



CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

SA4 Midterm Activity Report

Deliverable ID: DEISA-DSA4-4.2
Due date: October 31, 2006
Actual delivery date: November 24, 2006
Lead contractor for this deliverable: IDRIS-CNRS, France

Project start date: May 1st, 2004
Duration: 4 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Content

Table of Content.....	1
1. Executive Summary.....	2
2. Introduction.....	3
3. DEISA Common Production Environment.....	4
4. Software monitoring for the DEISA Common Production Environment.....	5
5. Support of scientific Joint Research Activities.....	7
5.1 JRA 1 (Material Sciences).....	7
5.2 JRA 2 (Cosmology Applications).....	7
5.3 JRA 3 (Fusion Research).....	8
5.4 JRA 4 (Life Sciences).....	8
5.5 JRA 5 (Industrial CFD).....	8
5.6 JRA 6 (Coupled Applications).....	9
6. DEISA Training Sessions.....	10
7. Other activities.....	11
7.1 Trouble Ticket System.....	11
7.2 DEISA Cluster Resource Management Package.....	11
7.3 DEISA Applications Test Suite.....	12
7.4 DEISA Test Users.....	12
7.5 User documentation.....	12
8. References and Applicable Documents.....	13
9. List of Acronyms and Abbreviations.....	14

1. Executive Summary

The main objective of this service activity is to deploy all the actions required to allow scientific users to adopt and use the DEISA supercomputing infrastructure. This is mainly done by providing and maintaining a *Common Production Environment* on all the platforms of the infrastructure, and also providing documentation on its usage, the organization of training sessions in various European countries and a decentralised *Help Desk* service.

This document gives a detailed overview of all the tasks achieved or currently undertaken in the *User Support and Applications* Service Activity (SA4) during the reporting period (PM 25 to PM 30). This constitutes the report of the mid-term annual activities.

This document is publicly available.

2. Introduction

During the last six months, the *Applications and User Support Service* Activity has mainly concentrated its efforts on the following tasks:

- Maintenance of the *DEISA Common Production Environment* (DCPE).
- Preparation for the migration to the completely new version 2.0 of the INCA software, used to monitor all our software environment.
- Provision of help to the users of the scientific *Joint Research Activities*.
- Organization of the first two DEISA training sessions.
- Set up of the internal procedures to host a new category of users, called *DEISA Test Users* (DTU).
- Port and operation of the *DEISA Applications Test Suite* (DATS) on nearly all the remaining sites which did not yet run it.

3. DEISA Common Production Environment

The unified *DEISA Common Production Environment* (DCPE) is a major feature in DEISA, as in all distributed infrastructures of this kind, which defines a coherent set of software accessible on the various sites of the infrastructure. It offers both a common interface to the users, independent of the target platform really used, and the ability to migrate jobs between different supercomputers of the same architecture. Of course, the level of coherence is not the same everywhere in the infrastructure, ranging from very high inside each subgroup of homogeneous computers to a lower level across the other subgroups.

Three main components define the CPE:

- a coherent set of software packages divided into six categories: environment (which does not include any software, but defines the environment needed by a DEISA user), shells, compilers, libraries, tools and applications;
- a uniform interface to access the software, provided by the *Modules* software [18];
- a framework to monitor the software (this will be described in section 4).

As reported in the deliverables D-SA4-3.2 [6] and D-SA4-4 [7], this initial DCPE has already been installed on all the seven AIX sites, 18 months ago for the core sites and later for the other ones and the AIX environment was already complete for the seven related partners six months ago. During the last period, only some minor updates were done to it, according to the commonly agreed evolutions of the software packages. These updates were to both the software stack and the *Modules* environment, as required.

In parallel, the work needed to nearly achieve the installation and operation of the DCPE on the non AIX sites was done. The two sites LRZ and SARA, which both operate now an SGI Altix system (SARA since several years and LRZ since the end of the summer) have now a common SGI Altix DCPE, according to what they mutually agreed in the past to be included in their common software basis. They are now in pre or full production modes.

On its side, BSC has also managed to build its own software stack for their Linux/PowerPC machine (MareNostrum), as well as the adapted *Modules* interface for it. They also operate it today in full production mode.

As initially planned in the two layer infrastructure of DEISA, there is, as required, a very strong homogeneity of the software stack inside each common set of platforms (IBM AIX and SGI Altix), with only minor discrepancies on the patch level of the software included, and a weaker one across the different set of platforms. Nevertheless, there is a general coherence and many common items.

The CVS server which has been set up at IDRIS to distribute the configuration files, allowing each partner to find easily the changes made since their last update and to download each time only the modified files, continued to be heavily used.

Our main objective, now that it is used in production mode in all sites, is obviously to keep a stable framework, with only the changes required by the evolutions of the software stack. Nevertheless, we agreed to make, in around one year from now, a deep analysis of the behaviour of this framework to take benefit of all the accumulated experience, and to look at this moment of possible improvements in this framework for its usage in the future, in its design as well as in its implementation.

4. Software monitoring for the DEISA Common Production Environment

It is an absolutely necessity that each centre's staff have both an up to date overview and a detailed status of the software environment, especially in a distributed infrastructure like DEISA. The monitoring of such a distributed software environment has become a requirement in all important grid projects, even if the requirements are rather high and sometimes difficult to handle, especially in a heterogeneous context, meaning that the development of such a general framework is a huge task by itself.

The first feature required in such tools is the ability to verify the accessibility of the various software packages installed on the different computers, the expected version level of these packages and also, when possible, their current behaviour. It is also important that these tools allow the verification of the status after software upgrades and offer the administrators and the user support services an updated view of what is installed and available on the various computers of the distributed infrastructure, alerting them of possible deficient components.

As explained in the deliverable D-SA4-3 [5], the INCA tool (*Test Harness and Reporting Framework*) [17], developed by the San Diego Supercomputer Center and the Argonne National Laboratory for the TeraGrid project [20], has been chosen after an evaluation process. As described in the deliverables D-SA4-3.2 [6] and D-SA4-4 [7], this framework was initially deployed on the AIX core sites to monitor the availability of the software of the AIX DCPE with the required version levels (with what is called *version reporters*), then extended to the other sites in the first half of 2006. In parallel, another kind of sensors, called *unit reporters*, have been deployed and operated, only on the core sites during the preceding period, to check periodically the correct behaviour of the critical software components. They are based on the *DEISA Applications Test Suite* (DATS) – see the paragraph 7.3. In the last months, most of the sites which did not yet install the DATS and run the *unit reporters* have done so (still using the version 0.10 of INCA). This task is mostly achieved today and both *version* and *unit* reporters are now operated successfully in full production on the platforms.

Nevertheless, an important work was started since last February, the date of a dedicated workshop organized at SDSC, and heavily continued during the reporting period, to prepare the future migration to the new version 2.0 of INCA. As explained in the deliverable D-SA4-4 [7], SDSC has been working since the middle of 2005 on a completely new version of INCA rewritten from scratch, which will have many improvements and advantages, with an improved architecture, new functionalities and enhanced customization opportunities. The main changes are the following:

- introduction of a new *Reporter Manager* to send the results of the *Reporters* to the *Depot*, which replaced the previous *Distributed Controller*,
- removal of the *Centralized Controller*,
- introduction of a new *Reporter Agent* to ease the deployment and maintenance of the *Reporters*,
- possibility to create some *Reporter Repositories*,
- introduction of a graphical user interface (*incat* tool) to allow the administrators to configure the INCA resources and suites,
- change of the database type required for the *Depot*, from a round robin to any SQL one,
- changes in the data display mechanism and techniques to define the *Data Consumers*, which display the results of the *Reporters*,
- improvement of the security model,

- change of the API for the reporters, which can be written in an alternative language to Perl, as long as their input conforms to the specifications.

We previously expected to migrate to this new version during last summer. However, as the INCA development team had important delays to release the first production version, we postponed our planned migration, as we cannot use alpha or beta versions for a major operational tool. The migration is now planned for December, immediately after the validation of the first production release expected now at the end of November.

A large amount of work was done during the reporting period to continue to prepare for this migration:

- The testbed set up in March between CINECA, IDRIS and LRZ, under the responsibility of LRZ, was continuously upgraded with the various pre-release versions available from the INCA development team. These were extensively tested, whilst maintaining a continuous interaction with the developers,
- All the version and unit reporters that we use with INCA 0.10 were converted for the new version 2.0, which will allow later a faster deployment when we will migrate.
- We prepared the possibility, for those partners that require it due to security constraints (about half of them), to install and operate a local *Reporter Manager*. This feature was not initially expected in the software, and required important efforts from us to be set up and tested on our testbed. Nevertheless, due to our requirement, the INCA development team integrates this feature and offer now an easier deployment of local *Reporter Managers* than the full hand made one initially required.

It must be emphasized that the close relationship established since nearly two years with the INCA development team has yet been increased during the last months, for the mutual benefits of the two sides. For our part, we had a very fast and efficient support, with an access to all the internal developments and a direct contact with the developers and the project manager. And for their part, they had from us detailed reports of our problems, especially concerning the portability ones that we experienced on the AIX framework, because all their developments were done in the Linux environment, and we helped them to improve the portability and robustness of their software, as well as giving them the opportunity to see it deployed on another major infrastructure grid project.

5. Support of scientific Joint Research Activities

All the projects included in these scientific JRAs have had a direct and strong connection with one of the core partners (RZG for JRA 1 and 3, IDRIS for JRA 4 and 6, CINECA for JRA 5), except for JRA 2 which has this connection with EPCC, but has in fact also now carried out work and experiments on the systems of many partners (BSC, HLRS, HPCx, IDRIS, RZG, SARA).

5.1 JRA 1 (*Material Sciences*)

In JRA 1, work was focused during the reporting period on the full integration of the CP2K package (see the deliverable D-JRA1-5 [8]). This includes support of the remaining components, mainly the classical molecular dynamics simulation code FIST.

In addition to the work on fully supporting CP2K, the materials science gateway has been enhanced by designing and implementing an interface to the so-called Application Hosting Environment (AHE) used in the materials science Extreme Computing project LIAMS. For this project, the NAMD and LAMMPS Molecular Dynamics software were integrated in the AIX DCPE, for which DCRMP distributions were built by SA4 in collaboration with SA3 (see paragraph 7.2). And the required *modulefiles* have been developed in cooperation with SA4 people.

5.2 JRA 2 (*Cosmology Applications*)

During the last six months, JRA2 (Cosmology Applications) has concentrated on completing work package WP3 (*Grid enablement of FLASH through code migration*) – see the deliverables D-JRA2-3.2 [9] and D-JRA2-3.3 [10]. During this period, WP3 has been completing the process of porting and profiling FLASH, a Fortran90 astrophysical code used by the Virgo Consortium, to several of the DEISA platforms, specifically: HPCx, IDRIS and RZG, the SGI Altix at SARA, the IBM Linux/PowerPC cluster at BSC (MareNostrum) and the NEC SX-8 at HLRS.

The SA4 team members at each of these sites (EPCC, IDRIS, RZG, SARA, BSC and HLRS) have all been instrumental in the ultimate success of JRA2 WP3 and have responded to requests for assistance or further information in a timely manner for WP3 staff. Further, HLRS has also granted access to a non-DEISA staff member, an MSc student at EPCC, who ported and profiled two versions of the FLASH simulation code on the NEC SX-8 at HLRS as part of his MSc in HPC dissertation.

At EPCC, the SA4 staff have assisted JRA2 staff members to install UNICORE clients and the ssh plugins, thus enabling access to the SGI at SARA, and the DESHL, all on local workstations. This was essential to successfully implement code migration via both UNICORE's Workflow or creating a DESHL workflow ourselves, where the simulation can now be shown to began at HPCx and complete at either IDRIS or SARA, seamlessly.

5.3 JRA 3 (*Fusion Research*)

During the reporting period, the work of JRA 3 (see deliverable D-JRA3-5 [11]) was focused on enabling work of a new important plasma physics code, EUTERPE. Work on ORB5 code was finalized. ORB5 and the previously enabled code GENE are used in the DECI project GYROKINETICS.

SA4 was involved in the preparational work of this DECI project on the new SGI Altix architecture of DEISA site LRZ.

5.4 JRA 4 (*Life Sciences*)

During the last six months, JRA 4 has been exploring the phylogeny field in search of parallelized and also widely used codes; in the mean time genomics code are not set aside. On another hand, JRA 4 is closely working with eSA3 helping on the life sciences portal project. For more details, see the deliverable D-JRA4-6 [12].

Three of the phylogeny codes especially studied by the DEISA user support group at IDRIS are: MrBayes, Phylml and Raxml. The MPI version of MrBayes has been tested on this platform but unfortunately the results were not as satisfactory as expected at IDRIS, partly because of the small number of processors needed. The MPI version of Phylml is still in progress. Raxml is a promising software having a MPI version and an OpenMP version, but may not yet be extensively used throughout the community.

The IDRIS user support team is still working with the Toulouse INSERM team in the adjustments of GeneHunter-TwoLocus, and also explores the MPI version of Blast (MpiBlast) and ClustalW (ClustalW-Mpi).

All those software are candidates for the first version of the life sciences portal, project managed by eSA3 in which JRA 4 has to help in the specification and requirements of the portal, and the selection and installation of the software. All this was done in order to fulfil the needs of the community, easing the access of highly efficient parallel codes to conduct large and demanding job or simulations.

5.5 JRA 5 (*Industrial CFD*)

JRA-5 is involved in modelling for aero-acoustic (CAA) and aerodynamic (CFD) simulations for some relevant automotive applications. The software used is CFD++ from Metacomp, a commercial package. The support activities to the JRA, during the last semester of the project, were mainly related to (see the deliverable D-JRA5-6 [13]):

1. Changes to the installation and configuration of the CFD++ software in order to accomplish the DEISA_HOME inter-sites sharing;
2. Testing and tuning of new releases of the CFD++ solver across CINECA and IDRIS sites, in order to assess and resolve some scalability issues and memory leaks;
3. Testing and production of CFD++ cases on an other architecture (Linux cluster);
4. Submission of *medium* production cases for relevant simulations.

5.6 JRA 6 (Coupled Applications)

Concerning the JRA6 activities, this last period was dominated by exploiting the coupled applications of the second set and by carrying out new releases for the three current projects (see the deliverable D-JRA6-5 [14] for more details).

We solved several important problems which have been solved in collaboration with the DEISA user support group. Here are the most important:

- The operation of the DECI coupling project called FOCUS required a very close interaction with the IDRIS operation team to adjust resource classes needed by the two coupled codes. Some mechanisms have been provided to avoid releasing the resources between two coupled job executions. This system avoids wasting resources at the operation level (an important part of the supercomputer – up to 348 Power 4+ processors – must be emptied before being able to run this coupled application).
- A numerical problem has been identified in the parallel radiative transfer code (module of the Convection/Radiation project). This code gave erroneous results for particular 3D cavities. The problem was detected without difficulty with tools harnessed by the DEISA user support team.
- Optimisation phases have been necessary on the SX-8 vector platform for the KOP3D project, to point out inefficient part of the codes. In collaboration with the HLRS DEISA user support group, solutions have been found to improve the vector code efficiency.

6. DEISA Training Sessions

Six training sessions are scheduled between mid-2006 and early 2008 in different European countries (in France, Germany, Spain, Finland, Italy and Germany). The purpose of the training is to enable fast development of user skills and know-how needed for the efficient utilisation of the DEISA infrastructure. Each session is divided in two parts: the first part will give a global description and introduction to the DEISA infrastructure and will describe the general middleware services, the usage of the *DEISA Common Production Environment* and the detailed utilisation of UNICORE. The second part of the training is dedicated each time to a different HPC topic, itself usually itself divided in two parts, one technical and one based on *use cases*, to give pedagogical summaries of experiments in different scientific fields.

The schedule defined is the following:

- Paris, July 3-5, 2006
- Jülich, October 23-25, 2006
- Barcelona, February or March 2007
- Helsinki, May or June 2007
- Bologna, October 2007
- Stuttgart, February 2008

The attendance is always limited to 30 participants and DEISA takes responsibility for the travel and living expenses of up to 15 participants arriving from outside the country which hosts the event.

The first session, organized by IDRIS, occurred in Paris in July 3-6, with a dedicated part devoted to the *Highly scalable parallel applications*. The attendees were 13 European scientists, 6 staff members and 9 speakers. The second session, organized by FZJ, was held in Jülich in October 23-25, with a dedicated part devoted to the topic of *Performance and portability*, with 22 European scientists, 4 staff members and 9 speakers. The feedback sent by the participants to these two sessions was fully positive.

According to both the feedback received after the first session and our wish to extend the opening of the infrastructure and to allow more scientists to experiment with it, we decided, starting with the second event, to add a special hands-on session, to allow the participants to experiment by themselves for a few hours with some of the major features of the hardware and software infrastructure (especially the usage of the Global File Systems, UNICORE, DESHL and the DCPE). Moreover, each participant to the training sessions will become a *DEISA Test User* (see paragraph 7.4) and will have a DTU account opened to continue to experiment with the infrastructure for a period of some weeks after the session that they attended.

For the second training session, the general opinion of the participants is that this hands-on session was a very important part of the training, which allowed them to really see and experiment by themselves with the software components of the infrastructure. According to their opinions we plan to further reinforce this section during the next four training sessions scheduled.

7. Other activities

7.1 *Trouble Ticket System*

As explained in the deliverable D-SA4-4 [7], the centralised *Trouble Ticket System*, based on the *Request Tracker* [19] software, set-up and operated for the DEISA purposes at RZG, and which is now used as a centralised service for all the DEISA service activities, especially the user support and operations teams, obviously continued to be used.

7.2 *DEISA Cluster Resource Management Package*

The *DEISA Cluster Resource Management Package* (DCRMP) helps the administrators of the DEISA sites in the installation and maintenance of their software environments, both for system and user oriented components. This activity, led by CINECA as an SA3 one, has already involved SA4, because, according to the work already achieved in SA3 to distribute some middleware components, and the potential interest of it for SA4 activities, it was decided to experiment it on a first limited set of application oriented packages (see the deliverable D-SA4-4 [7]).

The objective was to be able to distribute in this way some public domain software related to applications (mainly tools, graphical or numerical libraries and applications), to simplify and accelerate the installation process by providing distributions already packaged, which are easier to install for the supported dedicated platforms than by using the usual generic processes. It was initially decided to use two rather large and complex software packages of the DCPE: Python and the omniORB CORBA distribution. To handle the characteristics of this kind of software, which have some important differences from the ones used in SA3, it was necessary to extend the initial framework designed to build the first release of DCRMP. For this, a new tool was defined and built, called *DCRMP Package Creator* (DPC) (see deliverable D-SA3-5 [1]).

As explained in the deliverable D-SA3-7 [2], the two packages for Python and omniORB have been initially tested by the two sites involved in the development of DCRMP, CINECA and IDRIS. After this internal testing, two other sites, EPCC and FZJ, were especially interested to experiment with DCRMP distributions as a possible new easy way to install software packages. This is why they offered to test these two first application oriented ones. Both partners found that this mode of distribution and installation of software is really convenient, easy to use and allows fast installations which heavily reduced the required time. During their experiments, they found a few portability issues and some lack of precision in the initial documentation. Using their feedback, it has been possible to implement some improvements and corrections in the two packages, and to enhance the documentation and consistency checks of the DPC tool, to clarify some issues and avoid some pitfalls to its users.

As agreed with SA3, a DCRMP distribution of two public domain software packages in the Molecular Dynamics field (NAMD and LAMMPS) was recently built and released, according to the requirements of the newly launched LIAMS DECI project which will access several AIX platforms and needs to use these software in all places. Using these DCRMP packages will lower the deployment time to install these packages, and therefore the DECI project start-up time.

7.3 DEISA Applications Test Suite

As part of the various test suites developed inside DEISA to be able to check the behaviour of various components (global file systems, middleware, batch systems), an *DEISA Applications Test Suite* (DATS) has been developed in the SA4 activity to check the behaviour of some major software components (compilers, linkers, some scientific libraries, parallel libraries and execution framework). See the deliverable D-SA4-4 [7] for a complete description of the content of the DATS.

As explained in section 3, the DATS has been installed during the reporting period on nearly all the sites which did not yet have it, and they also integrated its different tests inside INCA, using dedicated *unit reporters*, to enable their automatic launch and to test on a regular basis some major aspects of the software infrastructure.

7.4 DEISA Test Users

As decided by the *DEISA Executive Committee* last summer, a new kind of access to the infrastructure has been opened, to allow any European scientist to test specific services like the global file systems and UNICORE, and to enable evaluation of the new opportunities that the DEISA infrastructure can offer to them, encouraging them to answer to the next DECI calls of proposals. For this purpose, a new category of users, the *DEISA Test Users* (DTU), were defined and set up, and in particular all the participants to the training sessions have now an account of this kind opened.

7.5 User documentation

The current public version (1.3) of the *Primer* [16] is available since June. Compared to the previous version, important improvements have been made to better describe the initial customization of the UNICORE environment, with additional explanations and clarifications. This also includes a precise example of the complex procedure that users must follow, especially the initial configuration of the certificates required to use this tool. The other major addition in this version was the description of the case of the so-called *DEISA Standard Users* (DSU), containing the particular information that this category of users must know. These users are not involved in the JRAs or DECI projects and remain “normal” local users, but will allow the jobs that they submit locally to be rerouted to another homogeneous platform of the DEISA infrastructure if the global operations management of the infrastructure consider it as an efficient procedure. For this job migration to take place, their jobs must fulfil some additional requirements, mainly to put the files used during the execution of their jobs in the global file systems defined across the DEISA sites and to use only software included in the DCPE or accessible in their own private environment.

The ongoing new version of the *Primer* (1.4) will have two new improvements. The first one will be to fully describe the situation of the SGI Altix and Linux/PowerPC sites, with some additional information about the characteristics of the usage of their own environments inside the DEISA infrastructure and the specificities of their own DCPE. The second one will be to give the small amount of specific information useful for the new kind of DEISA users recently introduced, the *DEISA Test Users*.

8. References and Applicable Documents

- [1] DEISA D-SA3-5 deliverable: *Second release of DCRMP*
- [2] DEISA D-SA3-7 deliverable: *Third release of DCRMP*
- [3] DEISA D-SA4-1 deliverable: *SA4 Service Definition and Operation*
- [4] DEISA D-SA4-2 deliverable: *Basic DEISA Infrastructure Documentation*
- [5] DEISA D-SA4-3 deliverable: *First SA4 Annual Report*
- [6] DEISA D-SA4-3.2 deliverable: *SA4 Mid-term Activity Report*
- [7] DEISA D-SA4-4 deliverable: *SA4 Second Annual Activity Report*
- [8] DEISA D-JRA1-5 deliverable: *DEISA Operation of full CP2K package*
- [9] DEISA D-JRA2-3.2 deliverable: *FLASH on the DEISA infrastructure*
- [10] DEISA D-JRA2-3.3 deliverable: *Documentation for D-JRA2-3.2*
- [11] DEISA D-JRA3-5 deliverable: *Activity report and roadmap evaluation*
- [12] DEISA D-JRA4-6 deliverable: *Final report of second set of applications*
- [13] DEISA D-JRA5-6 deliverable: *Intermediate results on medium heavy test cases*
- [14] DEISA D-JRA6-5 deliverable: *Production operation of second set of projects*
- [15] DEISA FAQ: <http://www.deisa.org/userscorner/faq.php>
- [16] DEISA Primer Documentation: <http://www.deisa.org/userscorner/primer.php>
- [17] INCA (Test Harness and Reporting Framework): <http://inca.sdsc.edu/>
- [18] Modules: <http://modules.sourceforge.net/>
- [19] Request Tracker: <http://www.bestpractical.com/rt>
- [20] TeraGrid: <http://www.teragrid.org/>
- [21] UNICORE (UNiform Interface to COmputing REsources): <http://unicore.sourceforge.net/>

9. List of Acronyms and Abbreviations

Ataskf	Applications Task Force
ATS	Applications Test Suite
CNRS	<i>Centre National de la Recherche Scientifique</i>
CORBA	Common Object Request Broker Architecture
DATS	DEISA Applications Test Suite
DCPE	DEISA Common Production Environment
DCRMP	DEISA Cluster Resource Management Package
DECI	DEISA Extreme Computing Initiative
DPC	DCRMP Package Creator
DSU	DEISA Standard User
DTU	DEISA Test User
Execution site	The site where the jobs of a user are executed
Home site	The site where a user usually works, logs in and submits jobs
IDL	Interface Definition Language
INSERM	Institut national de la Santé et de la Recherche Médicale
JRA	Joint Research Activity
Reference site	Same as <i>Home site</i>
RMIS	Resource Management Information System
SDSC	San Diego Supercomputer Center
TTS	Trouble Ticket System