# EGI-InSPIRE

## DEPLOYING SOFTWARE INTO THE EGI PRODUCTION INFRASTRUCTURE

## EU DELIVERABLE: MS402

| | |
|---|---|
| Document identifier: | EGI-MS402-53-V3-tf.odt |
| Date: | **11/07/2010** |
| Activity: | **SA1.3** |
| Lead Partner: | **LIP** |
| Document Status: | Draft |
| Dissemination Level: | **PUBLIC** |
| Document Link: | https://documents.egi.eu/document/53 |


Abstract

This document details the processes and procedures of software rollout into the EGI production infrastructure. This includes the EGI Unified Middleware Distribution (UMD) and Operational Tools provided by JRA1 activity. The UMD will be composed by middleware components from the EMI project, namely gLite, ARC and UNICORE, Globus from the Initiative for Globus in Europe (IGE) and other software component provided by the community.

## Delivery Slip

| | Name | Partner/Activity | Date |
|---|---|---|---|
| **From** | | | |
| **Reviewed by** | **Moderator:**<br>**Reviewers:** | | |
| **Approved by** | **AMB & PMB** | | |

## Document Log

| Issue | Date | Comment | Author/Partner |
|---|---|---|---|
| 0.3 | 06/25/10 | First draft (incomplete) | Mario David / LIP |
| 0.4 | 06/28/10 | Second draft (comments from Michaela Lechner) | Michaela Lechner / KTH |
| 0.5 | 06/29/10 | Additional material | Mario David / LIP |
| 1.0 | 06/30/10 | First complete draft | Mario David / LIP |
| 2.0 | 07/08/10 | Review and comments from sa1 | Mario David / LIP |
| 3.0 | | | |

## PROJECT SUMMARY

To support science and innovation, a lasting operational model for e-Science is needed − both for coordinating the infrastructure and for delivering integrated services that cross national borders.

The EGI-InSPIRE project will support the transition from a project-based system to a sustainable pan-European e-Infrastructure, by supporting 'grids' of high-performance computing (HPC) and high-throughput computing (HTC) resources. EGI-InSPIRE will also be ideally placed to integrate new Distributed Computing Infrastructures (DCIs) such as clouds, supercomputing networks and desktop grids, to benefit the user communities within the European Research Area.

EGI-InSPIRE will collect user requirements and provide support for the current and potential new user communities, for example the ESFRI projects. Support will also be given to the current heavy users of the infrastructure, such as high energy physics, computational chemistry and life sciences, as they move their critical services and tools from a centralised support model to one driven by their own individual communities.

The objectives of the project are:

1. The continued operation and expansion of today's production infrastructure by transitioning to a governance model and operational infrastructure that can be increasingly sustained outside of specific project funding.
2. The continued support of researchers within Europe and their international collaborators that are using the current production infrastructure.
3. The support for current heavy users of the infrastructure in earth science, astronomy and astrophysics, fusion, computational chemistry and materials science technology, life sciences and high energy physics as they move to sustainable support models for their own communities.
4. Interfaces that expand access to new user communities including new potential heavy users of the infrastructure from the ESFRI projects.
5. Mechanisms to integrate existing infrastructure providers in Europe and around the world into the production infrastructure, so as to provide transparent access to all authorised users.
6. Establish processes and procedures to allow the integration of new DCI technologies (e.g. clouds, volunteer desktop grids) and heterogeneous resources (e.g. HTC and HPC) into a seamless production infrastructure as they mature and demonstrate value to the EGI community.

The EGI community is a federation of independent national and community resource providers, whose resources support specific research communities and international collaborators both within Europe and worldwide. EGI.eu, coordinator of EGI-InSPIRE, brings together partner institutions established within the community to provide a set of essential human and technical services that enable secure integrated access to distributed resources on behalf of the community.

The production infrastructure supports Virtual Research Communities − structured international user communities − that are grouped into specific research domains. VRCs are formally represented within EGI at both a technical and strategic level.

# TABLE OF CONTENTS

# 1. INTRODUCTION

## 1.1. PURPOSE

This document describes the processes and workflows for the rollout of software components into the EGI production infrastructure.

EGI-InSPIRE will use the Unified Middleware Distribution (UMD). The UMD integrates middleware components provided by the European Middleware Initiative project (EMI), by the Initiative for Globus in Europe (IGE), and other external and internal sources called "Community Contributions". EMI will provide a set of services or components with a given base release provided by Product Teams (PT).

Services from the gLite, ARC and UNICORE middleware stacks will be included in the EMI release. Globus components, presently provided by the Virtual Data Toolkit (VDT), will be provided by IGE in the near future. Some components which are contributed by the community are for example; the batch systems support and integration in the middleware or the packages containing the Certification Authorities (CAs) approved by the International Grid Trust Federation (IGTF).

The Operational Tools such as the "Regional Dashboard" or the Nagios monitoring tools, are key software components for a reliable and stable operation and monitoring of the infrastructure. Only the tools which have been, or are being, regionalized will undergo through the staged-rollout process.

From here on we will call any entity providing any piece of software which falls in the previous description as "Software Providers".

We give below a summary of the major categories of software in use or expected to be used by EGI-InSPIRE:

- UMD provided by EGI-InSPIRE is composed of the following sets of components:
    2. EMI release containing gLite, ARC and UNICORE middleware stacks.
    3. Globus middleware provided by IGE.
    4. Community Contributions, as the Certification Authority packages or the batch system integration into the several middleware stacks.
- Operational Tools provided by EGI-InSPIRE JRA1 activity: Regional Dashboard, Nagios monitoring tools, etc.

## 1.2. APPLICATION AREA

This document is a formal deliverable for the European Commission, applicable to all members of the EGI-InSPIRE project, beneficiaries and Joint Research Unit members, as well as its collaborating projects.

## 1.3. REFERENCES

**Table 1: Table of references**

| | |
|---|---|
| **R 1** | MS503: Software Provisioning Process |
| **R 2** | MS702: Establishing the Operational Tool product teams |
| **R 3** | MS501: Establishment of the EGI Software Repository and associated support tools |

| | |
|---|---|
| **R 4** | MS502: Deployed Middleware Support Unit Operations Procedures |
| **R 5** | https://twiki.cern.ch/twiki/bin/view/EGEE/MiddlewareSupportJobDescription |
| **R 6** | |

## 1.4. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGI-InSPIRE "Document Management Procedure" will be followed:

https://wiki.egi.eu/wiki/Procedures

## 1.5. TERMINOLOGY

A complete project glossary is provided in the EGI-InSPIRE glossary:

http://www.egi.eu/results/glossary/.

The table below contains further terminology not provided in the previous location (t**his needs to be checked against the egi glossary**):

| | |
|---|---|
| ARC | Advanced Resource Connector (middleware stack) |
| EA | Early Adopter site |
| EGEE | Enabling Grids for E-sciencE |
| gLite | Lightweight Middleware for Grid Computing (middleware stack) |
| MU | EGI Middleware Unit |
| OU | EGI Operations Unit |
| PT | Product Team |
| UMD | Unified Middleware Distribution |
| UNICORE | Uniform Interface for Computing Resources (middleware stack) |
| VDT | Virtual Data Toolkit |
| WLCG | Worldwide LHC Computing Grid |

# 2. SOFTWARE ROLLOUT PROCESS

TF: an executive summary is needed

Staged-rollout is the process defined in EGEE-III to ensure that middleware is swiftly and safely deployed in the production infrastructure. It is based on the experience from the EGEE pre-production service, which consisted in the procurement of a dedicated infrastructure for software testing prior to release. Staged-rollout enhances the pre-production service by integrating new software releases into the production infrastructure without requiring extra testing effort from the production VOs supported by the sites contributing to the staged rollout. A detailed description of this process is documented in [R5]. TF: Which process?

While EGI staged rollout is based on the principles already adopted in EGEE, still EGI peculiarities need to be faced to ensure a smooth transition.

In EGEE, the staged-rollout process design and implementation was targeted to a single middleware stack, i.e. gLite, whereas the EGI infrastructure will be based on different middleware stacks. This fact leads to a broadening of the scope of the staged-rollout process and procedures.

While the variety of deployed middleware will increase in the short term the heterogeneity of the infrastructure, different middleware stacks will be harmonized in the medium term thanks to the effort of projects such as the European Middleware Initiative (EMI). More partners will need to contribute to staged-rollout in order to ensure that all components are properly verified. Also the regionalized operational tools developed in the framework of JRA1 will undergo the same release and testing process.

The Operational Tools used in the infrastructure will follow a similar procedure to the middleware stacks, (see details later). This was not the case for EGEE.

Most of the tools used in the Software Rollout process for EGEE, where hosted at CERN: web pages, twiki, task tracker, mailing lists, etc.. Similar tools will be needed for EGI, and will have to be migrated and adapted to the EGI era.

It should be noted that during the EGEE project series middleware development efforts were mostly internal to the project, and only the maintenance and development of operational tools is now within the EGI-InSPIRE project. This requires proper tools and clear reporting channels to be put in place.

## 2.1. INTEROPERABILITY AND NEW TYPES OF RESOURCES

The EGI infrastructure will have several middleware stacks deployed. Presently, as a result of the EGEE and WLCG projects, only gLite is fully integrated into all the operational tools, whilst ARC has been partially integrated, and Globus and UNICORE operational integration is still to be implemented. Comprehensive integration is a short-term objective of the project.

In a second phase, it is expected that site administrators and user communities will provide requirements for the interoperability between different middleware stacks, and that the EGI infrastructure will be integrated with new types of resource, such as digital libraries and repositories, desktop grids, High Performance Computing, etc.

## 2.2. SOFTWARE VERSIONING SCHEME

A new version of any software component can be categorized as follows [R1]:

1. **Emergency release** (when needed): it fixes critical functionality problems and/or serious security vulnerabilities. It is backwards compatible.

2. **Revision release** (at most once every two weeks): it provides bug fixes and is backwards compatible.
3. **Minor release** (at most once per month): it provides new functionality and is backwards compatible.
4. **Major release** (at most once per year)
    - It offers new functionality, not necessarily backwards compatible.
    - It includes a new service.

A given UMD major release will contain baseline major versions of a set of components. These major versions are subject to the agreement between EGI and the software providers, and will be detailed in a roadmap document.

Any middleware component can be updated only up to a minor release within any Major UMD release. The major releases of any given component may only be included in the next major UMD release, depending on the roadmap.

It is foreseen that all categories of component updates will undergo the staged-rollout process, but the time-lines and the extensiveness of the staged-rollout will vary according to the category. The sole possible exception is an emergency release, which may skip staged-rollout under exceptional well document circumstances to be evaluated in a case by case basis.

The components classified as "Community Contributions" and "Operational Tools" will follow a similar procedure.

EGI-InSPIRE will accept only certified and validated updates provided by the software providers. The validated components will undergo the staged-rollout procedure. If successful, can then be widely deployed in the production infrastructure.

If bugs or issues are found during the staged-rollout phase in a given component for which some solution or workaround is proposed, the fix(es) should be communicated and implemented by the respective software provider. Middleware components with workarounds to bugs or issues should be avoided in production.

Each middleware stack is in general composed of several services. As such, it is the responsibility of the EGI-InSPIRE Middleware Unit, to provide requirements for the integration of services into a given capability. TF: I don't understand this statement, can you clarify?

## 2.3. TOOLS SUPPORTING THE STAGED ROLLOUT PROCESS

Tools will be used at several stages of the staged-rollout process. These are already ready for deployment in EGI. A list including some usage details is given below:

1. **EGI Single Sign On** (EGI-SSO: TF put this into a reference, do the same for the urls below, they shouldn't be included in the document body https://www.egi.eu/sso/): contains user accounts and LDAP groups, such as the the Early Adopter group of users.
2. **EGI RT** (https://rt.egi.eu/): is a "Request Tracker". A queue has been created to track through tickets the various stages of the software rollout process, from the moment it is declared ready and made available by the software providers until it is released to the production infrastructure at large. TF: Provide a reference to the specific RT queue of interest here
3. **EGI Wiki** (https://wiki.egi.eu/): it is used to publish more dynamic information such as documentation of all releases with deployment advisories, links to release notes, and information about certification and validation of software components.

4. **EGI Repositories** (http://repository.egi.eu/): these provide access to the software packages that are part of the UMD distributions, during the various stages of the software lifecycle. Details are given in [R3].
5. **EGI Mailman** (https://mailman.egi.eu/mailman/listinfo): it hosts and manages several mailing lists needed by EGI staged-rollout, in particular:
    5.1. early-adopters-arc@mailman.egi.eu for ARC early adopter site administrators.
    5.2. early-adopters-glite@mailman.egi.eu for gLite early adopter site administrators.
    5.3. early-adopters-globus@mailman.egi.eu for Globus early adopter site administrators.
    5.4. early-adopters-opstools@mailman.egi.eu for operational tools early adopter site administrators.
    5.1. early-adopters-unicore@mailman.egi.eu for UNICORE early adopter site administrators.

## 2.4. SOFTWARE ROLLOUT WORKFLOW

This section details the workflow of new software component versions from the time when they are released by the software provider to the time when the component is distributed for deployment in the EGI production infrastructure.

1. Software providers
    1.1. A new version of a component has been produced. At this stage or earlier, the software provider creates a new ticket in the "**staged-rollout**" RT queue. It is required that the software provider is notified on any change in state of the ticket. This ensures a close and direct contact or a quick action if, at any step, there is a problem or issue with the component.
    1.2. At this stage the version of the component is in the "**Certified**" status.
    1.3. The Repository Manager has pulled the packages (rpm, deb, tar, etc.) into the EGI repository called "**Upload**", and triggers the move of the packages to the "**Scratch**" repository after automatic checksum verification. If this step fails, the process is repeated.
    1.4. The software provider has to provide the following information in the RT ticket:
        1.4.1. Release notes, or the link to them. The release notes should link to the advisory written by the Software Vulnerability Group (SVG), in the case of a software vulnerability.
        1.4.2. Changelog, or the link to it.
        1.4.3. Certification report(s) (from the agreed quality assurance documentation and tests) or a link.
        1.4.4. Link to documentation: users manual, system administration manual, etc. The documentation should be updated if applicable, for example if the release introduces new functionality.
        1.4.5. Links to all bugs, issues, features in this new release.
    1.5. The EGI Repository team, assigns the ticket to the EGI Middleware Unit group, after step 1.3 is successfully accomplished.
2. EGI Middleware Unit (MU) [R1]:

2.1. Verifies the new version. This includes the verification of all information provided according to step 1.4.

2.2. Triggers the packages into the "**Staged Rollout**" repository in case of successful verification. If there are problems or issues, the Software Provider is immediately notified. After discussion a countermeasure will be agreed upon to solve the issue at hand. This measure can include the rejection of the component.

2.3. Inserts the URL of the repository or the URL of all the packages in the RT ticket. This will be used later by the early adopters to preform the staged-rollout.

2.4. Sets the status of the RT ticket to "**verified**". At this point the ticket will be assigned to "**staged-rollout**" group which is thus notified that new packages are ready for the staged-rollout. [TF I would like to see a state diagram which illustrates the changes of the RT ticket status from one step to the other]

3. EGI Operations Unit (OU):

3.1. The "**staged-rollout**" group notifies the corresponding "**early-adopters**" group – through a ticket – that a new version of a given middleware component or operational tool is ready for staged-rollout.

3.2. Any early adopter participating staged-rollout of the relevant component should enter the RT and "**accept**" the ticket **within one working day** after the notification.

3.2.1. The first early adopter "**accepting**" the ticket, sets its status to "**rolling-out**".

3.2.2. Any other early adopter can also "**accept**" the same ticket.

3.2.3. Any number of early adopters can contribute to staged-rollout for any given component.

3.2.4. From this point forward, only the early adopter teams that accepted the staged-rollout for a given component, will receive notifications from that ticket, as well as the software providers.

3.3. The "**staged-rollout**" group checks after one day, if all tickets are in the state "**rolling-out**". If any are missing, the group will take action. This might be, by getting in contact with partners not yet in "rolling-out" state to solicit the start of staged rollout or by calling for new partners.

3.4. A given early adopter should primarily "**accept**" staged-rollout for the components it has officially committed to. However, early adopters are encouraged to participate to staged-rollout of additional components, even if not on a regular basis.

3.5. It is mandatory to notify the software providers if any problems or issues are found.

3.6. After staged-rollout, the partner is requested to fill in a report (a report template will be provided in the RT system) and to set the ticket status to "**done**", this meaning that the task was completed.

3.7. Only after all early adopters have set their status to "**done**" on the ticket, the ticket can be closed by the "**staged-rollout**" group. A summary in writing is produced to summarize the results and possibly provide further comments.

3.7.1. The status report can be: **<Success|Warning|Reject>**.

3.7.2. The "**Warning**" can be set when slight changes to the release notes, to installation or configuration are requested. The staged rollout manager is responsible of this.

3.8. A grace period of up to five working days, is foreseen for the new versions to be "exposed" to a production environment. [TF: I don't understand why this grace period is needed. Time between release and readiness for production has to be the minimum possible]

3.9. Closing the ticket with "Success|Warning" triggers the transfer of the packages from the "**Staged Rollout**" repository to the "**Production**" one. [TF Please also clarify how site managers are notified about a new release being ready for deployment]

Figure 1 depicts the workflow previously described, while the following table describes the expected time-lines in days, for each major step.

[TF: I don't understand the meaning of "item" in the table below. Is it "step"?]

[TF: for deadlines < 1 day, I think it is clearer if timelines are defined in hours]

[TF: is 1 day equivalent to 24 hours or 8 business hours?]

| | Item 1:<br>Software providers<br>Repository administrators | Item 2:<br>EGIMmiddleware<br>Unit Verification | Item 3:<br>EGI Operations Unit<br>Staged Rollout | Total number of days |
|---|---|---|---|---|
| Emergency | 0.4 | 0.3 | 0.3 | 1 |
| Revision | 0.5 | 0.5 | 1 (+5 days grace period) | 2 (to 7) |
| Minor | 0.5 | 0.5 | 1 (+5 days grace period) | 2 (to 7) |
| Major | 0.5 | 1 | 2 (+5 days grace period) | 3.5 (to 8.5) |

For "Emergency" releases the time-lines shown above can be made shorter, or some stages can be skipped. This will be evaluated on a case by case basis.

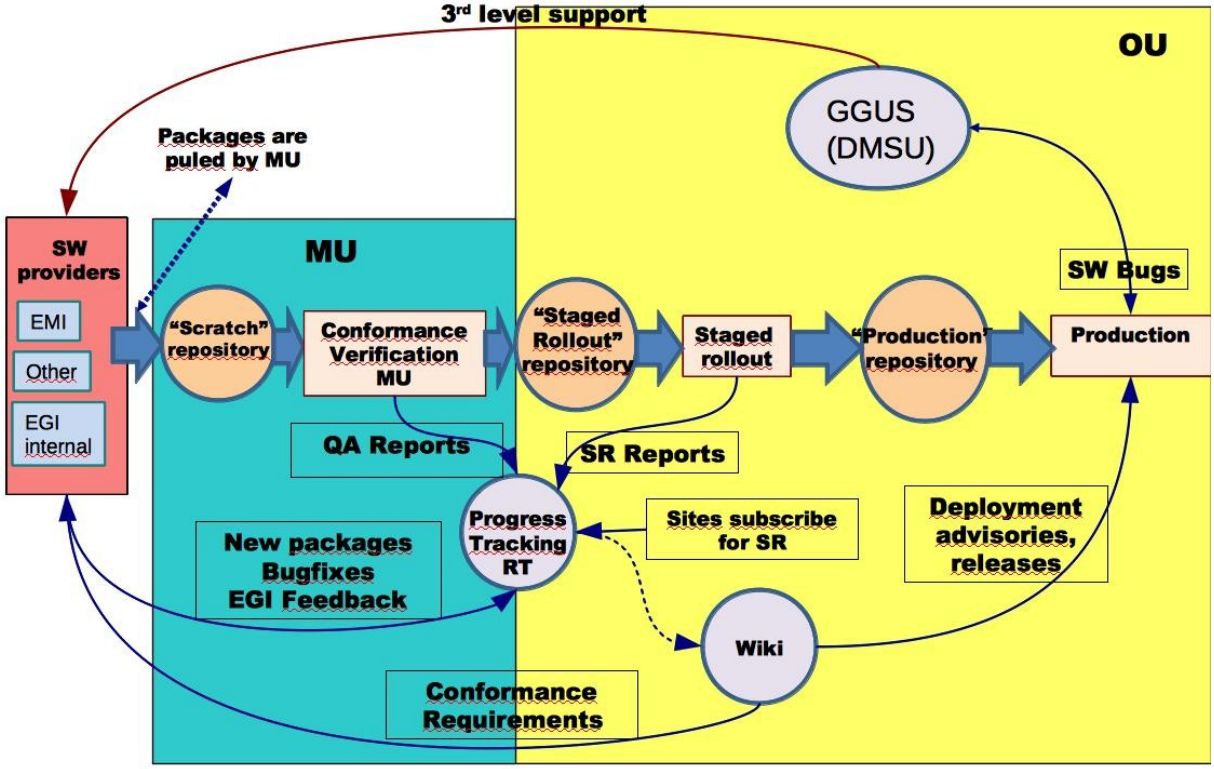


**Fig. 1: Middleware rollout process; from the software provider until deployment into the production infrastructure.**

[TF "plued" should be "pulled". The role of the DSMU (shown in the picture) is not explained, please add text]

## 2.5. PILOT SERVICES AND OTHER MIDDLEWARE TESTING PROCESSES

The staged-rollout is the common process to release new middleware component versions that are already certified into the production infrastructure. Additional processes are possible prior to certification for testing a new middleware component.

1. Pilot services
   1.1. Occasionally when a new middleware component has to be integrated with existing components, a pilot service instance can be needed to test this integration (the client is introduced first and the server part at a later stage). An example of this process was the testing of the pilot ARGUS service (gLite Authorization service).
   1.2. Middleware components may need pilot testing to verify robustness and scalability under high load, and/or to tune the service configuration. An example of this was the testing of the CREAM Computing Element to test its scalability and readiness as a replacement of the LCG Computing Element.
2. Alpha and Beta testing. Alpha and Beta testing are necessary during the early phases of the software development lifecycle. This is particularly important for complex and critical middleware components. Examples of these are: dCache, StoRM, and the File Transfer Service.

In any of the cases described previously, a tight collaboration between the interested parties is needed: a set of site managers, end-user communities testing the software in their own computing frameworks, and the developers. It is accepted that the versions of these middleware components are deployed and used in production by the interested sites and user communities, before these components reach the production repositories for general availability.

The staged rollout process is required from all components regardless of the amount of testing carried out during the early development stages before final certification by the Product Teams.

## 2.6. OPERATIONAL TOOLS

Availability and reliability measurement, monitoring, accounting, user and operational support in the EGI grid infrastructure all rely on several operational tools developed in the framework of the EGEE project series. Tool development is an ongoing effort and is part of the EGI-InSPIRE JRA1 work programme [R2].

While different harmonized middleware stacks are supported by EGI for deployment in the resource centres, the central and distributed instances of the operational tools are operated by a small number of partners committed to provide such services for National or Regional Grid Initiatives, or even for the whole EGI. The aforementioned staged rollout process is equally applicable to operational tools and middleware components.

The list of tools which will undergo the staged-rollout process is given below [R2]:

1. Nagios based monitoring infrastructure.
2. Regional operations dashboard.

These tools that are ready for distributed deployment by regional or National Grid Initiatives will follow the same path as the other software components. On the other hand, no staged rollout is foreseen for central tools for which a single instance exists. As soon as a central tool will be also available for distributed deployment, the tool will be requested to participate to staged rollout. The other tools expected to be regionalized soon are:

1. Accounting portal and repository.
2. Grid Operations Centre DataBase (GOCDB).

# 3. EARLY ADOPTERS

An Early Adopter (EA) is a **production** site committed to perform the staged-rollout for one or more middleware components or operational tools.

The EA sites should pursue as much as possible the deployment of these new versions in a production environment. The final deployment layout is always a final decision of the site manager.

It is assumed that any software component whose status was set to "verified" by the EGI Middleware Unit, is of production quality and can be safely deployed in production without causing interruption of the service provided. Thus the EA sites are just the first partners to be notified about a new version of any given software component, and are requested to report back about theoverall process outcome.

It is recommended that in the staged-rollout phase the new version of the software components is exposed to end-users under normal production loads.

The EA site manager defines the deployment configuration that is applicable to the respective site. It must be taken into account that the site "Reliability and Availability" **should not** be affected in case of problems with the staged rollout components. For that purpose, the service nodes in the staged-rollout might be identified through a specific flag (e.g. "**beta**") within the GOCDB, in order to let the Availability Calculation Engine ignore such components in its monthly statistics. Another possibility is to have a new special downtime state during the period the site is under the staged-rollout process.

## 3.1. ENGAGEMENT OF EA SITES

Currently there are several sites which have committed to do staged-rollout for several components. These include gLite, UNICORE, Globus and Nagios.

From the operational point of view, an EGI SSO group called "**staged-rollout**" has been created with the purpose of managing the SR process. This group is the manager of the corresponding RT queue identified by the same name.

The following EGI SSO groups and a related mailing list were also created:

- early-adopters-arc
- early-adopters-glite
- early-adopters-globus
- early-adopters-opstools
- early-adopters-unicore

The engagement of a new EA site is triggered by the creation of a ticket in RT by the respective site administrators, assigning it to the queue "**staged-rollout**". The following information should be provided in the ticket:

- Site name.
- Site administrators name.
- Site administrators e-mail.
- Component (possibly the OS and architecture).

The "**staged-rollout**" manager will manage the addition of the new team members in the EGI SSO groups mentioned above. From here on, the new EA administrators will receive notifications when new versions of software are available for staged-rollout.

### 3.2. CURRENT STATUS

The EGI wiki has a page containing a table with information about all EA sites that have committed to do staged-rollout on given middleware components: https://wiki.egi.eu/wiki/Early_Adopters TF: put this in a reference!. At the time of writing, 23 EA sites (or teams) committed to participate to staged rollout, including one for Nagios (operational tools), one for UNICORE and one for Globus, while all remaining ones are for gLite components.

Staged rollout is of paramount importance to ensure that software is safely deployed by a large scale pan-European infrastructure such as EGI. It is thus expected that more NGIs will participate to staged-rollout soon. These set of sites should form a stable core of staged-rollout partners. However, it is possible to have sites that only commit to staged-rollout for a limited period of time or only for certain releases of some middleware components.

Requesters of new functionalities or new services, if approved, should be required to engage in the staged-rollout phase too, eventually stimulating the participation of new sites.

## 4. METRICS

### 4.1. METRICS FOR EA SITES

Using the RT to track each staged-rollout process, allows some metrics to be calculated:

- The number of EA sites committed to staged-rollout: this value can be taken from the wiki page given in the previous section.
- The number of active EA sites.
- The number of staged-rollout tickets per site.
- The time spent on this activity per site.

### 4.2. METRICS FOR SW COMPONENT RELEASES

For a given software component (middleware or operational tool), the following metrics can be calculated:

- The total number of releases (up to minor releases) in a given time span.
- The number of times a given component was rejected, or had “warning” status, in the staged-rollout phase.

## 5. COMMUNICATION CHANNELS

The communication between the EA site administrators, the “staged-rollout” managers, the EGI Middleware Unit and the software providers, is expected to occur within the RT ticket, during the whole process of software rollout, until the component is released into the production infrastructure.

This allows a close and fast contact with the software providers in case of problems, during any of the stages described in this document. There might be cases where the release is approved for production, even with some workaround. It should be stressed that these situations should be avoided as much as possible. Furthermore, the “opening” of bugs against a given component during the staged-rollout phase, is not ruled out. [TF: can you clarify? Is it possible to open bugs while in staged rollout? Who is allows to do this? The site managers? Or the staged rollout manager? Which bug tracking system is used for this?]

All problems and issues found in the production infrastructure should be reported through GGUS tickets. These tickets will be routed to the second level support team called "Deployed Middleware Support Unit" (DMSU) [R4].

The announcement of any given new version of a software component for general availability, is done through the operations portal broadcast tool to the following groups or mailing lists:

1. Production Site Administrators (taken from the GOCDB).
2. WLCG Tier-1 contacts (static CERN maintained list).
3. inspire-sa2@mailman.egi.eu: EGI SA2 team including the repository administrators, the DMSU support unit, the staff responsible of defining the middleware quality criteria and of their verification.

## 6. PROCESS MANAGEMENT

The staged-rollout process is managed by the "**staged-rollout**" EGI SSO group. This group contains one or two expert users [TF: users of site managers?] from each middleware stack, from the grid interoperation team and from the operational tools community.

Such composition allows everyone on this group to be aware of any new software components versions being processed. The management of the staged-rollout process will be performed by the corresponding experts, including the last step which is the announcement of the new release to production.

## 7. SERVICE LEVEL AGREEMENTS (SLA)

Service Level Agreements (SLAs) will be signed between EGI and EMI for the gLite, UNICORE and ARC middleware stacks, and possibly between EGI and IGE for Globus.

The SLAs will define the level of commitment between third-party Software Providers and EGI, and should drive the stability and robustness of the software through a high level of trust.

The SLA terms and definitions are planned for a later stage of the project. More details will be documented in milestone MS505: "Service Level Agreements with Software Providers".

## 8. YEAR ONE ROADMAP

The roadmap for the first year concerning the TSA1.3 task starts with the transition from the processes and procedures that have been in place in the EGEE project to the ones defined in this milestone for the EGI-InSPIRE project.

This transition should proceed through several phases:

1. Before the EGI project started, the transition from the EGEE staged-rollout manager to the new EGI staged-rollout manager, took place. The procedure has not changed in this phase in order to ensure continuity of the process as perceived by the infrastructure operational staff and site managers performing the staged-rollout.
2. As the EGI started and several tools become available, the staged-rollout manager started the operational transition.
3. The current status is the following:

3.1. New EGI SSO groups have been created and populated for this task. These groups are linked to synchronized mailing lists. These lists are already in use in the staged-rollout process, and the old mailing list have been deprecated.

3.2. The wiki has been populated with information about the staged-rollout process, one page contains the future workflow and another page contains the current one. Another page contains a table with EA sites and contacts. [TF: provide references to these pages]

3.3. Currently he process is still driven through CERN Savannah patches under the "jra1mdw" project.

3.4. The announcement of new versions to the production infrastructure, is performed through the broadcast tool in the Operations Portal.

3.5. The process only includes the gLite middleware stack.

3.6. The repositories are hosted at CERN. Mirroring to the EGI production repositories for gLite, UNICORE and ARC middleware stacks is ongoing.

3.7. [TF the lcg-rollout mailing list is a primary forum for exchange of information about software deployment issues. Are you planning to migrate this?]

4. The next steps are the following:

4.1. To experiment the proposed EGI staged-rollout procedure based on RT starting from the new versions of the operational tools. This will allow to tune technical details in the RT queue "staged-rollout".

4.2. To start using the wiki for software release information [TF: is a RSS feed foreseen?]. This will be initially a duplication of information published in the CERN web pages for the gLite middleware stack. This will also be done for the other middleware stacks. For the operational tools, the EGI wiki should be used from the start.

4.3. To perform staged-rollout also on selected software components, testing the whole workflow, this including the interaction with the EGI repositories.

The main objective by the end of the first year of the project, is to fine tune the new procedure and to apply it to any software component relevant to the EGI production infrastructure using the tools and the workflow described earlier.