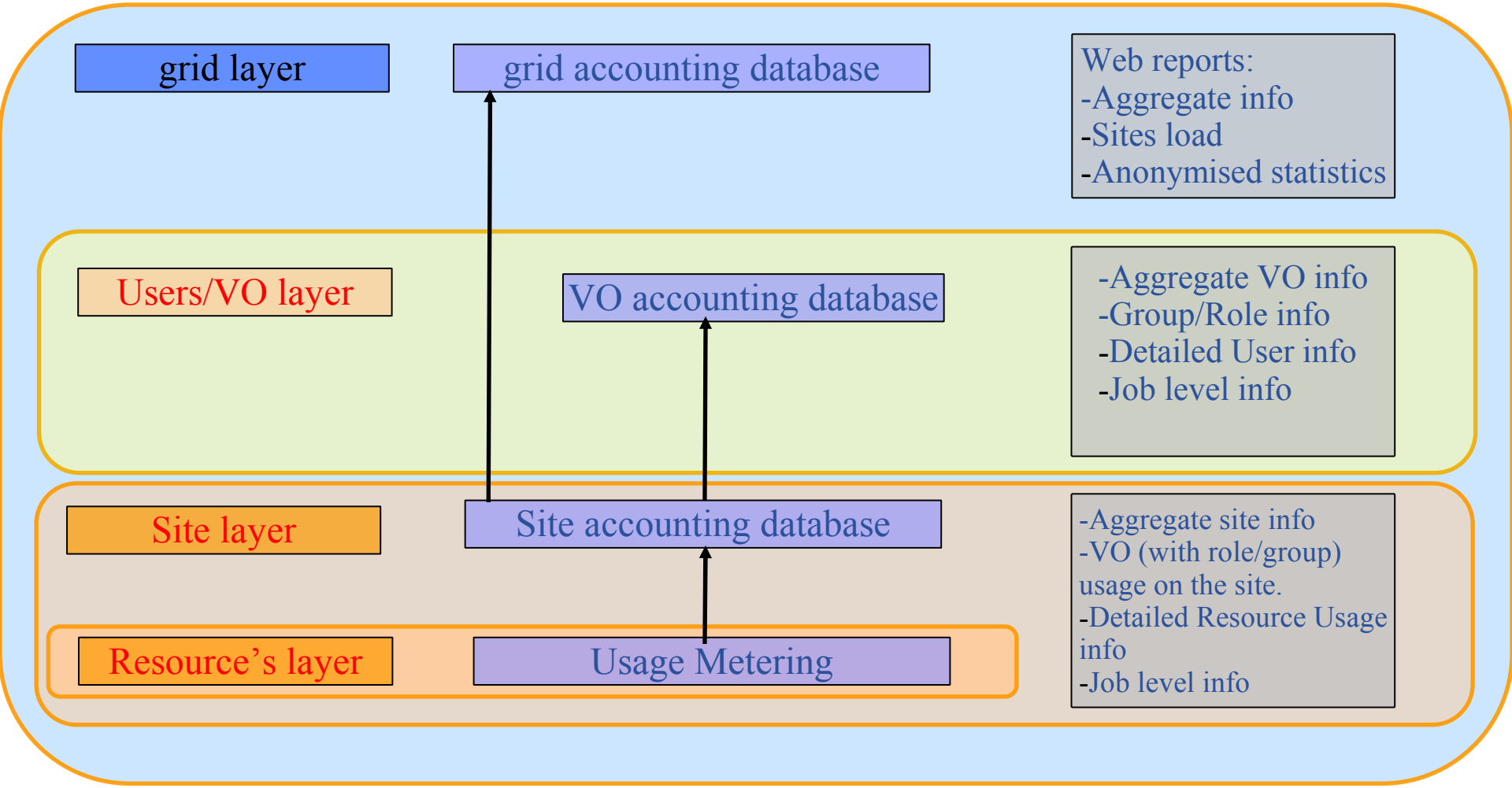
# APEL & DGAS
# A vertical infrastructure for Grid Accounting

**Rosario Piro & Dave Kant**

JRA1 All-Hands Meeting, CERN, March 23rd, 2006.

# A Grid accounting infrastructure

**A generic grid accounting infrastructure should cover the requirements of tree type of customers: Sites, VOs and their users, grid administrators.**
**While fulfilling the requirements it must also be secure, flexible and scalable.**



# Requirements / Use cases

**How much resource has been provided to each VO?**

Grid Level (High Level)
How much resource has been provided to each VO
Grid Level (High Level) Aggregation / Data Consolidation
Administrative Domains :-
VOs, VOMS Groups, Countries (LHC), Regions (EGEE)
Grids, (EGEE, OSG, TeraGrid, Nordic)
Organisations (GridPP, INFNGrid, SEEGrid, …)

**If 10,000 CPU hours were consumed by Atlas VO, who are the users that submitted the work?**

Finer Granularity (Low level)
User Level, Job Level, Top Users within a VO and Group
Confidentiality

**Quota enforcement / fairshare scheduling: how much resource has been provided to a particular user on a particular set of CEs?**

User Level, Job Level
Confidentiality
Reliability

# Accounting & policies

The accounting information can be also used to implement dynamic resource access policies and help in fair sharing of available resources.

It is clear that in order to reach this goal is necessary to have real time access to accounting information.

It is also necessary to be sure that the information contained in the accounting databases can't be maliciously altered by the users (reliability).

# APEL features

**Three Components:**
**Sensors (Deployed at site) :-**

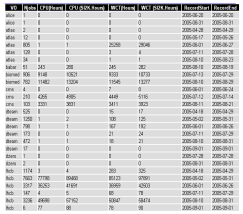
- Process log files; maps DN to Batch Usage; builds accounting records
- Accounts from Grid Jobs Only
- Supports PBS, SunGridEngine, Condor, and LSF
- Not REAL-TIME accounting

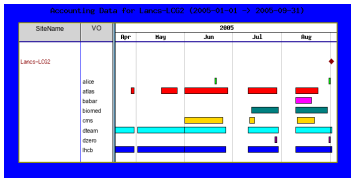
**Data Transport:-**

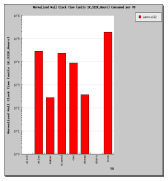
- Uses RGMA to send data to a central repository
- 196 sites publishing, 7.5 Million Job records collected
- Can use other transport protocols
- Allows sites to control DN information from leaving site

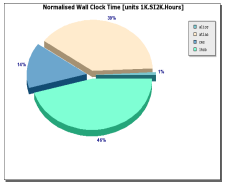
**Presentation (GOC and Regional Portal)**

- Reporting based on data aggregation
- LHC View, EGEE View, GridPP View, Site View
- Metrics (e.g. Time Integrated CPU Usage)

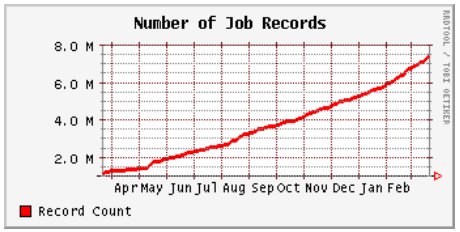




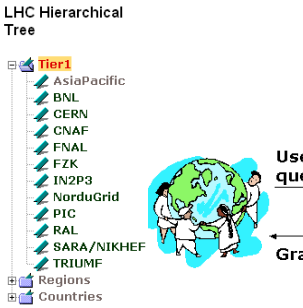




Tables, Pies, Gantt Charts,



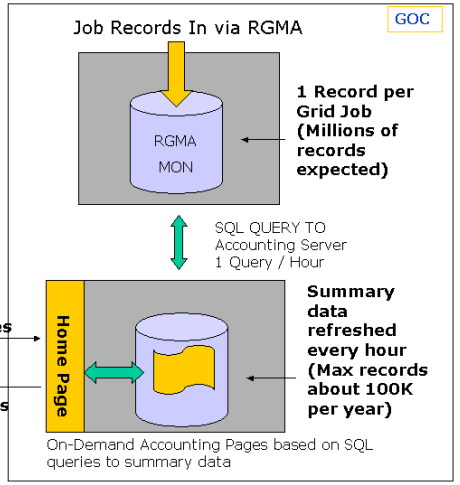

**Consolidation of Data**






User queries

Graphs



# APEL Issues

**Data Privacy and Confidentiality**

- **Provides sites with the option to publish or suppress (default) DN information**
  - Many sites suppress DN, some sites publish DN.
- **Any person with a trusted certificate and access to a UI can query RGMA and get Job Record information from APEL database**
  - x509-based authorization features in RGMA in development
- **Does not currently encrypt the usage record**
- **Does not currently sign the data**

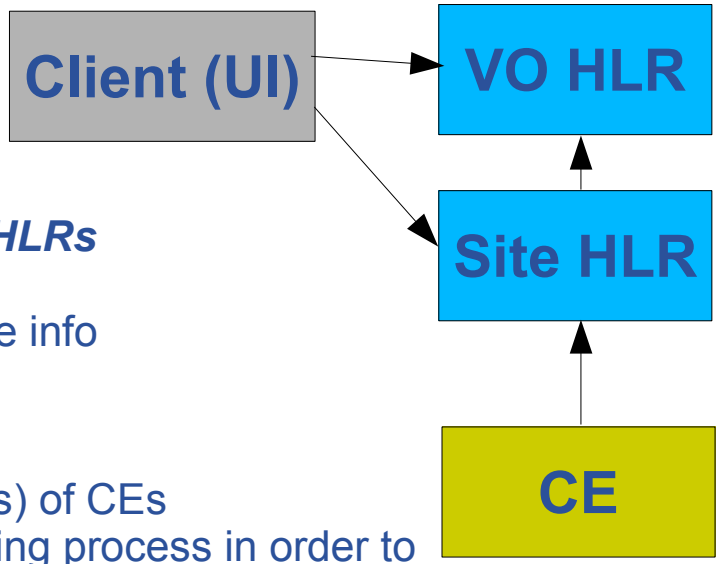
**General Issue For Accounting: Normalisation in Heterogeneous Farms**

- **APEL Uses SpecINT2000 published by sites in the information system because sites have different strategies/policies for setting up their batch systems**
  - Some sites publish SI2K of slowest worker nodes
  - Other sites publish a weighted mean, because they use internal scaling factors
  - Many Batch systems currently in use don't report performance/benchmark information of the worker nodes in the batch logs.

# DGAS features

**Gianduia (sensors deployed at site)**

- can build accounting records for *grid jobs* as well as *local jobs* (accounted on different HLR accounts)
- for grid jobs: integrates *local usage information* with *grid-related information* (user DN, user FQAN (VOMS), CE ID, grid job ID).
- supports PBS and LSF (work on SGE and Condor)
- ***"next-to-real-time" accounting*** (after job completion)




**HLR service (accounting servers)**

- ***fully distributed*** architecture (scalability)
- deployed as ***User (VO) HLRs*** and/or ***Resource (Site) HLRs***
- command line tools and C++ API for queries
- strict ***authorization control*** (x509) for accessing private info
- on Resource/Site HLRs: DGAS2APEL

**(optional) PA service (resource pricing servers)**

- for manually or dynamically setting prices (virtual credits) of CEs
- price information might be included in the RB's scheduling process in order to establish a grid resource market (may help in balancing demand and supply)

**Data transport:**

- all communication between the components as well as between client tools and servers is ***encrypted*** and ***secure*** (x509-based, GSI)
- mostly ***asynchronous*** data transport between the components for more robustness

# DGAS Issues

**Distributed architecture:**

- *advantage*: more scalable, more robust (no single point-of-failure)
- *disadvantage*: comparison of data from different VOs and different sites requires querying multiple HLR servers (as well as being authorized to access them)

==> ***suitable for reliable low-level accounting (e.g. for quota enforcement, charging of users, statistics for single VOs / sites, etc.), but high-level aggregate accounting and grid-wide statistics are difficult to obtain***

**Interface:**

- *command line interface* and *C++ API* that can be used by users, VO admins and site admins to retrieve their accouting information
- some customizable graphical output (gnuplot), but *no flexible graphical web interface* for easy comparison of aggregate accounting information (such as the APEL website at GOC)

==> ***suitable for integration with other components (through scripts and API calls) or retrieving fine grained and detailed information about single jobs, not suitable for the end user that needs a graphical presentation of accounting data***

# APEL + DGAS

**When used together APEL and DGAS can leverage each others strengths and give to their customers the following benefits:**

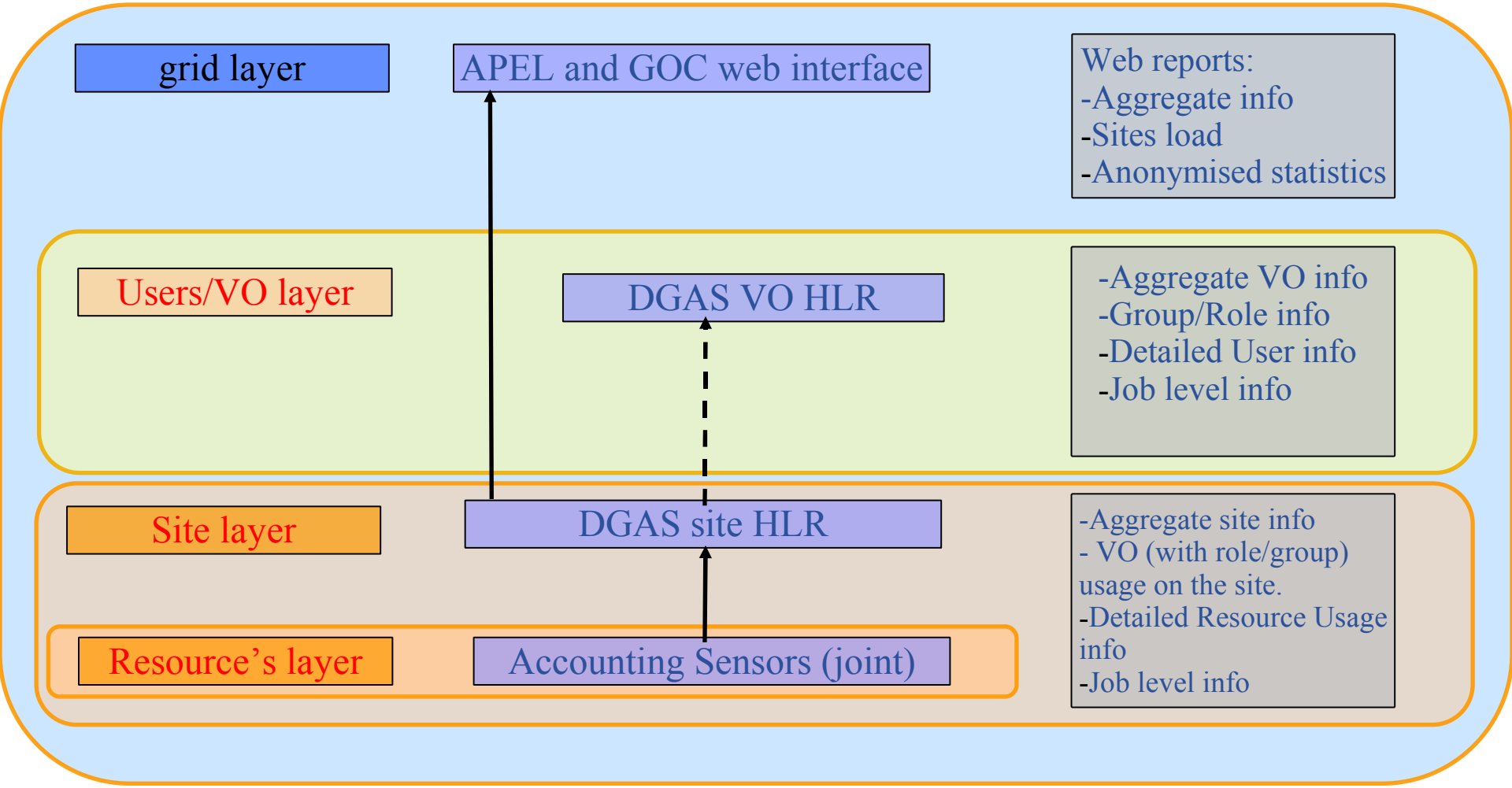
- Fully distributed architecture
- From single job to grid-level accounting information.
- Web access to aggregate usage information.
- VOMS based accounting (using VOMS FQAN in usage records).
- Real time access to job accounting info for users and site managers.
- Security and authorisation mechanisms to guarantee information integrity and confidentiality.
- Possibility to access non-confidential information via RGMA.

**APEL and DGAS, if used alone, have their own specific strengths and limitations.**

**None of the two can fulfill the previously highlighted requirements for grid accounting.**

**However, APEL and DGAS have complementary functionalities that if used in sinergy actually can fulfill the requirements.**

## DGAS and APEL as components of the vertical accounting infrastructure (DGAS2APEL example)



# Coming soon in gLite . . .

**Accounting information taken from:**

- **LRMS log files for usage metrics (CPU time, wall clock time, ...)**
- **New log file on the CEs**
  - For mapping grid-related information to the local job ID
  - Independent of submission procedure (WMS or not ...)
  - No services or clients required on the WN
  - Format (one line per job, one file per day):

    *"timestamp=<submission time to LRMS>"*
    *"userDN=<user's DN>"*
    *"userFQAN=<user's FQAN>"*
    *"ceID=<CE ID>" "jobID=<grid job ID>"*
    *"lrmsID=<LRMS job ID>" "localUser=<uid>"*
  - Notes:
    - userFQAN may appear multiple times
    - key-value pairs can appear in any order
    - some fields may be empty (e.g. userFQAN if not a VOMS proxy)
  - Already implemented for BLAH (and CREAM)
    (work in progress for LCG)
  - Sensors: work in progress (DGAS in testing phase, APEL in development)

# APEL + DGAS deployment issues

**Differences between APEL and DGAS sensors (on the CEs):**

- ***APEL*:**
  - parses log files once a day (running as a cron job)
  - publishes job records via RGMA to the GOC database
- ***DGAS:***
  - frequently (configurable) reads new entries in log files (running as a daemon)
  - GSI-based transmission of job records to the DGAS Site HLR server (and from there optionally to the VO HLR server)

**Which accounting sensors on the CEs? Three possibilities:**

- ***One common sensor for both systems:***
  - Advantage: shared development effort; better batch system support
  - Disadvantage: different requirements (e.g. real-time accounting)
- ***One system as client of the other*:**

  One system forwards accounting records to the other (DGAS2APEL + RGMA?); already prototyped.
  - Advantage: no redundant processing of LRMS/CE log files; reduced processing load
  - Disadvantage: in case of failures, both systems are affected.
- ***Two distinct sensors on the CE:***
  - Advantage: control of completeness of accounting information (consistency check between the systems)
  - Disadvantage: double processing of log files, more CPU load

# Discussion

**Proposal for deployment:**

- 1) deploy both sensors for smooth transition (also useful for consistency checks and more experience)
- 2) use DGAS2APEL (work in progress, easiest solution towards a single-sensor-approach)
- 3) eventually develop a common sensor (in the long-term)

**Start seriously discussing normalisation of accounting data:**

- a common *agreement on normalisation procedures* is required (for example: how to compute SpecInt/Float benchmarks? Slowest CPU? Mean value?)
- think about having more *fine grained performance information* published: SpecInt/Float for each WN, not only for the entire CE? At least min., mean and max. value for the CE?
- accounting systems *should* be provided with *raw, non-normalised* accounting data, such that normalisation procedures may be altered in the future without loss of information (e.g. PBS allows for CPU time scaling, LSF doesn't).
- **CPU time**: should be normalised according to ***performance per CPU***
- **Wall clock time**: should be normalised according to ***performance per utilised job slot***

**Accurate normalisation is necessary for:**

- charging users (not an issue now, in the future?)
- compare usage across different sites, VOs, countries, ...
- partnerships with industry

# Future issues

**Storage Accounting:**

- needed soon ...
- what exactly should be accounted? storage itself, access to data stored by others, ...

**Accounting of Generic Services:**

- account for provided grid services or provided software/licenses? ...
- not an issue now.

**Standardization:**

- *GGF Usage Record* format
  - very batch job specific ... (what about storage? generic services?)
- *GGF Resource Usage Service (RUS)*
  - Web Service-based interface for exchanging *GGF Usage Records*
  - Interoperability with other Grids (OSG, TeraGrid)

## APEL? DGAS? APEL+DGAS!

**For more information:**

- **APEL accounting website:**
  *http://goc.grid-support.ac.uk/gridsite/accounting/*
- **APEL User Guide:**
  *http://goc.grid-support.ac.uk/gridsite/accounting/glite-apel-rpms/apel-user-guide.pdf*

- **DGAS website:**
  *http://www.to.infn.it/grid/accounting/*
- **DGAS User Guides:**
  *https://edms.cern.ch/document/571271/1*