The RUS service defines a set of administration interfaces for advanced users who need to perform modifications on usage records through RUS. The comparison between administrative SIDs in versions 1.7 and 1.9 are illustrated in the following table

Table B-6: RUS Administrative SIDs

SID (version 1.7)			SID (version 1.9)			Description
Interface Name	Input	Output	Interface Name	Input	Output	
incrementUsageRecordPart	RUS record identity and XPath statement	Operational results and a list of RUS usage record identities				Add a numeric increment to an element in a single record
modifyUsageRecordPart	RUS record identity, and XUpdate statement	Operation Result and XUpdate Result	modifyUsageRecords	XUpdate statement	Operation result containing updated usage record identities	Modification of usage records as XUpdate statement
			modifySepcUsageRecords	A list of usage record identities	Operation result containing updated usage record identities	Modification of specific usage records
replaceUsageRecords	RUS record Id and a list of OGF-UR usage records, where each id corresponds to a single usage record.	RUS record identity list and operation result	replaceUsageRecords	OGF-UR usage records as a single XML file	Operation result containing replaced usage record identities	Replacement of a list usage records matched by record identities
deleteRecords	Search Term to identify target RUS	RUS record identity list and operation	deleteUsageRecords	XPath statement as search term	Operation result containing a list of	Deletion of all usage records that match the specified

	usage records	result			deleted usage record identities	criteria.
deleteSpecificRecords	RUS record identity list	RUS record identity list and operation result	deleteSpecUsageRecords	A list of usage record identities	Operation result containing a list of deleted usage record identities	Deletion of usage records with the specified record identities

Configuration

The only configuration information required for RUS SIDs is the mandatory element list, which specifies the mandatory usage properties in usage records to be populated through RUS. Therefore RUS provides an interface definition for querying the mandatory usage properties. There is no difference for this interface definition in version 1.7 and version 1.9. The definition is illustrated in the following table.

Table B-7: RUS Configuration SID

SID (version 1.7 / 1.9)			Description
Interface Name	Input	Output	
ListMandatoryUsageRecordElements	NONE	A list of OGF-UR metric elements	List mandatory usage properties as configuration.

Advanced Features

Advanced features such as aggregation and data replication are currently being discussed by the OGF working group. There is no RUS specification relating to these features as yet.

- Aggregation

The aggregation advanced feature of RUS is intended to be an extension of extraction SIDs to allow the usage records to be returned in a summary format, grouped by certain grid entities. This feature of RUS contributes to the aggregate accounting.

- Data Replication

Data replication is a feature to provide synchronously or asynchronously replication amongst usage record storage devices.

B.3 EGEE/WLCG Summary Usage Schema

Table B-8: EGEE/WLCG Aggregate/Summary Usage Schema

Aggregate Metric Name	Base Data Type (SQL)	Description
ExecutingSite	varchar (50)	The executing site name for jobs summarised in the usage record
LCGUserVO	varchar (50)	The VO Name for jobs summarised in the usage record
Njobs	int (11)	The total number of jobs summarised in the usage record
SumCPU	decimal(10, 0)	The total CPU time of jobs summarised in the

		usage record
NormSumCPU	decimal (10, 0)	The normalised total CPU time of jobs summarised in the usage record
SumWCT	decimal (10, 0)	The total wall clock time of jobs summarised in the usage record
NormSumWCT	decimal (10, 0)	The normalised total wall clock time of jobs summarised in the usage record
Month	int (11)	For fast accounting of a specific month
Year	int (11)	For fast accounting of a specific year
RecordStart	date	The start date of jobs summarised in the usage record
RecordEnd	date	The end date of jobs summarised in the usage record

B.4 EGEE/WLCG Storage Usage Schema

Table B-9: EGEE Storage Accounting Schema

StorageRecords	Base Data Type (SQL)	Description
RecordIdentity	VARCHAR	Unique record identity
ResourceIdentity	VARCHAR	Storage Identity defined in GLUE Schema
Grid	VARCHAR	EGEE, OSG, etc
ExecutingSite	VARCHAR	The site name providing the storage resource
VO	VARCHAR	The virtual organization name
SpaceUsed	INT	The space consumed
SpaceAvailable	INT	The space available
Total	INT	The total space allocated
Unit	VARCHAR	The storage unit (GB, MB and etc.)
SEArchitecture	VARCHAR	The type of storage (Tape, Disk, Multidisk and etc.)
Type	VARCHAR	The type of storage in terms of its persistence (Permanent, volatile and etc.)
EventDate	Date	The event date (in the format of YYYY-MM-DD) when storage usage a metered
EventTime	Time	The event time (in the format of HH:MM:SS) when storage usage a metered
MeasurementDate	Date	The measure date (in the format of YYYY-MM-DD) provided by RGMA at publish time
MeasurementTime	Time	The measure time (in the format of HH:MM:SS) Provided by RGMA at publish time

C Accounting Tool Assessment

C.1 Schema Mapping

Grid Accounting Schema

Table C-1: Grid Usage Metric Mapping to OGF-UR

OGF-UR	APEL & DGAS System		NGS Accounting System	
Metric Name	Metric Name	Base Data Type (SQL)	Metric Name	Base Data Type (SQL)
RecordIdentity	RecordIdentity	VARCHAR	RUSId	VARCHAR
			RecordId	
GlobalJobId	LCGJobId	VARCHAR	—	—
LocalJobId	—	—	LocalJobId	—
ProcessId	—	—	—	—
LocalUserId	LocalUserId	VARCHAR	LocalUserId	VARCHAR
GlobalUserName	LCGUserId	VARCHAR	X509SubjectName	VARCHAR
JobName	LocalJobId	VARCHAR	JobName	VARCHAR
Charge	—	—	—	—
Status	—	—	Status	VARCHAR
WallDuration	ElapsedTime	VARCHAR	WallDuration	NUMBER
CpuDuration	BaseCpuTime	VARCHAR	CpuDuration	NUMBER
EndTime	StopTime	VARCHAR	EndTime	DATE
	StopTimeUTC	VARCHAR		
StartTime	StartTime	VARCHAR	StartTime	DATE
	StartTimeUTC	VARCHAR		
MachineName	—	—	MachineName	VARCHAR
Host	ExecutingCE	VARCHAR	—	—
SubmitHost	—	—	SubmitHost	VARCHAR
Queue	—	—	—	—
ProjectName	LCGUserVO	CHARVAR	—	—
Network	—	—	—	—
Disk	—	—	—	—
Memory	MemoryReal	INT	—	—
	MemoryVirtual	INT		
Swap	—	—	—	—
NodeCount	—	—	—	—
Processors	—	—	Processors	NUMBER
TimeDuration	—	—	wallTimeRequested	NUMBER

			cpuTimeRequested	NUMBER
TimeInstant	EventDate	DATE	timeGlobusSubmitted	DATE
	EventTime	Time		
	MeasurementDate	DATE	pbsLogDate	DATE
	MeasurementTime	TIME		
ServiceLevel	—	—	—	—

Cluster Accounting Schema

Table C-2: Cluster accounting System Usage Metrics Mapping to OGF-UR

OGF-UR	Gold		ARCO System	
Metric Name	Metric Name	Base Data Type (SQL)	Metric Name	Base Data Type (SQL)
RecordIdentity	g_id	INT(11)	j_id	INTEGER
GlobalJobId	—	—	—	—
LocalJobId	g_job_id	VARCHAR(255)	j_job_number	INTEGER
ProcessId	—	—	j_task_number	INTEGER
LocalUserId	g_user	VARCHAR(255)	j_owner	TEXT
GlobalUserName	—	—	—	—
JobName	g_executable,	VARCHAR(255)	j_job_name	TEXT
	g_application	VARCHAR(255)		
Charge	g_charge	DOUBLE	ju_slots	TEXT
Status	—	—	ju_exit_status	INTEGER
WallDuration	g_wall_duration	INT(11)	ju_ru_walclock	INTEGER
CpuDuration	—	—	ju_ru_utime	DOUBLE
			ju_ru_stime	DOUBLE
EndTime	g_end_time	INT(11)	ju_end_time	TIMESTAMP
StartTime	—	—	ju_start_time	TIMESTAMP
MachineName	—	—	—	—
Host	g_machine	VARCHAR(255)	ju_hostname	TEXT
SubmitHost	—	—	—	—
Queue	g_queue	VARCHAR(255)	ju_qname	TEXT
ProjectName	g_project	VARCHAR(255)	j_project	TEXT
Network	—	—	ju_ru_io	DOUBLE
Disk	—	—	ju_mem	DOUBLE
			Ju_maxvmem	DOUBLE
Swap	—	—	ju_ru_nswap	INTEGER

NodeCount	g_nodes	INT(11)	—	—
Processors	g_processors	INT(11)	—	—
TimeDuration	—	—	—	—
TimeInstant	g_submission_time	INT(11)	J_submission_time	TIMESTAMP
	g_creation_time	INT(11)		
	g_modification_time	INT(11)		
ServiceLevel	g_quality_of_service	VARCHAR(255)	—	—

C.2 Custom OGF-UR Extensions

Table C-3: Metric Extensions from OGF-UR

OGF-UR Extension	Gratia Extension	SGAS Extension	APEL and DGAS Extensions	Description
Resource	VOName	VOName	—	The VO Name to which the usage record of the job belongs
	ReportableVOName	—	—	The VO Name that is actually used when reporting the usage record
	NJobs	—	—	The number of jobs recorded in the usage records. The property is specific to aggregate accounting
	ProbeName	—	—	The probe identity that metered the usage record
	SiteName	—	ExecutingSite	The site name on which the job recorded is executed
ConsumableResource	—	—	—	OGF extension for additional measured resource associated with the usage record
PhaseResource			ElapsedTimeSeconds	Wall Duration in seconds
			BaseCpuTimeSeconds	CPU Duration in seconds
	—	—	SpecInt2000	The GLUE host benchmark Sloo
	—	—	SpecFloat2000	The glue host benchmark SFoo
VolumeResource	—	—	—	OGF extension for additional measured storage resource associated with the usage record

C.3 APEL Accounting System

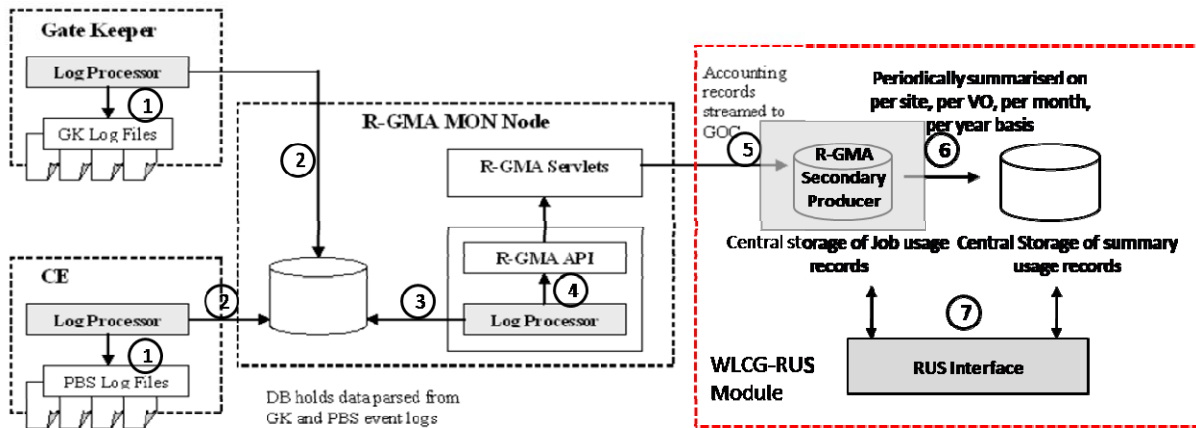


Figure C-1: EGEE APEL Accounting Workflow

- ① APEL interprets gatekeeper and batch system logs to produce accounting data at a site. The CPU processor of APEL normalises site-specific CPU duration according to CPU performance information queried from an information service.
- ② Metered accounting data are streamed into a local database for further processing.
- ③ The APEL join processor then enforces the accounting data at local sites and produces complete accounting records in well-formed format.
- ④ The join processor calls R-GMA API's primary producer service.
- ⑤ The accounting records are then streamed into a centralised accounting storage at GOC.
- ⑥ These job accounting records are periodically aggregated and inserted into secondary database tables on per VO, per site, and per month bases.
- ⑦ The WLCG-RUS module is intended to provide querying of the aggregated accounting usage, and possibly job-level records, and also allow the population of summary accounting records from other accounting systems (SGAS, DGAS-RUS, and Gratia).

C.4 DGAS Accounting System

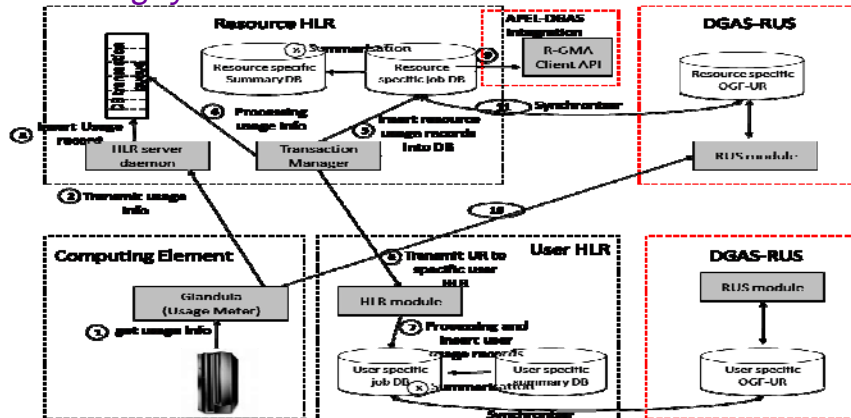


Figure C-2: EGEE DGAS Accounting Workflow

* Preliminary design plan

- ① A lightweight sensor, Gianduia, meters job usage information on completion of a job.
- ② The batch-system job usage information is posted to a specific resource HLR server where the resource account is registered.
- ③ The HLR daemon listens for the incoming message and puts the received usage information into a buffer queue for further processing.
- ④ A transaction manager module in the HLR server then fetches the job usage information from the buffer and processes it into an accounting record.
- ⑤ The accounting record then is inserted into a relational database in the HLR server.
- ⑥ The transaction manager is also responsible for notifying the user's HLR for processing job usage information related to registered user account.
- ⑦ The user HLR processes and inserts the user accounting records into the local database.
- ⑧ The HLR server also maintains summary usage information for each user or resource account.
- ⑨ The APEL-DGAS integration is realised through the R-GMA module, exposing the HLR job usage database as a primary producer. Accounting records are thus sent to GOC database.
- ⑩* With the development of DGAS-RUS, an implementation of OGF-RUS service, the usage information can be directly inserted into a RUS database through the RUS upload interfaces.
- ⑪* A synchronizer process will be running at the backend to ensure synchronicity between the HLR and RUS databases.

C.5 SGAS Accounting System

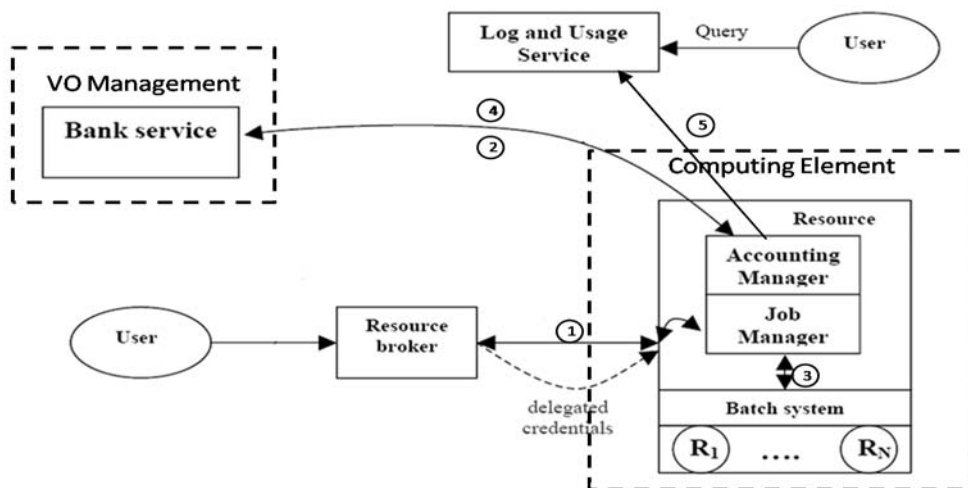


Figure C-3: SGAS Accounting Workflow

- ① A user submits a job to a resource broker designating a project/VO that should be charged.
- ② On receiving the job request, the job account reservation manager (JARM) computes the maximum cost based on the maximum runtime specified in the job submission. The JARM then contacts the banking service belonging to the given project/VO. A hold is placed against an account corresponding to the calculated cost.
- ③ The job request is forwarded into local batch job manager for execution.

④ On completion, JARM is triggered to collect usage information from the local system and convert it into OGF-UR format (XML documents). JARM also contacts the banking service to perform charging based on the actual usage.

⑤ Finally, the usage information is then inserted into a centralised storage facility (XML:DB) through the Log and Usage Tracking Service (LUTS), an implementation of the OGF-RUS service.

C.6 Gratia Accounting System

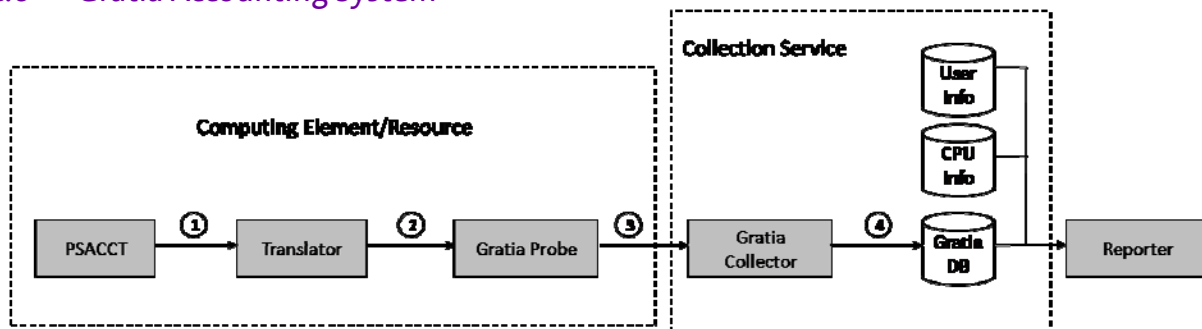


Figure C-4: Gratia Accounting Workflow

- ① The usage metering is based on PSACCT, a linux utility that logs resource usage daily.
- ② A translator script will be scheduled to run daily after PSACCT service and convert the usage information into well-formed OGF-UR records.
- ③ The gratia probe component running on each node pushes the OGF-UR records into the Gratia collector. Each probe is registered to a particular collector service for publishing usage records.
- ④ The collector is running as a standalone Web service that maintains usage records as well as related user and CPU information, which are currently administrated manually.

C.7 NGS RUS Accounting

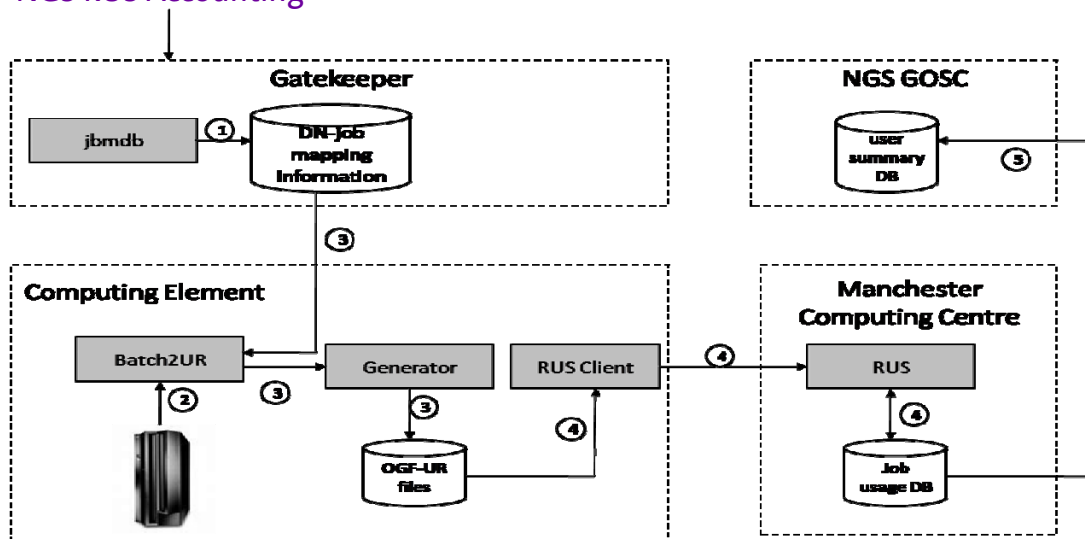


Figure C-5: NGS Accounting Workflow

- ① The “jbmdb” module generates a database of globus logs to capture the user/job mapping information
- ② The Batch2UR (formerly PBS2UR) scripts provide log parsing facilities for LSF and PBS batch systems. The extracted information is sent to the usage record generator with the globus user/job mappings.

- ③ The usage record generator formats the records into OGF-UR format and stores them into local file system.
- ④ Sites that provide accounting data are required to run a RUS client to populate usage records to a centralised job usage repository (relational database) through RUS.
- ⑤ The NGS GOSC at RAL periodically (daily) summarises and stores usage records on per user basis.

C.8 UNICORE RUS Accounting

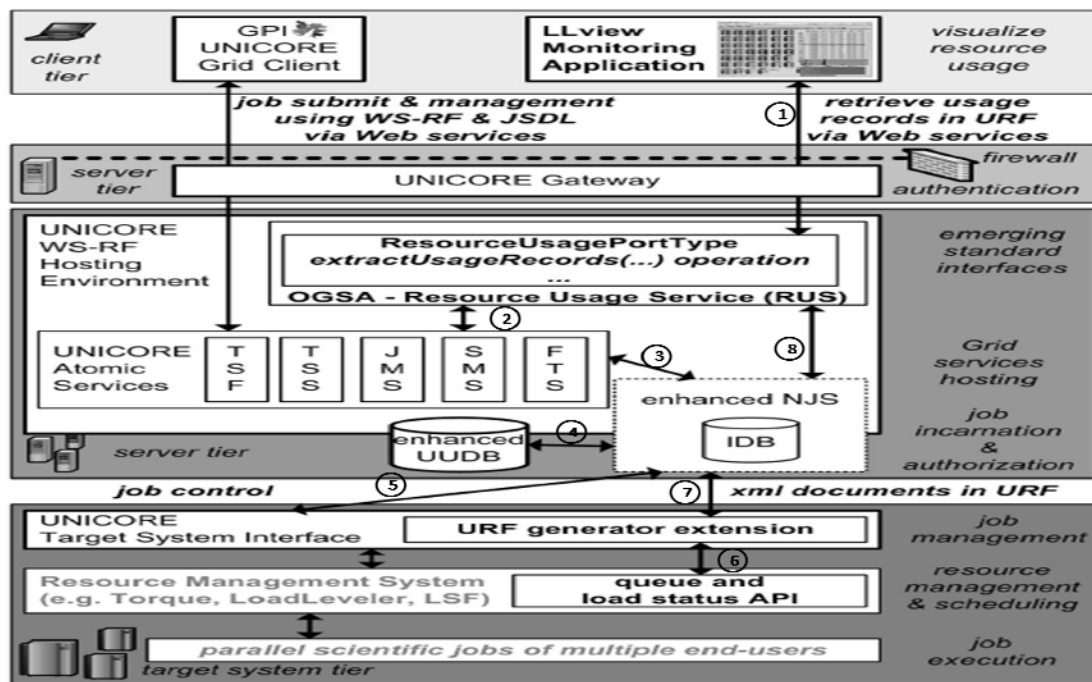


Figure C-6: UNICORE Accounting Service Workflow

- ① The RUS service is implemented as a WSRF-compatible service in the core service layer of UNICORE so that the client-side application is able to obtain up-to-date resource usage information of jobs executing on target local batch systems.
- ② As a high-level service on top of the UNICORE Atomic Service (UAS), the invocation of RUS interfaces creates a target system session.
- ③ Hence, the invocation of a RUS operation leads to the definition and submission of an internal job to the Network Job Supervisor (NJS).
- ④ The NJS authorises the RUS request by interrogating on UNICORE User Database (UUDB). After successful authorisation, the rather abstract job definition execution is translated into non-abstract job descriptions.
- ⑤ Finally the execution request is forwarded to a URF generator, an extension to the Target System Interfaces (TSI), rather than being allocated to local batch system.
- ⑥ The URF generator calls the batch system API to get queue and job status.
- ⑦ The usage information is then composed into well-formed OGF UR format.
- ⑧ The XML documents are returned to the clients through RUS.

D Assessment: Usage Monitoring Deployment and Client View

D.1 EGEE Accounting Portal

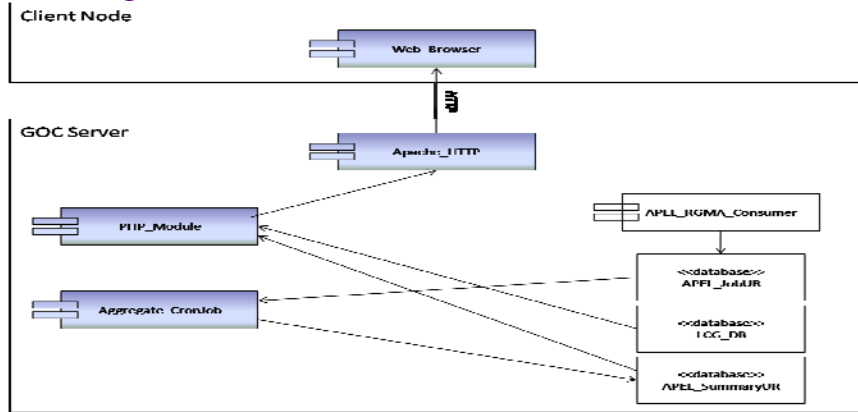


Figure D-1: EGEE Accounting Portal (Deployment Diagram)

Server-side modules:

- PHP_Module: the representation layer operates on summary usage records and the LCG_DB to provide a tree-view of LCG sites and VOs.
- Aggregate_CronJob: a lightweight script that periodically aggregates APEL job usage records to summary usage storage.
- Apache_HTTP: the third-party module listening for HTTP requests from clients;

Client-side modules:

This is no specific client-side modules provided by the APEL Web application. The client accesses the web portal via a URL.

The PHP web site provides a client-side GUI interface for users to specify query criteria, with the target sites organised in a hierarchical tree structure in the left navigation bar.

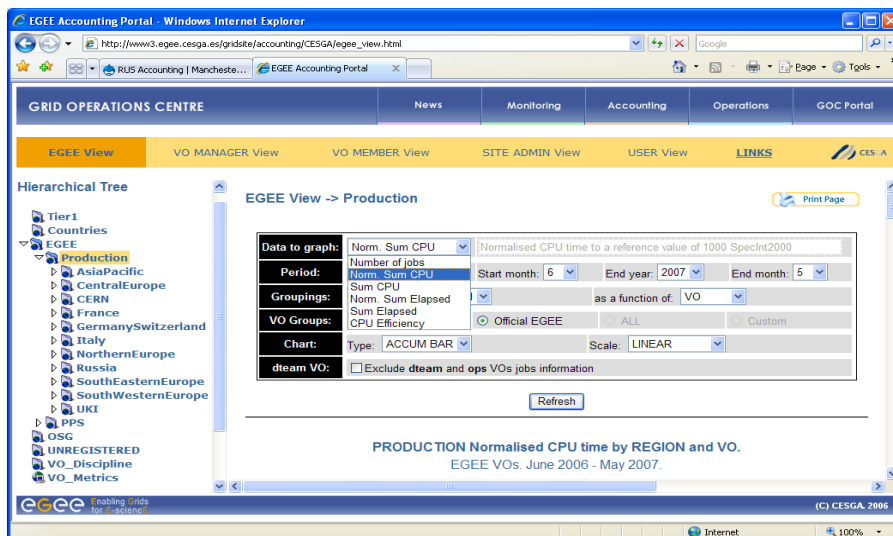


Figure D-2: EGEE CPU Usage Accounting Portal Snapshot (User Interface)

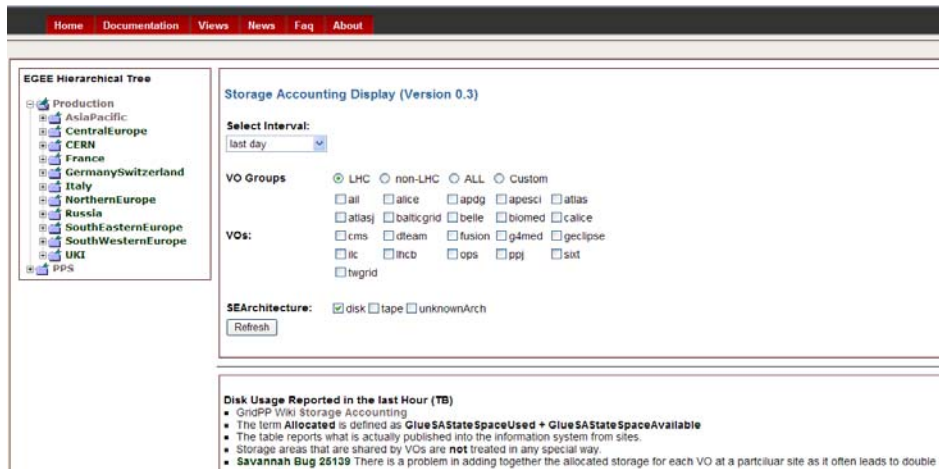


Figure D-3: EGEE Storage Usage Accounting Portal Snapshot (User Interface)

The results of a query can be displayed graphically or in table form. The snapshot example shows a query of CPU usage by VO at the Brunel site.

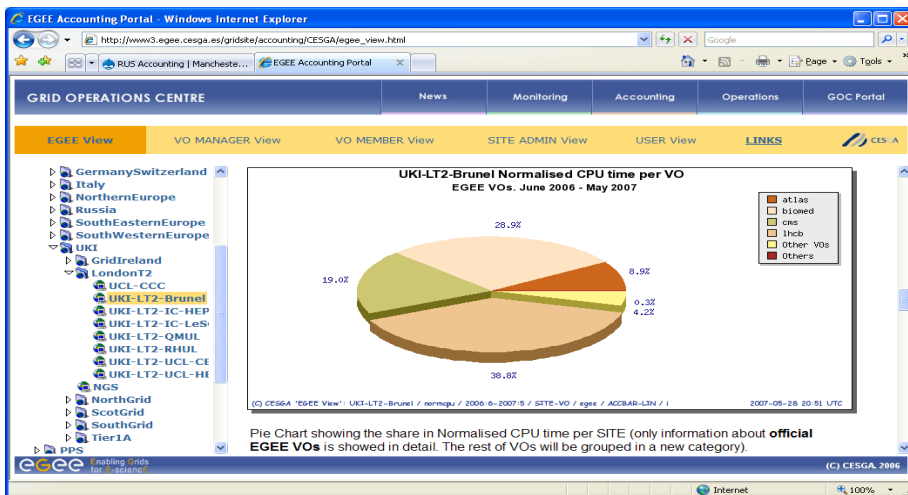


Figure D-4: EGEE Snapshot of CPU Usage Statistics of London Tier 2 Site in Brunel

D.2 HLR

As one of the main components of DGAS, the HLR is deployed as a service endpoint where a client communicates a request using XML-based protocol to the HLR server. The current HLR server is composed of a set of modules for user authentication/authorisation, usage record population and usage monitoring. These modules are loaded on demand by HLR server daemon to provide accounting and usage monitoring facilities. The HLR deployment diagram below (figure D-5) highlights four modules relating to usage monitoring.

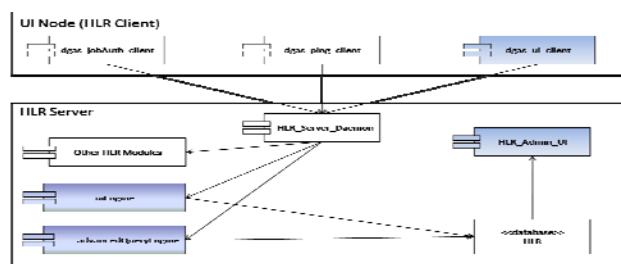


Figure D-5: HLR Usage Monitoring Deployment UML Diagram

Server-side modules:

- **uiEngine:** the server backend module for the simple user interface requests. It allows a remote user interface to retrieve information about account status, transactions (usage records) and so on.
- **advancedQueryEngine:** advanced user interface module on the server side. It allows more complex queries to the underlying database.
- **HLR_Admin_UI:** the server-side API that allows the HLR server manager to execute local data management on the local HLR server.

Client-side module:

- **dgas_ui_client:** the client-side user interface of the HLR service that forwards usage monitoring queries on behalf of users to uiEngine/advancedQueryEngine through the HLR server daemon.

Table D-1: HLR Usage Monitoring Functional Specification

HLR Usage Monitoring API	Function Description	Target HLR Module
<i>glite_dgas_hlrUserInfoClient</i>	Used to query an HLR server for information about a specific user or group of users. The output of this command describes the total CPU usage (CPU/Wallclock duration) with the number of jobs belonging to specific user or a group of users. The command is specific to the user HLR server.	uiEngine
<i>glite_dgas_hlrResourceInfoClient</i>	Used to query an HLR server for information about a specific resource. The output of this command describes the total CPU usage (CPU/Wallclock duration) with the number of jobs executed on a specific resource. This command is specific to the resource HLR server.	uiEngine
<i>glite_dgas_hlrAdvancedQueryClient</i>	Used to retrieve lists of transactions as well as aggregated information over users, resources, account groups and VOs. <ul style="list-style-type: none"> • Aggregate usage records for a user within a specified period; • Aggregate usage records for a resource within a specified period; • Aggregate usage records for a group of users or resource accounts within a specified period; • Aggregate usage records for the users belonging to a VO within a specified period; • List of job usage records by a user within a specified period; • List of job usage records executed by a resource within a specified period; 	advancedQueryEngine

The following example gives a client-side view following the execution of a *glite_dgas_hlrAdvancedQueryClient* command on the UI machine to obtain aggregate usage information of jobs belonging to a particular user over specified period of time. The result is displayed on per day basis.

```

>$GLITE_LOCATION/bin/glite_dgas_hlrAdvancedQueryClient \
-H "hlr02.to.infn.it:56568:"-QuserAggregate \
-u "/C=IT/O=INFN/OU=Personal Certificate/L=Torino/CN=Andrea Guarise/Email=Andrea.Guarise@to.infn.it" \
-t 1117164757-1118164757-F day
+
+ date | jobs | totCpuTime | totWallTime | totMem(MB) | totVMem(MB) |
+ 27/5/2005-28/5/2005 | 0 | 0 | 0 | 0 | 0 |
+ date | jobs | totCpuTime | totWallTime | totMem(MB) | totVMem(MB) |
+ 28/5/2005-29/5/2005| 0 | 0 | 10 | 0 | 0 |
+ date | jobs | totCpuTime | totWallTime | totMem(MB) | totVMem(MB) |
+ 29/5/2005-30/5/2005| 0 | 0 | 10 | 0 | 0 |
+ date | jobs | totCpuTime | totWallTime | totMem(MB) | totVMem(MB) |
+ 30/5/2005-31/5/2005| 61 | 779 | 1041 | 109 | 415 |
+ date | jobs | totCpuTime | totWallTime | totMem(MB) | totVMem(MB) |
+ 31/5/2005-1/6/2005 | 0 | 933 | 1222 | 176 | 686 |
+ date | jobs | totCpuTime | totWallTime | totMem(MB) | totVMem(MB) |
+ 1/6/2005-2/6/2005| 177 | 2317 | 3108 | 398 | 1608 |
+ date | jobs | totCpuTime | totWallTime | totMem(MB) | totVMem(MB) |
+ 2/6/2005 | 363 | 4850 | 6079 | 865 | 3417 |
    
```

Figure D-6: HLR Command Client View Snapshot

SGAS Reporting Framework

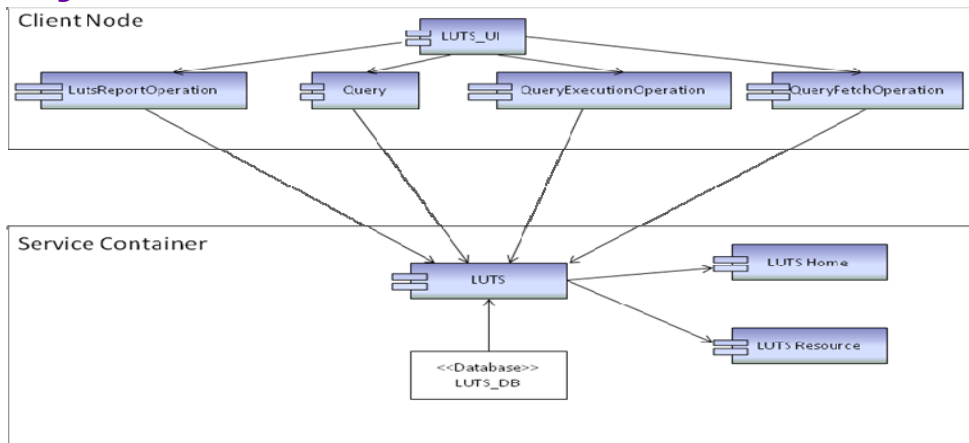


Figure D-7: SGAS Report Framework (Deployment Diagram)

Server-side modules:

- LUTS: a module deployed at the server side as a standalone RUS implementation that accepts job usage records from SGAS JARM into a centralised usage data storage, the LUTS_DB (XML:DB).

Client-side modules:

- LutsReportOperation: one of the components of the SGAS reporting framework that enables the querying of reports from LUTS and to publishes them to external sources such as Web servers.
- Query: the query module executes a query against the LUTS service.
- QueryExecutionOperation: a module that contributes to a cursor-style query against the LUTS service.
- QueryFetchOperation: a module that allowz clients to fetch query results.
- LUTS_UI: a set of command interfaces corresponding to client operation modules.

Table D-2: Functional sepcifcaiton (SGAS LUTS)

Client API	Functional Description	Target Module
<i>Luts-query</i>	Executes a query against the LUTS. The results must be fetched through the "query fetch" operation	Query
<i>luts-report</i>	Generates LUTS reports	LutsReportOperation

<i>query-execute</i>	Executes a cursor-based query	QueryExecutionOperation
<i>Query-fetch</i>	Fetch part of a query result.	QueryFetchOperation

The following snapshot (figure D-8) gives an example of LUTS report operation that generates HTML reports based on a user query.

LUTS Report

Another sample report script is available in \$SGAS_LOCATION/etc/sgas-luts/luts-report.sh. It queries LUTS service listed in \$SGAS_LOCATION/etc/sgas-luts/luts-services with full line-separate service URLs. The XPath expressions are in \$SGAS_LOCATION/etc/sgas-luts/luts-xpaths and the stylesheet used is in \$SGAS_LOCATION/etc/sgas-luts/luts-report.xsl.

Sample LUTS report output:

RecordIdentity rec1

GlobalJobId:	globid1
LocalJobId:	locald1
Charge:	10
Status:	Completed
GlobalUserId:	thomasglob
LocalUserId:	thomaslocal
WallDuration:	10
NodeCount:	2
StartTime:	Mon 20:20
StartTimeEpoch:	1110305525
ProjectName:	http://proj1
SGASProjectName:	test
SubmitHost:	subhost1
Queue:	queue1

Figure D-8: HTML Snapshot (Generated by SGAS Report Framework)

Gratia Reporter

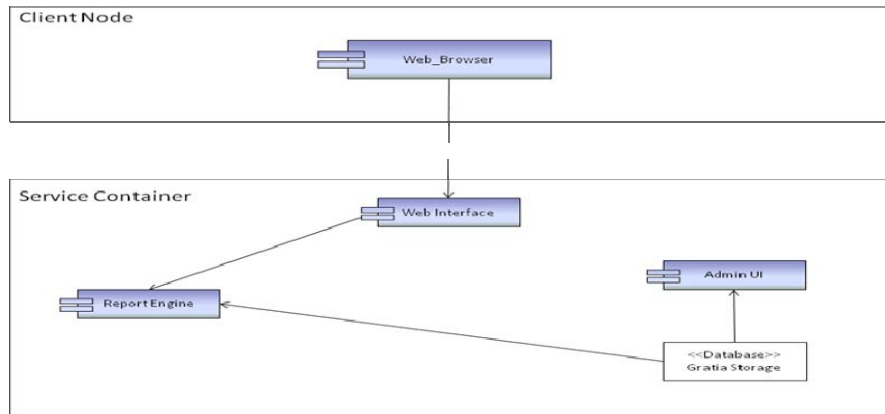


Figure D-9: Gratia Reporter (Deployment Diagram)

The current reporting system of Gratia is deployed as a standalone Web application based on the BIRT reporting engine, which plots accounting data in chart form.

Server-side modules:

- ReportEngine: the Business Intelligence and Reporting Tools (BIRT) creates dynamic database-driven reports with a full range of formatting capabilities.
- Web Interface: The web interface is composed of a set of JSP pages that accepts user requests and contacts the Report Engine to produce dynamic reports.
- AdminUI: the admin UI is used to provide data management facilities for advanced users to manage CPU information, User Information and Usage Records.

Client-side modules:

This is no specific client-side modules required to access the Gratia reporting system. The client can access a web front end by URL.

The current implementation of Gratia reporting system does not provide access control and authorisation facilities but simply providing public-view usage statistics on VO, Site, and probe basis. The Web interfaces (table are implemented as a set of JSP Web pages that invokes the BIRT reporting engine to interrogate Gratia database for usage reporting.

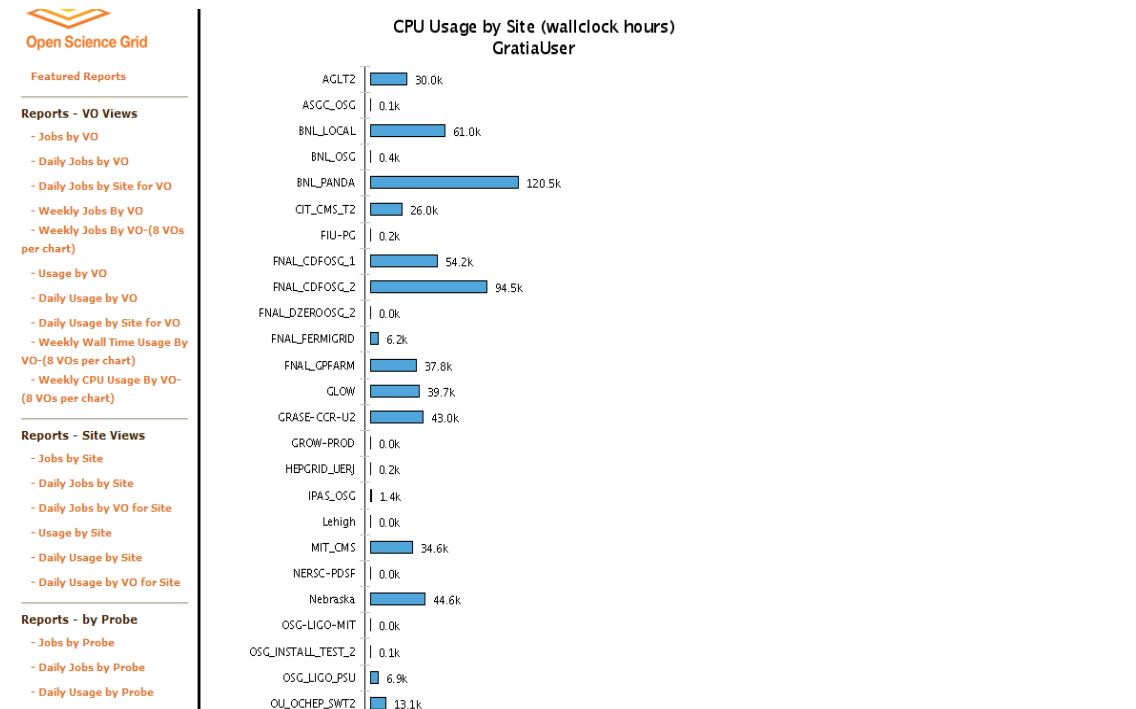


Figure D-10: Gratia Reporting Snapshot (Client View)

D.3 UAS

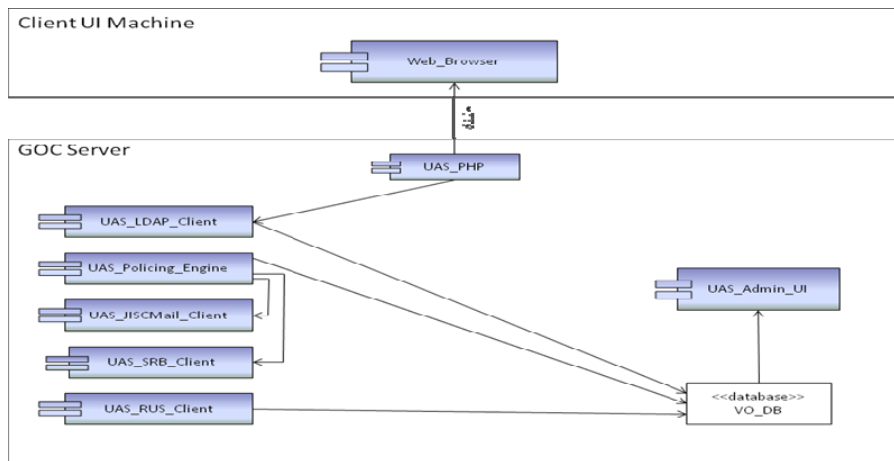


Figure D-11: User Accounting System (Deployment Diagram)

The UAS modules are built on other components of the NGS system, including the NGS Resource Usage Service, VO management, Storage Resource Broker (SRB) and JISC mail management. It is designed provide users with a single interface for usage monitoring and user and project management.

Server-side Modules:

- UAS_PHP: this module provides a basic Web interface for users to view CPU usage information relating to his or her account.
- UAS_LDAP_Client: this module contacts the VO database to get usage information of the user through LDAP protocol.
- UAS_Policing_Engine: this module provides tools for automatic policing of accounts. This policing engine monitors usage information and compares it to allocations.
- UAS_JISCMail_Client: this module is used to invoke a JISC mail management service. When a new user is registered and activated, the JISC mail management service is invoked to notify user account status. A warning mail is forwarded to a user if a user's account goes over 90% of quota for CPU usage. If a user exceeds the quota he/she is emailed with notification that the account is locked (the user's DN is removed from the LDAP server).
- UAS_RUS_Client: the RUS client contacts the RUS service to query the CPU/disk usage for each user and updates the usage information stored within the VO database.
- UAS_SRB_Client: the UAS SRB client contacts the SRB server for user account management. When a new user is activated, the SRB client creates a new user in the SRB server. When the user's usage exceeds 100% of allocated quota, the user's account is removed from SRB server.

Client-side Module

The user accesses the PHP website via certificate authentication. User specified querying is not currently available.

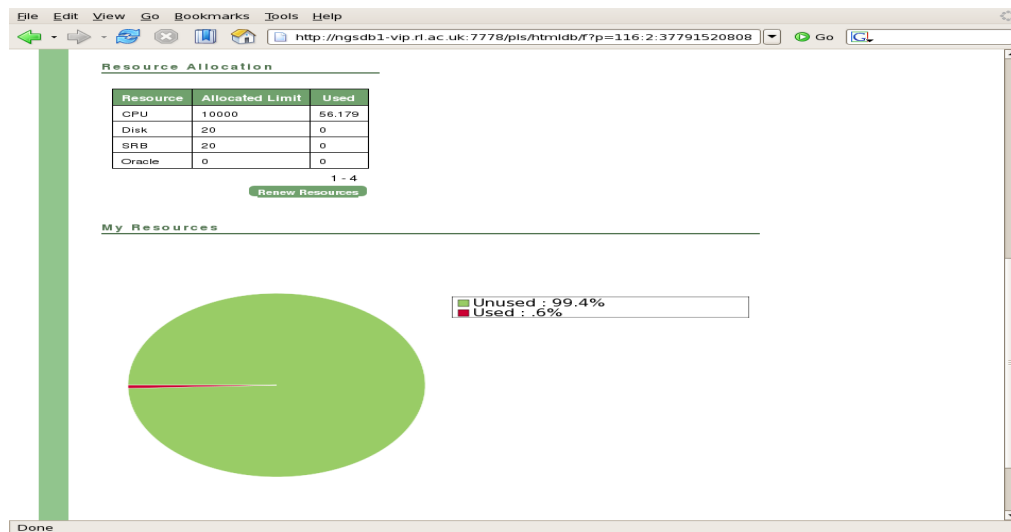


Figure D-12: UAS Snapshot (Client View)

D.4 LLview

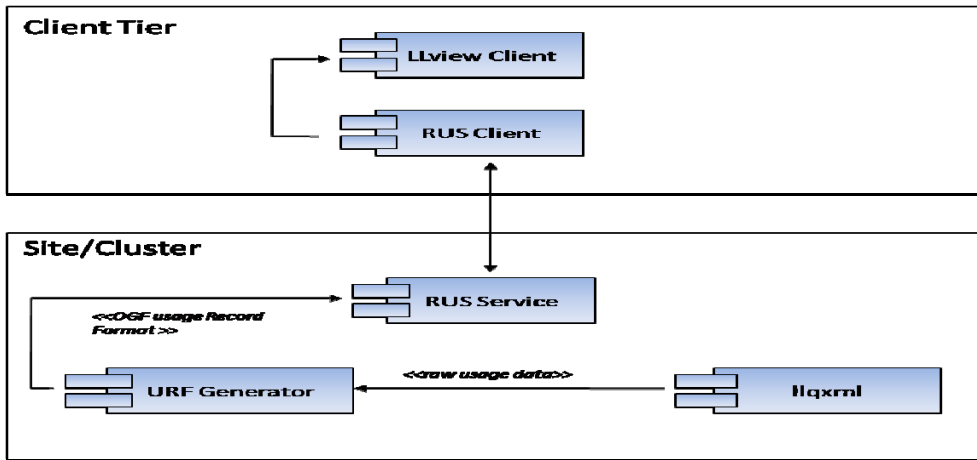


Figure D-13: LLview (Deployment Diagram)

Server-side Module

- RUS Service: WSRF-compatible RUS implementation
- URF Generator: the component used to generate OGF-UR records
- llqxml: a C program that is built on the LoadLeveler API to provide information about running and waiting jobs. The information is stored in XML format.

Client-side Module

- LLview Application Client: The core component of the LLview client is the node display, which colours nodes according to running jobs and displays additional information about the node such as status, memory usage and cpu load.
- RUS Client: the client-side module of the RUS service.

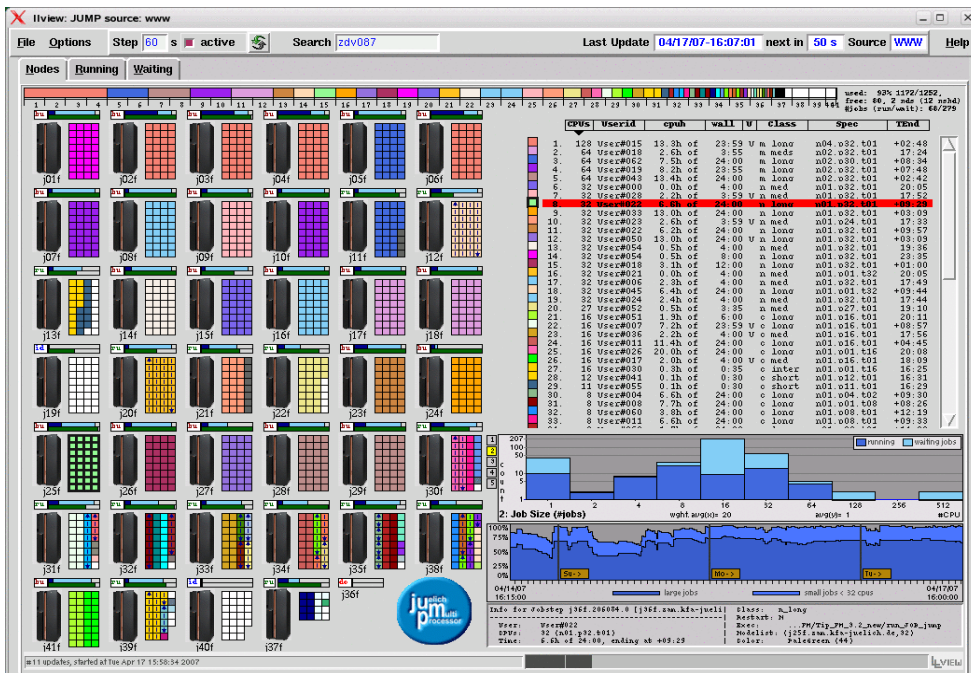


Figure D-14: LLview Application Snapshot (Client View)

Review of Accounting and Usage Monitoring

Final Report

Copyright (2007) The University of Manchester
Manchester Computing
The University of Manchester
Manchester M13 9PL
United Kingdom

<http://www.manchester.ac.uk/manchestercomputing>