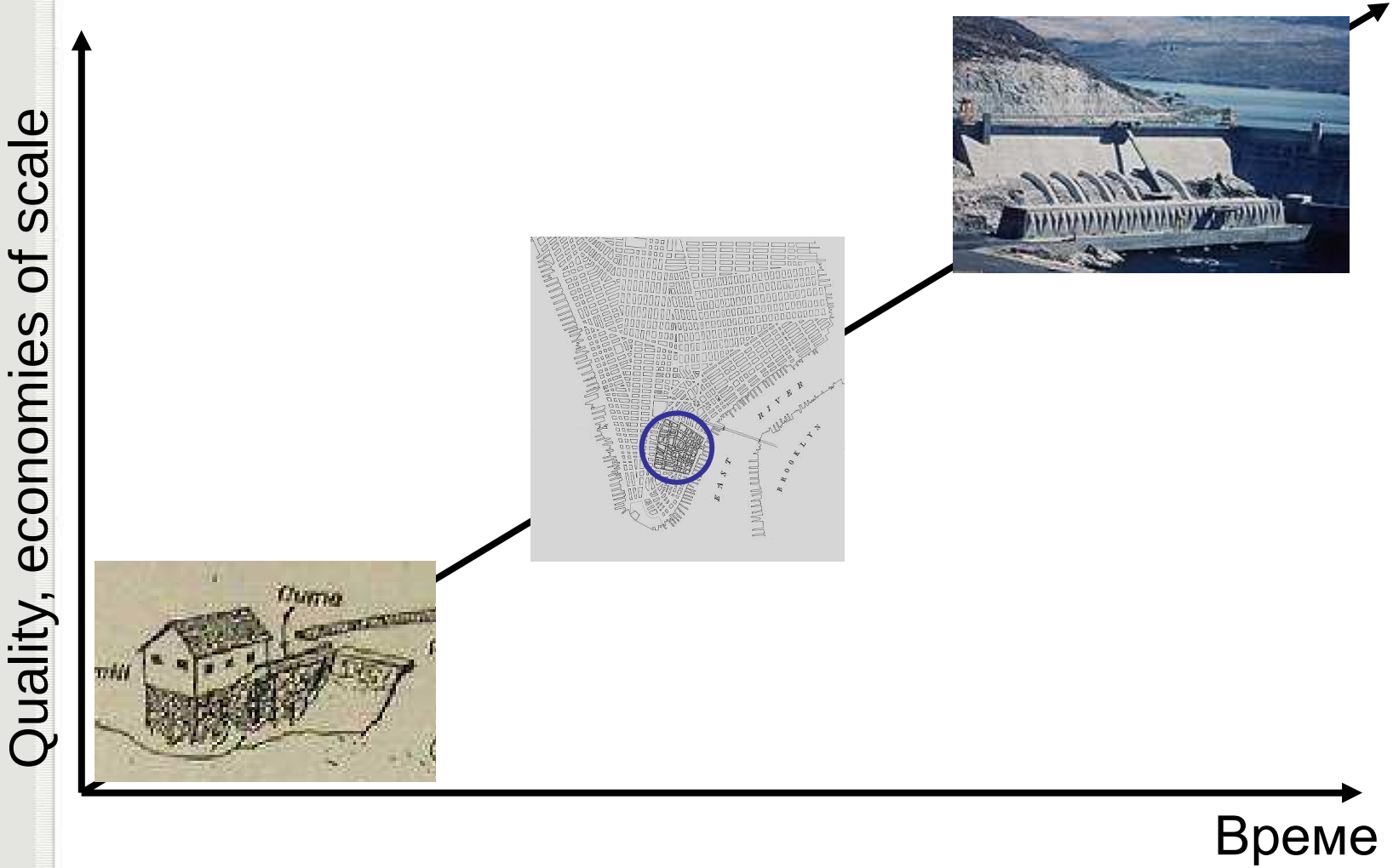
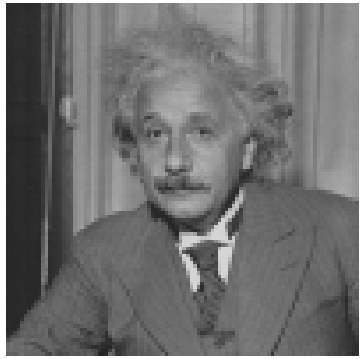
GRID Computing

Увод

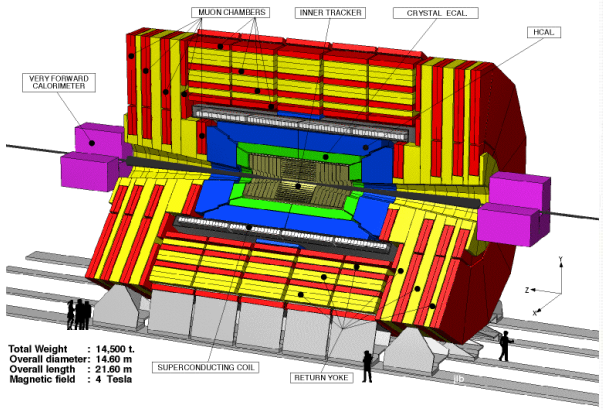


Развој е-науке

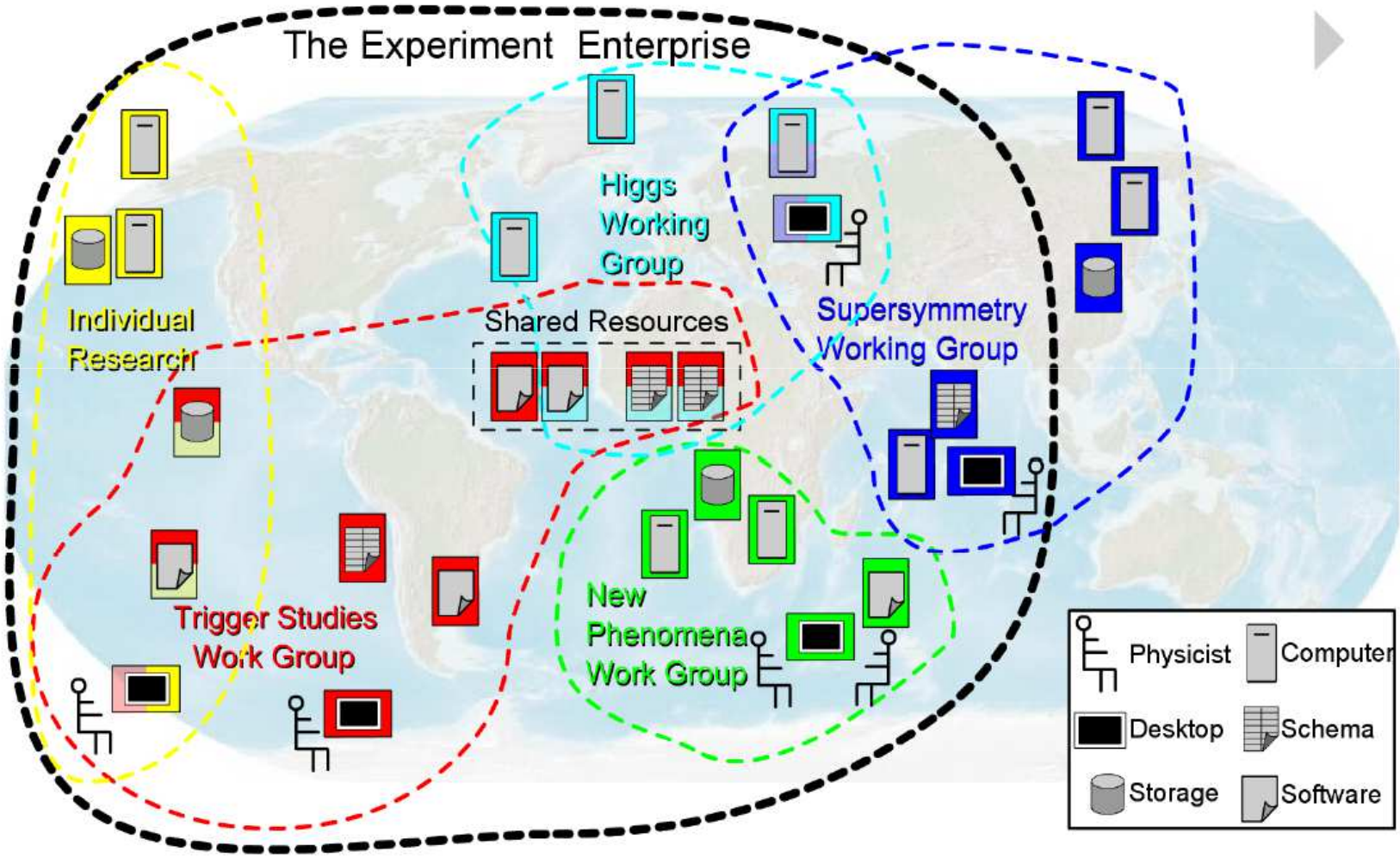
- Пре појаве Интернета
  - Теоријски приступ и/или експерименти који се изводе самостално или у малим групама; објављивање радова



- Након појаве Интернета
  - Креирање и истраживање великих база података добијених посматрањем или симулацијама
  - Развој симулација и њихова анализа
  - Удаљен приступ специјализованим инструментима
  - Размена информација међу удаљеним мултидисциплинарним тимовима

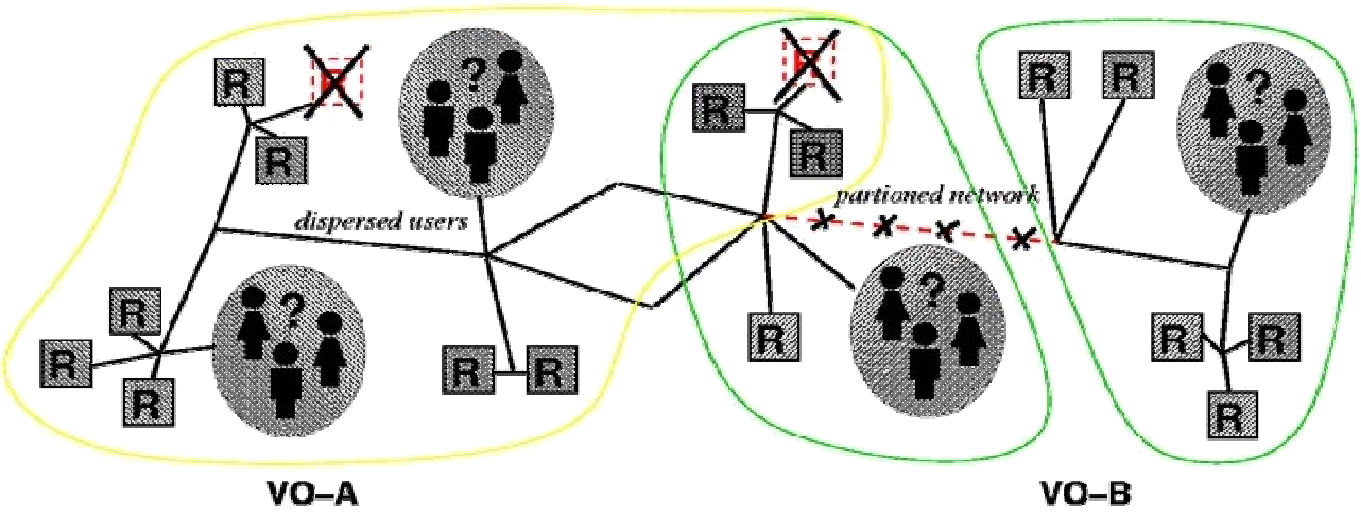


Постојећи системи



Заједнички концепти

- **Дељење ресурса и координирано решавање проблема у динамичким, мулти-институционалним виртуелним организацијама**



Пример

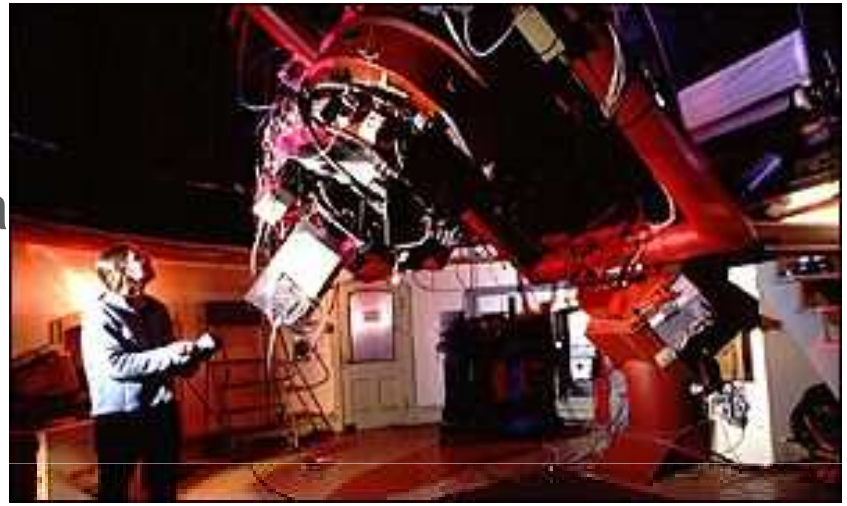
- Грид на суперрачунарима
  - UK e-Science Project, CERN
    - IBM’s Globus (www.globus.org)
- Грид на личним рачунарима
  - SETI
  - Climate*prediction*
  - Screensaver Lifesaver

CERN

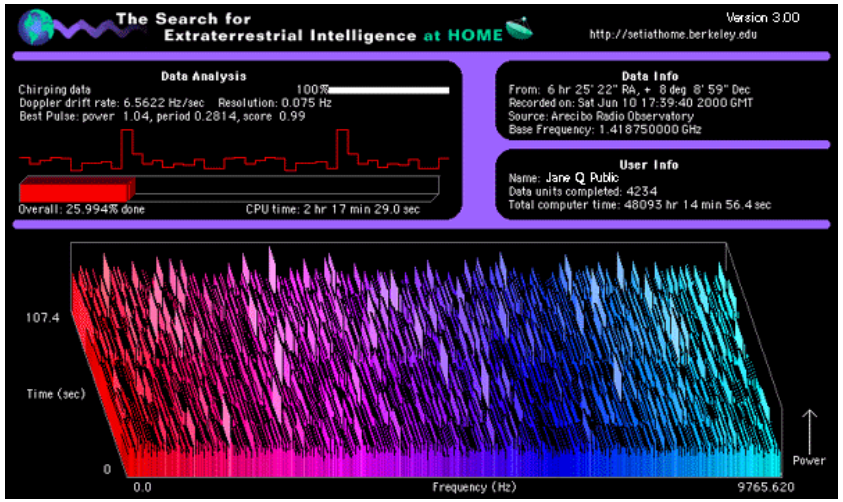
- Computing net promises vast power (BBC) Sept 2003
- Грид технологија искоришћена за повезивање суперрачунара у свету преко Интернета да би се створила нова генерација изузетно моћних рачунара

SETI@home

- Search for Extraterrestrial Intelligence (Потрага за ванземаљском интелигенцијом)



- Програм који анализира податке примљене са радио телескопа

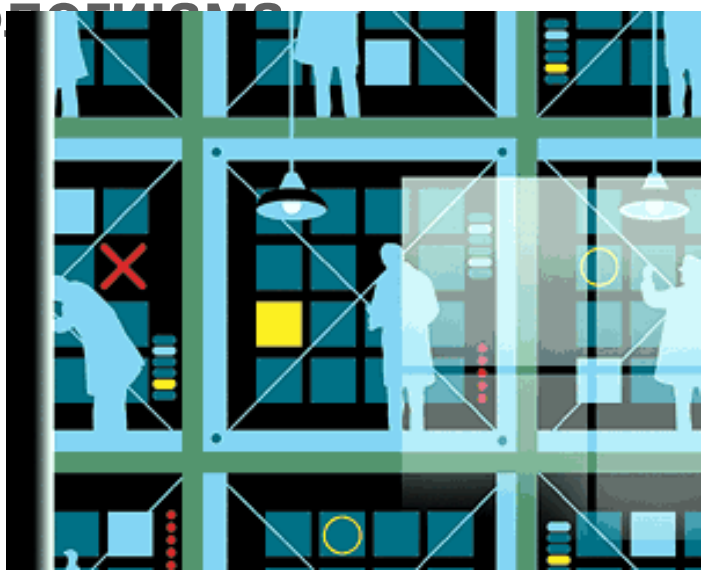


Шта је то Grid Computing

- Grid computing harnesses the massive processing power made possible by distributed computing and applies it to business problems.
- It does this through virtualising computers, storage and networking resources such that a grid appears as one large computer.
- A grid computing infrastructure is capable of managing large peaks of work; managing large amounts of processing and automatically adjusting capacity to meet workload.
- When this infrastructure is combined with business systems that allow users to describe their processes, substantial benefits can be obtained.

Шта су користи од Grid Computing

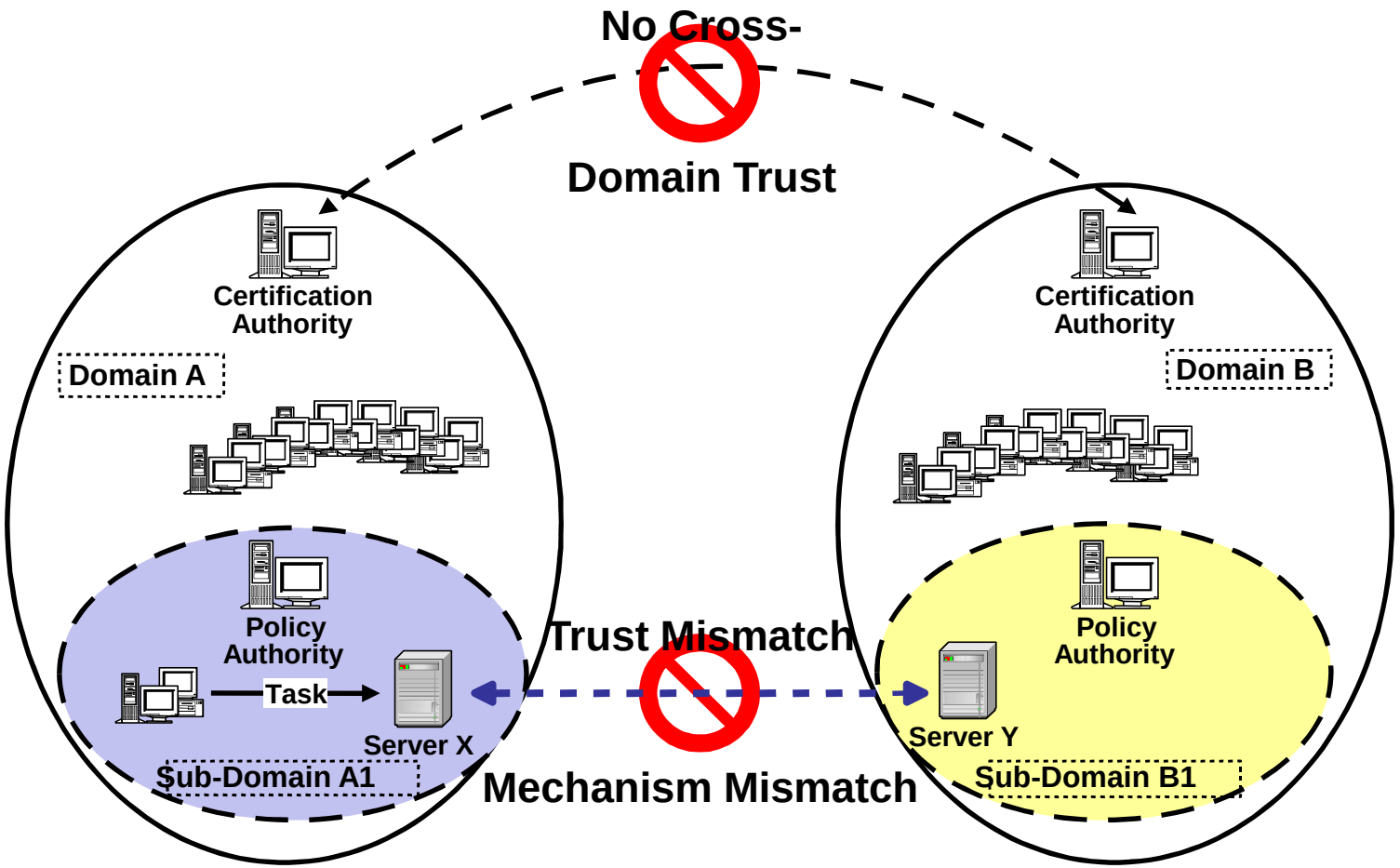
- Improved Return on Assets
- Повећање скалабилности (могућност проширења)
- Business Flexibility & Agility
- Управљање комплексним информационим технологијама



Сервиси Грида

- Покретање посла/надгледање
- Трансфер датотека (individual, queued)
- Приступ базама података
- Data management (replication, metadata)
- Monitoring/Indexing system information

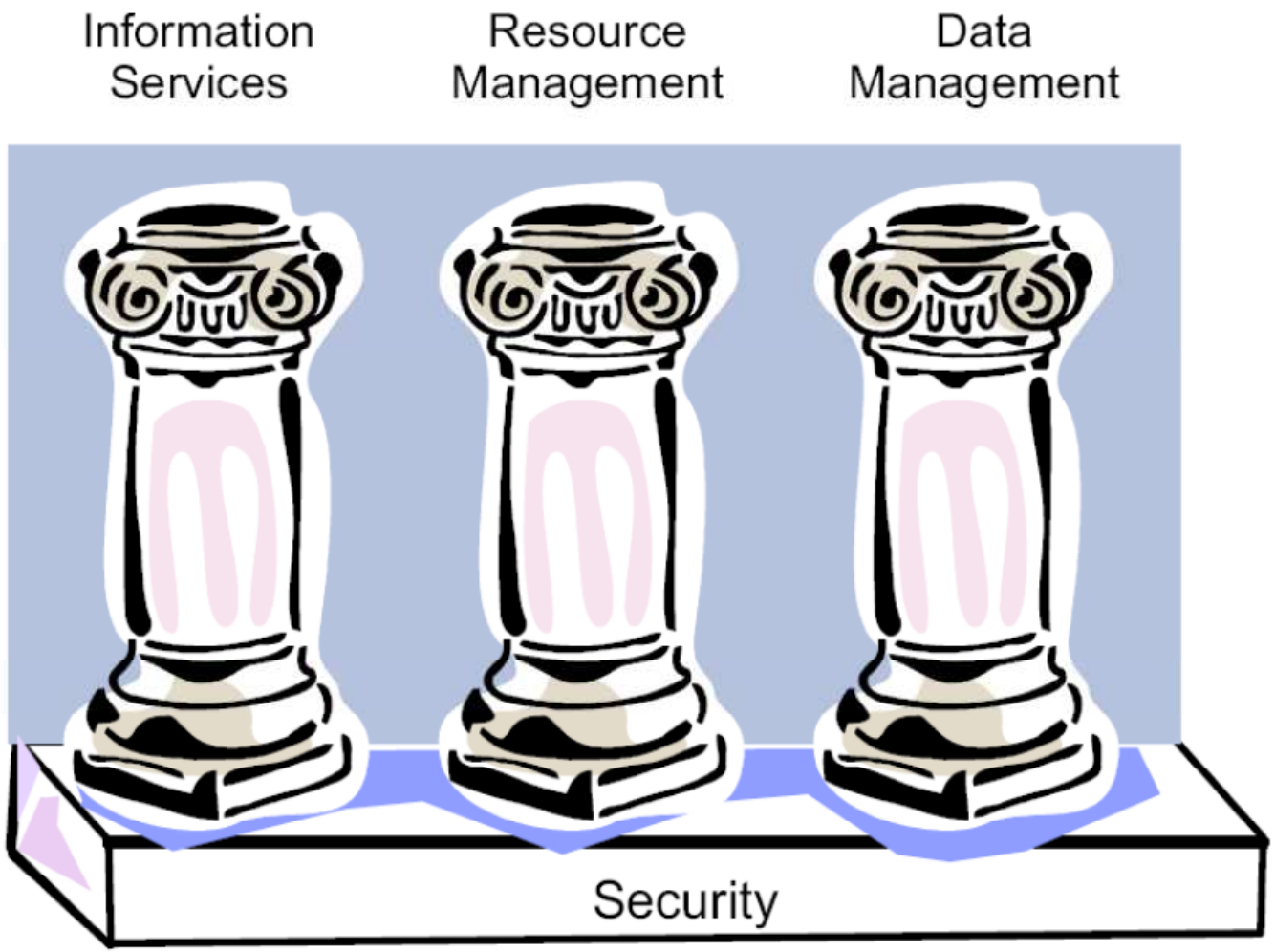
Међуповезаност институција



Виртуална организација

- Скупови организација које користе Грид да би наменски делиле ресурсе за специјалне потребе
- Подржавање јединствене заједнице
- Развој компатибилних технологија и договарање о полисама
  - Сигурносне полисе
- Развој различитих мрежних сервиса:
  - Grid Information
  - Grid Resource Brokering
  - Grid Monitoring
  - Grid Accounting

Основе Грида



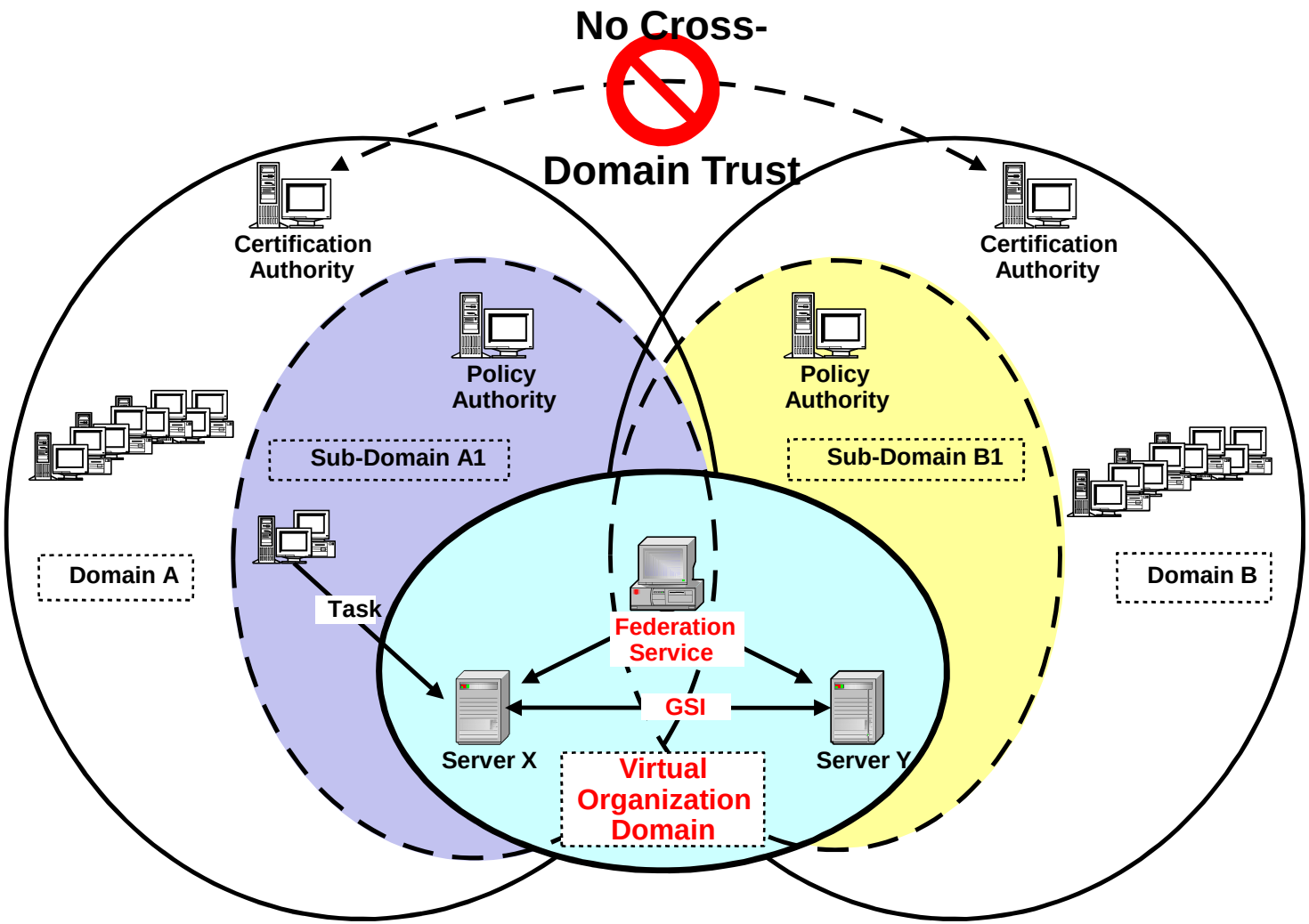
Заштита

- Resources being used may be valuable & the problems being solved sensitive
  - Both users and resources need to be careful
- Dynamic formation and management of virtual organizations
  - Large, dynamic, unpredictable…
- VO Resources and users are often located in distinct administrative domains
  - Can’t assume cross-organizational trust agreements
  - Different mechanisms & credentials

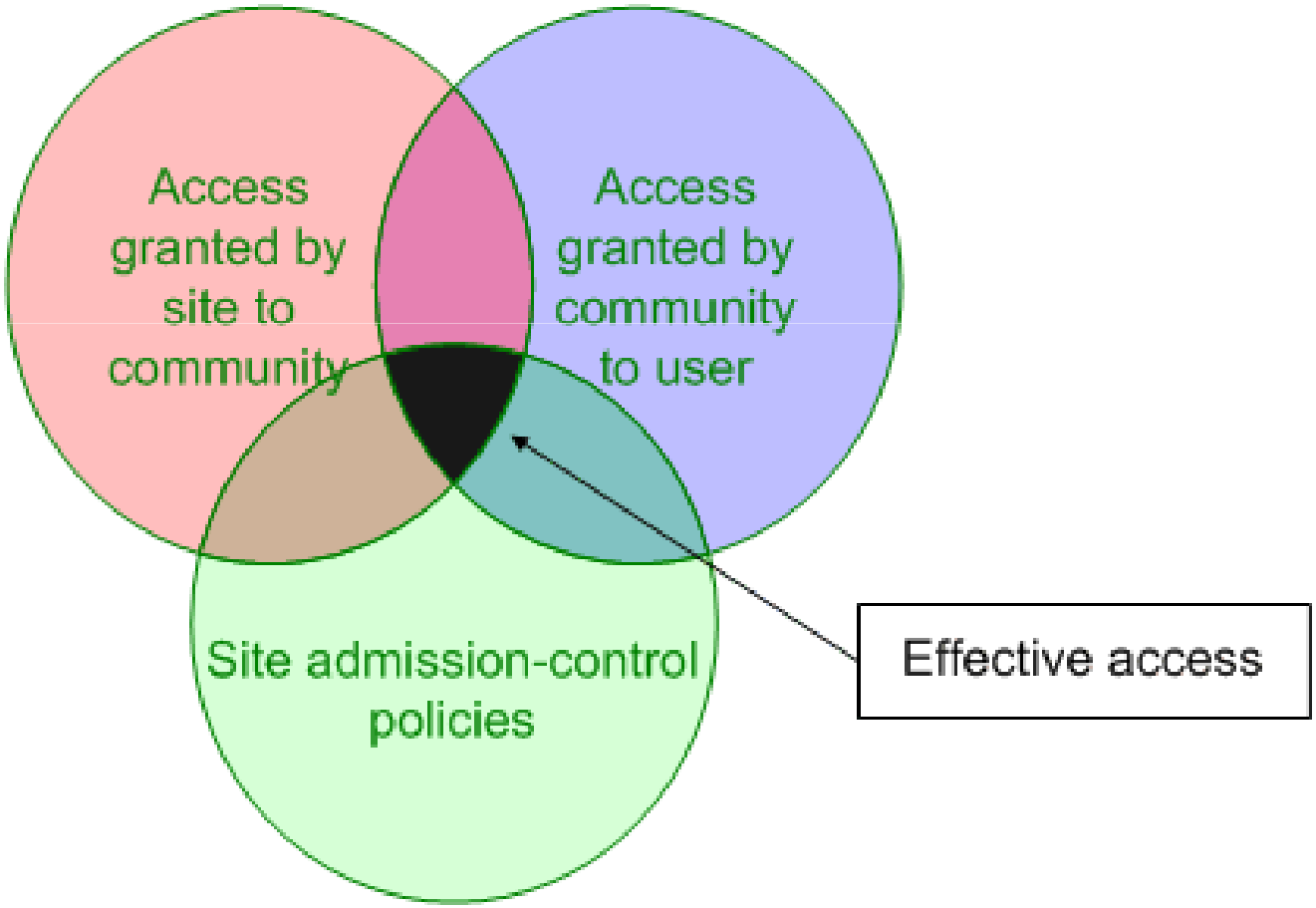
Заштита

- Interactions are not just client/server, but service-to-service on behalf of the user
  - Requires delegation of rights by user to service
  - Services may be dynamically instantiated
- Standardization of interfaces to allow for discovery, negotiation and use
- Implementation must be broadly available & applicable
  - Standard, well-tested, well-understood protocols; integrated with wide variety of tools
- Policy from sites, VO, users need to be combined
  - Varying formats
- Want to hide as much as possible from applications!

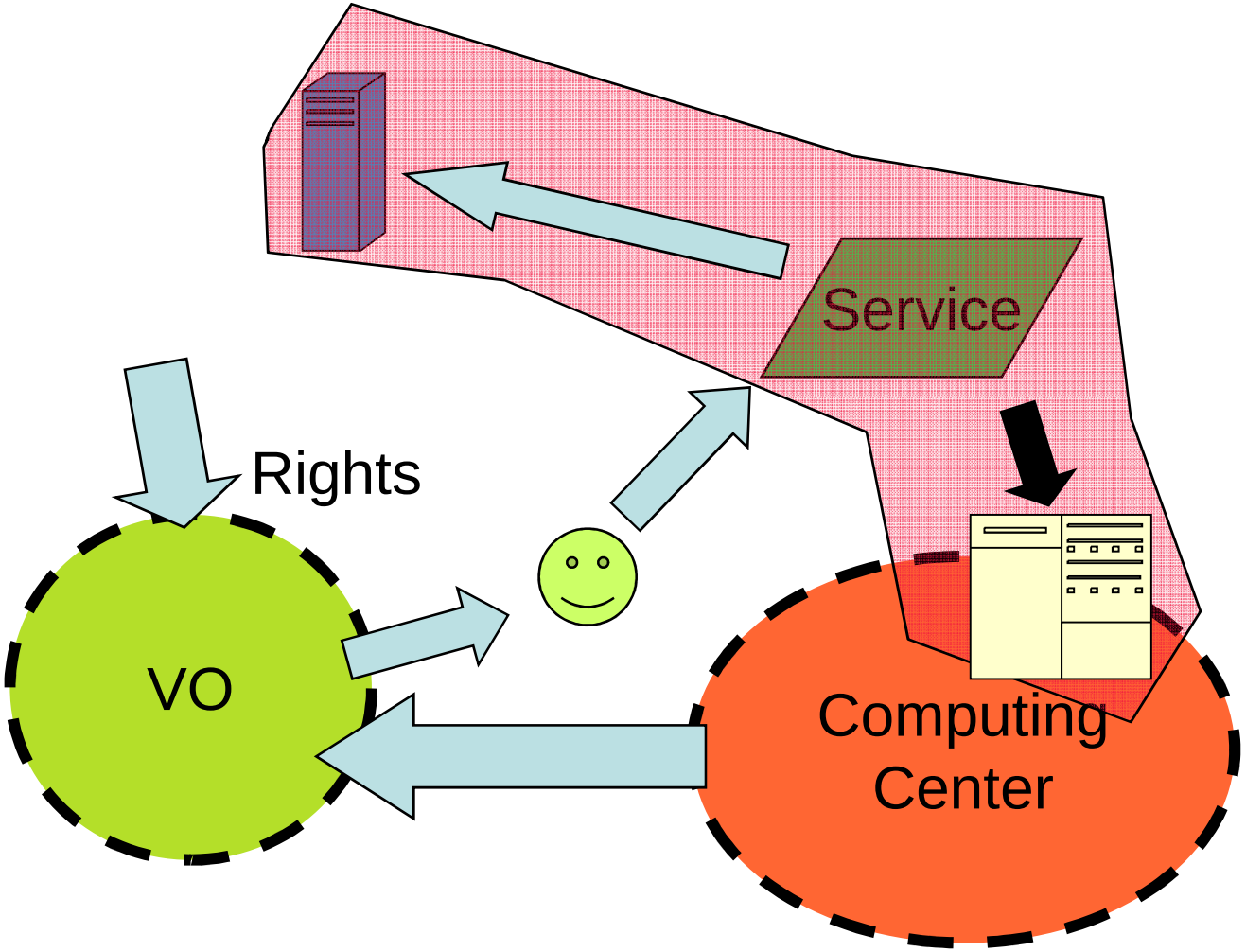
Виртуалне организације



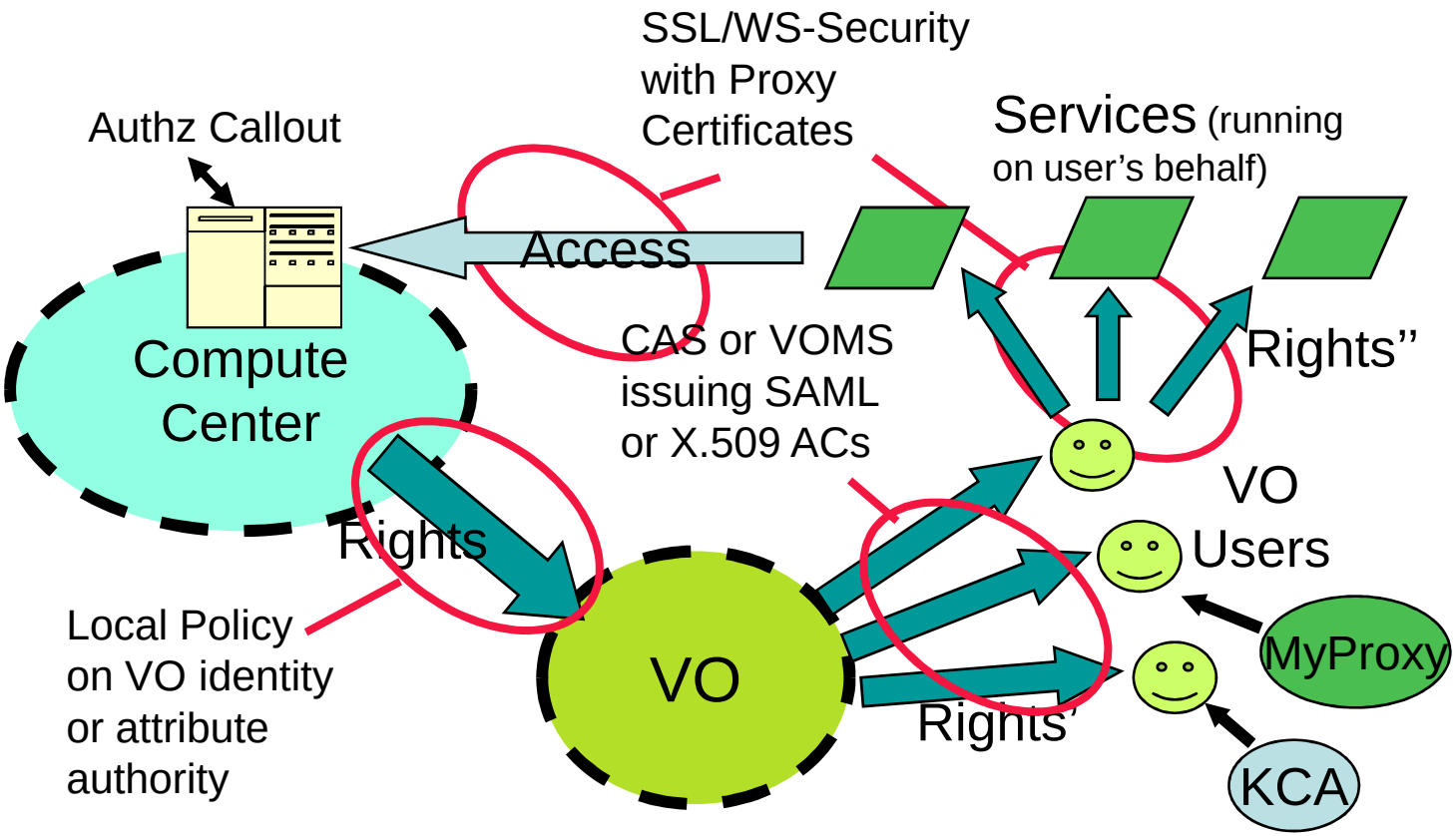
Права приступа



Коришћење делегата



Имплементација GSI



Коришћење Грида

- Покретање програма у Гриду користећи делегата:

  voms-proxy-init –voms VONAME

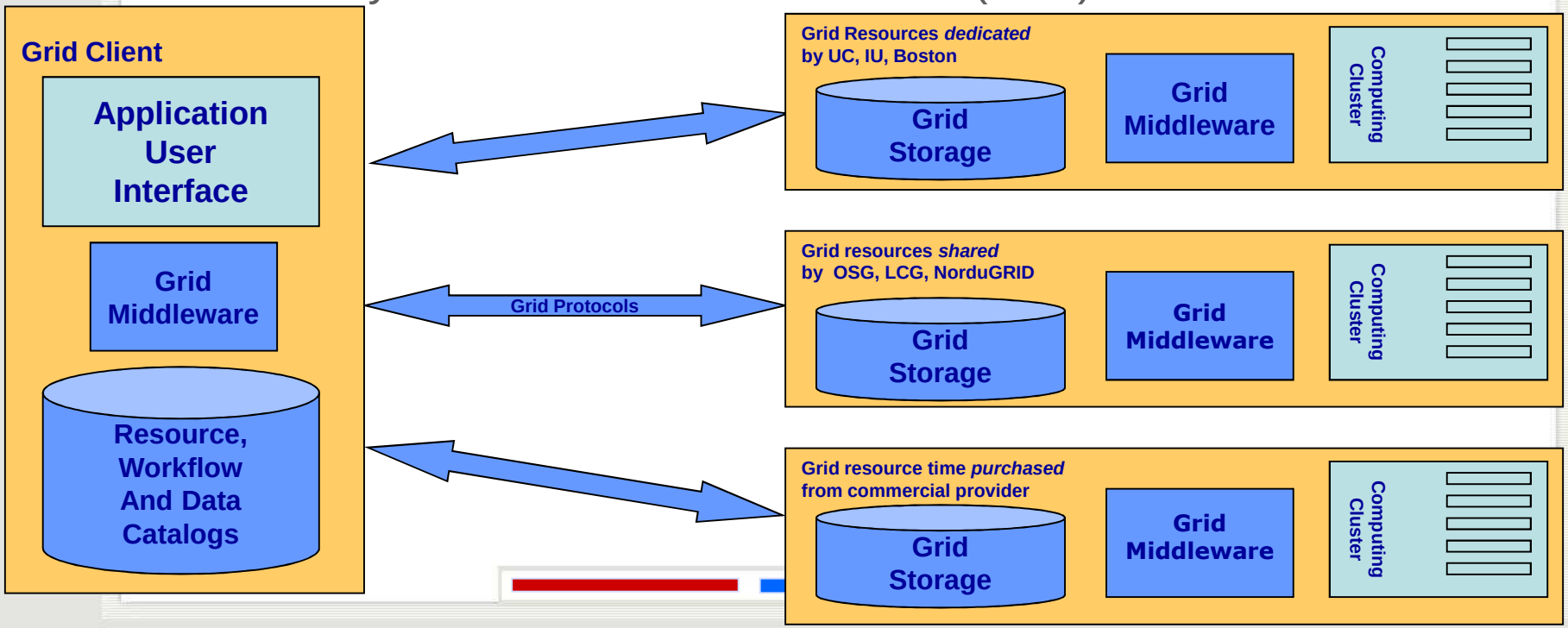
  Enter PEM pass phrase: **********

- Креирање привременог, локалног, делегата (proxy)

- Delegation = remote creation of a (second level) proxy credential, which allows remote process to authenticate on behalf of the user

Од чега се Грид састоји

- Security to control access and protect communication (GSI)
- Directory to locate grid sites and services: (VORS, MDS)
- Uniform interface to computing sites (GRAM)
- Facility to maintain and schedule queues of work (Condor-G)
- Fast and secure data set mover (GridFTP, RFT)
- Directory to track where datasets live (RLS)

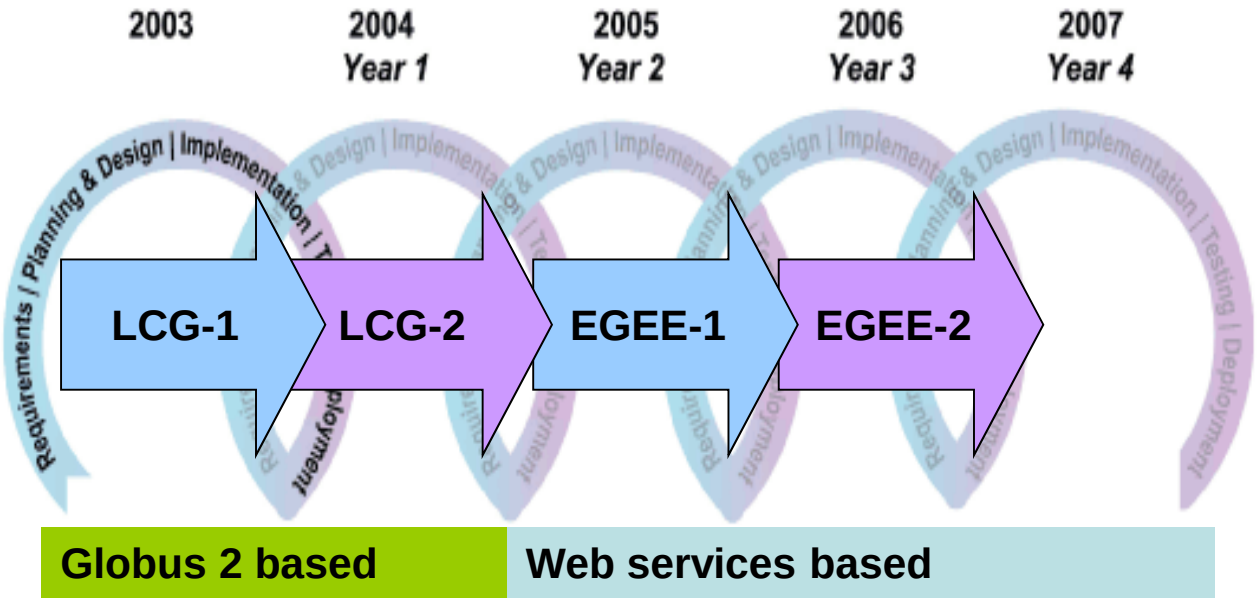


Middleware

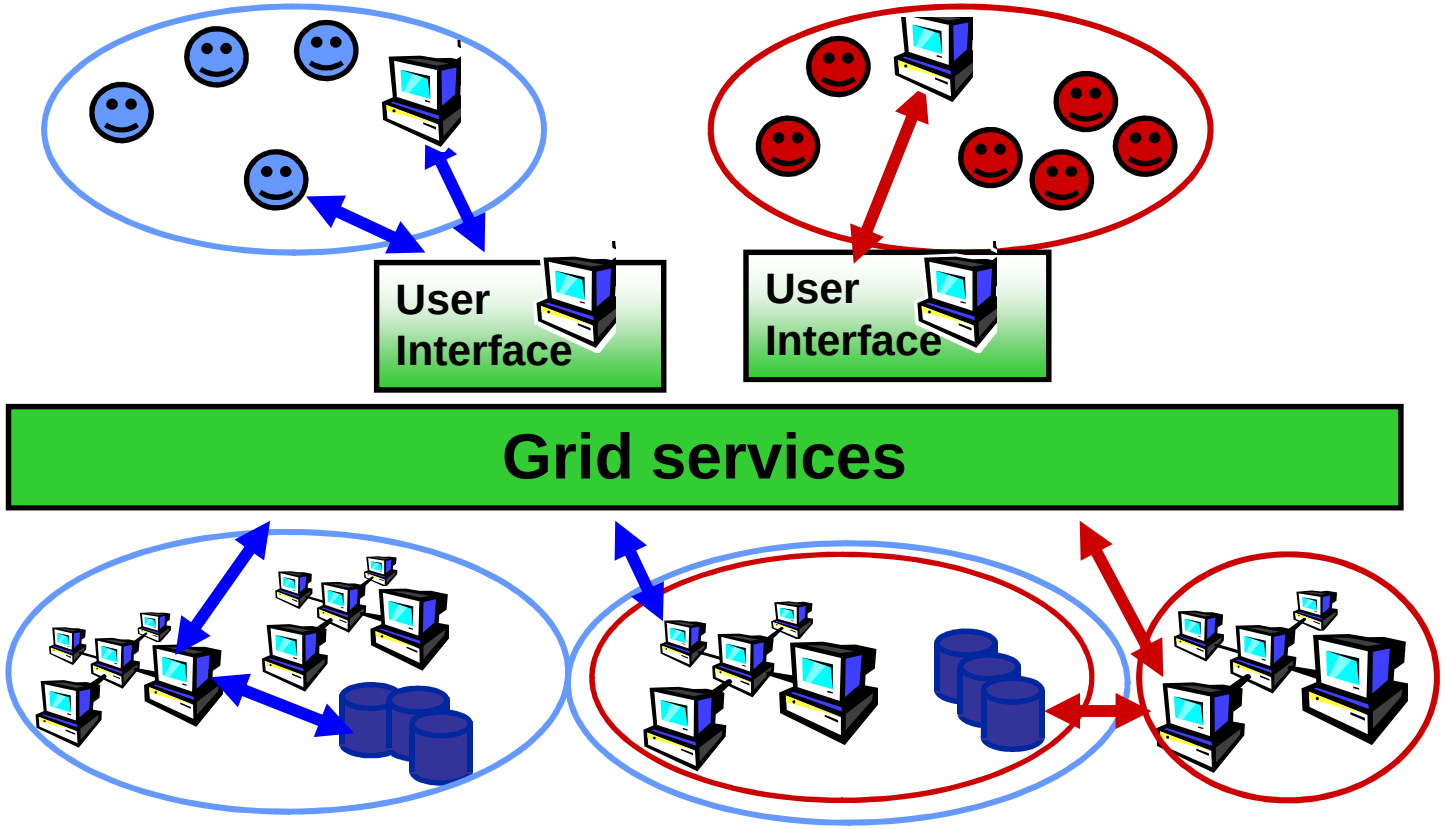
- Сакривање дистрибуције апликације
- Сакривање хетерогености система
- Обезбеђивање униформног, стандардног интерфејса високог нивоа развојним тимовима апликацијама
- Обезбеђивање скуп сервиса различите функционалности

Middleware

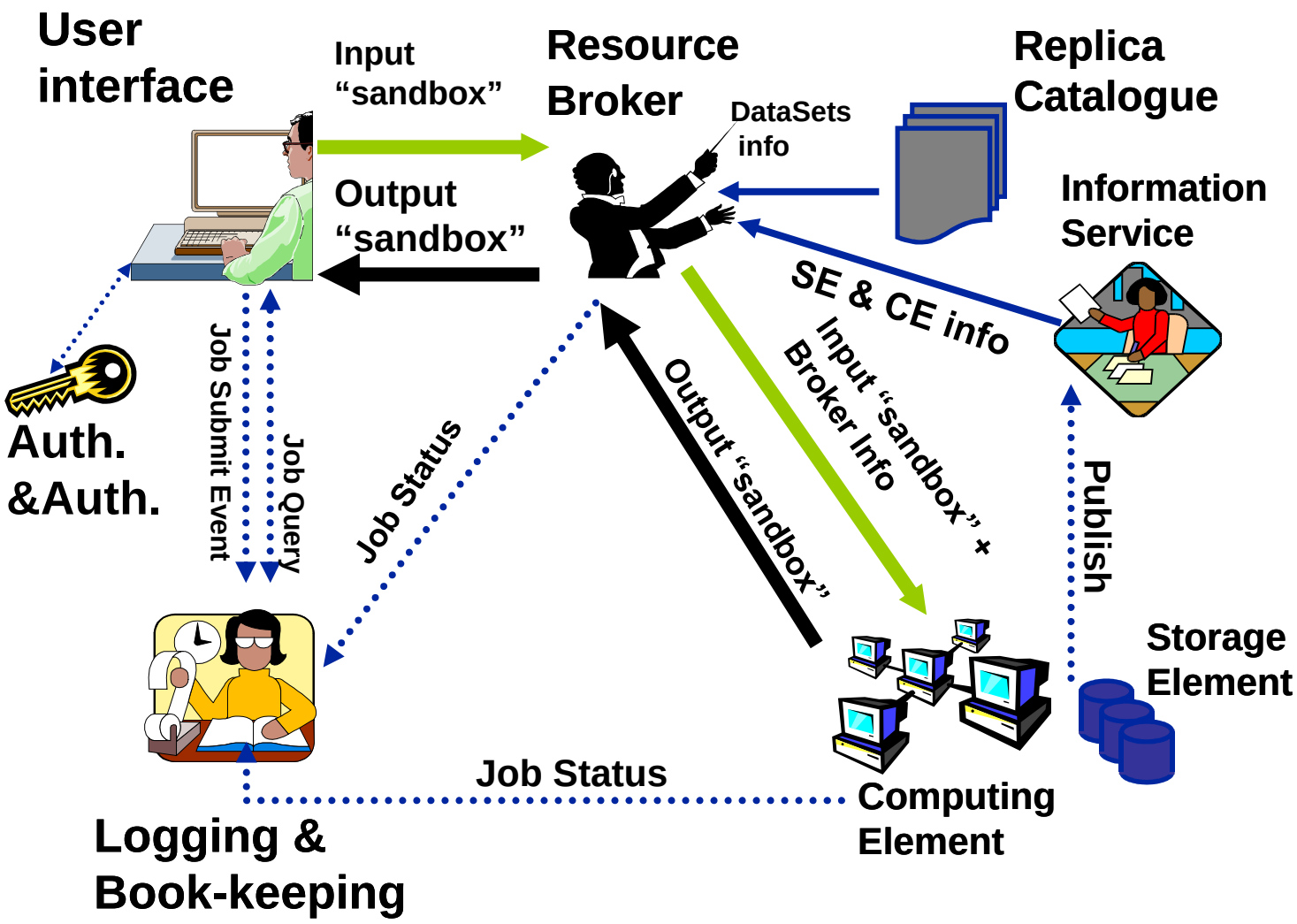
- LCG: Large Hadron Collider Computing Grid
- LCG infrastructure running LCG-2 is “EGEE-0”
- In parallel producing new web-service-oriented middleware (“gLite”), which will replace LCG-2 as production facility this year



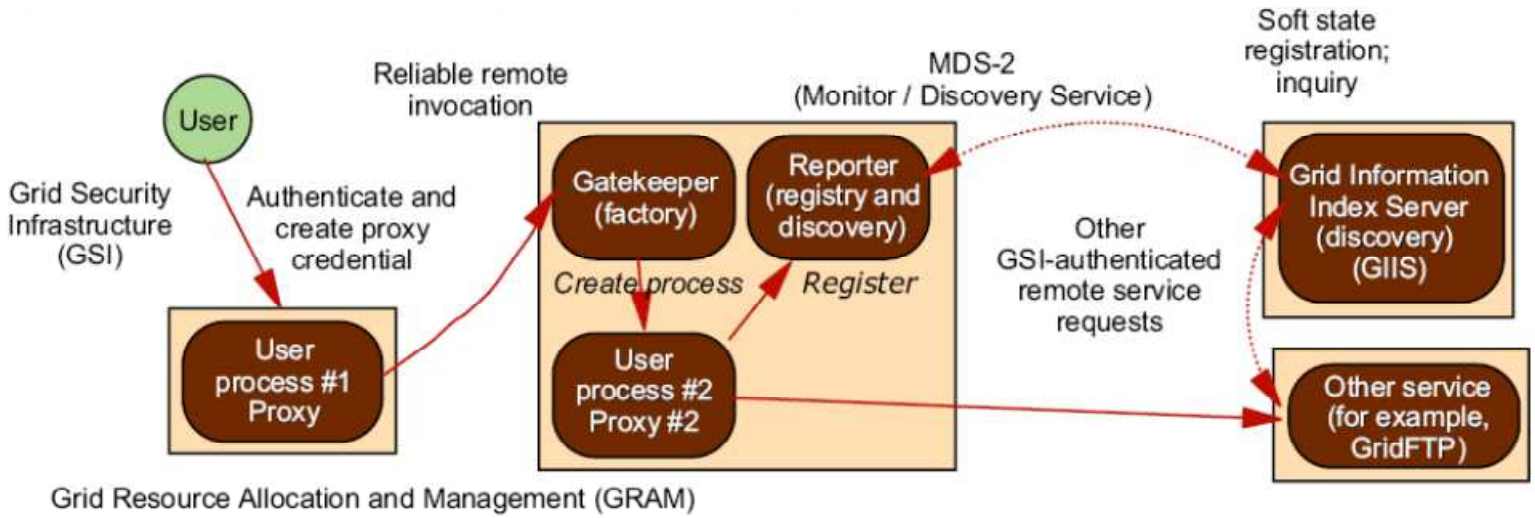
Кориснички поглед на Грид



Функционисање



Функционисање



Workload Management System (WMS)

- Distributed scheduling
  - multiple UI's where you can submit your job
  - multiple RB's from where the job can be sent to a CE
  - multiple CE's where the job can be put in a queuing system

- Distributed resource management
  - multiple information systems that monitor the state of the grid
  - Information from SE, CE, sites

Authentication and Authorization

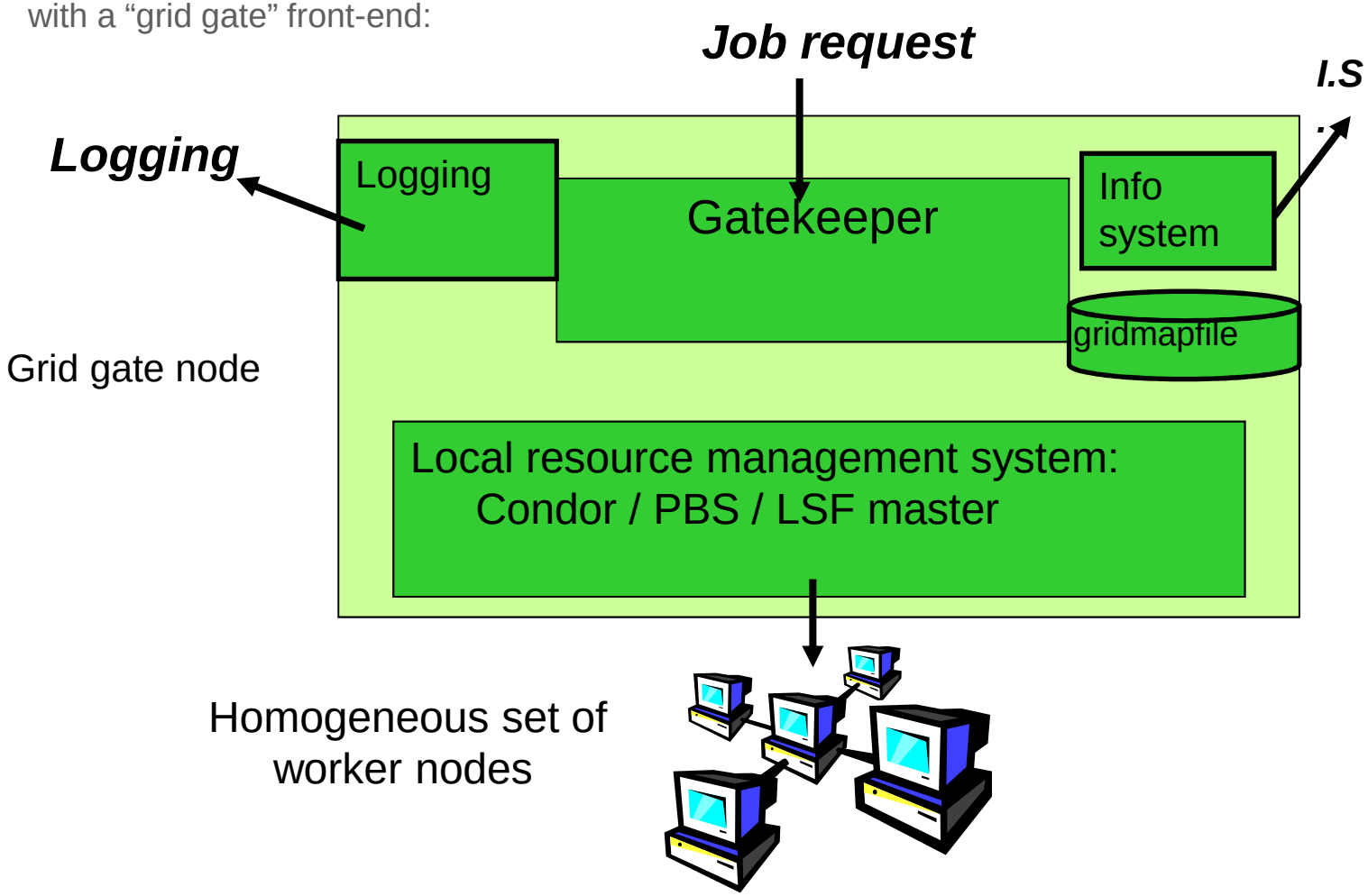
- Authentication
  - User obtains certificate from CA
  - Connects to UI by ssh
  - Downloads certificate
  - Invokes Proxy server
  - Single logon – to UI - then Secure Socket Layer with proxy identifies user to other nodes
- Authorization - currently
  - User joins Virtual Organisation
  - VO negotiates access to Grid nodes and resources (CE, SE)
  - Authorization tested by CE, SE: gridmapfile maps user to local account

User Interface (UI)

- Листање ресурса који одговарају потребама посла који треба да се изврши
- Подношење посла на извршење
- Отказивање посла
- Преузимање резултата извршеног посла
- Прегледање информација о статусу посла који се извршава,
- Манипулисање (смјештање, копирање и брисање) датотекама на *Гриду*
- Прегледање информација о статусу различитих ресурса у оквиру *Грид* мреже.
- Покретање посла коришћењем посебног језика JDL (Job Description Language)

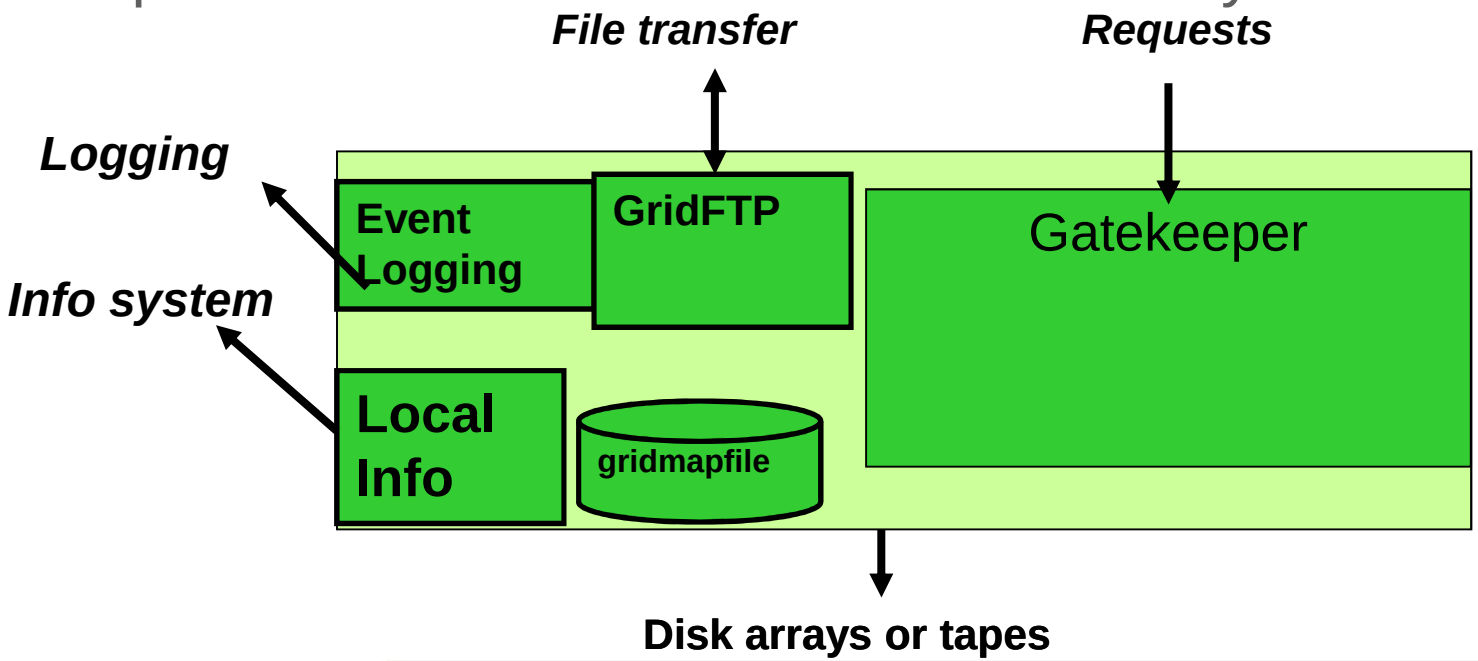
Computing Element (CE)

A CE is a grid batch queue
with a “grid gate” front-end:



Storage Element (SE)

- Storage elements hold files: write once, read many
- Replica files can be held on different SE:
  - “close” to CE; share load on SE
- Replica Catalogue - what replicas exist for a file?
- Replica Location Service - where are they?



Resource Broker

- Run the Workload Management System
  - To accept job submissions
  - Dispatch jobs to appropriate Compute Element (CE)
  - Allow users
    - To get information about their status
    - To retrieve their output
- A configuration file on each UI node determines which RB node(s) will be used
- When a user submits a job, JDL options are to:
  - Specify CE
  - Allow RB to choose CE (using optional tags to define requirements)
  - Specify SE (then RB finds “nearest” appropriate CE, after interrogating Replica Location Service)

Logging and Bookkeeping

- Who did what and when?
- What’s happening to my job?
- Usually runs on RB node

Information System

- Receives periodic (~5 min) updates from CE, SE
- Used by RB node to determine resources to be used by a job
- Currently BDII is used

Коришћење Грида

Сертификати за прокси

- Креирање проксија

```
$ voms-proxy-init --voms <vo>
```

- Прикупљање информација о проксију

```
$ voms-proxy-info
```

- Гашење проксија

```
$ voms-proxy-destroy
```

Покретање посла

- Покретање посла, ver={edg, glite}

```
$ ver-job-submit <jdl_file>
$ ver-job-submit –o <job_ID_file> <jdl_file>
$ ver-job-submit –r <CE_ID> <jdl_file>
```

- Дохватање информација о послу

```
$ ver-job-status <job_ID>
$ ver-job-status –i <job_ID_file>
```

- Дохватање излаза посла

```
$ ver-job-get-output <job_ID>
$ ver-job-get-output –i <job_ID_file>
$ ver-job-get-output –dir <path> <job_ID>
```

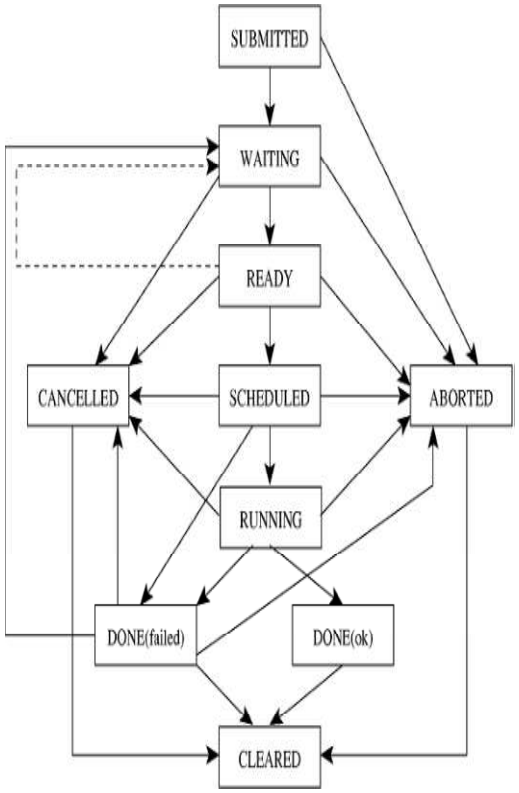
- Отказивање посла

```
$ ver-job-cancel <job_ID>
$ ver-job-cancel –i <job_ID_file>
```

- Листање погодних CE

```
$ ver-job-list-match <jdl_file>
```

Стања посла


SUBMITTED
WAITING
READY
CANCELLED
SCHEDULED
ABORTED
RUNNING
DONE(failed)
DONE(ok)
CLEARED

Атрибути JDL фајла

- **Type**
  for LCG2 only “Job” type supported
- **JobType**
  values: “Normal”, “MPICH”, “Interactive”…
- **Executable**
  command name
- **Arguments**
  command line arguments
- **StdOutput**
  file where the job standard output goes
- **StdError**
  file where the job standard error goes
- **InputSandbox**
  files for transfer from UI to the WN
- **OutputSandbox**
  files for transfer from WN to the UI

Атрибути JDL фајла

- Requirements

  to send a job to a particular CE

  Requirements = other.GlueCEUniqueID ==

  “ce.phy.bg.ac.yu:2119/jobmanager-pbs-seegrid”;

  job must run on a CE where a particular experiment software is installed

  Requirements = Member("VO-atlas-release-11.0.2",

  other.GlueHostApplicationSoftwareRunTimeEnvironment);

  for MPI job

  Requirements = Member("MPICH", other.GlueHostApplicationSoftwareRunTimeEnvironment)

Пример JDL фајла

```
Executable = "/bin/hostname";
Arguments = "";
StdOutput = "message.txt";
StdError = "stderror.txt";
OutputSandbox = {"message.txt","stderror.txt"};
Requirements =
   RegExp("rti29.etf.bg.ac.yu*",other.GlueCEUniqueID);
```

Пример JDL фајла

```
Executable = "/usr/java/jdk1.6.0/bin/java";
Arguments = " -classpath
   /storage/exp_soft/drs/gravitacion/Gravitational
StdOutput = "message.txt";
StdError = "stderror.txt";
OutputSandbox = {"message.txt","stderror.txt"};
Requirements =
   RegExp("rti29.etf.bg.ac.yu*",other.GlueCEUniqueID);
```

Задаци

N Body Gravitational Problem

У свемиру постоји N небеских тела која међусобно интерагују. Користећи програмски језик Јава потребно је симулирати кретање ових тела користећи више рачунара у паралели. При томе треба узети да је један рачунар задужен за више небеских тела. Број небеских тела је много већи од броја расположивих рачунара.

N Body Gravitational Problem

$$m_j \ddot{q}_j = \gamma \sum_{k \neq j}^{n} \frac{m_j m_k (q_j - q_k)}{|q_j - q_k|^3}, j = 1, \ldots, n \qquad (1)$$

Питања?

Захарије Радивојевић
Електротехнички Факултет
Универзитет у Београду
zaki@etf.rs