Towards next generations of software for distributed infrastructures : the European Middleware Initiative

Cristina Aiftimiei[††], Alberto Aimar[†], Andrea Ceccanti[*], Marco Cecchi[*], Alberto Di Meglio[†], Florida Estrella[†],
Patrick Fuhrmann[‡], Emidio Giorgio[*], Balázs Kónya[§], Laurence Field[†], Jon Kerr Nilsen[¶], Morris Riedel[||], John White[**]

[††]National Institute of Nuclear Physics, INFN - Padua, on leave from NIPNE-HH (Romania)
email:cristina.aiftimiei@pd.infn.it
[*]National Institute of Nuclear Physics, INFN
email:{cristina.aiftimiei@pd, andrea.ceccanti@cnaf, marco.cecchi@cnaf, emidio.giorgio@ct}.infn.it
[†]European Center for Nuclear Research, CERN
email: {alberto.aimar, alberto.di.meglio, florida.estrella, laurence.field}@cern.ch
[‡]German Electron Synchrotron, DESY
email: patrick.fuhrmann@desy.de
[§]Institute of Physics, Lund University
email:balazs.konya@hep.lu.se
[¶]Department of Physics, University of Oslo
email:j.k.nilsen@fys.uio.no
[||]Juelich Supercomputing Centre, FZJ
email:m.riedel@fz-juelich.de
[**]Helsinki Institute of Physics
email:john.white@cern.ch

***Abstract*—The last two decades have seen an exceptional increase of the available networking, computing and storage resources. Scientific research communities have exploited these enhanced capabilities developing large scale collaborations, supported by distributed infrastructures. In order to enable usage of such infrastructures, several middleware solutions have been created. However such solutions, having been developed separately, have been resulting often in incompatible middleware and infrastructures. The European Middleware Initiative (EMI) is a collaboration, started in 2010, among the major European middleware providers (ARC, dCache, gLite, UNICORE), aiming to consolidate and evolve the existing middleware stacks, facilitating their interoperability and their deployment on large distributed infrastructures, establishing at the same time a sustainable model for the future maintenance and evolution of the middleware components. This paper presents the strategy followed for the achievements of these goals : after an analysis of the situation before EMI, it is given an overview of the development strategy, followed by the most notable technical results, grouped according to the four development areas (Compute, Data, Infrastructure, Security). The rigorous process ensuring the quality of provided software is then illustrated, followed by a description the release process, and of the relations with the user communities. The last section provides an outlook to the future, focusing on the undergoing actions looking toward the sustainability of activities.**

I. Background

European scientific research has benefited in the past two decades from the increasing availability of computing and data infrastructures that have provided unprecedented capabilities for large scale distributed scientific initiatives. A number of major projects and endeavours, like EGEE[**?**], DEISA[**?**], WLCG[**?**], NDGF[**?**], OSG[**?**] and others, had been established within Europe and internationally to share the ever growing amount of computational and storage resources. This collaborative effort involved hundreds of participating research organizations, academic institutes and commercial companies. The major outcome was a number of active production infrastructures providing services to many research communities, such as High Energy Physics, Life Sciences, Material Science, Astronomy, Computational Chemistry, Environmental Science, Humanities and more.

At the core of these rich infrastructural facilities lies the grid middleware, a set of High Throughput Computing (HTC) and High Performance Computing (HPC) software services and components that enable the users to access the distributed computing infrastructures: manage data resources, discover services, execute jobs, collect results and share information.

The idea of distributed computing infrastructures and the underlying middleware followed the rapid development in internet technologies and can be considered an indirect result of internet standardization. Shortly after TCP/IP was accepted as a standard in the late eighties of the last century, the first approaches to harness distributed and yet interconnected computing power, were made. The Condor project[**?**] is often considered as a precursor, by setting in 1988 the goal of creating software that enabled a new type of computing environment with heterogeneous distributed resources. Legion/Avaki[**?**] has provided a wide-area distributed computing solution since

1993, and SETI@Home has released a distributed desktop resource scavenging service 1993 followed by BOINC[?] in 2002. While conceptually very sound, these approaches lack two key features: security and an information system. These features were added years later in the Globus Toolkit[?], which pioneered the idea of a single sign-on to the computing infrastructure. The Globus Toolkit became the first true open source Grid software. Still, being a toolkit, Globus did not offer a turnkey solution; the UNICORE project[?] in 1997 started developing the first European software that provided end-to-end services as an alternative to Globus in the area of high-performance computing.

The attraction of the Grid vision was its decentralized nature based on open protocols and services. This allows for cross-domain resource sharing as opposed to resource allocation in traditional high-performance computing. This proved to be particularly appealing for large distributed research communities in need of high-throughput computing services: physicists working on the Large Hadron Collider at CERN were the first to come with the idea of a worldwide computing Grid, WLCG5, in 2003, and are still the single heaviest Grid user community.

After the necessary initial period of research and consolidation of the early middlewares, a handful of production quality solutions emerged. In Europe, middleware like gLite[?] from the EGEE project, ARC[?] from the NorduGrid Collaboration, UNICORE and dCache[?] allowed thousands of scientific researchers to access grid enabled resources and produce scientific results.

UNICORE originally was conceived as a secure, unified and seamless interface to high performance computing resources, fitting neatly and non-intrusively into the existing infrastructure and administrative procedures at a large HPC site. UNICORE also came integrated with workflow capabilities allowing end users to tackle complex problems using the Grid without the need to resort to custom tools.

ARC was designed and implemented as a reliable, efficient, highly portable and easy-to handle middleware. It was optimized for serial data-intensive computational tasks, such that input and output data manipulation is considered an integral part of a computing service.

The gLite middleware stack was a modular ensemble of components that, in general, comprises a client, a job brokering system, a computing element and execution (or worker) nodes and data storage and management services. Integral to gLite are the security and information systems that protect the infrastructure from misuse and damage and provide information on capacity and usage respectively.

dCache is a data management technology designed for storing, retrieving and managing huge amounts of data, distributed among a large number of heterogeneous server nodes, under a single virtual file system tree. A variety of standard methods is offered to access and manage data.

Historically the above middleware stacks had been developed simultaneously and even though there was overlap in their capability domains, the delivered solutions were not compatible, thus usage of these frameworks created isolation of the infrastructures and separation of the user communities. A clear need for interoperability and standardization-based convergence appeared. The growing usage of these software solutions required the transformation of the fragmented European middleware landscape into a harmonized software infrastructure based on professionally managed and standardized services. EMI was the first project proposed to bring together four major European grid middleware providers, ARC, gLite and UNICORE and dCache in order to capitalize on their long history of competitive development, which has shaped their approaches, but also contributed to the overall quality and clear understanding of key grid propositions and problems.

By the start of the EMI project it became obvious that a number of problems that still prevented users from easily accessing and using the existing computing infrastructures must be urgently addressed:

- Usability had to be enhanced, removing redundancy and consolidating the services, simplifying the security management without compromising its strength
- Compatibility had to be improved by removing proprietary interfaces in the middleware services and ensuring true interoperability through the adoption of agreed community standards, thus leading to the removal of implementation islands.
- Manageability had to be improved by providing standard service configuration, monitoring and making accounting and other operational information more readily accessible.
- Sustainability was an issue that had to be improved by establishing long-term collaboration programs and turning to open source models.

II. The EMI impact

EMI aims to deliver a consolidated set of middleware components for deployment in EGI [?] and other distributed computing infrastructures, extending the interoperability and integration between grids and other computing infrastructures, establishing a sustainable model to support, harmonize and evolve the middleware, ensuring it responds effectively to the requirements of the scientific communities relying on it. The growing availability and use of compute and data infrastructures now requires their transformation into a professionally managed and standardised service. It is of strategic importance for the establishment of permanent sustainable research infrastructures to lower the technological barriers still preventing resource owners and researchers from using grids as a commodity tool in their daily activities. The EMI development roadmap is contributing to the realisation of this vision, organizing the software development into the following pillars:

1) Support existing infrastructures by providing reactive and proactive maintenance for software components used in production.
2) Implement best-practice, service-oriented procedures based on clear Service Level Agreements (SLA) and

work out transition and phase-out plans. Harmonize and consolidate the software portfolio originating from the middleware consortia by removing duplications and simplifying usage and maintenance. The EMI software stack must be consolidated and streamlined by removing unnecessary duplication, replacing proprietary technologies with off-the-shelf and community supported technologies wherever possible. The harmonization should be carried out via the adoption of either standard interfaces from well- established international collaborations or interfaces defined via EMI agreements reflecting defacto standards used by the majority of implementations.

3) 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, implementing new requested features 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.

As the ultimate result of the EMI software development activity, by the end of April 2013, EMI will deliver a high quality consolidated middleware distribution of modular intercompatible components with unified interfaces offering advanced functionalities that can be swapped depending on what kind of feature set is needed. The EMI-Final software stack will consist of reliable and interoperable solutions for the core capabilities needed to operate and manage a distributed computing infrastructure. In particular, EMI will provide services within the compute and data functionality areas, forming an integrated ecosystem via the common security mechanism and the information system backbone. On the user-side simplified management of security credentials will hide the complexity of Grid security and considerably lower the entry level barrier, while usability and maintainability will be improved through the unification of user interfaces, APIs, error messages, installation mechanisms and management procedures.

The development roadmap is divided into three phases (years):

1) The first phase of the development is marked as EMI 1. This phase was completed with the Kebnekaise release delivered on 12 May 2011 [?]. During the first EMI development phase important technical agreements, component design and early implementations were delivered in addition to the enormous integration efforts that had been deployed for EMI 1 release preparation. Among these efforts, are worth mentioning the adherence, for all EMI packages, to Filesystem Hierarchy Standard (FHS, [?]), and the adoption of Fedora [?] and EPEL [?] packaging guidelines. Endorsing these well known open source practices marked a significant step toward standardization, as well as grid middleware usability and sustainability. Furthermore, most of the Kebnekaise products came with numerous improvements as a result of individual components evolution.
2) The second development phase, leading to the EMI 2 release [?], completed the work on the four area consolidation plans (data, security, compute and infrastructure) that already had started back in year 1. The second phase constituted the most development intensive period of the EMI project. This was the phase that delivered harmonized solutions based on the first year agreements, products such as the EMI Execution Service Interface implementations, the EMIR service and the CANL security library.
3) During the third and final phase, the work will focus on completing all the open development tasks and thus bring the three-year EMI developments to production level. Not yet released products such as STS and the EMI datalib will be the new product highlights of the third release. Apart from these, the final phase development objectives are mostly targeting hardening of existing EMI features, improving non-functional aspects such as reliability, usability and interoperability. Another important objective is the roll-out of the latest EMI products with the rest of the EMI software portfolio. The broad usage of the EMIR information index service and the migration to the CANL EMI security library are such planned activities.The phase will conclude with the Monte Bianco release due February 2013.

In what follows, the planned and already ongoing development work of the final two phases is presented along the four technical areas of security, infrastructure, compute and data.

A. Security

Security area includes those services and components enabling the Grid security model, allowing the safe sharing of resources on a large scale. Such services cover identity management, Virtual Organization membership management, authentication, delegation and renewal of credentials, and authorization. Grid security has been largely based on a Public Key Infrastructure [?]. Although strongly reliable, production usage of PKI based mechanisms have shown in the years some limitations, in particular on the usability aspects, that eventually became barriers preventing wide usage of grid infrastructures [?]. On another side, the middleware flavours in EMI implemented slightly different security frameworks. Therefore, the overall strategy in EMI-Security was twofold : firstly, to integrate the interoperable services of each middleware, in order to reduce the duplications and paving the way for an unification of the security models. At the same time, simplification and ease of access have been taken into account, pursuing interoperability with well established security frameworks as Kerberos [?] and Shibboleth[?].

1) Common SAML and XACML profiles: Specification of common authorization policies, formerly managed through different, often non-standard services, was a mandatory step pursuing interoperability of security services. In order to have a common attribute authority that issues attributes either to X.509 proxies or SAML assertions, a SAML common profile was needed. A first agreement on a common set of SAML authorization attributes was planned and delivered by the EMI SAML group [?]. For the same reasons, a common XACML profile[?] has been produced, in order to define a minimal common set of security attributes to be used in policies. The work performed for this objective started with a general agreement that Argus will become the common authorization system for the middleware stacks in EMI. The XACML group thus

- Gathered the XACML profile requirements of the different Compute Elements.
- Determined the work needed to modify/extend the current (CREAM) CE XACML profile.
- Clarified that the full XACML specification is met within Argus.
- Attempted to collect requirements from the EMI Execution Service (EES) and Data Management.

2) Security Token Services: A key aim in the EMI Security Area is to make the security credential management more accessible to ordinary users. This is to be achieved by simplifying the management of security credentials by reducing the complexity of handling certificates and transparently integrating different security mechanisms such as Shibboleth and Kerberos into the EMI stack. This development will allow users to use their own authentication system to access a 'Grid'. In order to enable this access, a new security service, the Security Token Service (STS) is needed to translate these external credentials into the X.509 credentials needed by most Grid infrastructures.

The STS implements the service defined by the WS-Trust specification. STS is a Web service that issues security tokens, a collection of claims, for the authenticated clients. As the clients can authenticate to the service using different security token formats, the service can be seen as converting a security token from one format into another. The current plan is that the STS will be implemented on top of the upcoming Shibboleth IdP version 3 and OpenSAML3 implementations. The advantage of reusing the Shibboleth / OpenSAML3 code base is twofold, as it allows the reuse of non-trivial Web service libraries, permitting at the same time to leverage the STS from the beginning against the code base of the most used AAI system in Europe.

3) EMI Common Authentication Library: The security area consolidation of the EMI products is driven by the definition, implementation and migration over to the common EMI authentication library (CANL). EMI has provided common authentication libraries supporting X.509 and optionally SAML and is available for Java, C and C++. The implementations of the libraries have almost completed and prototype versions of CANl for all the three languages were included into the EMI Matterhorn release. Adoption of the library in UNICORE already started, at first the UNICORE Gateway and UNICORE security libraries are updated to use the new library. The adoption of the C library by other EMI components is expected during the EMI 3 phase. EMI security is built around X509 technology where proxies play a central role. Therefore, a considerable part of the security area development targets proxy functions, in particular proxy handling features to address SHA2 signing, default key-size and OCSP requests. Grid sites will start to receive certificates and proxies from Certificate Authorities that will be signed with a SHA2 hash rather than the current SHA1 or even MD5. The possibility to configure a default key size for generated proxies is required as currently this is not universally enforceable. The Online Certificate Status Protocol (OCSP) is an Internet protocol,, alternative to static revocation lists, used for obtaining the revocation status of an X.509 certificate. In order to provide a common solution, all these proxy handling features will be implemented in the Common Authentication libraries (CANl). Subsequent usage of these common libraries by other EMI components will automatically provide these needed proxy features.

4) Encrypted Storage: The security area is involved in providing a transparent solution for encrypted storage utilizing ordinary EMI Storage Elements. The realization of an encrypted storage within EMI is relatively simple: the necessary services to protect data and user identities on a Grid have to be provisioned. These services are requested by user communities that have stringent data protection requirements. EMI offers the pseudo-anonymity (pseudonymity) service and the key storage service (Hydra) as a solution. Pseudonymity is a certified EMI product, while Hydra is undergoing certification and release for EMI-2.

B. Infrastructure

The Infrastructure Area embraces a wide set of topics, from information services and service monitoring to client-side accounting and messaging technology. Information services enable users, applications and other services to discover which services exists in a infrastructure along with further information about their structure and state. It is clear how convergence on this aspect was crucial to achieve the overall harmonization and standardization goals pursued by the project. For this reason, a strong focus has been placed on harmonization and interoperability of the EMI middleware components [?], developing on this purpose several new products.

1) GLUE2.0 support: To ensure interoperability between the different EMI middleware components, it was agreed to adopt the GLUE 2.0 information model from the Open Grid Forum (OGF) [?] as a common information exchange model. Support for GLUE 2.0 within the information services themselves was delivered with the EMI 1 release and in the EMI 2 release, all EMI services publish GLUE 2.0 based information. By the EMI 3 release, it is anticipated that all EMI services make use full use of the GLUE 2.0 information model.

2) EMI Resource Information Service: The EMI Resource Information Services (ERIS) has been introduced into the EMI middleware stack as a common interface for obtaining information directly from services themselves and is one of the major results from the harmonization activity. The ERIS provides an LDAP v3 interface to GLUE 2.0 information. Information providers, in the classic sense, extract information from the under laying Grid service and provide GLUE 2.0 information in the LDIF format. These two together represent the external and internal interfaces for obtaining local information from EMI services. This approach aims to be a minimal-cost solution for existing EMI products and has a low-impact on existing infrastructures. The definition of this common interface not only ensures interoperability between the EMI middleware components, the provided implementation also reduces duplication of functionality an hence simplifies the EMI middleware stack.

3) EMI registry: Another major result achieved has been the development of a common service registry (EMIR), enabling the discovery of service endpoints and other information about the services deployed in an infrastructure. EMIR aims to facilitate reliable service discovery through provision of a tailored service designed specifically for this task. An initial version has been released in EMI 2 and is currently being evaluated for use in productions infrastructures.

4) Accounting: Historically, there were a number of different accounting solutions in the accounting landscape, which did not interoperate; APEL and DGAS[**?**] for gLite, JURA/SGAS for ARC, and the OGSA-RUS interface in UNICORE. As part of the harmonization activity within EMI, accounting records have been defined for computing (CAR) and storage (StAR), based on the OGF Usage Record [**?**]. The compute and storage services are being modified to be able to generate CAR and StAR records and furthermore, the transport system that moves records from the resource to the accounting server (APEL) is also being adopted to be used these records. These modified records are being used as important input to the next generation OGF Usage Record.

5) Messaging Services: Messaging services are increasingly being used in distributed systems and solutions based on such services have been chosen for Grid applications. For example, messaging technology has been adopted as an integration framework for service monitoring [**?**] and other Grid operational tools including accounting, ticketing and operational dashboards. It has been shown that the use of messaging technology can simplify such tools and improve their reliability through the adoption of either commercial or open source implementations. In this respect, the EMI Messaging Product Team has facilitated the adoption of messaging technologies within EMI by selecting, testing and documenting the available technologies as well as providing additional software, documentation and consultancy to the other Product Teams.

6) Service Monitoring: An early investigation looked into the possibility of use of messaging technology, with the aim of providing a common interface for service monitoring and management that could be adopted by all EMI middleware components and a survey of Grid sites was jointly conducted with EGI to investigate the requirements in this field. The feedback from this survey was that while there is general agreement that such a solution is on everyones wish list, this would only make sense within the wider context of standards in data centers, which is out-of-scope for EMI. In practice smaller more concrete objectives would have greater impact for service monitoring and management. The result of consultations with EGI has re-focused the plans with respect to Service Monitoring and Management[**?**]. In particular, for service monitoring, EGI requested that each service should provide a Nagios probe which can be used to measure the availability. 90% of the service Nagios probes were delivered and released in the EMI 2. For service management, as a number of different fabric management tools are used to provision services based on EMI components, it was requested that these components should conform to the operating system guidelines for the platforms which EMI supports.

C. Compute

Compute area services include middleware services involved in the processing and management requests concerning the execution of a computational task. They cover the interaction with Local Resource Management Services (as LSF, Torque/Maui, PBS, SGE), the provision of a common interface to the computational resources of a site (the so-called Computing Element), and the availability of high-level meta-scheduling, workflow execution and task tracking functionality. As number of services were already present and well established in the originary middleware stacks, the focus on this area has been given on making such solutions interoperable, through the support of common/standard tools, such as Argus for authorization purposes, or, when needed, the development of ad-hoc interfaces. The most notable results achieved in the first two project years in this area follows:

1) Full support for GLUE 2.0: The EMI Computing Elements (CE, ARC CE, gLite CREAM and UNICORE/X) fully support the publication of local-level resource information expressed according to the GLUE2 OGF proposed recommendation standard. The remaining activity for a complete GLUE2 support in the compute area is the development of GLUE2 support in the match-making modules and client tools. This concerns the implementation of a new module in the WMS, responsible for querying over LDAP a GLUE2 enabled BDII and for fetching information in the WMS internal cache, ready to be queried by the clients through the Job Definition Language (JDL).

2) Common job submission and management interface through the EMI Execution Service: The so called EMI Execution Service is a common job management interface to be implemented and adopted by all the middleware Computing Elements. This is one of the most distinguished developments of the project, and, fostered by the adoption of the common authentication library (CANL), will allow seamless execution of user jobs to the three EMI CEs (ARC CE[**?**], gLite CREAM[**?**] and UNICORE/X[**?**]). At the time of writing, the overall completion status of EMI-ES implementation for each middleware is estimated to be above 80%. Parallel to the ongoing implementation of the EMI-ES specification, interoperability tests have started already, involving all the three middleware services in all the possible combinations of clients and servers.

3) Definition and implementation of the EMI Compute Accounting Record(CAR): A compute accounting record (CAR) is typically defined in reflecting practical, financial and legal requirements of resource consumption, including CPU time, wall-clock time and memory usage. An agreement, in terms of XML schema definitions, over detailed and aggregated usage records was reached among all the EMI producers. 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 of the existing middleware layers and taking into consideration existing grid use cases.

4) Integrated solutions to interface with batch systems: An important compute area development task is to provide the ability for all the EMI computing elements to fully support a well advertised set of batch systems. This initial set included PBS/Torque family, Sun/Oracle/Univa Grid Engine and LSF. Then, SGE was added and now its fully supported by all the three CEs. Support for SLURM is planned to be supported by the end of EMI.

5) Common parallel execution framework: Another convergence task within the EMI compute area is the identification of a common parallel execution framework. In particular, rather than simply trying to identify a common back-end, a somewhat difficult and even not necessary step, it was decided to find an agreement on a definition of parallel jobs across the three 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, while the expected behavior is defined by the interface. The implementation of this solution required slight adaptations to the EMI-ES definitions, that were promptly done in order to accommodate for different kind of requests of parallel applications, coming from the long experience of MPI-Start.

D. Data

The EMI-Data portfolio provides components to store, manage, access and transfer data in the 100 Petabyte range, supporting highly distributed infrastructures. An high-level view of EMI-Data services shows systems both able to keep track on data locations as well as to manage and operate on the associated metadata. The major building blocks, as there are dCache, DPM[**?**], StoRM[**?**], the LFC[**?**], AMGA[**?**] and FTS[**?**], are already in production for several years and with that reached a high degree of stability. One of the main goal of EMI-Data is to allow costumers to combine those components according to their needs and to build a scalable and easy to maintain data infrastructure. Another, similarly important objective is to equip EMI-Data components with standard interfaces to allow them to be plugged into existing IT systems or to easily replace parts or complete industry software stacks by EMI open source software. Below, we briefly describe some most prominent activities.

1) POSIX File access: With the second major release of EMI, all provided storage elements support direct POSIX[**?**] access to storage, either through their storage backend as with StoRM or by implementing the new NFSv4.1/pNFS[**?**] industry file system standard. As a consequence, applications from all scientific communities can gain seamless access to EMI storage without being modified or without providing specific proprietary data access libraries.

2) From http(s) / WebDAV access to federated storage: Initially implementing a location catalogue for all EMI storage elements and local files, at the end of the second year of EMI, LFC was released with an http/WebDAV interface , while a WebDAV federation component was added with the beginning of the third year. This component is supposed to incorporate http/WebDV storage endpoints to a single storage space by overlaying their namespaces. This is either done by probing known endpoints for the requested data or by interrogating file location catalogues. A plugging framework allows to find the most appropriate storage endpoint for each individual client request by using proximity, storage element load or network topology information. Based on those pieces of information, WebDAV requests to the central service are redirected to the most appropriate endpoint.

3) Improved data transfer service, FTS: Based on the experience in transferring enormous amounts of data with the first generation of the WLCG File Transfer Service, FTS, the FTS team is building a second generation service, more flexible in handling shared networks and being able to incorporate load information from storage endpoints.

III. Quality assurance

The EMI software Quality Assurance (QA) process has been created for providing adequate confidence that the software products and processes in the project lifecycle conform to their specified requirements and adhere to their established plans. In order to provide a sound strategy for the whole EMI project, the specific context in this project has to be taken into account: the EMI middleware was developed by the major middleware providers in Europe, ARC, gLite, dCache and UNICORE. These middleware providers have been developing software in the grid domain for the past several years and they all had their own practices and tools therefore it has been very important to plan these practices into a single EMI QA activity (EMI Software QA Plan)[**?**].

The Quality Assurance activity had to define and foster common practise and tools and facilitate the transition towards an open source approach (standards, tools, repositories, etc), which is the part of the model chosen by EMI in order to ensure sustainability of the products after project completion. The main services and quality assurance tasks provided to the EMI projects are:

- Common policies, documents, templates (EMI QA Policies Documents)[?]
- Software metrics, with automated reports and dashboards
- Build infrastructure for all development teams (EMI QA Tools Documentation)[?]
- Testing infrastructure for all EMI software (EMI Certification and Integration Testbed)

A. Tools

ETICS[?] was the tool selected to integrate, build, test and package the different pieces of software that are part of EMI. After an initial survey circulated to collect the different requirements and tools used, ETICS was the tool better positioned. But some modifications were needed to fulfill all the requirement: standard tools, such as Mock and PBuilder were integrated to generate packages compliant with the Fedora and Debian packaging guidelines. It was needed to introduce support to new platforms, as Scientific Linux 5/6 and Debian 32/64 bits (this requiring support to APT repositories) These modifications include a new virtualization system and a change of paradigm about how the worker nodes are started. For generating the different reports, a complete new report generator framework was developed to analyze the different values collected by the plugins during the builds and generate the different charts.

B. Results

The compliance level of QA policies and procedures have been measured by the QC team which was mainly focused on:

- Verifying that software products, being included in major releases, were compliant with the Production Release Criteria as defined within the Software QA Plan
- Performing the security assessment of a sub-set of released products to ensure they were not containing any vulnerabilities or security holes
- Verifying that released products were compliant with EPEL packaging criteria
- Preparing reports where results of periodic control activities are collected and any existing non-conformity or deviation are pointed out.

Collected results and measurements, which are reported in the following sections with more details, present a progressive improvement on quality performance. The good improvement made on regression testing and EPEL/Debian compliance metrics has given a real impact on the quality of the product and in its future sustainability after the end of the project.

Metrics reports have helped to detect deviations from the EMI QA policies. A process to follow up these deviations is also defined in the EMI Quality Model and the Quality Control (QC) has been responsible for tracking the situation and proposing the necessary corrective actions as described in the rest of the document.

A general major improvement on all metrics occurred moving from EMI 1 to EMI 2. Large part of the updated products is accompanied with unit and regression tests and the coverage of RfCs is clearly improving. Best results were achieved for EMI 1 Updates as probably developers had more time to fulfill agreed policies:

- 97% of the EMI 2 code passed all the mandatory checks
- 100% of High/Emergency RfCs are covered by associated tests in all EMI 1 updates
- 90% of the products are basically EPEL compliant and have less than 15 errors
- 90% have a trend improving their EPEL compliance

The measurements are collected either using the automatic procedures provided by the other tasks of the QA activity such as the above mentioned, RfC and Verification dashboards and the statics metrics extracted by the ETICS tool. During this second year the work started in the first year has been completed and the QC task has verified all the EMI 1 Kebnekaise updates and the full EMI 2 release. Comprehensive feedback has been provided to the Product Teams and EMI products have always successfully passed the EGI acceptance criteria[?].

Besides the performing of quality checks on the software, the QC collected also measurements to evaluate the work package performance on the basis of quality indicators like the respect of the published deadlines stated in the release process.

IV. Maintenance, Support and Release

The maintenance and evolution of the EMI software is implemented inside Product Teams (PTs). PTs are small teams of software developers, fully responsible for the successful release of a particular software product, or a group of tightly related products, compliant with agreed sets of technical specification and acceptance criteria. Individual PTs are requested to implement the necessary processes and procedures and respect established requirements and priorities as globally defined and endorsed by the EMI project executive and technical leadership.

The software development process is driven by two concurring demands:

- Software evolution, harmonization and consolidation, as defined in EMI Technical Development Plans, to address user requirements for new functionality and rationalize the EMI software stack.
- Software adaptive and corrective maintenance : to address problems reported by the middleware users or changes in the software operating environment.

A. Support and maintenance policies

The natural outcome of the maintenance activity is the release of the updated components in production, following the EMI updates release process, while the main new developments are released in periodic major releases, delivered once a year. In particular, backward-incompatible changes to the interface or to the behavior of a component that is part of the EMI distribution can be introduced only in a new EMI major release. Changes to interfaces that are visible outside the node where the component runs need to be preserved even across

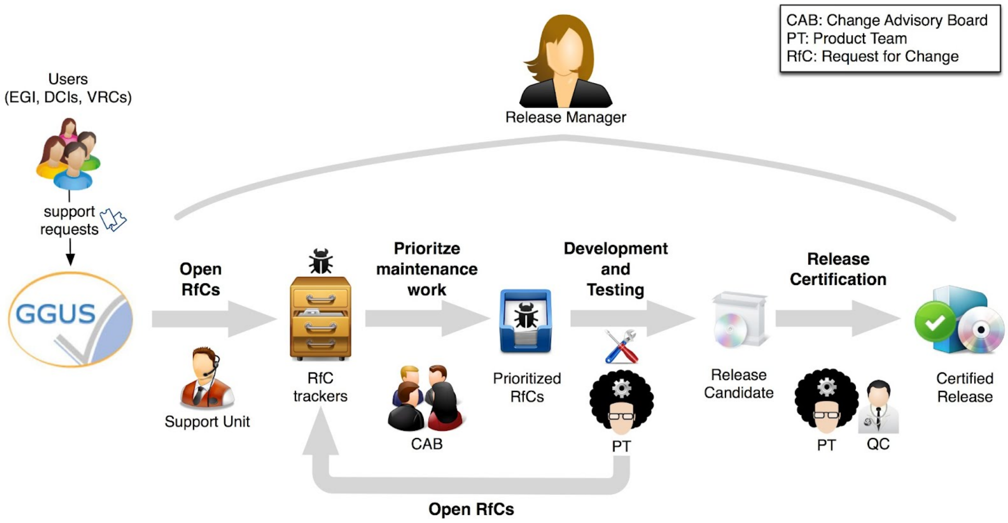


Fig. 1. The EMI release process

major releases, according to end-of-life policies to be defined on a case-by-case basis.

EMI distributes its software through the EMI repository based on the AFS and HTTPD services provided by CERN. The repository hosts all the software components developed during the lifetime of the project, while EMI releases are announced through various channels, e.g., the EMI website, mailing lists and dedicated RSS feeds.

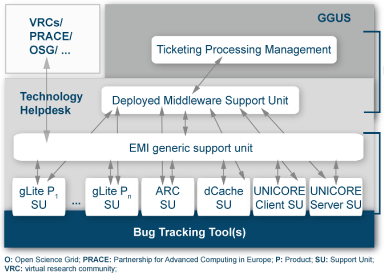
EMI major releases are supported and maintained for approximately two years after the release date. The availability of a new major release of EMI does not automatically obsolete the previous ones and multiple major releases may be supported at the same time according to their negotiated end-of-life policies. It is foreseen, however, that only the latest two EMI major releases will be supported at any time. For each supported EMI major release, the maintenance schedule is organized in the following periods:

1) Full maintenance, 12 months: in this period, updates address issues in the code and provide new features.
2) Standard maintenance, 6 months: this period follows full maintence. Updates address issues in the code, but no new features are introduced.
3) Security updates, 6 months: after the standard maintenance period, only updates targeting security vulnerabilities are provided.
4) End-of-life: this 'period' starts after the end of the security updates period. Provision of updates and support cease.

Specific exceptions to the above maintenance periods can be negotiated between the users and the PTs, depending on various criteria that include critical run-time of experiments or projects that do not allow upgrades to new versions, particularly complex deployment and migration conditions, etc.

B. The maintenance and release processes

The EMI software maintenance activity (see Figure **??**) is mainly driven by Request for Changes (RfCs) targeting EMI components. RfCs are typically created as consequence of incidents reported by users through the support channels when the cause of the incidents is traced to an actual problem in the code by EMI Support Units. RfCs also originate from PTs, when problems are found in the code during development or when introducing minor unplanned improvements. Although RfCs are tracked in several independent PT bug trackers, the RfC reporting and aggregation tools implemented by the EMI QA team provide a common view on submitted RfCs. In particular, a weekly report summarizes the status of Immediate and High priority RfCs affecting released products. The Change Advisory Board analyzes the content of the weekly report to assess and validate the priority assigned to each RfCs and select the items that need to be fixed in the next maintenance release for the affected components. A task is created in the EMI Release Tracker for each component which links to the RfCs that will be addressed At this stage, PTs start the development and testing work required to address the problems reported in the task. Once PTs are reasonably confident that all problems have been fixed and that changes in the code do not break the ETICS continuous integration build, the code is tagged in the VCS and the build configuration is attached to the Release Candidate (RC) build for the targeted EMI major release. At this stage, PTs complete the product internal certification following the Testing and Certification policies defined by the project. If required, the documentation is updated and release notes for the new release are produced. The task in the release tracker is moved to the *Certified* status, and the release candidate is handed over to the QC team and Release Manager for the final verification phase, which includes deployment on the EMI testbed and additional validation. If the final validation step yields positive results, the EMI repository is updated with the new components and



O: Open Science Grid; **PRACE:** Partnership for Advanced Computing in Europe; **P:** Product; **SU:** Support Unit; **VRC:** virtual research community;

Fig. 2. The EMI support model

the release is announced.

C. The EMI support structure

The EMI support model (Figure **??**) integrates in the overall support structure adopted in EGI, which foresees an organization in three levels. The EGI Helpdesk represents the main contact point for a user where to get support. Within the Helpdesk the Ticket Processing Management (TPM) is responsible for the monitoring and routing of all active tickets to the appropriate support units (SUs). In EGI the Helpdesk is a distributed infrastructure consisting of a central Helpdesk interconnected with a collection of local NGI Helpdesks. If the Helpdesk is unable to resolve the incident, this is escalated for further investigation to a 2nd- level support unit.

The Deployed Middleware Support Unit (DMSU) ensures the availability of more specialized skills than those offered by the Helpdesk in the investigation and resolution of incidents. The DMSU includes people that together can cover all middleware areas: job and compute management, data management, security, information systems, accounting, etc. The DMSU is an integral part of EGI. If the DMSU is unable to resolve the incident, this is escalated for further investigation to a 3rd- level SU. 3rd-level SUs offer the most specialized skills needed for the investigation and resolution of an incident and are typically represented by the developers of the affected software component. 3rd-level SUs are not normally part of EGI but are integrated in the organization of the software providers, such as EMI.

This industry-standard model provides the most effective use of resources, for it involves the ultimate technical experts only when their detailed knowledge is indispensable for the investigation of an incident. Support tickets should not normally flow from the Helpdesk directly to the EMI SUs, unless it is evident that the incident is caused by a software problem. Incidents occurring to users on the production infrastructure, even if initially reported through other means (typically mailing lists) should always be reported through GGUS and their processing tracked through GGUS tickets. This allows EMI to compute user-oriented metrics completely from GGUS data.

V. Outlook to the Future

The aforementioned fruitful results of collaboration among ARC, dCache, gLite, and UNICORE raise the question of how the significant parts of these harmonization activities can be sustained in the future. Although not every aspect is clear, there are some activities that give a glimpse into the future of the Grid middleware solutions in Europe.

The EMI partners have committed themselves to take over maintenance to the most possible degree supporting their strongly-related scientific communities. At the time of writing, we foresee that at least all the major scientific communities taking advantage of EGI and PRACE can expect that EMI products will exist after the project end. At the same time, EMI activities will expand to the market of distributed services. The simplification and standardization work done will permit an alternative approach, allowing exploitation of single products instead of the traditional full-featured middleware solutions.

These activities will be conveyed into the *ScienceSoft Open Software for Open Science* initiative [**?**], which emerged from the EMI collaboration in order to expand the activities into the broader open science market, promoting a community driven approach. ScienceSoft aims to explore the feasibility of creating an open source community for software specific to scientific communities. This inherently includes the sustainability of the EMI collaboration on a more loosely coupled basis in order to let new partners to join collaborations beyond the currently middleware focused work. The fundamental goal of ScienceSoft would be thus to use open source software for science in a more transparent manner making it truly collaborative across communities and projects. Future activities in this market are expected to also have side effects in implementing a potentially mixed sustainable business model over time based on existing examples such as Apache [**?**] or Eclipse [**?**].

VI. Conclusions

This paper has been written right after the EMI 2nd major release, EMI 2 Matterhorn. After an intense 24 months of planning, development, integration and harmonization, the project has delivered a consolidated software stack for deployment in distributed computing infrastructures. A streamlined set of compute, data, infrastructure and security services have been made available to resource providers addressing requirements from its communities, some of them heavy users from the fields of high energy physics, life sciences and biology.

EMI takes pride in being part of the discovery of a new particle consistent with the Higgs boson based on data collected and processed in 2011 and 2012. This exceptional discovery has been attributed to the global effort of the experiments, the infrastructure and the grid computing[**?**]. A significant number of EMI products have been released by EGI, the largest research infrastructure in Europe with an estimated 20,000 users to date. This is a major milestone in providing a consistent platform, software and infrastructure, for all users in the European Research Area to gain access to suitable and integrated computing resources. EMI's last year will be no less demanding, with the release of its final distribution EMI 3

Monte Bianco. An even wider adoption of core EMI services is the target for the last year, with many service owners pledging to continue supporting and maintaining its services after the end of the project.

The EMI project vision is part of a more general context of European and international collaborations, where relevant stakeholders actively take part in establishing a functional distributed computing infrastructure ecosystem. Interactions, collaborations and cross-fertilization of expertise, ideas and results are paramount to achieving this objective. The EMI collaboration represents engagement and commitment of a significant number of, on one side, middleware developers and resource providers, and on the other side, users of distributed computing resources, an important step in shaping the distributed computing ecosystem.

Complementing this commitment is the implementation of ScienceSoft, an EMI initiative to build a network of developers and users. The target communities are those leveraging open source solutions - developers, service providers, researchers, platform integrators, projects, companies and funding bodies. These communities will have access to an information hub of software catalogues, service catalogues, people directory and profiles. ScienceSoft will provide a marketplace for scientific communities to find open source software and services they need and the people who can provide them. The plan is to make available in ScienceSoft information about EMI-developed software for everyone's use, even after the end of the project.

Acknowledgment

This work has been partially funded by the European Commission as part of the EMI (Grant Agreement INFSO-RI-261611) project.

References

[1] EGEE (Enabling Grids for E-sciencE) web site, retrieved July 2012, http://www.eu-egee.org/

[2] DEISA (Distributed European Infrastructure for Supercomputing Applications) web site, retrieved July 2012, http://www.deisa.eu/

[3] WLCG (Worldwide LHC Computing Grid) web site, retrieved July 2012, http://wlcg.web.cern.ch/

[4] NDGF (Nordic DataGrid Facility) web site, retrieved July 2012, http://www.ndgf.org

[5] OSG (Open Science Grid) web site, retrieved July 2012, http://www.opensciencegrid.org/

[6] M. Litzkow, M. Livny and M. Mutka *Condor - A Hunter of Idle Workstations*, Proceedings of the 8th International Conference of Distributed Computing Systems, pag. 104-111, June 1988

[7] A. J. Ferrari, A. S. Grimshaw et al. *From Legion to Avaki: The Persistence of Vision*, chapter 10 in *Grid Computing: Making the Global Infrastructure a Reality*, published by John Wiley & Sons in March 2003, pages 265–298, ISBN 0-470-85319-0

[8] SETI@Home web site, retrieved September 2012, http://setiathome.berkeley.edu/sah_about.php

[9] Anderson, D.P. *BOINC: a system for public-resource computing and storage* Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on Digital Object Identifier: 10.1109/GRID.2004.14 Publication Year: 2004 , Page(s): 4 - 10

[10] I. Foster, C. Kesselman *Globus: A Metacomputing Infrastructure Toolkit*, Intl J. Supercomputer Applications, 11(2): pag. 115-128, 1997.

[11] M. Romberg *The UNICORE Architecture: Seamless Access to Distributed Resources*,Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing (HPDC-1999), Redondo Beach, USA, IEEE Computer Society Press, 1999, pages 287-293

[12] E. Laure, F. Hemmer, A. Di Meglio et al. *Middleware for the Next Generation Grid Infrastructure*. Proceedings of Computing in High Energy and Nuclear Physics (CHEP) 2004, September 2004

[13] M. Ellert et al. *Advanced resource connector middleware for lightweight computational grids*, Future Generation Computer Systems, Vol. 23, Issue 2, February 2007, Pag. 219-240

[14] P. Fuhrmann et al. *dCache, a distributed storage data caching system*, Proceedings of Computing in High Energy and Nuclear Physics (CHEP) 2001, September 2001

[15] EGI official web site, retrieved July 2012 : http://www.egi.eu

[16] EMI 1 Kebnekaise page, retrieved July 2012 : http://www.eu-emi.eu/emi-1-kebnekaise/

[17] Filesystem Hierarchy Standard Group web page, retrieved September 2012 : http://www.pathname.com/fhs/pub/fhs-2.3.html

[18] Fedora Packaging Guidelines, retrieved September 2012 : http://fedoraproject.org/wiki/Packaging:Guidelines

[19] EPEL Packaging Guidelines, retrieved September 2012 : http://fedoraproject.org/wiki/EPEL/GuidelinesAndPolicies

[20] EMI 2 Matterhorn page, retrieved July 2012 : http://www.eu-emi.eu/emi-2-matterhorn/

[21] Neuman, B.C., *Kerberos: an authentication service for computer networks*, Communications Magazine, IEEE, vol. 32, issue 9, pag. 33-38

[22] Shushan Zhao, A.Aggarwal, R.D. Kent *PKI-Based Authentication Mechanisms in Grid Systems*. Proceedings of International Conference on Networking, Architecture, and Storage, 2007. NAS 2007. Page(s): 83 - 90

[23] Bruce Beckles, Von Welchb, Jim Basney *Mechanisms for increasing the usability of grid security* International Journal of Human-Computer Studies, Volume 63, Issues 12, July 2005, Pages 74101

[24] R. Sinnott, J. Jiang, J. Watt, O. Ajayi *Shibboleth-based Access to and Usage of Grid Resources* Proceedings of the 7th IEEE/ACM International Conference on Grid Computing, Pag. 136-143

[25] First set of common SAML authorization attributes, retrieved July 2012, https://twiki.cern.ch/twiki/bin/view/EMI/CommonSAMLProfileV102

[26] The XACML profile commonly agreed for the EMI components, retrieved July 2012, https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T4XACML

[27] L. Field, S. Andreozzi, B. Konya, *Grid Information System Interoperability: The Need For A Common Information Model*, eScience '08. IEEE Fourth International Conference on, pag. 501-507

[28] A. Guarise, R.M. Piro and A. Werbrouck *An Economy-based Accounting Infrastructure for the DataGrid*, 4th Int. Workshop on Grid Computing (GRID2003), November 17, 2003, Phoenix, AZ

[29] R. Mach, R. Lepro-Metz and S. Jackson *Usage Record - Format Recommendation*, OGF recommendation (GFD-R-P.098) obtained through the Internet, July 2012 : http://www.ogf.org/documents/GFD.98.pdf

[30] P. Nyczyk, Z. Sekera et al. *Monitoring the Availability of Grid Services Using SAM and Gridview*,Grid Computing - proceedings of International Symposium on Grid Computing (ISGC2007),Springer US 2009, isbn 978-0-387-78417-5, pag. 163-168

[31] *Towards an Integrated Information System* workshop minutes, https://indico.egi.eu/indico/conferenceDisplay.py?confId=654

[32] M. Sgaravatto et al. *Design and implementation of the gLite CREAM job management service*, Future Generation Computer Systems Vol. 26, Issue 4, April 2010, Pag. 654667

[33] A. Streit, P. Bala et al. *UNICORE 6 Recent and Future Advancements* Annals of Telecommunications, Vol. 65, Numbers 11-12 (2010), 757-76

[34] R. Zappi et al. *StoRM, an SRM Implementation for LHC Analysis Farms* Computing in High Energy Physics (CHEP 2006), India, Feb. 13-17, 2006.

[35] A. Frohner et al. *Data management in EGEE*, Journal of Physics: Conference Series, 2010, Vol. 219 par. 6

[36] POSIX standard specifications (IEEE Std 1003.1,2004 Edition) http://www.unix.org/version3/ieee_std.html, retrieved July 2012

[37] pNFS web site, retrieved July 2012 : http://www.pnfs.com/

[38] M. Alandes et al. *Why are common quality and development policies needed ?* http://cdsweb.cern.ch/record/1457987/files/CHEP-poster-paper-policies-reviewed.pdf

[39] *EMI Quality assurance tools documentation*, EU deliverable, http://cdsweb.cern.ch/record/1277591/files/EMI-DSA2.2.3-1277591-QA_Tools_Documentation-v1.0.pdf

[40] A. Di Meglio et al. *ETICS: the International Software Engineering Service for the Grid* Journal of Physics Conferences Series, 2008: Conf. Ser. 119 042010 (11pp)

[41] *Software Provisioning Process*, EU deliverable 508, retrieved in August 2012, https://documents.egi.eu/document/505

[42] Science Soft web site, July 2012 : http://www.sciencesoft.org

[43] Apache foundation web site, July 2012 : http://www.apache.org/

[44] Eclipse Foundation web site, July 2012 : http://www.eclipse.org/

[45] Press release: CERN experiments observe particle consistent with the Higgs boson, July 2012 : http://press.web.cern.ch/press/PressReleases/Releases2012/PR17.12E.html