



Enabling Grids for E-scienceE

UMD, gLite Consortium, EGEE-III Y2

Francesco Giacomini

CCR & INFN-Grid Workshop

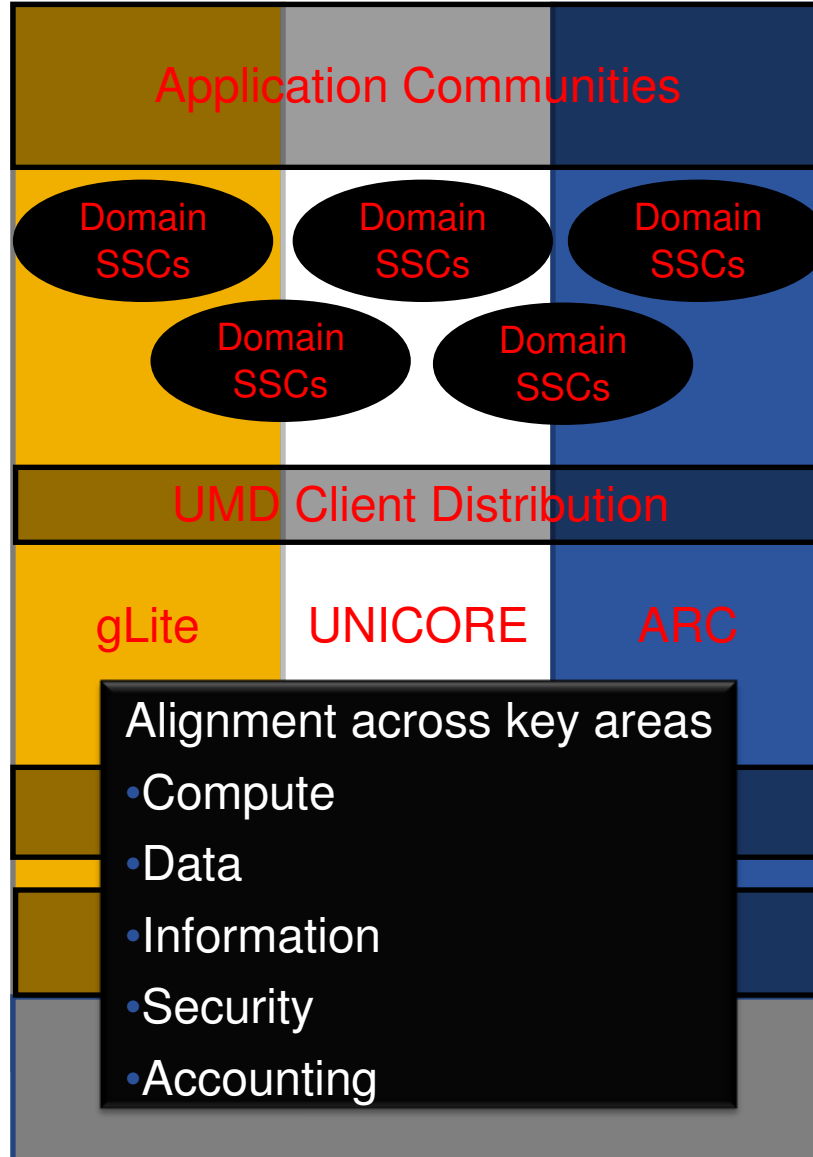
Palau, 11-15 January 2009

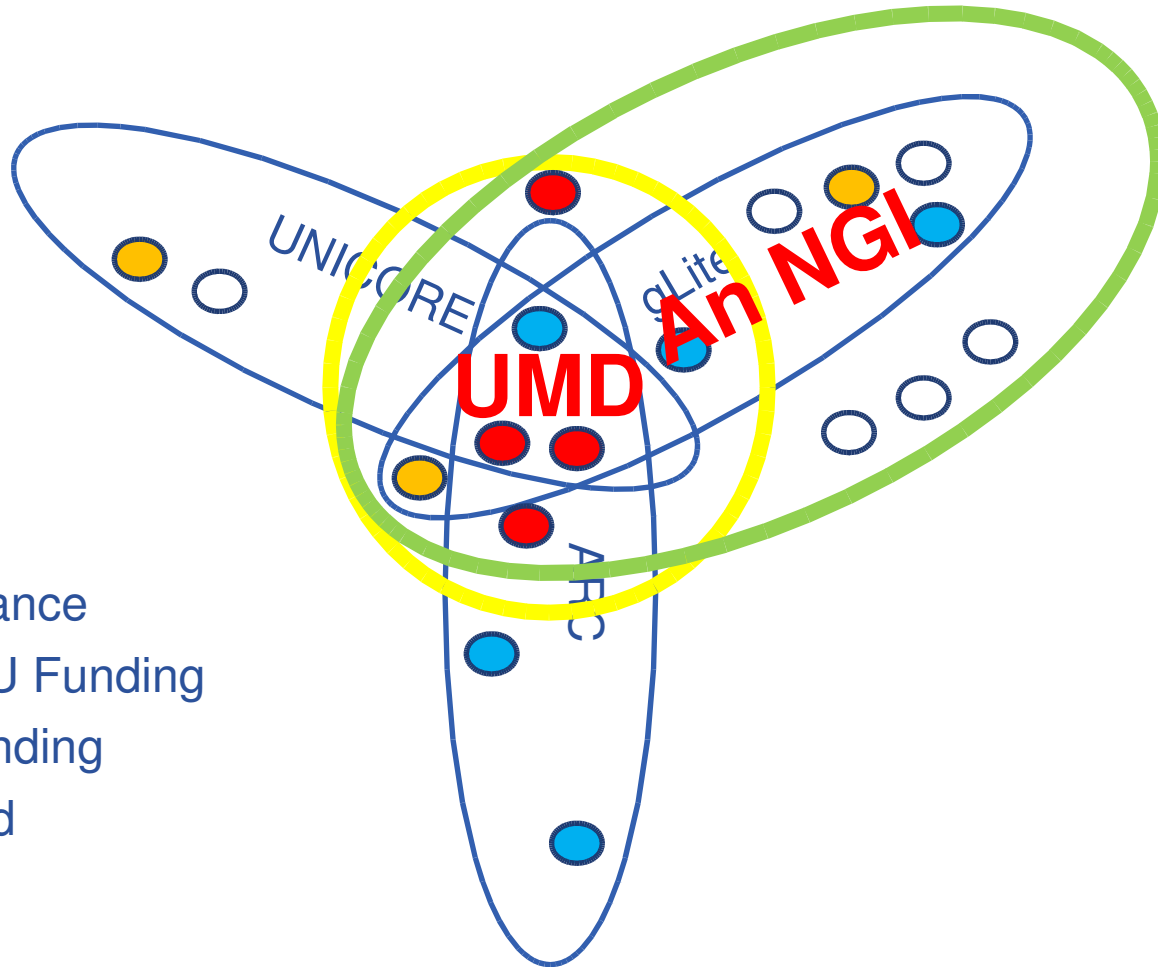
www.eu-egee.org



- The goal is to provide a usable Grid middleware distribution to EGI
- *The middleware consortia ARC, gLite and UNICORE propose to foster the convergence of their current solutions into a Unified Middleware Distribution (UMD), similar to what the Virtual Data Toolkit (VDT) is in the US for the Open Science Grid (OSG)*
- ARC, gLite and UNICORE are the biggest players in the Grid middleware arena in Europe, but others exist
 - dCache, Gridway, pGrade, ...
 - UMD is open

- **Work within the scope of UMD includes:**
 - the maintenance of the currently deployed middleware in production use across Europe that supports a diverse and active application community
 - the convergence of widely deployed components from different providers to a single, ideally standardized interface, that will enable communities to utilize services from any European middleware provider
 - the rationalization of European middleware components to eliminate duplication where there are components with converged interfaces that are deployed to meet identical use cases
 - the continued development of new functionality to meet the evolving needs of its application community and to ensure that Europe continues to be recognized as a leading provider of e-infrastructure software



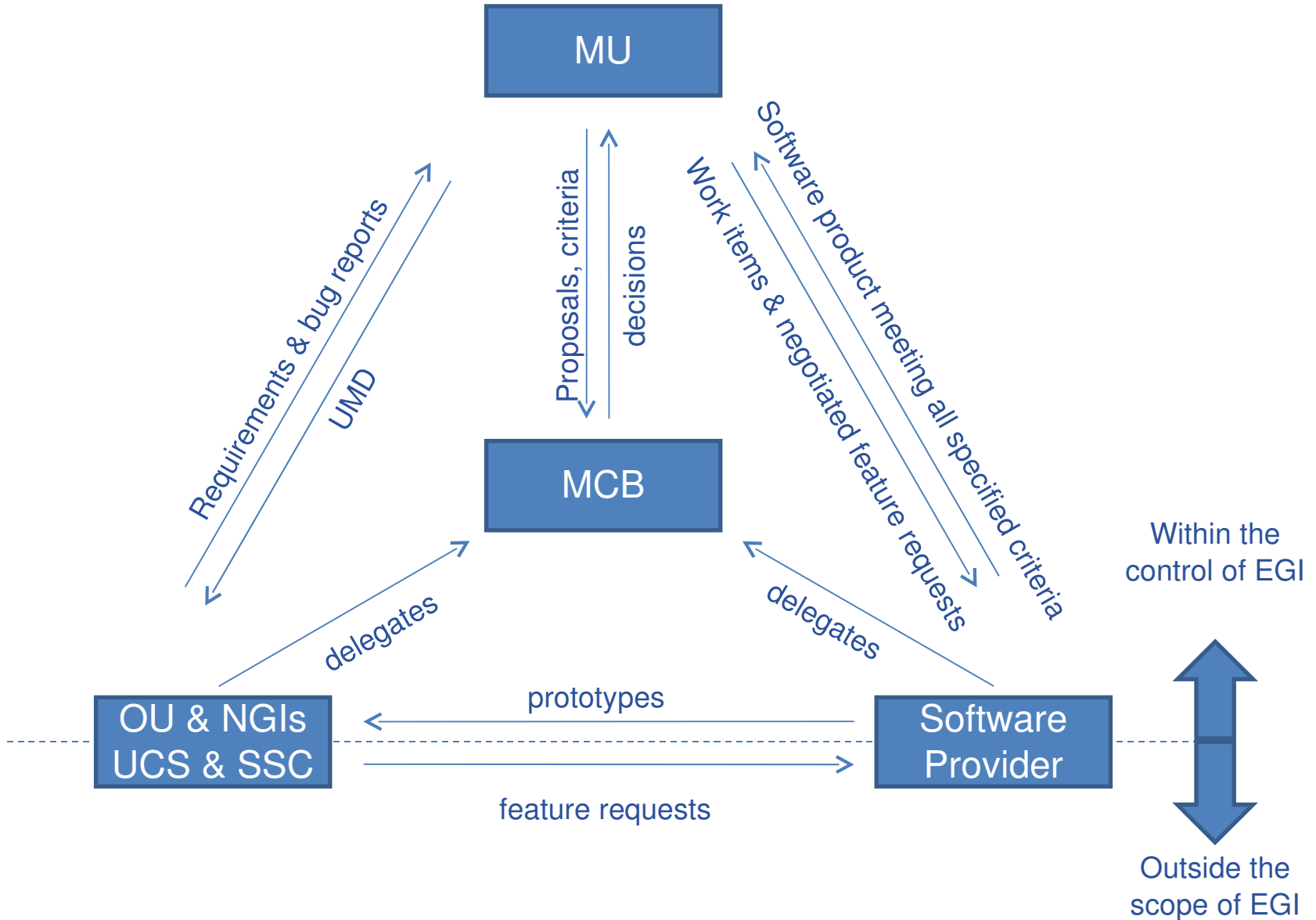


- EGI Maintenance
- Direct EGI EU Funding
- Other EU Funding
- National Fund

- <http://knowledge.eu-egi.eu/knowledge/index.php/UMD>
- **There are a series of meetings and three working groups**
 - Technical
 - Survey of existing components
 - Operations and User requirements
 - Recommendations for the future
 - Process
 - How to manage the software contributed to UMD

- **Middleware Providers**
 - e.g. Middleware consortia
- **Middleware Coordination Board**
 - Representatives from:
 - The user communities
 - The operations
 - The middleware providers
- **Middleware Unit**
 - EGI.org body
 - see next slide

MW Tasks in EGI.org	FTE
Maintain and document processes and quality criteria common to all middleware providers.	1
Provide and support tools to enable and monitor the processes (such as configuration management system, bug and task tracker, wiki).	1
Define quality and conformance criteria that UMD components need to satisfy in areas such as security, performance, scalability, functionality, usability, interoperability, adherence to standards. Verify that accepted components are certified according to the agreed process and satisfy the quality and conformance criteria, specifically targeted against security vulnerabilities.	3
Maintain a repository of certified middleware components or references thereto.	2
Follow the daily execution of the strategic plan endorsed by the MCB. Promote the EGI participation in standardisation bodies.	1
Sum of Resources in EGI.org Middleware Unit	8



- ***Generic criteria:*** interoperability, completeness, extensibility, deployability, simplicity, product documentation, platform portability, lightweight (i.e. only install the additional components that need to be installed), co-existence (i.e. require minimal, ideally none, changes to any other commonly installed software components), ...
- ***Component/element specific criteria:*** performance, stability, scalability, standards compliance, command line syntax, public API, graphical interface, ...

- *It is envisaged that the software providers will be responsible for delivering to EGI an integrated solution that includes all development, integration, testing and certification of the software element in the environment specified by the MU which will be representative of the expected environment found in production deployments.*

- **Why a Consortium?**

- Provide a long-term sustainable roadmap for the gLite software to meet the needs of its diverse user community beyond the EGEE series of projects
- Coordinate the maintenance and evolution of the software provided by the Consortium in response to requirements from its community
- Achieve interoperability with other Grid infrastructures, preferably through the adoption of established standards
- Contribute software for deployment within production infrastructures, such as the EGI UMD
- Provide community support, through mailing lists, discussion forums, help, training, documentation, etc.

- **A high-level “agreement” document has been prepared by a small group of people**
 - Antonio Candiello, Francesco Giacomini, Ales Krenek, Steven Newhouse, Markus Schulz
 - Motivation, benefits, sketch of legal and technical organization, ...
- **Presented at the EGEE-III Collaboration Board on the 3rd of March**
- **Available at <https://edms.cern.ch/document/998539>**

- **The Consortium is open (hence the name)**
- **Expected participation by organizations now contributing to the existing gLite middleware**
 - Not only for development, but also for testing, porting, training, support, etc.
- **It is not necessary to be a member in order to contribute to gLite**
 - Contributions by non-members (organizations and individuals) are encouraged, provided they are integrated in the overall software process
 - Promote the growth of a wide gLite community

- **An Activity:**
 - Is responsible for an aspect of the Consortium's technical work
 - Contributes components to the Consortium roadmap
 - Complies with the Consortium's technical requirements as defined by the TB
 - Communication and a clear process are fundamental
 - Has a leader (with deputies)
 - Similar to Product Teams in EGEE-III Y2
- **The Technical Board:**
 - Defines and maintains the Consortium roadmap
 - Broadly co-ordinates the Activities' work
 - Is composed of representatives for each Activity
 - Is led by a Technical Coordinator
 - A bit more than the EMT in EGEE-III

- **The copyright of the software and other material (e.g. documentation, training material, ...) contributed to gLite is copyrighted by the gLite Consortium**
 - EGEE has to transfer the copyright
 - Y2 goal: all the code and other material needs to have the proper copyright attribution
 - A prototype ETICS plugin is available to check copyright and license in all files
- **The software and related material are distributed under the Apache license, version 2**
 - <http://apache.org/licenses/LICENSE-2.0>
 - Adequate for commercial exploitation

- **The gLite roadmap has to be compatible with UMD**
- **A set of recommendations (wishes, dreams, ...) for UMD has been collected by the UMD User and Operations WG**
- **Time for ideas to address those recommendations, to be included in the UMD and possibly other proposals**
- **Probably a mini-workshop in the coming weeks**

- **Prepare for EGI already in year II of EGEE-III**
 - Re-organization
 - Clusters of competence → product teams
 - *Responsible for delivering working certified deployable software*
 - *Obviously this includes testing, certification and deployment*
 - *But also packaging, integration into the **node type** and documentation*
 - Some certification remains central – for the moment!
 - *NB: Underfunded SA3 resources at CERN*
 - Manage Workplan
 - Integrate core cleanup tasks (documentation, error codes, ...)
 - Group work items into releases and monitor progress of releases

Product Team	Software Components	Allocated Node Type(s)
Authorization	Authz Service, Shibboleth interoperability	PAP, PDP
VO Management	VOMS, VOMSAdmin	VOMS
Security Infrastructure	Delegation Framework, Trustmanager, Util-Java, Hydra, DICOM, myProxy Integration, LCAS/LCMAPS, gLExec, SCAS, Gridsite	Hydra
Information Systems	BDII, GLUE Schema	BDII
Compute Element	CREAM, CEMon, BLAH	CREAM
Job Management	WMS	WMS
Logging & Bookkeeping	Proxy and attribute certificate renewal, Logging & Bookkeeping, gSOAP-plugin	LB
Data Management	CGSI_gSOAP, DPM, FTS, GFAL/lcg_util, LFC	FTS, LFC, DPM

- **Responsible for delivering working software products**
 - development vs. integration vs. documentation vs. testing vs. certification
 - Fixed resources: delay releases, lower quality, reduce scope
 - Lower quality: not an option with refunding coming up!
 - Delay release: what are the downstream consequences?
 - Reduce scope: what is really needed?
- **Your software is a component in an infrastructure**
 - Change the logging format: break operations
 - Change an interface (WSDL, API, CLI): break applications
 - Change a library: force unexpected upgrades on partners
- **Really think about backwards compatibility**
 - Follow an agreed policy major & minor releases for:
 - Software Component (i.e. RPM)
 - Software Element (i.e. Node type)

- **Define ‘tasks’ to reflect the structure & dependencies**
 - Software Element (Node type) releases
 - Software Component (Library) releases
 - Work items that go into a software component
- **Cross team dependencies**
 - If your software element/component depends on another release then take a dependency on it
 - Allows teams to understand the impact of their delays/changes
 - All teams will use a base infrastructure to do their releases
 - ETICS configuration
 - Be transparent about dependencies and the impact of delays
- **This plan will form a ‘contract’ with the customer**
 - A snapshot of the plan will be in MJRA1.3.2

- **Clean up error codes & messages in all components**
 - Provide backwards compatibility
- **Dependencies between different software components**
 - Reduce dependencies between product teams
- **Review and document all public APIs and CLIs.**
- **Implement project priorities: portability, IPv6, ...**
- **Separate client and server binary deployment packages**
- **Correct copyright notices & licenses**
- **Generate source distributions from the ETICS builds**
 - SA3 will verify that the source distributions build binaries
- **What changes are NEEDED before the end of EGEE?**

- **Issues found in production use, certification or testing**
 - Triage aggressively... Is it really critical? Or just nice to have?
- **Issues defined by the TMB**
 - These are not done lightly... and are needed for project goals
- **Issues in your own roadmaps**
 - Is the item still valid? Is it needed? Is it critical? Can it wait?
- **Items must have regular review**
 - Daily within the product team?
 - Several times a week by the gLite consortium?
 - The role of the EMT?
 - Regular review by the 'customer' EGI MU (i.e. SA3)
 - Monitoring by the TMB (eventually the EGI MCB)
 - Metric: Predictable high quality releases

- **Migration out of CVS needed soon**
 - **Official CVS support at CERN ends this year**
 - unofficially?
 - **SVN or something better?**
- **Ticketing**
 - **Savannah or something better?**
- **Configuration management**
 - **ETICS? If not, what?**
- **Documentation**
 - **What about a common solution?**
- **...**

Experiences with the ETICS support for testing in vomms

Vincenzo Ciaschini

JRA1 All-Hands

- **Experimentation by Akos and Vincenzo**
- **Why DVCS?**
 - **Browsing history is fast (local)**
 - **Easy to branch, easy to merge**
 - **Developer/porter/integrator can have local branches**
 - Without commit right to central repository
 - Experimental ideas
 - Long, independent work
 - **Local branches**
 - Local history, revision control, commit comments
 - Can be updated from upstream HEAD
 - Can be sent to upstream to inclusion

```
$ cvs co -d LCG-DM-cvs LCG-DM  
# about 20 seconds
```

```
$ du -sh LCG-DM-cvs # current HEAD  
14M LCG-DM-cvs
```

```
$ git clone git://lxtank02.cern.ch/LCG-DM  
# about 11 seconds
```

```
$ du -sh LCG-DM/.git # all history back to  
1999  
4.6M LCG-DM/.git
```