



Enabling Grids for E-science

EGEE-II Middleware

Erwin Laure

EGEE-I Deputy Middleware Manager

www.eu-egee.org



Information Society



- **This presentation is not about JRA1**
- **We focus on middleware tasks that could be done within EGEE-II**
 - Not everything needs to be done within EGEE-II
 - partner projects could collaborate
 - Combines tasks currently being done by JRA1, JRA3, JRA4, and partially SA1
 - Adds new tasks currently not covered by EGEE-I
 - This was our understanding of the request from the editorial board
- **No attempt to match tasks to activities**
 - Tasks can be done in different activities or even projects
 - Not everything should be in one activity!

- **Consolidation of the current middleware stack (gLite, LCG-2)**
 - Reinforcement of
 - Quality
 - *Quality assurance*
 - *Integration*
 - *Testing*
 - Security
 - *Covering site and application aspects*
 - Continue moving to a Service Oriented Architecture and basic middleware infrastructure
 - Better enabling applications and other projects to leverage the middleware infrastructure for building higher level services
 - Reinforce collaboration with operations and applications

- **International Collaborations**
 - Continue close collaboration with US partners (in particular Condor and Globus)
 - Interoperability in particular with OSG
 - Establish better collaboration with other projects (e.g. NDGF aka NorduGrid, NAREGI, UNICORE, ...)
 - Reinforce standardization involvement

- **EGEE could be involved in 5 major middleware tasks:**
- **Integration**
- **Testing**
- **Grid foundation middleware**
- **Grid services**
- **Standardization**

- **Main integration tasks include:**
 - Tooling
 - Code repository, versioning, building
 - Management of distributed development testbed
 - Packaging, distribution
 - Integration process
 - Common infrastructure for installation, instrumentation, configuration (service management)
 - Integration with selected deployment tools (e.g. Quattor)
 - Documentation (in particular coherent installation guides with walkthroughs and deployment scenarios).
 - Release management
 - Releasing of internal and public version of the Grid middleware
 - Quality assurance
 - Definition of procedures
 - Note that most of QA work needs to be done outside middleware activities (cf. issues discussion later)

- **Need ONE core team at ONE place**
 - Responsible for releasing middleware for the infrastructure
 - No separation of gLite and LCG releases
 - *Only ONE codebase – only ONE release process (may be internal or public)*
 - Customization should be responsibility of ROCs
 - Combine current JRA1 and SA1 efforts
- **Needs to be independent**
- **Synergies with related projects (ETICS) possible**
 - Mainly on the infrastructure, probably not people
- **ALL developments inside EGEE need to follow the SAME procedures**

- **Main testing tasks include:**
 - Installation and deployment testing
 - Functionality and regression testing
 - Certification and validation testing
 - Support for application testing and integration as well as operations
- **Requires a set of distributed testbeds for different testing purposes**
 - Testbed management needs dedicated resources
 - Needs to simulate *real* sites as closely as possible
 - An automated distributed testing framework/infrastructure is essential and needs to be ready before project start
- **Requires a SINGLE, INDEPENDENT team (can be distributed) in ONE activity**
 - combining the current JRA1 and SA1 efforts
- **Application testing and integration needs to be separated**
- **Synergies with related projects (ETICS) possible**
 - Mainly on the infrastructure, probably not people

- **The “middleware infrastructure”**
 - Provide the basis for higher level services; higher level services will leverage the grid foundation middleware
- **Requires modular, well documented components with well defined interfaces**
 - Must avoid vertically integrated solutions
- **Strong requirements on accounting and auditing**
- **Strong requirements on dependability and interoperability**
 - Requires strong interactions with related projects
- **Focus is on maintenance and consolidation of current infrastructure**
- **Need functional description of components before committing manpower**

- **Grid foundation services are:**
 - Accounting
 - Computing Element
 - Information and Monitoring systems
 - Network monitoring and reservation
 - Security model and infrastructure (including auditing)
 - Storage Element
- **Basically covers all services sites need to provide. Higher level services may leverage and build upon these services**
- **Service management and deployment systems need to be developed/maintained together with Integration**
- **EGEE as an infrastructure project should provide the Grid foundation middleware**

- In the following, we list the Grid foundation services in more detail giving examples of existing technologies
 - The exact technologies to be covered in EGEE-II need to be worked out
 - **SINGLE set of services needs to be defined before project start**
- **Security**
 - VO management (VOMS and VOMS-admin)
 - Authorization (WSS, LCAS, LCMAPS, authorization framework, su-exec)
 - Auditing (EDG/EGEE job repository)
 - Policy definition and enforcement (G-Pbox, GAAAPI)
 - Dynamic connectivity service
 - Delegation service
 - Encrypted storage
- **CE**
 - Head node monitoring and management (GRAM, Condor-C)
 - VO scheduler (Condor-C, others)
 - LRMS abstraction (blaphp)
 - CREAM

- **SE**
 - SRM (Castor, dCache, DPM, NeST)
 - Local file access (gLite-I/O, xroot, nfs4)
 - Data transfer (gLite FTS, Condor Stork, Globus RFT)
 - Local catalogs – consistency service (FiReMan, LFC)
- **Network monitoring and reservation**
 - Monitoring tools and framework (EGEE-I JRA4 NPM)
 - Network reservation system (EGEE-I JRA4 BAR)
- **Accounting**
 - Local accounting sensors and caches (APEL, DGAS giandua, DGAS HLR)
 - Grid-wide accounting repository (APEL, DGAS HLR)
- **Information and Monitoring Systems**
 - Information providers and local caches (provider scripts, R-GMA, MonaLisa, ...)
 - Information registry and caches (R-GMA, MonaLisa, GridICE, ...)
 - Service discovery (R-GMA, UDDI,...)

- **Higher level services leveraging the Grid foundation middleware**
- **Sometimes quite application specific**
 - Probably different flavors of services depending on application needs
 - Different applications will require/use a different set of services
- **Development can be done in different places:**
 - EGEE (middleware and/or application activities)
 - Applications outside EGEE
 - Other projects
- **EGEE developments need functional description of components before committing manpower**

- **Grid Services include:**
 - Workload management systems
 - Logging & bookkeeping services
 - Replication services including catalogs
 - Visualization services
 - Workflow services
 - Grid economies
 - Advanced reservation systems
 - Metadata catalogs
 - ...

- **In the following, we list some of the Grid services in more detail giving examples of existing technologies**
 - The exact technologies to be covered in EGEE-II need to be worked out
- **Workload Management**
 - These are higher level services providing a Grid scheduler like functionality
 - LCG/gLite WMS
 - AliEn task queue and optimizers
 - LHCb DIRAC system
 - UNICORE broker
 - Etc.
- **Logging and bookkeeping, Job Provenance**
 - These are services keeping track on activities performed at the Grid level.
 - The gLite L&B and Job Provenance service
- **Replica Management**
 - These are services to reliably schedule data movement and catalog updates.
 - LCG replica management system (lcg-utils)
 - gLite File Placement Service and Data Scheduler
 - SDSC Storage Resource Broker (SRB)
 - Etc.

- **Monitoring visualization**
 - This is on the borderline between Grid foundation and Grid services
 - MonaLisa
 - Grid-ICE
 - Network Diagnostics
 - Etc.
- **Workflows**
 - Workflow services typically provide an application specific abstraction of the workflow specific to an application; they may include advance reservation systems.
 - LHC experiments frameworks
 - GridLab project's Triana system
 - Mygrid Taverna workbench
 - Etc.
- **Grid Economies**
 - Extend workflow and workload mgmt systems with economy based scheduling (e.g. based on the gLite DGAS/HLR system).
- **Advanced Reservation and Co-scheduling systems**
 - Manage reservations on reservable resources (e.g. SRM, CE, network)
 - gLite Agreement service
 - Etc.
- **Metadata catalogs**
 - Manage application specific metadata
 - HEP metadata catalogs
 - ARDA metadata catalog

- **Main focus remains on providing dependable production-quality Grid middleware**
- **This is encouraged by**
 - Distinction between Grid foundation MW (middleware infrastructure) and Grid services
 - Grid foundation needs to be dependable and interoperable
 - Better facilitate building of higher level services by applications and other projects
 - Common core integration and testing teams
 - Avoid duplication of work
 - Improve communication
 - Focus on security
 - Better collaboration with applications and operations

- **SA1**
 - Avoid duplication of integration and testing efforts
 - Maintenance of deployed software must not be separated from consolidation of Grid foundation
- **NA4**
 - Close collaboration with (pilot?) applications is needed
 - Developers commitment needed
 - We need a way to come up with a commonly agreed workplan
 - Boundaries between application tasks and middleware tasks must be clear and agreed upon early
 - Several models in EDG, EGEE-1 and LCG:
 - Task Forces
 - PTF
 - ARDA
 - Baseline working group
 - EIS (Experiment Integration Support)

- **NA3**
 - Help with training material and holding courses – effort needs to be allocated
- **JRA2**
 - Quality assurance need to be independent from middleware. Must not rely on middleware people being honest!
 - Needs to be appropriately staffed
- **JRA3 and JRA4**
 - Middleware related tasks are now integrated in what has been discussed before
- **NA2/5**
 - EU Concertation effort needs to be planned; the same holds for dissemination efforts

- **As discussed before cross-activity coordination needs to take place in the following areas**
- **Application support**
- **Operational issues**
- **Quality assurance**
- **Training**
- **Standardization/Concertation**
- **Security coordination**

- **Priority process**
 - How to come up with a common plan between middleware, operations, and applications?
- **Collaboration with other (additional) projects**
 - Which projects? How will that be coordinated?
- **Quality assurance**
 - Need to be independent from middleware activity and appropriately staffed
- **Standardization**
 - Standardization work is very time (and money) consuming
 - Need to have a project-wide agreement early where to focus and this agreement needs to be endorsed by the affected partners
 - Work on interoperability is a first step towards standardization
 - This is the short term part
 - Standardization work is more long term and tedious!
 - *No stable standards yet – don't really see them on the horizon*
 - This does not mean to adopt early standards in the middleware infrastructure
 - No proof of concept implementations
 - Requested from reviewers

- **Organization**

- Depends on:
 - Selection of tasks
 - Which tasks will be allocated to which activities
 - Who will be the partners
 - Staffing levels

- Need to be confined to a “small” number of (geographically bound) centers
- Staff commitment needs to be at an appropriate level (>70%)
 - Administrative overheads account for a significant part of the effort

- Retain (and enlarge) distributed development testbed

- Retain international Design Team
- Retain Middleware Security Group
- Retain the eXtended Integration Team