EUROPEAN MIDDLEWARE INITIATIVE

DJRA1.1.3 - COMPUTE AREA WORK PLAN AND STATUS REPORT

EU DELIVERABLE: D5.1.3

Document identifier:	EMI-DJRA1.1.3-1277612-Compute_Area_Work_Plan-M24-v1.0.odt
Date:	30/04/12
Activity:	JRA1
Lead Partner:	INFN
Document status:	Final
Document link:	http://cdsweb.cern.ch/record/1277612

Abstract:

This deliverable contains the detailed work plan of the Compute Services technical area, compliant with the overall EMI Technical Development Plan. The plan is released early in the project lifetime and updated every year, including a status report on the achievements of the past 12 months compared to the planned objectives. The status report at M03 will cover the state-of-the art while the work plan at M36 will provide recommendations for further work.

I. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Marco Cecchi	INFN/JRA1	24/04/12
Reviewed by	Ivan Marton, Krzysztof Bendyczak	NIIFI/JRA1 UWAR/JRA1	
Approved by	PEB	-	25/05/12

II. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1	11/04/12	ToC for review	Marco Cecchi/INFN
2	24/04/12	v0.4 for review	Marco Cecchi/INFN
3	25/05/12	v1.0 PEB approved version	PEB

III. DOCUMENT CHANGE RECORD

Issue	Item	Reason for Change
1		
2		

IV. DOCUMENT AMENDMENT PROCEDURE

This document can be amended by the authors further to any feedback from other teams or people. Minor changes, such as spelling corrections, content formatting or minor text re-organization not affecting the content and meaning of the document can be applied by the authors without peer review. Other changes must be submitted for peer review and to the EMI PEB for approval.

When the document is modified for any reason, its version number shall be incremented accordingly. The document version number shall follow the standard EMI conventions for document versioning. The document shall be maintained in the CERN CDS repository and be made accessible through the OpenAIRE portal.

V. GLOSSARY

Acronym	*Long Name*
A-REX	ARC Resource-coupled EXecution
API	Application Programming Interface
ARC	Advanced Resource Connector
BES	Basic Execution Services
BLAH	Batch Local Ascii Helper
CE	Computing Element
CLI	Command Line Interface
CRL	Certificate Revocation List
CREAM	Computing Resource Execution And Management
DCI	Distributed Computing Infrastructure
DEISA	Distributed European Infrastructure for Supercomputing Applications

DoW	Description of Work
EGI	European Grid Infrastructure
EMI	European Middleware Initiative
FTS	File Transfer Service
HED	Hosting Environment Daemon
HEP	High Energy Physics
HiLA	High Level API
HPC	High Performance Computing
HPC-BP	HPC-Basic Profile
ICE	Interface to CREAM Environment
JC	Job Controller
JDL	Job Description Language
JSDL	Job Submission Description Language
JMS	Job Management Service
JURA	Job Usage Reporter of ARC
LAM-MPI	Local Area Multicomputer-MPI
LB	Logging and Bookkeping
LRMS	Local Resource Management System
MCT	Minimum Completion Time
MPI	Message Passing Interface
NGI	National Grid Initiative
OGF	Open Grid Forum
OS	Operating System
PACX-MPI	PArallel Computer eXtension-MPI
PGI	Production Grid Infrastructure
PBS	Portable Batch System
PT	Product Team
RTE	Run Time Environment
SE	Storage Element
SGAS	SweGrid Accounting System
SGE	Sun/Oracle Grid Engine
SLURM	Simple Linux Utility for Resource Management

SMS	Storage Management Service
TSF	Target System Factory
TSS	Target System Service
UCC	UNICORE Commandline Client
UNICORE	UNiform Interface to COmputing REsources
XACML	eXtensible Access Control Markup Language
XNJS	eXtensible Network Job Supervisor
WG	Working Group
WLCG	Worldwide LHC Computing Grid
WMS	Workload Management System
WS	Web Service

The complete EMI glossary is available at https://twiki.cern.ch/twiki/bin/view/EMI/EmiGlossary .

VI. COPYRIGHT NOTICE

TABLE OF CONTENTS

1. INTRODUCTION

1.1. EXECUTIVE SUMMARY

One of the most important goals of the EMI project is achieving a unified and consolidated middleware distribution. In the past years, the grid computing services now part of the EMI project were separately implemented, mainly in the field of High Energy Physics, in the context of three different middleware stacks: ARC, gLite and UNICORE. To be able to converge to such a harmonized distribution, much of the effort during the first and second year was devoted to providing a common ground, through use of same protocols and agreements, in such a way that both end users and services are able to seamlessly interoperate throughout both existing and new DCIs. The focus on the third year will basically move to consolidation, further work on the agreements previously taken and fulfilling new requirements from the user community that have come in the meanwhile.

This document gives the status of the Compute Area activity after the second year of the EMI project and the plans for the third year. These plans exclude work such as regular maintenance and bug-fixing of components in production usage. This document is organized into a status report and the third-year work plan.

The Compute Area status report section gives the current status of work performed to achieve the objectives as described in DNA1.3.2 and DJRA1.1.2. These reports are organized by objective and give the actual state of the relevant work. To give an overall status, the vast majority of the year two objectives, as by DNA1.3.2, were successfully achieved. There are cases where the schedule has fallen behind due to circumstances beyond the control of the area groups or unforeseen extra work taken by product support and release process.

The main points are as follows: a considerable amount of work during the second year was devoted to the implementation of the EMI Execution Service. **The EMI Execution Service** (EMI-ES) deals with the submission and management of computational tasks on a computing resource and covers important use cases from both high-throughput and high-performance computing. Worth of mention is the ability to compute multi-scale workflows on both HPC and HTC environments, described by the Compchem community [COMPCHEM]. The EMI-ES has always represented a significant part of the EMI vision, its fully interoperable and production-quality implementation become now of primary importance. Acknowledgments by official standardization bodies would also be welcome and can surely be pursed during the third project year, even if not being directly a primary goal. Once this interface is finalized, the **harmonization** process of the existing clients will finally take place, built on top of the EMI-ES, so as to achieve uniform and high-level access to computing resources without reducing the present functionality required by each DCI.

The **execution of parallel jobs** is also another important aspect when dealing with the enlarged scenario composed of both HPC and HTC resource. Although the current middleware stacks already provided basic access to such applications, this support resulted to be limited and not always usable in practice by many. EMI wanted to face the challenge of providing extended and seamless access to the execution of MPI jobs across all the middleware it supports. Again, the EMI-ES, already an established agreement that also comprises the execution of parallel jobs, reveals to be the gluing factor. While the previous years were devoted to the harmonization of MPI-Start as a backend for parallel jobs, the focus will now move to providing a unified framework, that is a front-end, to define and submit parallel jobs in a seamless way throughout the EMI middleware.

Another important aspect of harmonization requires improving sustainability through the consolidation of transversal areas such as security and infrastructure at large. For the former, the first steps started with the harmonization of Argus [ARGUS] into the EMI computing elements, which happened for all A-REX [AREX], CREAM [CREAM], WMS [WMS] and UNICORE/X [UX]. All the three middleware are already able to interface to the gLite Argus authorization framework and enforce authorization policies based on Argus. This consolidation process will now move to the harmonization of all the operations needed for authentication, enabled by the EMI common authentication library.

About the interaction with infrastructure, the adoption of information service related standards and the adoption of a uniform way to measure resource utilization are important targets that the compute components are pursuing. The second year has seen the agreement on the compute accounting record taking place, while, for what concerns the well-known standard for the specification of the characteristics and status of grid entities, that is GLUE, the work on v. 2.0 has progressed to the point of raising some open points. Such issues were in particular catalyzed by its implementation in the gLite WMS, whose implementation will help producing a mature profiling. In particular, it is important that the WMS and all the EMI Compute clients and are provided with a common way of retrieving resources part of the stack. The common EMI service registry [EMIR] will remarkably allow for easier interoperability, cooperation and uniformity in operation of DCIs. In this respect, the main EMI services are required to uniformly implement service discovery through it.

1.2. PURPOSE AND SCOPE

This document covers in detail the progress in the development of the Compute Area work plan during the second project year and describes its evolution for the third and final year. It is meant to be a reporting/planning document, its style it is not the one of an in-depth technical paper. While, when the matter is particularly relevant, a description of the technical aspects of the issue is not avoided and richness of references is also provided, generally speaking, this document is supposed to mainly focus on the planning of the remaining activities and the distribution of the effort. The intended audience is presumed to be the document reviewers, project reviewers and interested parties in affected projects. Some side comments that are better to be known in advance before reading the document:

- not every single objective is mapped to a project milestone or deliverable. For those which are not, adequate and tangible references in terms of produced papers, plots, proof of concepts, examples etc. will be given to try and illustrate in the best way how each achievement has been carried out;
- for any deviation from the original work plan, the required justifications and underlying motivations will be provided as well, so as to give a possibly clear and concise understanding, in a broader context, of the reasons for the decision to move away from some specific original commitment.

1.3. DOCUMENT ORGANIZATION

This document is organized in two major parts. In the first one, Section 2, a status report of the current status of the EMI Compute Area components is presented. The second part, Section 3, presents a detailed description and is planning the objectives for the second and third project year.

1.4. REFERENCES

DJRA1.1.1	http://cdsweb.cern.ch/record/1277608
DJRA1.1.2	http://cdsweb.cern.ch/record/1277610
DNA1.3.1	http://cdsweb.cern.ch/record/1277540
DNA1.3.2	http://cdsweb.cern.ch/record/1277543
COMPCHEM	Antonio Laganà, Carlo Manuali, Marco Cecchi, Antonia Ghiselli, Alessandro Costantini, Michele Carpenè, Elda Rossi, *Efficient workload distribution bridging HTC and HPC in Scientific Computing*, 12th International Conference on Computational Science and Its Applications
A-REX	www.nordugrid.org/documents/arex_tech_doc.pdf
CREAM	http://grid.pd.infn/it/cream
CLASSAD	http://research.cs.wisc.edu/condor/classad/
UX	http://www.unicore.eu/documentation/manuals/unicore6/unicorex/
ARGUS	https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework
GLUE20	http://www.ogf.org/documents/GFD.147.pdf
EMIR	https://twiki.cern.ch/twiki/bin/view/EMI/EMIRegistry
EMIES	https://twiki.cern.ch/twiki/bin/view/EMI/EmiExecutionService Specification: https://twiki.cern.ch/twiki/pub/EMI/EmiExecutionService/EMI-ES-Specification_v1.0.odt Schema https://twiki.cern.ch/twiki/pub/EMI/EmiExecutionService/es-schemas-1.0-src.tar
PBS	http://www.openpbs.org
LSF	http://www.platform.com/workload-management/high-performance-computing
SGE	Oracle Grid Engine http://gridengine.sunsource.net/
SLURM	https://computing.llnl.gov/linux/slurm/
CAR_TF	https://twiki.cern.ch/twiki/bin/view/EMI/ComputeAccounting
GRIDSITE	http://www.gridsite.org/
FHS	http://www.pathname.com/fhs/

2. COMPUTE AREA STATUS REPORT

This chapter will review one by one all the work plan tasks, as defined in DNA1.3.2 and, more specifically, in DJRA1.1.2, to be completed by the second project year (M24) or that have been completed in advance or where progress has to be reported or that have been canceled due to change of priorities.

2.1. GLUE 2.0 SUPPORT IN JOB MANAGEMENT SERVICES

DNA1.3.3 ref C1; Status: Delivered with EMI 2

A remaining activity for the achievement of this objective was the implementation of the dynamic support in the CREAM CE, to publish LDAP rendered GLUE 2.0 [GLUE2] information retrieved by the batch system and the implementation of a so called 'cluster' service, able to optimize the publication of GLUE information in those sites where multiple CREAM CE nodes are installed.

For what concerns the former, the CREAM team has decided to support and maintain, under its own full responsibility, all the various gLite modules that were originally used for this purpose and not to carry on with the exercise that was successfully accomplished to integrate/adapt the ARC LRMS modules, as it was reported in the DJRA1.1.2 description of this task. Since the beginning, in fact, the project never considered the unification of the modules for the interaction with the batch system a potential area for unification and, despite looking like a decision against sustainability, the CREAM PT decided, after a careful evaluation, that the adaptation of the ARC modules in CREAM would probably have required much more work that maintaining and evolving the existing scripts. Some differences in fact were found, where some intervention on the ARC modules would have been needed as they could not be used as they are for gLite.

The activity on the 'cluster' service testifies how EMI is attentive to deliver software that can effectively be utilized in production environments, and of course also testifies some delays that the work plan can have. This special layout, in fact, is not needed in ARC or UNICORE distributions. This work consisted of a non trivial work, given that in gLite there can be more Computing Services at a site, to be deployed in grouped configurations, also referred to as clusters. This makes more difficult the GLUE 2.0 publication, as both ‘with cluster’ and ‘without cluster’ modes must be taken into account, and, in the former case, a different layout must be adopted to coordinate the publishing of shared information among the CE instances.

There were no deviations from the agreed plan of the past period to be reported.

References

- https://wiki.italiangrid.it/twiki/bin/view/CREAM/CreamGlue2

2.2. GLUE 2.0 SUPPORT IN MATCH-MAKING MODULES AND CLIENT TOOLS

DNA1.3.3 ref C8; Due date: M26; Status: Partially delivered with EMI 2

This activity concerns the implementation of a new module in the WMS engine, responsible for querying over LDAP a GLUE 2.0 enabled BDII and fetching information in the WMS internal cache. For the record, this modules are called 'purchasers' in the WMS architecture. A conversion from LDIF to Classad [CLASSAD] is performed by the WMS engine and the result stored into an internal cache, the so called Information Supermarket. Both GLUE 1.3 and GLUE 2.0 resources need to be fetched, paying attention that the same endpoint publishing both v. 1.3 and 2.0 is not accounted twice.

After a careful investigation, it actually turned out that the GLUE 2.0 schema was deliberately kept generic enough to allow for several implementations to be fully compliant, yet unable to interoperate in practice. This is quite typical of standards, whereby an in-depth profiling is usually needed to refine all the harmonization aspects that come into play while progressing. The same also happened in other areas, for example security, leading to the EMI XACML profiling and similar things. The implementation in the WMS, that must be able to consume information produced by all the kind of EMI Computing Elements, acted as a the 'driving force' to sort out all the missing pieces and actually glue together the three middleware providers. Just as an example of the required complexity, in those cases where the same resource is both publishing in GLUE 1.3 and 2.0, there must be a way for the WMS to understand that the underlying resource is actually the same, not to represent it twice. This is not always easy, and requires a profiling, because the standard is deliberately generic enough not to foresee a unique handle to represent an *ObjectClass*. For the EMI CEs, it was agreed that the queue name would have to be published in a specific attribute (*GLUE2.Computing.Share.ID*) of the Share ObjectClass, in such a way that the GLUE 1.3 CE identifier could be retrieved from there or from *GLUE2.Computing.Share.Policy* if the former is missing.

The current status is that purchasers and match-making modules are fully implemented for what concerns Compute Elements (in particular the objectclass *GLUE2ComputingService*). The corresponding work for Storage Elements (referring to objectclass *GLUE2StorageService*) has not been completed at the time of writing, because the complexities requiring profiling revealed to be even harder with the Storage. During the first tests, in fact, several inconsistencies and sometimes errors have been found in the way each middleware is publishing its content. The present phase consists in reviewing the purchaser modules is such a way to sort out all these issues, find a way for all Storage Elements to publish information in a consistent way and find a workable solution for the current implementations. Indeed, there are some deviations from the agreed plan of the past period to be reported. Full support for StorageElement has not been finalized yet. As said, this is not only matter of work that concerns the WMS. The implementation in the WMS triggered new issues that still have to be worked out with the data area product teams. The LDAP rendering of GLUE2.0 [OGF_LDAP] has been kept generic enough to allow for several interpretations that producers and consumers need to agree upon. For what concerns support in the clients through the JDL, the formalism is already capable to handle GLUE2.0 resource descriptions, for both CEs and SEs. For such definitions to be properly interpreted by the server, the aforementioned support in the match-making and purchaser modules must be finalised.

References

- OGF LDAP rendering: http://forge.ogf.org/sf/go/doc15518?nav=1

2.3. COMMON JOB SUBMISSION AND MANAGEMENT METHODS

DNA1.3.3 ref C5; Due date: M27; Status: Partially delivered with EMI 2

DNA1.3.3 ref C6; Due date: M27; Status: Partially delivered with EMI 2

These tasks refer to the implementation of the EMI Execution Service [EMIES] in all the EMI computing services and their clients, as by its reference specification, that was defined in year one. This is one of the most distinguished developments of the project, and it will allow, also fostered by the migration to the common authentication library in the Compute Area components, seamless execution of complex workflows to HPC and HTP environments through a single entry point, i.e. the gLite WMS. Delegation and authorization aspects will still need to be finalised in the security area, especially for what concerns interoperability with ARC/gLite and UNICORE. To final goal is to have

each client of each of the three middleware solutions able to send and manage jobs to each different computing service and, conversely, to have all computing services able to accept jobs sent by each different client. In some interesting use-cases that have been discussed with user communities in the reporting period, this can enabled by the architecture described in the picture below, whereby the results of a huge computation performed on a super-computer, are repeatedly taken and independently executed in a more opportunistic way on a HTC Grid until a certain termination condition is met:

At the time of writing, the overall completion status for each middleware can be summarised by the following table. It is important to mention that the percentages are based on personal esteems of the work to be done as felt by the involved PTs.They are not meant to have an absolute value, as only the teams know exactly the amount of work that a given feature might require, whether it requires a few modifications to existing code or if it requires a new redesign of some core components etc.

Progress in the completion of EMI-ES in services and their clients	A-REX	CREAM	UNICORE
Overall completion	85.00%	85.00%	70.00%
Server	90.00%	85.00%	70.00%
Client	80.00%	85.00%	70.00%
General comments	- There is no complete validation of activity description. - There is no schema validation. - Service capabilities validation is partial. - GLUE2 rendering implementation is still undergoing.	- Both server and client miss the ResourceInfo port type. For the server, GetResourceInfo has being mostly developed and is being tested. QueryResourceInfo requires integrating Xpath. - The 85% complete prototype can be released in EMI 2 as CREAM 1.14.0. - The missing development should be finished and released as EMI 2 update.	- Can be considered complete except for the queryResourceInfo, totally missing for now, and the getActivityInfo function will be very limited. - The server will be released in EMI-2, the client will not.

Table 1: Overall status of the implementation of the EMI-ES in the three middleware providers.

At the same time, interoperability tests have started, involving all the three middleware services in the possible combinations of clients and servers. Detailed progress can be found in the included reference

(*). The interface covers a very large spectrum of submission use-cases, because it was meant to cover all the possible combinations of present and future needs. In this respect, adapting this interface to the existing functionality revealed to be a non trivial operation, that sometimes required a re-design of some back-end functionality. This is basically the reason for the not particularly significant overall delays that have been reported. Worth of notice, is that the first interoperability tests started among ARC, gLite and UNICORE were successful in getting basic job submission and status information.

References

- https://savannah.cern.ch/task/?func=detailitem&item_id=22396

- https://savannah.cern.ch/task/?func=detailitem&item_id=22397

- https://savannah.cern.ch/task/?func=detailitem&item_id=22398

- https://savannah.cern.ch/file/emi-es-server-plan.txt?file_id=23118

- *https://savannah.cern.ch/task/?func=detailitem&item_id=22401*

- *https://savannah.cern.ch/task/?func=detailitem&item_id=22400*

- *https://savannah.cern.ch/task/?func=detailitem&item_id=22399*

- https://twiki.cern.ch/twiki/bin/view/EMI/TestPlan26 (*)

2.4. IMPROVE INTERACTIVE ACCESS

DNA1.3.3 ref C3; Status: Could be done depending on available effort

This task was more about exploring ways to improve and evolve support for interactive access in Grids, rather than providing a specifically targeted tool for allowing interactive access, either in a limited or fully functional way. After a review of what functionality was already provided by the existing middleware and third-party tools to enable support for interactive access, it turned that what we had was already complete enough to provide a valid basis for supporting interactivity at various levels. Also referring to the possible directions for investigation that were suggested in DJRA1.1.2, the computing services already provide features to inspect the output files while the job is still running, and for what concerns the provisioning of shell-like access, that can be accomplished through well known third-party tools. In this respect, this item will be superseded by more important tasks in the planning for the third year. There were no deviations from the agreed plan of the past period to be reported.

References

- http://grid.ifca.es/wiki/Middleware/i2glogin

- https://edms.cern.ch/document/590869/1 (look up for ‘job perusal’)

2.5. AGREEMENT ON COMPUTE ACCOUNTING RECORD

DNA1.3.3 ref C2; Status: Delivered with EMI 2

A compute accounting record is defined reflecting practical, financial and legal requirements of resource consumption, including CPU time, wall-clock time and memory usage. This task produced an agreement, in terms of XML schema definitions, over detailed and aggregated usage records. This was done by addressing the previous Usage Record limitations and by extending accounting records to include VO-aware storage usage accounting. This activity resulted in a description document and two XML schemas, one for each record type. Both were based on existing OGF standards, that has been slightly modified both in syntactical and semantic aspects to allow for extended interoperability for the existing middleware layers and taking into consideration existing grid use cases. There were no deviations from the agreed plan of the past period to be reported.

References

- https://savannah.cern.ch/task/?21761
- [CAR_TF]
- https://twiki.cern.ch/twiki/pub/EMI/ComputeAccounting/CAR-EMI-tech-doc-1.0.doc

2.6. INTEGRATED SOLUTIONS TO INTERFACE WITH BATCH SYSTEMS

DNA1.3.3 ref C7; Status: Delivered with EMI 2

This task refers to the ability of all the EMI computing services to fully support a set of batch systems identified by the project. These initially were:

- PBS/Torque family

- Sun/Oracle/Univa Grid Engine

- LSF

Among these, PBS and LSF were already supported by all the three CEs. SGE was supported in A-REX CE and UNICORE computing services and not in CREAM. CREAM started working on supporting SGE in the reporting year, and since EMI-1 Update 9 SGE is officially enabled as one of the supported batch systems in the EMI distribution. In this respect, the original task can be said to be complete. Later in the course of the year SLURM [SLURM] was also added. SLURM is fully supported in A-REX and UNICORE, CREAM will need to add support for it. For the record, full support indicates the ability to implement missing functionality and to perform proactive maintenance on existing functionality in terms of job submission, management and information retrieval - keeping up with new product releases, updates and so on. To be more specific, full support does not only mean the ability to interact with the batch system for job specific operations, but also the ability to query for specific and overall information about all the grid jobs delivered to a batch system (number of queued/running jobs, status, etc.). There were no deviations from the agreed plan of the past period to be reported. The way in which the CREAM PT decided to handle the interaction with the LRMS, that applies to this task as well, is the same that the one described in 2.1.

References

- https://savannah.cern.ch/task/?func=detailitem&item_id=20339

- https://savannah.cern.ch/task/?func=detailitem&item_id=21768

2.7. CONSOLIDATION/HARMONIZATION OF COMPUTE AREA CLIENTS

DNA1.3.3 ref C10; Due date: M25; Status: Ongoing

The activity concerning the definition of a plan for the consolidation and harmonization of the computing clients and APIs took place as planned in the reporting year. Several agreements have been reached and documents produced, in particular for what concerns the shape of the high-level APIs in C and particularly in Java. A taxonomy on the command line options of the three compute clients has been produced, as shown in the references. Unfortunately a global agreement among the involved components of the devoted task force is still lacking, so that this activity cannot be considered to be fully achieved, as it will be also described in paragraph 3.1 when addressing the planning for the third year. In particular, what is still being discussed is the need for a unified client, a CLI interface: whether it is worth at all and, in case, in which language it should be written and accordingly what teams should be more involved in its eventual development. As said, a more refined planning for this task will be addressed in the second part of this document.

References

- https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T2ComputeClientWorkPlan

- https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T2ComputeClientProposal

- https://twiki.cern.ch/twiki/bin/view/EMI/EmiEsClientApiDefinition

- https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T2ComputeClientOpportunities

- https://savannah.cern.ch/task/?22428

2.8. IMPLEMENT CLOUD STRATEGY IN COMPUTE AREA

DNA1.3.3 ref C11; Status: Ongoing, partially delivered with WNoDeS

This activity is ongoing with the recent inclusion of WNodes in the EMI stack.

2.9. COMMON PARALLEL EXECUTION FRAMEWORK

DNA1.3.3 ref C4; Status: Delivered with EMI 2

In the reporting year, the devoted task force was successful in identifying the mentioned common parallel execution framework. In particular, more than simply trying to identify a common back-end, a somewhat difficult and even not necessary operation, the task force decided to find an agreement on a common definition of parallel jobs across the middleware. The idea was to adopt the ParallelEnvironment as defined by the EMI-ES. This also leaves the implementation of the back-end up to each single service, leaving the expected behaviour defined by the interface. This proposal, when publicly presented, was greeted with enthusiastic approval by the involved product teams. It required slight adaptations to the definition of the EMI-ES that were promptly done in order to accommodate for different kind of requests of parallel applications, coming from the long experience of MPI-Start, that were not initially considered. For what concerns the implementation of this proposal when using MPI-start as a common back-end, some adaptationsfor ARC RunTimeEnvironments and UNICORE ParallelEnvironment were already done in the reporting year. The rest will automatically come with the implementation of the the EMI-ES ParallelEnvironment

with the adaptations that were requested in the meantime by the TF to support for hybrid MPI-openMP applications and memory/CPU affinity. There were no deviations from the agreed plan of the past period to be reported.

References

- https://indico.cern.ch/contributionDisplay.py?sessionId=4&contribId=46&confId=147484

- https://devel.ifca.es/mpi-start

2.10. INTEGRATION WITH ARGUS

DNA1.3.3 ref X5; Status: Delivered with EMI 2

The only missing service in the Compute area to be made interoperable with Argus was the WMS. During the reporting period, this integration was accomplished through the PEP client C APIs. Other than this, a restructuring activity took place to clean up old code and to provide a lean design in the way authentication and authorization were handled in the WMS interface (called WMProxy). This has significantly reduced the number of lines of code, even in front of an augmented functionality, and will make easier the integration with the EMI Common authentication library. There were no deviations from the agreed plan of the past period to be reported. For what concerns the current support for Argus PDP in UNICORE production environments (DJRA1.1.2 C1.1), given that from the development point of view the task was accomplished, it was decided, after a more careful evaluation of the present work and the future commitments to the objectives for the third year, that any further request will have to be handled as a support activity, so that the present task can be considered as complete. About the ability for the ARC services to directly query the PDP engine by directly defining their XACML expressions (C1.2), after a more careful investigation with the Compute Area leader and the PTB, it turned out that the Argus library provided by the Security Area is expressive enough to properly interpret all the use-cases needed in ARC. In case, some bugs can eventually be opened to the relevant security team, but, the idea to by-pass a functionality already provided by a devoted team has to be considered against the overall planning of the project activities. In this respect, task C1.2 can be reported as complete.

References

- https://savannah.cern.ch/task/?func=detailitem&item_id=21060

2.11. GSI REPLACEMENT

DNA1.3.3 ref X12; Due date: M32; Status: Delivered with EMI 2

After a careful review of all the involved components, it finally turned out that no service in the Compute area depends on the GSI protocol. In particular, the rest of the investigation that was started on the previous work plan, addressed the Logging&Bookkeeping service, that does rely on the Globus libraries, but only for abstracting more easily the operation on security tokens handling, not to perform delegation, as operating on behalf of the user is not needed in the L&B. The WMS relies on Gridsite delegation [GRIDSITE], so not depending on GSI as well. ARC software, moreover, can be configured to use GSI-free production software. As always said, UNICORE has never relied on the

Globus software since its very beginning. There were no deviations from the agreed plan of the past period to be reported, the task was complete in advance with respect to its schedule.

2.12. IMPROVE USABILITY OF CLIENT TOOLS

DNA1.3.3 ref X8; Due date: M32; Status: Ongoing

There have been no specific user requests complaining about the usability of client tools and interfaces in the reporting period. For what concerns the coherency of command line options, discussed as part of the client harmonisation task, there is still a debate over this matter: the proposal is in fact to provide one common CLI that will automatically fix the problem of the harmonisation of the options. What wasdone, to start this evaluation with a clear vision of the impact, was the provision of a synoptic table comparing all the command line options of each client among A-REX, CREAM and UNICORE/X. Then, this table was also extended so as to take in consideration products, such as SAGA, that are not directly in the EMI portfolio, but that could help harmonization if the project decides to integrate those. The deviations from the original work plan items are basically the same as in 2.8.

References

- https://twiki.cern.ch/twiki/pub/EMI/EmiJra1T2ComputeClientOpportunities/CLI_commands_options.ods

2.13. PUBLISH COHERENT GLUE 2.0 INFORMATION

DNA1.3.3 ref X1; Due date: M26; Status: Partially delivered with EMI 2

This task refers to the implementation of agreements meant to unify the way service description is handled in EMI, in terms of precise version information, description and status of provided functionality etc., all this published in the GLUE2.0 schema. The remaining services that needed to finalise their GLUE2.0 description were A-REX and CREAM. This task (a minor development) was accomplished in the reporting year and released in EMI updates.

References

- *https://savannah.cern.ch/task/?21820*

- https://savannah.cern.ch/task/?21916https://bugzilla.nordugrid.org/show_bug.cgi?id=2591

- https://savannah.cern.ch/task/?22268

2.14. ADHERE TO OPERATING SYSTEM STANDARDS

DNA1.3.3 ref X6; Due date: M26; Status: Partially delivered with EMI 2

As of EMI-1, the majority of the Compute Area components was already FHS compliant, with only minor details missing (see references). The EMI release process, defined by SA2, clearly states that all components in an EMI release should follow the operating system-defined methods for service

control. This being the case, any computing service released through the EMI process has been thoroughly checked by the quality group (EMI SA2) in this respect, so that the EMI-2 release Compute Area services can be said to comply to this task. There were no deviations from the agreed plan of the past period to be reported.

References

- [FHS]

- https://savannah.cern.ch/bugs/?63007

- https://savannah.cern.ch/bugs/?func=detailitem&item_id=79712

- https://savannah.cern.ch/bugs/?func=detailitem&item_id=79415

- https://savannah.cern.ch/bugs/?func=detailitem&item_id=79416

2.15. PORT, RELEASE AND SUPPORT ON AGREED PLATFORMS

DNA1.3.3 ref X7; Due date: M34; Status: Partially delivered with EMI 2

All EMI components are required to perform this task by the SA1.3 release management, so that specific planning is not needed in this case. The affected components in this respect are all EMI Compute Area components, as taken from DNA1.3.2: A-REX, ARC gridftp jobplugin interface, ARC LRMS modules, pre-WS compute CLI (ng*), WS compute CLI (arc*), BLAH, CEMon, CREAM server, CREAM client, LSF module, CREAM PBS module, CREAM, EMI-UI, gLite CLUSTER, gLite-MPI, L&B, TORQUE WN config, TORQUE server config, UNICORE Client, UNICORE HILA, UNICORE TSI, XNJS, UAS-C, U-BES, U-EMIEX, WMS server, WMS client. The components listed above have been attempted to be ported by their respective product teams. At the time of writing this document various components within the EMI-2 release candidate can be built manually directly on the required operating systems listed above. It is not possible to build the complete set within the ETICS build system.

EMI-2 build status of the EMI Compute Area components	Scientific Linux 6	Scientific Linux 6	Debian 6	Comments
ARC CE	100.00%	100.00%	N/A	Problems with dependencies, in particular Gridsite, for Debian 6.
ARC clients	100.00%	100.00%	N/A	
ARC gridftp plugin	100.00%	100.00%	N/A	
CREAM	100.00%	100.00%	N/A	
CREAM UI	100.00%	100.00%	N/A	

L&B	100%	100%	85.00%	Local builds and packages for Debian6 100% complete.
WMS	100%	100%	N/A	Several dependencies are missing or packaged with different names in Debian6: Gridsite, Globus, gSOAP, Apache mod_ssl and fast-cgi.
WMS UI	100%	100%	N/A	
UNICORE U-EMIEX	100%	100%	100%	
UNICORE U-BES	100%	100%	100%	
UNICORE TSI6	100%	100%	100%	

Table 2: synoptic view of the status of the EMI-2 builds for the Compute Area components.

2.16. MONITORING PROBES

DNA1.3.3 ref X4; Due date: M26; Status: Partially delivered with EMI 2

In the reporting period, all the product teams involved in the Compute Area were able to include in their distribution the requested monitoring probes, as also testified by the numerous links provided. Whether pre-existing or newly developed, the teams were able to provide proactive maintenance and no significant incident was reported during this time in the EGI/NGI infrastructures. There are no deviations from the agreed plan of the past period to be reported.

References:

- https://savannah.cern.ch/task/?21823
- https://twiki.cern.ch/twiki/bin/view/EMI/NagiosProbes
- https://tomtools.cern.ch/confluence/display/SAM/WMS
- https://unicore-life.svn.sourceforge.net/svnroot/unicore-life/monitoring/
- https://tomtools.cern.ch/confluence/display/SAM/Probes#Probes-CREAMCE
- https://tomtools.cern.ch/confluence/display/SAM/CREAMCE+DJS

2.17. SEMI-AUTOMATED CONFIGURATION OF BACKENDS

DNA1.3.3 ref X13; Status: Delivered with EMI 2

This objective in its entirety is foreseen to be completed by month 30. In the reporting period, WMS has anyway introduced some semi-automatic tweaking of its database (i.e. MySQL), through its configuration module, yaim-wms, to obtain higher performance. These settings were usually

suggested to site administrators only in case of performance issues, but now are being provided as a default. There were no deviations from the agreed plan of the past period to be reported.

References:

- https://savannah.cern.ch/task/?func=detailitem&item_id=23845

2.18. INCREASE PERFORMANCE

DNA1.3.3 ref X15; Due date: M32; Status: Ongoing

For what concerns last year's planning, C12.1, A-REX better scalability for the job control directory, was addressed by splitting status information by state and limiting number of active tasks. C12.2 (better memory management in CREAM), was addressed by the relevant PT and subsequently cancelled as invalid. At the same time, C12.4 (performance of L&B WS interface) was closed for similar reasons. These were requests coming from the user community that were managed as part of support, so that they did not require any devoted development in JRA1. C12.3: WMS and DAG jobs. Document/certify throughput in terms of jobs/day. A document has been written together with the EGI administration personnel to identify and address the more frequently asked questions about the expected hardware configuration, tweaking and performance of an average WMS+LB installation (*). About C12.5, improve job submission rate by introducing parametric job descriptions in UNICORE XNJS, some of the received requests were addressed in release 6.4.2. The original requirements, in fact, were discussed and better assessed after a more careful analysis.

There were no significant deviations from the agreed plan of the past period nor incidents to be reported for this task.

References:

- https://savannah.cern.ch/task/?20356

- https://savannah.cern.ch/task/?24121

- https://wiki.egi.eu/wiki/WMS_best_practices (*)

2.19. EXTEND PARALLEL COMPUTING CAPABILITIES

DNA1.3.3 ref C13; Status: Delivered with EMI 2

This activity was originally planned for the third year. The reason why it is mentioned in this status report is that, after discussing the new requirements for the last year at the Project Technical Board, it was decided to cancel in favor of new and more concrete tasks that have superseded this task in priority.

3. COMPUTE AREA WORK PLAN

This chapter presents the Compute Area work plan for the third project year, that has been prepared to address the technical objectives stated in DNA1.3.3. These technical objectives have been created from requests and requirements gathering both from users communities, other projects and the EMI original planning itself.

One more year of experience on the field has also offered the opportunity to better identify and refine some of the previously defined objectives, so that their description might have been revised in order to concentrate the available effort on a more effective and pragmatic work. This is the reason why the objectives listed in this section might not necessarily match the ones described in the previous chapter, even if the the core activities stayed essentially the same. Each objective will be elaborated in terms of technical description – including harmonization aspects and affected key performance indicators - involved responsibilities and planning.

The components affected by this work are All EMI Compute Area components, as taken from DNA1.3.3, reported in Annex one.

3.1. CONSOLIDATION/HARMONIZATION OF COMPUTE AREA CLIENTS

DNA1.3.3 ref C10; Due date: M25; Status: Ongoing

Given the delays that this task was subject to due to lack of overall agreement, the PTB decided to manage this incident by creating a renewed task force and starting the discussion over again, so as to give each new and old member the possibility to mutually review their opinions about the consolidation plan. To start, an extended survey of existing client products, not only limited to CREAM, A-REX, UCC and not restricted to EMI products will be performed. This will be used as the basis for identifying potential overlaps. With this renewed scenario, partly known from the earlier work on harmonisation, possible developments for harmonization and consolidation will be described in a document to be submitted to the Project Technical Board for review. The risk is that finding an agreement on the whole issue might not be easy, given also the past experience, as . At least an agreement on a common API in the major language bindings is foreseen and encouraged to be feasible. Again, more risks associated with the objective are that none of the envisioned scenarios could be implementable by end of 2012. Relevant KPI is KJRA1.4: number of reduced released/ products.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C10.1	Define a plan for client consolidation of all EMI clients, as plus SAGA	Harmonization TF		M25

References:

- https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T2Compute_Client
- https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T2ComputeClientSurvey

3.2. COMMON PARALLEL EXECUTION FRAMEWORK

DNA1.3.3 ref C13; Status: Delivered with EMI 2

The final realization of the common parallel execution framework depends on the availability of the EMI-ES implementations. Once each stack the EMI-ES interface implemented, MPI-Start will be adapted for each stack to be used as *ParallelEnvironment*. The use of MPI-Start as the back-end will not be mandatory, but it will be ready for any site administrator to use it if it fits their resources. In the mean time, MPI-Start was modified in order to make it possible to use it as a UNICORE Execution Environment or ARC Runtime Environment (work presented on EGI UF 2011, see the reference). This task has always been quite ahead of time, and none or little risks are foreseen for its achievement. Relevant KPI is KJRA1.2: number of interoperable interface usage, given that the EMI-ES will be utilised.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C12.1	Adapt MPI-Start to interface with EMI-ES *ParallelEnvironment*	Glite MPI	C5.1	M32

References:

- https://www.egi.eu/indico/contributionDisplay.py?sessionId=13&contribId=28&confId=207

3.3. EMI-ES SERVER SIDE IMPLEMENTATION

DNA1.3.3 ref C5; Due date: M27; Status: Partially delivered with EMI 2

The remaining parts of the implementation of the present EMI-ES specification will have to be finalized by each Computing Service. A possible revision of the specification and consequent further developments should also be taken into account at this stage. It is in fact very unlikely that the specification will be fully suitable at its first release and the need for some updates already emerged from the first interoperability tests. Given all the past work that has been documented in the status report and the pace that the involved teams are sustaining towards the full implementation, no new impediments are foreseen (for example, the CREAM PT had initially to deal with the migration to Axis 2.0) and the objective is within reach. Needless to say, the deadline that has been set is short, for these very reasons; on the other hand, particularly demanding support activities might always interrupt this activity. Possible discussions about new upgrades of the specification might, in case, require sometime for proper evaluation. KPI is KJRA1.2: number of interoperable standard interfaces

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C5.1	Finalize EMI-ES implementation in A-REX.	ARC CE		M27
C5.2	Finalize EMI-ES implementation in CREAM	Glite Compute		M27
C5.3	Finalize EMI-ES implementation in UNICORE-EMIEX	UNICORE services		M27
C5.4	Finalize EMI-ES implementation in UNICORE TSI6	UNICORE services		M27

3.4. EMI-ES CLIENT-SIDE IMPLEMENTATION

DNA1.3.3 ref C6; Due date: M27; Status: Partially delivered with EMI 2

The remaining parts of the present EMI-ES specification will have to be finalized by each Computing client, similarly to what described in the previous paragraph. An utterly new task is instead the implementation of submission through the EMI-ES in the gLite WMS, that needs to be specifically planned afresh. Whether the EMI-ES API should be directly invoked in the WMS or a harmonized API is used, that is left to the team to choose.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C6.1	Finalize EMI-ES implementation in ARC WS compute CLI	ARC clients		M27
C6.2	Finalize EMI-ES implementation in CREAM CLI	CREAM		M27
C6.3	Finalize EMI-ES implementation in HILA	UNICORE clients		M27
C6.4	Implement submission through EMI-ES in WMS	Glite Compute		M32

3.5. GLUE 2.0 SUPPORT IN JOB MANAGEMENT SERVICES

DNA1.3.3 ref C8; Due date: M26; Status: Partially delivered with EMI 2

As described in paragraph 2.2, support for *GLUE2StorageService* will have to be profiled, working in cooperation with the data area teams, and finalized in the WMS match-making engine and in its clients, through appropriate handling of the Job Definition Language (JDL). The risk with not achieving this little remaining step towards the full implementation of the GLUE 2.0 specification in the WMS is that all the work that has been done until now will have to be considered of little use, as *GLUE2ComputingService* alone, even though far more complex than *GLUE2StorageService* in its structure, cannot considered to be usable without support for the latter. The deadline is short and an agreement for proper profiling in the Storage Elements could not be found. A relevant KPI addressed by this item is KJRA1.1: number of adopted open interface usage.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C8.1	Finalize GLUE 2.0 support for Storage Elements in WMS purchaser modules and in the client	Glite Compute		M26

3.6. ACCOUNTING RECORD IMPLEMENTATION

DNA1.3.3 ref C9; Due date: M27; Status: Ongoing

This task refers to the implementation of the specification for the EMI Compute Accounting Record (CAR) in each relevant sensor, as defined in paragraph 2.6. This task was originally planned in

DJRA1.1.2 to be completed by M22. Actually the plan, even if delivered on time, is still waiting for a formal endorsement by the PTB. Once this happens, implementation can start. For CREAM this will be managed by the APEL PT, which is a product f the infrastructure area. This activity represents an interesting opportunity for harmonising the EMI software stack at an operational level. The agreement that has been found was not made too distant from each existing implementation, so the risks for not fulfilling it are low, as this activity will mainly require some adaptations in the present code base of each service. A relevant KPI addressed by this item is KJRA1.2: number of interoperable interface usage.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C9.1	Implement EMI CAR in A-REX	ARC CE		M27
C9.2	Implement EMI CAR in L&B	L&B		M27
C9.3	Implement EMI CAR in UNICORE Computing Services	UNICORE services		M27
C9.4	Implement a producer of EMI CAR for use by CREAM	APEL PT		M27

3.7. SUPPORT FOR EXCLUSIVE NODE OR MULTI-CODE ALLOCATION

DNA1.3.3 ref C14; Due date: M32; Status: New year 3 objectiveTraditionally, most experiments, in particular WLCG, have been running their workloads using one core per job. This normally applies to both early- and late-binding jobs. However, these experiments have for some time required the possibility to run their jobs on more than one core. This has led in some cases to issues related to memory consumption. These issues can be mitigated if memory is shared across several cores on worker nodes. Currently a few sites are running prototypes to support multi-core jobs through dedicated “whole-node” queues, i.e. entry points where experiments would send jobs to, with the assurance that the jobs would run on hardware exclusively allocated to them. This solution has a few advantages: once a job has landed on a given node, it can simply look up how many cores there are, how much memory, etc.; the job will be able to manage the resources simply using the standard OS tools, without having to care about other jobs being running on the same node. However, from a site perspective, this solution often leads to resource partitioning: some resources are allocated to one or more “whole-node” queues, and cannot be used (without greatly sacrificing resource exploitation) by traditional single-core jobs. This partitioning may also be aggravated by the fact that systems have an increasingly number of CPU cores; therefore, a single system accounts for a relatively larger share of resources. Several sites in general, and sites serving more than one experiment in particular, rather prefer dynamic resource sharing making use of the appropriate scheduling mechanisms provided by the site LRMS (Local Resource Management System, or batch system: e.g., PBS, LSF, SGE, Condor). It will be anyway a site decision whether to offer a "strict whole node" solution, or rather generic multi-core access, but the middleware must be able to support it. With the rise of multi-/many- cores architectures and the experiments’ needs to limit the increased memory requirements that cannot be solved by running single core jobs on such CPUs, the risk with not achieving this task is that a growing part of the computing resources will be misused. For this reason, being far from representing a technical obstacle, it is important that this activity is well tracked and looked after by all the computing services in EMI. This work does not directly address any JRA1 KPIs.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C14.1	Provide support for multiple/exclusive node allocation in A-REX and all the EMI supported batch systems	ARC CE		M32
C14.2	Provide support for multiple/exclusive node allocation in CREAM and all the EMI supported batch systems	Glite Compute		M32
C14.3	Provide support for multiple/exclusive node allocation in TSI6 and all the EMI supported batch systems	UNICORE services		M32

3.8. PROCESS JOBS WITH DIFFERENT CHARACTERISTICS

DNA1.3.3 ref C15; Due date: M32; Status: New year 3 objective

Sites have used general purpose batch queues (LRMS) for many years to provide highest throughput and fairest allocation of resources between the communities that they serve. Many sites have also expressed interest with scheduling based on Resource Constraints, particularly storage. More options should be given to schedule resources as they assert that higher throughput could be achieved if jobs requirements are known better to the LRMS scheduler. I/O throughput on sites storage systems have an optimum load, and global throughput can be reduced when a load above a certain threshold is reached. Excessive Storage load could also lead to Storage system instability. For example a site may decide that by spreading users jobs on different nodes, the start up time is more staggered and so preventing resource contention on a single file. The introduction of pilot jobs has prevented scheduling decisions based on user name. Although user name only gave an indirect interpretation of job load, I/O-bound and CPU-bound tagging intends to better scheduling based on user name, by giving more useful information to the LRMS. These Resource Constraint Tags should be honoured by the CE and passed to the LRMS in an agreed way, allowing sites to customize scheduling if they so desire. Sites that do not want to make use of Resource Constraint Tags will not need to support them; furthermore, sites are encouraged to publish in the Information System the Tags that apply to a given queue, so that this mechanism will serve user needs other than being only at the advantage of the sites themselves. This will require some syncronisation among producers and consumers about the way the bits of information will have to be advertised and retrieved. The implementation risk is low. The risk of not having this task released in production in a short time is that many sites could not optimise job load to better reflect their resources, so making less efficient use of the available ones. This work does not directly address any JRA1 KPIs.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C15.1	A-REX must be able to forward resource constraint tags to the underlying batch system	ARC CE		M32
C15.2	CREAM must be able to forward resource constraint tags to the underlying batch system	Glite Compute		M32
C15.3	UNICORE must be able to forward resource constraint tags to the underlying batch system	UNICORE services		M32

3.9. INTEGRATED SOLUTIONS TO INTERFACE WITH BATCH SYSTEMS

DNA1.3.3 ref C7; Status: Delivered with EMI 2

As anticipated in paragraph 2.7, SLURM was added to the list of batch systems to be supported by the EMI project. While it is fully supported in A-REX and UNICORE already, CREAM will start supporting SLURM with the EMI-3 release and this requires planning a subtask for this activity, which is considered to be low risk, given the entity of the work. No KPIs are involved.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
C7.1	Implement full support to SLURM in CREAM, BLAH and CREAM information system modules	Glite Compute		M32

3.10. INCREASE PERFORMANCE

DNA1.3.3 ref X15; Due date: M32; Status: Ongoing

User requests about this task will arrive through the approved channels and be dealt with according to their severity and priority. The EMI task associated with this objective has not produced any specific tasks for Compute Area product teams. An activity for documenting the performance in WMS is still left open, in case the user community represented by EGI would like to be provided with more information with respect to the document that has been written in the second project year.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
X15.1	Document performance of WMS	Glite Compute		M32

3.11. EVOLVE EMI COMPONENTS

DNA1.3.3 ref X16; Due date: M32; Status: Ongoing

User requests about this task will arrive through the approved channels and be dealt with according to their severity and priority. For now, one activity to be planned is the ability to support logs to the L&B service from the Computing Elements. This was requested by experiments supporting pilot jobs and it is also required to provide a more detailed tracking in scenarios where the gLite WMS is involved, if the logs from the CE will be able to affect the overall Grid state machine. Implementation by the Computing Elements is not mandatory, but it is important that the L&B supports this feature. Another activity that requires planning for this task is the implementation of a highly available and fault tolerant solution for the CREAM CE. Many sites, especially those in the WLCG community, are requiring this feature (that they usually refer to as 'virtual CE'), because they need to deploy and

advertise one single CE, instead of many, for all the exposed queues without introducing neither a single point of failure nor having one single instance to sustain all the traffic to their sites (these are typically huge, 'Tier 1' sites) . This feature requires several CREAM instances be hidden by a single endpoint (that each rime corresponds to a different node by using well-known techniques such as DNS aliasing) that can work in cooperation and in a synchronised way on a common persistent job directory. This development is far from being trivial and represents one of the most notable features that CREAM will implement in the third project year. Not only this will allow more fault tolerant and load balanced instances of the CREAM service at sites, but it will also allow to run updates on single machines without service interruption, as the involved node will simply have to be removed from the pool for the required time. This work does not directly address any JRA1 KPIs. There are no major risks foreseen with this technical objective.

3.12. MIGRATION TO AUTHENTICATION LIBRARY

DNA1.3.3 ref X17; Due date: M32; Status: Starts in year 3

Each involved service and its clients should be able to handle authentication and all related operations (for example, security token handling for issuing and checking requests, checking of the credentials etc.) exclusively depending on the EMI authentication library. How backwards compatibility is addressed is left to each product team. This activity represents a notable step, together with the implementation of the EMI-ES, towards the unification of the middleware services in ARC, gLite and UNICORE; therefore, its achievement is of primary importance and the risk is that actual interoperability, for example between HPC and HTC, is hindered by the specific layer that each middleware implements for security. Relevant KPIs are KJRA1.2: number of interoperable interfaces usage and KJRA1.3: number of reduced lines of code.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
X17.1	Migrate the present authentication mechanisms to work with the EMI authentication library in A-REX.	A-REX		M32
X17.1	Migrate the present authentication mechanisms to work with the EMI authentication library in ARC clients.	ARC clients		M32
X17.2	Migrate the present authentication mechanisms to work with the EMI authentication library in CREAM.	Glite Compute		M32
X17.3	Migrate the present authentication mechanisms to work with the EMI authentication library in CREAM CLI and APIs.	Glite Compute		M32
X17.4	Migrate the present authentication mechanisms to work with the EMI authentication library in WMS.	Glite Compute		M32
X17.5	Migrate the present authentication mechanisms to work with the EMI authentication library in WMS CLI and APIs.	Glite Compute		M32
X17.6	Migrate the present authentication mechanisms to work with the EMI authentication library in L&B server.	L&B		M32
X17.5	Migrate the present authentication mechanisms to work with the EMI authentication library in all L&B clients.	L&B		M32
X17.6	Migrate the present authentication mechanisms to work with the EMI authentication library in UNICORE U-EMIEX.	UNICORE services		M32
X17.7	Migrate the present authentication mechanisms to work with the	UNICORE		M32

	EMI authentication library in UNICORE TSI6.	services		
X17.8	Migrate the present authentication mechanisms to work with the EMI authentication library in UCC.	UNICORE clients		M32
X17.9	Migrate the present authentication mechanisms to work with the EMI authentication library in 'UNICORE client libs'.	UNICORE clients		M32
X17.10	Migrate the present authentication mechanisms to work with the EMI authentication library in HiLA	UNICORE clients		M32

3.13. DELEGATION

DNA1.3.3 ref X3; Due date: M32; Status: Ongoing

Of course this task is mostly relevant to the security area: the agreement that was reached has clarified that gridsite [GRIDSITE] delegation will become the standard delegation method for X.509 credentials within the EMI stack (taking into account that UNICORE uses Explicit Trust Delegation for internal delegation of SAML assertions). This main objective stays open as a compute area task in case possible future activities with the OGF might affect in some way the compute area components. No activities need to be planned for now.

3.14. PORT, RELEASE AND SUPPORT ON AGREED PLATFORMS

DNA1.3.3 ref X7; Due date: M34; Status: Partially delivered with EMI 2

All the Compute Area components are required to perform this task according to the platforms that the project will require to support. Specific planning is not needed here, as the task affects all the services, components and libraries at large. The risk foreseen for this objective is that, similar to the experience of the release of EMI-2, porting to other platforms of the current build system is a non-trivial activity. Dealing with a totally different naming, packaging and versioning of dependencies has also created several delays, for example for what concerns the Debian port. Developers scheduled to start other work are delayed due to overrunning porting tasks. This work does not directly address any JRA1 KPIs.

3.15. IMPROVE USABILITY OF CLIENT TOOLS

DNA1.3.3 ref X8; Due date: M32; Status: Ongoing

The EMI Compute Area will make components easier to use, as properly pondered requests are received from the user community. At the time of writing this document, no specific requests to make products easier to use have been received. Risks are that this activity requires too much effort than the current workload can afford to be properly discussed and dealt with. Changes can be minor, only at cosmetic level, nonetheless requiring a non trivial effort to be solved, especially as compared to the added value that the EMI software is supposed to provide in other topics.

3.16. ADHERE TO OPERATING SYSTEM STANDARDS

DNA1.3.3 ref X6; Due date: M26; Status: Partially delivered with EMI 2

As shown in paragraph 2.15, all the Compute Area components comply to standard guidelines for the

currently supported Operating Systems. The work plan for this objective is that, as other operating systems platforms are added to the EMI release, the Product Teams will make sure their components will follow the corresponding guidelines. No new operating systems are foreseen at the time of writing. The risk is minimal, as the rate of adding operating systems to the EMI build has been very low and therefore this task cannot be seen to take up much effort. This work does not directly address any JRA1 KPIs.

3.17. EMIR ROLLOUT

DNA1.3.3 ref X18; Due date: M30; Status: New objective in year 3

The Compute Area servers and clients will be affected by the rollout of the EMI Registry, in that they will basically have to be integrated with it by allowing to publish services and querying for them. This impacts primarily the clients, especially wherever complex service discovery mechanisms have been implemented. In the gLite architecture, where a Grid meta-scheduler is foreseen as the main entry point for end users, service discovery is implemented in the WMS client, not in the CREAM CLI which is only meant for direct submission to an already available endpoint. Some adaptations are needed on the Compute services as well, mostly to register the service to EMIR, as what follows is expected be handled transparently by the EMIR architecture.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
X18.1	Integration with EMIR in A-REX	ARC CE		M32
X18.2	Integration with EMIR in libarcclient	ARC Clients		M32
X18.3	Integration with EMIR in WS compute CLI (arc*)	ARC Clients		M32
X18.4	Integration with EMIR in CREAM	Glite Compute		M32
X18.5	Integration with EMIR in WMS	Glite Compute		M32
X18.6	Integration with EMIR in WMS client	Glite Compute		M32
X18.7	Integration with EMIR in L&B server	L&B		M32
X18.8	Integration with EMIR in L&B clients	L&B		M32
X18.9	Integration with EMIR in UCC	UNICORE clients		M32
X18.10	Integration with EMIR in ‘UNICORE client libs’	UNICORE clients		M32
X18.11	Integration with EMIR in HiLA	UNICORE clients		M32

3.18. MANDATORY CONFIGURATION VARIABLES

DNA1.3.3 ref X19; Due date: M28; Status: New objective in year 3

This item is quite self-explanatory and refers to well written documentation that is able to provide service administrators with a detailed list comprising all the mandatory variables needed for properly

running the specific configuration tools that are shipped with each single service. It was raised by the EGI user community at large and needs a specific planning. Of course there is little risk with this activity, as all the bits of information are readily available. This work does not directly address any JRA1 KPIs.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
X19.1	Identification of the mandatory configuration variables in A-REX	ARC CE		M32
X19.2	Identification of the mandatory configuration variables in CREAM	Glite Compute		M32
X19.3	Identification of the mandatory configuration variables in CREAM Cluster	Glite Compute		M32
X19.4	Identification of the mandatory configuration variables in WMS	Glite Compute		M32
X19.5	Identification of the mandatory configuration variables in L&B	L&B		M32
X19.6	Identification of the mandatory configuration variables in UNICORE UAS-C	UNICORE services		M32
X19.7	Identification of the mandatory configuration variables in UNICORE U-EMIEX	UNICORE services		M32
X19.8	Identification of the mandatory configuration variables in UNICORE U-BES	UNICORE services		M32
X19.9	Identification of the mandatory configuration variables in WNoDES	WNoDES		M32

3.19. STANDARD SUPPORT FOR CRL HANDLING

DNA1.3.3 ref X21; Due date: M29; Status: New objective in year 3

This task is about checking that each service is able to import Certificate Revocation Lists from the EMI distribution and about checking the present behaviour with respect to 'live' re-loading without service interruption. Little risk is involved in this task. This work does not directly address any JRA1 KPIs. It has to be noted that the migration to the common authentication library, due by M32, will automatically solve this issue, as it will be able to transparently deal with CRLs.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
X21.1	Verify CRLs handling in A-REX	ARC CE		M26
X21.2	Verify CRLs handling in ARC gridftp jobplugin interface	ARC CE		M26
X21.3	Verify CRLs handling in CREAM	Glite Compute		M26
X21.4	Verify CRLs handling in WMS	Glite Compute		M26
X21.5	Verify CRLs handling in L&B	L&B		M26

X21.6	Verify CRLs handling in UNICORE TSI6	UNICORE services		M26
X21.7	Verify CRLs handling in UNICORE U-EMIEX.	UNICORE services		M26
X21.8	Verify CRLs handling in UNICORE U-BES.	UNICORE services		M26

3.20. SERVICE MIGRATION AND HOT SWAPPING

DNA1.3.3 ref X22; Due date: M32; Status: New objective in year 3

This activity is the result of the discussions about requirements that were raised by the EGI. In particular, the request was to have guidelines specifically addressing how to migrate a stateful service while preserving its state and, wherever it is supported, how to set up complex layouts comprising high-availability and load balancing at a site. In both cases, all EMI computing services are required by this task to produce clear documentation stating if they support service migration/hot swap or high-availability and load sharing, and, in case, what operations are required to a site administrator (i.e. what FS mounts need to be persistent, where paths are specified in the configuration, etc.) to install the required layout, to recover in case of failures and where special hardware is needed. This work does not directly address any JRA1 KPIs.

Definition of subtasks with owner and due date:

Subtask	Name	Owner	Not before	Due date
X22.1	Document service migration and hot swap in A-REX	ARC CE		M32
X22.2	Document service migration and hot swap in ARC gridftp jobplugin interface	ARC CE		M32
X22.3	Document service migration and hot swap in CREAM	Glite Compute		M32
X22.4	Document service migration and hot swap in WMS	Glite Compute		M32
X22.5	Document service migration and hot swap in L&B	L&B		M32
X22.6	Document service migration and hot swap in UNICORE TSI6	UNICORE services		M32
X22.7	Document service migration and hot swap in UNICORE U-EMIEX.	UNICORE services		M32
X22.8	Document service migration and hot swap in UNICORE U-BES.	UNICORE services		M32

3.21. DEFINE CONSOLIDATED API SET

DNA1.3.3 ref X23; Due date: M29; Status: New objective in year 3

This objective should result in a comprehensive set of APIs that a developer may use to code against the various EMI services and libraries. The APIs of the EMI Compute Area components are documented as a standard part of the release process. The process Quality Control will make sure that the minimal documentation set is present before a component is released. Currently there are no developer resources assigned to this task, apart from what overlaps with the consolidation task C10, that has been planned separately, therefore, no planning is needed.

4. CONCLUSIONS

This document described the status of the work performed by the EMI Compute Area during the second year of the project. The work plan for the second year was described in DJRA1.1.2. The status section generally follows the format of the first year work plan and also gives reasons if the work has not progressed according to the schedule. This document also described the third year work plan for the compute area components. Relevant decisions have been taken in order to allow for a definite and clear roadmap that the involved teams should commit to the best of their efforts. This plan is based on the best information available at the time of writing.

5. ANNEX 1: COMPUTE AREA COMPONENTS

Product	Element	PT	Type
ARC CE	ARC gridftp jobplugin interface	ARC CE	service
ARC CE	ARC LRMS modules	ARC CE	internal
ARC Clients	pre-WS compute CLI (ng*)	ARC Clients	client
ARC Clients	WS compute CLI (arc*)	ARC Clients	client
ARC Clients	libarcclient	ARC Clients	library
BLAH	none	Glite Compute	internal
CEMon	none	Glite Compute	Service, client
CREAM LSF module	none	Glite Compute	internal
CREAM SGE module	none	Glite Compute	internal
CREAM Torque module	none	Glite Compute	internal
CREAM SLURM module	none	Glite Compute	internal
CREAM	CREAM service	Glite Compute	service
CREAM	CREAM client	Glite Compute	client
EMI-UI	none	EMI common	client
EMI-NAGIOS	none	EMI common	client
EMI-WN	none	EMI common	client
gLite CLUSTER	none	Glite Compute	service
gLite-MPI	MPI-start	Glite MPI	internal
gLite-MPI	MPI-utils	Glite MPI	internal
L&B	LB server	L&B	service
L&B	LB client	L&B	client
TORQUE server config	Yaim-torque-server, yaim-torque-utils	EMI common	internal
TORQUE WN config	Yaim-torque-client, lcg-pbs-utils	EMI common	internal
UNICORE Client6	U. client libs	U. Clients	library
UNICORE Client6	ucc	U. Clients	client
UNICORE HILA	none	U. Clients	client
UNICORE TSI6	none	U. Services	internal
UNICORE/X6	XNJS	U. Services	internal
UNICORE/X6	UAS-C	U. Services	service
UNICORE/X6	U-BES	U. Services	service
UNICORE/X6	U-EMIEX	U. Services	service
WMS	WMS server	Glite Compute	service
WMS	WMS client	Glite Compute	client, library

Product	Element	PT	Type
ARC CE	ARC gridftp jobplugin interface	ARC CE	service
WNoDES	none	WNoDES	service

6. ANNEX 2: YEAR 3 COMPUTE AREA OBJECTIVES

#	Objective	Due	Priority	Not before	Involved components
C10	Consolidation of the compute area clients	M25	5		ARC clients, CREAM client, WMS client, UNICORE clients, SAGA, L&B clients
C12	Common parallel execution framework implementation	M32	3		MPI-Start
C5	Server-side implementation of EMI-ES	M27	5		A-REX, CREAM, U. EMI-ES, U. TSI
C6	Client-side implementation of EMI-ES	M27	4		WMS server
C7	EMI computing services should provide fully integrated solutions to interface with identified set of batch systems	M32	3		CREAM
C8	Glue 2.0 support in compute clients	M26	5		WMS client
C9	EMI accounting record implementation by compute services	M27	3		A-REX, CREAM, U. TSI
C11	Implement the EMI cloud strategy within compute area.				WNoDES
C14	Provide better support for jobs requiring exclusive node and/or multiple core allocation	M32	3		A-REX, CREAM, BLAH, U. TSI
C15	Ability to process job with different characteristics in a suitable environment	M32	3		A-REX, CREAM, BLAH, U. TSI

7. ANNEX 3: YEAR 3 CROSS AREA OBJECTIVES RELEVANT TO COMPUTE AREA

#	Objective	Due	Priority
X15	Increase performance of EMI services.	M32	3
X16	Evolve EMI components to meet specific user requests.	M32	3
X17	Complete the rewrite of components utilizing the new emi_authlib libraries.	M32	4
X3	Agreement on common EMI delegation method.	M32	4
X7	Port, release and support EMI components on identified platforms (full distribution on SL6 and Debian 6, UI on SL5/32 and latest Ubuntu).	M34	2
X8	Improve usability of client tools based on customer feedback by ensuring a) better more informative, less contradictory error messages b) coherency of commands line parameters.	M32	4
X18	EMIR rollout: service and consumer-side integration	M30	5
X19	Clear identification of the mandatory configuration variables	M28	5
X20	Make EMI products installable as a non-privileged user	M26	4
X21	Verify the standard support of CRL handling of EMI services	M29	3
X22	Support service migration and hot swapping	M32	3
X23	Define a consolidated EMI API set	M29	3

8. ANNEX 4: YEAR 3 COMPUTE AREA SUBTASKS

Subtask	Name	Owner	Not before	Due date
C5.1	Finalize EMI-ES implementation in A-REX	ARC CE		M27
C5.2	Finalize EMI-ES implementation in CREAM	Glite Compute		M27
C5.3	Finalize EMI-ES implementation in UNICORE-EMIEX	UNICORE servers PT		M27
C5.4	Finalize EMI-ES implementation in UNICORE TSI6	UNICORE servers PT		M27
C6.1	Finalize EMI-ES implementation in ARC WS compute CLI	ARC clients		M27
C6.2	Finalize EMI-ES implementation in CREAM CLI	CREAM PT		M27
C6.3	Finalize EMI-ES implementation in HILA	UNICORE clients PT		M27
C6.4	Implement submission through EMI-ES in WMS	WMS PT		M32
C7.1	CREAM to support SLURM	Glite Compute		M32
C8.1	Finalize GLUE 2.0 support for Storage Elements in WMS purchaser modules and in the client	WMS PT		M27
C9.1	Implement EMI CAR in A-REX	ARC CE		M27
C9.2	Implement EMI CAR in L&B	L&B		M27
C9.3	Implement EMI CAR in UNICORE Computing Services	UNICORE services PT		M27
C9.4	Implement a producer of EMI CAR for use by CREAM	APEL PT		M27
C10.1	Define a plan for client consolidation of all EMI clients, as plus SAGA	Harmonization TF		M25
C12.1	Adapt MPI-Start to interface with EMI-ES ParallelEnvironment	Glite MPI	C5.1	M32
C14.1	Provide support for multiple/exclusive node allocation in A-REX and all the EMI supported batch systems	ARC CE		M32
C14.2	Provide support for multiple/exclusive node allocation in CREAM and all the EMI supported batch systems	Glite Compute		M32
C14.3	Provide support for multiple/exclusive node allocation in UNICORE TSI and all the EMI supported batch systems	UNICORE services PT		M32
C15.1	A-REX must be able to forward resource constraint tags to the underlying batch system	ARC CE		M32
C15.2	CREAM must be able to forward resource constraint tags to the underlying batch system	Glite Compute		M32
C15.3	UNICORE must be able to forward resource constraint tags to the underlying batch system	UNICORE services PT		M32
X15.1	Identification of the mandatory configuration variables in A-REX	ARC CE		M32
X15.2	Identification of the mandatory configuration variables in CREAM	Glite Compute		M32

X15.3	Identification of the mandatory configuration variables in CREAM Cluster	Glite Compute		M32
X16.1	LB logs from the Computing Element	L&B		M32
X16.2	Implement High-Availability in CREAM	Glite Compute		M32
X17.1	Migrate the present authentication mechanisms to work with the EMI authentication library in A-REX.	A-REX		M32
X17.1	Migrate the present authentication mechanisms to work with the EMI authentication library in ARC clients.	ARC clients		M32
X17.2	Migrate the present authentication mechanisms to work with the EMI authentication library in CREAM.	Glite Compute		M32
X17.3	Migrate the present authentication mechanisms to work with the EMI authentication library in CREAM CLI and APIs.	Glite Compute		M32
X17.4	Migrate the present authentication mechanisms to work with the EMI authentication library in WMS.	Glite Compute		M32
X17.5	Migrate the present authentication mechanisms to work with the EMI authentication library in WMS CLI and APIs.	Glite Compute		M32
X17.6	Migrate the present authentication mechanisms to work with the EMI authentication library in L&B server.	L&B		M32
X17.5	Migrate the present authentication mechanisms to work with the EMI authentication library in all L&B clients.	L&B		M32
X17.6	Migrate the present authentication mechanisms to work with the EMI authentication library in UNICORE U-EMIEX.	UNICORE services		M32
X17.7	Migrate the present authentication mechanisms to work with the EMI authentication library in UNICORE TSI6.	UNICORE services		M32
X17.8	Migrate the present authentication mechanisms to work with the EMI authentication library in UCC.	UNICORE clients		M32
X17.9	Migrate the present authentication mechanisms to work with the EMI authentication library in ‘UNICORE client libs’.	UNICORE clients		M32
X17.10	Migrate the present authentication mechanisms to work with the EMI authentication library in HiLA.	UNICORE clients		M32
X18.1	Integration with EMIR in A-REX	ARC CE		M32
X18.2	Integration with EMIR in libarcclient	ARC Clients		M32
X18.3	Integration with EMIR in WS compute CLI (arc*)	ARC Clients		M32
X18.4	Integration with EMIR in CREAM	Glite Compute		M32
X18.5	Integration with EMIR in WMS	Glite Compute		M32
X18.6	Integration with EMIR in WMS client	Glite Compute		M32
X18.7	Integration with EMIR in L&B server	L&B		M32
X18.8	Integration with EMIR in L&B clients	L&B		M32
X18.9	Integration with EMIR in UCC	UNICORE clients		M32
X18.10	Integration with EMIR in ‘UNICORE client libs’	UNICORE clients		M32

X18.11	Integration with EMIR in HiLA.	UNICORE clients		M32
X19.1	Identification of the mandatory configuration variables in A-REX	ARC CE		M32
X19.2	Identification of the mandatory configuration variables in CREAM	Glite Compute		M32
X19.3	Identification of the mandatory configuration variables in CREAM Cluster	Glite Compute		M32
X19.4	Identification of the mandatory configuration variables in WMS	Glite Compute		M32
X19.5	Identification of the mandatory configuration variables in L&B	L&B		M32
X19.6	Identification of the mandatory configuration variables in UNICORE UAS-C	UNICORE services		M32
X19.7	Identification of the mandatory configuration variables in UNICORE U-EMIEX	UNICORE services		M32
X19.8	Identification of the mandatory configuration variables in UNICORE U-BES	UNICORE services		M32
X19.9	Identification of the mandatory configuration variables in WNoDES	WNoDES		M32
X21.1	Verify CRLs handling in A-REX	ARC CE		M26
X21.2	Verify CRLs handling in ARC gridftp jobplugin interface	ARC CE		M26
X21.3	Verify CRLs handling in CREAM	Glite Compute		M26
X21.4	Verify CRLs handling in WMS	Glite Compute		M26
X21.5	Verify CRLs handling in L&B	L&B		M26
X21.6	Verify CRLs handling in UNICORE TSI6	UNICORE services		M26
X21.7	Verify CRLs handling in UNICORE U-EMIEX.	UNICORE services		M26
X21.8	Verify CRLs handling in UNICORE U-BES.	UNICORE services		M26
X22.1	Document service migration and hot swap in A-REX	ARC CE		M32
X22.2	Document service migration and hot swap in ARC gridftp jobplugin interface	ARC CE		M32
X22.3	Document service migration and hot swap in CREAM	Glite Compute		M32
X22.4	Document service migration and hot swap in WMS	Glite Compute		M32
X22.5	Document service migration and hot swap in L&B	L&B		M32
X22.6	Document service migration and hot swap in UNICORE TSI6	UNICORE services		M32
X22.7	Document service migration and hot swap in UNICORE U-EMIEX.	UNICORE services		M32
X22.8	Document service migration and hot swap in UNICORE U-BES.	UNICORE services		M32