Middleware Reengineering (JRA1)

Frédéric Hemmer, JRA1 Manager, CERN
On behalf of JRA1
EGEE 1st EU Review
9-11/02/2005

Introduction

- **This presentation highlights the organization, processes and achievements of JRA1 (Middleware Engineering and Integration)**
  - Overview
  - Services
  - Processes & Tools

Overview of the activity

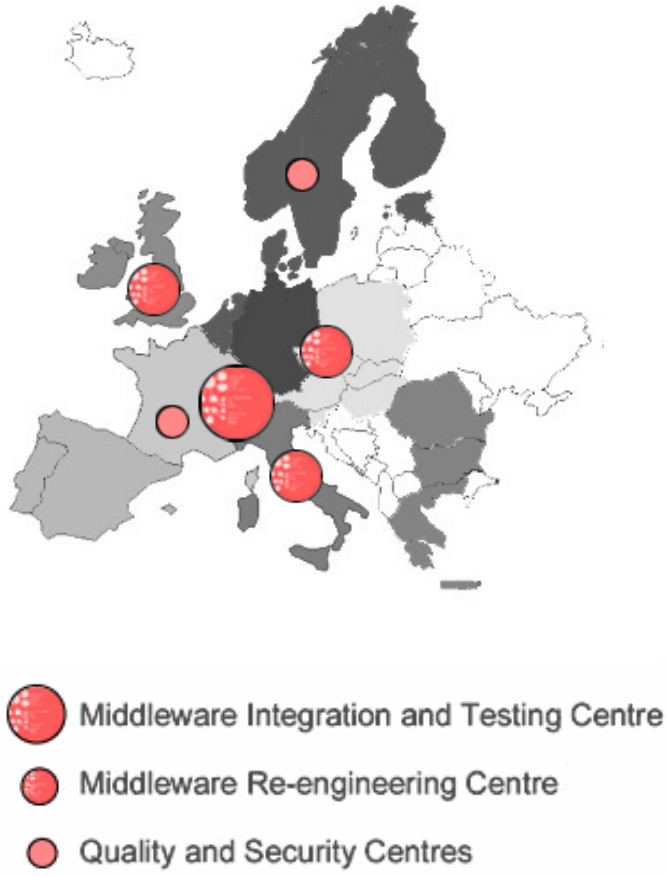
Original objectives

- **Provide robust, supportable middleware components**
  - Select, re-engineer, integrate identified Grid Services
  - Evolve towards Services Oriented Architecture
  - Adopt emerging OGSI standards*
  - Multiple platforms
- **Selection of Middleware based on requirements of**
  - The Applications (Bio & HEP)
    - In particular requirements from LCG's ARDA & HepCALII
  - The Operations
    - E.g. deployment, updates, packaging, etc..
- **Support and evolve the middleware components**
  - Evolution towards OGSI*
  - Define a re-engineering process
  - Address multiplatform, multiple implementations and interoperability issues
  - Define defect handling processes and responsibilities

*: **Challenged by the WSRF announcement on January 20, 2004. The strategy is to use plain Web Services (WS-I compliant) and review the situation when implementation(s) become available (GT4-WRSF::Lite).**

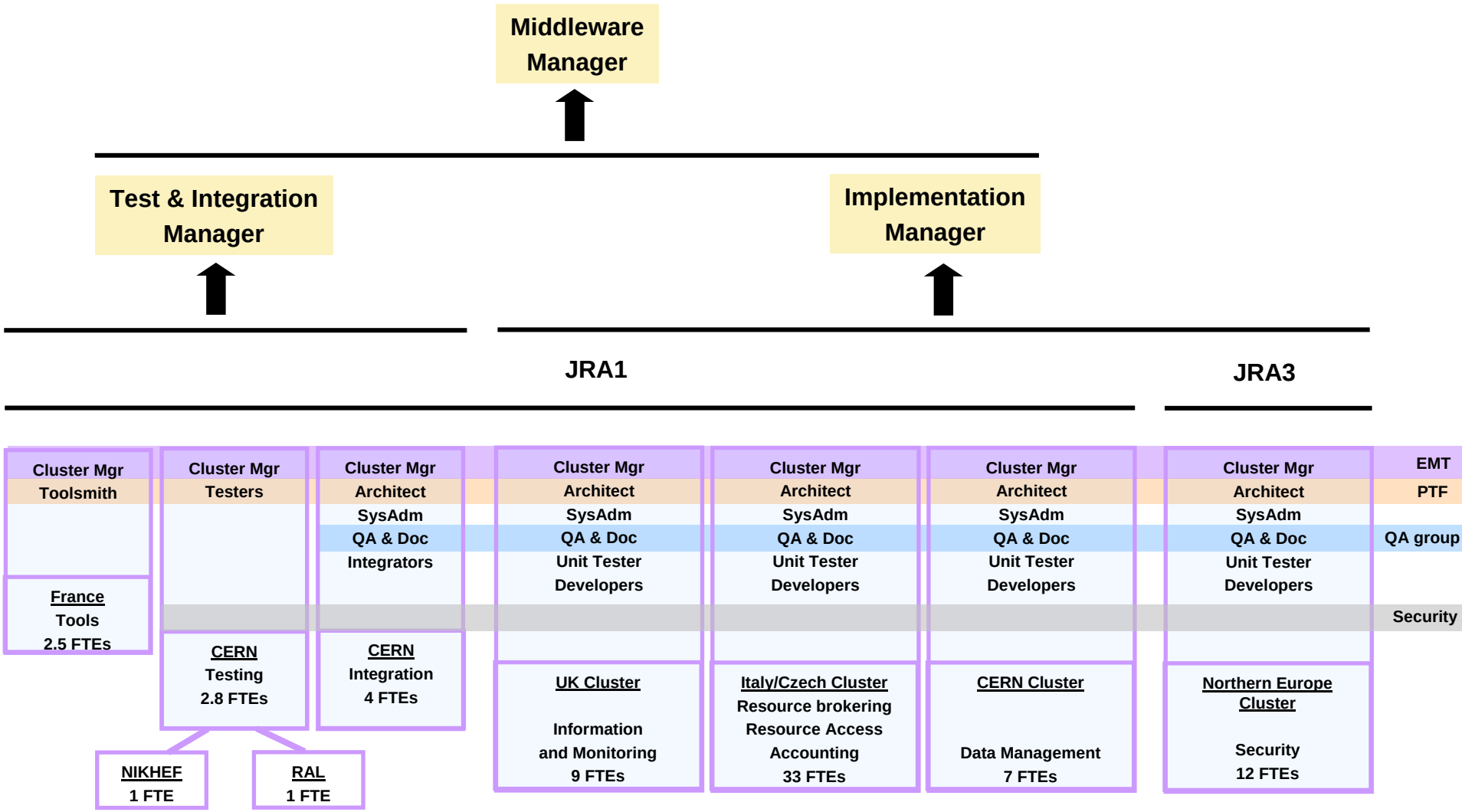
Software Clusters

- **Hardening and re-engineering of existing middleware functionality, leveraging the experience of partners**
- **Activity concentrated in few major centers and organized in "Software clusters"**
- **Key services:**
  - Data Management (CERN)
  - Information and Monitoring (UK)
  - Resource Brokering, Accounting (Italy-Czech Republic)
  - Quality Assurance (France)
  - Grid Security (Northern Europe)
  - Middleware Integration (CERN)
  - Middleware Testing (CERN)



- **Clusters collaborate with US partners**
  - University of Chicago
  - University of Southern California
  - University of Wisconsin Madison

Organization

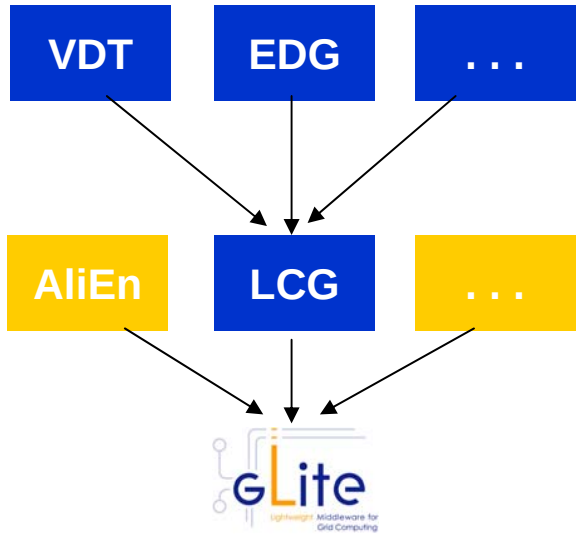


Middleware Reengineering

Services

Guiding Principles

- **Service oriented approach**
  - Allow for multiple interoperable implementations
- **Lightweight (existing) services**
  - Easily and quickly deployable
  - Use existing services where possible
    - Condor, EDG, Globus, LCG, …
- **Portable**
  - Being built on Scientific Linux and Windows
- **Security**
  - Sites and Applications
- **Performance/Scalability & Resilience/Fault Tolerance**
  - Comparable to deployed infrastructure

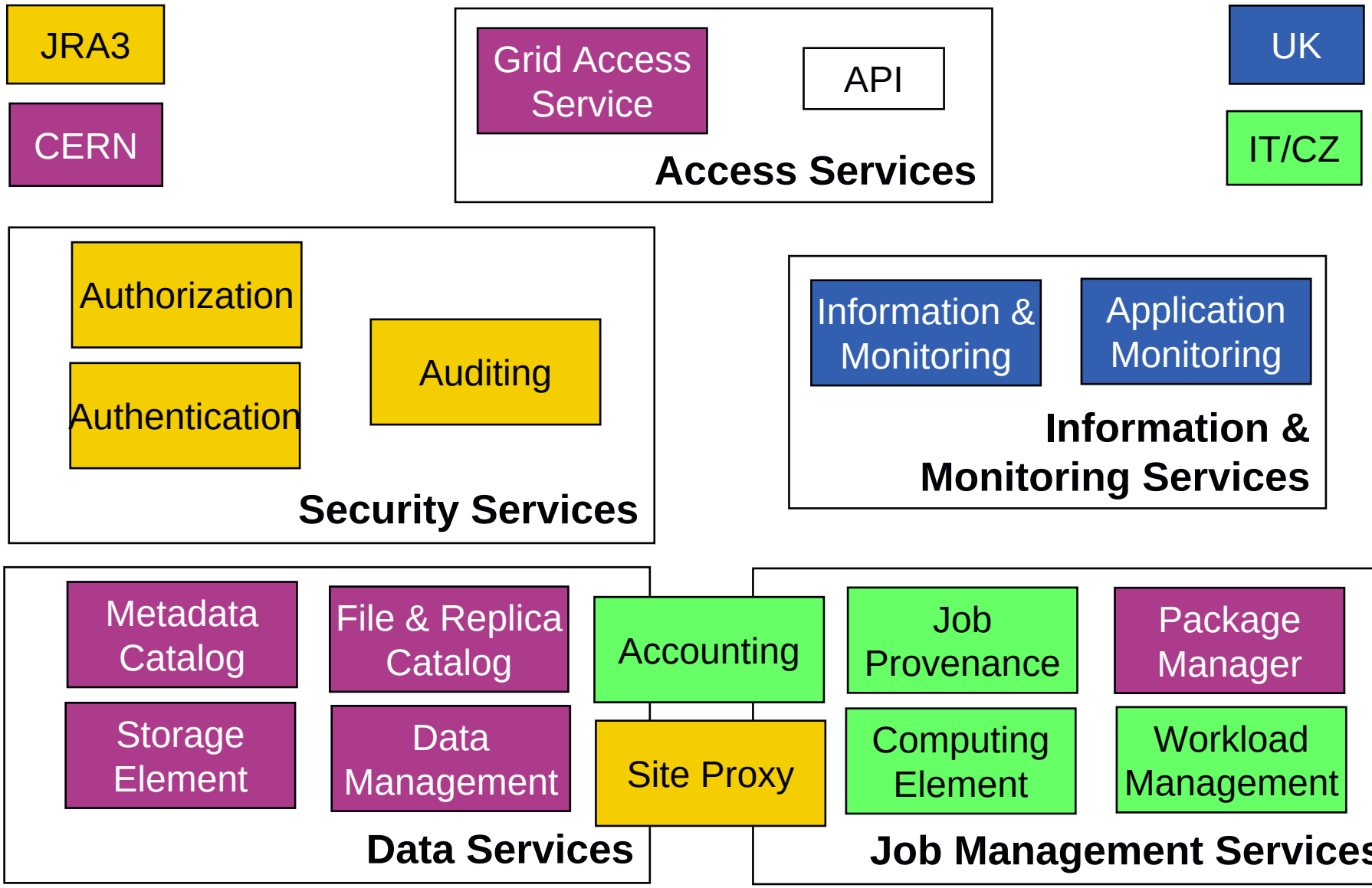


- **Co-existence with deployed infrastructure**
  - Co-existence with LCG-2 and OSG (US) are essential for the EGEE Grid services
- **Site autonomy**
  - Reduce dependence on ‘global, central’ services
- **Open source license**

Architecture & Design

- **Design team including representatives from Middleware providers (AliEn, Condor, EDG, Globus,…) including US partners produced middleware architecture and design.**
- **Takes into account input and experiences from applications, operations, and related projects**
- **DJRA1.1 – EGEE Middleware Architecture (June 2004)**
  - https://edms.cern.ch/document/476451/
- **DJRA1.2 – EGEE Middleware Design (August 2004)**
  - https://edms.cern.ch/document/487871/
- **Much feedback from within the project (operation & applications) and from related projects**
  - Being used and actively discussed by OSG, GridLab, etc. Input to various GGF groups

gLite Services and Responsible Clusters



Job Management Services

- **Efficient and reliable scheduling of computational tasks on the available infrastructure**
- **Started with LCG-2 Workload Management System (WMS)**
  - Inherited from EDG
  - Support partitioned jobs and jobs with dependencies
  - Support for different replica catalogs for data based scheduling
  - Modification of internal structure of WMS
    - Task queue: queue of pending submission requests
    - Information supermarket: repository of information on resources
    - Better reliability, better performance, better interoperability, support push and pull mode
  - *Under development*
    - Web Services interface supporting bulk submission (after V1.0)
      - *Bulk submission supported now by use of DAGs*
    - Distributed superscheduling (interaction among multiple WMSs)

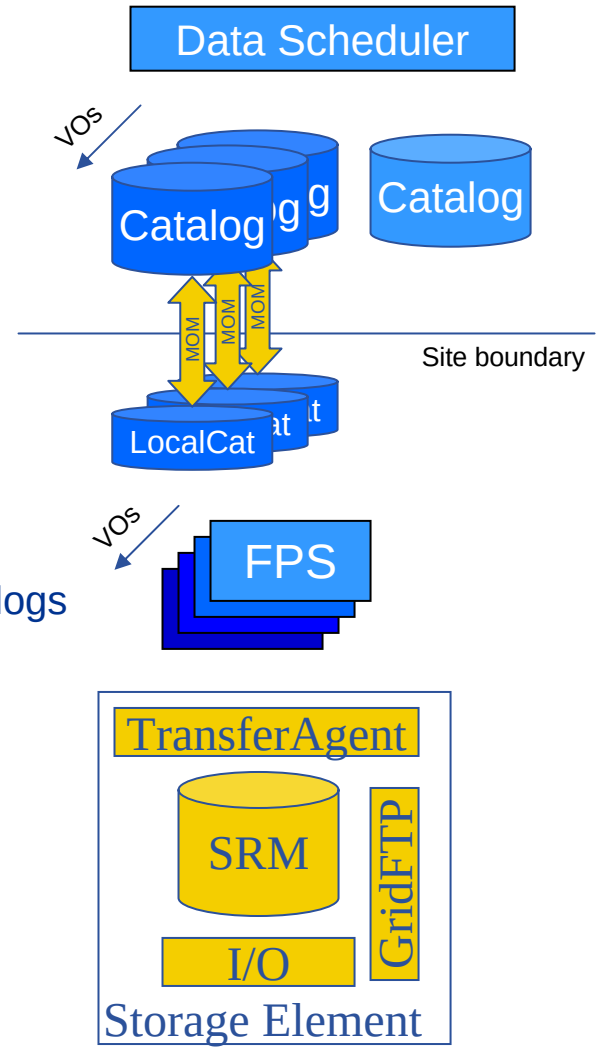
Job Management Services II

- **Computing Element (CE)**
  - Service representing a computing resource
    - CE moving towards a VO based local scheduler
    - Incorporates new technologies provided by Condor and Globus
  - Web service interfaces
    - job management requests (run, cancel, suspend, resume, …)
      - *Still under development*
    - Policy based notifications on changes of the CE (pull model)
- **Job Provenance**
  - Keeps track of definition of submitted jobs, execution conditions and job life cycle for a long time *under development*
- **Grid Accounting (DGAS)**
  - Accumulates Grid accounting information about the usage of Grid resources by users / groups (e.g. VOs) for billing and scheduling policies. *Under development*
- **VOMS**
  - Virtual Organization Membership Service
- **Advanced Reservation service *under development***
- **Assured backward compatibility (to facilitate migration from current SA1 infrastructure)**

Data Management Services

- **Efficient and reliable data storage, movement, and retrieval on the infrastructure**

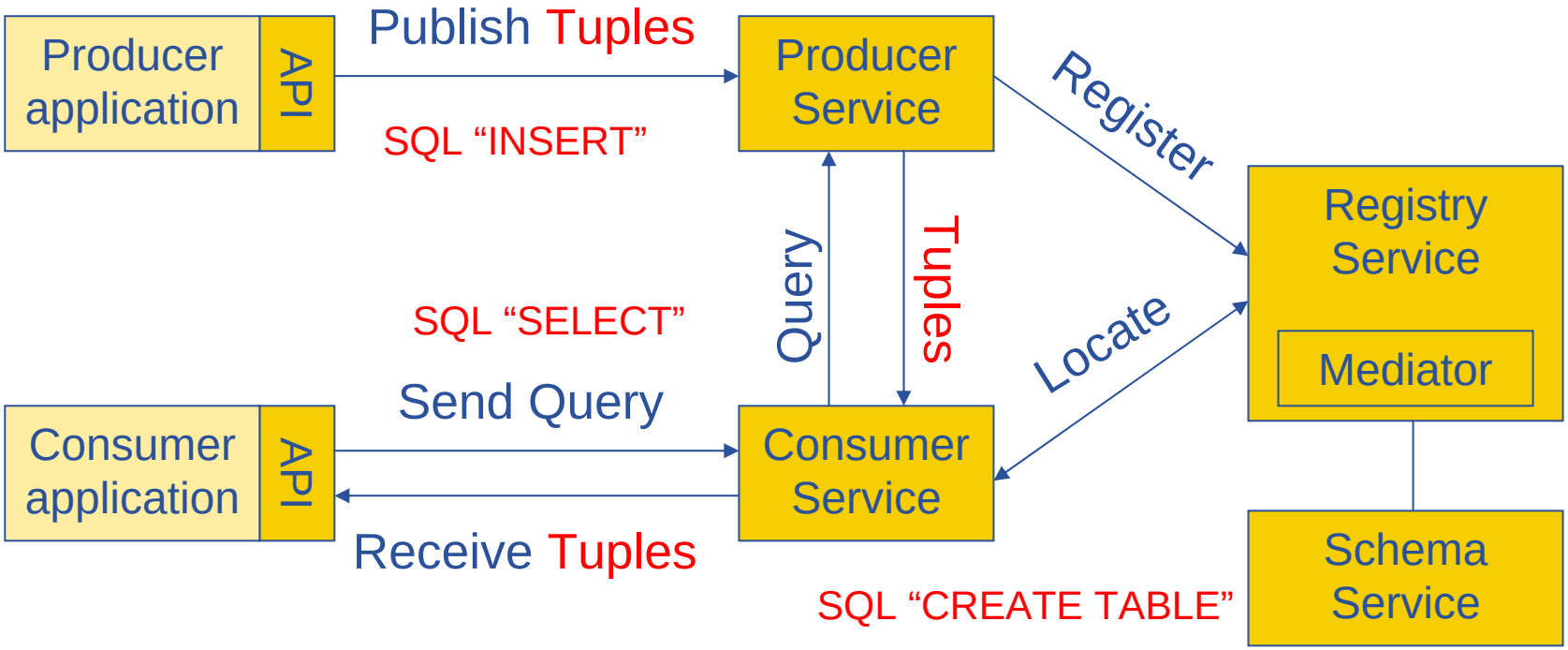
- **Storage Element**
  - Reliable file storage (SRM based storage systems)
  - Posix-like file access (gLite I/O)
  - Transfer (gridFTP)
- **File and Replica Catalog**
  - Resolves logical filenames (LFN) to physical location of files (URL understood by SRM) and storage elements
  - Hierarchical File system like view in LFN space
  - Single catalog or distributed catalog (*under development*) deployment possibilities
- **File Transfer and Placement Service**
  - Reliable file transfer and transactional interactions with catalogs
- **Data Scheduler**
  - Scheduled data transfer in the same spirit as jobs are being scheduled taking into account e.g. network characteristics (collaboration with JRA4)
  - *Under development*
- **Metadata Catalog**
  - Limited metadata can be attached to the File and Replica Catalog
  - Interface to application specific catalogs have been defined



Information and Monitoring Services

- **Efficient and reliable provision of Grid information and Grid and Application monitoring data**

- **R-GMA (Relational Grid Monitoring Architecture)**
  - Implements GGF GMA standard
  - Development started in EDG, deployed on the production infrastructure for accounting



R-GMA

- **Producer, Consumer, Registry and Schema services with supporting tools**
  - Registry replication
  - Simpler API – matching the next (WS) release
    - Provides smooth transition between old API and WS
  - coping with life on the Grid: poorly configured networks, firewalls, MySQL corruptions etc

- **Generic Service Discovery API**

- ***Under development***
  - Web Service version
  - File (as well as memory and RDBMS) based Producers
  - Native python interface
  - Fine grained authorization
  - Schema replication

Other Reengineering Activities

- **Prototypes of Grid Access Service and Package Manager implemented in the AliEn framework**

- **Grid Access Service**
  - Acts on user's behalf
  - Discovers and manages Grid services for the user

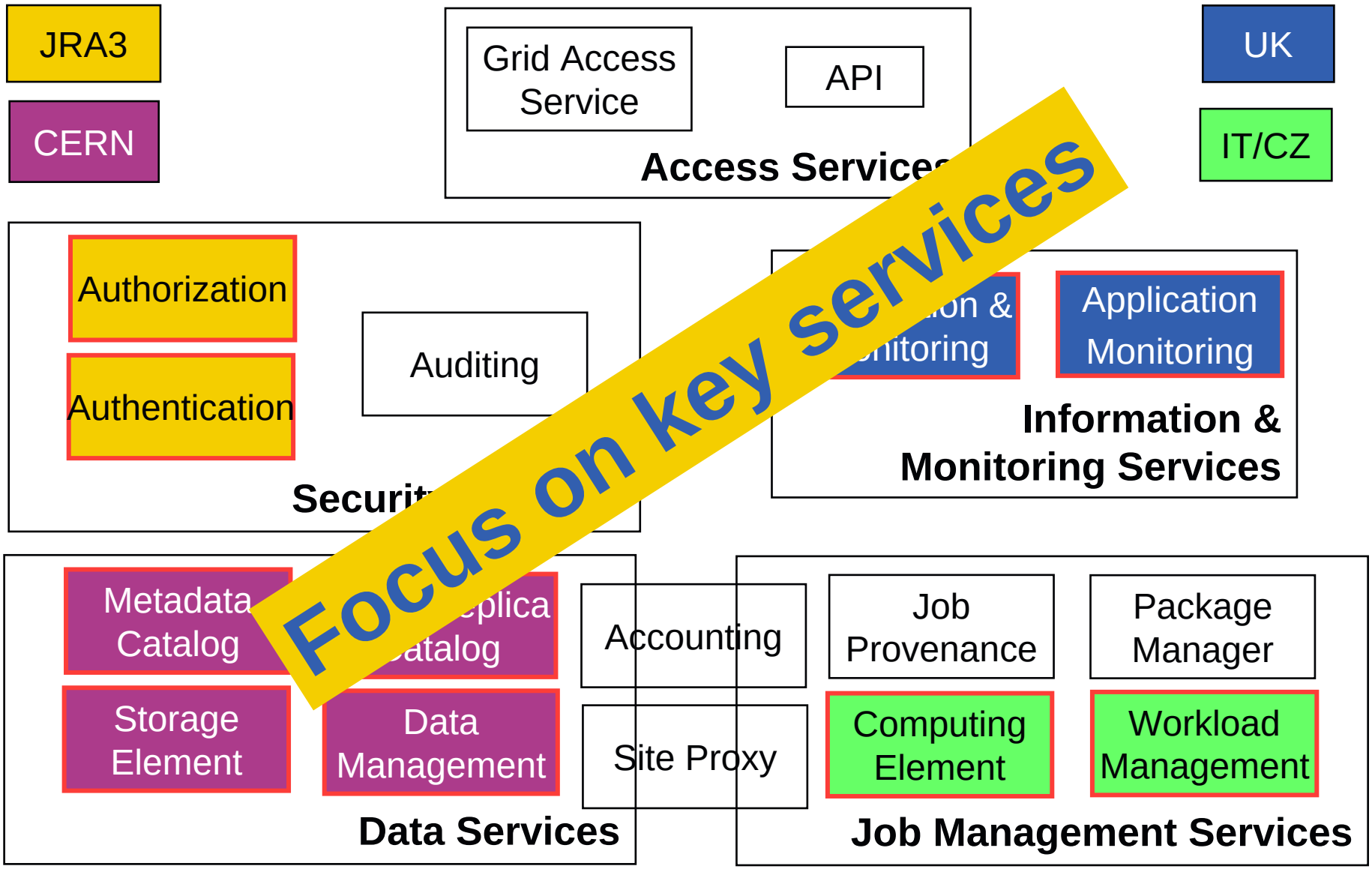
- **Package Manager**
  - Provides dynamically distribution of application software needed
  - Does not install Grid middleware

Security

- **Job Management Services**
  - WMS, LB, and CE implement authorization based o… VO, groups, and user information
- **Data Services**
  - Authorization: ACL and (Unix) p… ons
  - Fine-grained ACL on data … through gLite-IO and Catalogs
  - Catalog data itself … orized through ACLs
    - Currently s… through DNs
    - VOMS … tion being developed
- **Inf… tion Services**
  - …ine grained authorization based on VOMS certificates being implemented

Following talk by Ake Edlund will give more details

gLite Services for Release 1



gLite Services for Release 1
Software stack and origin (simplified)

- **Computing Element**
  - Gatekeeper (Globus)
  - Condor-C (Condor)
  - CE Monitor (EGEE)
  - Local batch system (PBS, LSF, Condor)
- **Workload Management**
  - WMS (EDG)
  - Logging and bookkeeping (EDG)
  - Condor-C (Condor)
- **Storage Element**
  - File Transfer/Placement (EGEE)
  - glite-I/O (AliEn)
  - GridFTP (Globus)
  - SRM: Castor (CERN), dCache (FNAL, DESY), other SRMs
- **Catalog**
  - File and Replica Catalog (EGEE)
  - Metadata Catalog (EGEE)
- **Information and Monitoring**
  - R-GMA (EDG)
- **Security**
  - VOMS (DataTAG, EDG)
  - GSI (Globus)
  - Authentication for C and Java based (web) services (EDG)

Main Differences to LCG-2

- **Workload Management System works in push and pull mode**
- **Computing Element moving towards a VO based scheduler guarding the jobs of the VO (reduces load on GRAM)**
- **Distributed and re-factored file & replica catalogs**
- **Secure catalogs (based on user DN; VOMS certificates being integrated)**
- **Scheduled data transfers**
- **SRM based storage**
- **Information Services:**
  R-GMA with improved API and registry replication
- **Prototypes of additional services**
  - Grid Access Service (GAS)
  - Package manager
  - DGAS based accounting system
  - Job provenance service
- **Move towards Web Services**

Standards

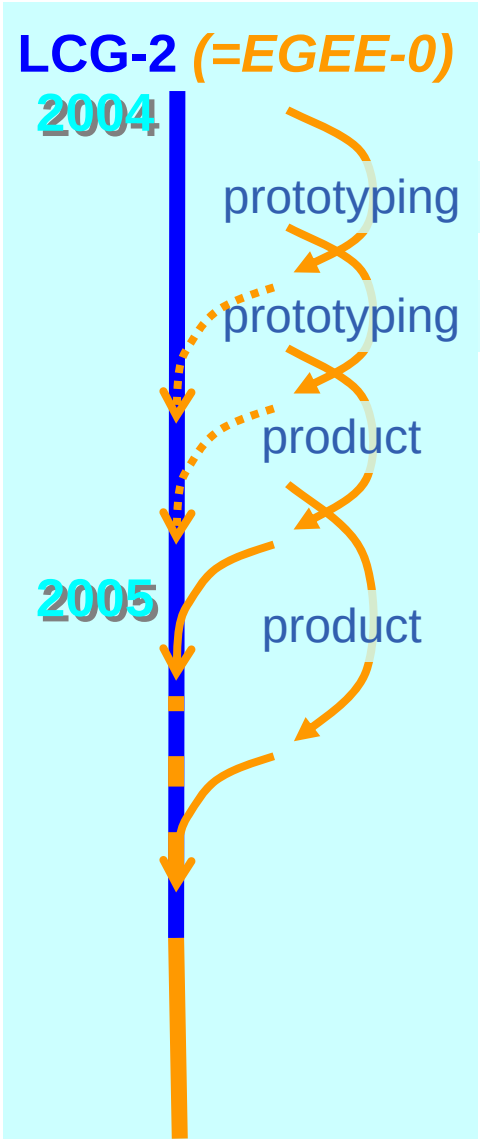
- **Web Services Fast moving area**
  - Follow WSRF and related standards but are not early adopters
  - WS-I compatibility is a target
    - Challenging to write WSDL which is WS-I compatible AND can be processed by all the tools
  - Industry strength tooling not always available
  - Trying to keep back from the bleeding edge

- **Work on standards bodies**
  - Active contributions to
    - GGF OGSA-WG
      - *GMA in OGSA*
      - *Data Design team*
    - GGF INFOD-WG
    - OASIS WS-N
    - GGF GSM-WG (SRM)
      - *Co-chairing WG*
    - Replica Registration Service
  - And following many, many others
  - Adopting mature standards is a goal

Middleware Reengineering

Processes and tools

From Development to Product



- **Fast prototyping approach allowing end users for rapid feedback**
- **Provide individual components to SA1 for deployment on the pre-production service**
  - See Markus Schulz talk on Certification and Deployment process
- **These components need to go through integration and testing**
  - To ensure they are deployable and basically work

Prototype

- **Started an early prototyping activity (was not in the original workplan) on applications request**
  - Small scale testbed
    - CERN, Univ. Wisconsin later also INFN and NIKHEF
    - Started with only 3 WNs, now 35 WNs (further expansion planned)
  - Many ideas of architecture & design inspired by AliEn system hence
  - Started from AliEn with additional components successively added
  - Get user feedback on service semantics and interfaces
    - HEP ARDA project
    - Biomed applications
  - ~70 users registered
  - Second VO has been set up (core services at Wisconsin)
    - Uses Globus RLS

Software Process

- **JRA1 Software Process is based on an iterative method**
- **It comprises two main 12-month development cycles divided in shorter *development-integration-test-release* cycles lasting 1 to 4 weeks**
- **The two main cycles start with full Architecture and Design phases, but the architecture and design are periodically reviewed and verified.**
- **The process is fully documented in a number of standard documents:**
  - Software Configuration Management Plan
  - Test Plan
  - Quality Assurance Plan
  - Developer's Guide

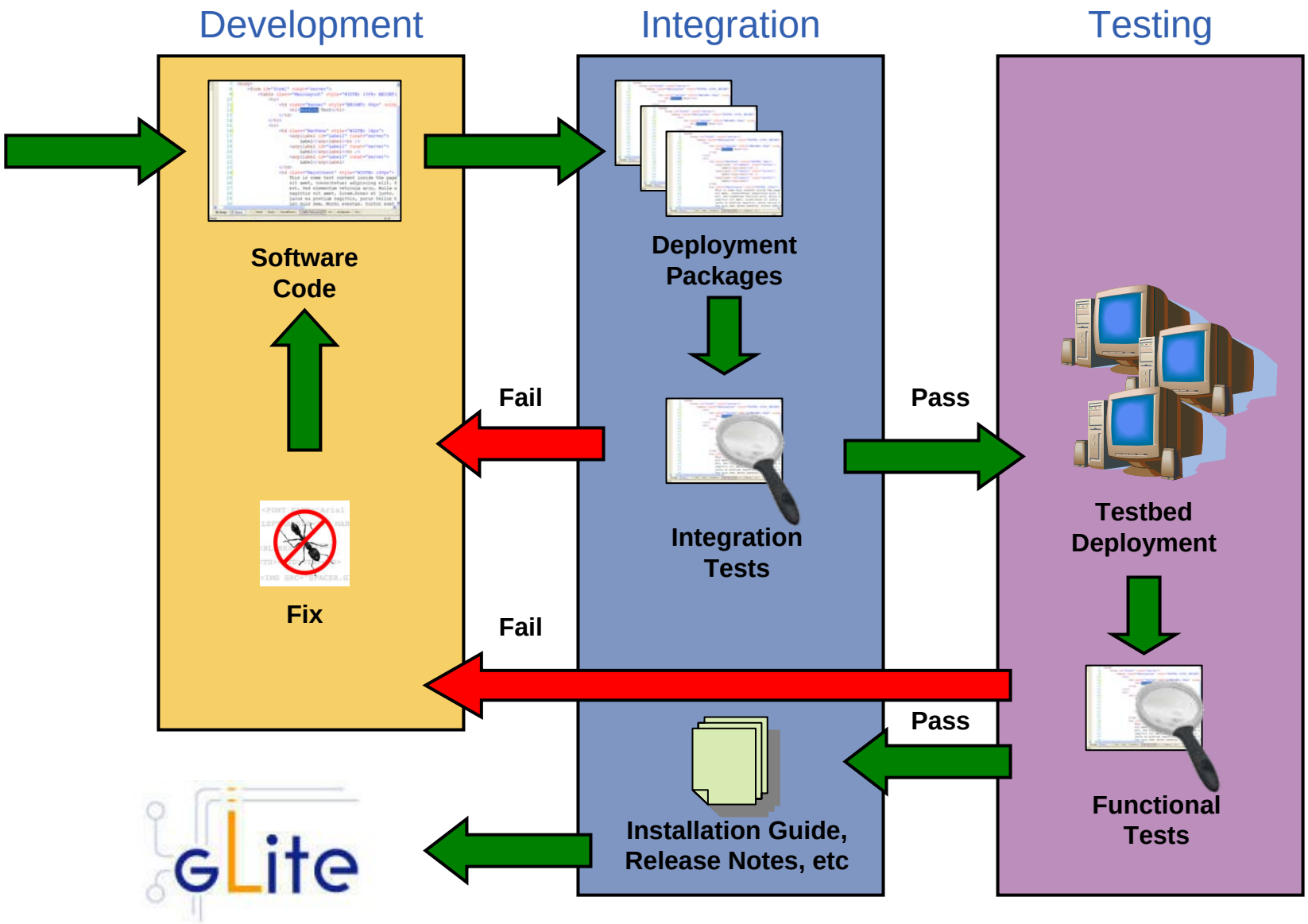
Software Configuration Management

- **The SCM Plan is the core document of the Software Process**

- **It contains a description of the processes and the procedures to be applied to the six SCM activity areas:**
  - Software Configuration and Versioning, tagging and branching conventions
  - Build Systems
  - Release Process
  - Change Control and the Change Control Board (CCB)
  - Bug Tracking
  - Process Auditing and QA Metrics
- **It is based on a number of standard methods and frameworks including:**
  - ISO 10007:2003 - Quality management systems -- Guidelines for configuration management, ISO, 2003
  - IEEE Software Engineering Guidelines (http://standards.ieee.org/reading/ieee/std/se)
  - The Rational Unified Process (http://www-306.ibm.com/software/awdtools/rup)
- **Standard best-practice solutions[1] have been adopted**

[1]S.P. Berczuk, Software Configuration Management Patterns, Software Patterns Series, Addison-Wesley, 2002
A. Di Meglio et al., A Pattern-based Continuous Integration Framework for Distributed EGEE Grid Middleware Development, Proc. CHEP 2004

Weekly Release Process



QA and SCM Metrics

- Several QA and SCM Metrics are mandated by the SCM and QA Plans
- Metrics are calculated periodically and published on the gLite web site
- Total complete builds done: 208
- Number of subsystems: 12
- Number of CVS modules: 343
  (development, integration modules, testsuites, documentation and tools)

SLOC

Total Physical Source Lines of Code (SLOC)

- SLOC = 583,620 (as of 7 February 2005)

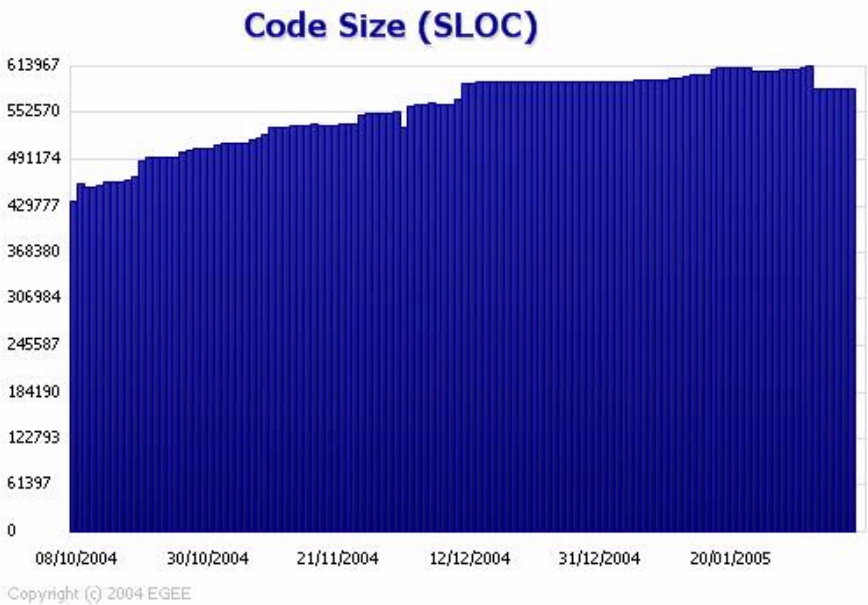
Total SLOC by language (dominant language first)

• C++	180816	(30.98%)
• Java	170749	(29.26%)
• Ansi C	134264	(23.01%)
• Perl	60972	(10.45%)
• Python	20039	(3.43%)
• sh	11859	(2.03%)
• Yacc	3635	(0.62%)
• jsp	640	(0.11%)
• Lex	335	(0.06%)
• csh	217	(0.04%)

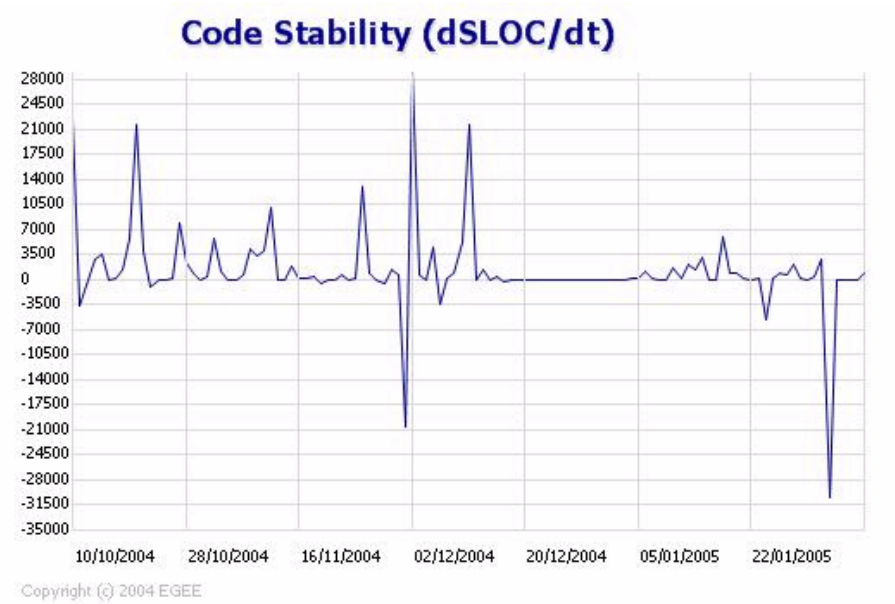
Software provided by different sources

Diversity increases the complexity

Code Size and Stability



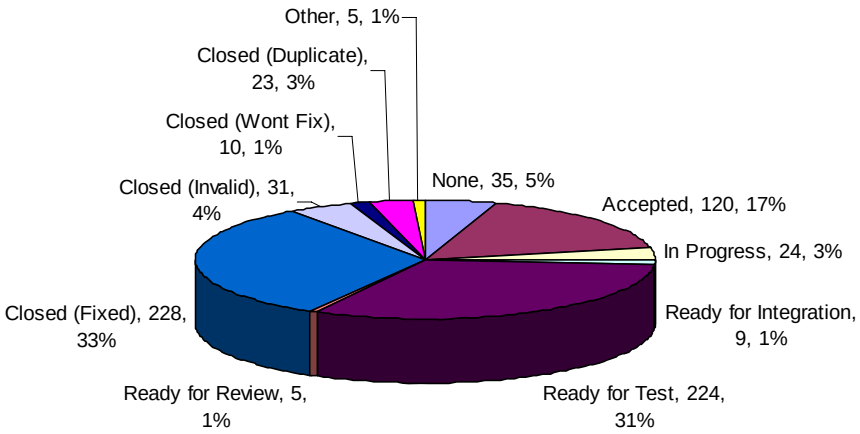
The Code Size chart shows the changes in total number of SLOCs during the life of the project



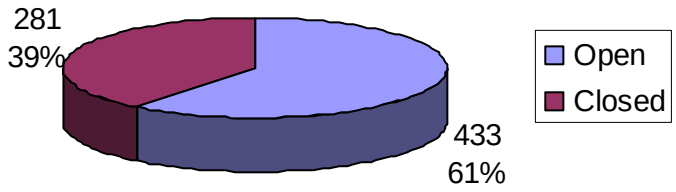
The Code Stability chart shows the change rate of code size during the life of the project.
As the project nears completion the rate should approach 0

Bug Counts and Trends

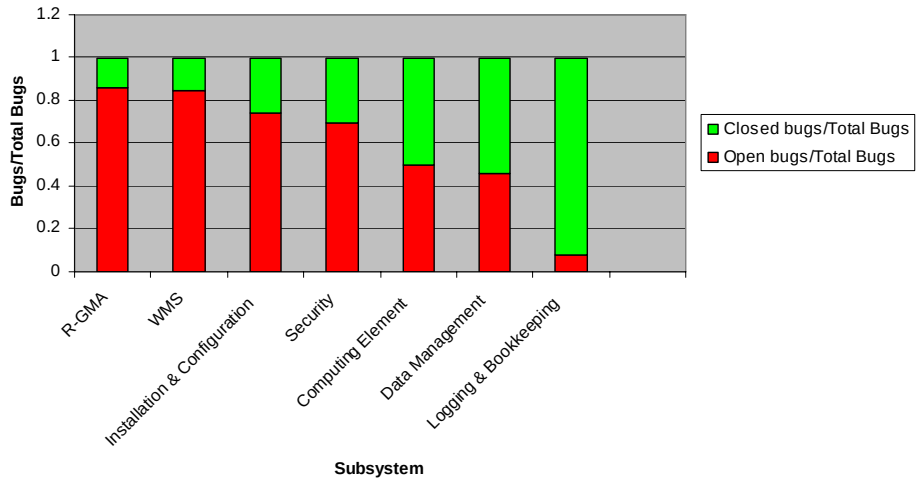
Bugs by status



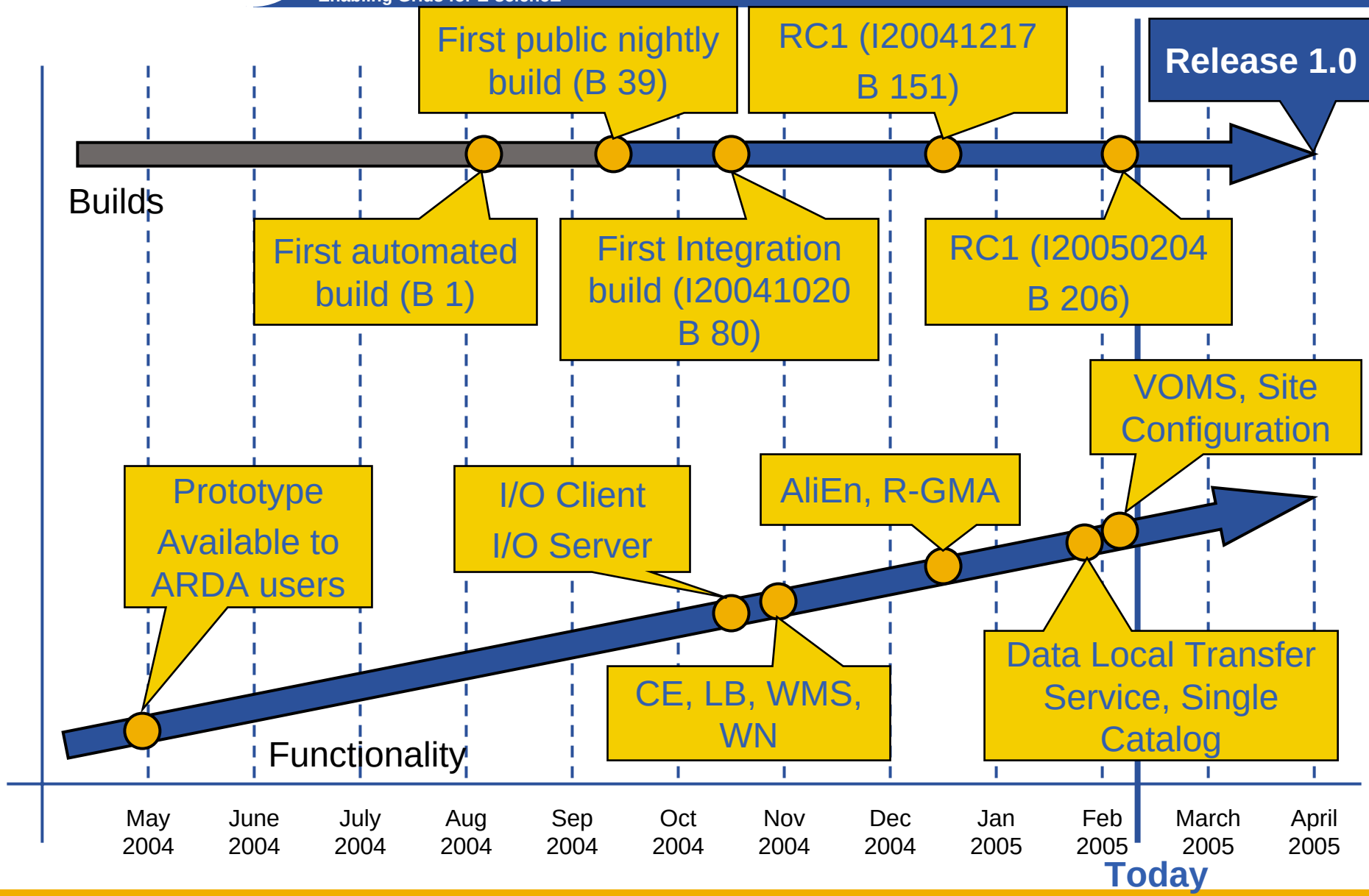
Total open/closed bugs
(Total = 714)



Relative bug status



Release Timeline



Testing Process

- **The process is documented in the Test Plan document (https://edms.cern.ch/document/473264)**
  - Organization of the testing activity: members and infrastructure
  - Features tested
  - Testing environment
  - Test methodologies and phases
  - Test development and execution
  - Schedule
  - Relationship with other testing activities
- **Distributed testbed, 3 sites, different variations of RH Enterprise Linux**
  - CERN: 20 machines
  - NIKHEF: 6 machines
  - RAL: 6 machines
- **Testing activities are coordinated with NA4 and SA1**

Available tests

Test suites for core components available:

- **Job Management**
  - The LCG WMS test suite (a part of the LCG certification test suite) has been ported, packaged and documented in collaboration with LCG to work with the gLite job submission components.
  - Regression and stress tests being developed
- **Data Management**
  - Testing done in collaboration of ARDA
  - gLite I/O: Basic functional test suite exists, including regression and stress test cases.
  - FPS: test suite being developed
  - Catalog: test suite being developed
- **R-GMA**
  - Test cases defined; test suite being implemented
- **Security**
  - Test cases being worked out

Common Tools used by JRA1

- **When available, in collaboration with LCG Software Process Infrastructure (SPI)**
  - Avoiding duplication of efforts and tools
    - Used by many EGEE activities and other projects
  - Savannah
    - Bug tracking
    - Task Management
    - Automatic Generation of QA reports
  - CVS
    - Code repository
    - Change tracking and control
  - QMTest, CppUnit, pyUnit (Testing tools)
  - CodeWizard (Code checking Tool)

- **Other Tools used:**
  - Build Tools: Ant, CruiseControl
  - Testing Tool: jUnit
  - Code checking Tool: Jalopy

Relationship to related projects

- **Condor**
  - Member of the Design Team
  - Specific Condor developments for the CE
  - Evaluating Stork for Data Management
  - Part of development testbed
- **Globus**
  - Member of the Design Team
  - Specific Globus developments for CE
- **UK Computing for Particle Physics (GridPP)**
  - Metadata
- **National Middleware Initiative (NMI)**
  - Discussions on processes and potential common testing
- **Open Middleware Infrastructure Institute (OMII) & National Grid Service (NGS)**
  - Mutual software evaluations on-going
  - Discussions on standards
- **Open Science Grid (OSG)**
  - Architecture & Design documents used for OSG blueprint
  - Discussions on common strategies and software
- **Many other interactions with DILIGENT, GridLab, GEMSS, MammoGrid, Unicore, …**

JRA1 Accomplishments and Issues

Accomplishments

- **Produced gLite Architecture and Design documents**
  - Already referenced outside EGEE
- **Software development processes in place**
  - being exercised by SA1 on the Pre-Production Service
- **Software available to pilot Applications and SA1**
- **Actively following and contributing to emerging standards (GGF, OASIS, etc.), but not early adopters**
- **Produced all deliverables but MJRA1.4 (one month slip) in time**

Issues

- **Disagreement on what software is made available**
  - Handled via a Management Task Force
- **Web Services technologies fulfilling requirements of EGEE applications**
  - Web Services industry support (tooling)
  - Standards are not yet mature, and do not always have a robust implementation
- **Staffing in Testing/Integration under pressure at this stage**
  - Being handled with partners
- **Communication with software clusters not optimal (difficult to follow each other's work)**
  - Make better use of tools

Plan for the next 15 months

- PM12: Release 1
- Incremental releases (bug fixes) are foreseen on a regular basis for the whole period
- PM14: Revision of the architecture document.
- PM15: Revision of the design document.
- PM18: Test plan for Release 2
- PM19: Functionality of Release 2 will be frozen.
- PM20: Integration of Release 2
- PM21: Release 2 delivered to SA1
- PM24: Final report including assessment of work completed and outstanding issues

Summary

- **gLite architecture, design and release plan have been produced**
- **A software process is in place which is being exercised by SA1 on the Pre-Production Service**
- **gLite release 1.0 with key functionalities implemented will be produced by end of March according to the schedule**
  - Moving towards Web Services