

SOFTWARE PROVISION PROCESS FOR EGI

Mário DAVID, Gonçalo BORGES, Jorge GOMES, João PINA

*Laboratório de Instrumentação e Física Experimental de Partículas
Lisboa, Portugal
e-mail: david@lip.pt*

Isabel CAMPOS, Enol FERNANDEZ, Alvaro LOPEZ, Pablo ORVIZ

*Instituto de Física de Cantabria, CSIC
Santander, Spain*

Javier López CACHEIRO, Carlos FERNANDEZ, Alvaro SIMON

*Fundacion Centro de Supercomputacion de Galicia
Santiago de Compostela, Spain*

Alvaro FERNANDEZ

*Instituto de Física Corpuscular CSIC/University Valencia
Valencia, Spain*

Abstract. The *European Grid Initiative* (EGI) provides a sustainable pan-European Grid computing infrastructure for e-Science based on a network of regional and national Grids. The middleware driving this production infrastructure is constantly adapted to the changing needs of the EGI Community by deploying new features and phasing out other features and components that are no longer needed. Unlike previous e-Infrastructure projects, EGI does not develop its own middleware solution, but instead sources the required components from Technology Providers and integrates them in the *Unified Middleware Distribution* (UMD). In order to guarantee a high quality and reliable operation of the infrastructure, all UMD soft-

ware must undergo a release process that covers the definition of the functional, performance and quality requirements, the verification of those requirements and testing in production environments.

Keywords: Grid infrastructures, Grid middleware, European Grid Infrastructure, European Middleware Initiative

1 INTRODUCTION

The *European Grid Initiative* (EGI) [1, 2] provides a computing infrastructure formed by a federation of cooperating resource providers integrated into Regional and National Grid Initiatives (NGIs) across Europe. EGI does not develop the software deployed in the grid infrastructure – all components are provisioned in partnership with Technology Providers. Technology Providers are organizations or projects collaborating with EGI that develop or deliver software for use within the production infrastructure.

Two types of software are deployed in the EGI Infrastructure: The middleware delivered by external Technology Providers such as the European Middleware Initiative (EMI) [4] or the Initiative for Globus in Europe (IGE) [5], is integrated by EGI in the *Unified Middleware Distribution* (UMD), and a set of Operational Tools – providing ticketing systems for user and operational support, accounting and monitoring – that support the operation and management of the production infrastructure.

EMI is the technology provider that will distribute the ARC, gLite, UNICORE and dCache middleware components in an integrated release, while IGE is the technology provider that will distribute and support the Globus middleware stack.

Most Operational Tools are developed within the framework of the EGI-InSPIRE [3] project, acting as an internal Technology Provider.

Each software component submitted by the Technology Providers must undergo the software provision process before its General Availability for deployment in the production infrastructure. This well-defined workflow assures the quality and reliability of the software by assessing it against a set of defined criteria. The workflow consists of two main phases: *Software Verification* and *Staged Rollout*.

This paper is organized as follows. Section 2 gives an overall description of the full Software Release Process. Section 3 describes the Software Verification phase including the Quality Criteria. The Staged Rollout workflow, the past year experience and Early Adopters are described in Section 4 and Section 5 gives a brief overview of the upcoming EMI1.0 release due at the end of April 2011. Conclusion and future prospects are drawn in Section 6.

2 SOFTWARE PROVISIONING WORKFLOW

The Software Provisioning activity in EGI [6] governs two processes to ensure software quality and its correct integration into EGI UMD repository. These activities include a continuous revision of the process to adapt any change required by the EGI community (including new features and phasing out features and components that are no longer needed), and the maintenance and improvement of the process quality as defined by EGI. The Software Provisioning activity acts as a filter to accept or to reject new middleware products or components into EGI's UMD repository.

The Software Provisioning workflow is depicted in Figure 1. It starts when a new software component release is made available by the Technology Provider.

The Technology Provider creates a ticket in the *Global Grid User Support* system (GGUS) [7]. The GGUS ticket is filled in with complete information about the new software release such as: release notes, documentation, installation and configuration notes and the list of packages contained in the release.

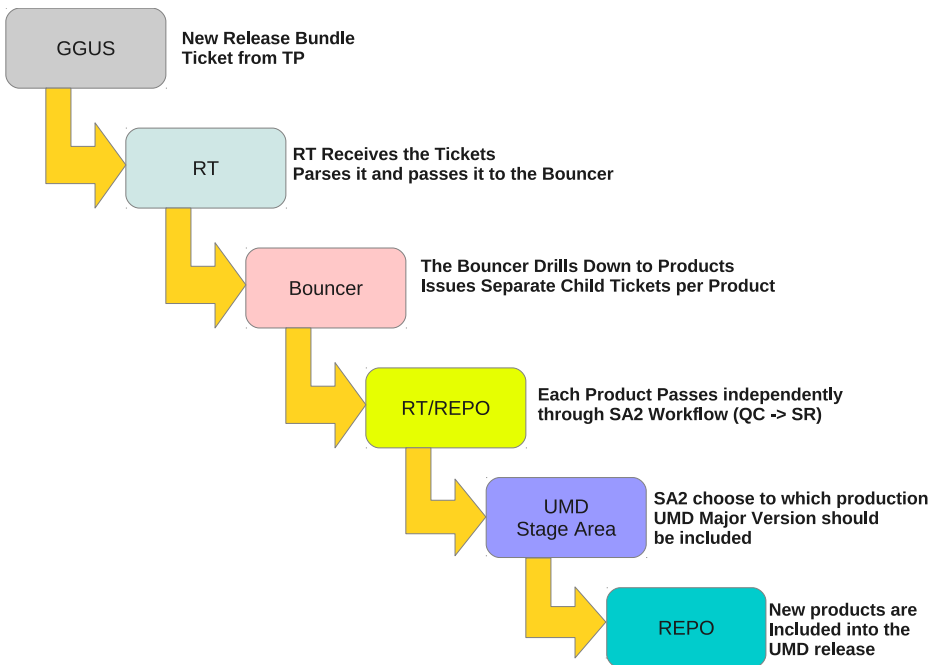


Fig. 1. Software provisioning workflow also showing the tools used in the whole process

The ticket is assigned to the EGI Technology Unit (SA2), triggering the creation of a ticket in the EGI RT system [8], in a dedicated queue named “sw-rel”. The state diagram of this queue is shown in Figure 2, and is the main tool to handle the

Software provisioning workflow. Upon the RT ticket creation, the Software Delivery phase (implemented by an external RT module called “Bouncer”) takes place and the new Technology Provider release is processed automatically.

The end result of the “Bouncer” is a list of product release descriptions specific for each platform and architecture (*Product-Platform-Architecture* **PPA**), creating one child ticket (in the RT queue “sw-rel”) per combination of Product-Platform-Architecture. From EGI point of view a *Product* is a solution delivered by Technology Providers which offer the functionality for one or more capabilities as one single and indivisible unit.

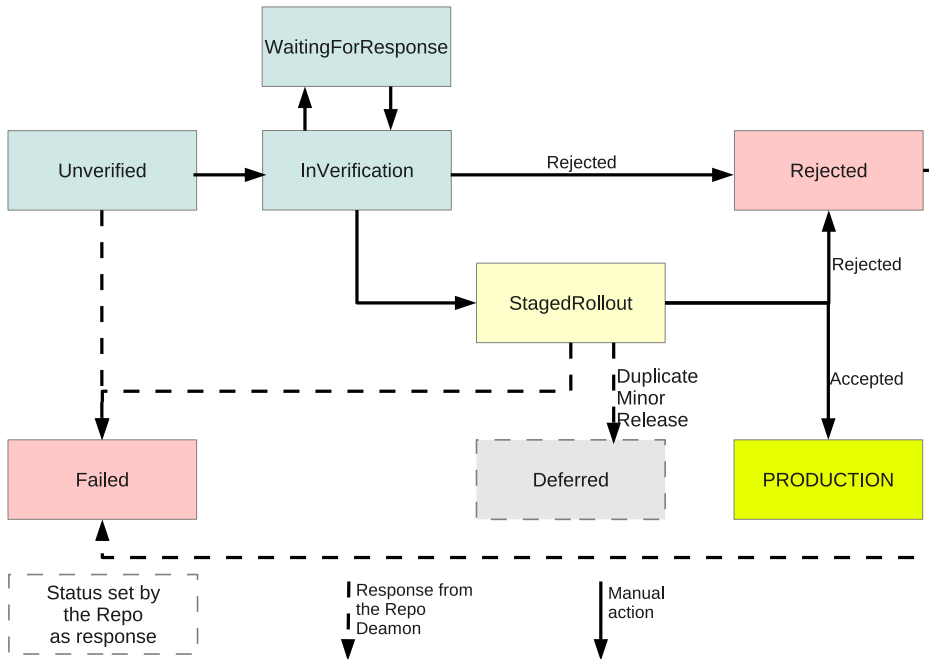


Fig. 2. Software release workflow implemented in the EGI RT and repository portal [9]

This concept is important due to the intrinsic architecture of the infrastructure, in particular to the resource infrastructure administrators and EGI Operations Unit (SA1). The previously mentioned parties have driven the requirements for the EGI repository structure [10] to be PPA oriented, e.g. there is one repository per PPA combination.

Each “*Product-Platform-Architecture*” combination is tracked in its own child ticket during the Verification and Stage Rollout phases. As such, several PPA combinations may be processed in parallel and independently from each other. The RT ticket state changes from “Unverified” to “InVerification” when the Verification process starts (see Figure 2).

If the outcome of the Verification phase is “**accepted**”, the RT ticket state changes from “**InVerification**” to “**StagedRollOut**” (Figure 2). This state transition triggers the creation of a child ticket in another specialized RT queue named “**staged-rollout**”. The Staged Rollout phase is fully tracked through tickets in this queue. This phase is complete when the ticket is set to “**resolved**” status with a given outcome. A more detailed description will be given below.

If the outcome of the Staged Rollout is “**accepted**”, a new release is prepared for General Availability to the EGI infrastructure. Several actions are performed: preparation of a new UMD release, transfer of software packages to the production repositories, broadcast of the new release to all resource infrastructure administrators and other involved parties. The PPA “**sw-rel**” child ticket is set to “**resolved**” with an outcome of “**accepted**”.

An outcome of “**rejected**” is set if problems or bugs are discovered in any of the previously described phases. In this case, the release is rejected and the “**sw-rel**” child ticket is closed.

Whatever the outcome, setting the RT ticket to “**resolved**” triggers the updating of the GGUS ticket providing a report to the Technology Providers and closing it. This is the final state of the workflow.

At the time of writing, the full Software Provision workflow is matured and largely implemented, both in terms of the process itself as well as of the supporting tools.

3 SOFTWARE VERIFICATION PROCESS

The EGI TSA2.3 task handles the Software Verification process. The objective of this task is to verify the software quality before the Staged Rollout phase. Software components should comply with the quality criteria in order to be deployed in the production infrastructure.

During the first project year, the verification process was designed and implemented based on Unified Middleware Distribution Quality Criteria capabilities.

The release of the first version of the Quality Criteria has allowed the creation of the verification templates that are publicly available in the EGI Document-DataBase [11]. The Verification phase of the workflow is described next.

When the “**sw-rel**” RT ticket changes into the state “**InVerification**” the Verification phase starts (Figure 2). The actions that a verifier has to perform are:

1. Check several documents such as: release notes, “*changelogs*” and which bugs or issues are solved, existence of updated documentation and the certification reports provided by the Technology Providers.
2. Check if there are security vulnerabilities that need more detailed treatment and testing.
3. Determine if all required capabilities are present.
4. In specific cases, do a deployment and configuration test of the component.

5. Produce a verification report (using the report templates mentioned above), and set an outcome for the release.

Depending on the outcome of the Verification phase, if the release is rejected, this is automatically communicated back to the original GGUS ticket together with the reasons of the rejection. If the outcome is “**accepted**”, the workflows proceeds to the Staged Rollout phase.

The Verification task is handled by Ibergrid EGI partners; it is foreseen to involve other EGI partners in the case of Products where other skills and experience is needed. This is the case for ARC and UNICORE components.

3.1 Quality Criteria

The EGI task TSA2.2 handles the Software Quality Criteria specification. This is a continuous process that is formally reviewed every 6 months producing new/updated versions of the Quality Criteria documents, with input from all involved parties: user community, operations, Technology Providers and the EGI Technology Unit (SA2).

There are two major types of Unified Middleware Distribution Quality Criteria (QC) documents: “*Generic*” and “*Specific*” QC.

The “*Generic Quality Criteria*” [12] is applicable to all Unified Middleware Distribution components and Operational Tools. The main sections are:

- documentation: Release notes, user guides and other software documentation.
- software Release: Component license, source code availability, etc.
- service Criteria: Service logs, management and monitoring.
- security: File access policy.

As such, the new software must be well documented, it should include a license that permits the deployment of that software in the EGI infrastructure, publicly available open source code, etc..

On the other hand, software capabilities depend on the product used, as such, the “*Specific Quality Criteria*” targets the specificity of each component. Due to the high heterogeneity of Unified Middleware Distribution software capabilities and sources, all the “*Specific Quality Criteria*” are grouped in six major groups:

1. Compute Capabilities: Products with Compute Capabilities QC are aimed at job execution (parallel and sequential) and jobs scheduling.
2. Data Capabilities: Product capabilities aimed at data management and data or metadata catalogs.
3. Information Capabilities: Grid information model schema and service discovery capabilities.
4. Operations Capabilities: monitoring and accounting capabilities.

5. Security Capabilities: Grid authentication, authorization, access and Virtual Organizations management capabilities.
6. Storage Capabilities: Products that target data transfer and storage capabilities.

The activity in the first EGI project year has been in the definition of all Quality Criteria documents where input and close collaboration from the Technology Providers was essential, aiming at the first version of those documents finalized for the first EMI release.

3.2 Verification of the Quality Criteria: Results

Some software components have been put through the workflow. These first uses of the workflow process were aimed at testing the system and the tools. They were also used to fine tune the tools and check for additional customization that needed to be implemented. As such, the following products were used for the tests:

- Trust Anchors updates: this component is internal to EGI and contains the root certificates of all Certification Authorities. Version 1.37-1 was the first component undergoing through the workflow. Due to the importance of the Certificate Authority updates, specific Quality Criteria were defined and made available in the egi.eu wiki [13] in order to ensure the correctness and validity of the certificates included in the package. This CA version was released to production without problems. After the CA update verification a new version of Trust Anchors (v1.38-1) was also verified following the same procedure. In this case this release was performed after the final release of the Quality Criteria documents.
- SAM/Nagios monitoring updates: This component is part of the Operational Tools and the Technology Providers are internal to the EGI project. The Update 6 of SAM monitoring probe was the first Operational Tools component that went through the verification process. Verification started with the development and definition of Quality Criteria prior to the actual release with the collaboration of monitoring experts from the Operations Community. The criteria, as with the Trust Anchors, were initially made available in the egi.eu wiki [13] and later as part of the first Quality Criteria release. The Verification team has verified, 4 updates so far.

At the time of writing, it is considered that all tools used to support the Software Provisioning workflow, as well as the procedures, are ready to process all other software components.

4 STAGED ROLLOUT

The EGI Staged Rollout is the procedure through which newly verified software releases are first deployed and tested by Early Adopter sites before General Availability to all sites in the production infrastructure.

In the general case, the Staged Rollout test is performed in a production service. As such, the new version of the software is exposed to a more heterogeneous/chaotic environment with different users and applications, compared with the certification and verification phases. During this process, if issues or problems arise, then workarounds must be added to the release notes or, in more extreme cases, the update may be rejected.

An initial description of the process can be found in [14, 15], thus in this paper we will concentrate on actions taken to implement the process during the first EGI project year.

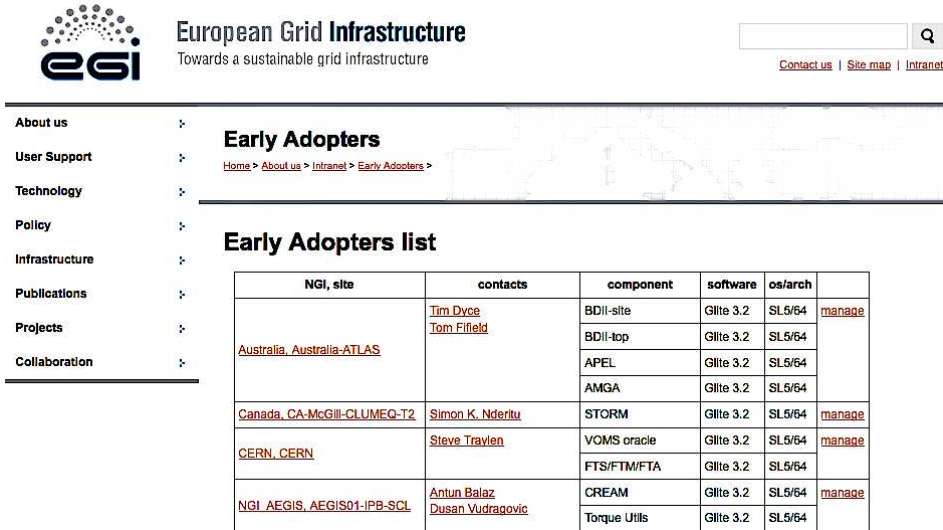
As previously described, the Staged Rollout phase is triggered when the state of the RT ticket in the queue “sw-rel” changes from “InVerification” to “StageRollOut” triggering the movement of the packages into the respective repository [10]. This means that the software component was accepted in the verification phase and a child ticket is created in the “staged-rollout” queue. At this point, the following actions are performed:

1. The newly created child ticket imports the relevant information from its parent ticket: Release notes, bugs or issues fixed, other software documentation.
2. A staged rollout manager assigns the ticket to all Early Adopter teams responsible for the test.
3. Each Early Adopter team deploys the new software in their corresponding production service.
4. Any problem found is either reported in the ticket, or if it is a serious bug through the GGUS system.
5. The new version of the software component is exposed to production load (environment) and users. This period may last between 5 to 7 days, but may be extended depending on the component under test.
6. Each Early Adopter team must fill a report of the test.
7. The staged rollout manager collects all reports, produces a summary with an outcome and make it publicly available through the EGI Document DB. The identifier of the reports is inserted into the ticket and ticket is closed with an outcome of “accept” or “reject”.

When the component passes the staged rollout phase, the child ticket is set to “resolved”, the outcome and relevant information about the Staged Rollout reports are communicated back to the parent ticket ending this part of the workflow.

4.1 EGI Project Year 1 Results

The Staged Rollout procedure is being coordinated within EGI through the TSA1.3 task. The transition process has had a smooth evolution from the one implemented during the *Enabling Grids for E-science* project (EGEE) [14]. It involved the



European Grid Infrastructure
Towards a sustainable grid infrastructure

Contact us | Site map | Intranet

Early Adopters
Home > About us > Intranet > Early Adopters >

Early Adopters list

NGI, site	contacts	component	software	os/arch	
Australia, Australia-ATLAS	Tim Dyce	BDII-site	Glite 3.2	SL5/64	manage
	Tom Fifield	BDII-top	Glite 3.2	SL5/64	
		APEL	Glite 3.2	SL5/64	
		AMGA	Glite 3.2	SL5/64	
Canada, CA-McGill-CLUMEQ-T2	Simon K. Nderitu	STORM	Glite 3.2	SL5/64	manage
CERN, CERN	Steve Travien	VOMS oracle	Glite 3.2	SL5/64	manage
		FTS/FTM/FTA	Glite 3.2	SL5/64	
NGI, AEGIS, AEGIS01-IPB-SCL	Antun Balaz	CREAM	Glite 3.2	SL5/64	manage
	Dusan Vudragovic	Torque Utils	Glite 3.2	SL5/64	

Fig. 3. EGI portal for Early Adopters management (<https://www.egi.eu/earlyAdopters/table>)

change of EGEE procedures into the new software release workflow, using the tools provided within EGI and deprecating the ones previously used.

One of the main points during this transition was to have a gradual and smooth transition in the procedures that the Early Adopter sites have to follow. On the other hand, the transition has had a larger impact on the coordination and management of the whole process, e.g. using partially the tools and processes inherited from EGEE together with tools and processes/workflow devised in EGI.

In the first iterations all involved parties (EGI Operation Unit and Technology Providers), it was agreed that Technology Providers should not interact with the EGI RT system. As such, the initial workflow was re-designed so that the process starts with a GGUS ticket that is opened by the external Technology Provider, followed by all steps previously described.

Therefore all communications between EGI and Technology Providers are carried through GGUS ticket(s), while allowing a single well determined point of exchange of information, public availability, traceability and easier extraction of metrics. This well determined communication channel does not exclude other means of more informal communication, such as e-mail or private communications.

Although some software components already went through the EGI Software Provision workflow, the gLite middleware components still go through the old process, i.e. using the CERN Savannah patches¹ to track the verification and staged rollout phases. Regarding gLite 3.1 and 3.2 components, it is not foreseen to adopt

¹ <https://savannah.cern.ch/patch/?group=jra1mdw>

the new procedure. Only the upcoming EMI releases (as well as IGE releases) are foreseen to undergo the new workflow.

Figure 4 shows the number of staged rollout tests undertaken per Regional or National Grid Initiative (NGI) during the period of first 10 months of the EGI project. It can be seen that a few NGI's are performing a large fraction of this task, and it is perceived that the workload has to be spread among the several NGI's.

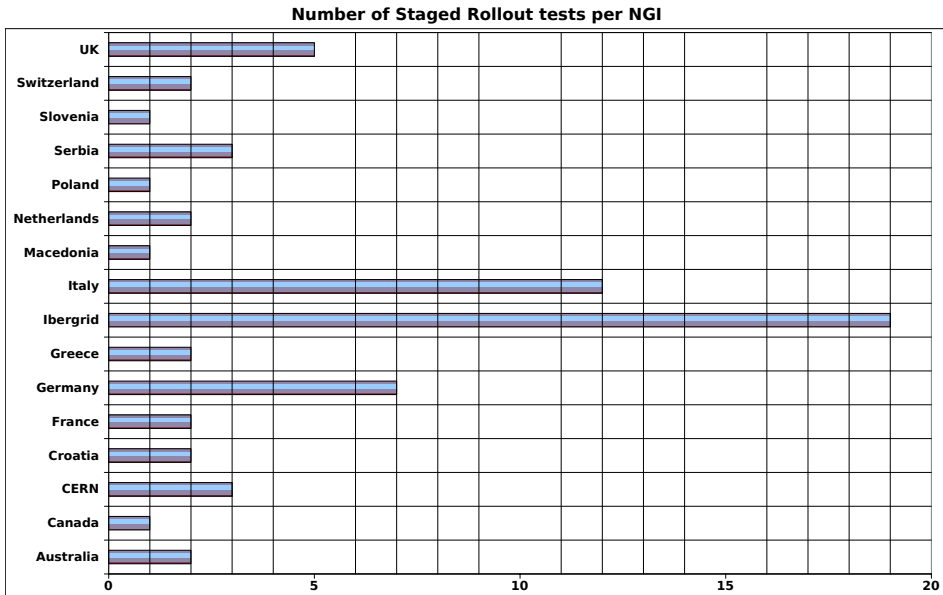


Fig. 4. The number of staged rollout tests per Regional or National Grid Initiative; period between May 2010 and March 2011

4.2 Early Adopters

The number of Early Adopter sites has increased since the beginning of the project, from an initial set of around 25 teams to around 42 at the time of writing. Consequently most gLite software components are covered for testing by at least one Early Adopter team, contrary to the situation in the beginning. Furthermore, ARC and UNICORE components are also covered, spanning all components that are part of the first EMI release².

The aim is to have as many Early Adopter teams as possible to cover different deployment and heterogeneous scenarios as well as redundancy to perform the test (when given team or teams are unavailable). A portal, as shown in Figure 5, was

² <https://savannah.cern.ch/task/?group=emi-releases>

made available to monitor the number of Early Adopter teams available per software component.

The work performed by the Early Adopter teams was briefly described above. One common issue with new teams is to find all the needed information to perform the staged rollout test, and produce the final outcome. Part of this issue comes from the use of repositories with unintuitive structure that changes quite often. The EGI repository structure should solve this problems gradually.

Furthermore, the initial information for installation and configuration provided by the Technology Provider has improved considerably during the course of the first project year.

Another aspect to bring more Early Adopter teams into the process was the possibility to declare the service under Staged Rollout test in the Grid Operations Centre database, so that the infrastructure operations teams monitoring the status of the production services are aware of those services, taking that fact into account in the reliability and availability service metrics.

Presently the major challenge is to move the Early Adopters of gLite 3.2 components to the corresponding EMI 1.0 ones where major deployment changes are expected while using a stable production infrastructure.

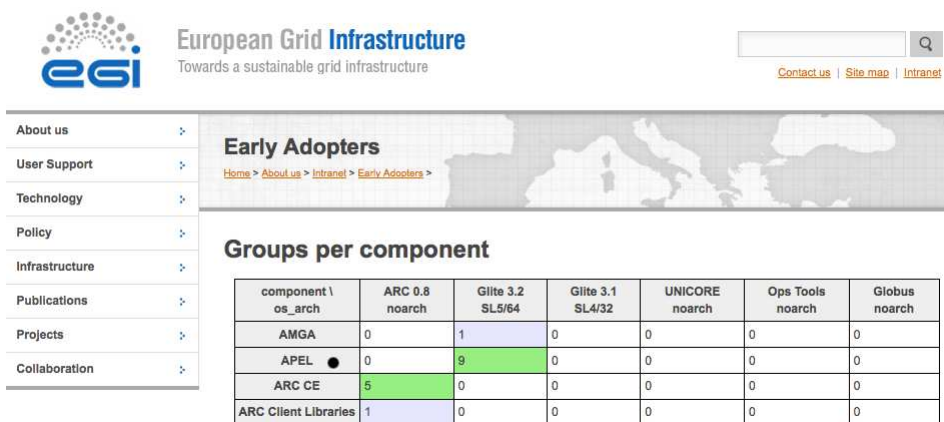


Fig. 5. The number of Early Adopters covering each software component. Hosted <https://www.egi.eu/earlyAdopters/teams>

5 NEAR FUTURE: EMI 1.0 RELEASE

The EMI project gathers components from four major middleware providers: gLite, ARC, UNICORE and dCache. The first major release called EMI 1.0 is expected by the end of April 2011.

All EMI 1.0 components will undergo the Software Provision workflow described in the previous sections, the ones accepted will be gathered in the UMD release. The

workflow has been designed and implemented in such way that each component can be processed independently from any other and in parallel. The workflow described earlier will be tested soon with some agreed components in order to exercise the procedure both in terms of tools and human players.

Currently the EGI infrastructure is largely based on gLite components as well as dCache, consequently all operational and monitoring tools have been targeted to these middleware components. There is also a significant fraction of sites deploying ARC components particularly in the Nordic countries. Some level of integration between ARC components and the operational tools has already been achieved. During the first year of the project a large effort has been made to achieve a seamless integration of both ARC and UNICORE with these operational tools.

The first major release will imply a large effort from all parties: “Quality Control” teams, “Verification” teams, Early Adopters and Staged Rollout managers. The main points are as follows:

- It is the first major release and consequently the first real use of the full workflow: tests with the full chain of procedures are undergoing and it may result in some more fine tuning.
- It is known to EGI that this first release of EMI is backward incompatible [4] in terms of deployment and configuration, but not in terms of interfaces and interaction with current production services. This will imply additional effort from the Early Adopter teams.

The latter point above is considered as the most problematic because the current Early Adopter teams committed for a given gLite component will have to move to the corresponding EMI one. Since the staged rollout test should primarily be done in production nodes, it will imply complete re-installation and configuration of production services. One way to work around this, at least for some services, is to deploy new instances of the service in parallel with the production ones. In any case it will imply additional effort with respect to the currently deployed components.

Going a step further: when the new release gains General Availability it is not expected that a large fraction of the production infrastructure performs a major deployment of the new release. Partly due to the points made above for the Early Adopter sites, but also not to disrupt the services being provided to the Virtual Organizations and users.

Furthermore, the support schedule for gLite 3.1 and 3.2, as well as for each major EMI release is already set. As such, all production resource infrastructure managers and users should be aware of it and plan ahead their schedules to upgrade the production instances. The gLite 3.2 has an end of support date of April 2012, coinciding with the EMI 2.0 major release.

6 CONCLUSIONS AND PROSPECTS

The *Software Provision Process for EGI* has its great challenge in the months following April 2011 with the first major release of EMI. This means that the full chain of the workflow will have to be fully implemented and and working, but more importantly to *convince* the Early Adopter teams, and after the rest of the resource infrastructure managers to move to the new versions of the software components.

It is perceived that the deployment of new software versions into the productions infrastructure will happen during a long time span, and it may happen that unsupported versions of components or services be found in the production infrastructure. This has happened in the past and even today, although one very important point is that a clear support schedule for any given piece of software is known well in advance as is the case we are facing now with both gLite 3.2 and EMI major releases.

Acknowledgments

This work is partially funded by the EGI-InSPIRE (European Grid Initiative: Integrated Sustainable Pan-European Infrastructure for Researchers in Europe) is a project co-funded by the European Commission (contract number INFSO-RI-261323) as an Integrated Infrastructure Initiative within the 7th Framework Programme. EGI-InSPIRE began in May 2010 and will run for 4 years. Full information is available at: <http://www.egi.eu/>.

REFERENCES

- [1] EGI Design Study. EGI Blueprint. <http://www.eu-egi.eu/blueprint.pdf>.
- [2] EGI Web portal. <http://www.egi.eu/>.
- [3] European Grid Initiative: Integrated Sustainable Pan-European Infrastructure for Researchers in Europe. EGI-InSPIRE proposal..
- [4] European Middleware Initiative (EMI). <http://www.eu-emi.eu/>.
- [5] Initiative for Globus in Europe (IGE). <http://www.ige-project.eu/>.
- [6] FERNÁNDEZ, C.: MS503: Software Provisioning Process. <https://documents.egi.eu/public/ShowDocument?docid=68>.
- [7] Global Grid User Support. <https://gus.fzk.de/pages/home.php>.
- [8] EGI Request Tracker (RT). <https://rt.egi.eu/>.
- [9] EGI SA2 and SA1.3, New Software Release Workflow. https://wiki.egi.eu/wiki/NSRW_IMPLEMENTATION_RT.
- [10] ANGELOU, E. et al.: MS504: EGI Software Repository Architecture and Plans. <https://documents.egi.eu/public/ShowDocument?docid=89>.
- [11] The EGI Document Database. <https://documents.egi.eu>.
- [12] FERNÁNDEZ, E. et al.: UMD Quality Criteria. <https://documents.egi.eu/document/240>.

- [13] The EGI wiki pages. https://wiki.egi.eu/wiki/Main_Page.
- [14] DAVID, M. et al.: Provisioning of Grid Middleware for EGI in the framework of EGI-InSPIRE. 4th Iberian Grid Infrastructure Conference Proceedings, Braga, Portugal, May 24–27, 2010, pp. 36–47.
- [15] DAVID, M.: MS402: Deploying Software into the EGI production infrastructure. <https://documents.egi.eu/public/ShowDocument?docid=53>.



Mário DAVID has received his Ph.D. in high energy physics in 2002 from the University of Lisbon. Since then he has been working in LIP Lisbon Computing Center in Grid technologies. He has been collaborating in several European Grid projects such as EDG, Crossgrid, the EGEE series (I, II and III), as well as in Int-EU-Grid. Currently he is the task leader of EGI TSA1.3: Service Deployment and Validation.