Enabling Grids for E-science in Europe

Vilnius, September 2004

EGEE Middleware Plans: gLite .

Lightweight Middleware for Grid Computing

Richard Hopkins

Training Team, National e-Science Centre, Edinburgh, UK

rph@nesc.ac.uk

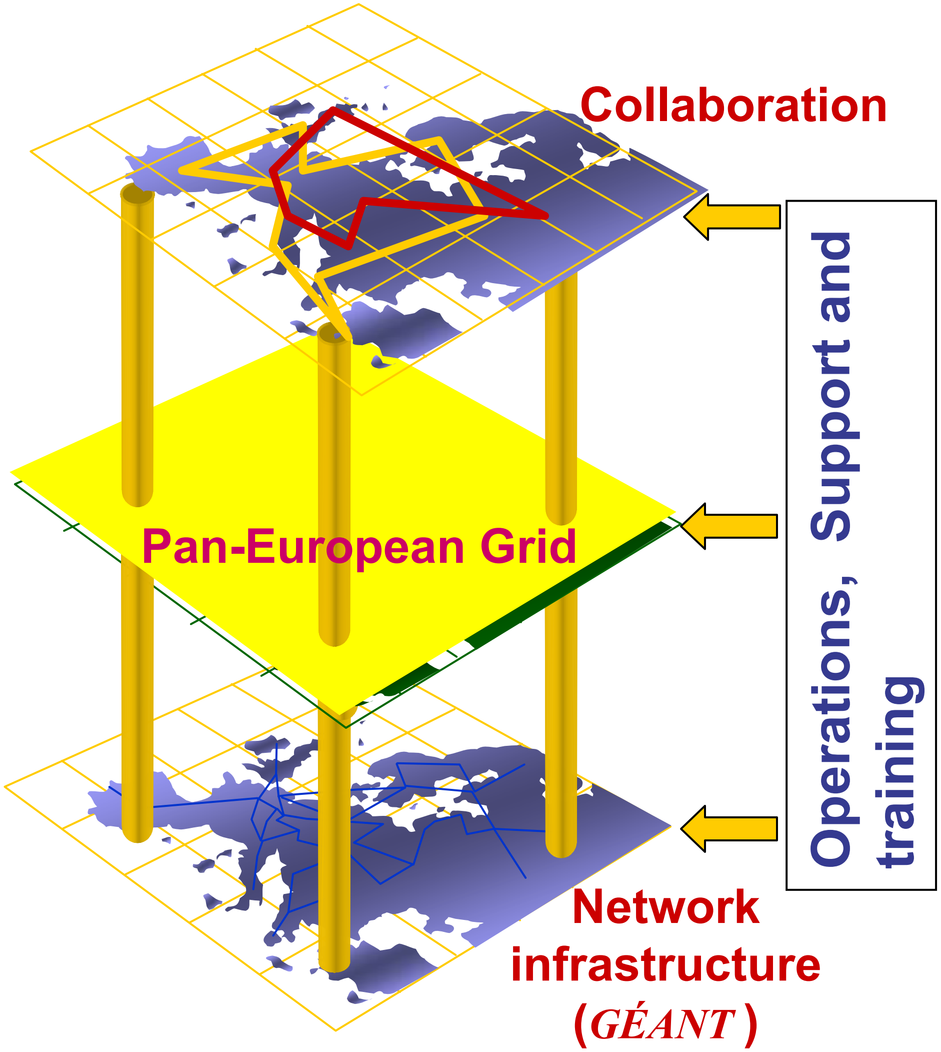
EGEE is a project co-funded by the European Commission under contract INFSO-RI-508833

- **What, Why, How**
- Service Oriented Approach
- The gLite Services

Objective : Grid Infrastructure

Build a large-scale production grid service to:

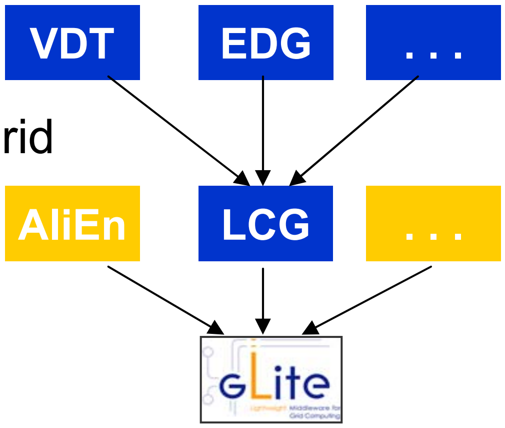
- Underpin European science and technology
- Link with and build on national, regional and international initiatives
- Foster international cooperation both in the creation and the use of the e-infrastructure


Collaboration
Pan-European Grid
Network infrastructure (*GÉANT*)
Operations, Support and training

Approach

- Role gLite is intended as the middleware for Pan-European Grid
- Approach:

  Combining and re-engineering existing components (as far as possible)
  - Middleware (MW) components now work reasonably well - problems usually elsewhere
  - Substantial body of experience of MW development, packing and use
- Based on LCG2
  - What is currently used within the EGEE project
  - Can be viewed as an initial version of Pan-European Grid middleware
- gLIte deals with
  - LCG2 shortcomings
  - Advanced application needs
  - State-of-the-art Internet approaches and standards



VDT

EDG

. . .

AliEn

LCG

. . .

gLite

Improvements

Enabling Grids for E-science in Europe

- Extended Functionality
  - Incorporates features from other products
  - Some new functionality
- Increased Modularity
  - Previous products tend to be monolithic – assumed to be used in entirety
  - Now need to move towards smaller scale of component combinability
- Improved Deployability
  - Modularity
  - Installation support
  - Non-dedicated machines
- Coherent architecture
  - extending functionalities and increasing modularity requires –
  - a revised picture of the organisation of functions
  - rationalisation of interface

Improvements

- Adopt Web Services Approach and Standards
  - The standard framework for developing multi-organisational distributed applications.
  - From a different starting point than grids, but now converging
  - Facilitates –
    - Cooperation between autonomously evolving components
      - A pan-European grid won’t work for long if it excludes autonomous evolution
    - Mix-and-match of components
      - As the overall functionality increases, increasing need to sub-set it
    - Combining Aspects of grids with other Web Service application domains
      - Important for extending uptake of grids into new areas
      - Maximising the pay-off from the grid development investment
    - Rapid Development of higher-level services
    - Leverage industrially-produced support packages

Architecture & Deployment Guiding Principles

- Lightweight - Easily and quickly deployable
  - Mix-and-match components
  - Multiple services running on the same physical machine (if possible)
- Interoperable
  - Client may talk to different independent implementations of the same service
- Resilient and Fault Tolerant
- Co-exist with deployed infrastructure
  - Reduce requirements on site components
  - Co-existence with LCG-2 essential for the EGEE Grid service
- Platform support- Goal is to have portable middleware
  - Building & Integration on RHEL 3 and windows
  - Initial testing (at least 3 sites) using different Linux flavours (including free distributions)

Architecture & Deployment Guiding Principles

- Service oriented approach
  - Based on web services standards

- Service autonomy
  - User may talk to services directly or through other services (like access service)

- Open Source software license
  - No restriction on usage (academic or commercial) beyond acknowledgement
  - That's for MW - for application software, Sites must obtain appropriate licenses before installation

- Main Documentation

  Application requirements: http://egee-na4.ct.infn.it/requirements/

  Architecture: https://edms.cern.ch/document/476451

  Design: https://edms.cern.ch/document/476451

  Release plan: https://edms.cern.ch/document/468699

Development & Deployment

- Design team works out architecture and design
  - Feedback and guidance from EGEE PTF & applications; Operations, LCG GAG & ARDA
- Components are initially deployed on a prototype infrastructure
  - Small scale (CERN & Univ. Wisconsin)
  - Get user feedback on service semantics and interfaces
- After internal integration and testing, components are delivered to grid operations group and deployed on the pre-production service
  - Early application access for new developments
  - Certification of selected components from gLite
  - Starts with LCG-2
- Smooth/gradual transition from LCG-2 to gLite for production operation

PTF – Project Technical Forum (http://egee-ptf.web.cern.ch/egee-ptf/default.htm)
GAG – Grid Application Group (http://project-lcg-gag.web.cern.ch/project-lcg-gag/)
ARDA - A Realisation of Distributed Analysis for LHC (http://lcg.web.cern.ch/LCG/peb/arda/Default.htm)

Development Cycle

- Prototyping short development cycles for fast user feedback –

Requirements

Planning & Design

Implementation

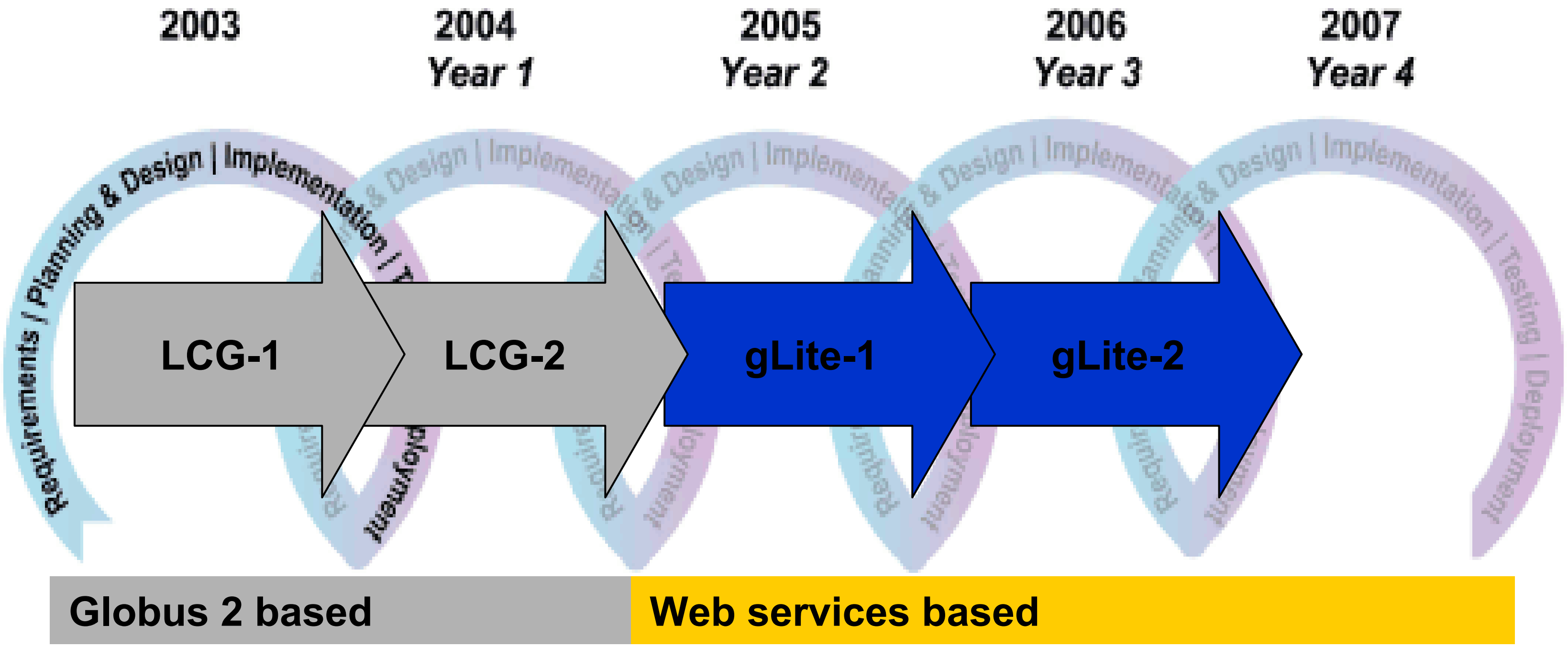
Testing

Deployment :

Prototype Infrastructure

→ Pre-production service

→ Production Service



- What, Why, How
- **Service Oriented Approach**
- The gLite Services

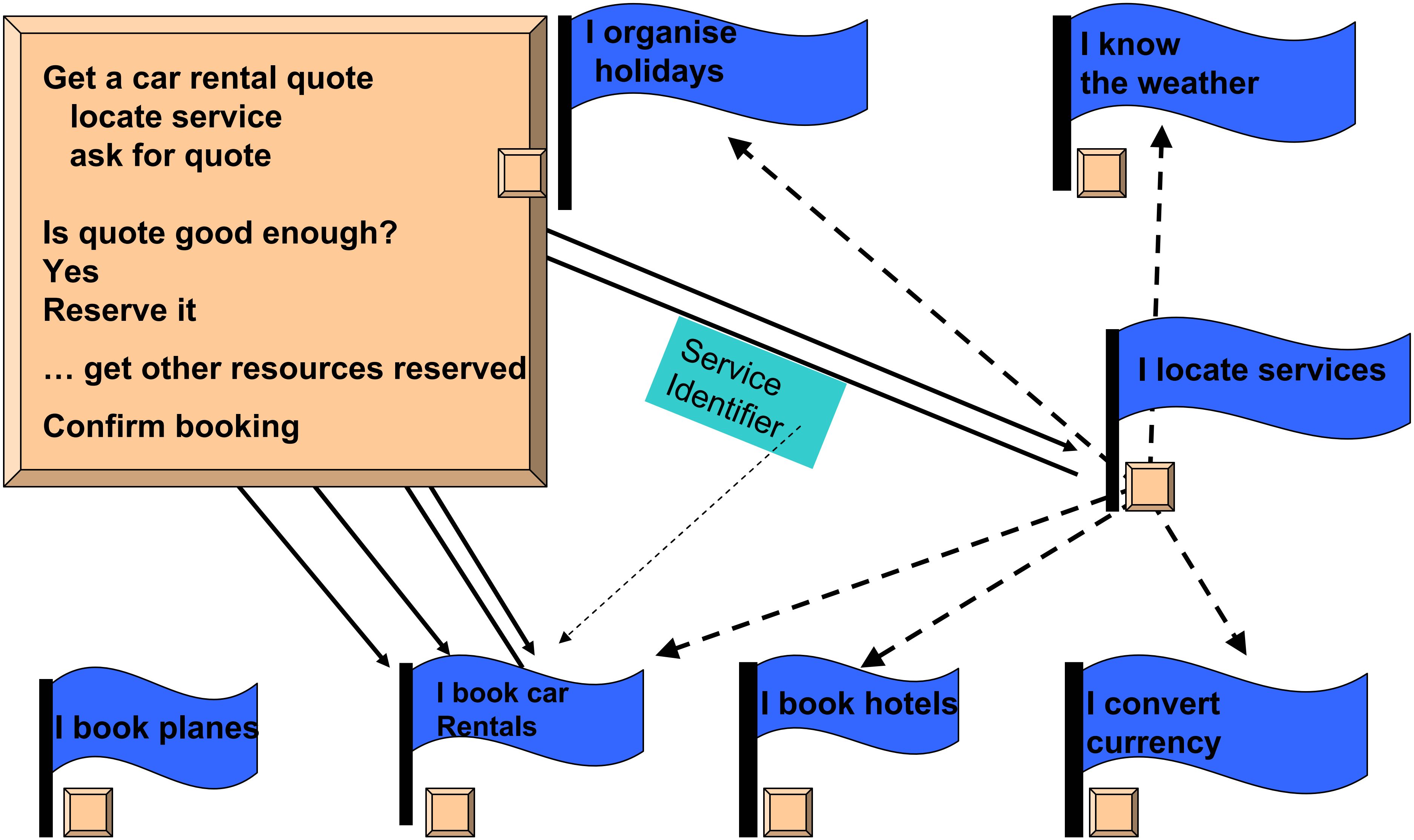
Web Services Approach

Web Services

- A person doing some complex task by surfing the web to look for and invoking services – book a hotel; book a plane; book a car hire; find best currency conversion rate …
- Web services is a framework that allows that same model to be used in writing an application – which itself becomes an “organise a holiday” service.
- Gives –
  - Coordinated **use of services**
  - Of up to international level of geographic separation
  - Crossing organisational boundaries

- Which are similar to the essential characteristics of grids –
- Coordination of **computational components**

Service Interaction



Get a car rental quote
locate service
ask for quote

Is quote good enough?
Yes
Reserve it

… get other resources reserved

Confirm booking

I organise holidays

I know the weather

I locate services

Service Identifier

I book planes

I book car Rentals

I book hotels

I convert currency

Essential Characteristics of Web Services Approach

Need to achieve effective cooperation even though

  - the different services are produced by different organisations, without any design collaboration
  - the services are autonomously evolving

- Loose coupling – minimum prior shared information between the designer of the two components of an interaction
  - Self-description – Meta data
  - Tolerance of partial understanding
  - Collaboration is on defining generic standards, rather than on specific design
- Easy for humans, hard for machines!!!!

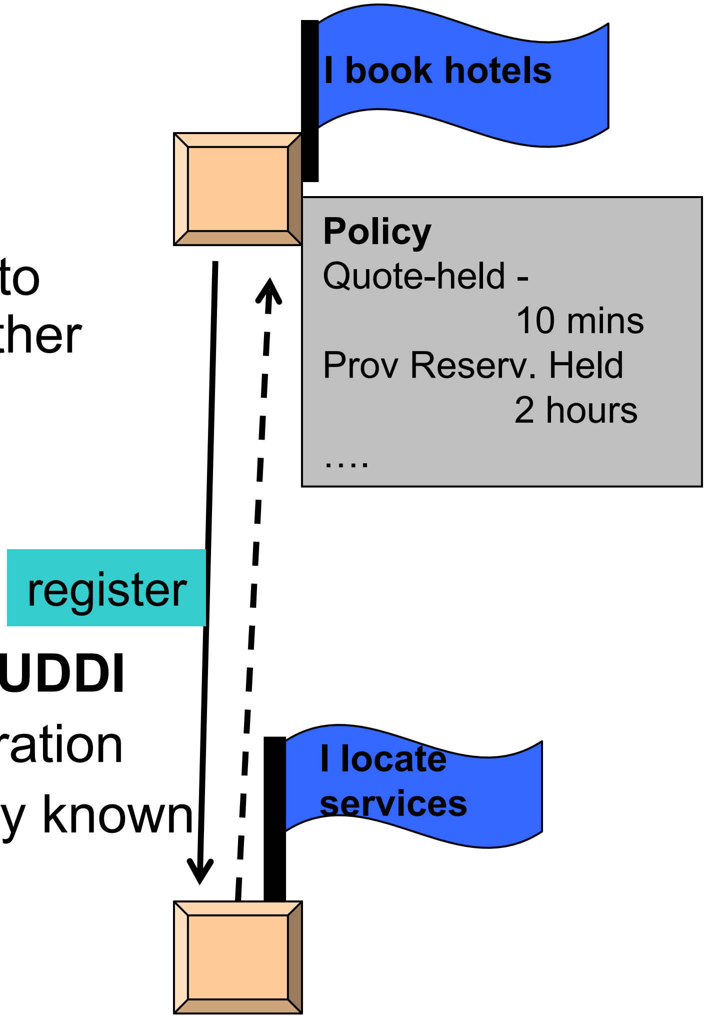
Basic WS

Basic Requirements

- **Service Description –WSDL**
  - Web services Definition language
  - Allows the code implementing one service to discover the functions and interface of another service

- **Usability Negotiation – Policies**

- **Service Registry and Discovery Services – UDDI**
  - Universal Description, Discovery and integration
  - Allows one service to register with a registry known by potential users

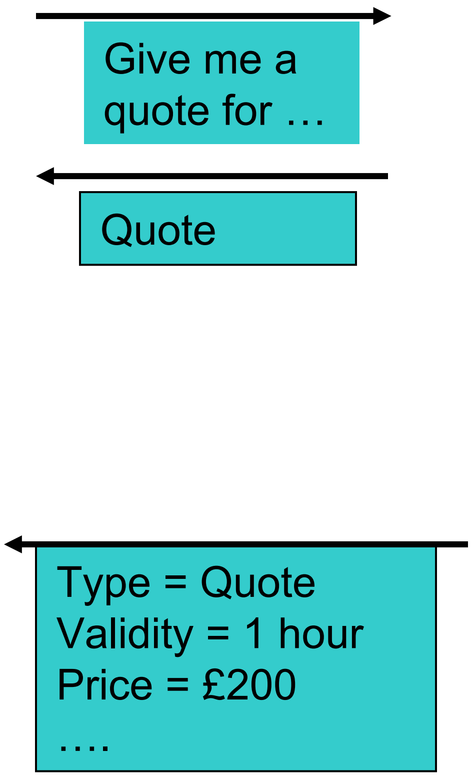


Basic WS

- Basic Requirement
  - **Service request and response** –
    - Service Oriented Access Protocol (SOAP)
    - Other models – notification

      Ask car rental service to notify me of any special deals

      To notify me if my reservation hold time is about to expire
  - **Common Language – XML Documents**
    - Tag-value pair
    - Defined by a schema
    - Used for messages, policies, service definition
  - **Self Description – tagged values, WSDL**
    - Necessary for loose coupling
    - Autonomous evolution
  - **Autonomous creation – discovery service**



Give me a quote for …

Quote

Type = Quote
Validity = 1 hour
Price = £200
….

Web Services – Evolving Standards

- Building more specific standards on top of basic WS standards
- WSRF – Web Services Resource Framework
  - E.g. a quote is a persistent entity which will need to be identified in subsequent interactions: quote → provisional booking → confirmed booking
  - Consistent standard framework for creating, identifying, destroying such entities
  - Destruction can be lifetime expiry
  - Essential part of Grid services – but not yet standardised
- WS-Addressing - Generalises communication of identities between services
- WS-Notification - Framework of notification interaction – subscribe, publish
- WS-Security – Framework for authentication and confidentiality

…….

Grids and Web Services

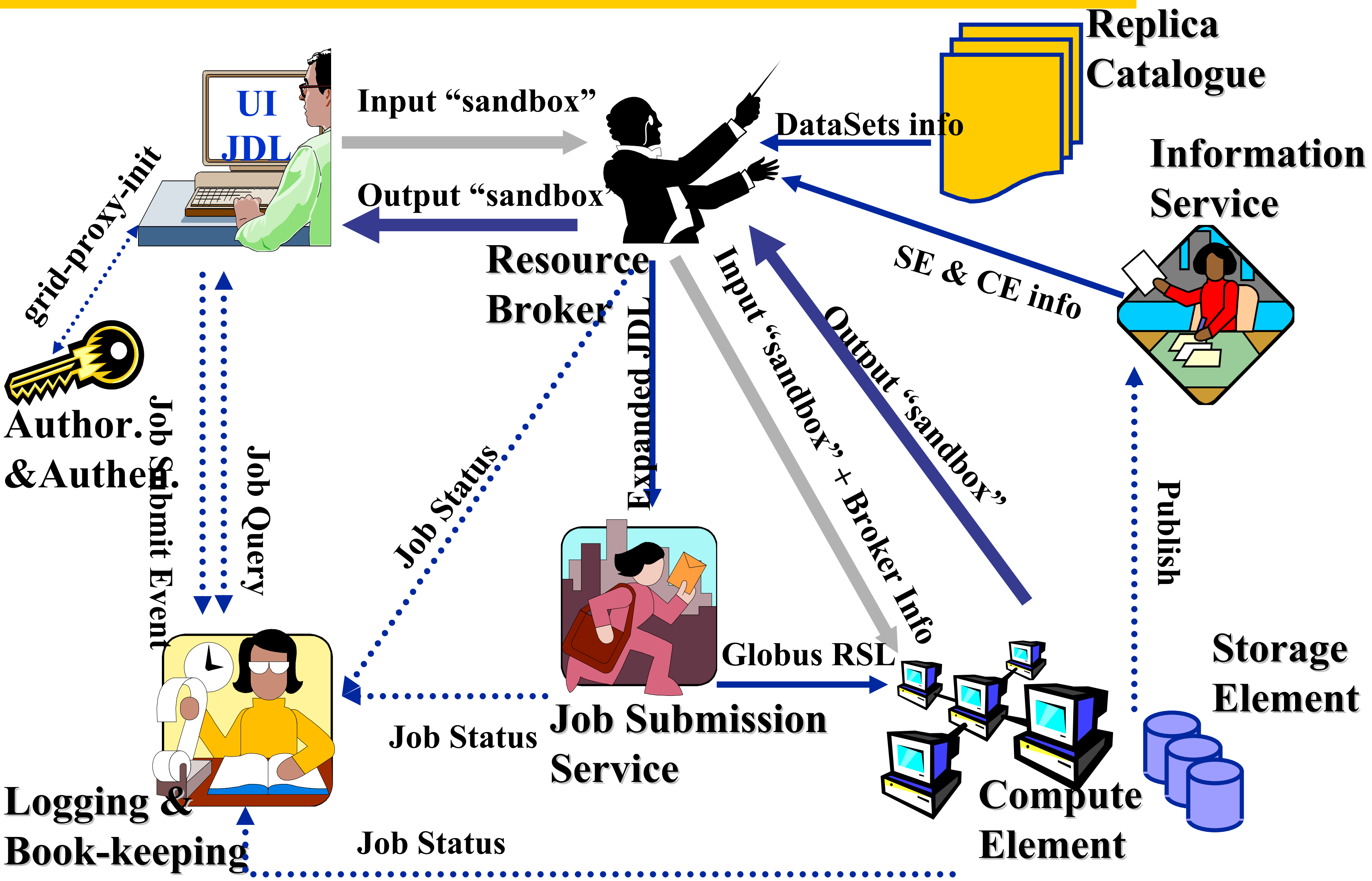
Grid Services

- Grid middleware can be seen as a special application area of web services dealing with – **VOs**, **data replication** and **job execution**
- Many advantages to basing grids on web services
  - Cooperation between autonomously evolving components
  - Mix-and-match of components
  - Combining Aspects of grids with other Web Service application domains
  - Rapid Development of higher-level services
  - Leverage industrially-produced support packages

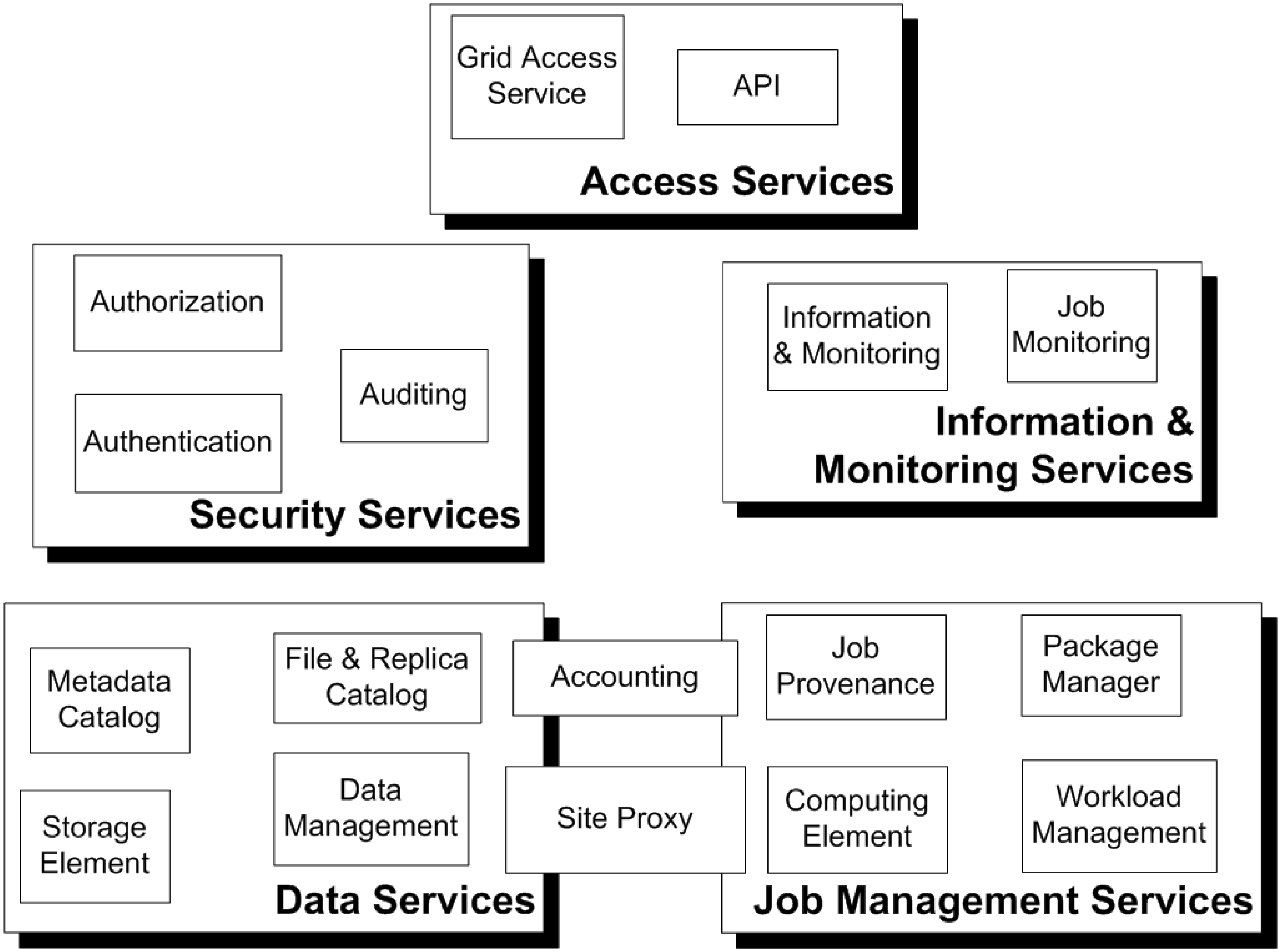
- Previously, independent simultaneous developments
  - Much of existing middleware components are not based on web services
- Now trying to bring together these two major developmental thrusts - OGSA – Open Grid Services Architecture
  - Re-engineering of existing components
  - Hampered by WS standardisation process being slower than the urgency in providing the next generation of pan-European grid middleware

- What, Why, How
- Service Oriented Approach
- **The gLite Services**

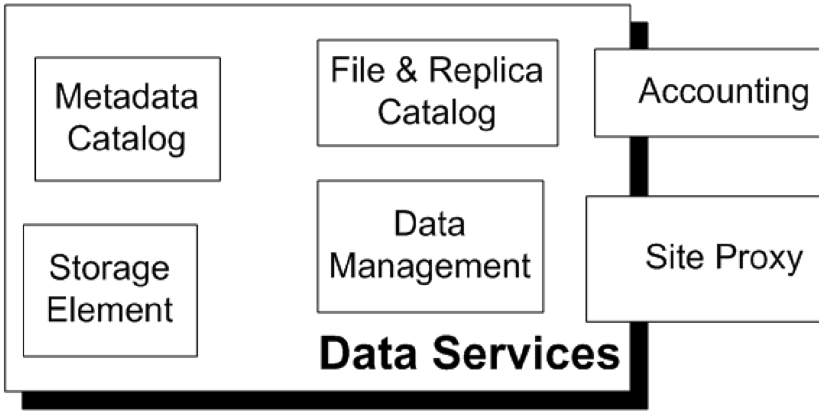
LCG2 (GILDA) Organisation



gLite Services



Data Services



Data Storage and Access

- Granuality is file, but different granularities can be supported
- Unix-like directory structure and navigation
- Tactical and Strategic storage
- For a job a "file name" can be a MetaData query

Storage Element

Resource providing physical storage runs MW to provide a standardised SE service interface

Subsumes a wide variety of storage characteristics

Operations – space management; data transfers, caching …

F&R Catalog –

Naming and replication

LFN – unique logical name within VO namespace

GUID – Globally unique identifier

SURL – to identify a file within the site's storage

Data Management

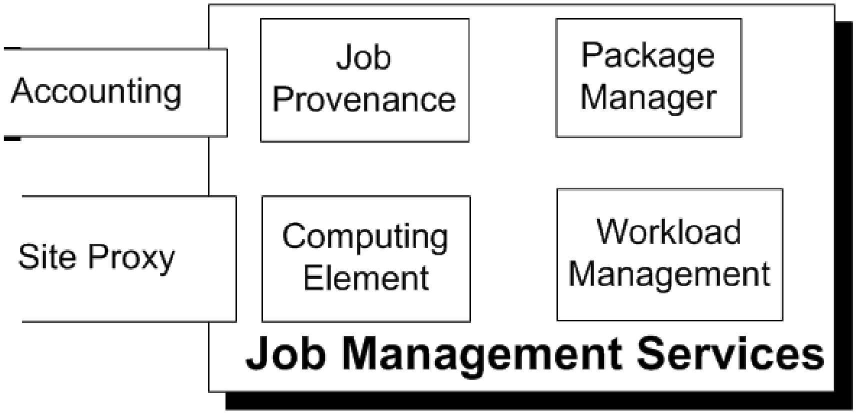
Organising data transfers between sites

VO - oriented

Metadata

Application specific description of data resources

Job Management



- **Job Management**

  manage jobs as they are entered into and progress through the system

  Presenting a consistent view of the status of any job in the system

  Common JDL for defining characteristics and requirments of jobs

Computing Element

Service provided by a computing resource

Run job – JDL

QoS preview

Cancel, suspend, resume

Status and signals

Workload M'mnt

Matchmaking

Scheduling jobs

Information supermarket

Both push and pull

CE autonomy – can always refuse job

Provenance

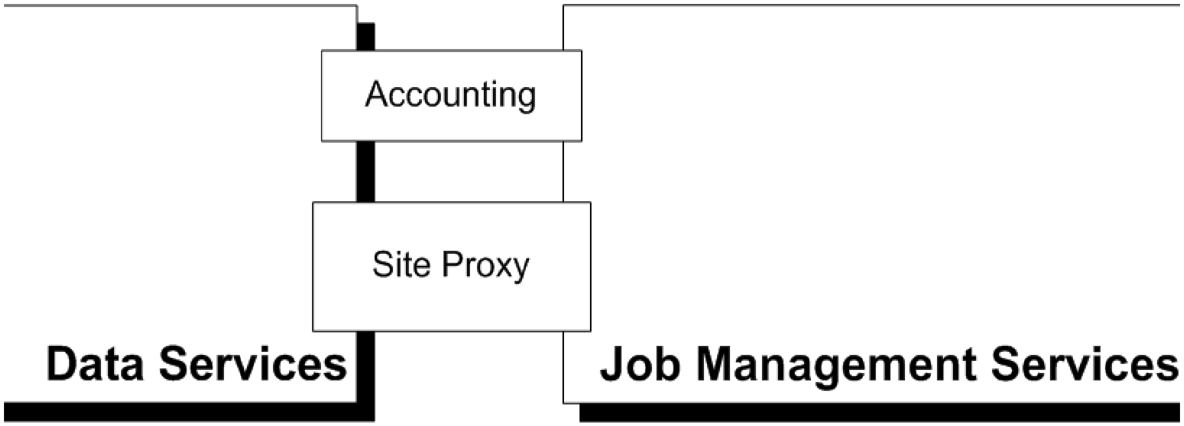
Completed execution archive -

De-bugging; performance tracking; re-execution; …

Package M'mnt

Assits S/W instal, configure, upgrade, remove

Usage Monitoring and Control



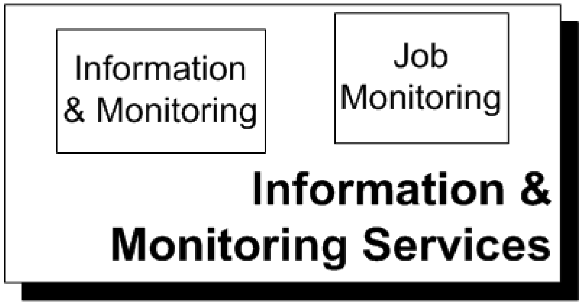
Accounting (Auditing)

- records information about resource usage by users and groups (e.g. a VO)
- statistics reports
- discovering and avoiding abuses
- charging users, maybe
- supporting quotas, maybe
- Maybe, market economy for load balancing
- Initially only within Job Management

Site Proxy

- SE and CE resources may be within the provider site’s firewall
- These resources accessed via the proxy which gives local control and monitoring

Information and Monitoring Services



Information & Monitoring

Job Monitoring

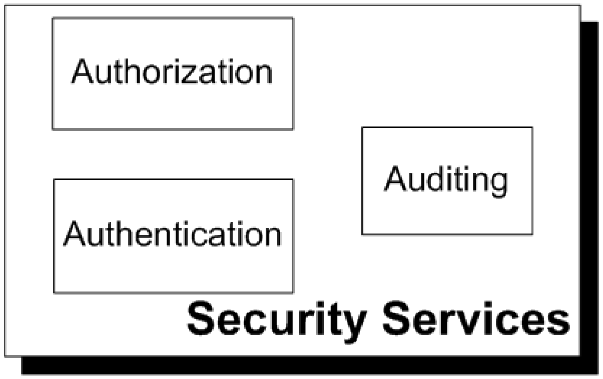
Information & Monitoring Services

Information & Monitoring Services –

Provide a means for users and other services to publish and obtain information, and use it for monitoring purposes

- **Basic I&M services**
  - Receives information from diverse producers
  - Presents as a single federated database on which consumer can run queries
  - Information is about status and events for jobs and resources
  - Provides a base service on which can be built
- **Job Monitoring**
  - Tracking and providing end user with
    - Status
    - Statistics

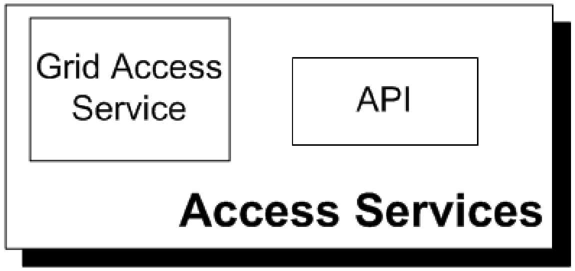
Security Services



Security Services

- Controlling Access to services and resources
- Authentication - Identification of entities – users, systems, services
- Authorisation -Allow or deny access
  - VOMS – VO membership service
- Auditing -Support post-mortem analysis of security related events

Access Services



- About Providing (and controlling) user access to whole of grid services

- **GAS – Grid Access Service**
  - Provides a façade for the other service – protects against changes
    - Similar to a Portal, but is for use by application software
    - Can build on it a browser user interface to it, like GENIUS
  - Create a new GAS instance for each user (at a location near to the user)
  - Has a lifetime determined by user privileges
  - Typical service oriented approach – GAS provides a "complete grid" service by using the other services

API – Application Programming Interface

- For specific programming languages
  - JAVA, … ????
- A library of code modules for building
  - Grid applications
  - Graphical user interfaces
  - Grid Portals
- A programming interface for accessing grid services

Summary

- Combines and re-engineers existing components
- LCG2 is starting point
- Extended functionality
- Web Services based
- Increased Deployability
- Interoperable
- Lightweight
- Co-exists with existing infrastructure

- Detailed release plan
- First components for pre-production service during autumn
- Continuous integration and testing scheme defined and adopted

- THE END