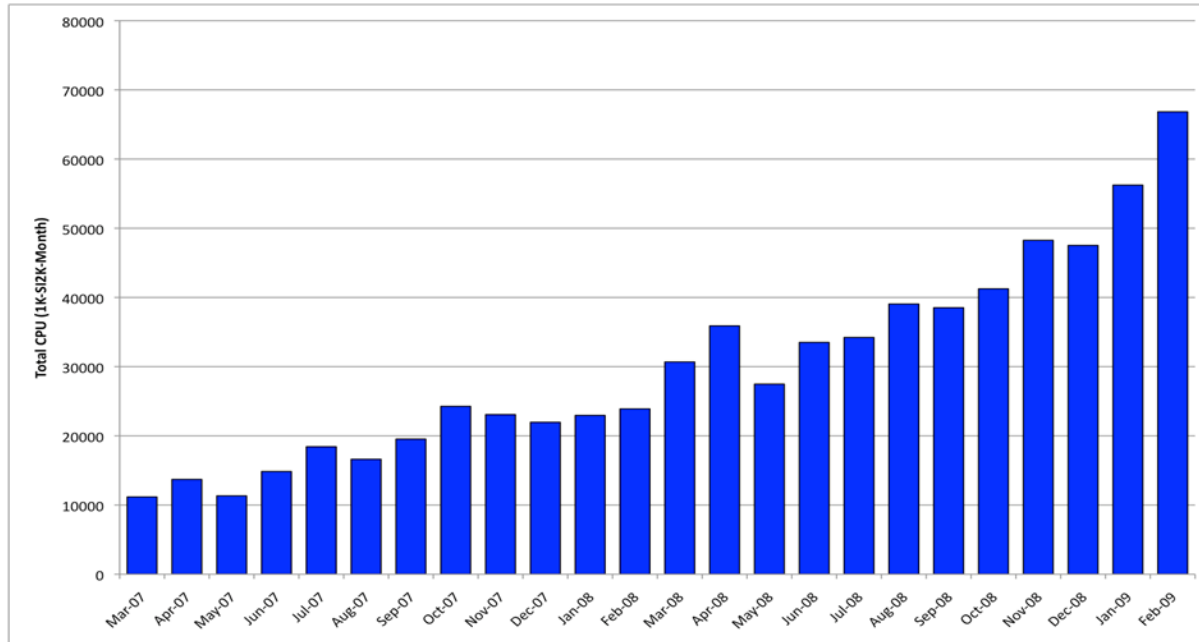


Moving towards EGI - JRA1 & SA3 All Hands

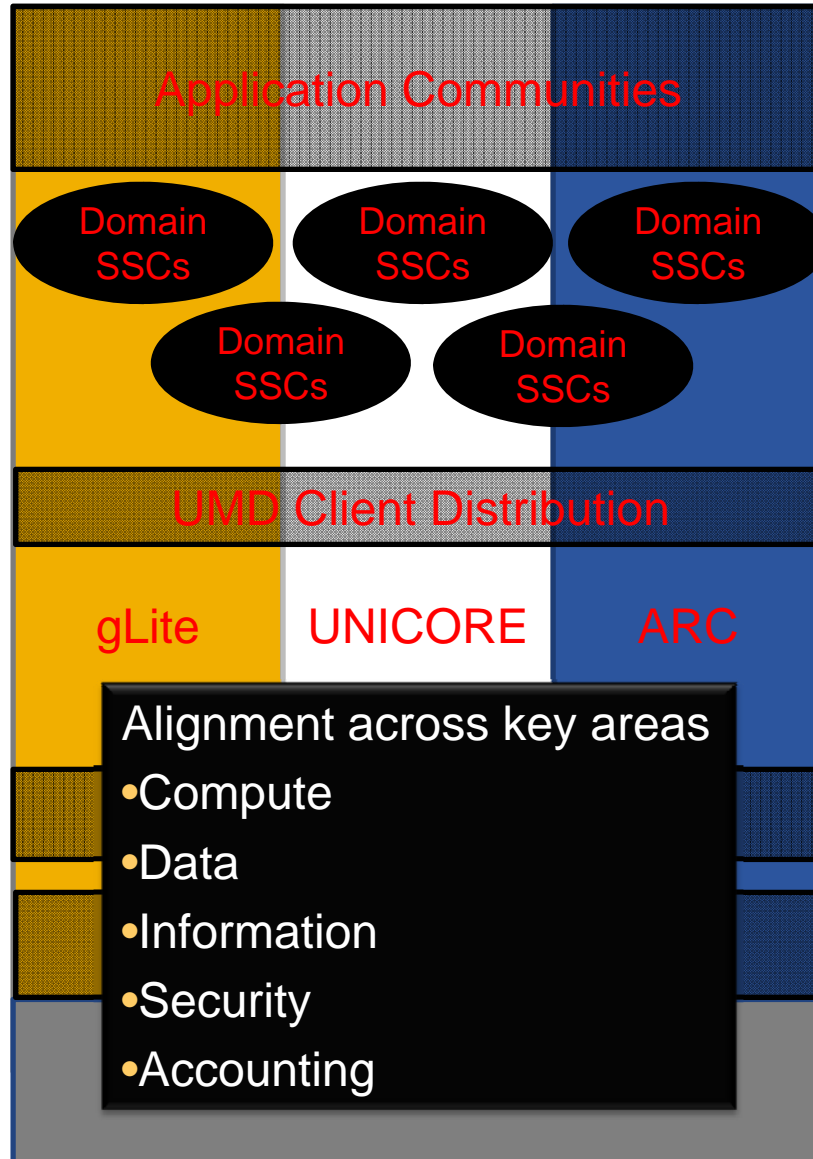
Steven Newhouse
Technical Director

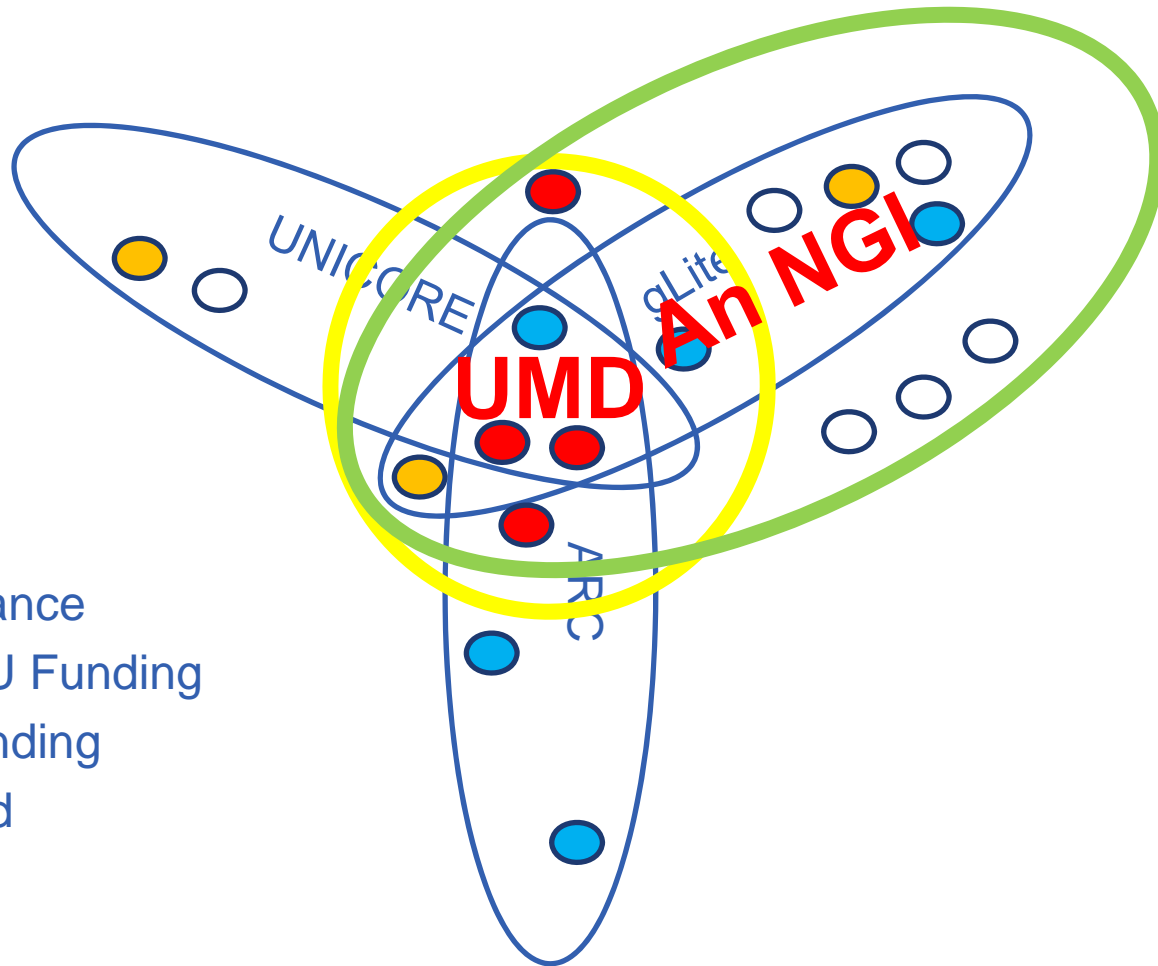
- **Approaching 10 years investment in ‘e-infrastructure’**
 - World leading development activities
 - Integration, testing & certification of production strength software
 - Deployment and operation on a worldwide 24/7/365 basis
- **User Community**
 - Number of users and their use of the infrastructure GROWS



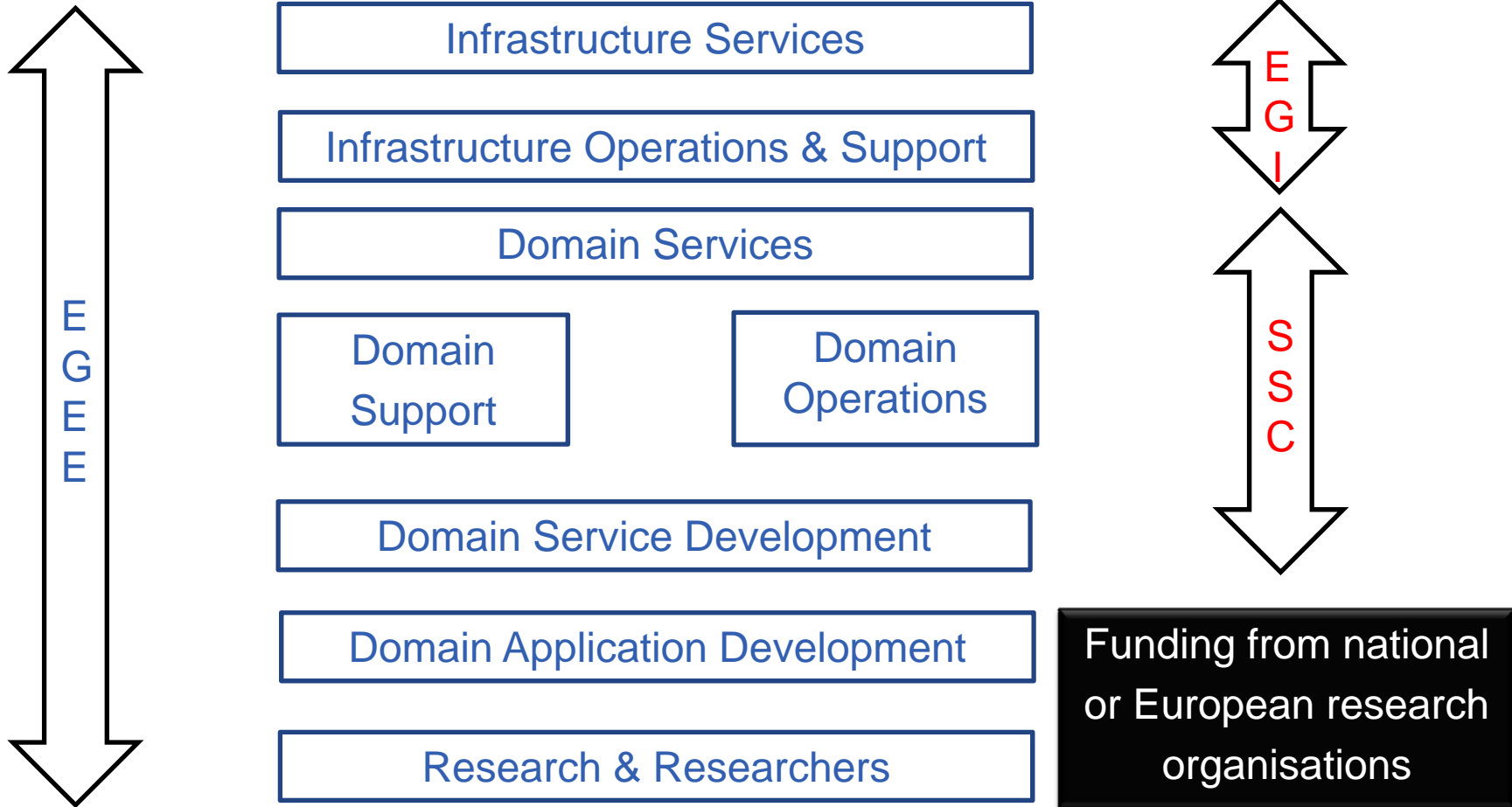
- **EGI: Moving towards sustainability**
 - What does that actually mean?
- **It's for *the* community not *one* community**
 - Driven by the NGIs to contribute resources for their researchers
 - Provide an integrated distributed electronic infrastructure
 - High Throughput Computing
 - High Performance Computing
 - Data Resources & Data Sources (Instruments)
- **Operations**
 - Support the integration into a single European Infrastructure
- **User Community**
 - Provide the bridge to get communities onto the e-infrastructure
 - i.e. training, domain specific operations, domain specific software

- **An integrated e-infrastructure → integrated software**
 - gLite
 - ARC
 - UNICORE
- **EGI will focus on maintenance & harmonisation**
 - Incremental conservative improvements
 - Development can take place independently in parallel
 - Move developments across to production IF proven & required
 - Research takes place elsewhere
- **Professional middleware for production use**
 - Take note... for this software community to have a future!





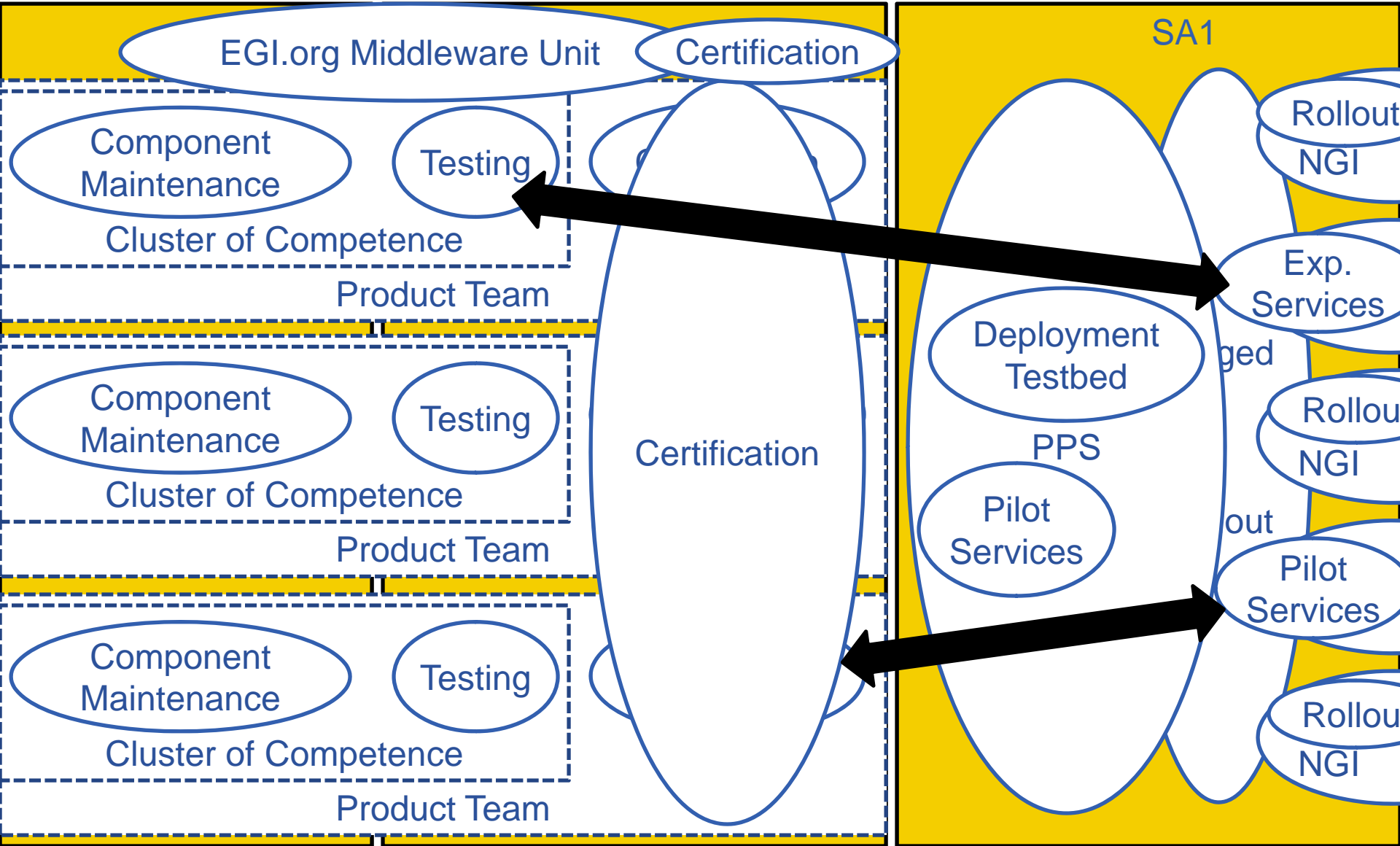
- EGI Maintenance
- Direct EGI EU Funding
- Other EU Funding
- National Fund



- **EGI provides the bridge between national activities**
 - Small central organisation
- **EGI (MU) owns the 'UMD' brand for the community...**
 - Has to work with user, operations & middleware to define it
 - Work with middleware consortia to deliver it
 - Minimal resources ($\rightarrow 0$) to integrate and verify releases
- **EGEE in Y2 has to transition to the EGI structures**
 - Start now... find out problems now... not later

- **EGI starting on 1st May 2010 is unlikely**
 - There *may* be scope to backdate start date
- **Extend EGEE resources for an additional 3 months**
 - Informed by the results from the CB questionnaire
 - Informed by the country reports & the activity view
 - Maintaining the critical tasks within the infrastructure
- **What we are not doing...**
 - Moving resources between partners
 - Look to move tasks not people
 - Stopping activities/tasks
 - May prioritise some tasks over others to provide extension
 - Formally moving work between activities
 - Different reimbursement rates between NA/SA/JRA
- **Do the most with the resources we will have!**

An Overview of JRA1/SA1/SA3



This is what we have now!

- **Identify and establish:**
 - A gLite consortium bringing together SA3 & JRA1 partners
 - A EGI.org Middleware Unit (effectively SA3 resources @ CERN)
- **Establish product teams**
 - Responsible for delivering working certified deployable software
 - **Obviously** this includes testing, certification and deployment
 - **But** also packaging, integration into the node type & documentation
 - Some certification remains central – for the moment!
 - NB: Underfunded SA3 resources at CERN being raised at PMB
- **Manage Workplan (upcoming MJRA1.3.2)**
 - Integrate core cleanup tasks (documentation, error codes, ...)
 - Group workitems into releases & monitor progress of releases
- **Reduce certification workload**
 - Group releases onto a node type

Product Name (Lead Manager & Partner)	Software Components (Partners)	Allocated Node Type(s)
Authorization (Christoph Witzig, SWITCH)	Authz Service (SWITCH, HIP, INFN, NIKHEF) Shibboleth interoperability (SWITCH)	PAP PDP
VO Management (Vincenzo Ciaschini, INFN)	VOMS (INFN) VOMSAdmin (INFN)	VOMS
Security Infrastructure Product Team (John White, HIP)	Delegation Framework (CERN, HIP, STFC) Trustmanager (HIP), Util-Java (HIP), Hydra (HIP), DICOM (HIP) myProxy Integration (HIP), LCAS/LCMAPS (NIKHEF) glExec (NIKHEF), SCAS (NIKHEF), Gridsite (STFC)	Hydra
Information Systems (Laurence Field, CERN)	BDII (CERN) GLUE Schema (CERN)	BDII
Compute Element (Massimo Sgaravatto, INFN)	CREAM (INFN) CEMon (INFN) BLAH (INFN)	CREAM
Job Management (Marco Cecchi, INFN)	WMS (INFN, ED)	WMS
Logging & Bookkeeping (Ales Krenek, CESNET)	Proxy and attribute certificate renewal (CESNET) Logging & Bookkeeping (CESNET) Gsoap-plugin (CESNET)	LB
Data Management (Ákos Frohner, CERN)	CGSI_gSOAP (CERN), DPM (CERN), FTS (CERN) GFAL /lcg_util (CERN), LFC (CERN)	FTS (various), LFC DPM (various)

Product Name (Lead Manager & Partner)	Software Components (Partners)	Allocated Node Type(s)
Integrated Clients (SA3, CERN)	Proxy Renewal (Elisa @ ???) GSI-SSH (External - TBC)	UI WN VO Box
Torque (SA3, NIKHEF)		Torque
LSF (SAn, ???)		LSF
Condor (SAn, PIC)		Condor
SGE (SAn, CESGA)		SGE
MPI (SAn, CERN)	Work with MPI-WG to find a new group to take this on.	GLITE-MPI
Other / Unassigned (SA3, CERN)	dCache (External) AMGA (Birger Koblitiz, KISTI)	

- **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

- **Removing explicit SA1 PPS deployment testbed**
 - Staged rollout on current deployment resources & NGI resources
 - Includes system testing on representative production resources
 - Exact mechanism TBC with SA1
- **Critical issues resolved by upgrading to ‘old’ release**
 - V1.0.4 – deployed in production use
 - V1.0.5 – new release with critical undiscovered issue
 - V1.0.6 – new release containing V1.0.4 code base
- **Must be able to make releases with JUST bug fixes**
 - When you release you MUST branch
 - Not acceptable for a new bug fix release to have new functionality

- **EGI presents a disruptive opportunity to the EU middleware community**
 - Will need to deliver working software on time
 - Will need to become more relevant to other e-infrastructures
 - Relationship with EU providers and elsewhere
 - Need to prove our worth
 - Focus on maintenance and development as distinct activities
- **Changes planned for Y2 will smooth transition**
 - Delivering certified software, recognising SA3@CERN issues
 - Work together within and between the product teams
 - Ensure relevance to user and operations community

Any questions?