



Middleware Rollout: Task TSA1.3

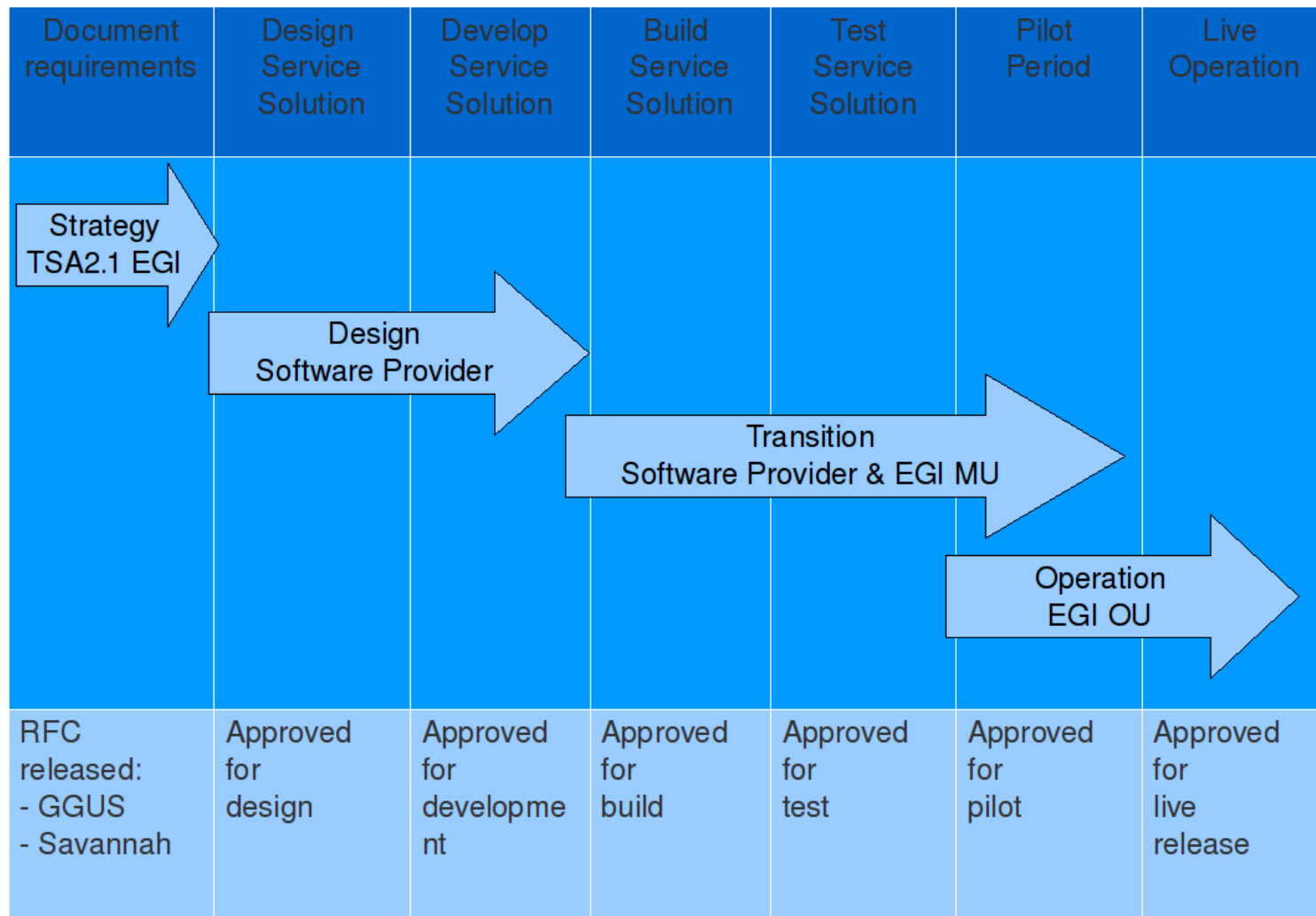
Mario David <david@lip.pt>

(on behalf of IBERGRID ES-NGI and PT-NGI)

- **“Proposal” procedures for EGI InSPIRE**
- **Version scheme of MW components**
- **Operational tools for the MW rollout task**
- **Service Level Agreements**
- **Early Adopters:**
 - Engagement.
- **Pilot Services and other MW testing processes**

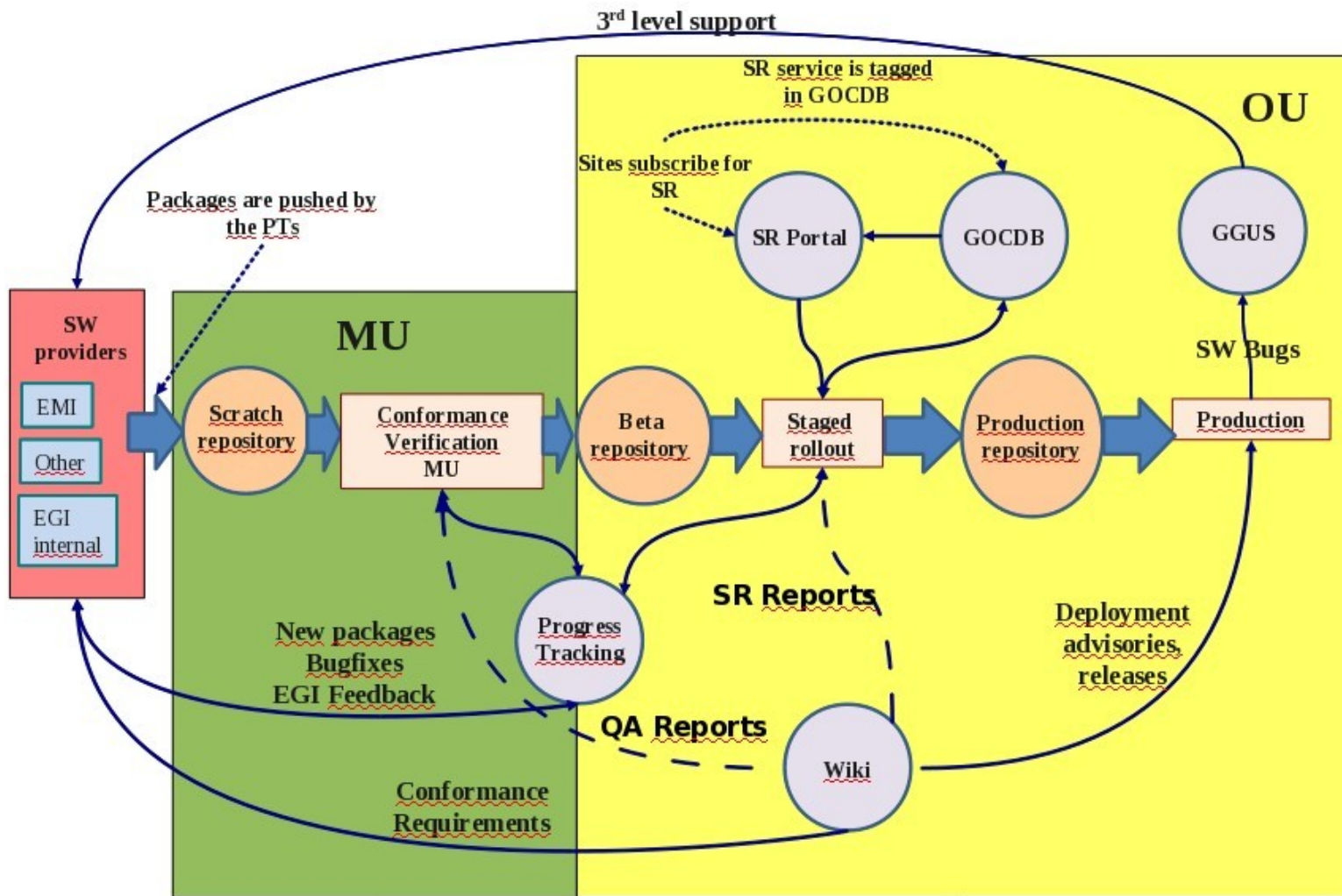
Javier Lopez

SERVICE LIFECYCLE



- **Categories:**
 - 1. Emergency (when needed): bug fixes, or security vulnerability, backward compatible.**
 - 2. Revision release (at most once every two weeks): bug fix, backward compatible.**
 - 3. Minor release (at most once per month): new functionality, backward compatible.**
 - 4. Major release (at most twice per year):**
 - a. New functionality not necessarily backward compatible.**
 - b. New service.**

- **For a given UMD major release, any given MW component can be updated only up to a “Minor Release”:**
 - The “Major Releases” of any given component may only be included in the next major UMD release, dependent on the roadmap.
- **All categories of updates to the components will undergo the SR procedure, though the timelines and depth of the SR may vary with the category:**
 - The sole exception is an Emergency release, for which, under exceptional circumstances to be evaluated in a case by case basis, may skip the SR.
- **EGI-InSPIRE will accept only certified and validated updates provided by the PTs.**
- **In the SR phase, if bugs or issues are found in a given component for which some solution or workaround is proposed, the fix(es) should be communicated and implemented by the PT:**
 - MW components with workarounds to bugs or issues should be avoided in production.
- **The UMD release will be based on services, which may affect one or more node types.**
- **It is the responsibility of the EGI-InSPIRE MU, to provide requirements for the integration of services into node types as is the case in gLite MW.**



- The PTs will interact/communicate with the MU using the tools provided by EGI-InSPIRE.
- Those tools will be under the “egi.eu” domain:
 - **Wiki:**
 - Documenting all releases with deployment advisories,
 - Links to release notes
 - Links to certification and validation of components provided by the PTs.
 - https://wiki.egi.eu/wiki/Main_Page
 - For the MW rollout process: https://wiki.egi.eu/wiki/Middleware:Release_Process
 - Links to the monitoring (nagios, gridview) of all services participating in the SR.
 - *Filter the services based in GOCDB tag “beta”.*
 - *This may give an integrated view of the quality of these services, and eventually of the MW component.*
 - **Progress tracking and task tool:**
 - Follow all the MW process from the moment it is declared ready and made available by the PTs until released into production.
 - Manage all the SR process.
 - *Creation of task teams for each service in SR,*
 - *Notification and reporting capabilities.*
 - *Record and display the effort of each team.*
 - *Define and display metrics for the effort and for the usage of the service during the period of SR.*
 - Feedback to the Software Providers:
 - *Interface or link with GGUS.*
 - Interface with GOCDB:
 - *Fetch the site names,*
 - *Get the services tagged as beta, eventually other information.*

- **Early Adopters Portal:**
 - **web form** where sites can subscribe to do the SR as Early Adopters (EA).
 - Integrated or linked from the WIKI.
 - Inject this information in “**Progress tracking and task tool**” in order to create the “squad” associated to any given SR service.
 - A site subscribes as EA:
 - Site name: *it should be selectable from the GOCDB.*
 - Responsible persons: site squad.
 - Services which the site proposes to perform the SR: all that where tagged “beta” in the GOCDB, and eventually others which are not tagged and the squad decide to do the test in a production instance.
 - The coordinator is notified.
 - The coordinator may contact the site for more information and planing.
- **Repositories: next session**

Entity Level 1	Entity Level 2	Entity Level 3	Comment
Squad Mangement[n]	Site	Site Name	Information taken from the GOCDB: Interface to GOCDB subscription date to the staged rollout
		Site ID	
		Creation date End date	
	Squad	E-mail [n]	Information about the squad given when it subscribed to SR
		Name[n]	email of each person name of each person
	Service[n]	Service name	Some interface to "supported" components/OS/Arch
		Op System	name of package/component
		Architecture	operating system
		Major UMD Release	Architecture
		Creation date	Major UMD Release
End date	start date of subscription to SR for this service		
Task[n]			Automatic creation of tasks and assignment to 1 or more squads
	Subject		containing the name of the component, OS, arch, major UMD release
	ID		task ID (from the task tracker tool)
	Status		status of the staged rollout for that component
	Report		report of the SR, when closing the task
	Creation date		date the task was created
	End date		dated when task is set to done or close
	Patch[j]	Release notes and other documentation List of packages	Interface to product teams trackers? To get all patches/or bugs for this version of the component release notes, documentation location, configuration changes, etc. list of rpms, tarballs, deb, etc.
Monitoring			Link or interface to monitoring tools (nagios, gridview?) selecting only services in SR: interface to GOCDB selecting "beta" tag
WIKI			Advisory deployment: taken from "Release notes and other documentation" of "Patch" and eventually from the "Report" of "Task" Link to "Production" repositories

- **From Javier Lopez.**
 - Operational Level Agreements (OLA) should be established following the best-practices defined in ITIL. In this case OLAs should be agreed between:
 - Software Providers (EMI and others) and EGI Middleware Unit.
 - In a longer term, between EGI Middleware Unit and EGI Operations Unit.
- **Should cover important aspects like:**
 - **A common Configuration Management Database (CMDB):**
 - A database used to store relevant information about each of the middleware components part of UMD and the relationships between them.
 - **Definitive Media Library (DML):**
 - One or more locations in which the definitive and approved versions of all the middleware components are stored.
 - Should be part of the repository information.
 - **Release Unit:** Components that will be normally released together.
 - **Release Identification:** A naming convention used to uniquely identify a Release.

- **An Early Adopter:**
 - Site which has committed to perform the SR for one or more MW components or services and report that test.
- **The following situations may occur:**
 - The site deploys a new service in parallel with the production instance, just for the SR process.
 - The site may have a “*clone*” of the production system (virtual machines for example), where the new version is deployed, but if some problem occurs, the instance can be quickly changed to the one initially in production.
 - The site preforms the SR in the production instance itself, having rollback procedures in place in case of problems.
- **This decision should be made by the EA, though it can have input from the Operations Team.**
- **The EAs choice may evolve with time. More confidence and more robust MW releases.**
- **Whatever the procedure that the site decides to follow in the SR process, it must be taken into account that the site “**Reliability and Availability**” **should not be affected** if problems occur due to the new versions of the MW components. For that purpose it is advisable to “**beta**” tag in the GOCDDB the services participating in the SR.**

- **Open for discussion**; a node type can be divided into:
 - Non-Critical: CE's, WMSs, clients (UI, WNs)
 - Critical : SE's and Catalog services (LFC, AMGA, etc.), VOMS server.
- **Middleware services affecting Non-Critical node types are easier to release into a production environment.**
 - Problems or bugs that may arise will have in general a limited impact to the users and the site.
- **Middleware services affecting Critical node types will have to be dealt more carefully, so has not to lead to data loss or major infrastructure availability blackout.**
- **The MW services affecting the information system are another component where special care must be taken. At the level of a “Top-BDII” and the “Site-BDII”, it may be advisable to have an instance running in parallel or cloned from the one in production.**

- Presently there are several sites which participate in the Pre-Production service, some of those sites have moved services to the production infrastructure, and others are in the process of doing so. This is one essential step towards the adoption of the SR procedure, and for those sites to become EAs.
- A mailing list has been created containing the contacts of these sites:
 - early-adopters@cern.ch.
 - New participants in the SR should be automatically subscribed in this mailing list.
 - This mailing list must be migrated to the EGI domain. The proposal is: early-adopters@mailman.egi.eu.
- It is expected that most of these sites will form the core of the SR, assuring the continuity of this activity with the least disruption as possible during the transition phase from EGEE to EGI.
- It is known that some of the sites, and possibly most will continue beyond EGEEIII assuring this task during the transition.
 - Known exceptions are CERN-PPS and KIAM from the Russian ROC, which will decommission their PPS sites.
- **It is expected that the NGIs that have requested effort in the EGI-InSPIRE proposal for this task, will commit sites to SR as early as possible.**
 - Answers to Gabriel's questionnaire.
 - These set of sites should form a stable core of the SR process.
 - There might be sites that will commit to the SR process only for a limited period of time or only for certain releases of some MW component.
- **Requesters of new functionalities or new services, if approved, should be required to engage in the SR phase eventually committing new sites.**

- **The Staged Rollout is the supported process to release new middleware component versions into the production infrastructure.**
- **Nonetheless, other processes for middleware testing may occur:**
 - Pilot services:
 - Occasionally it may be justified to setup this process when a new middleware component has to be integrated with existing components, phased introduction into production, with careful testing at all stages. Example: ARGUS.
 - Middleware components may need strengthening, improve robustness, test under high loads with configuration tuning. Example: CREAM-CE.
 - Testing Alpha and Beta releases of middleware components under development:
 - In particular complex and critical middleware components. Examples: dCache and StoRM, or the FTS.
- **On any of the cases described previously, there is a tight collaboration between the interested parties:**
 - A set of sites, users communities and the developers of the component.
 - It is usual that the versions of these middleware components are deployed and used in production by the participating sites and user communities, before they reach the production repositories for general availability.
- **Nonetheless, though these components went under heavy testing when they reach the EGI Scratch repositories, they should still be put under the Staged Rollout process.**

- **Do we need a support unit in GGUS: “Staged-Rollout” ?**
- **How is it done the interaction (technical point of view) of MU with PT's/EMI.**
 - Everything through GGUS, or some other “more direct” means?
 - Communication with an EMI body/task or with PT's: decide in a case by case basis??
 - (Production sites/VOs will always go through GGUS).
- **GOCDB4 has a “beta” tag for services participating in SR:**
 - Site wants the service to be production without any reference to that tag.
 - It may be the (only) way to filter all services in SR for monitoring purposes by the MU/OU, a way to check (even indirectly) the quality of the new version of the component.
- **Reliability and Availability, how will it be calculated?**
- **Progress tracking and task tool: interfaces or links to GGUS, GOCDB4, wiki, EMI tools...? Where and how is it needed?**

<https://twiki.cern.ch/twiki/bin/view/EGEE/ProductTeams>

- AMGA
- Authorization
- Batch System Integration
- Compute Element
- Data Management
- dCache
- Information Systems
- Integrated Clients
- Job Management
- Logging and Bookkeeping
- MPI
- Security Infrastructure
- Service Discovery
- VO Management
- **Configuration: YAIM (core)**
- **DM: does it include DPM? LFC?**
- **StoRM: status?**
- **lcg-CE: included in Compute Element?**
- **MyPROXY: included in Sec. Infrastr.?**
- **Roadmap updated?**
- **Dependency tree of the components?**