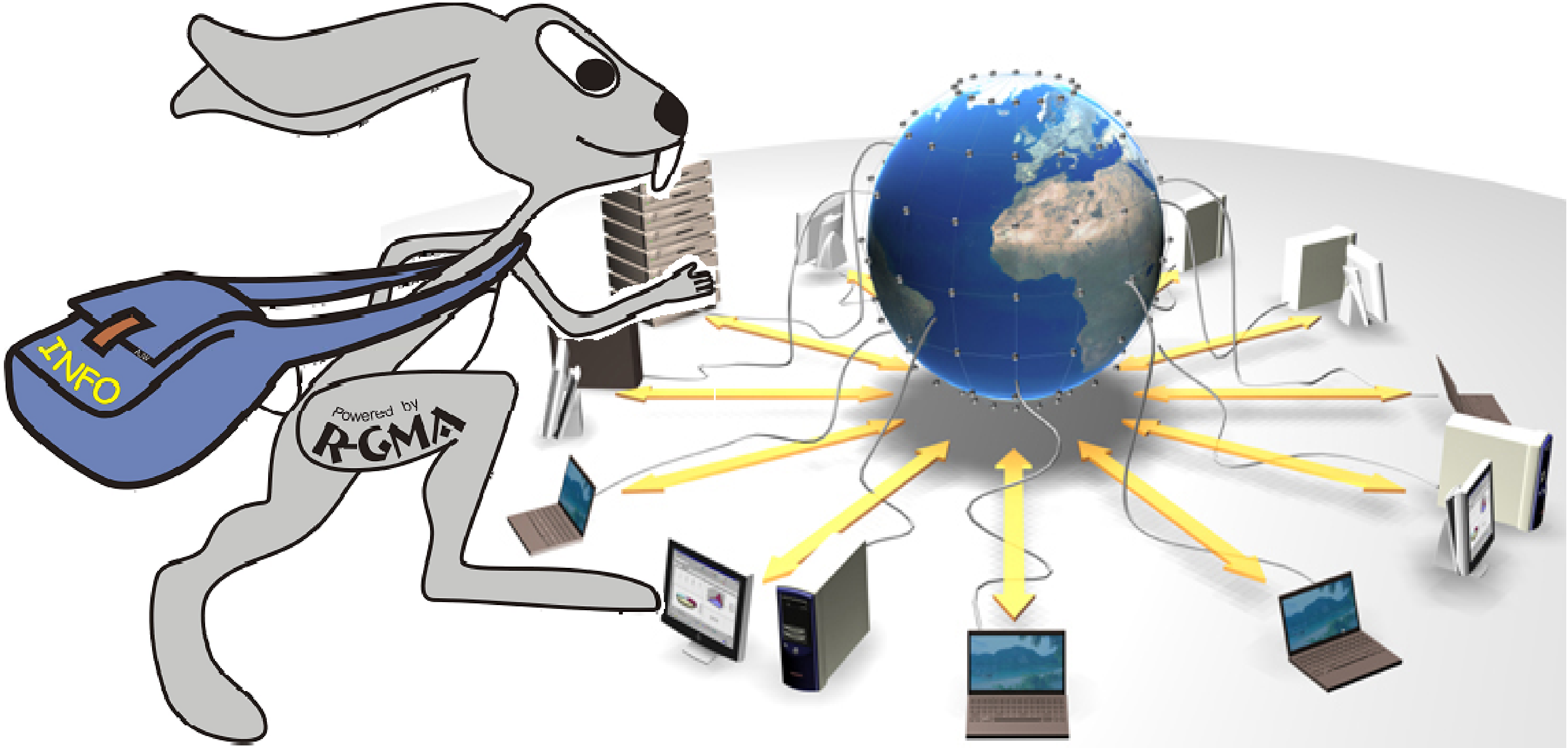
Information and Monitoring

Data GRID

Information Society Technologies

The European DataGrid Project Team

http://www.eu-datagrid.org



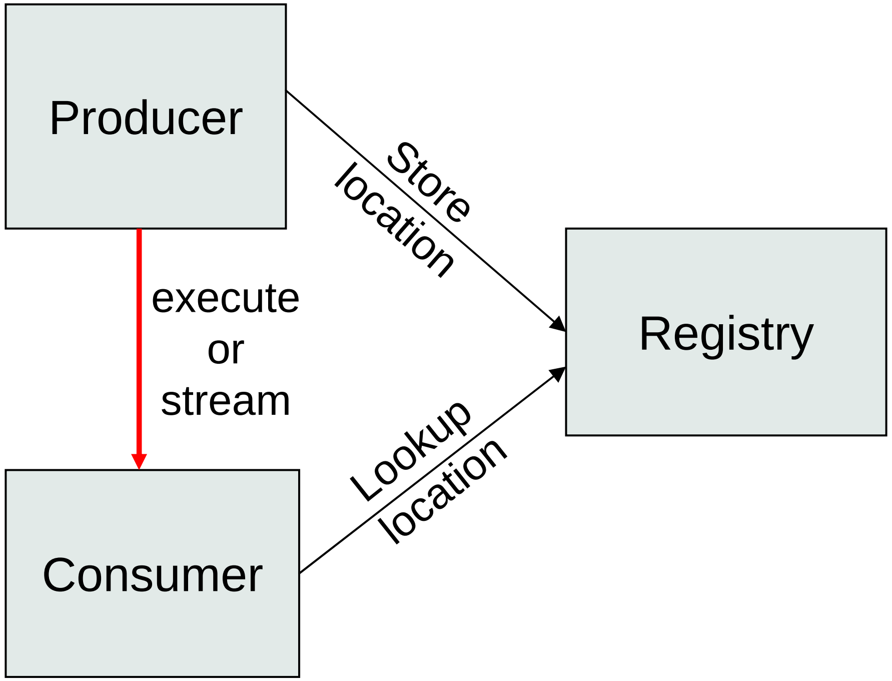
Contents

- Grid Information Systems
- GMA and R-GMA
- Topologies of components
- Monitoring the monitoring system
- Tools and APIs

Features of a grid information system

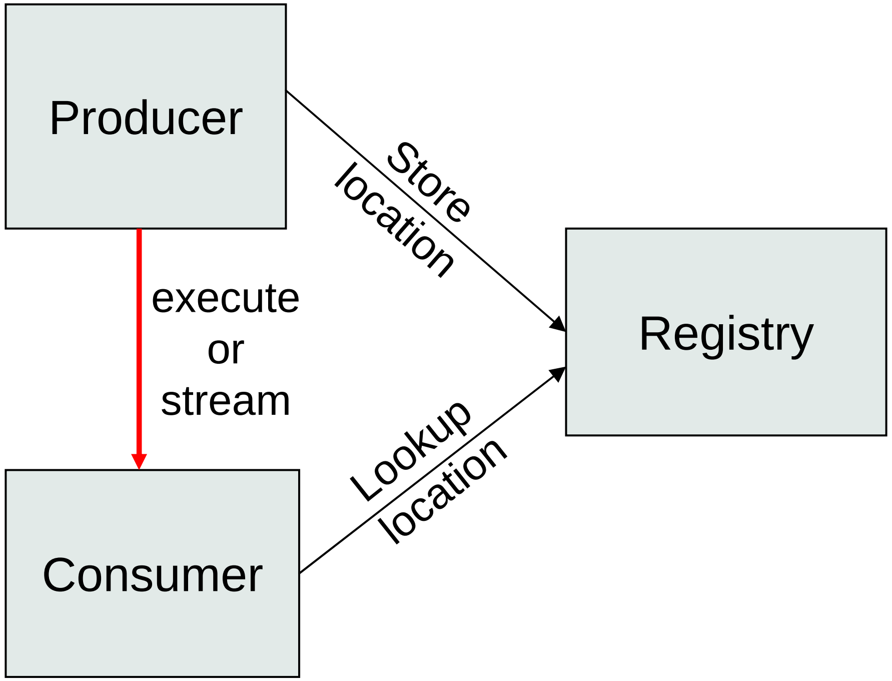
- Provides information on both:
  - The Grid itself
    - Mainly for the middleware packages
    - The user may query it to understand the status of the Grid
  - Grid applications
    - For users
- Flexible infrastructure
  - Able to cope with nodes in a distributed environment with an unreliable network
  - Dynamic addition and deletion of information producers
  - Security system able to address the access to information at a fine level of granularity
  - Allow new data types to be defined
  - Scaleable
  - Good performance
  - Standards based

GMA



- From GGF
- Very simple model
- Does not define:
  - Data model
  - Data transfer mechanism
  - Registry implementation

R-GMA



- Use the GMA from GGF
- A relational implementation
  - Powerful data model and query language
    - All data modelled as tables
    - SQL can express most queries in one expression
- Applied to both information and monitoring
- Creates impression that you have one RDBMS per VO

Relational Data Model in R-GMA

- Not a general distributed RDBMS system, but a way to use the relational model in a distributed environment where global consistency is not important

- Producers announce: SQL “CREATE TABLE”
  publish: SQL “INSERT”

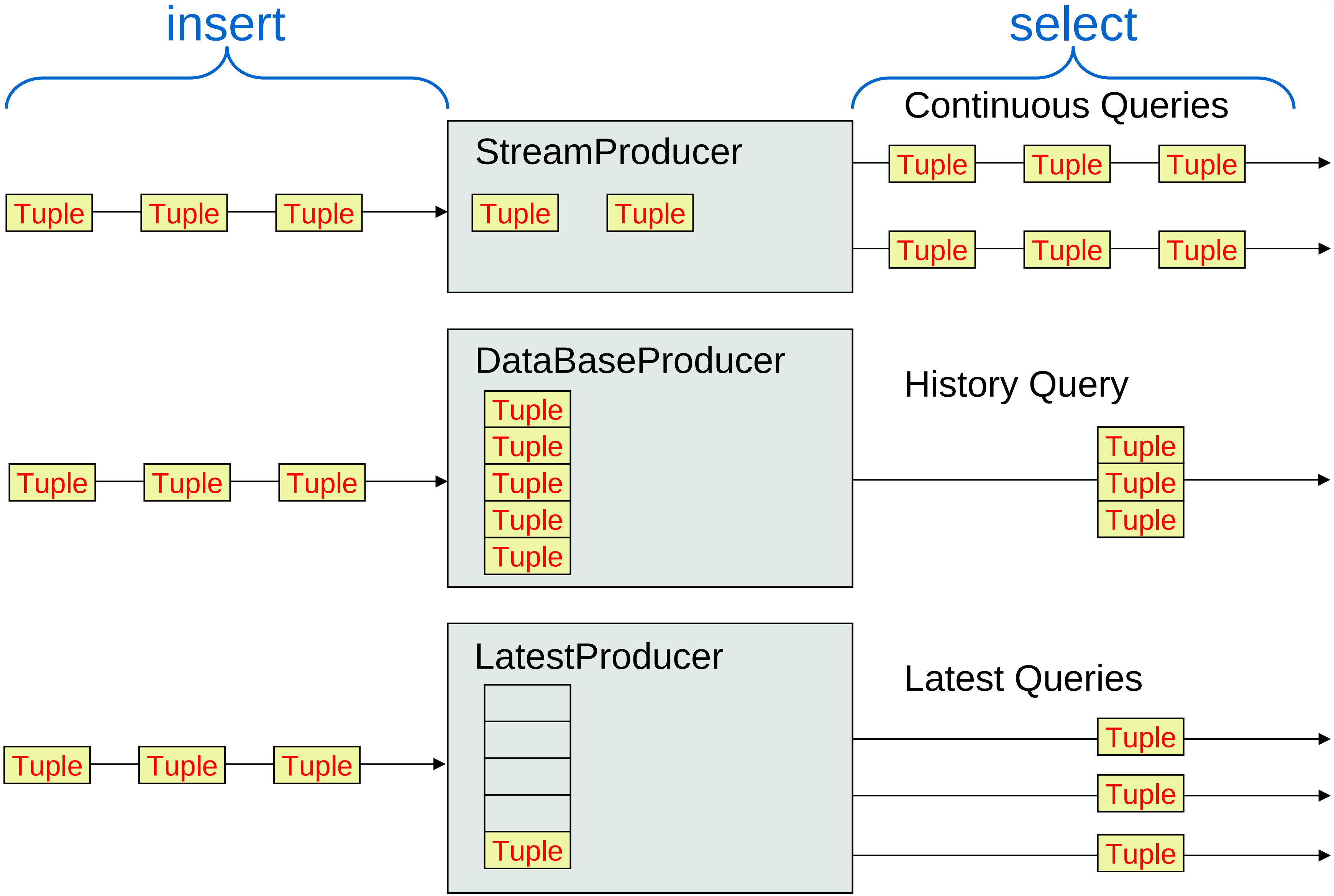
- Consumers collect: SQL “SELECT”

- Some producers, the Registry and Schema make use of RDBMS as appropriate – but what is central is the relational model

Data Transfer: Producer → Consumer

- Consumer can issue one-off queries
  - Similar to normal database query
- Consumer can also start a continuous query
  - Requests all data published which matches the query
    - As data matching the query is produced it is streamed to the Consumer
    - Can be seen as an alert mechanism

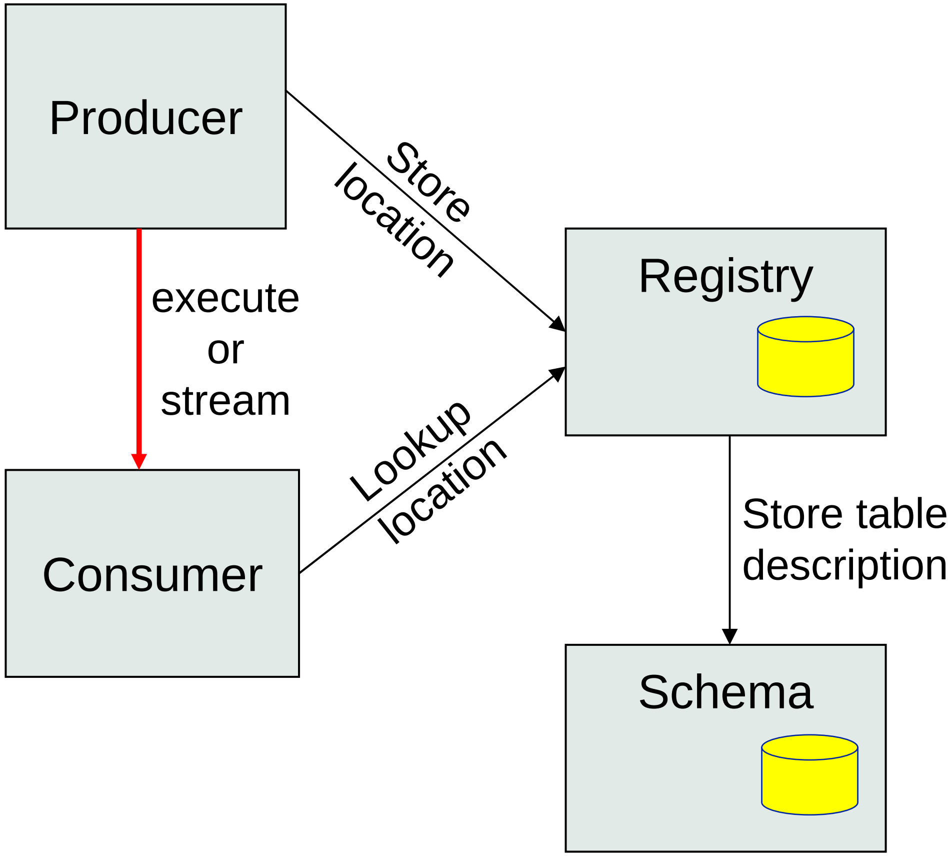
3 Kinds of Query



Producers

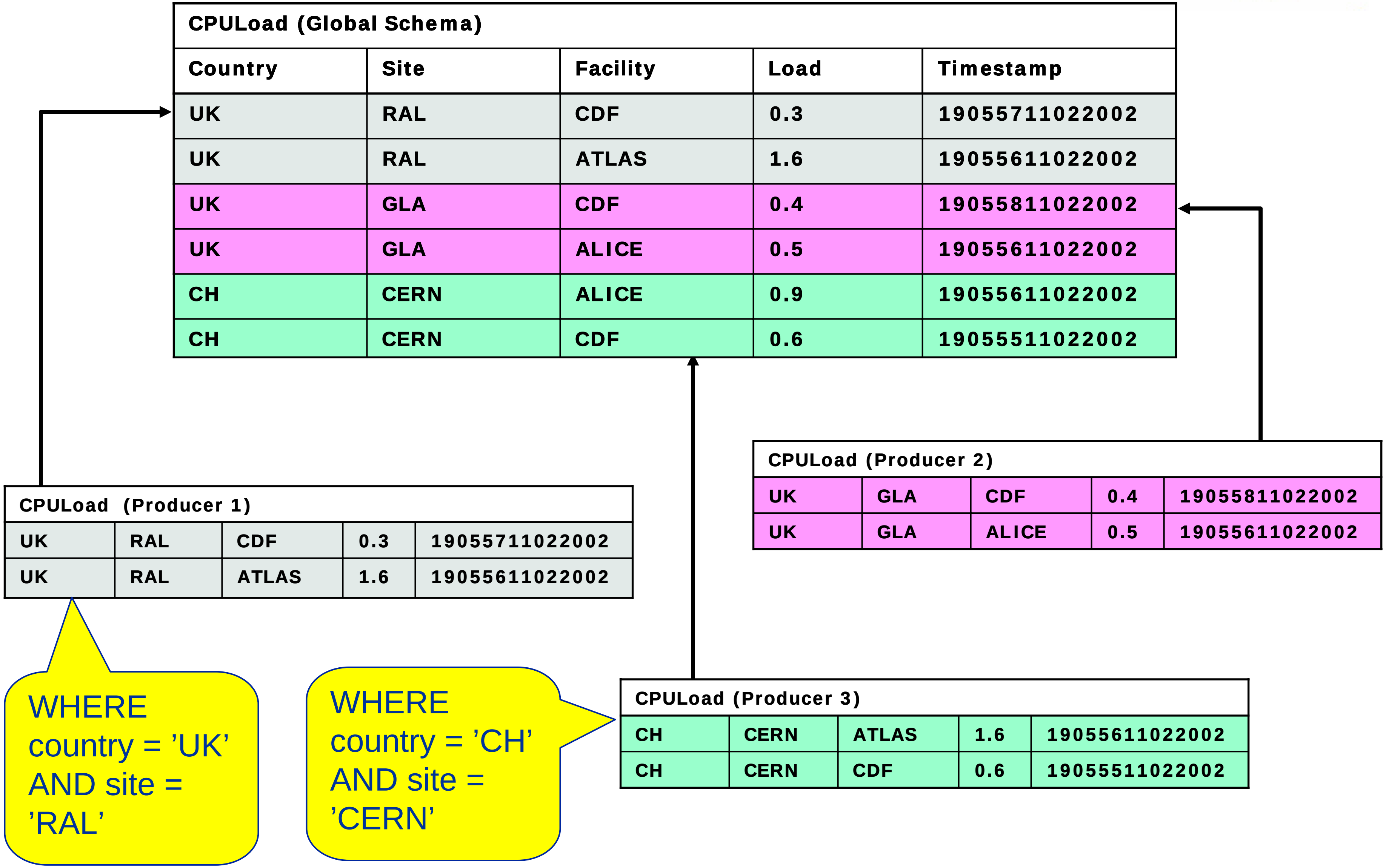
- StreamProducer – Supports Continuous Queries
  - In memory data structure
  - Can define minimum retention period
- ResilientStreamProducer – Supports Continuous Queries
  - Like the StreamProducer but won’t lose data if system crashes
  - So slightly slower
- DataBaseProducer – Supports History Queries
  - Information not lost
  - Supports joins
  - Clean up strategy
- LatestProducer – Supports Latest Queries
  - Just holds the latest information for any “primaryish” key
  - Supports joins
- CanonicalProducer – Supports anything
  - Offers “anything” as relations
  - User has to write code to handle SQL etc.

Registry and Schema



- Registry has two main tables:
  - Producer
    - Table name
    - Predicate
    - Location
  - Consumer
    - Query
    - Location
- Schema holds description of tables
  - Column names and types of each table
- Registry predicate defines subset of "global" table

Contributions to the "global" table



CPULoad (Global Schema)

Country	Site	Facility	Load	Timestamp
UK	RAL	CDF	0.3	19055711022002
UK	RAL	ATLAS	1.6	19055611022002
UK	GLA	CDF	0.4	19055811022002
UK	GLA	ALICE	0.5	19055611022002
CH	CERN	ALICE	0.9	19055611022002
CH	CERN	CDF	0.6	19055511022002

CPULoad (Producer 2)

UK	GLA	CDF	0.4	19055811022002
UK	GLA	ALICE	0.5	19055611022002

CPULoad (Producer 1)

UK	RAL	CDF	0.3	19055711022002
UK	RAL	ATLAS	1.6	19055611022002

CPULoad (Producer 3)

CH	CERN	ATLAS	1.6	19055611022002
CH	CERN	CDF	0.6	19055511022002

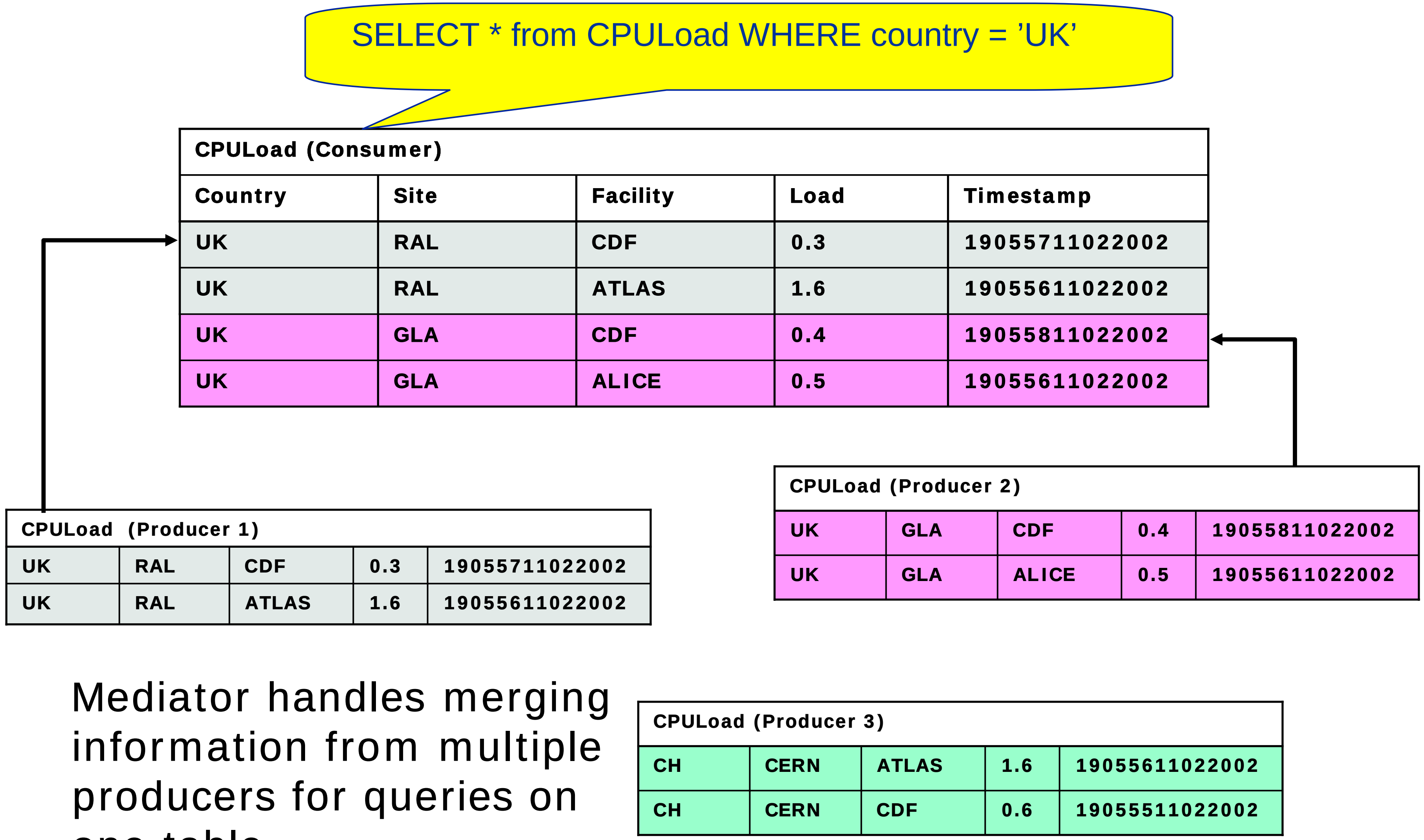
WHERE country = 'UK' AND site = 'RAL'

WHERE country = 'CH' AND site = 'CERN'

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

Queries over “global” table – merging streams



SELECT * from CPULoad WHERE country = ’UK’

CPULoad (Consumer)				
Country	Site	Facility	Load	Timestamp
UK	RAL	CDF	0.3	19055711022002
UK	RAL	ATLAS	1.6	19055611022002
UK	GLA	CDF	0.4	19055811022002
UK	GLA	ALICE	0.5	19055611022002

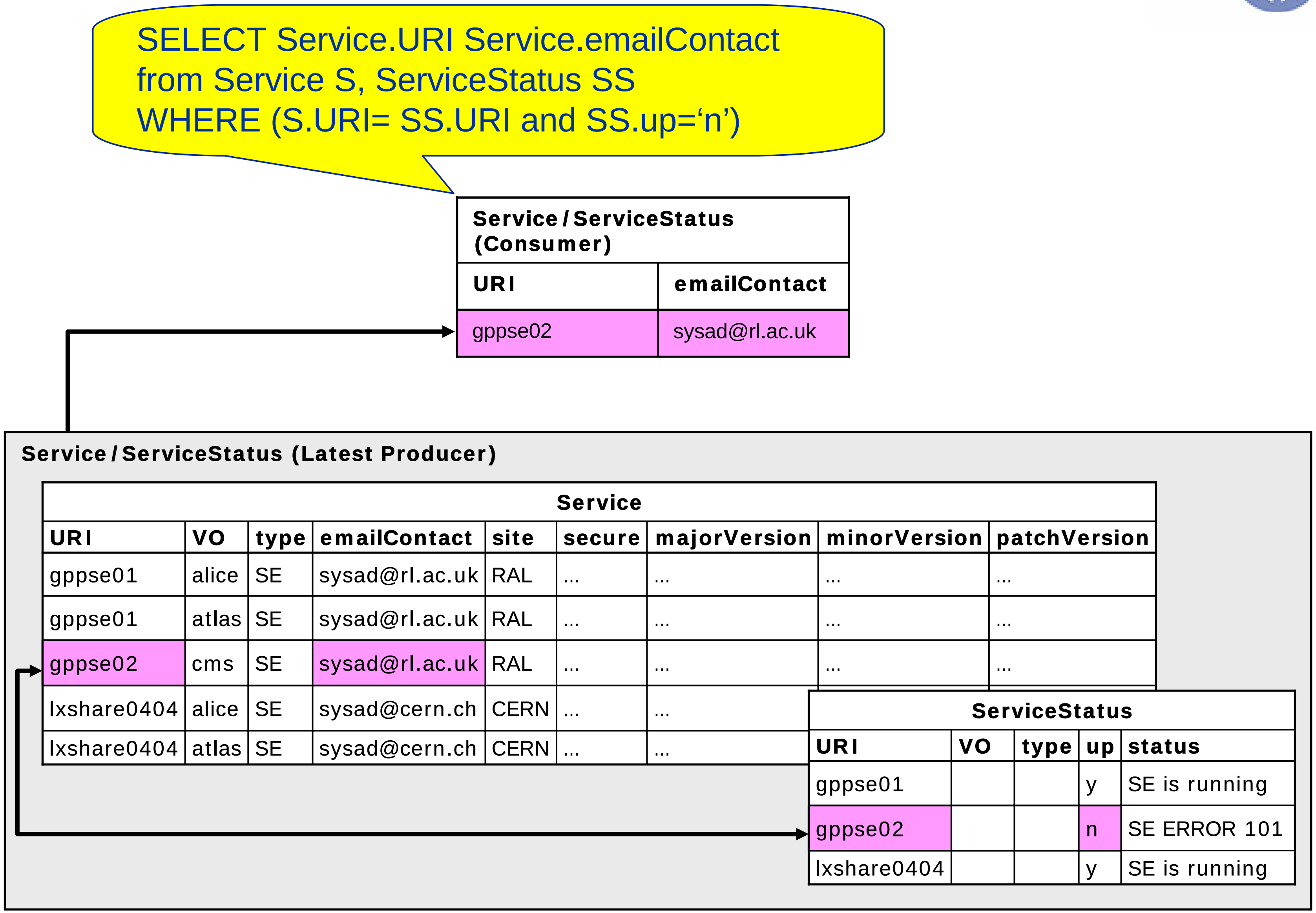
CPULoad (Producer 1)				
UK	RAL	CDF	0.3	19055711022002
UK	RAL	ATLAS	1.6	19055611022002

CPULoad (Producer 2)				
UK	GLA	CDF	0.4	19055811022002
UK	GLA	ALICE	0.5	19055611022002

Mediator handles merging information from multiple producers for queries on one table

CPULoad (Producer 3)				
CH	CERN	ATLAS	1.6	19055611022002
CH	CERN	CDF	0.6	19055511022002

Queries over “global” table – joining tables



SELECT Service.URI Service.emailContact
from Service S, ServiceStatus SS
WHERE (S.URI= SS.URI and SS.up=‘n’)

Service/ServiceStatus (Consumer)

URI	emailContact
gppse02	sysad@rl.ac.uk

Service/ServiceStatus (Latest Producer)

Service

URI	VO	type	emailContact	site	secure	majorVersion	minorVersion	patchVersion
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	...	...		

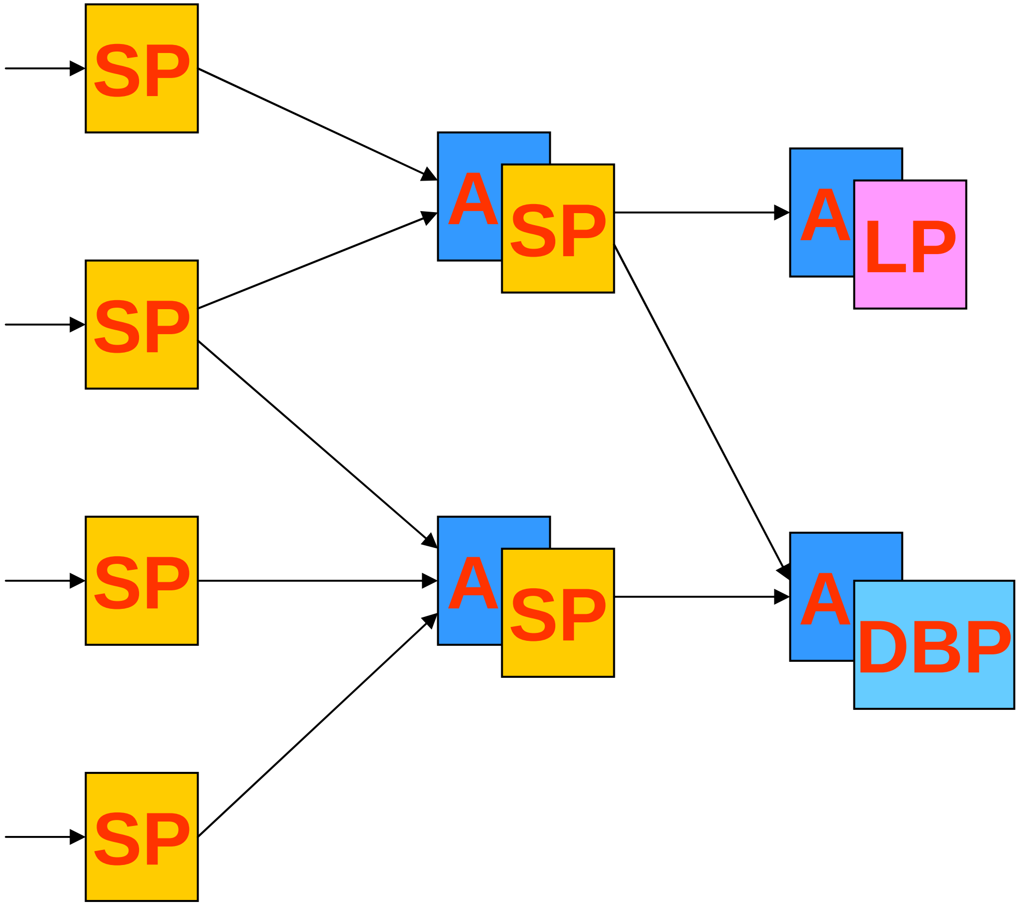
ServiceStatus

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

Archiver (Re-publisher)

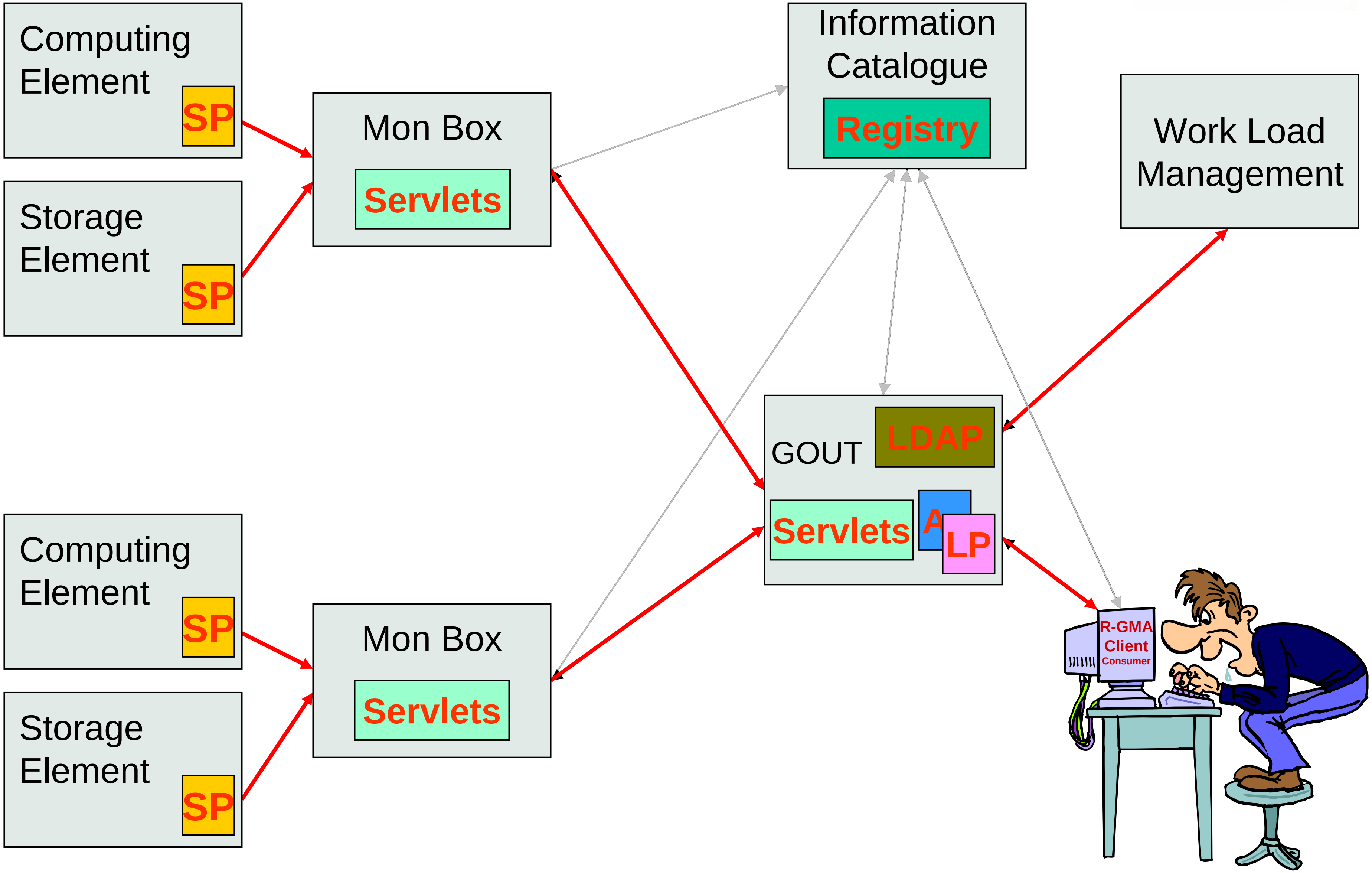
- It is a combined Consumer-Producer
  - Follows the GMA concept but packaged for ease of use
- You just have to tell it what to collect and it does so on your behalf
- Re-publishes to any kind of “Insertable” (i.e. not to the CanonicalProducer)
  - Can support joins if archiving to a DataBaseProducer or a LatestProducer

Topologies



- Normally publish via a StreamProducer SP
- Archivers A instantiated with a Producer and a Predicate.
  - May re-publish via:
    - StreamProducer
    - LatestProducer LP
    - DataBaseProducer DBP
- Must avoid cycles in the connections – i.e. must be a DAG.

Topology for EDG



Computing Element SP

Storage Element SP

Mon Box Servlets

Information Catalogue Registry

Work Load Management

GOUT LDAP Servlets A LP

Computing Element SP

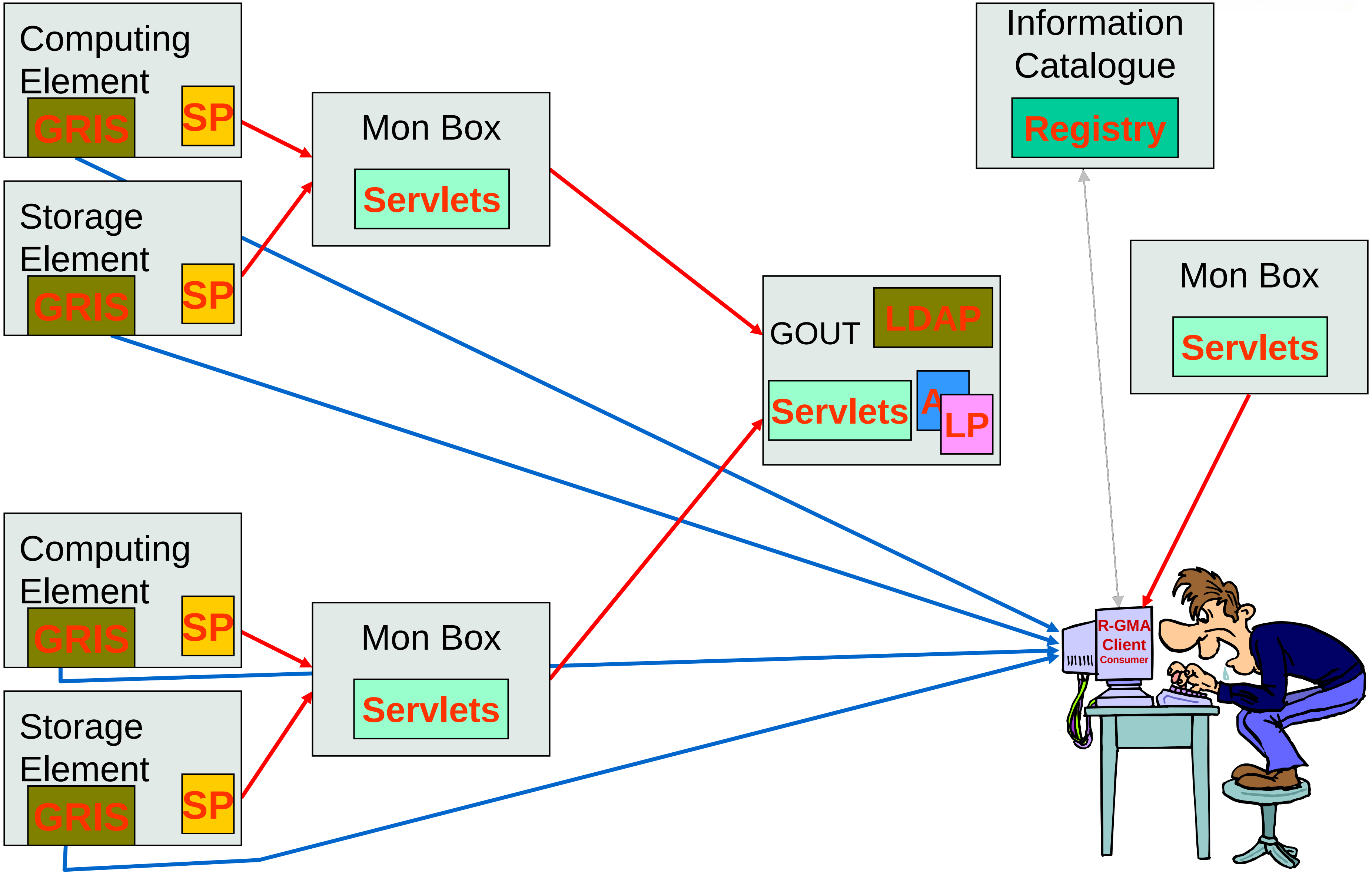
Storage Element SP

Mon Box Servlets

R-GMA Client Consumer

Monitoring the monitoring

GRISs



Computing Element

GRIS

SP

Storage Element

GRIS

SP

Mon Box

Servlets

Information Catalogue

Registry

GOUT

LDAP

Servlets

A

LP

Mon Box

Servlets

Computing Element

GRIS

SP

Storage Element

GRIS

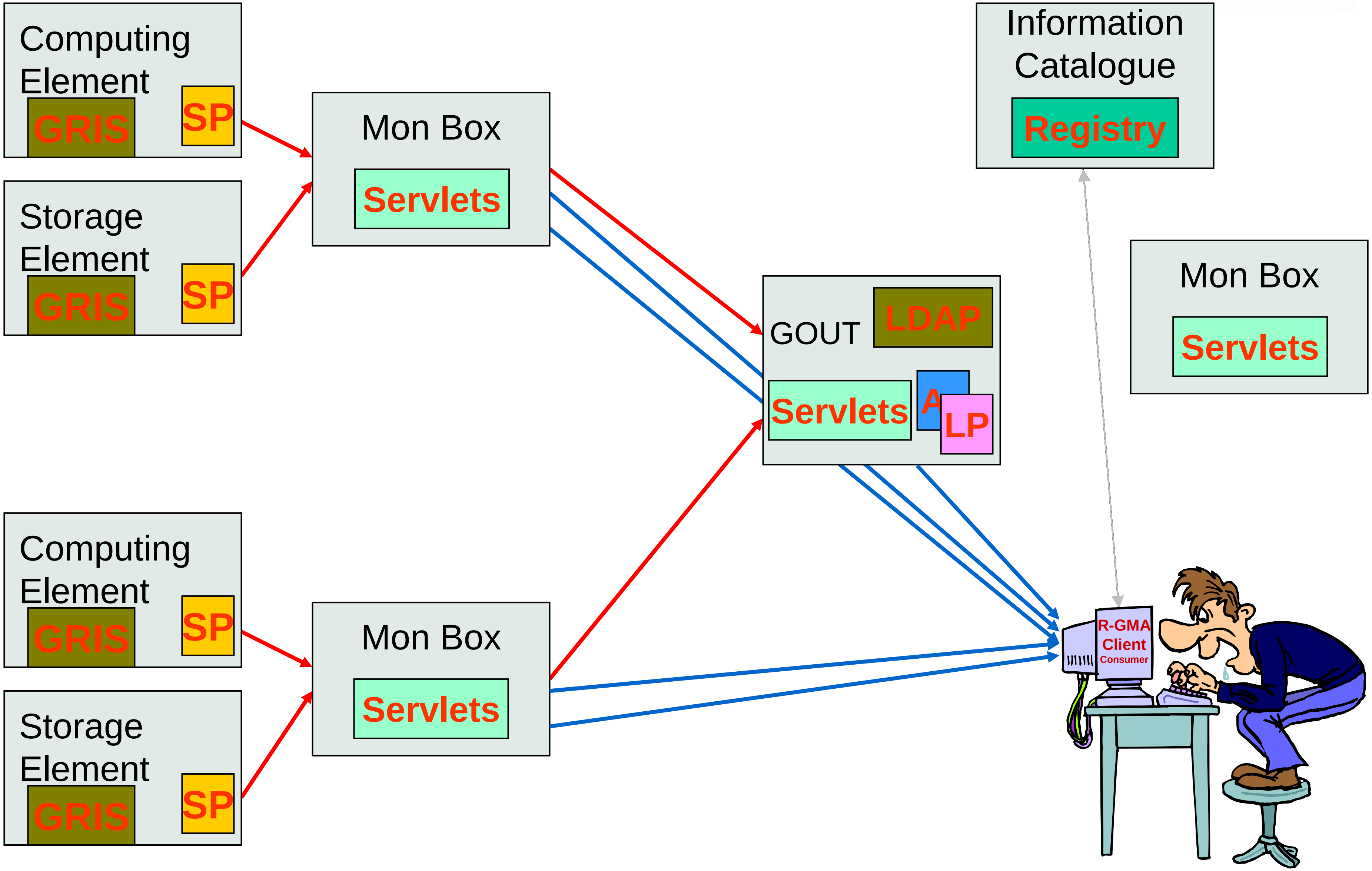
SP

Mon Box

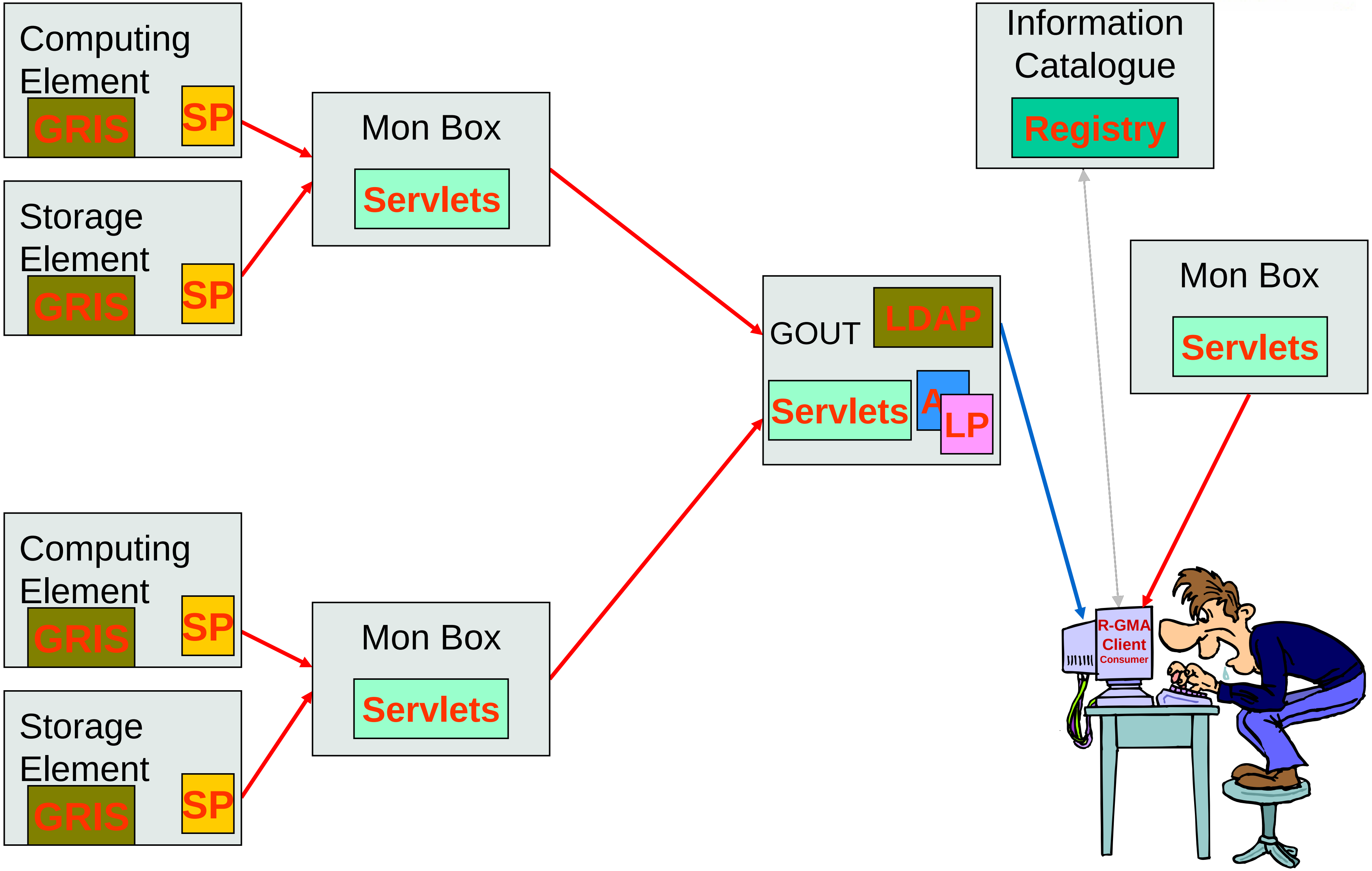
Servlets

R-GMA Client Consumer

R-GMA Producers



GOUT LDAP

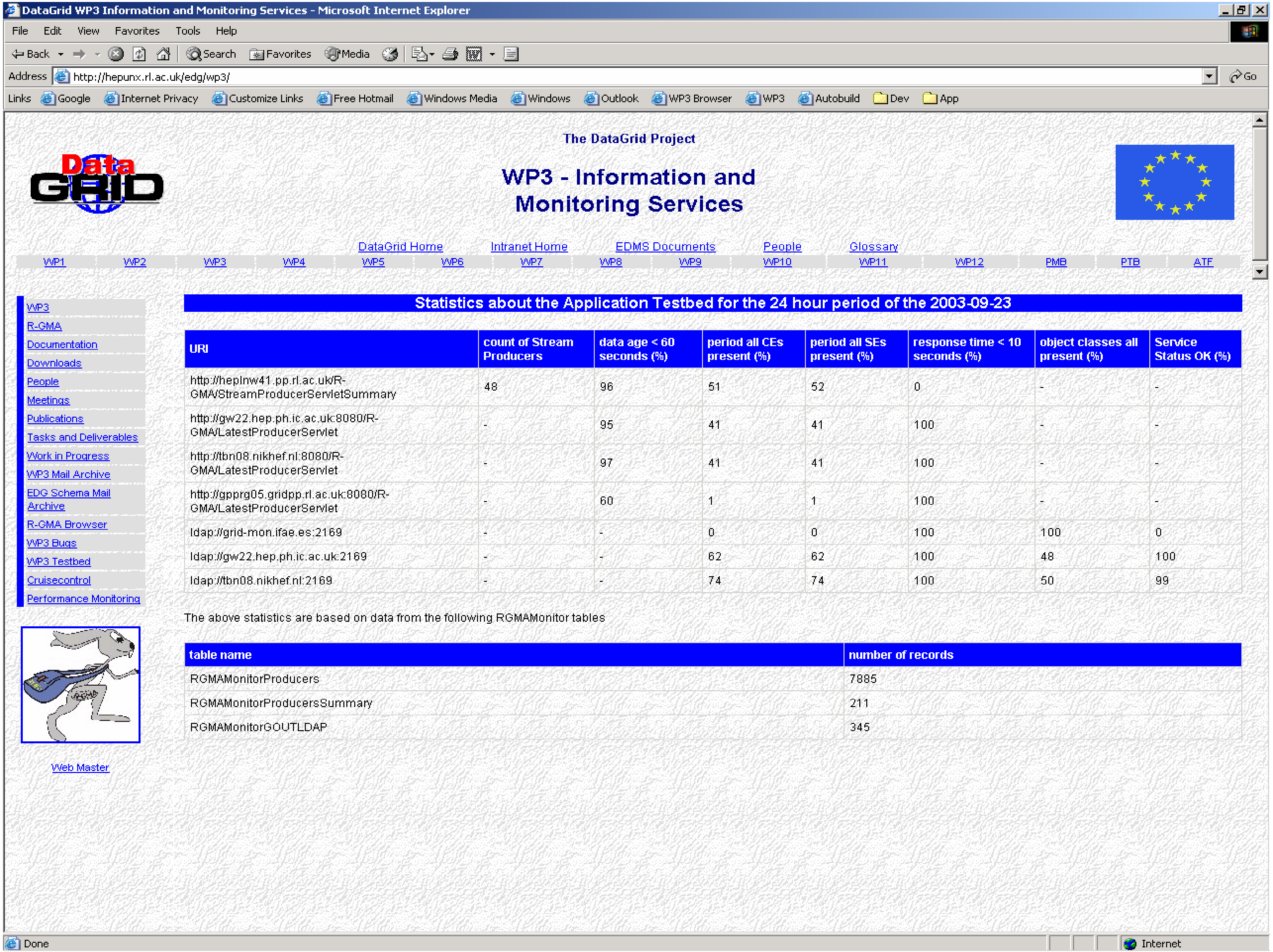

Computing Element
GRIS
SP
Storage Element
GRIS
SP
Mon Box
Servlets
Information Catalogue
Registry
GOUT
LDAP
Servlets
A
LP
Mon Box
Servlets
Computing Element
GRIS
SP
Storage Element
GRIS
SP
Mon Box
Servlets
R-GMA
Client
Consumer

Monitoring R-GMA: schema

- RGMAMonitorGRIS
  - URI, CECount, SECount, lineCount, responseTime, MeasurementDate, MeasurementTime
- RGMAMonitorProducers
  - serviceName, serviceType, producerType, producerUrl, responseTime, dataAge, MeasurementDate, MeasurementTime
- RGMAMonitorProducersSummary
  - URI, type, CECount, SECount, GRISCECount, GRISSECount, MeasurementDate, MeasurementTime
- RGMAMonitorGOUTLDAP
  - URI, CECount, SECount, lineCount, responseTime, objectClassStatus, objectClassMessage, GRISCECount, GRISSECount, GRISLineCount, MeasurementDate, MeasurementTime

Lots of Data

- 4 RGMAMonitor tables
  - Each with Latest and History Producers
- The monitoring scripts have a dependency on the Service table
- There are Latest and History Producers for the Service and ServiceStatus tables on our monitoring boxes
- We have all the data we need stored in a database on the monitoring machines
- So what do we do with all of this data?


DataGrid WP3 Information and Monitoring Services - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://hepunx.rl.ac.uk/edg/wp3/
Go
Links Google Internet Privacy Customize Links Free Hotmail Windows Media Windows Outlook WP3 Browser WP3 Autobuild Dev App
Data GRID
DataGrid Home Intranet Home EDMS Documents People Glossary
WP1 WP2 WP3 WP4 WP5 WP6 WP7 WP8 WP9 WP10 WP11 WP12 PMB PTB ATF
WP3
R-GMA
Documentation
Downloads
People
Meetings
Publications
Tasks and Deliverables
Work in Progress
WP3 Mail Archive
EDG Schema Mail Archive
R-GMA Browser
WP3 Bugs
WP3 Testbed
Cruisecontrol
Performance Monitoring
Web Master
Done
Internet

The DataGrid Project

WP3 - Information and Monitoring Services

Statistics about the Application Testbed for the 24 hour period of the 2003-09-23

URI	count of Stream Producers	data age < 60 seconds (%)	period all CEs present (%)	period all SEs present (%)	response time < 10 seconds (%)	object classes all present (%)	Service Status OK (%)
http://heplnw41.pp.rl.ac.uk/R-GMA/StreamProducerServletSummary	48	96	51	52	0	-	-
http://gw22.hep.ph.ic.ac.uk:8080/R-GMA/LatestProducerServlet	-	95	41	41	100	-	-
http://tbn08.nikhef.nl:8080/R-GMA/LatestProducerServlet	-	97	41	41	100	-	-
http://gpprg05.gridpp.rl.ac.uk:8080/R-GMA/LatestProducerServlet	-	60	1	1	100	-	-
ldap://grid-mon.ifae.es:2169	-	-	0	0	100	100	0
ldap://gw22.hep.ph.ic.ac.uk:2169	-	-	62	62	100	48	100
ldap://tbn08.nikhef.nl:2169	-	-	74	74	100	50	99

The above statistics are based on data from the following RGMAMonitor tables

table name	number of records
RGMAMonitorProducers	7885
RGMAMonitorProducersSummary	211
RGMAMonitorGOUTLDAP	345

Lots of Data

- Use Nagios to send alerts when
  - Any of the counts < the GRIS count
  - Queries timed out
  - etc.
- Improve visualisation of statistics
  - Provide graphical displays on our web site

Ranglia

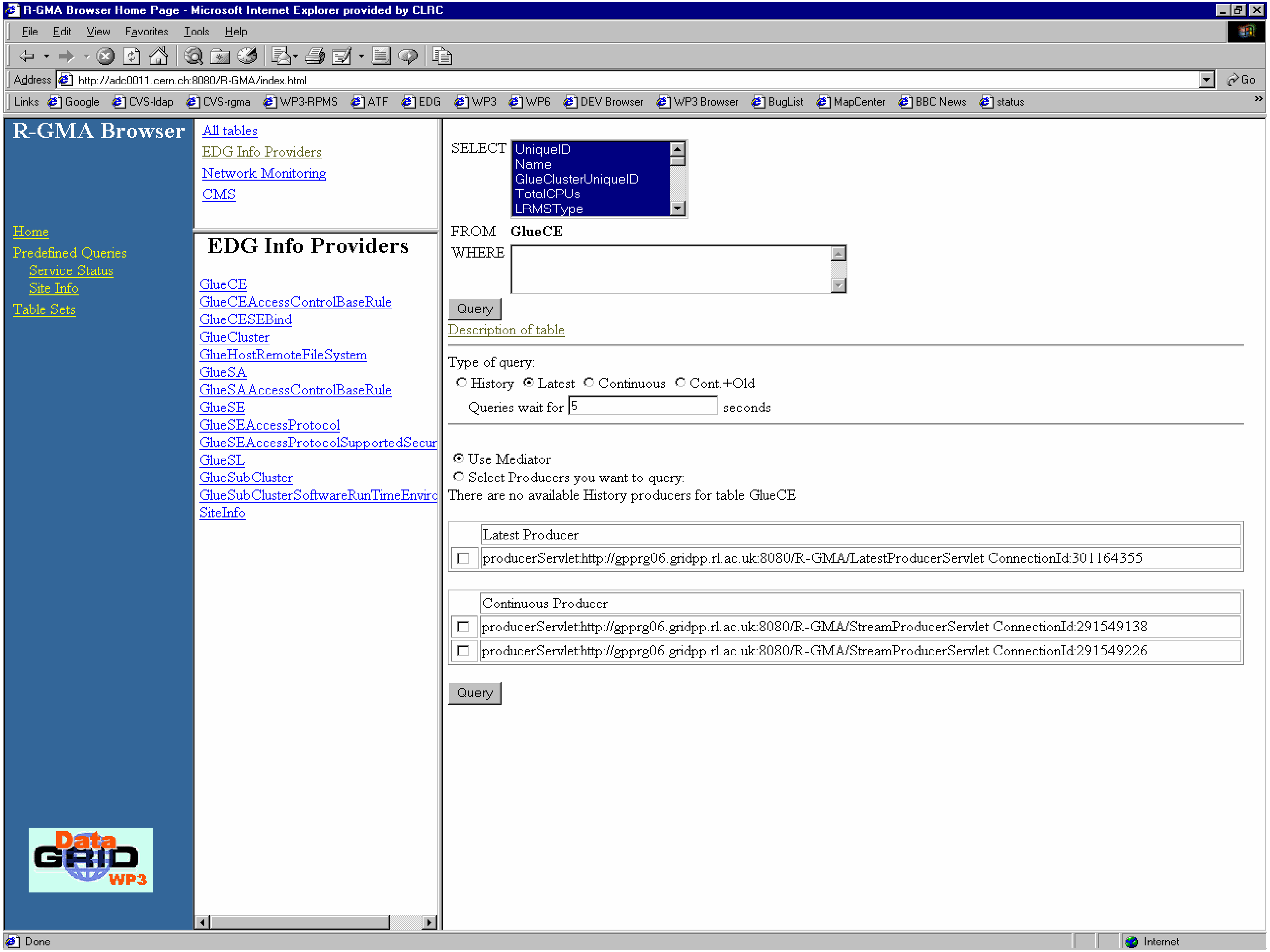
- R-GMA meets Ganglia
- A CanonicalProducer is used to interface Ganglia
- We have a working version
  - Still in early stages of development
  - Allows R-GMA queries to be made to Ganglia

Security

- Authentication has been implemented
  - Currently turned off on the development and application testbeds
- Denied hosts file
  - The Registry uses a file to check if a R-GMA server has been denied permission to register tables
  - A banned site can publish as much junk as it likes, as it will not have an entry in the Registry no one would see it

R-GMA Tools

- R-GMA Browser
  - Application dynamically generating web pages
  - Supports pre-defined and user-defined queries
- R-GMA CLI (edg-rgma)
  - Command Line Interface (similar to MySQL)
  - Supports single query and interactive modes
  - Can perform simple operations with Consumers, Producers and Archivers


R-GMA Browser Home Page - Microsoft Internet Explorer provided by CLRC
File Edit View Favorites Tools Help
Address http://adc0011.cern.ch:8080/R-GMA/index.html
Go
Links Google CVS-ldap CVS-rgma WP3-RPMS ATF EDG WP3 WP6 DEV Browser WP3 Browser BugList MapCenter BBC News status
Done
Internet
Data GRID WP3

R-GMA Browser

Home

Predefined Queries

- Service Status
- Site Info

Table Sets

All tables

EDG Info Providers

Network Monitoring

CMS

EDG Info Providers

GlueCE
GlueCEAccessControlBaseRule
GlueCESEBind
GlueCluster
GlueHostRemoteFileSystem
GlueSA
GlueSAAccessControlBaseRule
GlueSE
GlueSEAccessProtocol
GlueSEAccessProtocolSupportedSecur
GlueSL
GlueSubCluster
GlueSubClusterSoftwareRunTimeEnviro
SiteInfo

SELECT UniqueID
Name
GlueClusterUniqueID
TotalCPUs
LRMSType

FROM **GlueCE**

WHERE

Query

Description of table

Type of query:
○ History ● Latest ○ Continuous ○ Cont.+Old

Queries wait for 5 seconds

● Use Mediator
○ Select Producers you want to query:
There are no available History producers for table GlueCE

	Latest Producer
□	producerServlet:http://gpprg06.gridpp.rl.ac.uk:8080/R-GMA/LatestProducerServlet ConnectionId:301164355

	Continuous Producer
□	producerServlet:http://gpprg06.gridpp.rl.ac.uk:8080/R-GMA/StreamProducerServlet ConnectionId:291549138
□	producerServlet:http://gpprg06.gridpp.rl.ac.uk:8080/R-GMA/StreamProducerServlet ConnectionId:291549226

Query

edg-rgma

- show tables
- describe Service
- show producers of Service
- latest select * from Service
- old continuous select * from Service

edg-rgma – Example

```
edg-rgma \
   -c "timeout 0.1" \
   -c timeout \
   -c "str decl Service" \
   -c "str minr .2" \
   -c "str minr" \
   -c "stream INSERT into Service (URI, VO, type, secure,
    emailContact, site, majorVersion, minorVersion,
    patchVersion) values ('a','b','c','y', 'd','e',1,2,3)" \
   -c "old continuous select * from Service
```

```
+-----+----+------+---------------+------+--------+---------------+---------------+---------------+----------------
-+------------------+
| URI | VO | type | emailContact | site | secure | majorVersion | minorVersion | patchVersion | MeasurementDate
| MeasurementTime |
+-----+----+------+---------------+------+--------+---------------+---------------+---------------+----------------
-+------------------+
| a   | b  | c    | d             | e    | y      | 1             | 2             | 3             | 2003-07-08
| 10:26:58          |
+-----+----+------+---------------+------+--------+---------------+---------------+---------------+----------------
-+------------------+
1 Rows in set
```

APIs

- Exist in Java, C++, C, Python and Perl
- C, Python and Perl follow an object based style reflecting the Java and C++ APIs

Java

```
myProducer = new StreamProducer();
```

C++

```
myProducer= new edg::info::StreamProducer();
```

C

```
myProducer = StreamProducer_new();
```

Perl

```
$myProducer = rgmainfo::StreamProducer_new();
```

Python

```
myProducer = rgmainfo.StreamProducer_new()
```

C++ Consumer - Example

```
#include <string>
#include <iostream>
#include <unistd.h>
#include <stdio.h>

#include "info/Consumer.hh"
#include "info/ResultSet.hh"

int main(){
  try {
    edg::info::Consumer myConsumer("SELECT * FROM userTable", edg::info::Consumer::LATEST);
    edg::info::TimeInterval Timeout(60);
    myConsumer.start(Timeout);
    while(myConsumer.isExecuting()){
      sleep(2);
    }
    if(myConsumer.hasAborted()){
      std::printf("Consumer query timed-out\n");
    }
    edg::info::ResultSet resultSet = myConsumer.popIfPossible();
    std::printf("ResultSet: %s\n", resultSet.toString().c_str());
    myConsumer.close();
  } catch (edg::info::RGMAException& e) {
    std::printf("Exception: %s\n", e.what());
  }
}
```

C++ Producer - Example

```
#include <string>
#include <iostream>
#include "info/StreamProducer.hh"

int main(int argc, char* args[]) {

  if (argc != 2) {
    std::cout << "Exactly one argument must be specified\n" << std::endl;
    exit(1);
  }

  try {
    edg::info::StreamProducer myProducer;
    std::string astring = std::string("WHERE (userId = '") + std::string(args[1]) +
      std::string("')");
    std::cout << "Predicate: " << astring << std::endl;
    myProducer.declareTable("userTable", astring);
    myProducer.setTerminationInterval(edg::info::TimeInterval(1200));
    myProducer.setMinRetentionPeriod(edg::info::TimeInterval(600));
    astring = std::string("INSERT INTO userTable (userId, aString, aReal, anInt)
      VALUES ('") + std::string(args[1]) + std::string("', 'C++ producer', 3.1415962, 42)");
    std::cout << astring << std::endl;
    myProducer.insert(astring);
  } catch (edg::info::RGMAException& e) {
    std::cout << "Exception " << e.what() << std::endl;
  }
}
```

C++ Producer

- TerminationInterval
  - Period by which the producer must re-announce its existence
    - If it fails to do so it will be removed from the registry
    - Default is 20 minutes
    - Don't set it too long
- RetentionPeriod
  - Period which the published data will remain available, even after the Producer has been closed
  - Default is 0

Summary

- R-GMA
  - is suitable for Information **and** Monitoring
  - is a relational implementation of the GGF's GMA
  - has different Producer types
  - components can be deployed in various topologies
  - mediator creates the impression of a single RDBMS
  - has authentication using grid certificates
  - has been integrated with Ganglia and Nagios
  - has an API available in multiple languages

Further Information

- Information and Monitoring Services
  - http://hepunx.rl.ac.uk/edg/wp3/

- R-GMA
  - http://www.r-gma.org/