Generic Middleware Services

LCG Workshop
23-24 March 2004

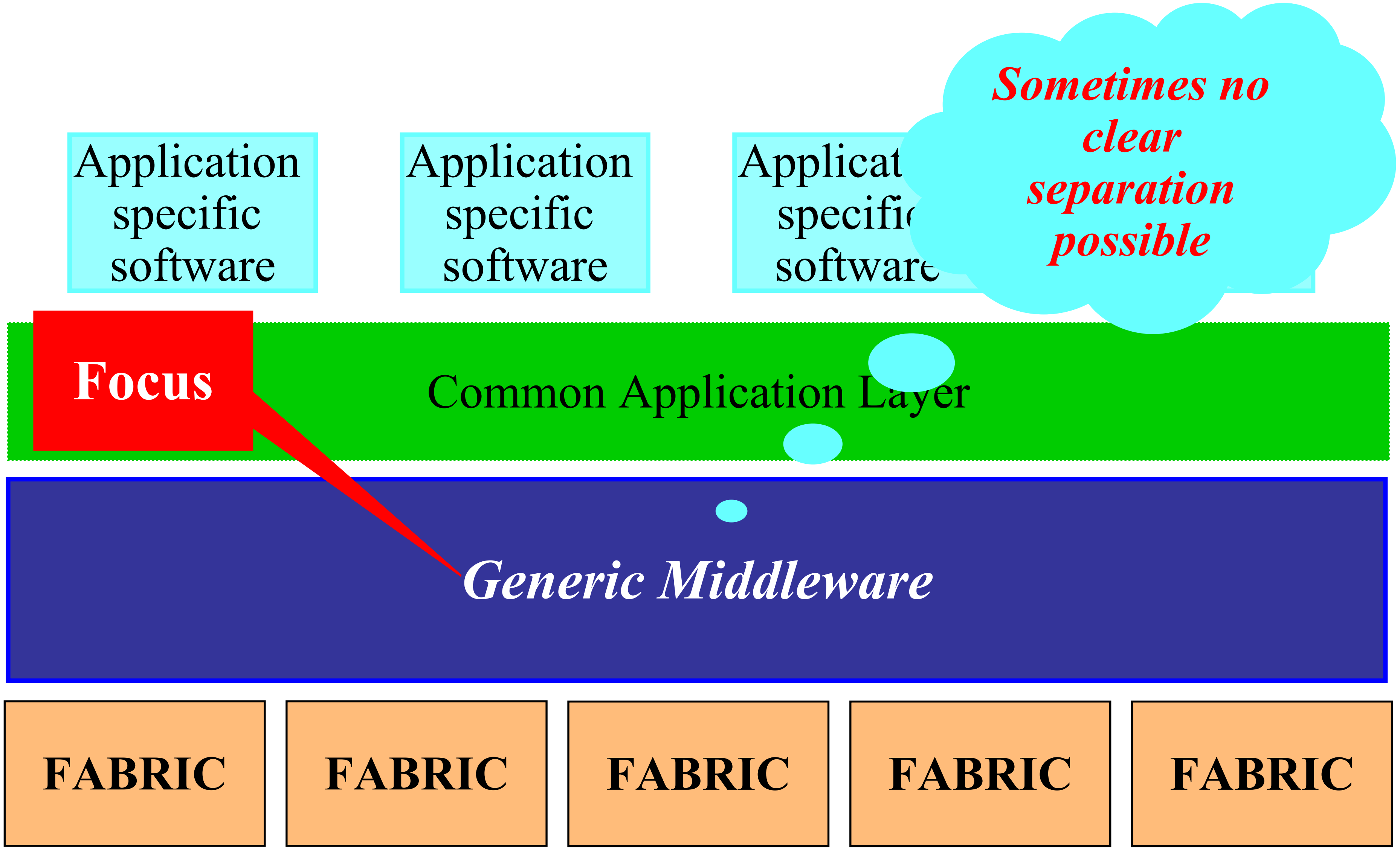
Erwin Laure (Cern)
Miron Livny (Univ. Wisconsin), Francesco Prelz (INFN), Steve Fisher (RAL)
Frederic Hemmer, Predrag Buncic, Peter Kunszt (CERN)
David Groep (NIKHEF), Torre Wenaus (LCG AA)

EGEE is proposed as a project funded by the European Union under contract IST-2003-508833

Contents

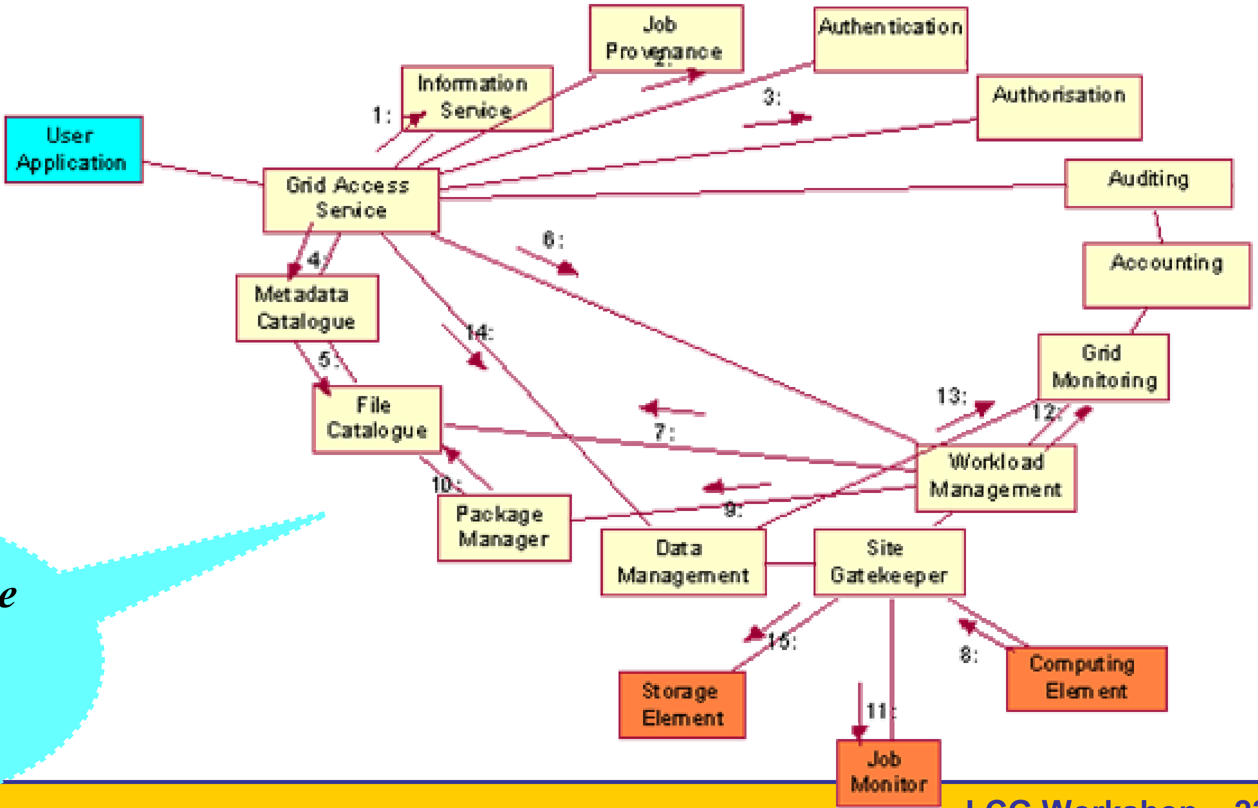
- Motivation
  - The message from the ARDA RTAG – and what we learned from it
- Initial core services
  - Data management
    - Storage Element
  - Workload management
    - Compute Element
  - Information and monitoring
- The path towards a prototype

Motivation
A high level end-to-end Grid Architecture


Sometimes no clear separation possible
Application specific software
Application specific software
Applicat specifi software
Focus
Common Application Layer
Generic Middleware
FABRIC
FABRIC
FABRIC
FABRIC
FABRIC

The message from the ARDA RTAG

- Blueprint of an architecture for distributed analysis on the Grid
- ARDA defines this architecture in terms of a set of collaborating Grid services with well-defined interfaces



Many of these services are generic middleware

The message from the ARDA RTAG

- Propose to build a prototype of the ARDA architecture
- Recommend that the ARDA architecture should be implemented as an OGSI compliant set of services
  - Note: OGSI in the meanwhile superseded by WSRF
- Propose a four-prong approach:
  - Re-factoring of AliEn and other services into ARDA, with a first release based on OGSI::Lite; consolidation of the API working with the experiments and the LCG-AA; initial release of a fully functional prototype. Subsequently implementation of agreed interfaces, testing and release of the prototype implementation.
  - Modeling of an OGSI-based services infrastructure, performance tests and quality assurance of the prototype implementation

    Generic Middleware
  - Interfacing to LCG-AA software like POOL and ROOT
  - Interfacing to experiment's frameworks, with specific meta-data handlers and experiment specific services

… how we interpret it

Deliver end-to-end capabilities (*from user to fabric*) and stability (*deployable*) at the price of services offered (*functionality*)

Services provide a natural abstraction and powerful software engineering constructs

AliEn provides a useful and stable suite of services as it meets the expectations of the Alice experiment

… and what we learned from it

- Provide a prototype of generic Grid middleware quickly, which experiments can interface to
- Use a service oriented approach
  - OGSI is not applicable anymore – use plain webservices
  - Follow WSRF
  - Migration to WSRF should be "easy" once it is settled
- Formed a design team with members from
  - AliEn
  - VDT
  - EDG

  all members will be part of EGEE as of April 1st
- Started intense technical discussion to
  - Break down the proposed architecture to real components
  - Identify critical components (and what existing software to use for the first instance of a prototype)
  - Define semantics and interfaces of these component
  - Coordinate with LCG AA (e.g. POOL)

Ranking Services (and functionality)

Reality: virtually all the services identified by the ARDA are essential. However, only a subset of them can be (fully) developed and implemented in the first phase.

- Which one can be given low priority in the first round?
- Which one can be only partially addressed?
  - Functionality
  - Centralized or distributed
  - Robustness, stability, performability
  - Ease of use, deployment and management
- Which one unify our middleware base?
- Which one are ready “out of the box”?

Initial Services

- Data management
  - Storage Element
- Workload management
  - Computing Element
- Information and monitoring

- Guiding principles:
  - Lightweight services
    - Easily and quickly deployable
  - Interoperability
    - Allow for multiple implementations
      - Being based on WS should help
  - Co-existence with deployed infrastructure
    - Run as an application

Mid- to long-term goals

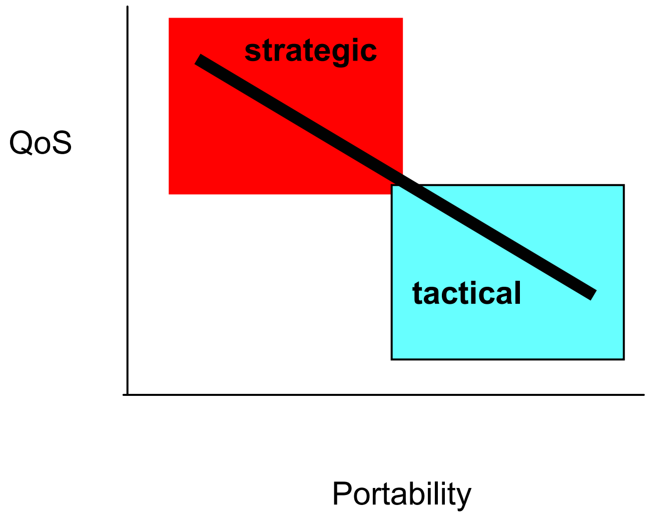
- Note: security not explicitly mentioned
  - Start with a minimalist approach
  - A few words on it later

Data Management

- Main components:
  - Storage element
    - Administration of space
    - File access
  - Catalogs
    - File catalog
    - Replica catalog
    - (Metadata catalog)
  - File transfer service

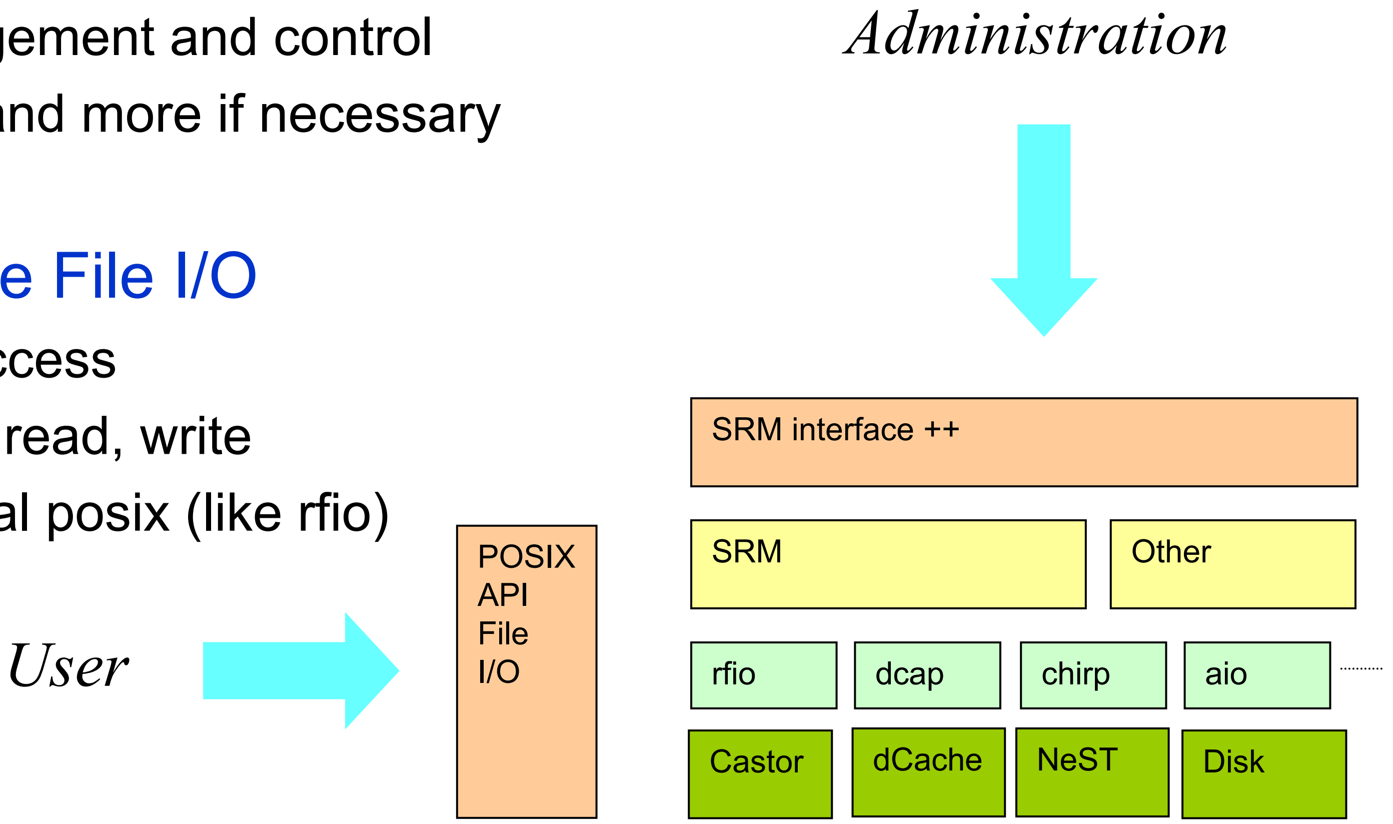
Storage Element

- ‘Strategic’ SE
  - High QoS: reliable, safe..
  - Has usually an MSS
  - Place to keep important data
  - Needs people to keep running
  - Heavyweight
- ‘Tactical’ SE
  - Volatile, ‘lightweight’ space
  - Enables sites to participate in an opportunistic manner
  - Lower QoS



Storage Element Interfaces

- SRM interface
  - Management and control
  - SRM and more if necessary
- Posix-like File I/O
  - File Access
  - Open, read, write
  - Not real posix (like rfio)



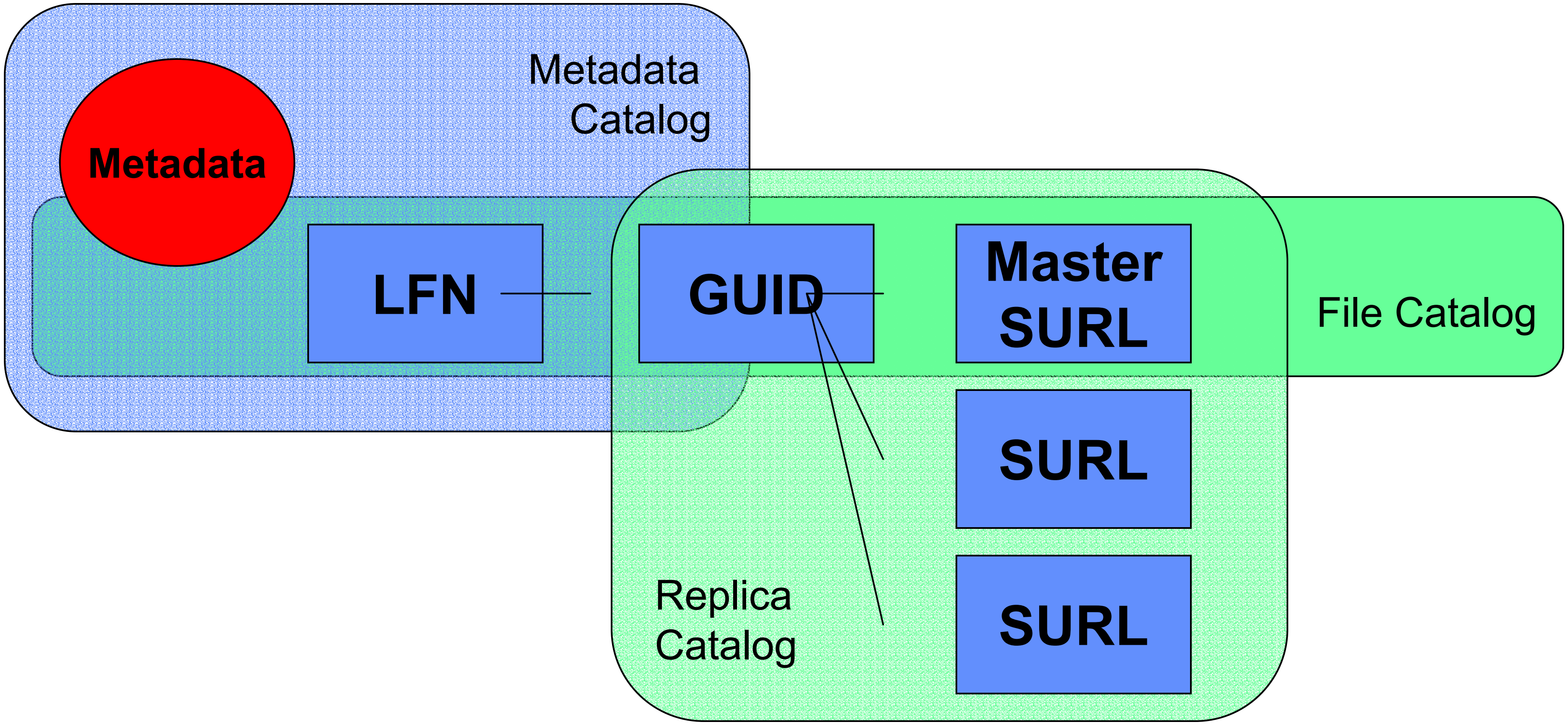
File Transfer Services

- Essentially a queue overseeing actual transfers over some wire protocol
- Start with GridFTP as default wire protocol
- Advantages to File Transfer Service:
  - Local control of network and storage resource possible
  - Avoid requests for the same file over the WAN by 500 simultaneous jobs (currently a problem in LCG)
  - Fail safety and recovery mechanisms
  - Asynchronous transfer API possible
  - Should allow to schedule file transfers much like jobs

Catalogs

- File Catalog
  - Filesystem-like view on logical file names
- Replica Catalog
  - Keep track of replicas of the same file
- (Meta Data Catalog)
  - Attributes of files on the logical level
  - Boundary between generic middleware and application layer

Files and Catalogs Scenario



File Placement Service

- Service to ‘bring files into/out of the grid’ and to replicate files
- Makes use of File Transfer Service
- Includes catalog interaction

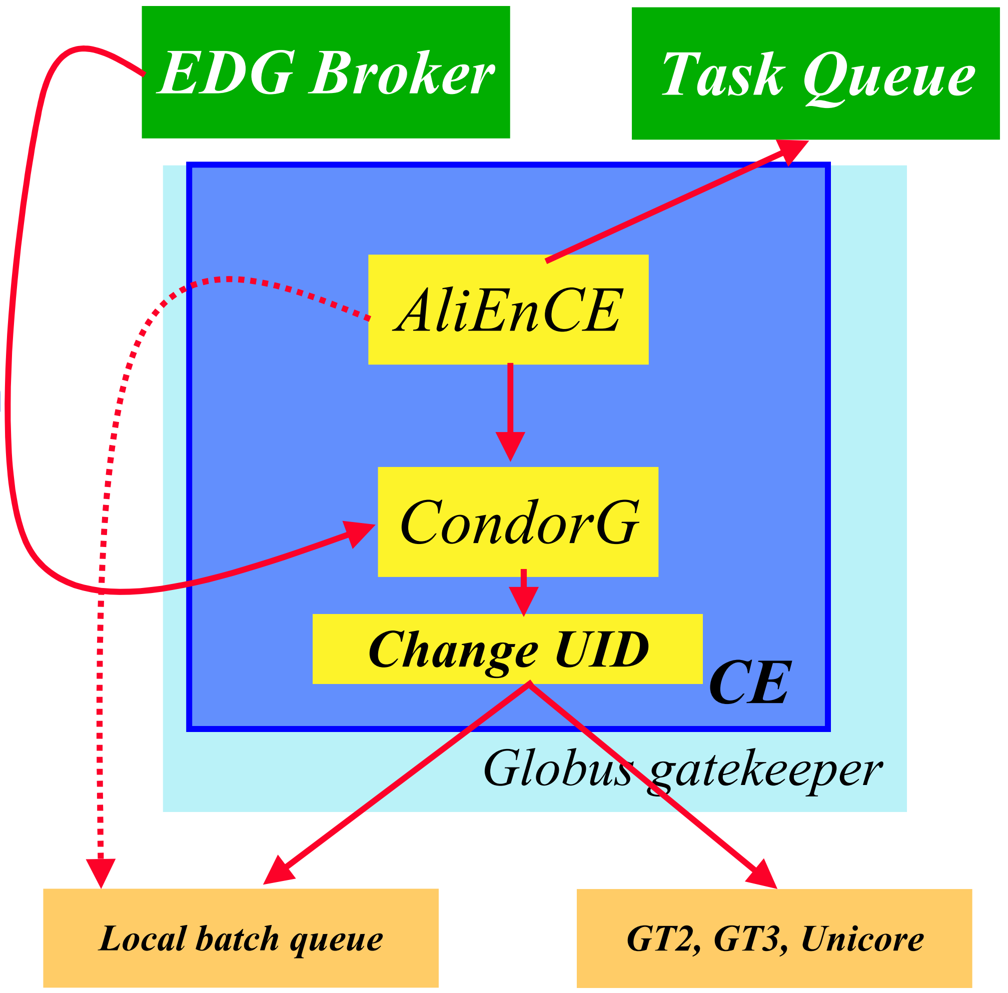
- Schedule file transfers much like jobs

Workload Management Services

- Distributes the workload on the Grid resources
  - AliEn uses a pull model (can also push jobs to other Grid systems); EDG a push model, VDT supports both (uses mainly push)
- Site must control (access and priority) consumption of ALL local resources
  - head node, worker nodes, storage resources, network bandwidth
- All resources must be protected by a claim and leasing protocol
- Requires a hierarchy of optimizers, planners, matchmaker, …
  - capable of dynamic (eager, lazy, just in time) workload management
- Requires flexible local resource management policies

Computing Element

- Layered service interfacing
  - various batch systems (LSF, PBS, Condor)
  - Grid systems like GT2, GT3, and Unicore
- CondorG as queuing system on the CE
  - Allows CE to be used in push and pull mode
- Call-out module to change job ownership (security)
- Lightweight service
  - should be possible to dynamically install e.g. within an existing globus gatekeeper



Information Service

- Adopt a common approach to information and monitoring infrastructure.
- There may be a need for specialised information services
  - e.g. accounting, package management
  - these should be built on an underlying information service
- A range of visualisation tools may be used

Access Services (and APIs)

The Access services and APIs expose the capabilities of the computational environment to the end user.

- User entry point to distributed analysis services
- Stateful service authenticating and authorizing the user
- One instance per analysis session
- Instantiates the proper UI according to user roles
- Service calls can be routed through these access services

AliEn offers a rich set of such services and APIs.

Authentication/Authorization

- Different models and mechanisms
- Authentication based on Globus/GSI, AFS, SSH, X509, tokens
- Authorization
  - AliEn: exploits mechanism of RDBMS backend
  - EDG: gridmap file; VOMS credentials and LCAS/LCMAPS
  - VDT: gridmap file; CAS, VOMS (client)

Security and protection at a level acceptable by fabric managers and end users needs to be discussed and “blessed” in advance.

A minimalist approach to security

- Need to integrate components with quite different security model
- Start with a minimalist approach
  - Based on VOMS (proxy issuing) and myProxy (proxy store)
  - User stores proxy in myProxy from where it can be retrieved by access services and sent to other services
  - Credential chain needs to be preserved
    - Allow service to authenticate client
  - Local authorization could be done via LCAS if required
  - User is mapped to group accounts or components like LCMAPS are used to assign local user identity

Middleware Working Document

Abstract: This working document is used to break down the high level services defined by ARDA to actual components and tries to define the initial set of services provided by the prototype, their interfaces, as well as the technology/systems exploited. The appendix maps these components to existing implementations coming from Alien, EDG, and VDT. The structure and initial AliEn input is taken from Chapter 5 of Draft v0.2 of the ARDA document (unpublished)

Started after a meeting in December as a vehicle to exchange and record information and ideas among the middleware providers.

- Identification of services
  - Service interplay and semantics
- Understand how existing MW could implement these services
  - Input from AliEn, EDG, VDT, commercial, …. (others?)
- Specify interfaces to applications
- Current draft v0.16 available at: http://cern.ch/erwin/ARDA-WD.0.16.zip

Towards a prototype

- Focus on key services discusse   nents
- Initially an ad-hoc installatior
- Aim to have first instance r
  - Open only to a small use
  - Expect frequent changes (
    feedback and integration of
  - Cf. talk by Massimo tomorrow
- Enter a rapid feedback cycle
  - Continue with the design of rem   services
  - Enrich/harden existing service   sed on early user-feedback

Initial prototype components for April'04 To be extended/ changed

- Access service:
  - AliEn shell, APIs
- Information & Monitoring:
  - R-GMA
- CE:
  - AliEn CE, Globus gatekeeper, CondorG, LCAS/LCMAPS
- Workload mgmt:
  - AliEn task queue
- SE:
  - SRM (Castor), GridFTP, GFAL, aoid
- File Transfer Service:
  - AliEn FTD
- File and Replica Catalog:
  - AliEn File Catalog, RLS

Summary

- Develop a lightweight stack of generic middleware useful to experiments based upon existing components
- Focus is on re-engineering and hardening
- Early prototype and fast feedback turnaround envisaged
- Expected to be the generic middleware component of the ARDA project
- Development done in the context of the EGEE project
  - One set of middleware
  - No competing ARDA, LCG, EGEE flavors
- Feedback and contribution via the ARDA project (cf. Massimo's talk tomorrow) highly welcome