

EUROPEAN MIDDLEWARE INITIATIVE

TECHNICAL DEVELOPMENT PLAN

EU DELIVERABLE: DNA1.3.1

Document identifier:	EMI_DNA1.3.1_draft.doc
Date:	19/11/2010
Activity:	NA1 - Administrative and Technical Management
Lead Partner:	LU
Document status:	Final, (v0.15)
Document link:	https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDNA131

Abstract:

This deliverable provides the details of the technical development plan for all EMI services. The plan contains the details for the first year of development and longer-term high-level plans for the following years. It is revised periodically and at least every twelve months with increasing level of details. It is coordinated by the Technical Director, but requires input and active engagement from all WP leaders and Product Team leaders.

Copyright notice:

Copyright © Members of the EMI Collaboration, 2010.

See www.eu-emi.eu for details on the copyright holders.

EMI (“European Middleware Initiative”) is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 7th Framework Programme. EMI began in May 2010 and will run for 36 months.

For more information on EMI, its partners and contributors please see www.eu-emi.eu

You are permitted to copy and distribute, for non-profit purposes, verbatim copies of this document containing this copyright notice. This includes the right to copy this document in whole or in part, but without modification, into other documents if you attach the following reference to the copied elements: “Copyright © Members of the EMI Collaboration 2010. See www.eu-emi.eu for details”.

Using this document in a way and/or for purposes not foreseen in the paragraph above requires the prior written permission of the copyright holders.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE EMI_DS COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Delivery Slip

	Name	Partner / Activity	Date	Signature
From	Balázs Kónya	LU/NA1		
Reviewed by				
Approved by				

Document Log

Issue	Date	Comment	Author / Partner
V0.1	11.06.2010	Table of content was created	Balázs Kónya/LU
V0.3	02.07.2010	Input from L&B and glite security PT leaders; Data, Compute and Security area leaders	Balázs Kónya/LU
V0.4	08.07.2010	Added EMI component table, Infra area plans, some extra input to Data and Security area plans	Balázs Kónya/LU
V0.5	14.07.2010	Created the technical objectives table, updated the security area harmonization and evolution sections	Balázs Kónya/LU

v0.6	17.07.2010	Project technical objectives were collected and decided upon (objectives table) with the exception of the infrastructure area. Updated the EMI PT list with AMGA and Cesnet Security PTs	John White, Patrick Fuhrmann, Massimo Sgaravatto, Morris Riedel, Balázs Kónya
v0.7	21.07.2010	Filled the component table, removed placeholder section with raw PT input, removed duplicated entries from the objectives table	Balázs Kónya
v0.8	1.08.2010	Component table: corrections	Balázs Kónya (based on PT leader feedback)
v0.9	4.08.2010	Infrastructure area objectives, Component table corrections	Laurence Field, Patrick Fuhrmann, Massimo Sgaravatto, Morris Riedel, Balázs Kónya also corrections from PT leaders
v0.10	6.08.2010	Completed Executive Summary and Introduction chapters (Purpose, Amendment Procedure, Doc organization). Structural reorganization: removed tech area plans because those are already covered in the respective detailed workplan deliverables, moved Requirement section. Component table corrections	Balázs Kónya, also corrections from PT leaders
v0.11	4.10.2010	AMGA PT info, completed Requirements section, restructuring of the component and the objectives tables	Balázs Kónya
v.0.12	18.10.2010	Components and objectives modifications to reflect the outcome of the Geneva architecture meeting	Balázs Kónya and area leaders
v0.13	22.10.2010	Completed "High level view" section	Balázs Kónya
v0.14	28.10.2010	Corrections implemented after the final PTB call	Balázs Kónya
v0.15	18.11.2010	Revision addressing review notes. Addition of missed jobwrapper component and three new data objectives.	Balázs Kónya

Document Change Record

Issue	Item	Reason for Change
1		
2		
3		

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY	5
2. INTRODUCTION	6
2.1. PURPOSE	6
2.2. DOCUMENT ORGANISATION.....	6
2.3. DOCUMENT AMENDMENT PROCEDURE	7
3. REQUIREMENTS	8
4. EMI COMPONENTS AND PRODUCT TEAMS	9
5. TECHNICAL OBJECTIVES.....	16
5.1. HIGH LEVEL VIEW.....	16
5.2. DETAILED OBJECTIVES	18
6. REFERENCES	23
7. APPENDIX	24

1. EXECUTIVE SUMMARY

This deliverable presents the technical development plan for all EMI services. The document starts with the definition of the EMI software stack: an inventory of components and the product teams behind. The EMI component table (Section 4), listing 98 components and 29 product teams, serves as a key reference for the entire project. The table identifies and categorizes the initial EMI software portfolio by specifying the type, area and status and of every EMI component. Furthermore, the component table carries important information about the foreseen convergence of the EMI software stack by specifying which components are planned to be phased out, merged or further developed as part of the EMI harmonization strategy. This is provided via table columns showing the maintenance, harmonization, evolution and optional phase out plans for every component. Based on the component inventory, a technical workplan is formulated through the presentation of a high-level EMI vision (roadmap) that is supported and followed by the table of technical objectives (Section 5.2). For each objective a completion date and the list of associated components are given. The presented plan covers the entire project duration.

2. INTRODUCTION

2.1. PURPOSE

This deliverable (DNA1.3.1) outlines the Technical Development Plan of the EMI project. The main **purpose of the document** is to complement the EMI Description of Work [R1], to define the precise scope, to provide more accurately formulated goals for EMI development by presenting:

- definition of EMI software portfolio, the bases of the EMI technical development plan,
- the high-level vision of the evolution of the EMI software stack
- a comprehensive, technical formulation of development objectives coupled to components and delivery dates.

The DNA1.3.1 **document does not provide**:

- low level, implementation-specific details of the technical objectives, fine-grained, detailed development planning for EMI components and product teams
- standardization and integration plans
- a schedule of the development tasks formulated via a Release Plan
- software quality assurance aspects of the EMI development

The above listed areas are covered in their respective EMI workplan documents:

- DJRA1.1.2 - Compute area work plan and status report
- DJRA1.2.1 - Data area work plan and status report
- DJRA1.3.1 - Security area work plan and status report
- DJRA1.4.1 - Infrastructure area work plan and status report
- DJRA1.5.1 - Standardization work plan and status report
- DJRA1.6.1 - Integration work plan and status report
- DSA1.2 - Software Release Plan
- DSA2.1 - Software Quality Assurance Plan

The Technical Development Plan (DNA1.3.1) represents the approved development strategy for the EMI software stack, a plan that has been prepared within the EMI Project Technical Board (PTB) during the first five months of the project. Its content is highly relevant for the entire project and is of interest to all DCI projects, user communities, developers and operations teams. The content is revised periodically (DNA1.3.2, DNA1.3.3).

2.2. DOCUMENT ORGANISATION

The Technical Development Plan document is organized as follows:

- Chapter 1 provides the **Executive Summary**, the high-level description of the document.
- Chapter 2 **Introduction** is this section, explaining the purpose, scope and organization of the document.

- Chapter 3 is dedicated to **Requirements** which drove the creation of the EMI technical development plan.
- Chapter 4 contains the **EMI components and Product Teams** inventory.
- Chapter 5 **Technical Objectives** describes the EMI development plan on two levels by presenting both a higher level roadmap and a table with detailed development tasks.
- The last section, Chapter 6 contains the **References**.

Definitions of terms, acronyms, and abbreviations required to properly interpret the document are available online within the EMI project glossary [R3].

2.3. DOCUMENT AMENDMENT PROCEDURE

This document can be amended by the EMI Project Technical Board further to any feedback from other teams or people. Amendments, comments and suggestions should be sent to the PTB (ptb@eu-emi.eu).

Minor changes, such as spelling corrections, content formatting or minor text reorganisation not affecting the content and meaning of the document can be applied by the document editor, the EMI TD without peer review. Other changes must be submitted to peer review and to the EMI PTB 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.

The EMI Technical Development Plan is revised periodically (DNA1.3.2, DNA1.3.3).

3. REQUIREMENTS

The selection of the EMI software components and the EMI software development plan has been governed by the project main objective to support efficient, reliable operations of EGI, PRACE and other DCIs. Therefore, the initial requirements influencing the EMI project setup and the EMI DoW had been already provided during the project preparation phase by the *UMD Operation and user requirements Working Group* [R2]. The UMD requirements document collected preliminary wishes from developers, operations and user communities on missing functionalities and proposed a preliminary roadmap for the future evolution containing concrete ideas for middleware developments. In particular, the UMD requirement document identified the following common feature list requested by DCI user communities (see Page 8-9 of [R2]):

- continuous push for uniform interfaces to the resources
- support for a great variety of OS'es and OS versions
- tighter integration with the local batch system
- concise set of clearly specified APIs
- a coordinated infrastructure activity, potentially crossing all three infrastructures to address the administration of users, their authentication, authorization
- need for an accounting system, independent of the access method, but hierarchical and federated supporting most optimally the administrative domains in a common European grid infrastructure
- data management to cover system-wide virtual file hierarchy as well as transparent data storage and access with a high level of security (access control, encryption).
- data transfer capabilities between infrastructures is needed to ensure inter-operability between DCIs
- scientific portals as important user tools.

The selection of the EMI components (Section 4) and the technical development objectives (Section 5.2) addresses the UMD working group requirements¹.

This section, for the second year document, will contain requirements expected to be received from the major customers of EMI. EGI-Inspire, PRACE and other distributed computing infrastructure or user community projects will be able to communicate their requirements concerning EMI development through well-established collaboration channels. These requirements will be able to influence the developments leading to EMI-2 release scheduled by spring 2012.

¹ EMI is the main software provider for the Unified Middleware Distribution (UMD) software stack to be deployed within EGI, the flagship European Distributed Computing Infrastructure.

4. EMI COMPONENTS AND PRODUCT TEAMS

The EMI software stack currently consists of selected components provided by the ARC, gLite, UNICORE and dCACHE middleware consortia. Later, the EMI stack will also include some new components to be developed through a common effort. New component development is foreseen as part of harmonization, i.e. when a need for a common new service or library is identified. One possible example for such a new service is the EMI Service Registry. The maintenance and development of the EMI components (or products) is carried out by their respective Product Teams.

This section presents the inventory of the EMI software stack via the **EMI component table**. The table lists all the components of the EMI software regardless their current status or played role in the EMI development activities. Therefore, the table contains EMI components which are target for phase out, components which require only maintenance, components which will undergo extensive development and some planned new components. The table provides the organization of the EMI products into product teams as well.

One of the main goals of EMI project is to achieve a convergence of the components contributed by the middleware consortia. The starting set of EMI components contains numerous redundancies, overlapping solutions and non-interoperable components. During the course of the project the complexity will be reduced, unjustified redundancies and overlaps will be eliminated. The ultimate goal of the EMI harmonization development is to create an integrated software stack where only properly integrated and interoperable components are kept. The initial status assessment of the EMI software landscape with respect to the convergence and harmonization objective is presented in the table through the “Status” and “Maintenance and development” columns. The details of the harmonization plan are given per technical areas (see Section 2.1).

The EMI component table serves as an important reference point not only for the software development activities but also for the entire EMI project.

Table Legend:

Component: name of the component (product)

PT: name of the product team responsible for the component

Area: (C)ompute, (D)ata, (S)ecurity, (I)nfrastructure

Type: service, client, library, internal

Status: planned, alpha, beta, ready, (in-production)

Phase out: no, investigate (including the possibility for merge), yes

Maintenance: no (for early development components), support, pro-active

Harmonization: no, merge, integrate (implementation of the EMI agreement)

Evolution: no, yes

	Component	PT	Area	Type	Status	Maintenance and Development plan			
						Phase out	Maintenance	Harmonization	Evolution
1.	A-REX	ARC Compute Element	C	service	ready	no	Pro-active	integrate	yes
2.	ARC Grid Manager		C	service	in-prod	yes	support	no	no
3.	ARC gridftp jobplugin interface		C	service	in-prod	yes	support	no	no
4.	ARC CE-Cache		D	internal	in-prod	no	pro-active	no	yes
5.	ARC CE-staging		D	internal	in-prod	no	support	integrate	yes
6.	ARC LRMS modules		C	internal	in-prod	no	pro-active	no	yes
7.	JANITOR		C	internal	beta	no	pro-active	yes	yes
8.	JURA accounting hook		I	internal	beta	investigate	support	integrate	no
9.	pre-WS compute CLI (ng*)	ARC Compute Clients	C,I	client	in-prod	yes	support	no	no
10.	WS compute CLI (arc*)		C,I	client	in-prod	no	pro-active	integrate	yes
11.	libarcclient		C,I	library	in-prod	no	pro-active	integrate, merge	yes
12.	libarcdata2	ARC Data Libraries	D	library	in-prod	no	pro-active	integrate, merge	yes
13.	pre-WS data CLI (ng*)		D	client	in-prod	yes	support	no	no
14.	WS data CLI (arc*)		D	client	in-prod	no	support	integrate	yes
15.	ARC DMCs		D	internal	in-prod	no	support	integrate	yes
16.	ARC gridftp server	ARC Classic SE	D	service	in-prod	yes	support	no	no

	Component	PT	Area	Type	Status	Maintenance and Development plan			
						Phase out	Maintenance	Harmonization	Evolution
17.	LocalLDAP	ARC Information System	I	service	in-prod	investigate	pro-active	integrate	no
18.	EGIS		I	service	in-prod	investigate	support	integrate	no
19.	ARC Grid Monitor		I	client	in-prod	no	support	integrate	yes
20.	ARC infoproviders		I	internal	in-prod	no	support	integrate	yes
21.	update-crls	ARC Security Utils	S	internal	in-prod	yes	support	no	no
22.	nordugridmap		S	internal	in-prod	investigate	support	merge	yes
23.	arcproxy		S	client	in-prod	no	support	integrate, merge	yes
24.	HED	ARC Container	I	service	ready	no	pro-active	no	yes
25.	HED security		S	internal	ready	no	pro-active	integrate, merge	yes
26.	HED LIDI		I	internal	ready	no	pro-active	integrate	yes
27.	HED language bindings		I	internal	in-prod	no	pro-active	no	yes
28.	MPI-start	gLite MPI	C	internal	in-prod	no	support	merge integrate	yes
29.	MPI-utils		C	internal	in-prod	no	pro-active	merge, integrate	yes
30.	CREAM	gLite Job Management	C	service, client	in-prod	no	pro-active	integrate	yes
31.	BLAH		C	internal	in-prod	no	pro-active	integrate	no
32.	CEMon		C,I	service, client	in-prod	no	support	integrate	no
33.	jobwrapper		C	internal	in-prod	no	support	integrate	no
34.	WMS		C	service, client	in-prod	no	pro-active	integrate	yes
35.	FTS	CERN Data	D	service	in-prod	no	pro-active	integrate	yes
36.	DPM		D	service	in-prod	no	pro-active	integrate	yes
37.	LFC		D	service	in-prod	no	pro-active	integrate	yes

	Component	PT	Area	Type	Status	Maintenance and Development plan			
						Phase out	Maintenance	Harmonization	Evolution
38.	GFAL		D	library	in-prod	no	pro-active	merge, integrate	no
39.	lcg_util		D	client	in-prod	no	pro-active	merge, integrate	no
40.	dCache server	dCache	D	service	in-prod	no	support	integrate	yes
41.	dCache client		D	client	in-prod	no	support	integrate	yes
42.	StoRM SE	StoRM	D	service	in-prod	no	pro-active	integrate	yes
43.	AMGA server	AMGA	D	service	in-prod	no	support	integrate	yes
44.	AMGA client		D	client	in-prod	no	support	integrate	yes
45.	APEL parsers	APEL Client	I	internal	in-prod	no	support	integrate	no
46.	APEL publisher		I	client	in-prod	no	support	integrate	no
47.	HLR-Clients	DGAS Client	I	internal	in-prod	no	support	integrate	yes
48.	HLR-sensors		I	internal	in-prod	no	support	integrate	yes
49.	BDII	gLite Information System	I	service	in-prod	no	pro-active	integrate	yes
50.	Glue model		I	internal	in-prod	no	support	integrate	no
51.	gLite service info providers		I	internal	in-prod	no	support	integrate	no
52.	gLite site info provider		I	internal	in-prod	no	support	integrate	no

	Component	PT	Area	Type	Status	Maintenance and Development plan			
						Phase out	Maintenance	Harmonization	Evolution
53.	gstat-validation		I	client	in-prod	no	support	integrate	no
54.	lcg-info and lcg-infosites		I	client	in-prod	yes	support	no	no
55.	SAGA-SD	SAGA-SD-RAL	I	library	ready	investigate	support	integrate, merge	yes
56.	SAGA-ISN		I	library	ready	investigate	support	integrate, merge	yes
57.	VOMS	VOMS	S	service, client	in-prod	no	pro-active	integrate	yes
58.	VOMS-Admin		S	service	in-prod	no	pro-active	integrate	yes
59.	Trustmanager	gLite security	S	library	in-prod	investigate	support	integrate	no
60.	Util-Java		S	library	in-prod	investigate	support	integrate	no
61.	LCAS		S	internal	in-prod	investigate	support	no	no
62.	LCMAPS		S	internal	in-prod	investigate	support	no	no
63.	LCMAPS-plugins-c-pep		S	internal	in-prod	investigate	support	integrate	no
64.	gLExec		S	internal	in-prod	investigate	pro-active	no	no
65.	SCAS		S	service	in-prod	yes	support	no	no
66.	Hydra		S	service	in-prod	no	support	no	yes
67.	STS		S	service	planned	no	no	integrate	yes
68.	Delegation Java		S	library	in-prod	no	support	integrate	no
69.	SLCS	S	service	in-prod	no	support	integrate, merge	no	
70.	Pseudonymity	S	service	alpha	investigate	support	integrate	no	
71.	org.glite.security.gss	CESNET Security	S	internal	in-prod	yes	support	no	no
72.	org.glite.security.gsoap-plugin		S	internal	in-prod	yes	support	no	no

	Component	PT	Area	Type	Status	Maintenance and Development plan			
						Phase out	Maintenance	Harmonization	Evolution
73.	org.glite.security.proxyrenewal		S	internal	in-prod	investigate	support	integrate, merge	no
74.	org.gridsite		S	internal	in-prod	no	support	integrate, merge	no
75.	Argus	Argus	S	service	in-prod	no	pro-active	integrate	yes
76.	Argus-EES		S	service	beta	no	pro-active	integrate	yes
77.	L&B Server	L&B	C	service	in-prod	no	pro-active	integrate	no
78.	L&B Client		C	client	in-prod	no	support	integrate	no
79.	U. TSI	U. Target System	C,I	internal	in-prod	no	support	integrate	yes
80.	U. XNJS		C,I	internal	in-prod	no	support	integrate	yes
81.	UAS-C	U. WS Interfaces	C	service	in-prod	no	support	integrate	yes
82.	UAS-D		D	service	in-prod	no	support	integrate	yes
83.	U. BES		C	service	in-prod	no	pro-active	integrate	no
84.	U. Registry		I	service	in-prod	investigate	support	integrate	no
85.	CIP		I	service	in-prod	no	support	integrate	no
86.	UCC		C	client	in-prod	no	support	integrate, merge	no
87.	U. client libs	U. Client and APIs	C	internal	in-prod	no	support	no	no
88.	HILA		C	library	in-prod	no	support	integrate, merge	no
89.	UNICORE/X	U. Container	I	service	in-prod	no	support	integrate	yes
90.	WSRFLite		I	internal	in-prod	no	support	integrate	yes
91.	UNICORE Gateway	UNICORE Security	S	service	in-prod	no	support	integrate	no
92.	XUADB		S	service	in-prod	no	support	integrate	no

	Component	PT	Area	Type	Status	Maintenance and Development plan			
						Phase out	Maintenance	Harmonization	Evolution
93.	UVOS		S	service & client	in-prod	yes	support	no	no
94.	U. XACML Entity		S	internal	in-prod	no	support	integrate	no
95.	U. authorization data providers		S	internal	in-prod	no	support	integrate	no
96.	U. security libraries		S	library	in-prod	no	support	integrate, merge	yes
97.	EMI Service Registry	Registry	I	service	planned	no	no	integrate	yes
98.	EMI messaging layer	Messaging	I	library	planned	no	no	integrate	yes

5. TECHNICAL OBJECTIVES

This chapter defines the EMI development strategy: the presentation of the three pillars of the EMI development is followed by a high-level time-line view of the overall development plan. Finally, a table with the concrete technical objectives is provided.

The EMI software development is organized around the following three pillars:

Support existing DCI infrastructures by providing **re-active and pro-active maintenance** for software components used in production. Implement best-practice service-oriented procedures based on clear Service Level Agreements. Work out transition and phase out plans.

Harmonize and Integrate the EMI-0 software portfolio originating from the middleware consortia by removing duplications and simplifying usage and maintenance. The middleware components must be **consolidated and streamlined** by removing unnecessary duplication, replacing proprietary technologies with off-the-shelf and community supported technologies wherever possible and adopting either standard interfaces from well-established international collaborations or de-facto standards used by the majority of implementations.

Evolve the middleware by addressing the requirements of the growing infrastructures as they become more stable and pervasive. The focus is more on hardening the reliability of existing services, **evolving their operational capabilities** and addressing clear and present needs, rather than producing new prototypal technology to be deployed in a few years time. The development preferably should be based on existing code or off-the-shelf 3rd party solutions, this way avoiding the creation of yet another prototype-level solution.

5.1. HIGH LEVEL VIEW

As the ultimate result of the EMI software development activity, by the end of the project, EMI will deliver a high quality consolidated middleware distribution of modular interchangeable components with unified interfaces offering advanced functionalities. The *EMI-Final* software stack will deliver reliable interoperable solutions for the core capabilities needed to operate and manage a distributed computing infrastructure. In particular, EMI will provide services within the compute, data, security and infrastructure functionality areas. The EMI services will form an integrated ecosystem via the common security mechanisms, the information system backbone and the monitoring and instrumentation solutions. *EMI-Final* will also bring substantial simplification and streamlining into the current middleware landscape due to the harmonization and consolidation efforts and the removal of unnecessary overlaps and duplications.

The high level view of the EMI development roadmap is shown in Figure 1. The workplan is divided into three phases (years):

- The first phase of the development is marked as EMI-1. This development phase will deliver important technical agreements, consolidation plans, design and early prototypes and additional new capabilities for production ready components. The latter will be included into the EMI-1 release due April 2011.
- The second development phase, EMI-2, in addition to working out the consolidation plans for the security and information system components and delivering some design and prototypes will be the most intensive development phase resulting in numerous production ready features to be released under EMI-2 due April 2012.
- During the third, EMI-Final phase the work will focus on completing the consolidation plans and bringing the prototypes to production level. The phase will result with the EMI-3 (or Final) release due April 2013.

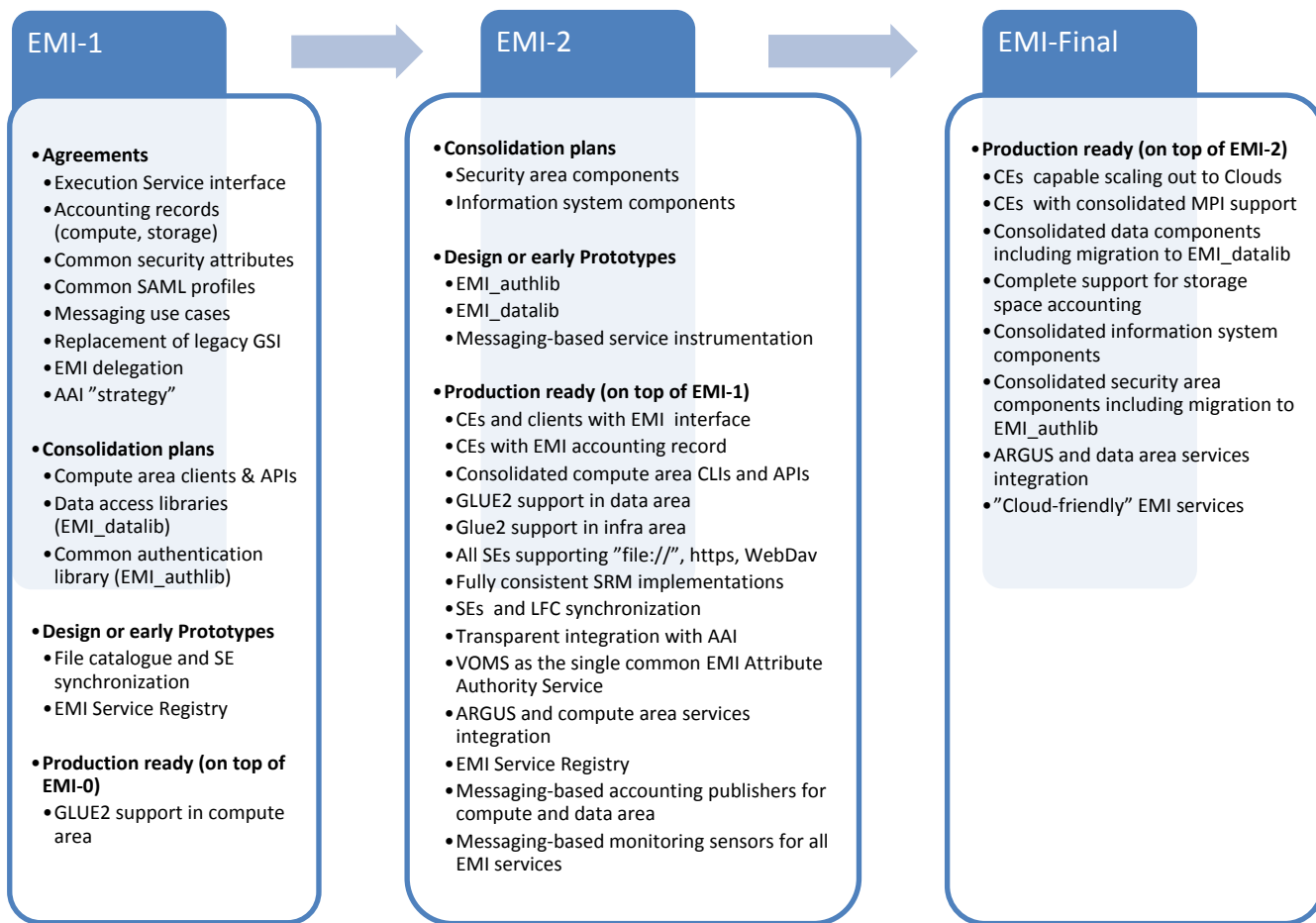


Figure 1: High level view of the main EMI development objectives.

5.2. DETAILED OBJECTIVES

The table presents the technical objectives of the EMI development plan. For each objective an implementation due date, expressed in project months, and the relevant EMI components are given. More precise formulation of the objectives is provided in the respective technical area development plans (see Section 2.1).

	Technical Objectives of Compute area	Due	Components
1.	Glue 2.0 support in job management services and client tools.	M12	A-REX, CREAM, U. TSI, U. XNJS, UAS-C, WSRFlite, WMS, libarcclient, arc*, UCC, HILA
2.	Implementation of the agreed common job submission and management methods in all the CEs and compute clients.	M18	A-REX, CREAM, U. TSI, U. XNJS, WSRFlite, U. EMIXS (new), libarcclient, arc*, CREAM client, UCC, HILA
3.	Provide limited interactive access for at least one EMI Computing element.	M18	A-REX, CREAM
4.	Support for the agreed compute accounting record (UR).	M18	CREAM, A-REX, UNICORE XNJS, TSI, UAS-C
5.	Consolidation and harmonization of compute area clients/APIs.	M24	libarcclient, arc*, CREAM client, WMS client, UCC and HILA
6.	Extend job definition language, resource information (GLUE model) and job management service capabilities so that EMI compute clients are able to request access to virtualized resource managers and appliances.	M24	A-REX, Janitor, WMS, CREAM, U. EMIXS (new), EMI compute clients
7.	Successful computational usage of emerging computing models i.e. clouds with EMI components (scaling out to clouds).	M30	A-REX, CREAM, U. TSI, U. XNJS, U. EMIXS,
8.	Provision of a common MPI execution framework, a “backend” across the different computing services to allow users to execute parallel applications in a uniform way.	M30	MPI-start, MPI-Utils, BLAH, CREAM, WMS, A-REX, U. TSI, U. XNJS, UAS-C
9.	Extend the parallel computing capabilities to better address multi-core jobs on all emerging architectures resources, multi-node execution on interconnected clusters; and special scenarios like advanced topologies, FPGAs, GPGPUs	M36	MPI-start, MPI-utils, BLAH, CREAM, WMS, A-REX, U. TSI, U. XNJS, UAS-C

	Technical Objectives of Data area	Due	Components
1.	All storage elements publishing initial GLUE 2.0 storage information.	M12	DPM, dCache, StoRM, UAS-D
2.	Using https instead of httpg for the SRM protocol as a prototype implementation in one storage element and client (library).	M12	dCache server plus one client
3.	All storage elements offering support for the http(s) protocol.	M12	dCache, StoRM, DPM
4.	All storage elements offering at least a prototype-level support for the "file://" access protocol.	M12	dCache, StoRM, DPM
5.	File Catalogue Access from UNICORE	M12	UAS-D
6.	One storage client is capable consuming GLUE 2.0 information published by storage elements.	M16	GFAL
7.	All storage elements publishing full set of GLUE 2.0 storage information and EMI clients are capable consuming that.	M24	DPM, dCache, StoRM, UAS-D, GFAL, libarcdata2
8.	Storage elements offering support for the WebDav protocol.	M24	dCache, StoRM
9.	Using https instead of httpg for the SRM protocol as a production implementation in all the storage elements and clients.	M24	All data clients and SEs
10.	Overall consolidation of data area by adopting a consistent interpretation of SRM.	M24	DCache, DPM, StoRM, FTS, GFAL, libarcdata2
11.	Providing a common set of data access libraries at least between gLite and ARC.	M24	GFAL, libarcdata2, <i>emi_data lib (new)</i>
12.	Solve the synchronization problem of the storage elements and the file catalogue.	M24	LFC, dCache, StoRM, DPM
13.	Integration of SRM-based access into UNICORE storage management	M24	UAS-D
14.	Completed migration to the common set of data access libraries.	M36	EMI data access clients
15.	Add support for storage space usage accounting on the SE/FTS side, including the refinement, definition and adoption (if/when applicable) of relevant standards.	M36	Dcache, Storm, DPM, UAS-D, FTS

	Technical Objectives of Security area	Due	Components
1.	Agreement on a minimal common set of security attributes to be used in policies.	M12	Argus, VOMS

	Technical Objectives of Security area	Due	Components
2.	Simplified management of security credentials by reducing the complexities of handling certificates and integrating different security mechanisms like Shibboleth and Kerberos across the EMI stack that allows users to use their own authentication system to access a ``Grid".	M18	STS, SLCS, VOMS
3.	Provide common authentication libraries supporting X.509 and optionally SAML.	M24	<i>Emi_authlib (new)</i>
4.	Consolidation and reduction in the number of security CLIs so that the users don't have to face the very different clients and utilities.	M24	EMI security clients and utilities
5.	Agreement and full support for a common single X.509 and SAML based Attribute Authority Service integrated with all EMI components.	M24	VOMS, UVOS
6.	Substantial simplification and reduction in the number of security area libraries, internal components and services.	M36	all security area services and internal components
7.	Provide a transparent solution for encrypted storage utilizing ordinary EMI SEs.	M36	Pseudonymity, Hydra

	Technical Objectives of Infrastructure area	Due	Components
1.	Provide early internal guidelines for integrating messaging into potential EMI target components.	M10	All EMI services and accounting sensors
2.	Design a common EMI service registry that is required in order to discover all the service endpoints of the different middleware components.	M10	<i>EMI registry (new component)</i>
3.	Investigate possible use cases for a common standard messaging system in the accounting area.	M12	APEL-publisher, DGAS HLR-sensors, JURA
4.	Investigate possible use cases for a common standard messaging system for the service monitoring and management.	M12	all EMI services
5.	Investigate possible use cases for a common standard messaging system for the information services and L&B.	M12	L&B, BDII, <i>EMI Registry (new)</i>
6.	Implement the common EMI Registry.	M24	<i>EMI Registry (new)</i>
7.	Fully utilize and support the GLUE2 information model.	M24	All EMI services, WMS, all infosys clients
8.	Provide guidelines for 3 rd parties to integrate messaging into their service/application based on the EMI experience.	M24	External products

	Technical Objectives of Infrastructure area	Due	Components
9.	Explore the modifications necessary in the EMI services to take advantages of the elasticity of the clouds resource management model while provisioning grid services within virtual machines (“grid in a cloud” scenario).	M24	all EMI services
10.	Implement or adapt the accounting record publishers of compute and data area services to use the common messaging system.	M24	DGAS HLR sensors, APEL-publisher, JURA, <i>new-publisher for data area services</i>
11.	Consolidation and reduction in the number of information system discovery APIs and CLIs.	M36	lcg-info, lcg-infosite, gstat-validation, GFAL, libarcclient, HILA, U. client libs, SAGA-SD, SAGA-ISN

	Cross area Technical Objectives	Due	Components
1.	Define the Information Flow architecture describing messaging and non-messaging based information exchange of the EMI components (e.g. service registry, information system, accounting, monitoring, and instrumentation). A common information exchange between the EMI components is preferable.	M9	all EMI services
2.	Investigate possible use cases for a common standard messaging system in the computing area.	M12	All EMI compute area services
3.	Investigate possible use cases for a common standard messaging system in the data area.	M12	All EMI data area services
4.	Evaluate integration scenarios with off-the-shelf computing cloud systems to be able to execute grid jobs on those (scaling out to clouds).	M12	A-REX, CREAM, WMS, UNICORE/X, U. TSI, U. XNJS, UAS-C
5.	An EMI-blessed delegation solution for at least the computing area.	M18	A-REX, CREAM, <i>U. EMIEX (new)</i>
6.	Definition and implementation of initial support for the common SAML profile all over the middleware stacks.	M18	A-REX, CREAM, CEMON, UNICORE services, UNICORE/X, ARGUS, VOMS, STS, SLCS, Hydra
7.	Integration of the compute area services with the ARGUS authorization framework.	M24	CREAM, CEMON, WMS, A-REX, HED, UNICORE/X, UAS-C, ARGUS
8.	Initial integration of the storage elements with the ARGUS authorization framework.	M24	DPM, dCache, StoRM, FTS, ARGUS

	Cross area Technical Objectives	Due	Components
9.	The legacy Globus security infrastructure (GSI) will be replaced with a common security solution based on TLS/SSL still including the delegation capability.	M24	DPM, dCache, StoRM, gfal, WMS, CREAM, A-REX, HED, libarcclient, libarcdata2
10.	Adapt or implement monitoring interfaces, sensors, providers for compute, data, security and infrastructure services to allow the use of standard monitoring tools preferably based on the common EMI messaging system.	M24	all EMI services
11.	Investigate service instrumentation interface for compute, data, security and infrastructure services, including remote configuration change and service management, utilizing the messaging system.	M24	all EMI services
12.	Complete migration to the new AuthN libraries.	M36	Dcache, DPM, StoRM, LFC, FTS, A-REX, HED, WMS, CREAM, CEMON, UNICORE services, U. gateway, U. sec. libs, ARGUS, VOMS, Hydra, SLCS, STS, Trustmanager, and all the corresponding clients

Further objectives from the EMI DoW that are categorized as low priority development targets:

- File Catalogue common front-ends, the availability of a standard interface to access the file catalogue, which simplifies the integration with user frameworks, portals and third party components.
- Improving usability by adding integrated support for high level gateways and portals.
- Improving manageability by providing standard service configuration for all EMI services.
- Extend interoperability between grids, supercomputers and emerging computing models like clouds and desktop grids.
- Provide a high-level application oriented client APIs whenever possible starting from existing community efforts like SAGA and provide practical and tested implementations across the supported middleware stacks.
- Agree upon and implement a standardized/common mechanism for obtaining service and resource information from Grid services (“local information”).
- EMI services provide command line and/or Web based service specific manageability (sysadmin toolbox) in addition to the messaging based common EMI monitoring solution.

6. REFERENCES

Table 1: Table of References

R 1	EMI Description of Work, public version, <i>https://twiki.cern.ch/twiki/pub/EMI/EmiDocuments/EMI-Part_B_20100624-PUBLIC.pdf</i>
R 2	UMD User and Operation Working Group Recommendations (2009), <i>https://twiki.cern.ch/twiki/pub/EMI/DeliverableDNA131/UMD-User-Operation-Req_draft.pdf</i>
R 3	EMI online glossary, <i>http://www.eu-emi.eu/about/glossary/</i>

7. APPENDIX

The appendix provides the inventory of EMI packages for every product team.

	Component	PT	Packages
1.	A-REX	ARC Compute Element	nordugrid-arc-nox-arex
2.	ARC Grid Manager		nordugrid-arc-grid-manager, arc-libs,
3.	ARC gridftp jobplugin interface		nordugrid-arc-gridftpd
4.	ARC CE-Cache		nordugrid-arc-grid-manager
5.	ARC CE-staging		nordugrid-arc-grid-manager
6.	ARC LRMS modules		nordugrid-arc-grid-manager
7.	JANITOR		nordugrid-arc-nox-janitor
8.	JURA accounting hook		nordugrid-arc-nox-arex
9.	pre-WS compute CLI (ng*)	ARC Compute Clients	nordugrid-arc-client, nordugrid-arc-libs
10.	WS compute CLI (arc*)		nordugrid-arc-nox-client
11.	libarcclient		nordugrid-arc-nox
12.	libarcdata2	ARC Data Libraries	nordugrid-arc-nox
13.	pre-WS data CLI (ng*)		nordugrid-arc-client, nordugrid-arc-libs
14.	WS data CLI (arc*)		nordugrid-arc-nox-client
15.	ARC DMCs		nordugrid-arc-nox-plugins-base, nordugrid-arc-nox-plugin-globus
16.	ARC gridftp server	ARC Classic SE	nordugrid-arc-gridftpd, -arc-libs

	Component	PT	Packages
17.	Classic Infoserver (localLDAP)	ARC Information System	nordugrid-arc-infosys-ldap
18.	Classic Infoindex (EGIIS)		nordugrid-arc-infosys-ldap
19.	Grid Monitor		nordugrid-arc-monitor
20.	ARC infoproviders		nordugrid-arc-infosys-ldap, -nox-arex
21.	update-crls	ARC Security Utils	nordugrid-arc-ca-utils
22.	nordugridmap		nordugrid-arc-gridmap-utils
23.	arcproxy		nordugrid-arc-nox-client
24.	HED	ARC Container	nordugrid-arc-nox, -hed,
25.	HED security framework		nordugrid-arc-nox, -hed
26.	HED LIDI		nordugrid-arc-nox, -hed,
27.	HED language bindings		nordugrid-arc-nox, -hed, -python
28.	MPI-start	gLite MPI	i2g-mpi-start
29.	MPI-utils		glite-MPI_utils
30.	CREAM	gLite Job Management	glite-ce-cream, -utils, glite-yaim-cream-ce, -client-api-c, -cli
31.	BLAH		glite-ce-blahp
32.	CEMon		glite-ce-monitor, -ce-plugin, -job-plugin, -monitor-client-api-c
33.	jobwrapper		glite-wms-helper, glite-cream-api-java
34.	WMS		glite-wms-utils-exception, -purger, -brokerinfo, -manager, -utils-classad, -utils-job, -ism, -configuration, -classad_plugin, -broker, -jobsubmission, -common, -helper, -wmproxy-interface, glite-yaim-wms, -wmproxy, -matchmaking, -ice, -ui-configuration, -wmproxy-api-cpp, -wmproxy-api-java, -api-python, -wmproxy-api-python, -ui-commands
35.	FTS	CERN Data Mana	glite-data-config-service, -delegation-api-c, -delegation-cli, -sd2cache, -srm-api-c, -srm-util-cpp, -srm2-api-c, -test-utils, -transfer-agents, -transfer-api-java, -transfer-cli, -transfer-fts, -transfer-interface, -transfer-load-generator, -transfer-monitor-gridview, -transfer-monitor-report, -transfer-monitor-schema, -transfer-proxyrenewal, -transfer-scripts, -transfer-url-copy, -util-c, glite-yaim-fts, -agents-common

	Component	PT	Packages
36.	DPM		DPM-DSI, DPM-dicom-copyd-mysql, DPM-dicom-server-mysql, DPM-dicom, LCG-DM-oracle, LCG-DM-py25, lcg-dm-common, glite-data-DPM, -DPM-client, -DPM-copy-server-mysql, -DPM-copy-server-oracle, -DPM-devel, -DPM-interfaces, -DPM-interfaces2, -DPM-libs, -DPM-name-server-mysql, -DPM-name-server-oracle, -DPM-perl, -DPM-python, -DPM-python25, -DPM-rfio-server, -DPM-server-mysql, -DPM-server-oracle, -DPM-srm-server-mysql, -DPM-srm-server-oracle, -LCGDM-devel, -LCGDM-libs, -dpm-httpd-cgi, -dpm-httpd-mod_dpmpu, -dpm-httpd-mod_keyauth, -dpm-httpd-service, -dpm-httpd-shell, glite-security-cgsi-gsoap, glite-yaim-dpm
37.	LFC		glite-data-LFC, -client, -devel, -interfaces, -interfaces2, -libs, -perl, -python, -python25, -server-oracle, -server-mysql, glite-yaim-lfc
38.	GFAL		gfal, glite-data-gfal-py25
39.	lcg_util		lcg_util, glite-data-dm-util-py25
40.	dCache server	dCache	dcache_server
41.	dCache client		dcache_SRM_client, dcap
42.	StoRM SE	StoRM	storm-backend-server, -backend-jars, -frontend-server, -checksum, glite-info-dynamic-storm
43.	AMGA server	AMGA	org.glite.amga.server
44.	AMGA client		org.glite.amga.client, org.glite.amga.api-java, org.glite.amga.api.python
45.	APEL parsers	APEL Client	apel-condor, apel-pbs, apel-sge, apel-lsf
46.	APEL publisher		apel-core, apel-publisher, apel-yaim
47.	HLR-Clients	DGAS Client	dgas-hlr-clients
48.	HLR-sensors		dgas-common
49.	BDII	gLite Information System	bdii, glite-yaim-bdii, bdii-config-site, bdii-config-top, glite-info-provider-ldap, glite-info-update-endpoints
50.	Glue model		glue-schema

	Component	PT	Packages
51.	gLite service info provider		glite-info-provider-service
52.	gLite site info provider		glite-info-site, glite-info-static
53.	gstat-validation		gstat-validation
54.	gLite info CLI (lcg-info, lcg-infosites)		lcg-info, lcg-infosites
55.	SAGA-SD	SAGA-SD-RAL	glite-saga-adapter-sd
56.	SAGA-ISN		glite-saga-adapter-isn
57.	VOMS	VOMS	org.glite.security.voms-server, voms-clients, voms-api-c, voms-api-cpp, voms-api-java, voms-api-noglobus, voms-config, voms-oracle, voms-mysql, voms-compatibility
58.	VOMS-Admin		org.glite.security.voms-admin-server, voms-admin-client
59.	Trustmanager	gLite security	glite-security-trustmanager
60.	Util-Java		glite-security-util-java
61.	LCAS		glite-security-lcas, -interface, -plugins-basic, -plugins-check-executable, -plugins-voms
62.	LCMAPS		glite-security-lcmaps, -plugins-basic, -plugins-verify-proxy, -plugins-voms, glite-security-saml2-xacml2-c-lib, glite-security-lcmaps-plugins-scas-client
63.	LCMAPS-plugins-c-pep		glite-security-lcmaps-plugins-c-pep
64.	gLEXec		glite-GLEXEC_wn, glite-security-glexec, glexec-wrapper-scripts
65.	SCAS		glite-SCAS, glite-security-scas, glite-security-saml2-xacml2-c-lib,
66.	Hydra		glite-data-hydra-cli, glite-data-hydra-service, glite-yaim-hydra
67.	STS		no packages yet
68.	Delegation Java		glite-security-delegation-java
69.	SLCS		glite-slcs-client, -slcs-server, -slcs-common
70.	Pseudonymity		glite-pseudo-server
71.	org.glite.security.gss	Ce sn et Se	glite-security-gss

	Component	PT	Packages
72.	org.glite.security.gsoap-plugin		glite-security-gsoap-plugin
73.	org.glite.security.proxyrenewal		glite-security-proxyrenewal
74.	org.gridsite		gridsite-all, -apache, -commands, -debuginfo, -devel, -gsexec, -service-clients, -services, -shared
75.	Argus	Argus	glite-ARGUS, -authz-pap, -authz-pdp, -authz-pepd, -authz-pep-c, -authz-c-cli, -yaim-argus_server
76.	Argus-EES		glite-security-ees
77.	L&B Server	L&B	glite-lb-build, -common, -doc, -glite-LB, -logger, -notif-logger, -proxy, -server-bones, -server, -state-machine, -types, -ws-interface, -ws-test, -yaim, glite-jobid-api-c, -api-cpp, -api-java, glite-lbjp-common-db, -common-jp-interface, -common-log, -common-maildir, -common-server-bones, -common-trio
78.	L&B Client		glite-lb-client, -common, -utils, -state-machine, -harvester, -client-interface
79.	U. TSI	U. Target System	unicore-tsi
80.	U. XNJS		unicore-unicorex
81.	UAS-C	U. WS Interfaces	unicore-unicorex
82.	UAS-D		unicore-unicorex
83.	OSGA-BES		unicore-unicorex
84.	Registry		unicore-registry
85.	CIP		unicore-cip
86.	UCC	U. Client and APIs	unicore-ucc
87.	U. client libs		unicore-uas-client
88.	HILA		unicore-hila
89.	UNICORE/X	U. Container	unicore-unicorex
90.	WSRFLite		unicore-wsflite
91.	UNICORE Gateway	UNICORE Security	unicore-gateway
92.	XUADB		unicore-xuadb-server, -admin-client
93.	UVOS		uvos-common, -client, -server, -rcpapp, -rcpcore

	Component	PT	Packages
94.	U. XACML Entity		uas-core
95.	U. authorization data providers		uas-authz
96.	U. security libraries		xfire-voutils, xfire-secutils, xfire-secutilsWithDSig, securityLibrary, samly2, samlTypes, crl-checking
97.	EMI Service Registry	Registry	no packages yet
98.	not yet know	Messaging	no packages yet