Grid middleware

www.see-grid.eu

Grid training days in Timişoara, 7-8 December 2006

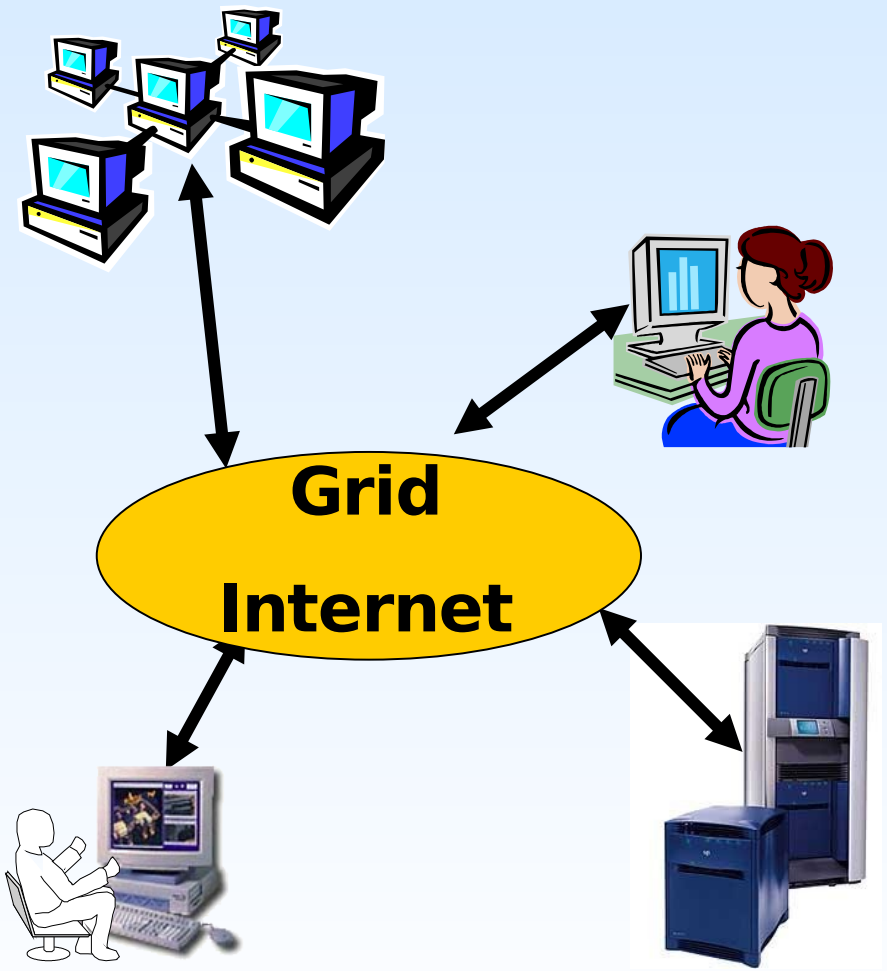
Alexandru Stanciu
ICI - RoGrid

Overview

- Grid Computing introduction
  - Grid Middleware
  - Virtual Organizations
- Major components of the Grid Middleware
  - Workload Management
  - Data Management
  - Information Services
  - Security

Grid overview

- A Grid is a collection of computers, storages, special devices, services that can **dynamically join and leave**
- They are **heterogeneous** in every aspect
- They are geographically **distributed** and connected by a **wide-area network**
- **They can be accessed on-demand by a set of users**

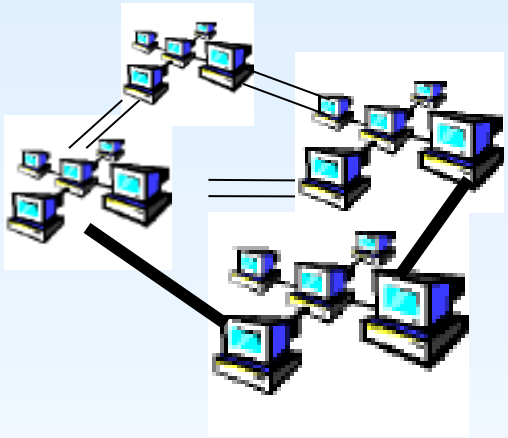


Characteristics of a Grid system

Numerous Resources

Ownership by Mutually Distrustful Organizations & Individuals

Connected by Heterogeneous, Multi-Level Networks



Different Security Requirements & Policies Required

Different Resource Management Policies

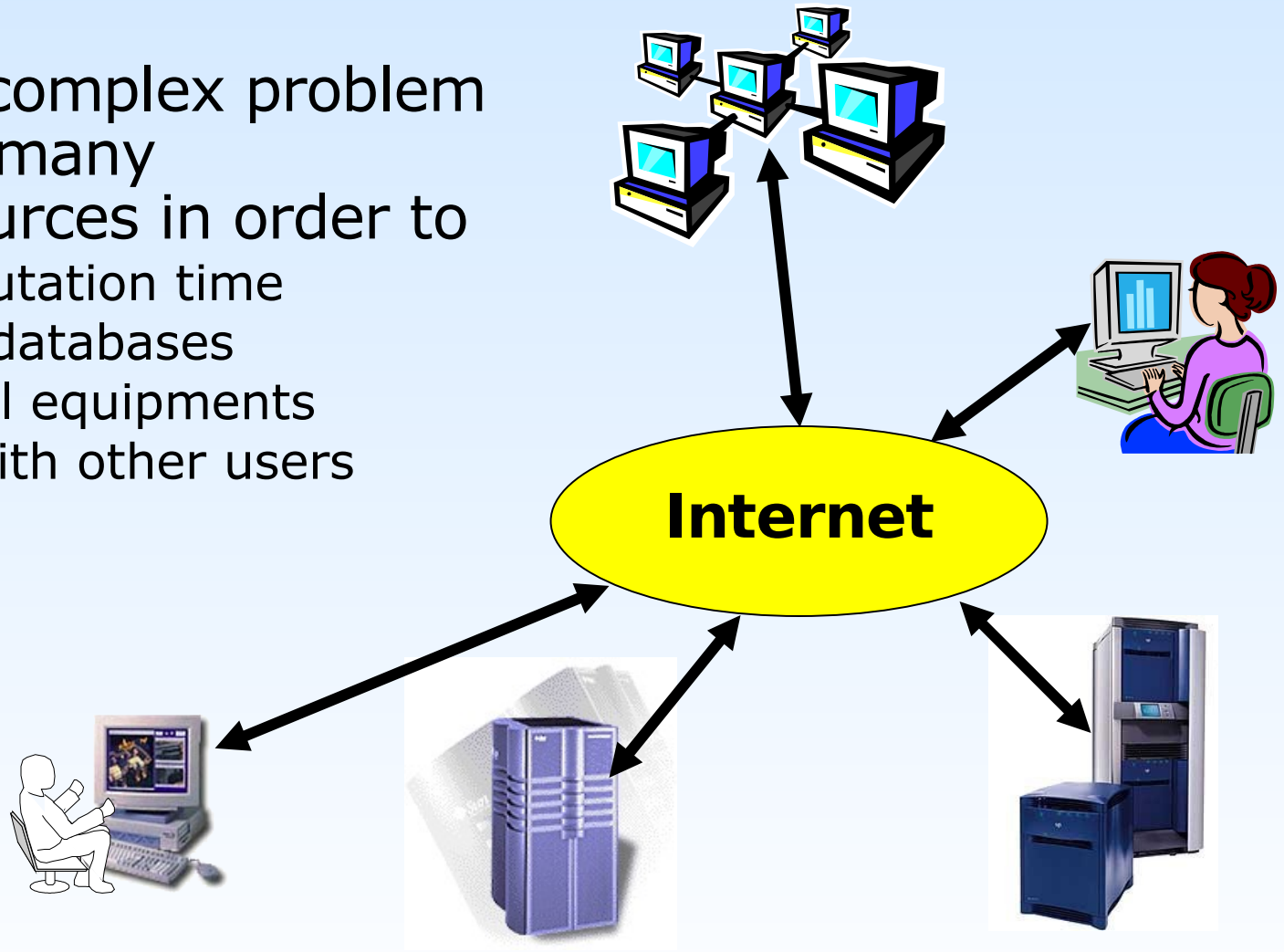
Potentially Faulty Resources

Geographically Separated

Resources are Heterogeneous

Why use a Grid?

- A user has a complex problem that requires many services/resources in order to
  - reduce computation time
  - access large databases
  - access special equipments
  - collaborate with other users

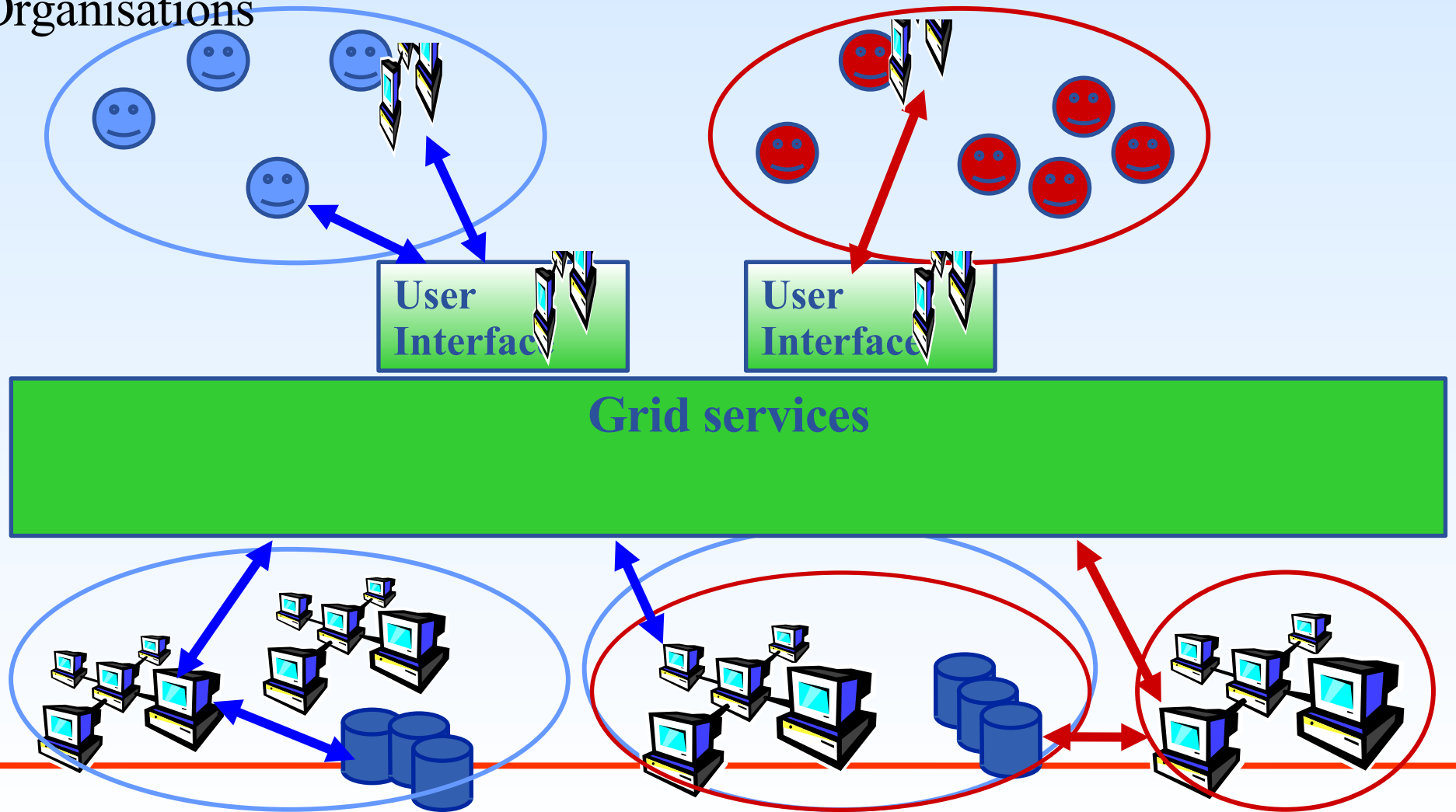


Key concepts

- Virtual Organization: people who collaborate by sharing resources – data, storage, CPU's, programs - across administrative and organizational boundaries
- Single sign-on
  - I connect to one machine – some sort of "digital credential" is passed on to any other resource I use, basis of:
    - *Authentication*: How do I identify myself to a resource without username/password for each resource I use?
    - *Authorization*: what can I do? Determined by
      - My membership of a VO
      - VO negotiations with resource providers
- Grid middleware – "the operating system of a grid"
  - on each resource
  - services that enable the grid
- User just perceives "shared resources" with no concern for location or owning organisation

A multi Virtual Organisation Grid

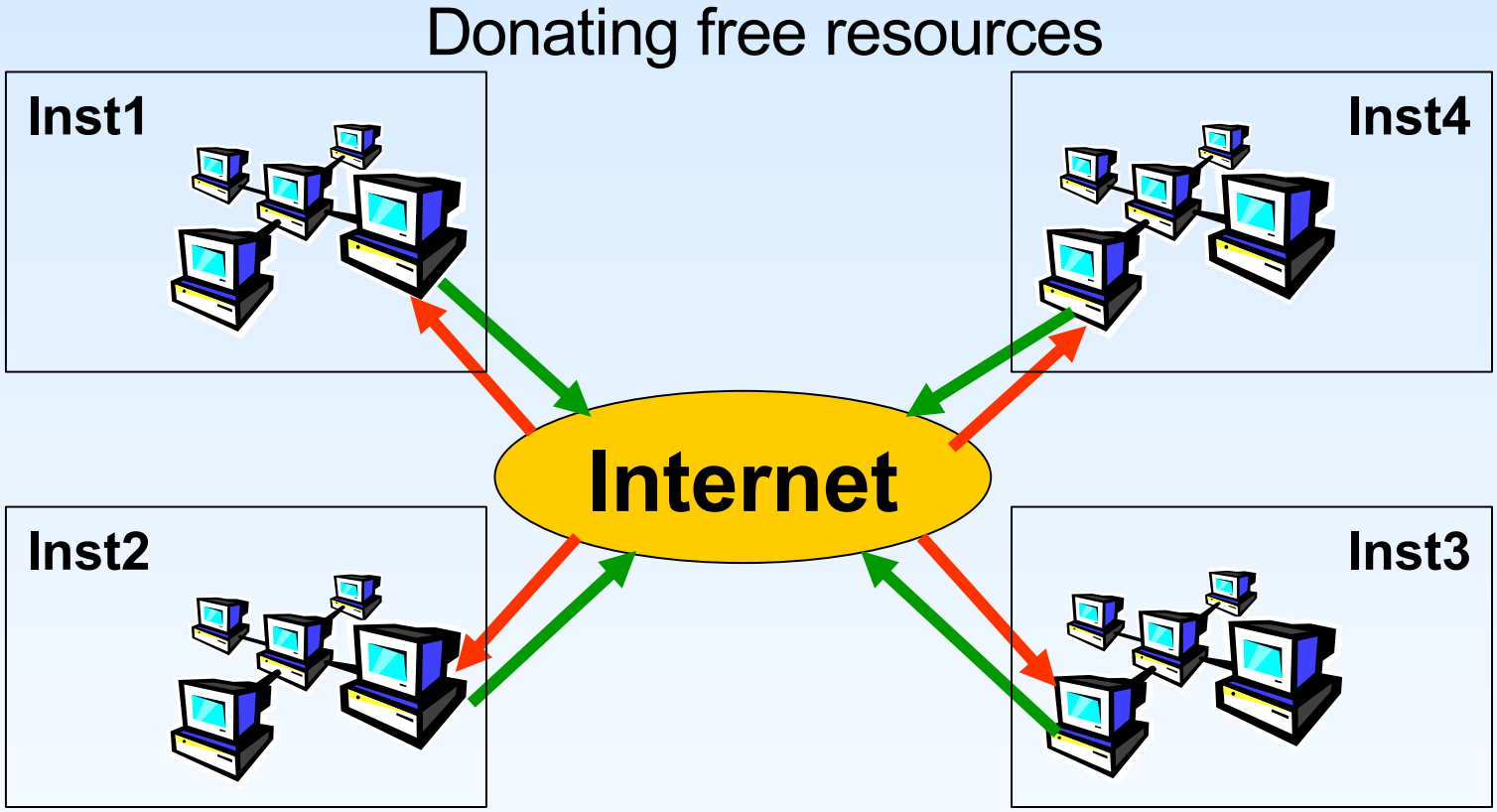
- Grid infrastructure should support multiple, diverse Virtual Organisations



Typical Grid application areas

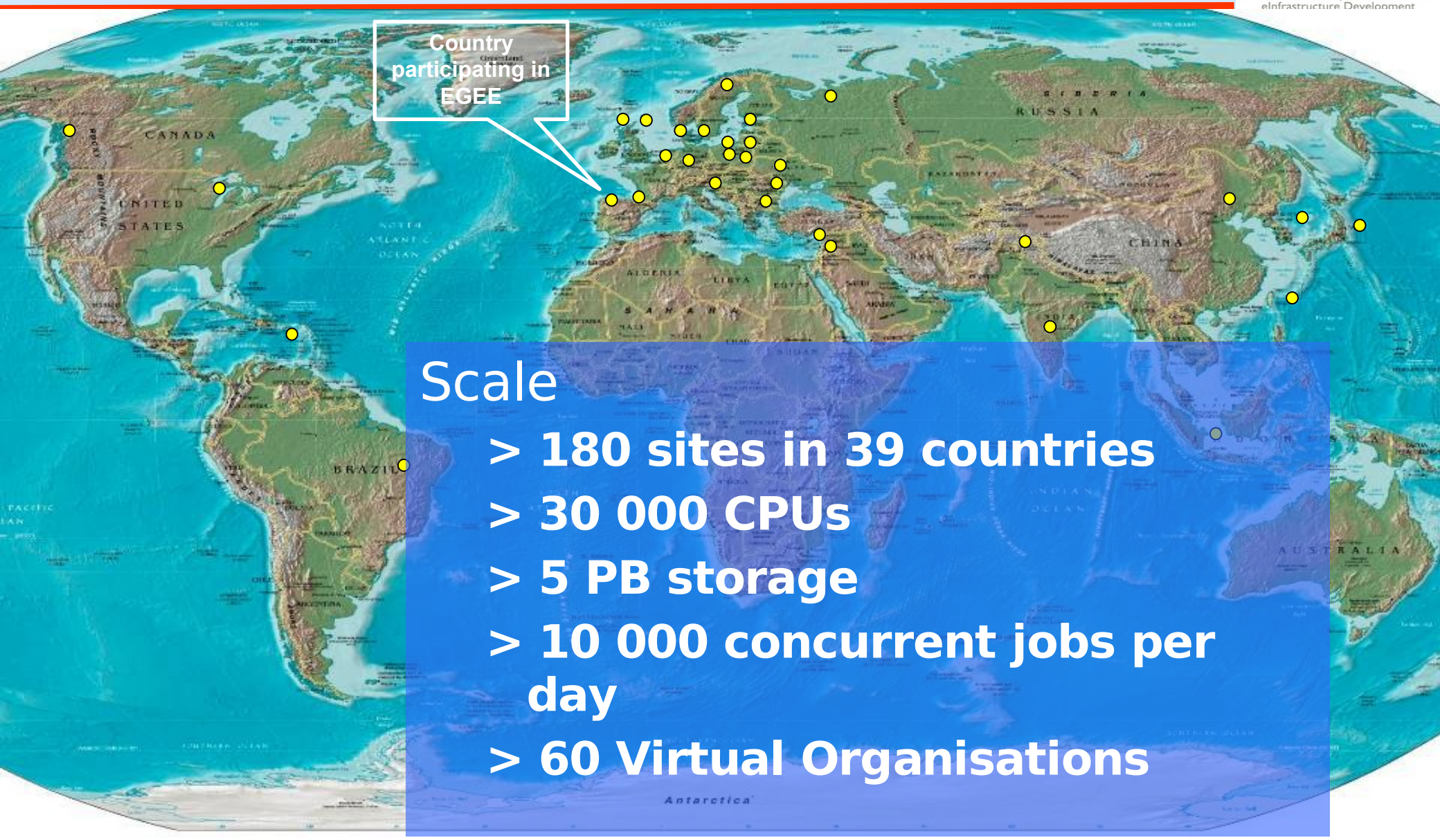
- **High-performance computing (HPC)**
  - to achieve **higher performance** than individual supercomputers/clusters can provide
  - Reguirement: **parallel computing**
- **High-throughput computing (HTC)**
  - To exploit the **spare cycles** of various computers connected by wide area networks
- **Collaborative work**
  - Several users can jointly and remotely solve complex problems

Generic Grid model



The largest production Grid: EGEE

SEE-GRID
South Eastern European GRid-enabled eInfrastructure Development



Scale

- > 180 sites in 39 countries
- > 30 000 CPUs
- > 5 PB storage
- > 10 000 concurrent jobs per day
- > 60 Virtual Organisations

Job Management

- The user interacts with Grid via a **Workload Management System (WMS)**
- The Goal of WMS is the **distributed scheduling and resource management in a Grid environment**.
- What does it allow Grid users to do?
  - To submit their jobs
  - To execute them on the "best resources"
    - **The WMS tries to optimize the usage of resources**
  - To get information about their status
  - To retrieve their output

The use of jobs for running applications

- Jobs are the way users execute applications on the grid.
- Information to be specified when a job has to be submitted:
  - Job characteristics
  - Job requirements and preferences on the computing resources
    - Also including software dependencies
  - Job data requirements
- Information specified using a Job Description Language (JDL)
  - Based upon Condor's *CLASSified ADvertisement language (ClassAd)*
    - Fully extensible language

JDL File

- JDL attributes are grouped in two categories:
  - **Job Attributes**
    - Define the job itself
  - **Resources**
    - Taken into account by the RB for carrying out the matchmaking algorithm (to choose the "best" resource where to submit the job)
    - *Computing Resource*
      - Used to build expressions of Requirements and/or Rank attributes by the user
      - Have to be prefixed with "other."
    - *Data and Storage resources (see talk Job Services With Data Requirements)*
      - Input data to process, SE where to store output data, protocols spoken by application when accessing SEs

Job Description Language: relevant attributes

- **JobType**
  - *Normal* (simple, sequential job), *Interactive*, *MPICH*, *Checkpointable*
  - Or combination of them
- **Executable** (mandatory)
  - The command name
- **Arguments** (optional)
  - Job command line arguments
- **StdInput**, **StdOutput**, **StdError** (optional)
  - Standard input/output/error of the job
- **Environment (optional)**
  - List of environment settings
- **InputSandbox** (optional)
  - List of files on the UI local disk needed by the job for running
  - The listed files will automatically staged to the remote resource
- **OutputSandbox** (optional)
  - List of files, generated by the job, which have to be retrieved
- **VirtualOrganisation** (optional)

JDL File

- **At least one has to specify the following attributes:**
  - the name of the executable
  - the files where to write the standard output and standard error of the job (recommended, not mandatory)
  - the arguments to the executable, if needed
  - the files that must be transferred from UI to WN and viceversa

```
[
Executable = "ls -al";
StdError = "stderr.log";
StdOutput = "stdout.log";
OutputSandbox = {"stderr.log", "stdout.log"};
```

Job Submission

- **glite-job-submit [-r <res_id>] [-vo <VO>] [-o <output file>] <job.jdl>**
  - -r the job is submitted directly to the computing element identified by <*res_id*>
  - -vo the Virtual Organisation (if user is not happy with the one specified in the UI configuration file)
  - -o the generated jobId is written in the <*output file*>

    Useful for other commands, e.g.:

    glite-job-status -i <*input file*> (or jobId)

    -i the status information about jobId contained in the <*input file*> are displayed

Possible job states

Flag	Meaning
SUBMITTED	submission logged in the LB
WAIT	job match making for resources
READY	job being sent to executing CE
SCHEDULED	job scheduled in the CE queue manager
RUNNING	job executing on a WN of the selected CE queue
DONE	job terminated without grid errors
CLEARED	job output retrieved
ABORT	job aborted by middleware, check ***reason***

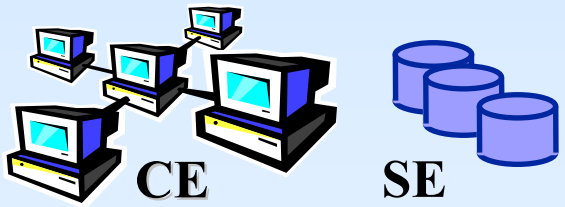
Data Management: general concepts

- What does "Data Management" mean?
  - Users and applications produce and require data
  - Data may be stored in Grid files
  - Granularity is at the "file" level (no data "structures")
  - Users and applications need to handle files on the Grid
- Files are stored in appropriate permanent resources called "Storage Elements" (SE)
  - Present almost at every site together with computing resources
  - We will treat a storage element as a "black box" where we can store data
    - Appropriate data management utilities/services hide internal structure of SE
    - Appropriate data management utilities/services hide details on transfer protocols

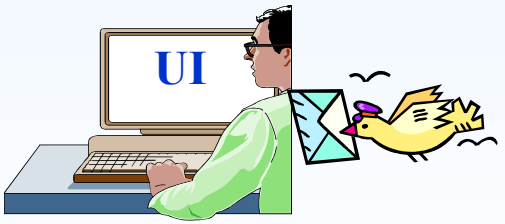
Data Management operations

Upload a file to the grid

- Users need to store data in SE (from a UI)
- Applications need to store data in SE (from a CE)
- Users need to store the application (to be retrieved and run on a CE)
- For small files the InputSandbox can be used



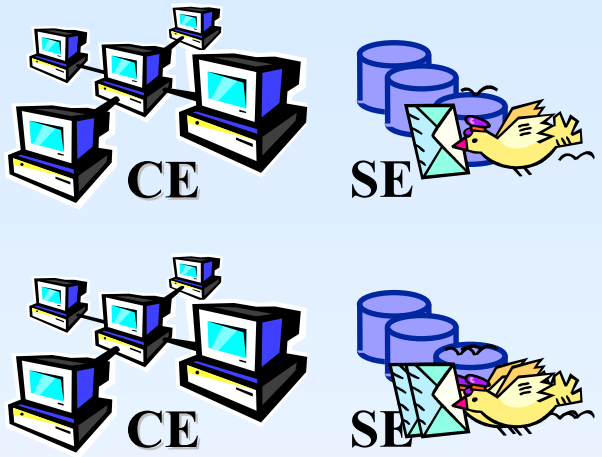
Several Grid Components



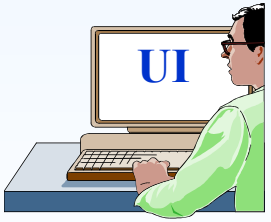
Data Management operations

Download files from the grid

- User need to retrieve (onto the UI) data stored into SE
  - For small files produced in WN the OutputSandbox can be used
- Applications need to copy data locally (into the CE) and use them
- The application itself must be downloaded onto the CE and run



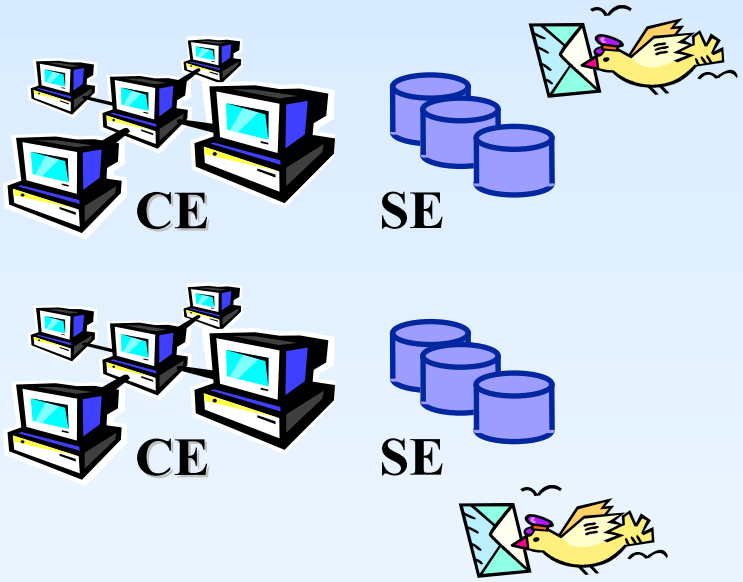
Several Grid Components



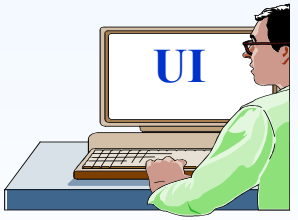
Data Management operations

Replicate a file across different Storage Elements

- Load share balacing of computing resources
  - Often a job needs to run at a site where a copy of input data is present
- Performance improvement in data access
  - Several applications might need to access the same file concurrently
- Important for redundancy of key files (backup)



Several Grid Components



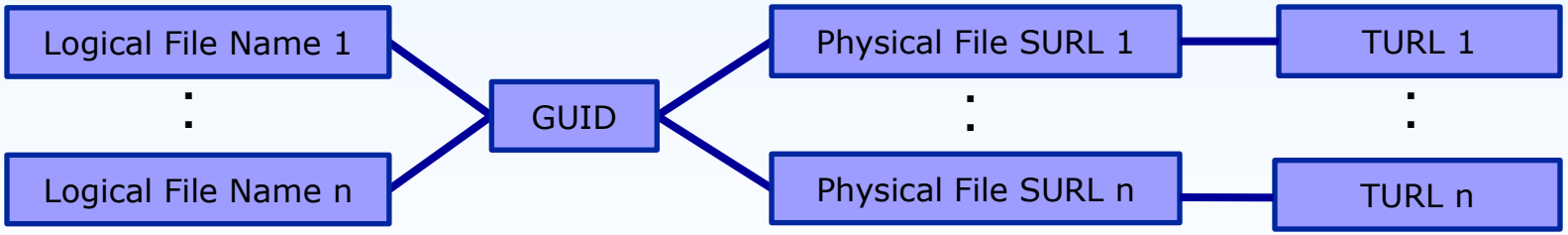
Files & replicas: Name Conventions

- Logical File Name (LFN)
- An alias created by a user to refer to some item of data, e.g. “lfn:cms/20030203/run2/track1”
- Globally Unique Identifier (GUID)
- A non-human-readable unique identifier for an item of data, e.g.

“guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”

- Site URL (SURL) (or Physical File Name (PFN) or Site FN)
- The location of an actual piece of data on a storage system, e.g. “srm://pcrd24.cern.ch/flatfiles/cms/output10_1” (SRM) “sfn://lxshare0209.cern.ch/data/alice/ntuples.dat” (Classic SE)
- Transport URL (TURL)
- Temporary locator of a replica + access protocol: understood by a SE, e.g.

“rfio://lxshare0209.cern.ch//data/alice/ntuples.dat”



File Catalogs

- **Issues:**
  - How do I keep track of all my files on the Grid?
  - Even if I remember all the lfns of my files, what about someone else files?
  - Anyway, how does the Grid keep track of associations lfn/GUID/surl?
- **Solution: FILE CATALOGUE**
- **Need to keep track of the location of copies (replicas) of Grid files**
- Replicas might be described by attributes
  - Support for METADATA
  - Could be "system" metadata or "user" metadata
- Potentially, milions of files need to be registered and located
  - Requirement for performance
- Distributed architecture might be desirable
  - scalability
  - prevent single-point of failure
  - Site managers need to change autonomously file locations

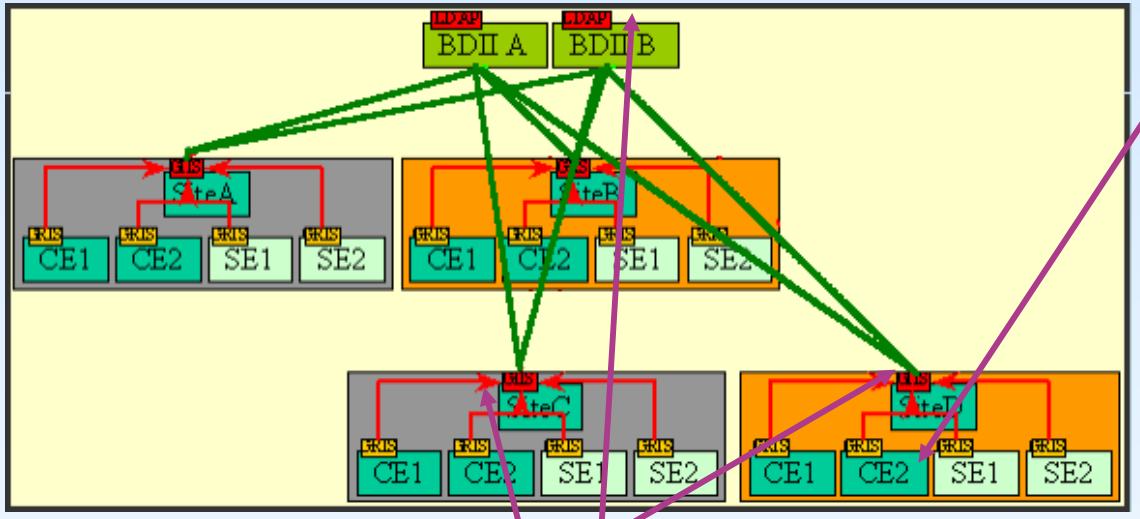
Information system

- The Information System (IS) provides information about the Grid resources and their status.
- The resources are hardware(CPU, Memory, Disk), software (Applications, services), storage, network etc.
- Both the UI (users) and other services (e.g. RB) need the IS.
- Computing and storage resources at a site implement an entity called Information Provider, which generates the relevant information of the resource (e.g.: the used space in a SE).
- In each site an element called the Site Grid Index Information Server (GIIS) collects all the information of the different providers and publishes it.

The Information Sytem

- Two main Information System technologies used in EGEE are: one LDAP based from Globus and one developed by the European DataGrid Project: R-GMA
- The LDAP based information system is based on Globus Monitoring and Discovery Service (MDS)
- In LCG-2, the Berkeley DB Information Index (BDII), based on an updated version of the Monitoring and Discovery Service (MDS), from Globus, was adopted as main provider of the Information Service.
- Relational Grid Monitoring Architecture (R-GMA) is also adopted in both the current LCG middleware (“LCG-2”) and in the new EGEE middleware (“gLite 3.0”) to which the production grid is currently transitioning

Information System architecture



- **Local GRISes** run on CEs and SEs at each site and report dynamic and static information regarding the status and availability of the services

```
ldapsearch -x -h
<hostname> -p 2135 -b
"mds-vo-name=local,o=grid"
```

- At each site, a site GIIS or site BDII collects the information of all resources given by the GRISs

```
ldapsearch -x -h <hostname> -p 2135 -b "mds-vo-name=<name>,o=grid"
ldapsearch -x -h <hostname> -p 2170 -b "mds-vo-name=<name>,o=grid"
```

- Each site can run a top level BDII It collects the information coming from the sites and collects it in a data base

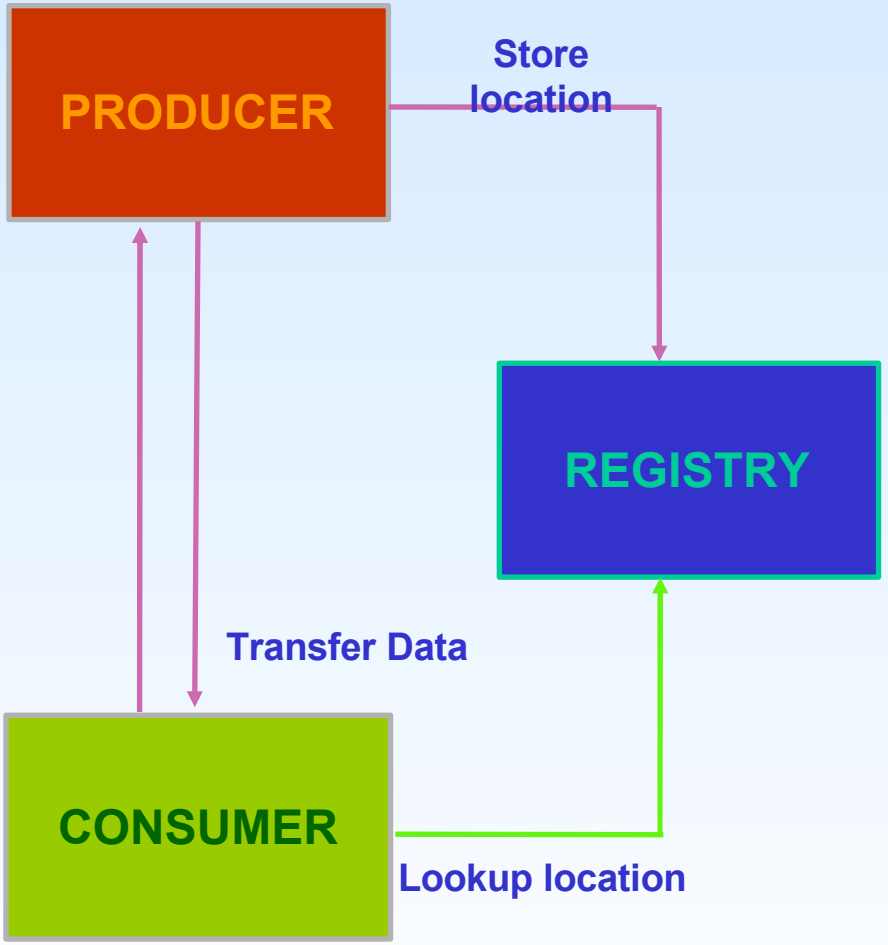
```
ldapsearch -x -h <hostname> -p 2170 -b "o=grid"
```

Introduction to R-GMA

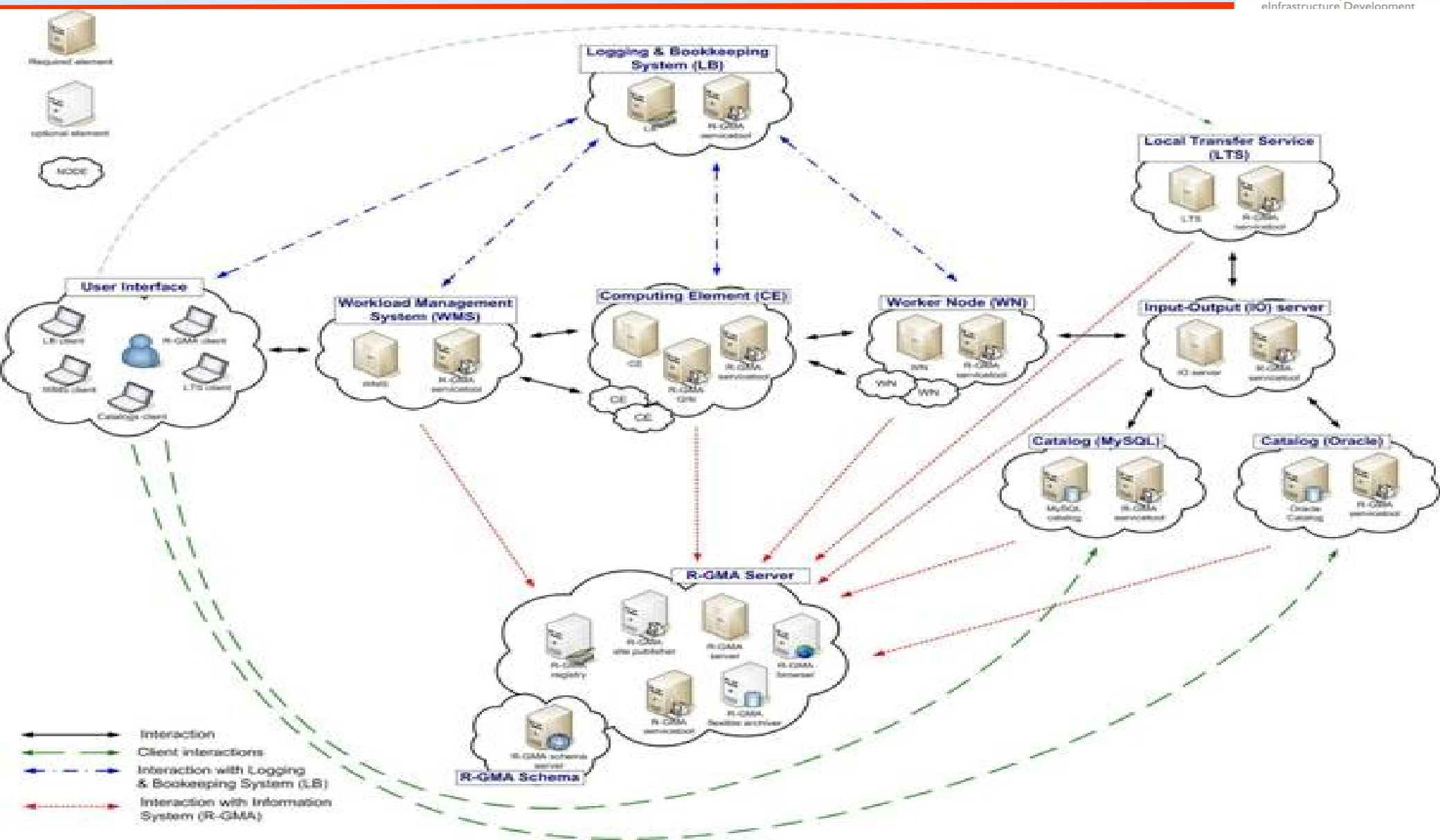
- Relational Grid Monitoring Architecture (R-GMA)
  - Developed as part of the EuropeanDataGrid Project (EDG)
  - Now as part of the EGEE project.
  - Based the Grid Monitoring Architecture (GMA) from the Global Grid Forum (GGF).
- Uses a relational data model.
  - Data are viewed as tables.
  - Data structure defined by the columns.
  - Each entry is a row (tuple).
  - Queried using Structured Query Language (SQL).

R-GMA Basics

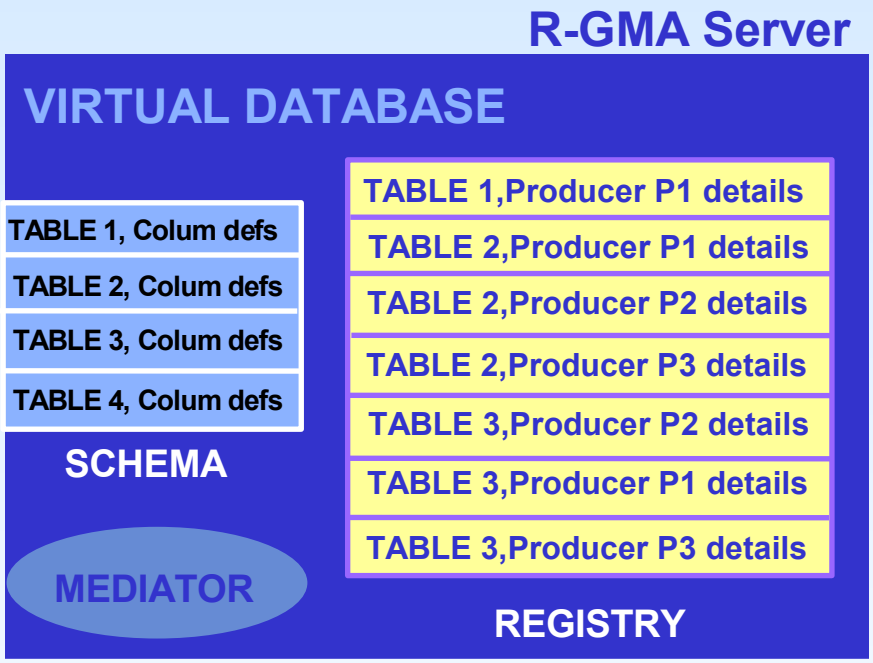
- The Producer stores its location (URL) in the Registry.
- The Consumer looks up producer URLs in the Registry.
- The Consumer contacts the Producer to get all the data or the Consumer can listen to the Producer for new data.



R-GMA within Testbed



R-GMA: Schema-Registry-Mediator



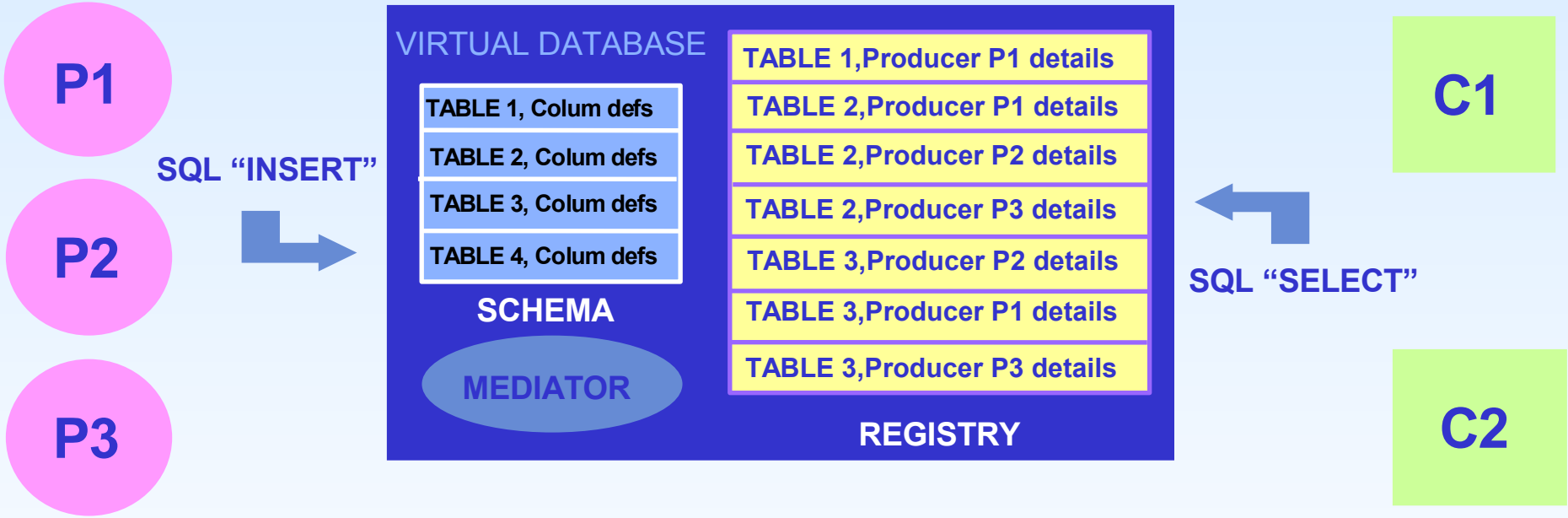
SCHEMA : it holds the names and definitions of all of the tables in the virtual database, and their authorization rules.

REGISTRY: It holds the details of all producers that are publishing to tables in the virtual database and it also holds the details of “continuous” consumers.

MEDIATOR: a set of rules for deciding which data providers to contact for any given query.

R-GMA: Producer-Consumer

Producers: are the data providers for the virtual database. Writing data into the virtual database is known as publishing, and data is always published in complete rows, known as tuples. There are three types of producer: Primary, Secondary and On-demand.

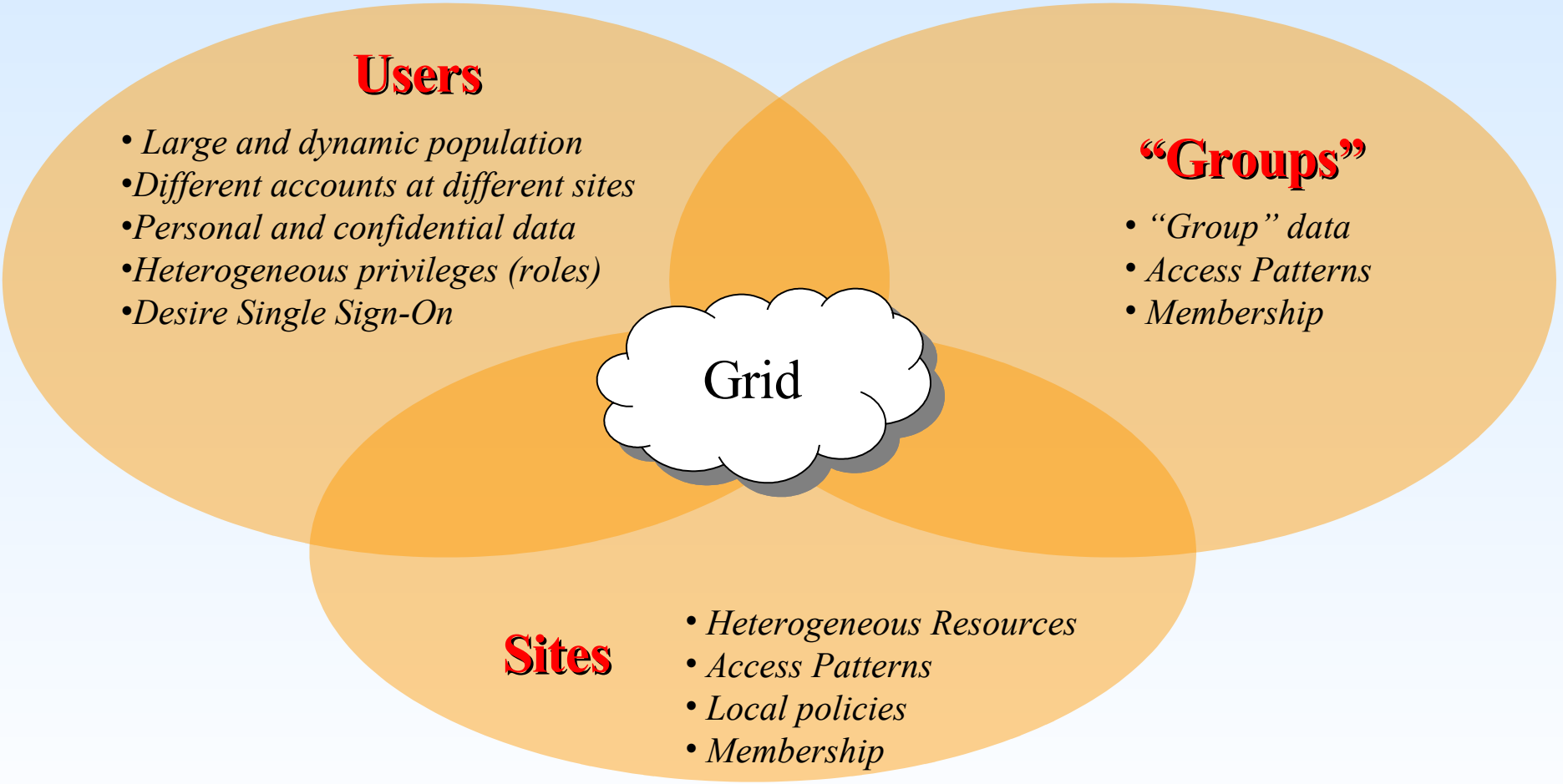


Consumer: represents a single SQL SELECT query on the virtual database. The query is matched against the list of available producers in the Registry. The consumer service then selects the best set of producers to contact and sends the query directly to each of them, to obtain the answer tuples.

R-GMA Usage

- Consumer users: who requests information
- Producer users: who provides information
- Site administrators: who runs R-GMA services
- Virtual Organizations: who “owns” the schema and registry
- Mutual Autentication: guaranteeing who is at each end of an exchange of messages
- Encryption: using an encrypted transport protocol (HTTPS)
- Authorization: implicit or explicit

Security in Grid



Security in Grid

- Distribution of resources: secure access is a basic requirement
  - secure communication
  - security across organisational boundaries
  - single “sign-on" for users of the Grid
- Two basic concepts:
  - Authentication: *Who am I?*
    - “Equivalent” to a pass port, ID card etc.

  - Authorisation: *What can I do?*
    - Certain permissions, duties etc.

X.509 Certificates

- *Certification Authority* (CA) issues Digital Certificates for users, programs and machines
- Check the identity and the personal data of the requestor
  - Registration Authorities (RAs) do the actual validation
- CA's periodically publish a list of compromised certificates
  - **Certificate Revocation Lists** (CRL): contain all the revoked certificates yet to expire
- CA certificates are self-signed

Questions

Thank you!

Questions & Open Discussion