

European Middleware Initiative (EMI) – Release Process

Doina Cristina Aiftimiei (INFN) EGI Technical Forum, Amsterdam 17. Sept.2010

Overview



EMI **EMI Releases & Release Policies Product Teams RfC & Release Candidate** SW-Rollout Workplan **EMI-1** Roadmap

EMI



European Middleware Initiative (EMI):

- brings together three major European grid middleware providers, ARC, gLite and UNICORE, and one of the leading grid storage solution providers, dCache.
- **Objectives:**
 - enhance Usability
 - removing redundancy and consolidating the services,
 - simplifying the security management without compromising its strength
 - adding integrated support for high level gateways and portals and transparently making use of virtualization to increase resource availability and management.

improve Compatibility

- removing proprietary interfaces in the middleware services and ensuring true interoperability through the adoption of agreed community standards.
- improve Manageability
 - providing standard service configuration, monitoring and instrumentation interfaces and making accounting and other operational information more readily accessible.
- extend Interoperability between grids, supercomputers and emerging computing models like clouds and desktop grids.
- Improve Sustainability
 - establishing collaboration programs with commercial companies, adopting off-the-shelf components to reduce maintenance costs and to facilitate easier adoption by wider user communities.
- Adopts best-practice service provision methods as the ITIL processes or the CMMi guidelines

EMI Release



An EMI distribution includes all the components that are developed within the project and that have reached production quality.

- The EMI release baseline all components from ARC, gLite, UNICORE and dCache used in production
- Successive releases incrementally add:
 - new services
 - new functionality
 - standard compliance
 - New technologies

based on user requirements

Releases & Release Policies

- periodic **major releases**, tentatively delivered *once a year*
 - good balance between stability and innovation.
 - well-defined interfaces, behavior, dependencies for all included components, available on a predefined set of platforms.
 - Can include backward-incompatible changes
 - multiple major releases may be supported according to negotiated end-of-life policies.

Releases & Release Policies (components)

Major Release

- well-defined interface & behavior, potentially incompatible with the interface or behavior of a previous release
- can be introduced only in a new major release of EMI.

Minor Release:

- significant interface or behavior changes backwards-compatible with those of the corresponding major release.
- Few times per year

Revision Release:

- changes fixing specific defects (bugs) found in production and represents the typical kind of release of a component during the lifetime of an EMI major release.
- Every week or two weeks

Emergency Release:

- only very specific bug fixes, typically security-related.
- As need, using emergency release procedures

Project Execution Structure



IME 116162

SFN

Product Teams



- the **services/components implementation teams** responsible to deliver software releases and all associated material.
- not independent from EMI the executive arm of EMI and are collectively responsible to achieve its technical objectives
- perform the required technical tasks:
 - From design to release through implementation, testing and certification and provide 3rd-level support.
- are flexible:
 - formed or closed as products are introduced or obsoleted
 - allow adding or removing services as needed even from external contributors
- provide a transparent and direct method to assign responsibility for a service
- Overall coordination of development, certification, release and repositories management is done at the WP level



Product Teams (2)





Request for Change

Change: addition, modification or removal of authorized, planned or supported service or service component and its associated documentation (ITIL).

Request for Change (RfC) - formal request

- GGUS tickets.
- User requirements (EGI, MCB or similar bodies).
- PTs fixing defects or introducing improvements.
- **Priority** driven development based on severity, impact, urgency, cost:
 - Immediate ASAP in all affected EMI major releases
 - High next release of the affected component, in all affected EMI major releases
 - Medium in the release of the affected component in the next EMI major release.
 - Low there is no target date
- Changes are handled via a tracking tool.



Release Candidate

Release Candidate (RC) - the way to release a change.

- It should contain the following information:
- List of packages and the URLs
- Associated EMI major release.
- Release Notes: non-technical text written in good English giving an overview of the change introduced by the packages.
- List of bugs/feature requests & links to the corresponding item in the tracking tool
- Link to the test report where the test results of the executed tests are included. This should prove that all mandatory tests relevant to the released packages have been executed successfully.
- Link to the relevant documentation (user guides, troubleshooting guides, etc...)
- List of known issues associated with the release.
- Changelog of each of the released packages containing the developer's comments associated with each change.



From RfC to Release

- A set of guidelines are prepared to guide the PTs activity:
 - Software configuration & integration
 - Packaging & releasing
 - Change management (RfC states, release candidates)
 - Metrics generation
 - Certification & Testing





Documentation

- <u>gLite</u>
- ARC
- UNICORE
- <u>dCache</u>

It will also contain the integration of all of them into EMI.

•

Workplan



Year 1

 Mostly focused on security, consolidation of services and libraries, usability. First major release at the end of the year (technical preview around PM6)

Year 2

- Mostly focused on integration and deployment of messaging services, integration of virtualization technology and standardization of interfaces. Second major release at the end of the year
- Year 3
 - New requirements, revision or business and exploitation plans

Sustainability vision after project end

 User-friendly, standard-based middleware is deployed through mainstream OS distributions and supported also by commercial companies



EMI-1 Roadmap





Thank you!

EMI is partially funded by the European Commission under Grant Agreement RI-261611