Архитектура и основные сервисы gLite (проект EGEE)

Н. Клопов ***(Петербургский Институт Ядерной Физики РАН)***

Что такое middleware gLite?

Middleware – в контексте Grid, это специфическое программное обеспечение, использующееся для функционирования распределенной компьютерной сети.

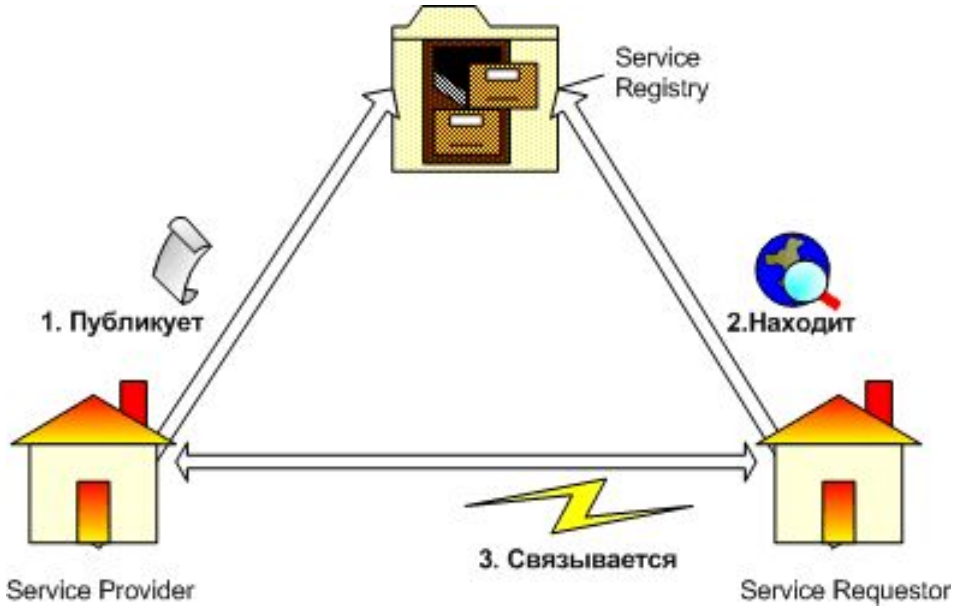
gLite: очередное поколение промежуточного программного обеспечения (ППО) проекта EGEE

Исходно EGEE использовал ППО своего предшественника - проекта EDG (European Data Grid). Это ППО затем было развито в пакет LCG, и именно LCG работал в инфраструктуре EGEE на ранней стадии проекта. Параллельно в EGEE были выполнены работы по модернизации большей части исходного пакета, и был создан новый продукт – gLite, который сейчас устанавливается в инфраструктуру

Принципы архитектуры gLite

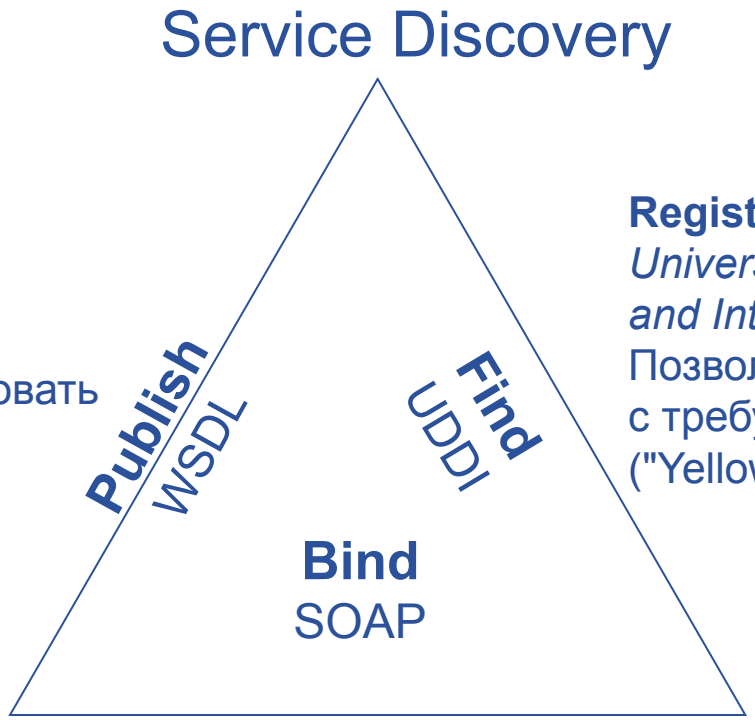
- EGEE middleware is supposed to be developed following Service Oriented Architecture (SOA) model. A service is a function which is well-defined, self-contained and does not depend on the context or state of other services.
- The services communicate with each other through well-defined interfaces and protocols (data passing or coordination of activities)
- Based on WEB service application that exposes its features using standard Internet protocol. WEB services interact by exchanging messages using Simple Object Access Protocol (SOAP) standard.
- Web Service Definition Language (WSDL) is used to specify the interface a service exposes.

Service Oriented Architecture



- Service Oriented Architecture(SOA) определяет, как несколько независимых, распределенных процессов должны взаимодействовать при выполнении общей задачи (CORBA, DCOM).

A Web Services Architecture



Interface
Web Services Description Language (WSDL)
Определяет как использовать Сервис

Registry
Universal Description, Discovery and Integration (UDDI)
Позволяет определить адрес сервиса с требуемой функциональностью ("Yellow Pages")

- **Веб-сервис – программная система, идентифицируемая URI, интерфейс внешнего доступа которой and bindings описываются при помощи WDSL. Другие программные системы могут обнаруживать и взаимодействовать с Веб-сервисами в соответствии с их описанием на основе использования XML-сообщений посредством протокола SOAP.**

Пример запроса

ReplicaManager::getAccessCost (LFN[], CE)

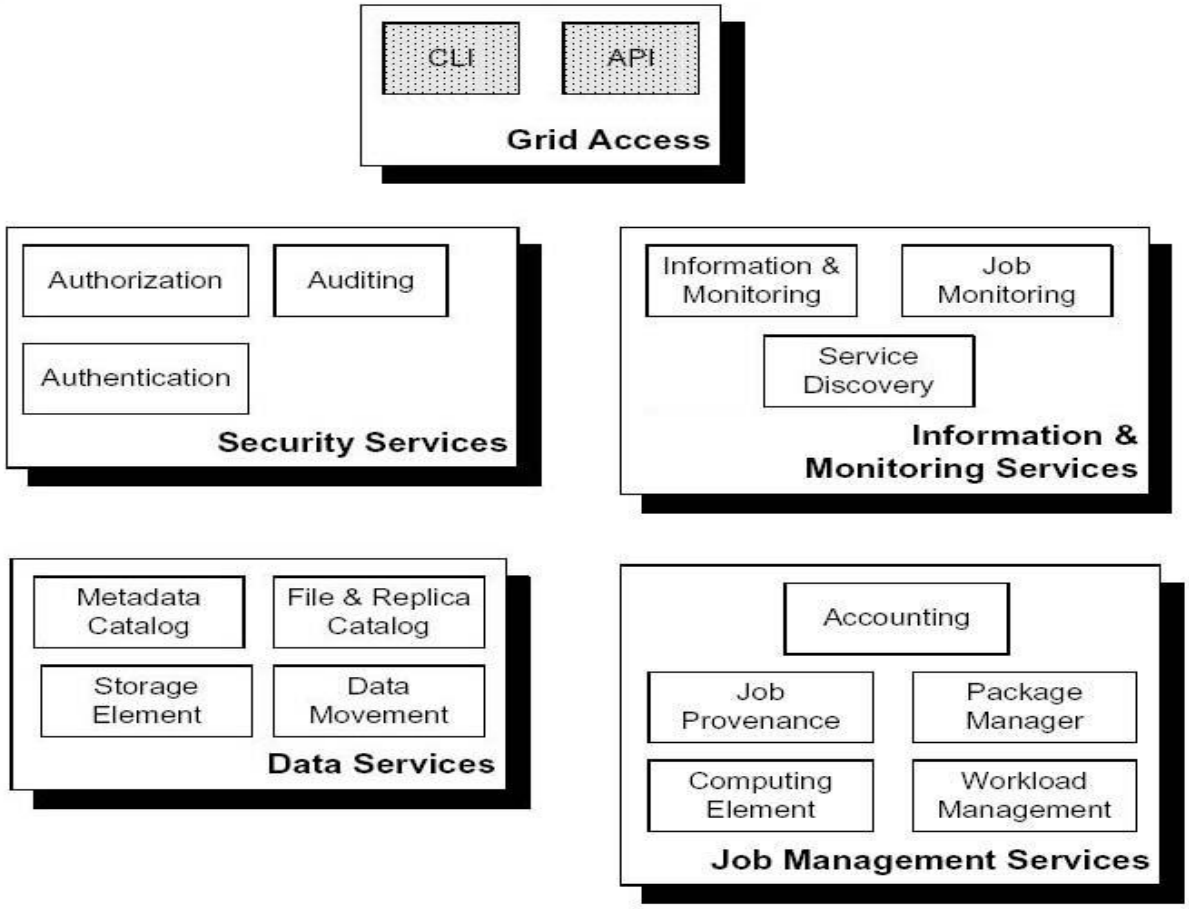
```
<SOAP-ENV: Envelope
        ……..
  <SOAP-ENV:Header> ….. </SOAP-ENV:Header>
     <SOAP-ENV:Body>
       <m:getAccessCost  xmlns:ns1=http://datagrid:ReplicaManager
       <LFN xsi:type="SOAP-ENC:ARRAY" SOAP-ENC:ArrayType="xsd:string[2]">
            <lfn> host1.cern.ch/path1/file1</lfn>
            <lfn> host1.cern.ch/path2/file2</lfn> </LFN>
        <CE xsi:type="xsd:string">myComputeElement  </CE>
        </m:getAccessCost>
    </SOAP-ENV:Body>
</SOAP-ENV: Envelope>
```

Пример ответа

Return message:

```
<SOAP-ENV: Envelope
        .....
     <SOAP-ENV:Body>
       <m:getAccessCostResponse
    xmlns:ns1=http://datagrid:ReplicaManager
           <return xsi type="SOAP-ENC:ARRAY"
    SOAP-ENC:ArrayType="xsd:string[2]">
       <pfn> host3.ral.ac.uk/path4/file2   </pfn>
                  <pfn> host3.ral.ac.uk/path7/file4 </pfn>
      </return>
       </m:getAccessCostResponse>
   </SOAP-ENV:Body>
</SOAP-ENV: Envelope
```

Основные сервисы gLite



Основные группы сервисов

GSI-grid security infrastructure

Три основных аспекта безопасности:

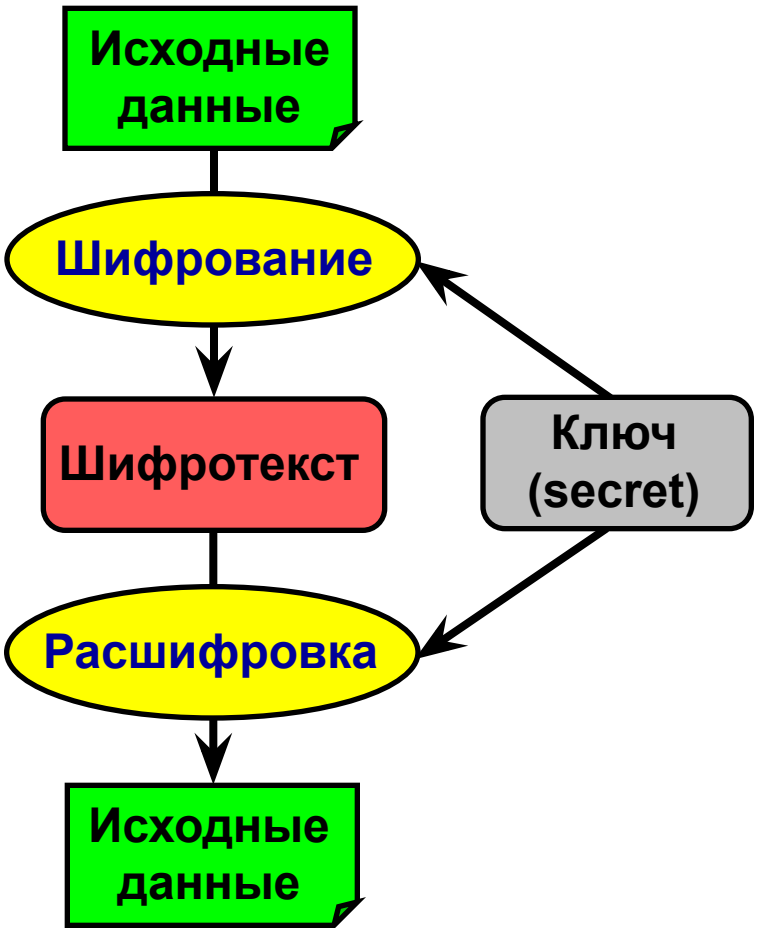
Privacy – Обмен сообщениями должен быть приватным.
(доступность передаваемых данных только участникам диалога)

Integrity – Целостность данных, т.е. неизменность передаваемых данных

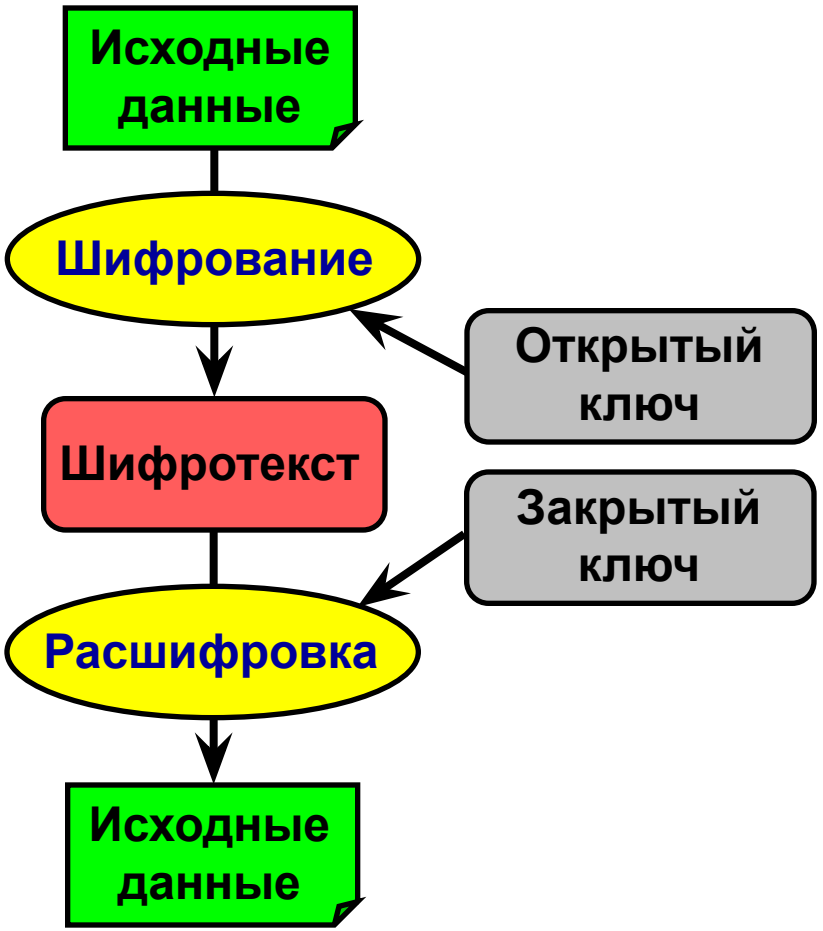
Authentication – Идентификация сторон, участвующих в диалоге (проверка подлинности сущности)

Алгоритмы шифрования

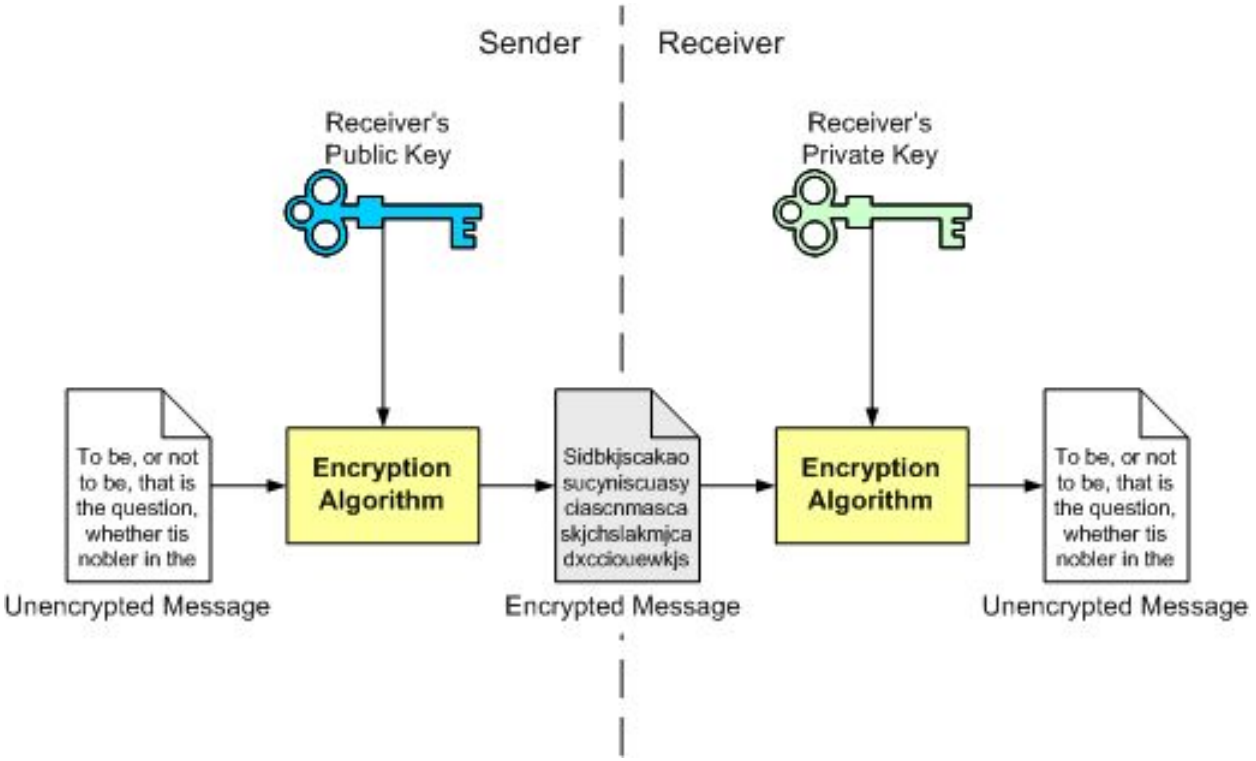
Симметричный алгоритм



Несимметричный алгоритм

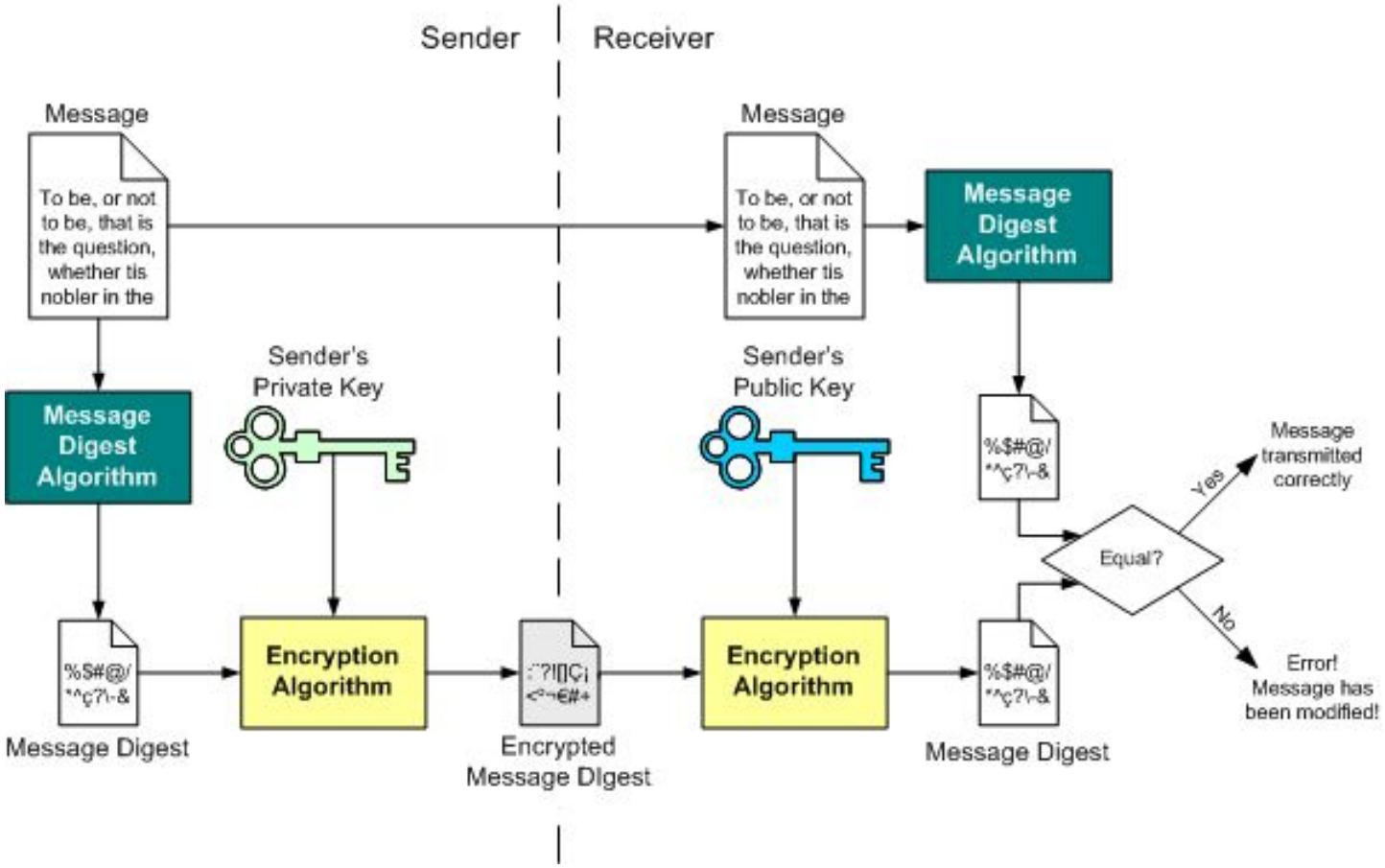


Privacy in public-key cryptography



Integrity in public-key systems

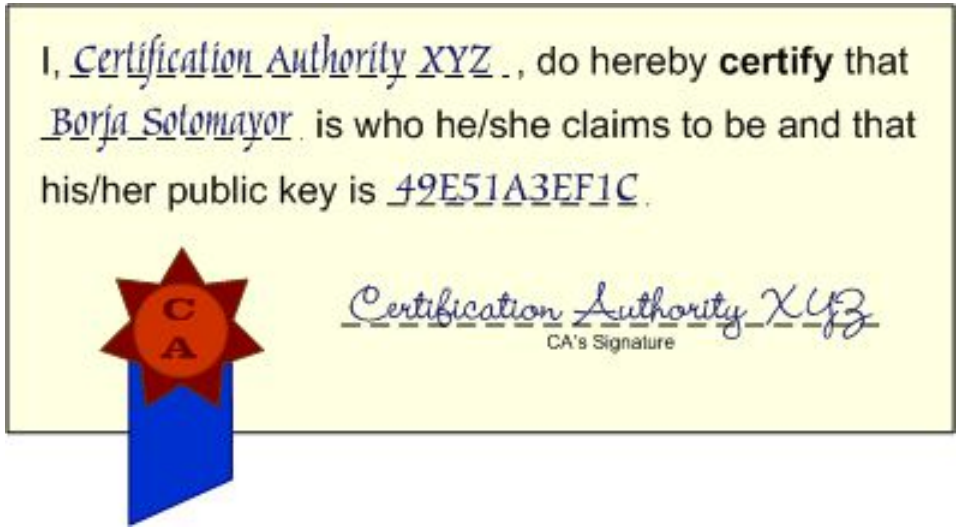
Digital signatures (Цифровая подпись)



Authentication

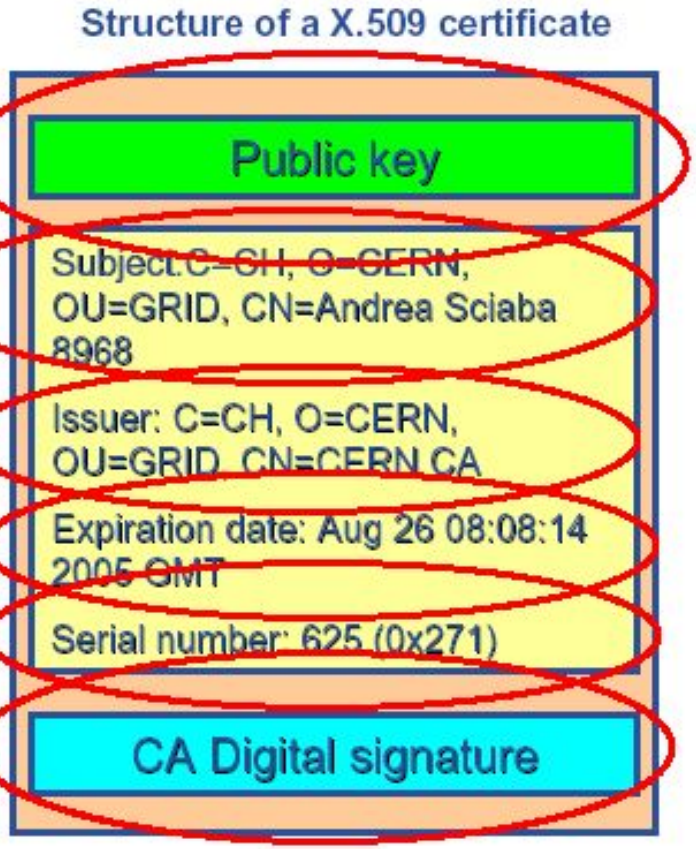
Certificates and certificate authorities

digital certificate – цифровой документ, подтверждающтй, что данный public key пренадлежит конкретному пользователю (системе, сервису). Этот документ подписан 3-й стороной, называемой certificate authority (CA). *Доверие сертификату строится на доверии третьей стороне, подписавшей этот сертификат*



X.509 сертификаты

- **An X.509 Certificate contains:**
  - owner's public key;
  - identity of the owner;
  - info on the CA;
  - time of validity;
  - Serial number;
  - digital signature of the CA



Challenge-response authentication

- **Алиса (А)** хочет аутентифицировать **Боба (Б).**
- **Б** посылает свой сертификат Алисе, она проверяет правильность сертификата и подпись (имеет РК CA).
- **А** посылает Бобу произвольную фразу (challenge) с просьбой зашифровать её закрытым ключом Боба.
- **Б** шифрует пришедшие данные и отсылает ответ (response) Алисе.
- **А** расшифровывает ответ Боба с помощью переданного ранее открытого ключа и сравнивает результат с эталонной фразой.
- Если сравнение успешно, то Боб действительно владеет закрытым ключом, сответствующим сертификату.

proxy-сертификат

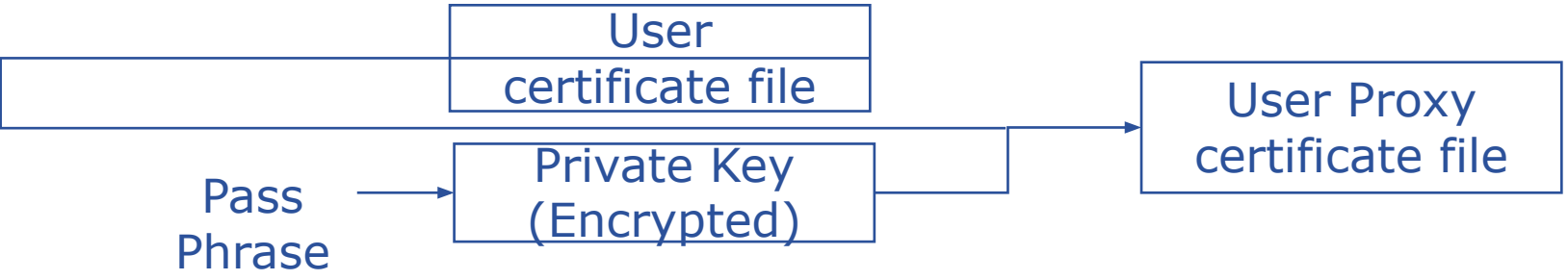
Проблемы:

Single sign-on

однократное предъявление
первичного закрытого ключа

Delegation

Proxy-certificate



Применение proxy-сертификата для аутентификации избавляет пользователя от необходимости вводить свой пароль при каждом взаимодействии с сервисами.

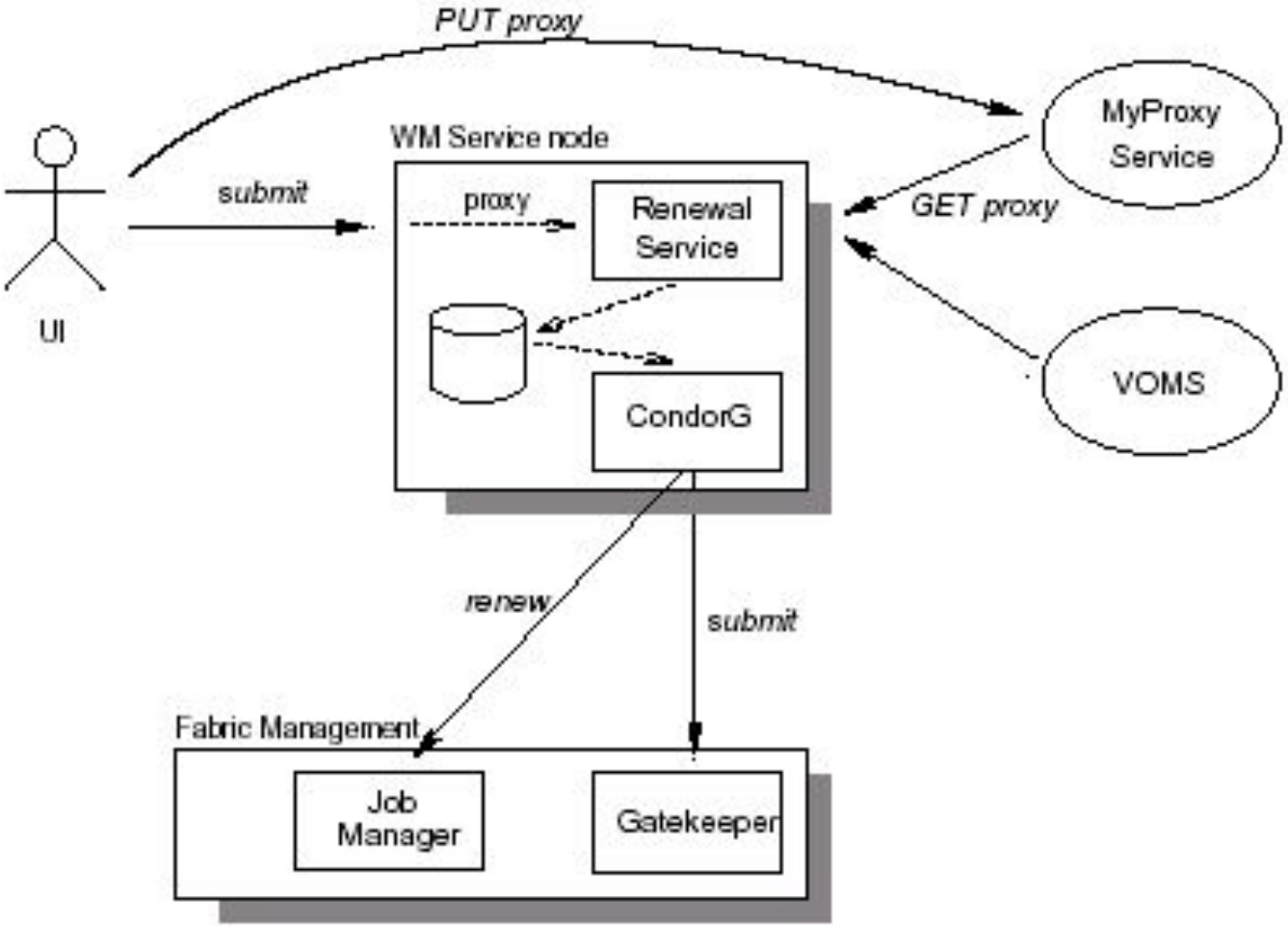
Можно передавать свои proxy-сертификаты другим субъектам для выполнения операций от своего имени.

Ограниченное время действия

proxy-сертификат

- voms-proxy-init –voms picard
- voms-proxy-info --all
- subject : /C=RU/O=RDIG/OU=users/OU=pnpi.nw.ru/CN=Nikolai Klopov/CN=proxy
- issuer : /C=RU/O=RDIG/OU=users/OU=pnpi.nw.ru/CN=Nikolai Klopov
- identity : /C=RU/O=RDIG/OU=users/OU=pnpi.nw.ru/CN=Nikolai Klopov
- type : proxy
- strength : 512 bits
- path : /tmp/x509up_u6901
- timeleft : 11:59:43
- === VO picard extension information ===
- VO : picard
- subject : /C=RU/O=RDIG/OU=users/OU=pnpi.nw.ru/CN=Nikolai Klopov
- issuer : /C=IT/O=INFN/OU=Host/L=CNAF/CN=cert-voms-01.cnaf.infn.it
- attribute : /picard/Role=NULL/Capability=NULL

Renewal Architecture



Authorization

Каждый Grid ресурс должен определить действия, которые может выполнять пользователь.

-Database server: read/write/create permission?

- Compute element: permission to execute?

-Storage Element: write/read access?

- Обычно выполняется через grid-mapfiles:

```
...
"/C=CH/O=CERN/OU=GRID/CN=Simone Campana 7461" .dteam
"/C=CH/O=CERN/OU=GRID/CN=Andrea Sciaba 8968" .cms
"/C=CH/O=CERN/OU=GRID/CN=Patricia Mendez Lorenzo-ALICE" .alice
...
```

- Много пользователей и привелегий -> трудно администрировать!

Пользователи объединяются в Виртуальные Организации (VO).

Виртуальные организации

- «Динамическое собрание одиночек и организаций, гибко, безопастно и координировано разделяющее ресурсы» -- LCG-2 User Guide.
- **VO с технической точки зрения:** LDAP или HTTP или VOMS ресурс, перечисляющий Distinguished Names сертификатов пользователей конкретной VO.
- **LCG-2:** файл /etc/grid-security/grid-mapfile, один сертификат – одна виртуальная организация, нет разделения пользовательских ролей внутри VO.
- **VOMS:** призвана для управления ролью пользователя внутри VO и создания пользовательских групп. Вместо использования LDAP/HTTP для отображения пользователей будет использоваться VOMS-запросы. Внутренняя структура хранилища – Relational Database.

Example

- VOMS presents a user's VO membership information as an extension to their X509 proxy certicate.
- `GROUP: {string that names the group}`
- `ROLE: {string that gives the user role(s) in this group}`
- `CAP: {special capabilities assigned to this role}`
- Example: A user works on ATLAS high energy physics experiment

GROUP	ROLE	Special CAPability
ATLAS	user	none
ATLAS CAL	user	none
ATLAS LARG	update	10G disk space
ATLAS FCAL	administrator	full write privileges

Data management

- **Гетерогенность**
  - Данные хранятся на различных устройствах (диски, ленты), использующих различные методы доступа
  - Необходим общий интерфейс к устройствам
    - Storage Resource Manager (SRM)
- **Распределенность**
  - Данные хранятся на различных сайтах, где отсутствует общая разделяемая файловая система
  - Данные могут перемещаться между различными сайтами
  - Необходим способ определения местоположения файлов
    - File and Replica Catalogs
  - Необходима управляемая, надежная передача файлов
    - File transfer and placement services
- **Различные административные домены**
  - Данные храняться там, куда обычный доступ вам запрещен
  - Необходима общая модель безопасности
    - ACLs enforcement based on Grid identities – DNs

Data Management Services

- **Storage Element – common interface to storage**
  - Storage Resource Manager — Castor, dCache, DPM, …
  - POSIX-I/O — gLite-I/O, rfio, dcap, xrootd
  - Access protocols — gsiftp, https, rfio, …
- **Catalogs – keep track where data is stored**
  - File Catalog — gLite File and Replica Catalog
  - Replica Catalog
  - File Authorization Service
  - Metadata Catalog — Application specific catalogs
- **File Transfer – scheduled reliable file transfer**
  - Data Scheduler
  - File Transfer Service — gLite FTS and glite-url-copy;
  - File Placement Service — gLite FPS
    (FTS and catalog interaction)

Storage Resource Management

- Данные хранятся на **disk pool servers** или **Mass Storage Systems**
- Управление этими ресурсами должно обеспечивать:
  - Прозрачный доступ к файлам (migration to/from disk pool)
  - Выделение места для файлов (Space reservation)
  - Управление временем жизни файлов (Life time management)
  - ….
- SRM (Storage Resource Manager) сервис реализует все эти требования:
  - SRM is a Grid Service that takes care of local storage interaction and provides a Grid interaface to outside world
- Interactions with the SRM is typically hidden by higher level services

File Naming problem

- **Имена файлов на SE имеют только локальное значение:**
  - `/tmp/picard/file1` (Unix)
  - `srm://castorgrid.cern.ch:8443/srm/managerv1?SFN =/castor/cern.ch/file1`

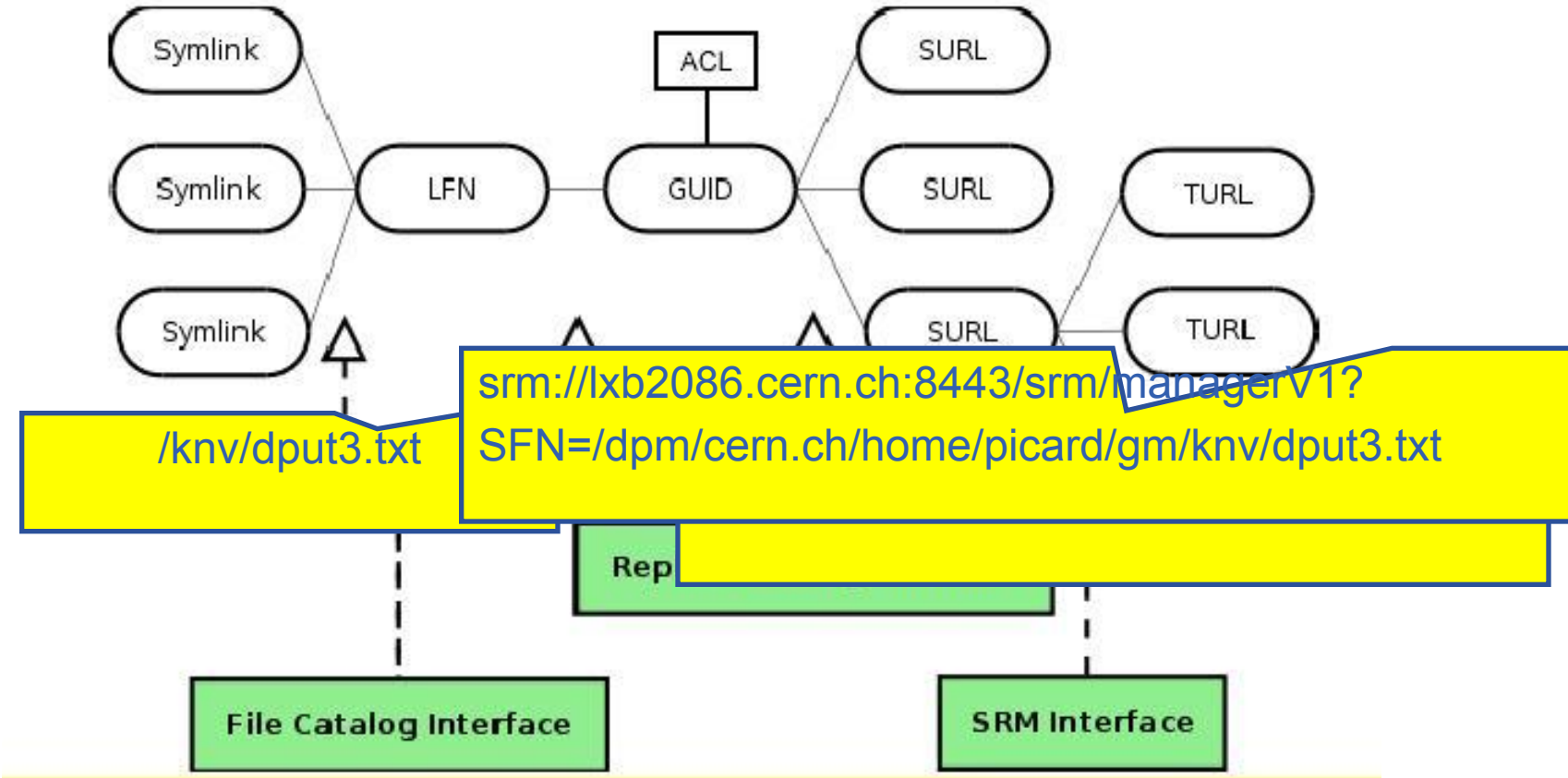
    (SRM Site URL – SURL)

Локальная система может преобразовывать имена файлов, напр. SURL не может использоваться прямо, он должен быть преобразован SRM в Transfer URL (TURL) :

`gsiftp://se05.cern.ch/scratch/file05`

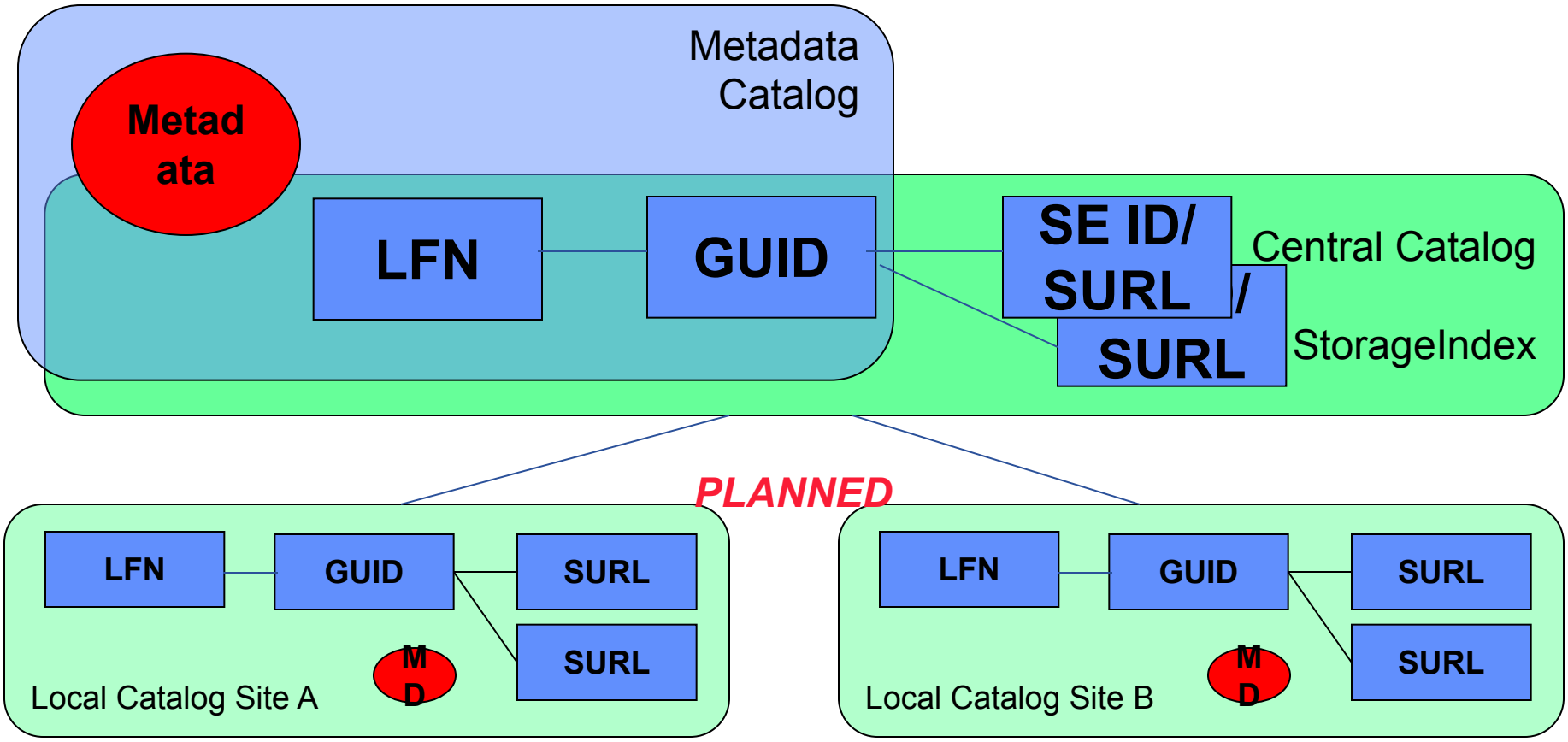
Для доступа к файлам необходим подход, позволяющий абстрагироваться от локальной системы имен и обеспечить общий для GRID среды механизм имен файлов

Data Naming


Symlink
Symlink
Symlink
LFN
ACL
GUID
SURL
SURL
SURL
TURL
TURL
/knv/dput3.txt
srm://lxb2086.cern.ch:8443/srm/managerV1?
SFN=/dpm/cern.ch/home/picard/gm/knv/dput3.txt
Rep
File Catalog Interface
SRM Interface

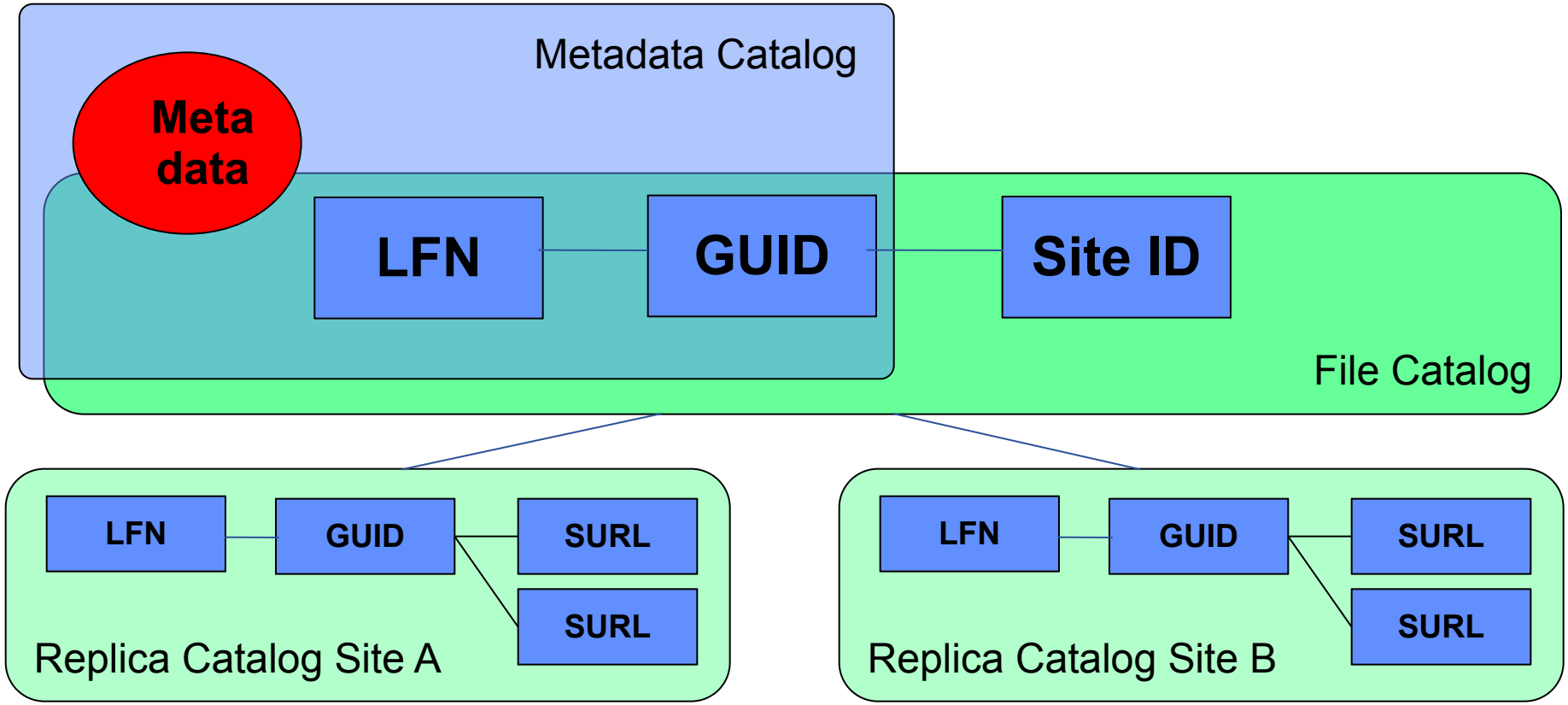
Каталоги данных

- Позволяет определить, где размещены файлы в Grid
- Может определить дополнительную семантику на LFN (атрибуты, ACLs,..)
- Позволяет определить местоположение реплик



Каталоги данных

- Позволяет определить, где размещены файлы в Grid
- Может определить дополнительную семантику на LFN (атрибуты, ACLs,..)
- Позволяет определить местоположение реплик



FiReMan Commands

- glite-catalog-ls
- glite-catalog-stat
- glite-catalog-mkdir
- glite-catalog-rmdir
- glite-catalog-mv
- glite-catalog-symlink
- glite-catalog-create
- glite-catalog-rm

- **glite-catalog-setreplica**
- **glite-catalog-getreplica**

- p - allow to change the permissions
- d - delete the entry
- r - read the file
- w - write to the file
- l - list contents
- x - execute
- g - get the meta data of the file
- s - set the meta data of the file

- **glite-catalog-getattr**
- **glite-catalog-setattr**

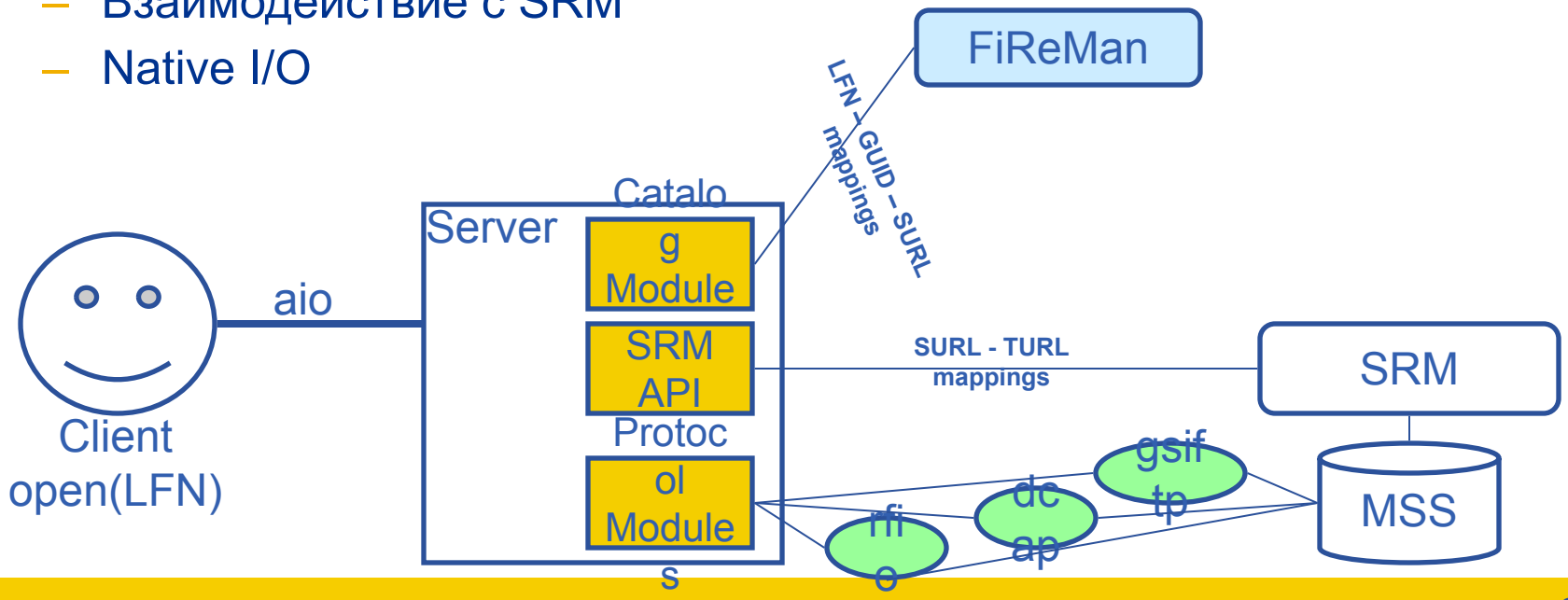
- **glite-catalog-chmod**
- **glite-catalog-getacl**
- **glite-catalog-setacl**

- glite-get
- glite-put
- glite-rm

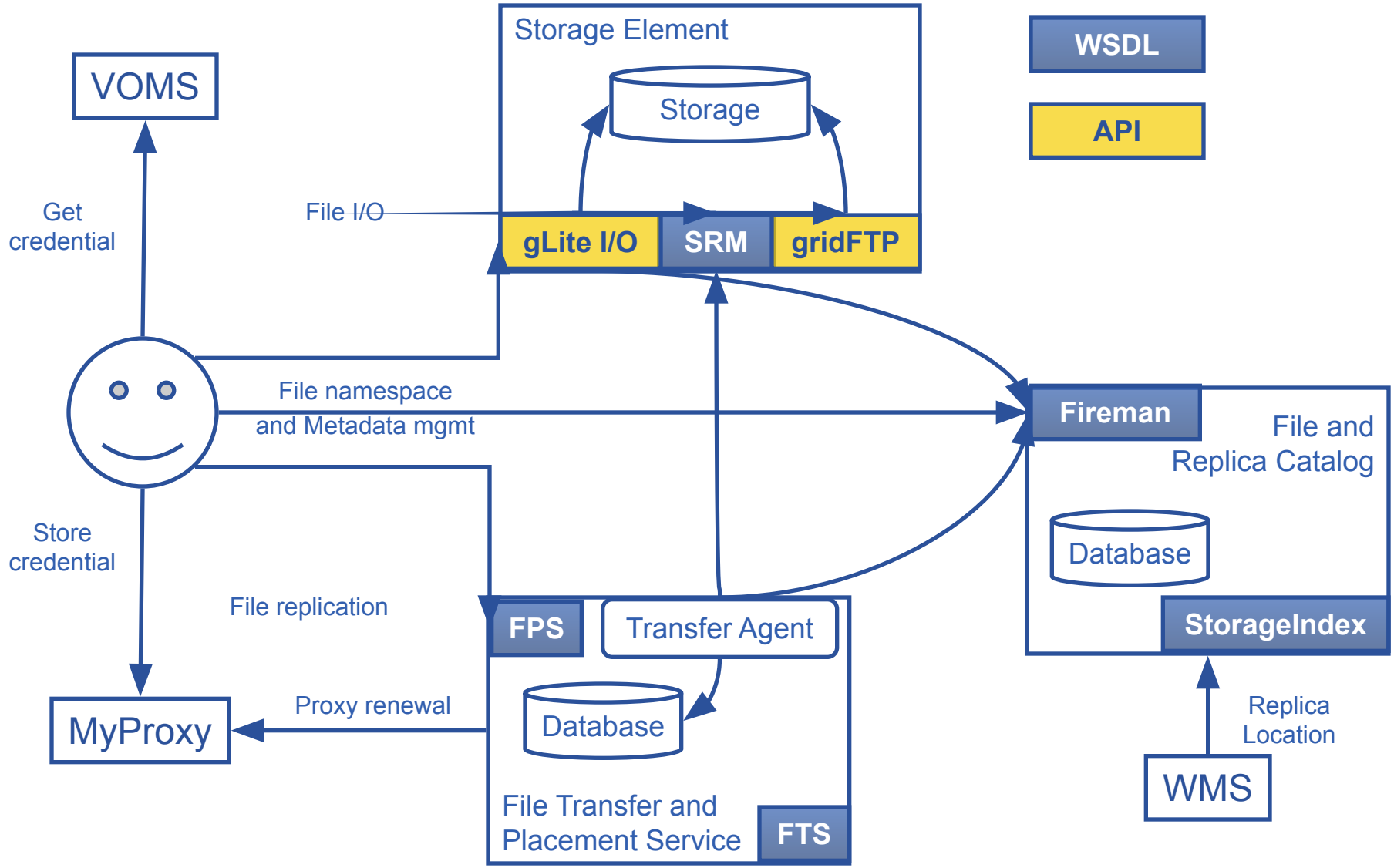
```
# LFN: /knv/dput4.txt
# User: /C=RU/O=RDIG/OU=users/OU=pnpi.nw.ru/CN=Nikolai Klopov
# Group: egee-group
# Base perms: user pdrwl-gs, group --r-l-g-, other --------
```

File Access Overview

- **Клиент использует API library или Command Line Interface**
  - GUID или LFN может использоваться, напр. open(“/grid/myFile”)
- **GSI Delegation to gLite I/O Server**
- **Сервер выполняет все операции от имени клиента**
  - Преобразование LFN/GUID в SURL и TURL
- **Встроенные операции:**
  - Взаимодействие с каталогом
  - Взаимодействие с SRM
  - Native I/O

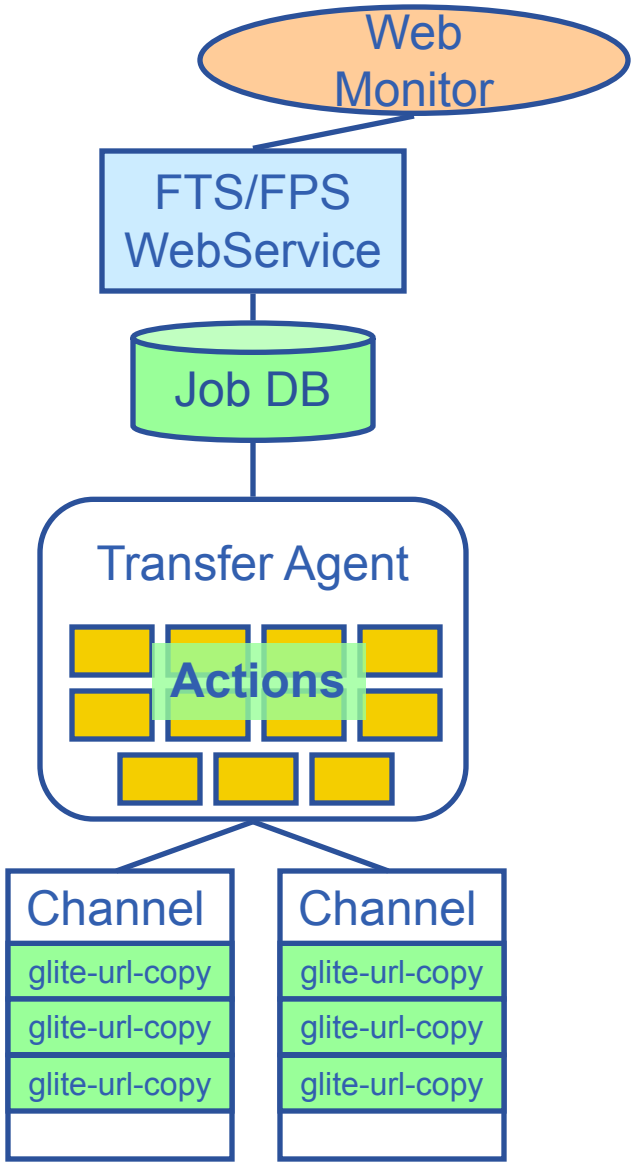


DM Interaction Overview



gLite FTS/FPS

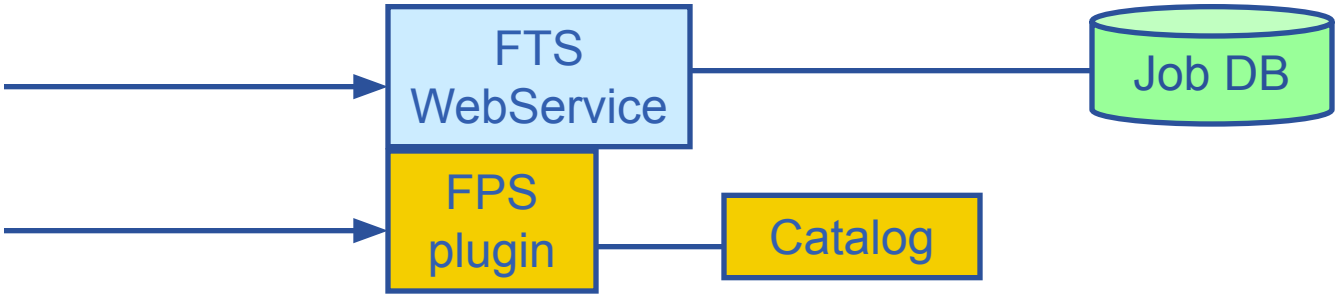
- **File Transfer/Placement Service (FTS,FPS)**
  - База данных заданий на передачу файлов
  - Предоставляет Transfer Web Service интерфейс для клиента (submit, cancel, status)
  - Имеет Web Interface
  - Модифицирует Catalog если необходимо
- **Transfer Agent**
  - Основные действия
    - Получает задания из Transfer Job Database
    - Управляет передачей через множество каналов
    - Мониторирует статус и модифицирует Transfer Job Database
- **Transfer Service (glite-url-copy)**
  - Фактически выполняет передачу: SRM – SRM, gsiftp – SRM, gsiftp – gsiftp
  - Мониторирование



FTS/FPS

- **File Transfer Service (FTS)**
  - Действует только через SRM SURLs or gsiftp URLs
  - `submit(source-SURL, destination-SURL)`

- **File Placement Service (FPS)**
  - Позволяет работать с LFNs
  - Взаимодействует с replica catalogs
  - Регистрирует реплики в каталоге реплик
  - `submit(transferJobs) (transferJob = sourceLFN, destinationSE)`



Информационная Система gLite

В распределенной среде важна возможность получать информацию о доступных в данный момент ресурсах. Эта информация может включать:

- сайты (CE), способные выполнить данное задание, их загрузка, ПО, установленное на них.
- сайты, предоставляющие возможности для хранения данных, включая их статус, максимальный размер и число файлов, которые могут быть сохранены.
- данные мониторирования процесса выполнения задания

Uses of the IS in EGEE/LCG

If you are a user

Retrieve information of Grid resources and status

Get the information of your jobs status

If you are a middleware developer

Workload Management System:
Matching job requirements and Grid resources

Monitoring Services:
Retrieving information of Grid Resources status and availability

If you are site manager or service

You "generate" the information for example relative to your site or to a given service

Information and Monitoring

- LCG-2 currently uses
  GT Monitoring and Discovery Service (MDS) architecture together with Berkley Database Information Indexes (BDII)

- The information system is built on *LDAP*
  Light-weight Directory Access Protocol
- A Schema describes the attributes and the types of the attributes associated with data objects
- Example:
  GlueSiteInfo

```
dataGridVersion: LCG-2_0_0
installationDate: 200404131100Z
objectClass: SiteInfo
siteName: nikhef.nl
siteSecurityContact: grid-support-admin@nikhef.nl
sysAdminContact: grid-support-admin@nikhef.nl
userSupportContact: grid-support-admin@nikhef.nl
```

Элементы Информационной Системы

Основные шаги:

1. На каждом сайте PROVIDERS передают статическую и динамическую информацию о своем состоянии на SERVERS

2. CENTRAL SYSTEM опрашивает эти сервера и сохраняет полученные данные

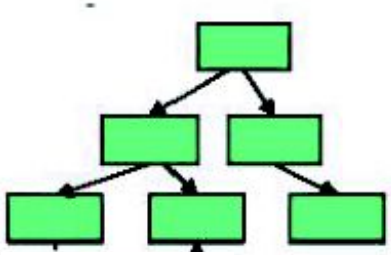
3. Эта информация доступна через заданный ACCESS PROTOCOL

4. Центральная система предоставляет информацию в соответствии со SCHEMA, которая определяет стандарт на описание сетевых ресурсов

BDII является текущей EGEE/LCG Информационной Системой и основывается на LDAP

Модель данных

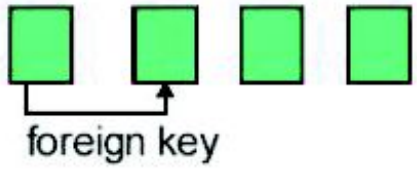
(a)



Иерархическая – структура типа дерево;
потомок имеет только одного родителя.
легко разделяется (partitions); легко отображается на физические устройства.

BDII, LDAP

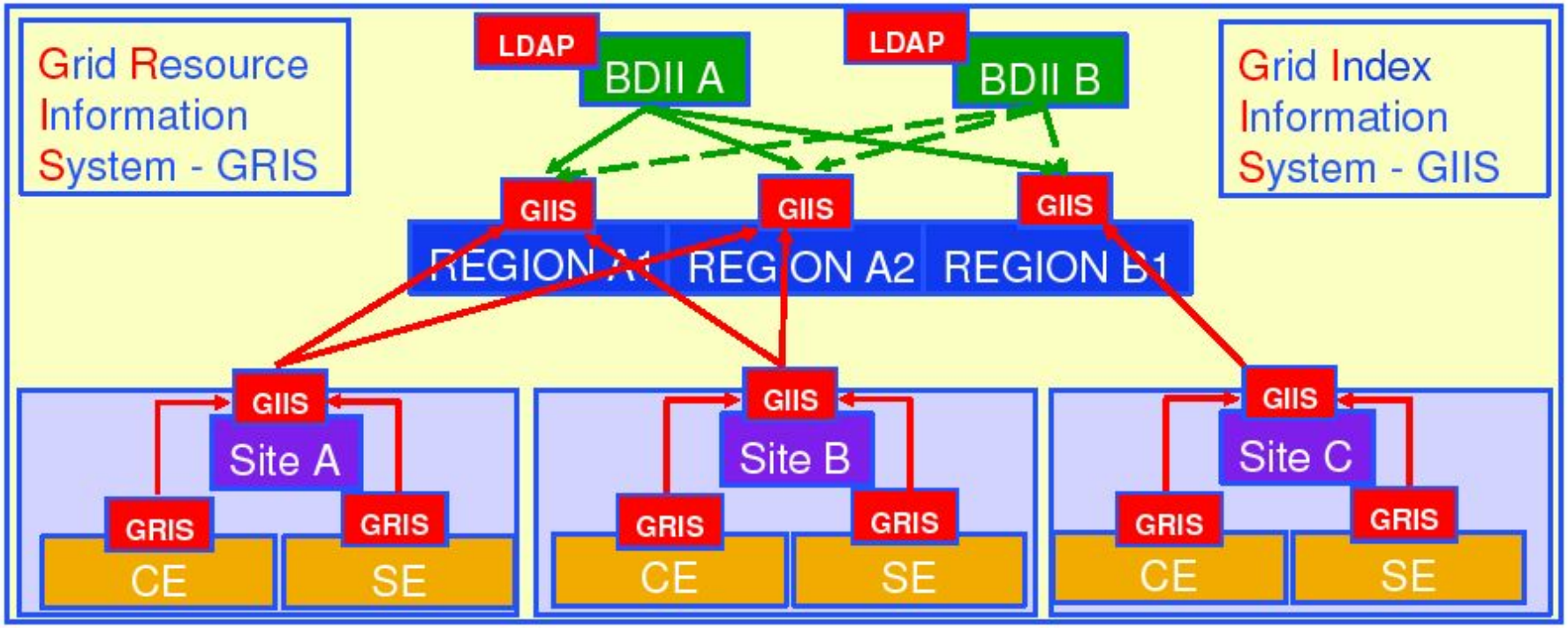
(b)



Реляционная – набор таблиц;
Язык запросов (SQL) эффективный,
широко распостраненный

R-GMA

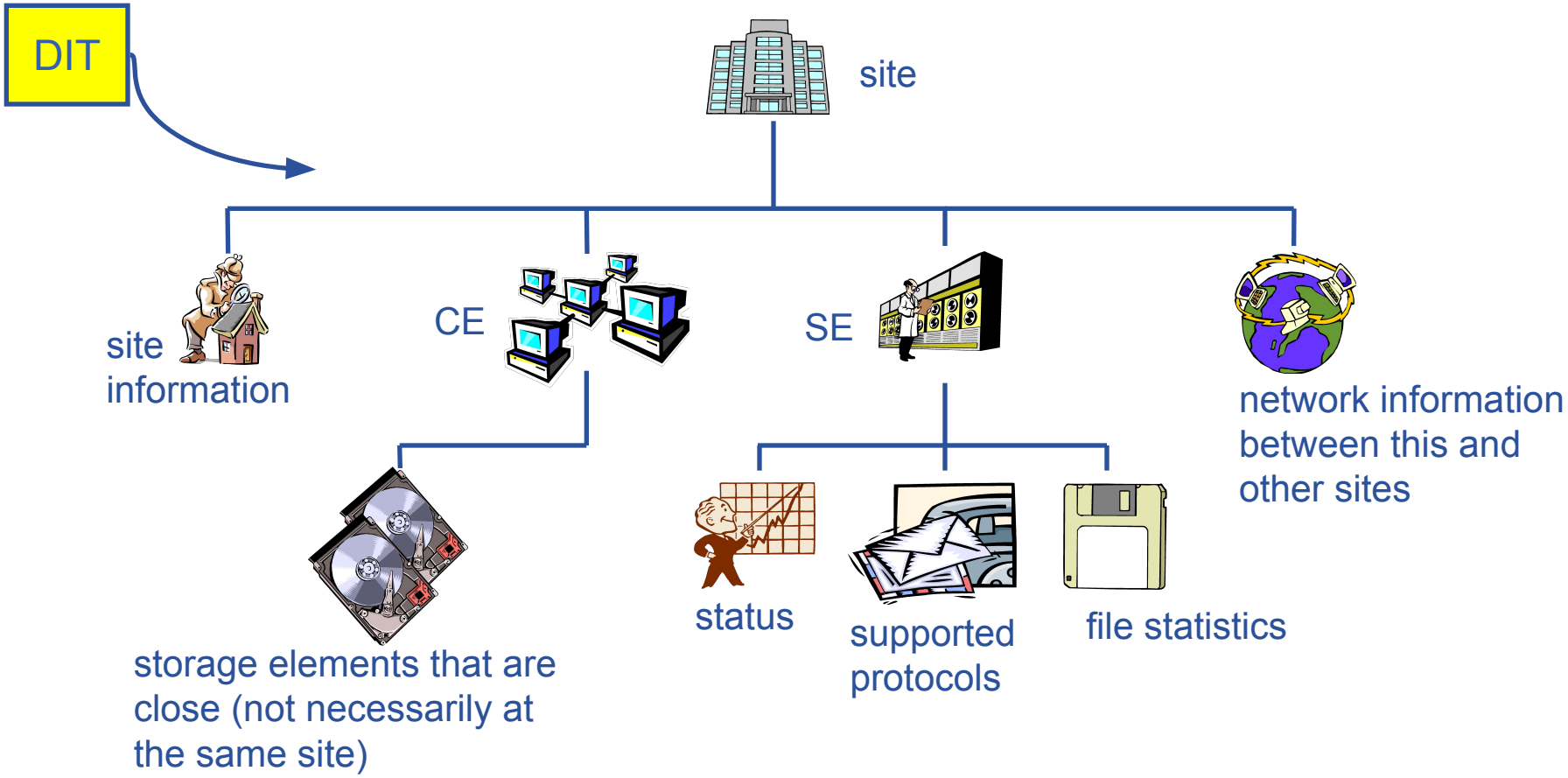
LCG IS



-- Иерархическая модель предоставления данных в информационной системе: **CE, SE GRIS GIIS BDII** (GIIS в настоящее время заменяется на **BDII**)

-- Ресурсы описываются при помощ и GLUE Schema.

Information Providers



Some examples of the Glue Schema (I)

ATTRIBUTES FOR THE CE

- Base Class for the CE information (`objectclass: GlueCETop`) : No attributes
- CE (`objectclass: GlueCE`)
  - `GlueCEUniqueID`: unique identifier for the CE
  - `GlueCEName` : human-readable name of the service
- CE Status (`objectclass: GlueCEState`)
  - `GlueCEStateRunningJobs` : number of running jobs
  - `GlueCEStateWaitingJobs` : number of jobs not running
  - `GlueCEStateTotalJobs` : total number of jobs (running + waiting)
  - `GlueCEStateStatus` : queue status: queueing (jobs accepted but not running), production (jobs accepted and run), closed (neither accepted nor run), draining (jobs not accepted but those already queued are running)
  - `GlueCEStateWorstResponseTime` : worst possible time between the submission of the job and the start of its execution

Some examples of the Glue Schema (II)

3. ATTRIBUTES FOR THE SE

- Base Class (`objectclass: GlueSETop`) : No attributes
- Architecture (`objectclass: GlueSLArchitecture`)
  - `GlueSLArchitectureType`: type of storage hardware (disk, tape, etc)
- Storage Service Access Protocol (`objectclass: GlueSEAccessProtocol`)
  - `GlueSEAccessProtocolType`: protocol type to access or transfer files
  - `GlueSEAccessProtocolPort`: port number for the protocol
  - `GlueSEAccessProtocolVersion`: protocol version
  - `GlueSEAccessProtocolAccessTime`: time to access a file using this protocol

4. MIXED ATTRIBUTES

- Association between one CE and one or more SEs (`objectclass: GlueCESEBindGroup`)
  - `GlueCESEBindGroupCEUniqueID`: unique ID for the CE
  - `GlueCESEBindGroupSEUniqueID`: unique ID for the SE

R-GMA

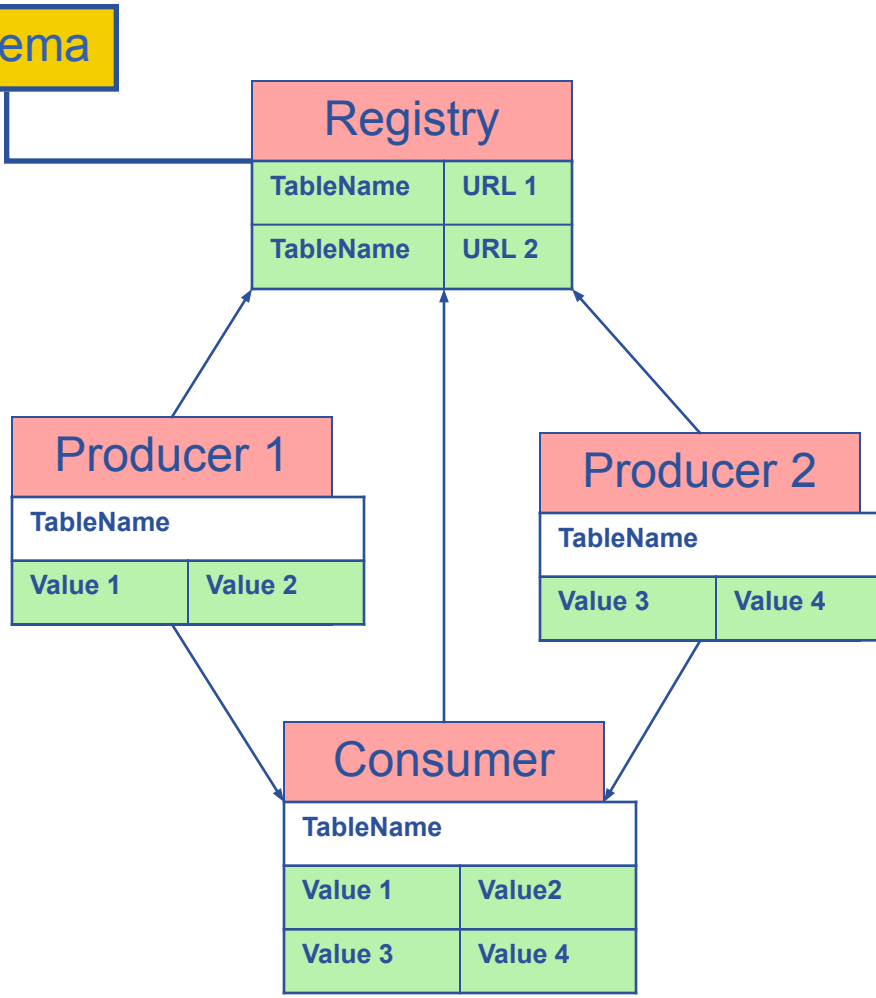
- LDAP не поддерживает агрегатные запросы на различные объекты, т.е. запрос основывается на атрибутах объекта.

R-GMA: Relation Grid Monitoring Archicecture

- **Использует реляционную модель данных.**
  - Данные представляются в виде таблиц.
  - Структура данных определяется по колонкам.
  - Каждая запись есть строка (tuple).
  - Язык запросов - Structured Query Language (SQL).
- **Поддерживает различные типы запросов:**
  - streams
  - archives
  - lates-value

Архитектура R-GMA

- Все **Producers** регистрируются в **Registry**, определяя **Schema**
- **Consumer** получает из **Registry** те URLs, которые могут выполнить его запрос.
- **Consumer** взаимодействует с этими **Producers**.
- **Producers** обрабатывают **query** и возвращают tuples **Consumer**.



Виртуальная база данных

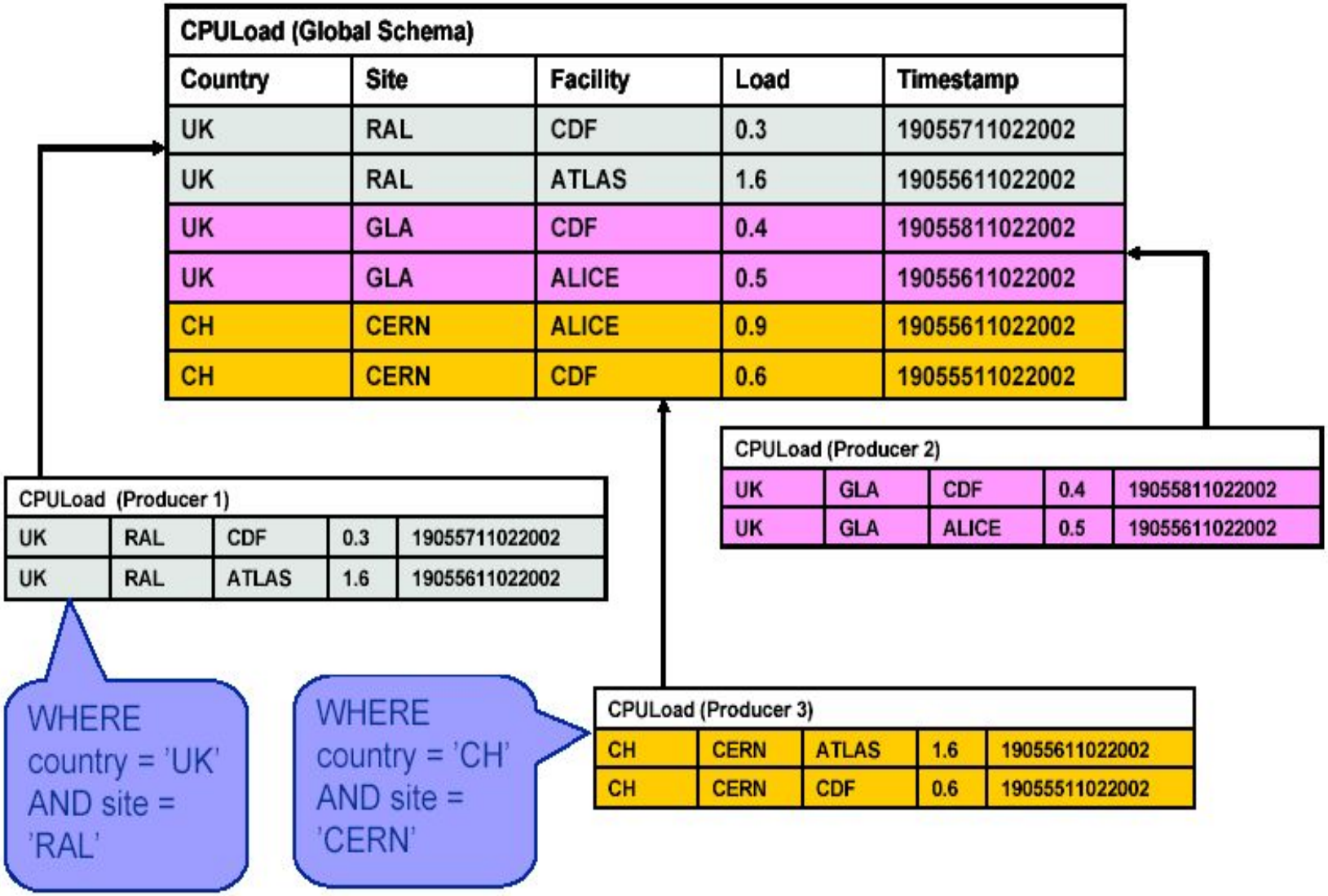
Mediator

- Queries posed against a virtual data base
- The Mediator must:

  -find the right Producers

  -combine information from them
- Hidden component – but vital to R-GMA
- Will eventually support full distributed queries but for now will only merge information from multiple producers for queries on one table or over multiple tables from one producer

Information and Monitoring



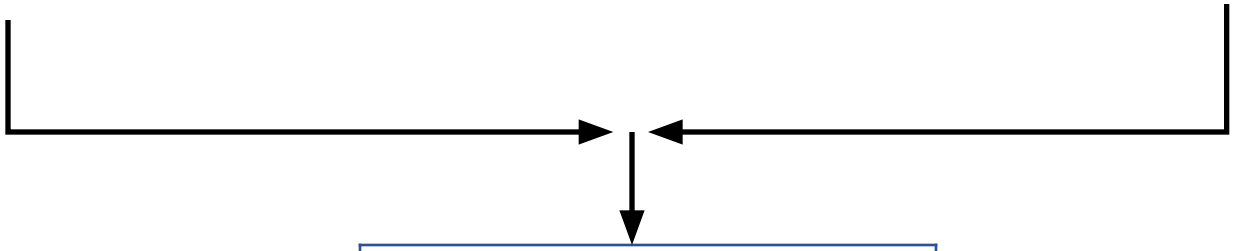
Information and Monitoring

Service

URI	VO	type	emailContact	site
gppse01	alice	SE	sysad@rl.ac.uk	RAL
gppse01	atlas	SE	sysad@rl.ac.uk	RAL
gppse02	cms	SE	sysad@rl.ac.uk	RAL
lxshare0404	alice	SE	sysad@cern.ch	CERN
lxshare0404	atlas	SE	sysad@cern.ch	CERN

ServiceStatus

URI	VO	type	up	status
gppse01	alice	SE	y	SE is running
gppse01	atlas	SE	y	SE is running
gppse02	cms	SE	n	SE ERROR 101
lxshare0404	alice	SE	y	SE is running
lxshare0404	atlas	SE	y	SE is running

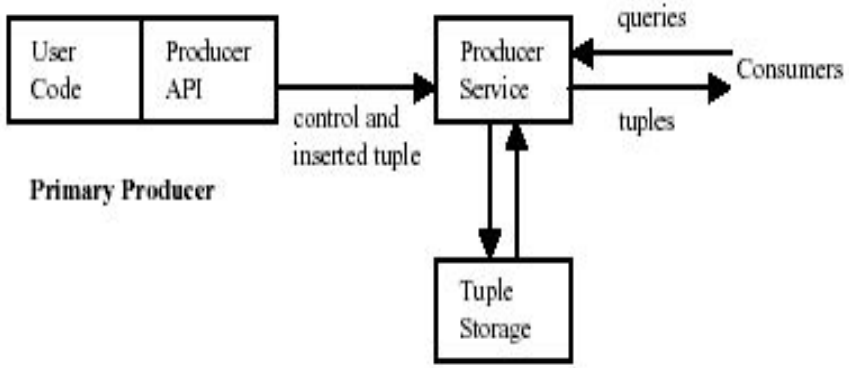


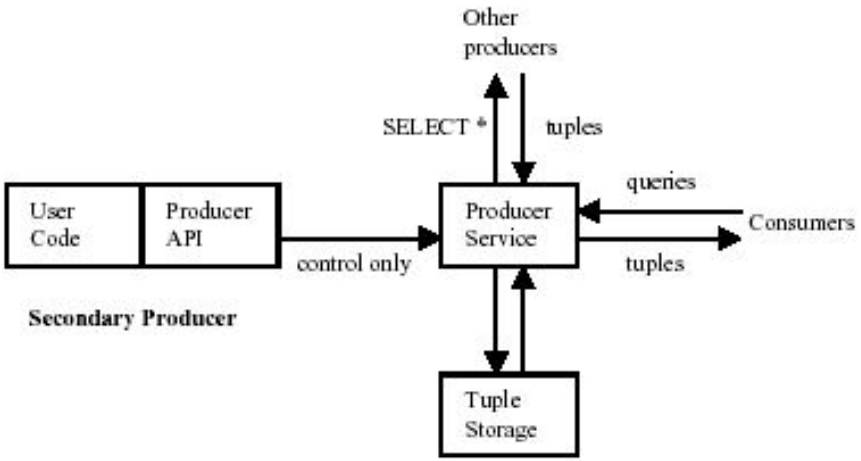
Result Set (Consumer)

URI	emailContact
gppse02	sysad@rl.ac.uk

SELECT Service.URI Service.emailContact FROM Service S, ServiceStatus SS
WHERE (S.URI= SS.URI and SS.up='n')

R-GMA: Producers



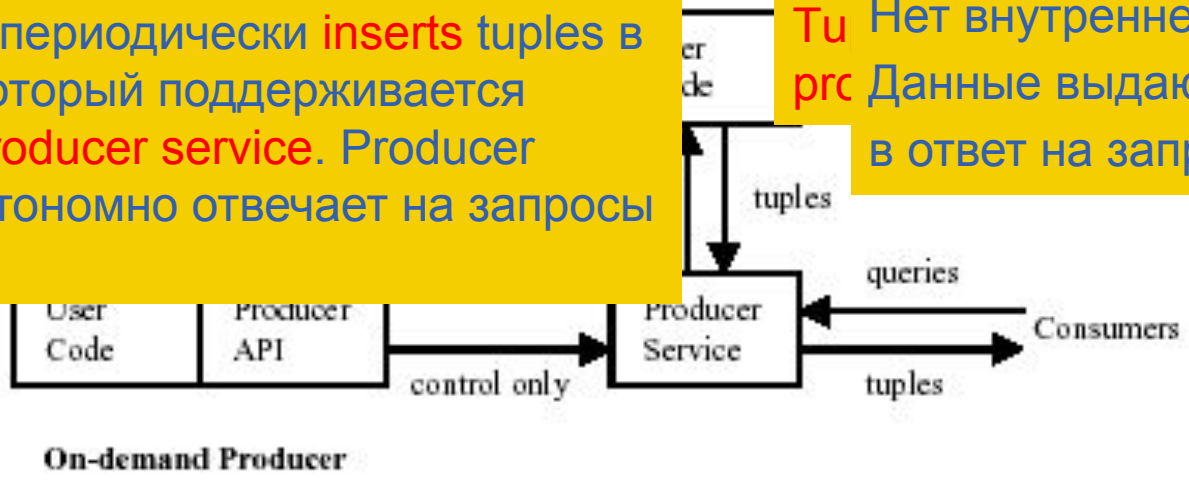


Primary Producer

user code периодически inserts tuples в storage, который поддерживается Primary Producer service. Producer service автономно отвечает на запросы consumer.

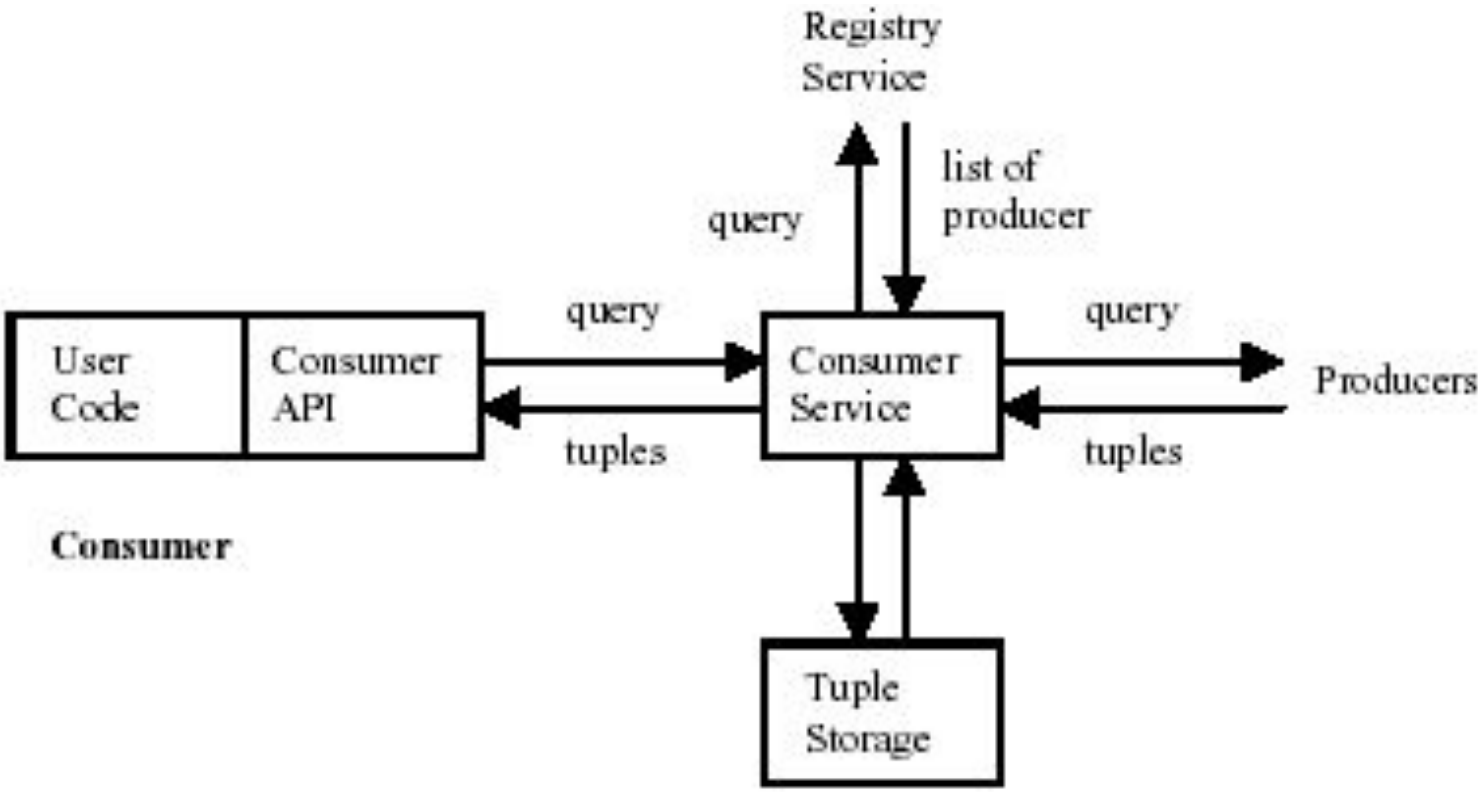
On-demand Producer

Нет внутреннего хранилища, Данные выдаются User Code в ответ на запрос



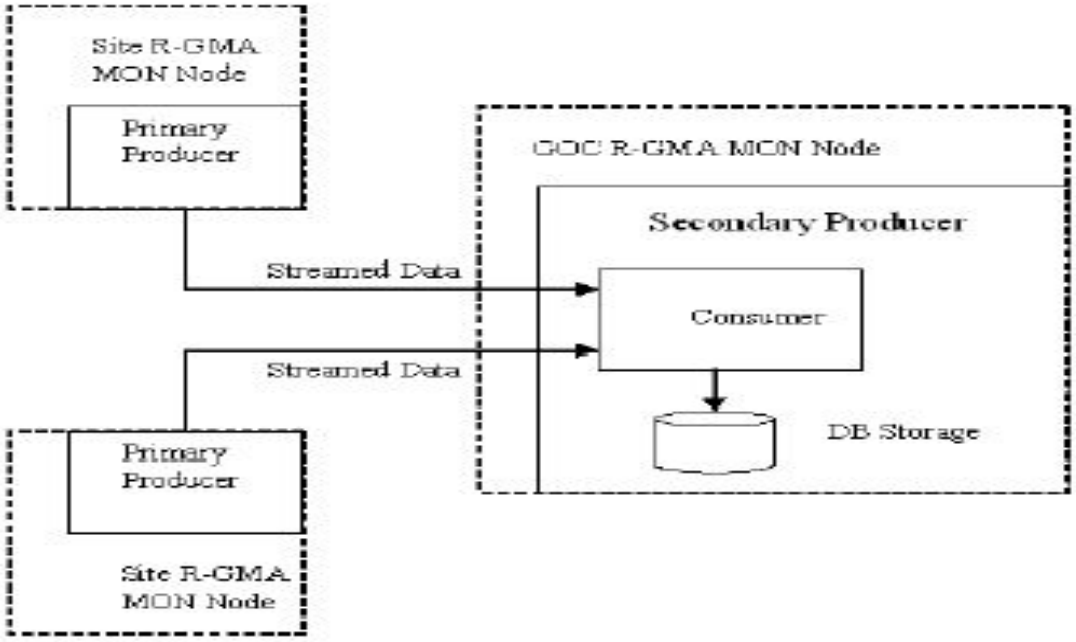
On-demand Producer

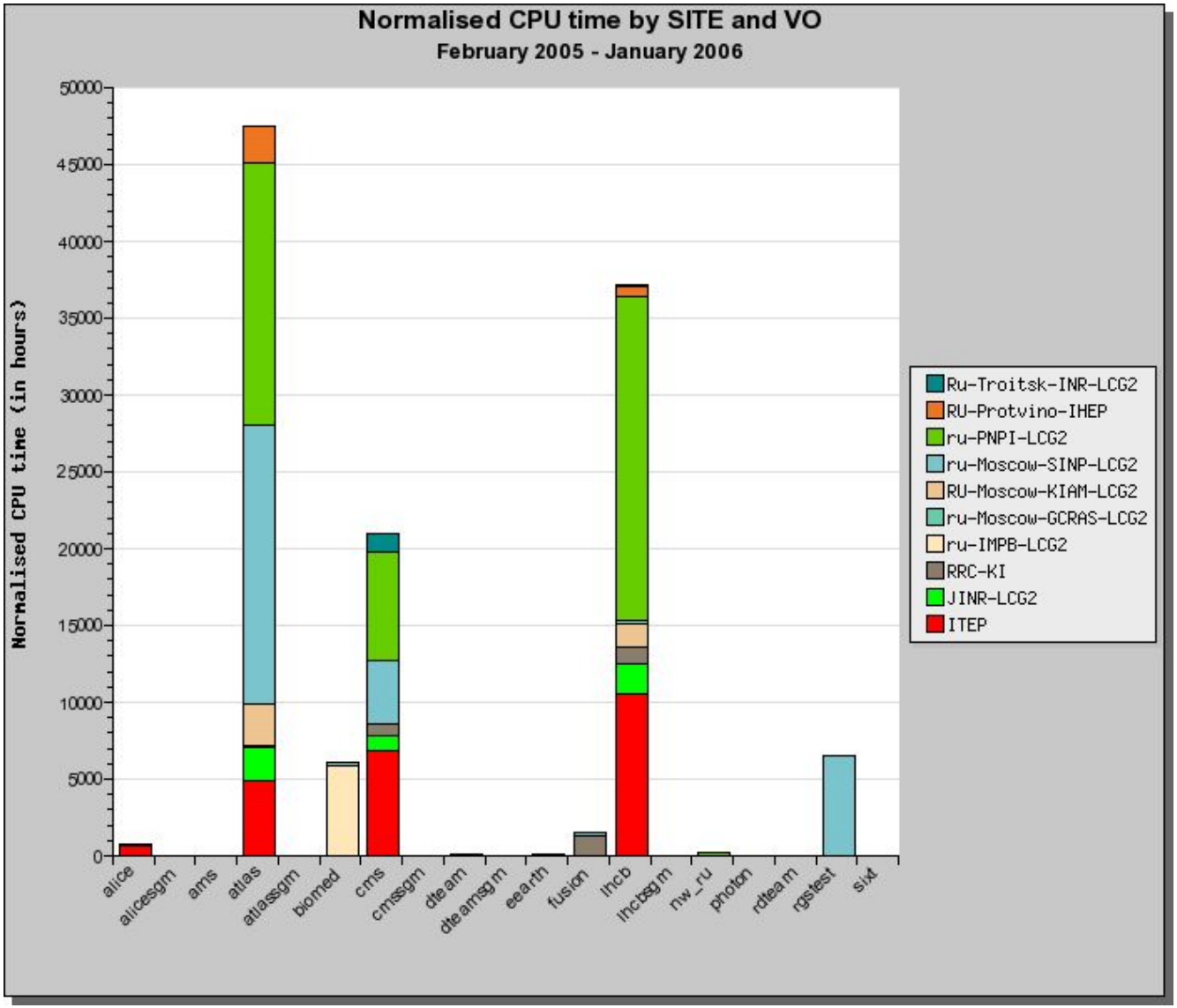
R-GMA: Consumer



R-GMA in Accounting

- Application runs on a site to process GateKeeper and PBS log files
  - Log files contain the User ID, Job Number and Resources used
  - Parsed log files are then published into a schema using R-GMA
- Log file data merged into one large accounting record
- R-GMA used to publish all accounting records per site
- Published records streamed to a Secondary Producer
  - Located on the GOC – aggregates records from all sites
  - Accounting records stored in a database

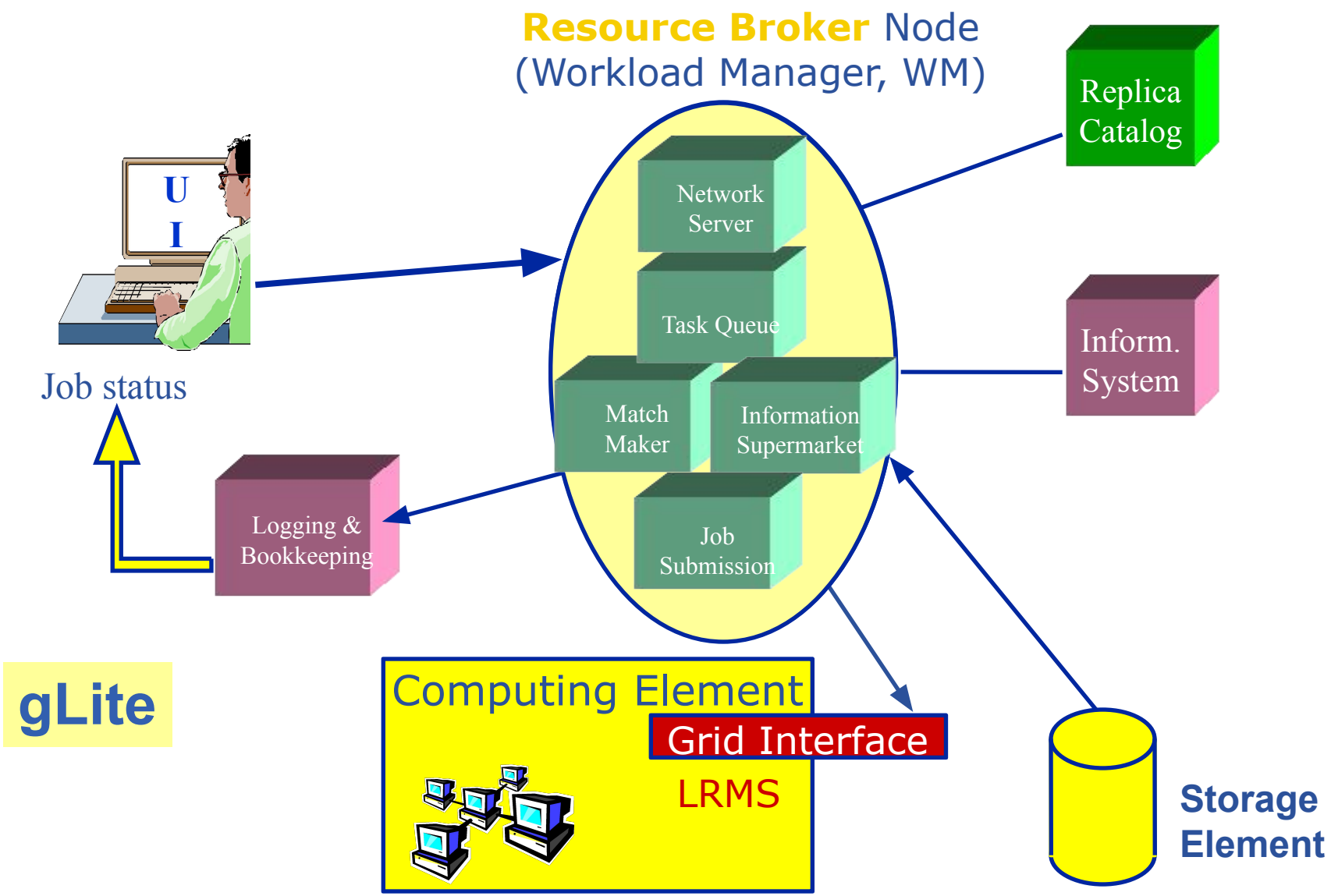



Normalised CPU time by SITE and VO
February 2005 - January 2006
Normalised CPU time (in hours)
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
alice
alicesgm
ams
atlas
atlassgm
biomed
cms
cmssgm
dteam
dteamsgm
eearth
fusion
lhcb
lhcbsgm
nw_ru
photon
rdteam
rgstest
sixt
Ru-Troitsk-INR-LCG2
RU-Protvino-IHEP
ru-PNPI-LCG2
ru-Moscow-SINP-LCG2
RU-Moscow-KIAM-LCG2
ru-Moscow-GCRAS-LCG2
ru-IMPB-LCG2
RRC-KI
JINR-LCG2
ITEP

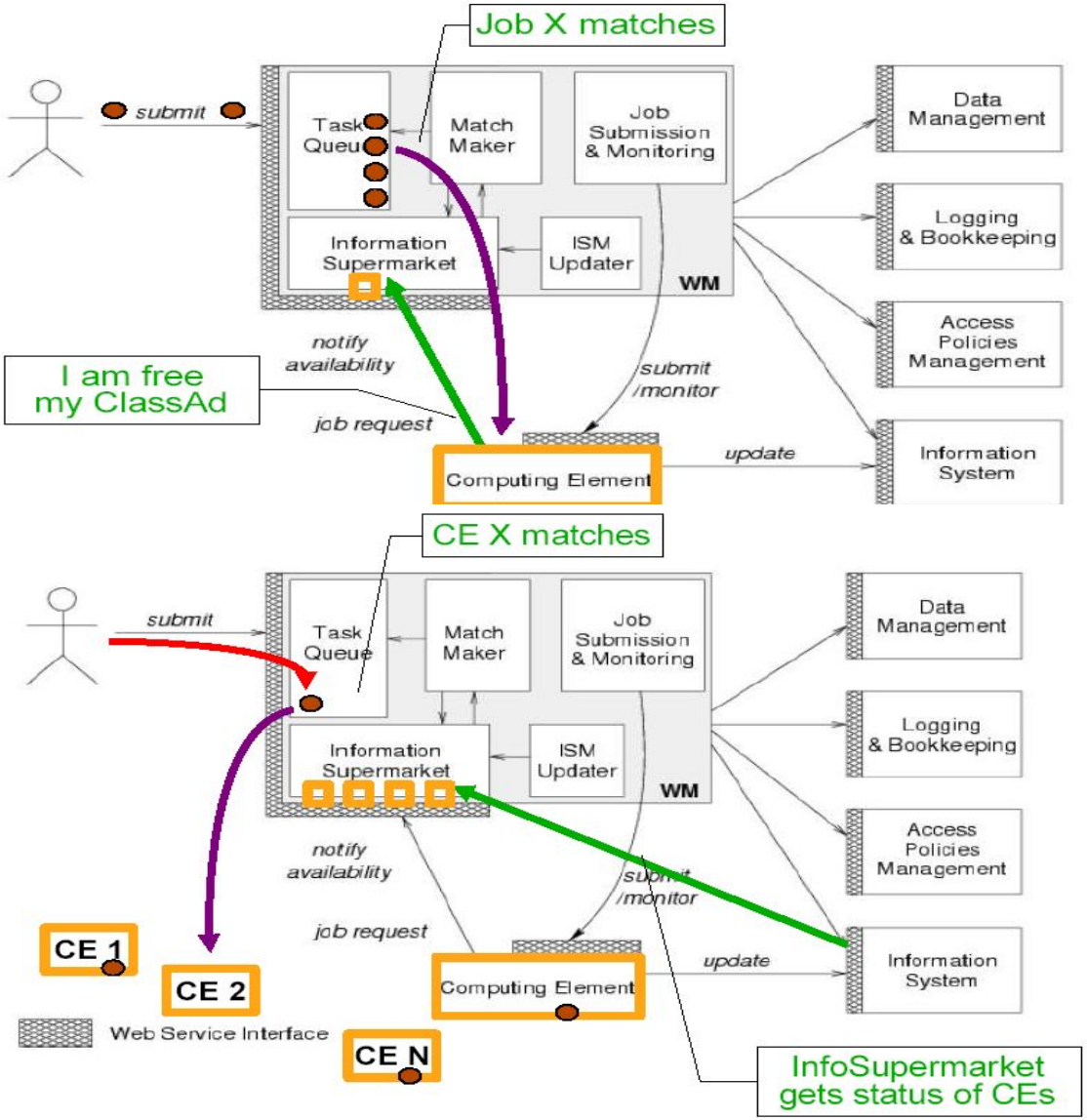
Job Management Services

- Сервисы управления заданиями (Job Management Services)
  - computing element
    - *job management (запуск и управление заданиями)*
    - *Информирование о своих характеристиках и статусе*
  - workload management
    - *управление запуском заданий*
  - accounting
    - computing, storage and network resources
  - job provenance
    - *Сохранение данных о запущенных заданиях, условий выполнения и окружения и т.п. На длительный период времени*
      - debugging, post-mortem analysis, comparison of job execution
  - package manager
    - *automates the process of installing, upgrading, configuring, and removing software packages from a shared area on a grid site.*
      - extension of a traditional package management system to a Grid

WMS Architecture Overview



WMS's Scheduling Policies



Lazy scheduling
(pool mode)

Eager scheduling
(push mode)

The Information Supermarket

- ISM represents one of the most notable improvements in the WM as inherited from the EU DataGrid (EDG) project
  - **decoupling between the collection of information concerning resources and its use**
    - **allows flexible application of different policies**
- The ISM basically consists of a repository of resource information that is available in read only mode to the matchmaking engine
  - **the update is the result of**
    - **the arrival of notifications**
    - **active polling of resources**
    - **some arbitrary combination of both**
  - **can be configured so that certain notifications can trigger the matchmaking engine**
    - **improve the modularity of the software**
    - **support the implementation of lazy scheduling policies**

The Task Queue

- The Task Queue
  - **Возможность сохранения запроса на запуск задания, если отсутствуют ресурсы, удовлетворяющие заданным требованиям (Non-matching requests)**

- Non-matching requests
  - **будут выбираться из очереди или периодически**
    - **eager scheduling**
  - **Или как только нотификация о доступности ресурса появится в ISM**
    - **lazy scheduling**

Logging & Bookkeeping

- L&B tracks jobs in terms of *events*
  - **important points of job life**
    - **submission, finding a matching CE, starting execution etc**
      - ***gathered from various WMS components***
- The events are passed to a physically close component of the L&B infrastructure
  - ***locallogger***
    - **avoid network problems**
      - ***stores them in a local disk file and takes over the responsibility to deliver them further***
- The destination of an event is one of *bookkeeping servers*
  - **assigned statically to a job upon its submission**
    - **processes the incoming events to give a higher level view on the job states**
      - **Submitted, Running, Done**
    - **various recorded attributes**
      - ***JDL, destination CE name, job exit code***
- Retrieval of both job states and raw events is available via legacy (EDG) and WS querying interfaces
  - **user may also register for receiving notifications on particular job state changes**

Job Submission Services

WMS components handling the job during its lifetime and performing the submission

- **Job Adapter**
  - **is responsible for**
    - **making the final touches to the JDL expression for a job, before it is passed to CondorC for the actual submission**
    - **creating the job wrapper script that creates the appropriate execution environment in the CE worker node**
      - ***transfer of the input and of the output sandboxes***
- **CondorC**
  - **responsible for**
    - **performing the actual job management operations**
      - ***job submission, job removal***
- **DAGMan**
  - **meta-scheduler**
    - **purpose is to navigate the graph**
    - **determine which nodes are free of dependencies**
    - **follow the execution of the corresponding jobs.**
- **Log Monitor**
  - **is responsible for**
    - **watching the CondorC log file**
    - **intercepting interesting events concerning active jobs**
      - ***events affecting the job state machine***
    - **triggering appropriate actions.**

Job Description Language (JDL

- **Поддерживаемые атрибуты можно разделить на 2 категории:**
  - Атрибуты задания (Job Attributes)
    - Определяют само задание
  - Ресурсы
    - Используются Workload Manager для matchmaking algorithm (выбрать “наилучший” ресурс для запуска задания)
    - ***Computing Resource***
      - *Используются для определения Requirements и Rank attributes*
    - ***Data and Storage resources***
      - *Input data, Storage Element (SE), где сохранять выходные данные, протоколы доступа к SE*

Example of JDL File

```
[
JobType="Normal";
Executable = "gridTest";
StdError = "stderr.log";
StdOutput = "stdout.log";
InputSandbox = {"/home/mydir/test/gridTest"};
OutputSandbox = {"stderr.log", "stdout.log"};
InputData = {"lfn:/glite/myvo/mylfn" };
DataAccessProtocol = "gridftp";
Requirements = other.GlueHostOperatingSystemNameOpSys
  == "LINUX"
                && other.GlueCEStateFreeCPUs>=4;
Rank = other.GlueCEPolicyMaxCPUTime;
]
```

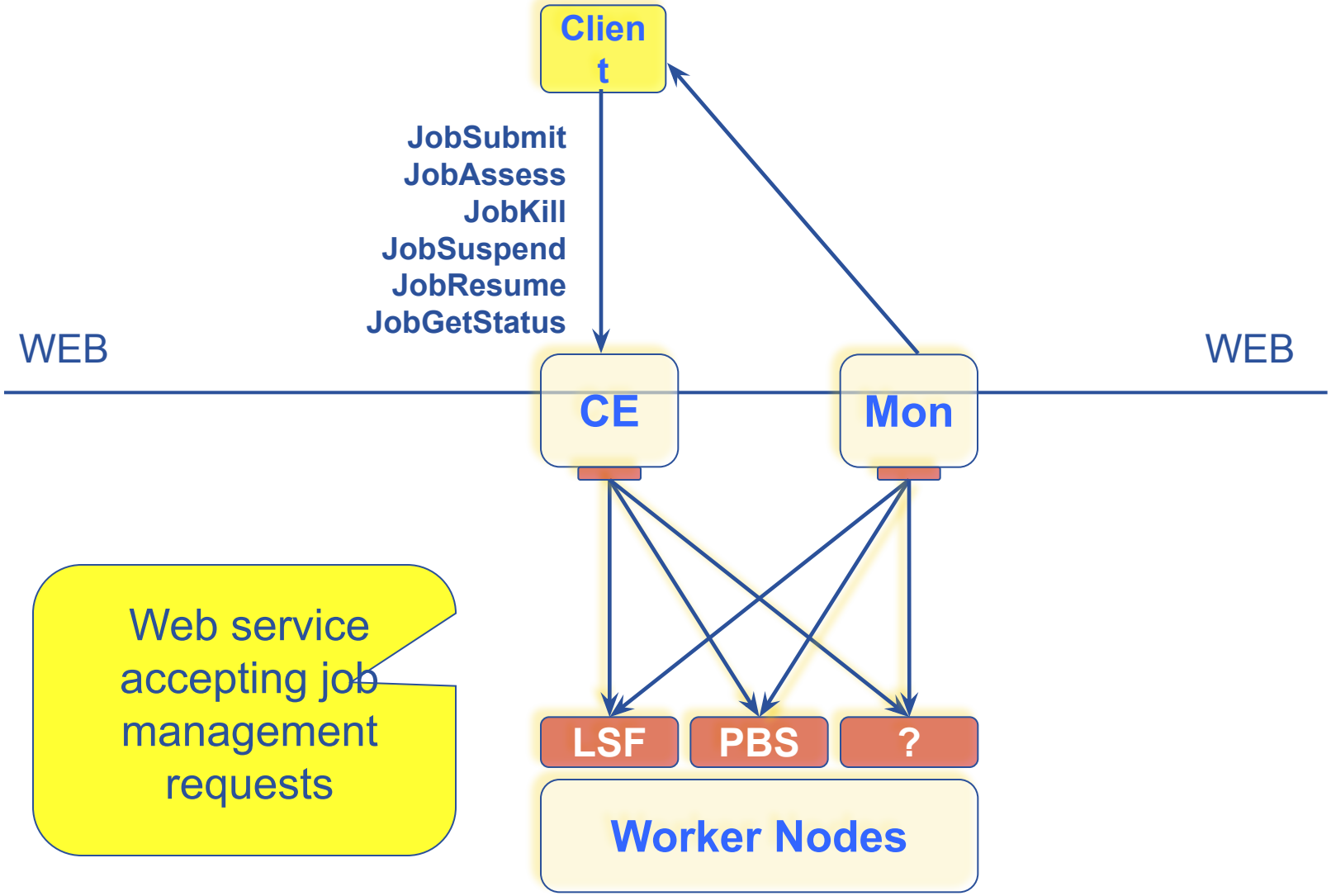
Job Resubmission

- **If something goes wrong, the WMS tries to reschedule and resubmit the job (possibly on a different resource satisfying all the requirements)**
- **Maximum number of resubmissions: min(RetryCount, MaxRetryCount)**
  - RetryCount: JDL attribute
  - MaxRetryCount: attribute in the “RB” configuration file

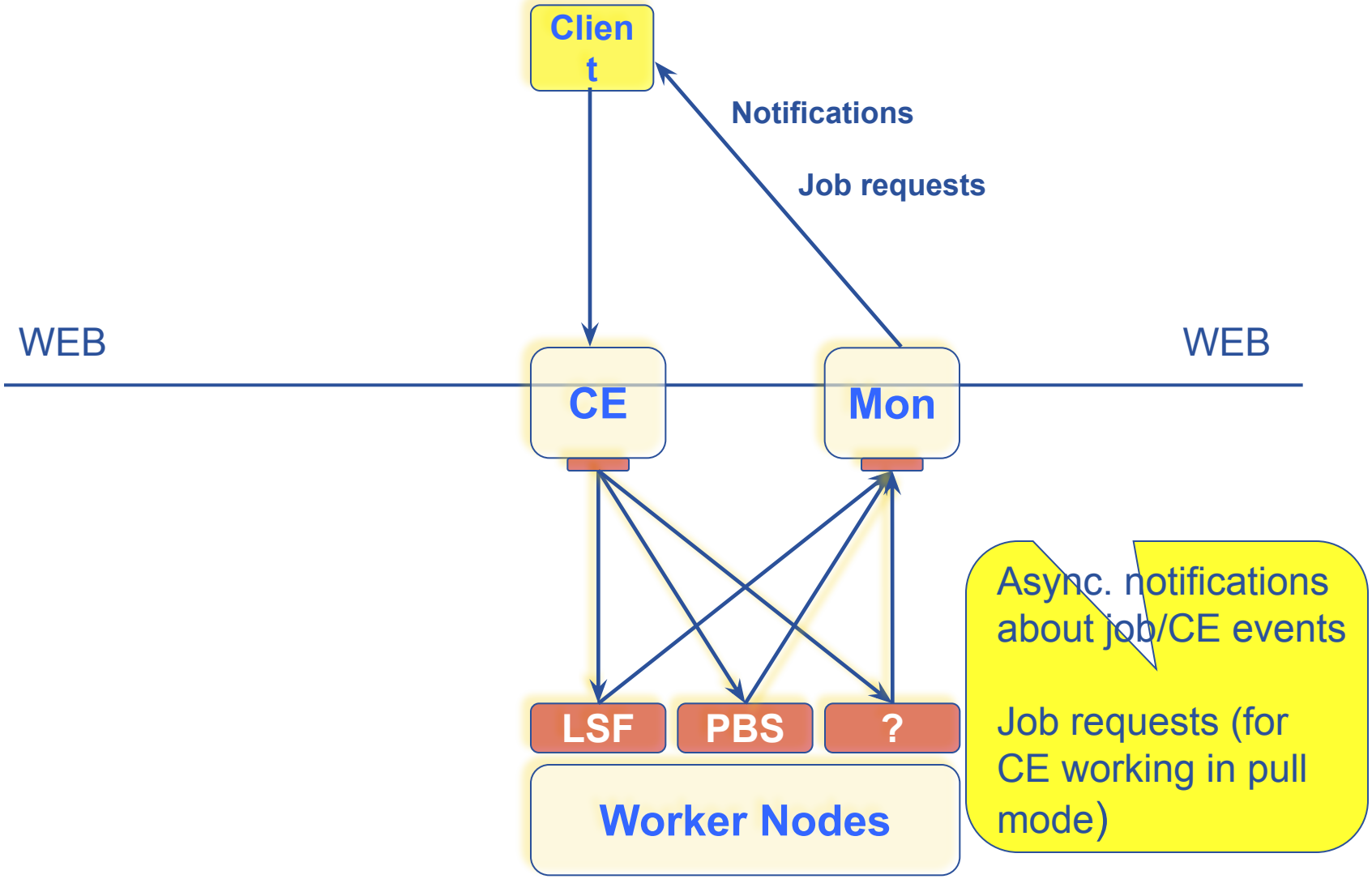
Computing Element

- **Service representing a computing resource**
- **Main functionality: job management**
  - Run jobs
  - Cancel jobs
  - Suspend and resume jobs
  - Provide info on “quality of service”
    - How many resources match the job requirements ?
    - What is the estimated time to have the job starting its execution ?
    - …
  - …
- **Used by the WM or by any other client (e.g. end-user)**
- **CE architecture accommodated to support both push and pull model**
  - Push model: the job is pushed to the CE by the WM
  - Pull model: the CE asks the WM for jobs
- **These two models are somewhat mirrored in the resource information flow**
  - In order to 'pull' a job a resource must choose where to 'push' information about itself

CE Architecture



CE Architecture



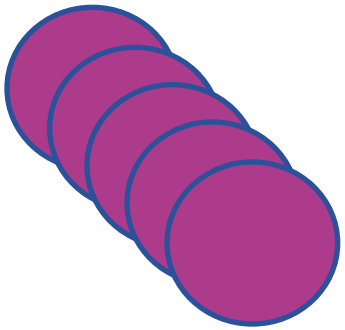
Типы заданий

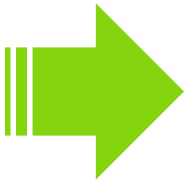
- Normal
- DAG - Directed Acyclic Graphs (DAG)
- MPICH - Message Passing Interface
- Checkpointable Jobs
- Partitionable
- Interactive Jobs
- Collection
- Parametric

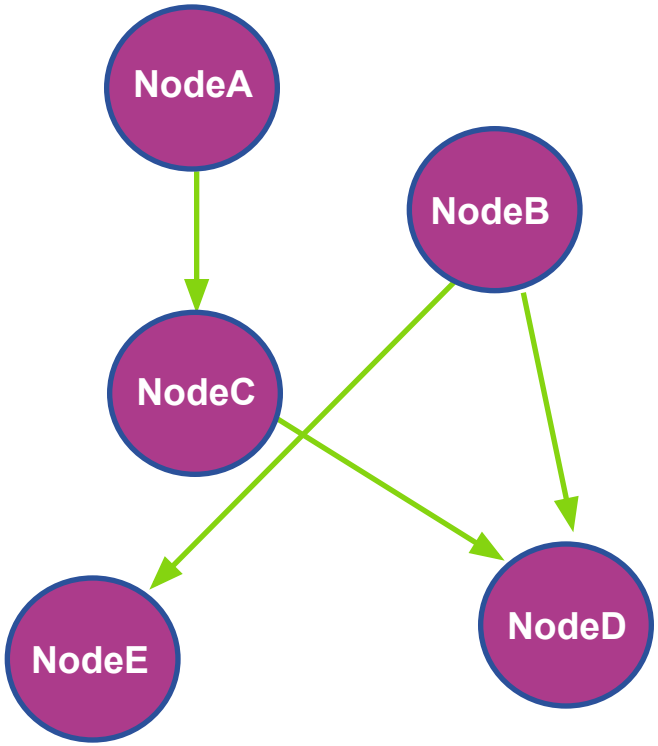
Directed Acyclic Graphs (DAGs)

- A DAG represents a set of jobs:

  Nodes = *Jobs* *Edges* = *Dependencies*

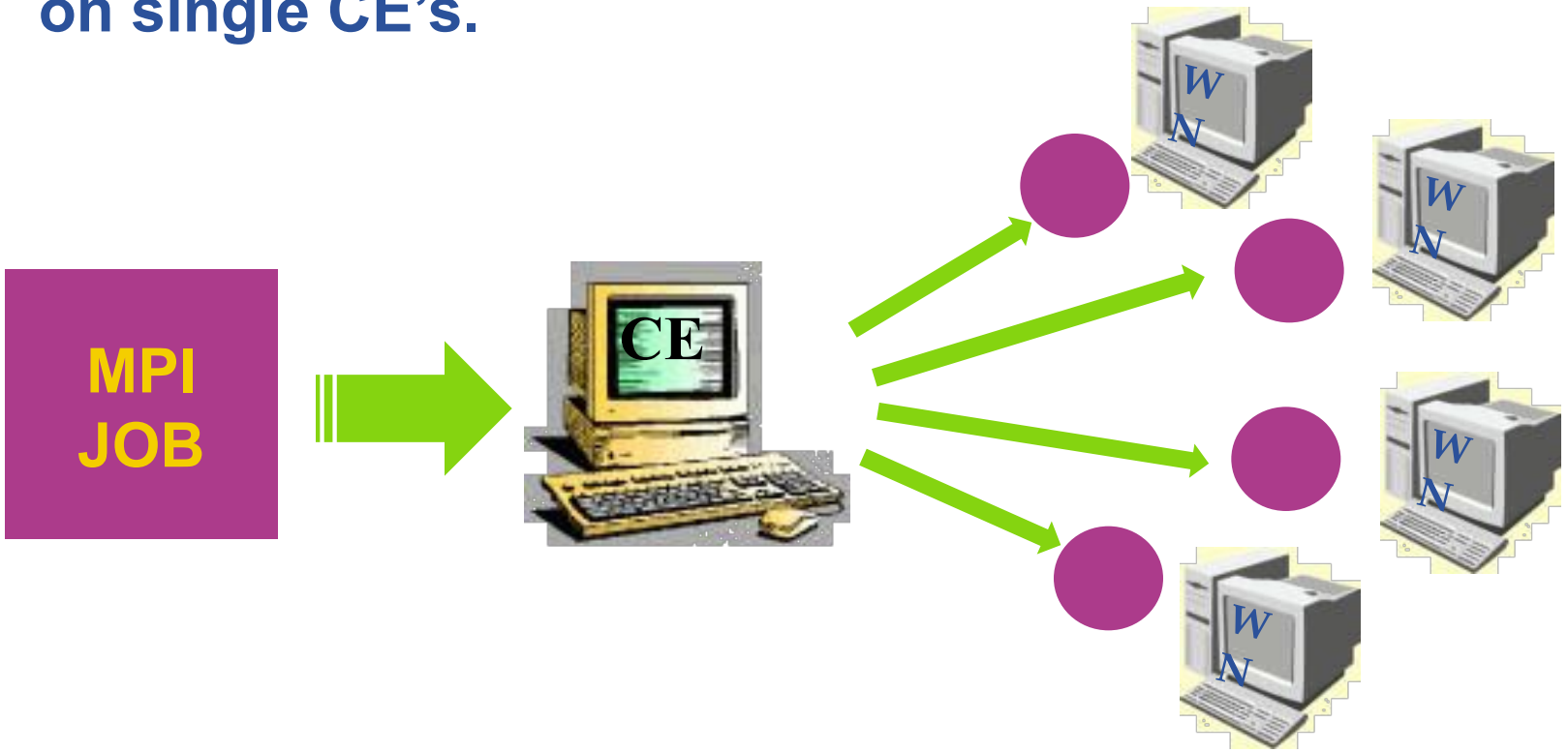






Message Passing Interface (MPI)

- The MPI job is run in parallel on several processors.
- Libraries supported for parallel jobs: MPICH.
- Currently, execution of parallel jobs is supported only on single CE's.



MPI: JDL Structure

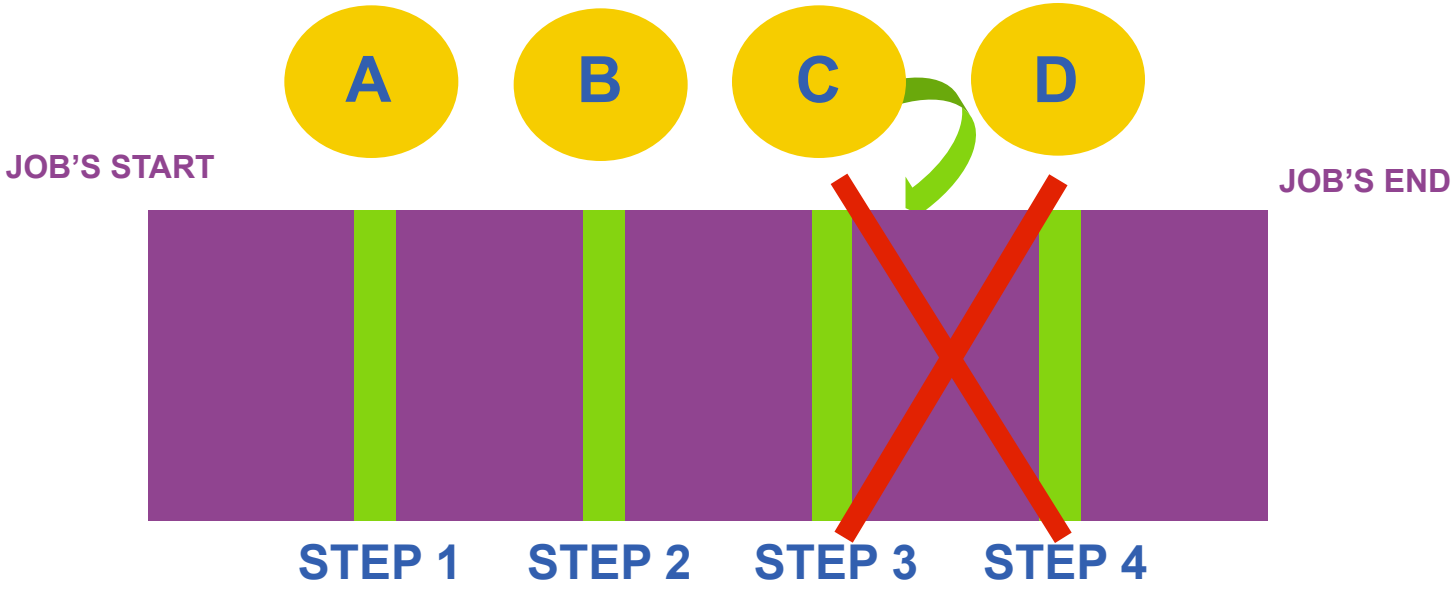
- Type = “job”; → *Mandatory*
- JobType = “MPICH”; → *Mandatory*
- Executable = “…”; → *Mandatory*
- NodeNumber = “int > 1”; → *Mandatory*
- Argument = “…”; → *Optional*
- Requirements = → *Mandatory*

  Member(“MpiCH”, other.GlueHostApplicationSoftwareRunTimeEnvironment)
  && *other.GlueCEInfoTotalCPUs >= NodeNumber* ;

- Rank = *other.GlueCEStateFreeCPUs;* → *Mandatory*

Logical Checkpointable Jobs

- It is a job that can be decomposed in several steps;
- In every step the job state can be saved in the LB and retrieved later in case of failures;
- The job can start running from a previously saved state instead from the beginning again.

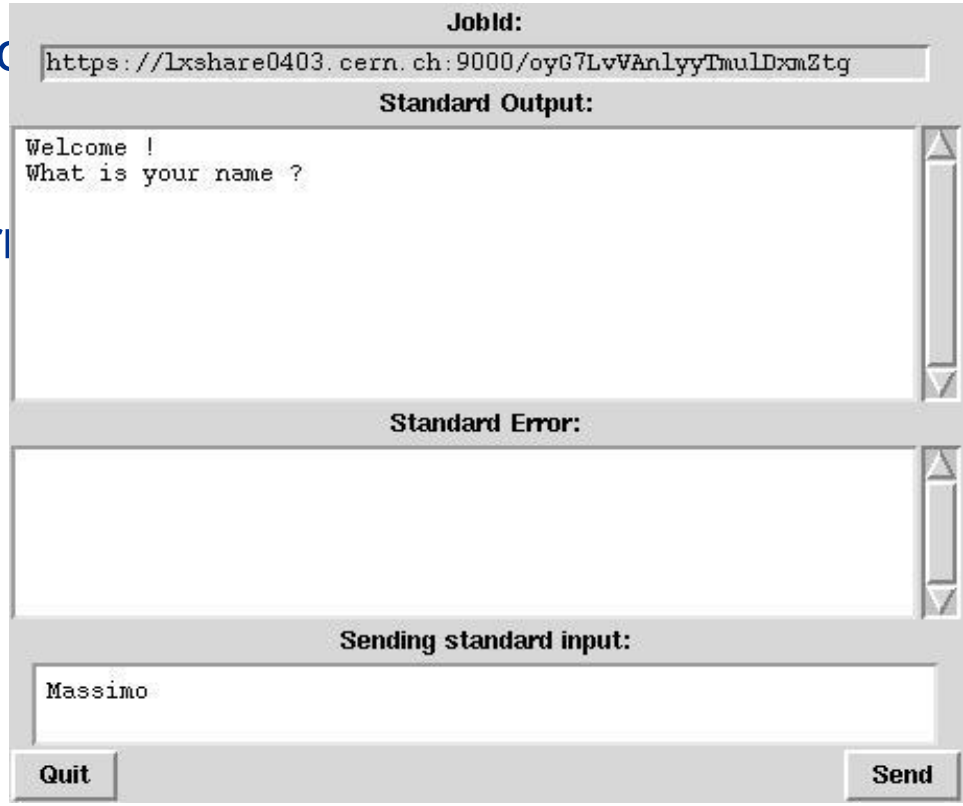


Checkpointable Jobs: JDL Structure

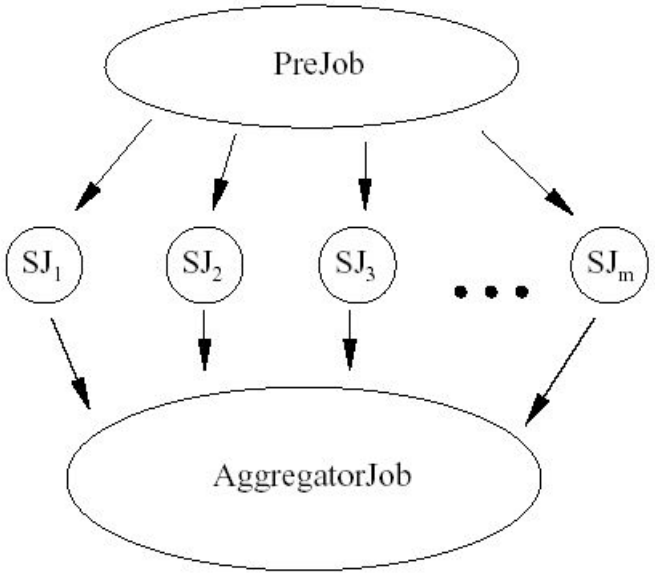
- Type = “job”; → *Mandatory*
- JobType = “checkpointable”; → *Mandatory*
- Executable = “…”; → *Mandatory*
- JobSteps = “list int | list string”; → *Mandatory*
- CurrentStep = “int > = 0”; → *Mandatory*
- Argument = “…”; → *Optional*
- Requirements = “…”; → *Optional*
- Rank = “”; → *Optional*

Interactive Jobs

- **JobType = “Interactive”**
- **When an interactive job is executed, a window for the stdin, stdout, stderr streams is opened**
  - Possibility to send the stdin to the job
  - Possibility the have the stderr and stdout of the job when it is running



Partitionable jobs



- `JobType=Partitionable`
- `JobSteps = {"cms0", "cms1", "cms2", "cms3", "orca"};`
- `StepWeight = {7.5, 25, 37.5, 15, 15};`
- `CurrentStep = 0;`

Parametric jobs

```
JobType = "Parametric";
      Executable = "cms_sim.exe";
      StdInput = "input_PARAM_.txt";
      StdOutput = "myoutput_PARAM_.txt";
      StdError = "myerror_PARAM_.txt";
      Parameters = 10000;
      ParameterStart = 1000;
      ParameterStep = 10;
      InputSandbox = {
"file:///home/cms/cms_sim.exe",
"file:///home/cms/data/input_PARAM_.txt "
};
      OutputSandbox = {
 "myoutput_PARAM_.txt",
 "myerror_PARAM_.txt" };
      Requirements = other.GlueCEInfoTotalCPUs > 2;
      Rank = other.GlueCEStateFreeCPUs;
```