Introduction to R-GMA: Relational Grid Monitoring Architecture

Acknowledgements

Slides are taken/derived from

- **the GILDA team**
- **Steve Fisher (RAL, UK) and the R-GMA team**

What is R-GMA ?

- **Uniform method to access and publish both information and monitoring data.**
- **From a user's perspective, an R-GMA installation currently appears similar to a single relational database.**
- **GMA (Grid Monitoring Architecture) was developed by the GGF**
- **R-GMA (Relational GMA) was created:**
  - To simplify use of GMA (servers “know” about registries, not the client software)
  - To give a relational view

Introduction to R-GMA

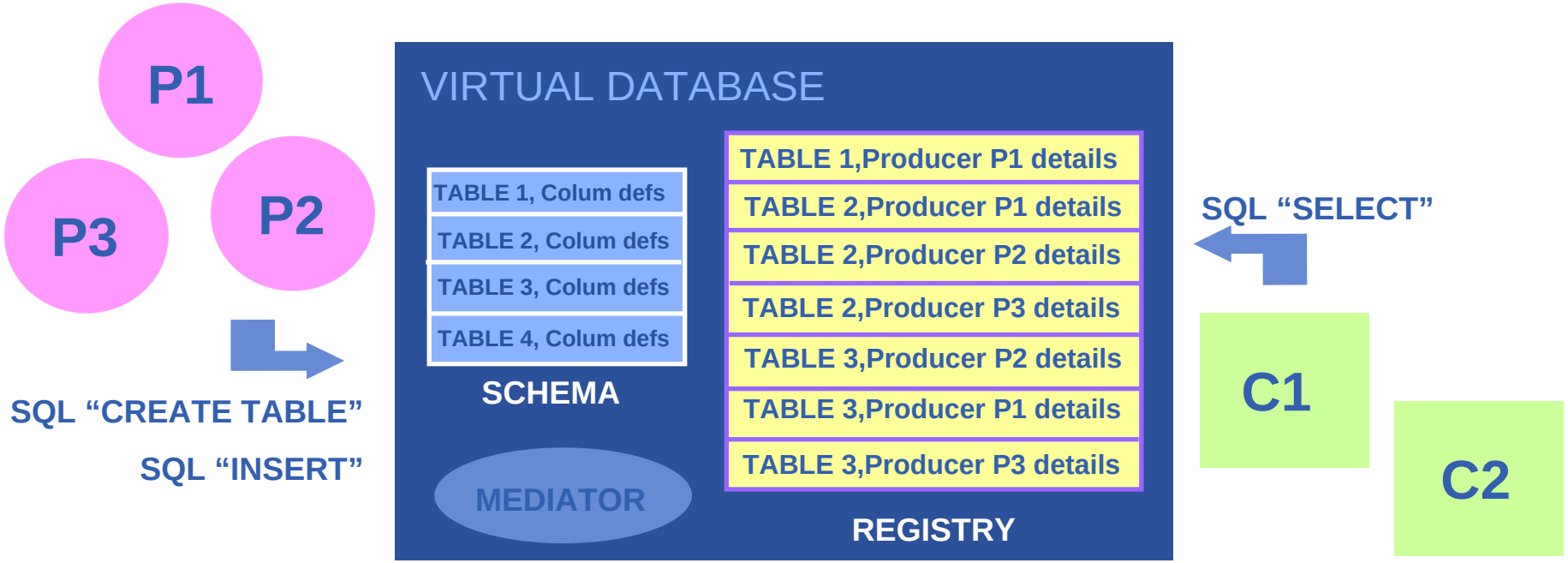
- **Relational Grid Monitoring Architecture (R-GMA)**
  - Developed as part of the EuropeanDataGrid Project (EDG)
  - Now as part of the EGEE project.
  - Evolution from the Grid Monitoring Architecture (GMA)

- **Uses a relational data model.**
  - Data are viewed as a table.
  - Data structure defined by the columns.
  - Each entry is a row (tuple).
  - Queried using Structured Query Language (SQL).

name	ID	birth	Group
Tom	4	1977-08-20	HR

SELECT * FROM people WHERE group='HR'

R-GMA



There is no central repository!!! There is only a "*Virtual Database*".

Schema is a list of table definitions: additional tables/schema can be defined by applications

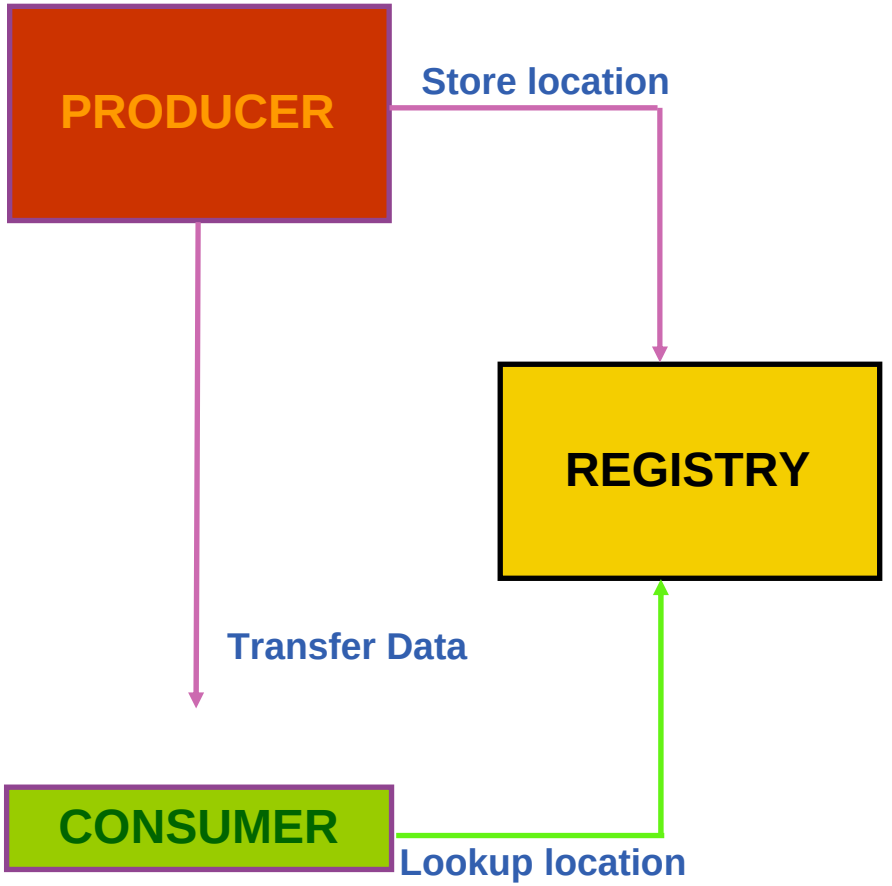
Registry is a list of data producers with all its details.

Producers publish data.

Consumers read data published.

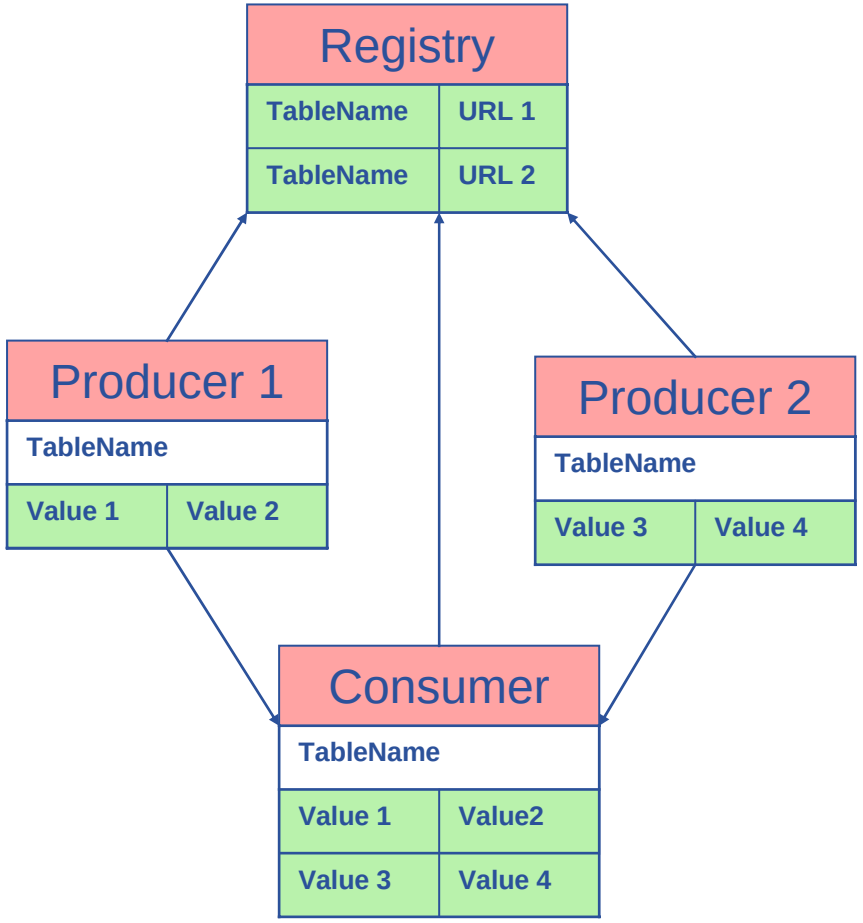
Service orientation

- 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.



Virtual database

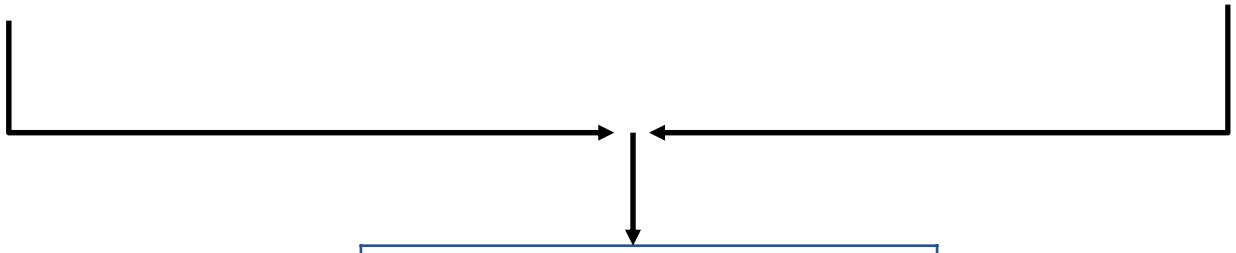
- The Consumer interrogates the Registry to identify all Producers that could satisfy the query.
- Consumer connects to the Producers.
- Producers send the tuples to the Consumer.
- The Consumer will merge these tuples to form one result set.



Joins

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

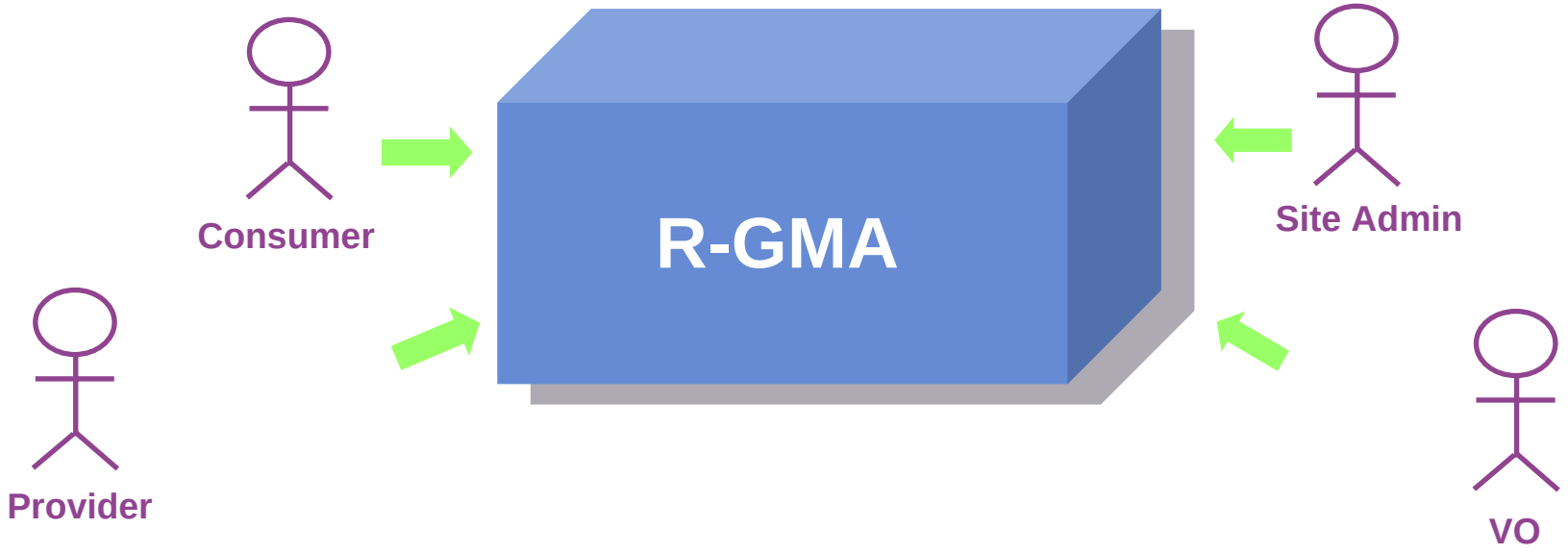
ServiceStatus				
URI	VO	type	up	status
gppse01	alice	SE	y	SE is running
gppse01	atlas	SE	y	SE is running
gppse02	cms	SE	n	SE ERROR 101
lxshare0404	alice	SE	y	SE is running
lxshare0404	atlas	SE	y	SE is running



Result Set (Consumer)	
URI	emailContact
gppse02	sysad@rl.ac.uk

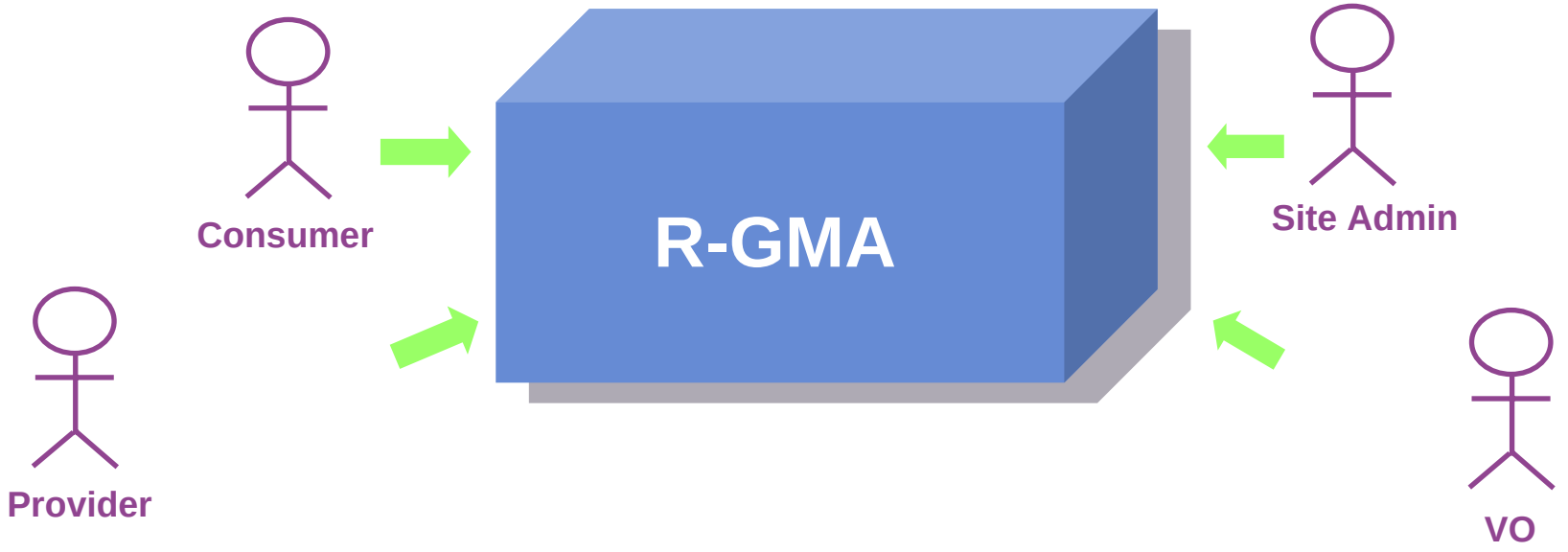
SELECT Service.URI Service.emailContact FROM Service S, ServiceStatus SS
WHERE (S.URI= SS.URI and SS.up='n')

Roles



- **Consumer users: who request information.**
- **Producer users: who provide information.**
- **Site administrators: who run R-GMA services.**
- **Virtual Organizations: who “own” the schema and registry.**

Security



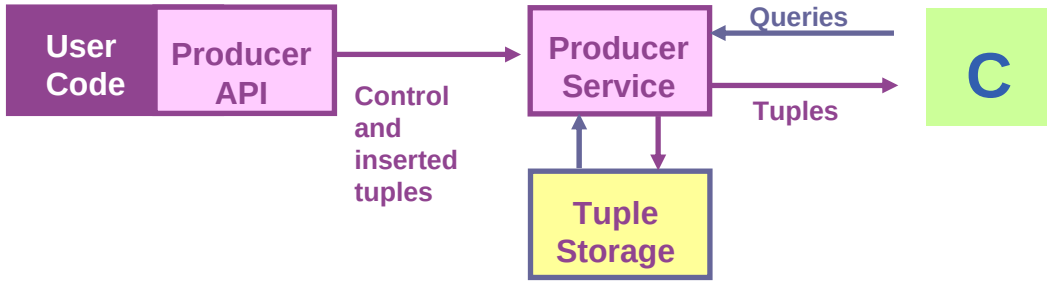
- **Mutual Authentication: guaranteeing who is at each end of an exchange of messages.**
- **Encryption: using an encrypted transport protocol (HTTPS).**
- **Authorization: implicit or explicit.**

Deployment

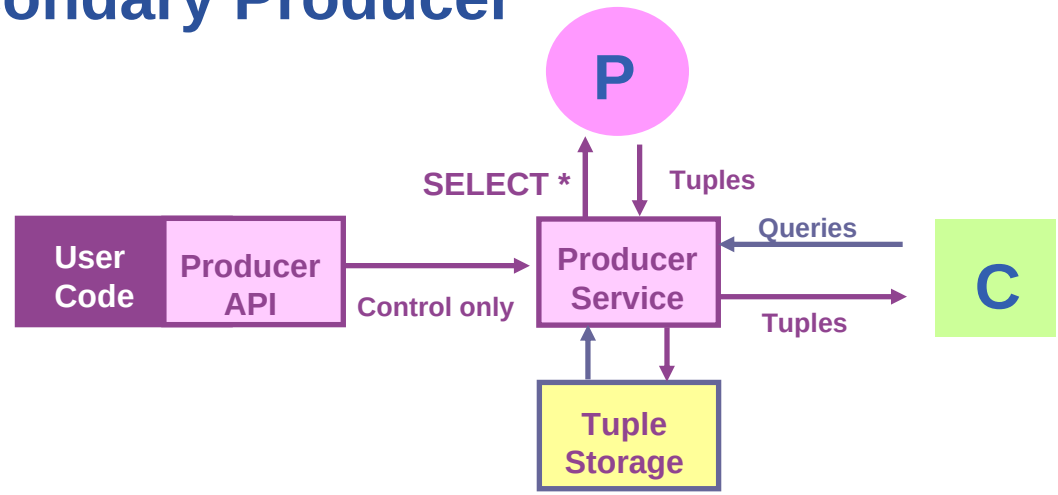
- **Producer and Consumer Services are typically on a one per site basis**
- **Centralized Registry and Schema.**
- **The Registry and Schema may be replicated, to avoid a single point of failure**
  - … when you use RGMA CLI you will see which are being used

Producer Types

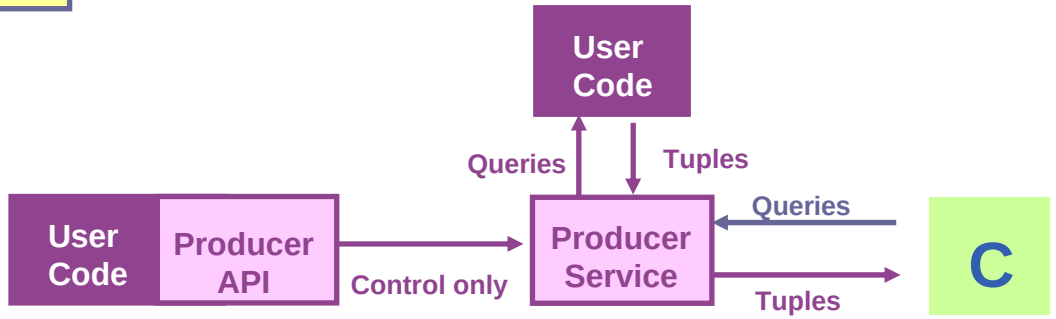
- **Primary Producer**



- **Secondary Producer**



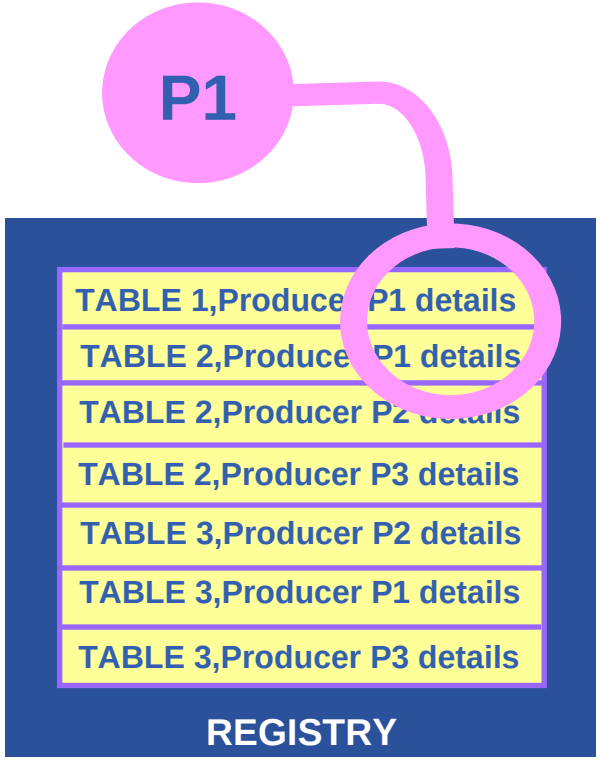
- **On-Demand Producer**
  - No internal storage
  - Queries passed to user code



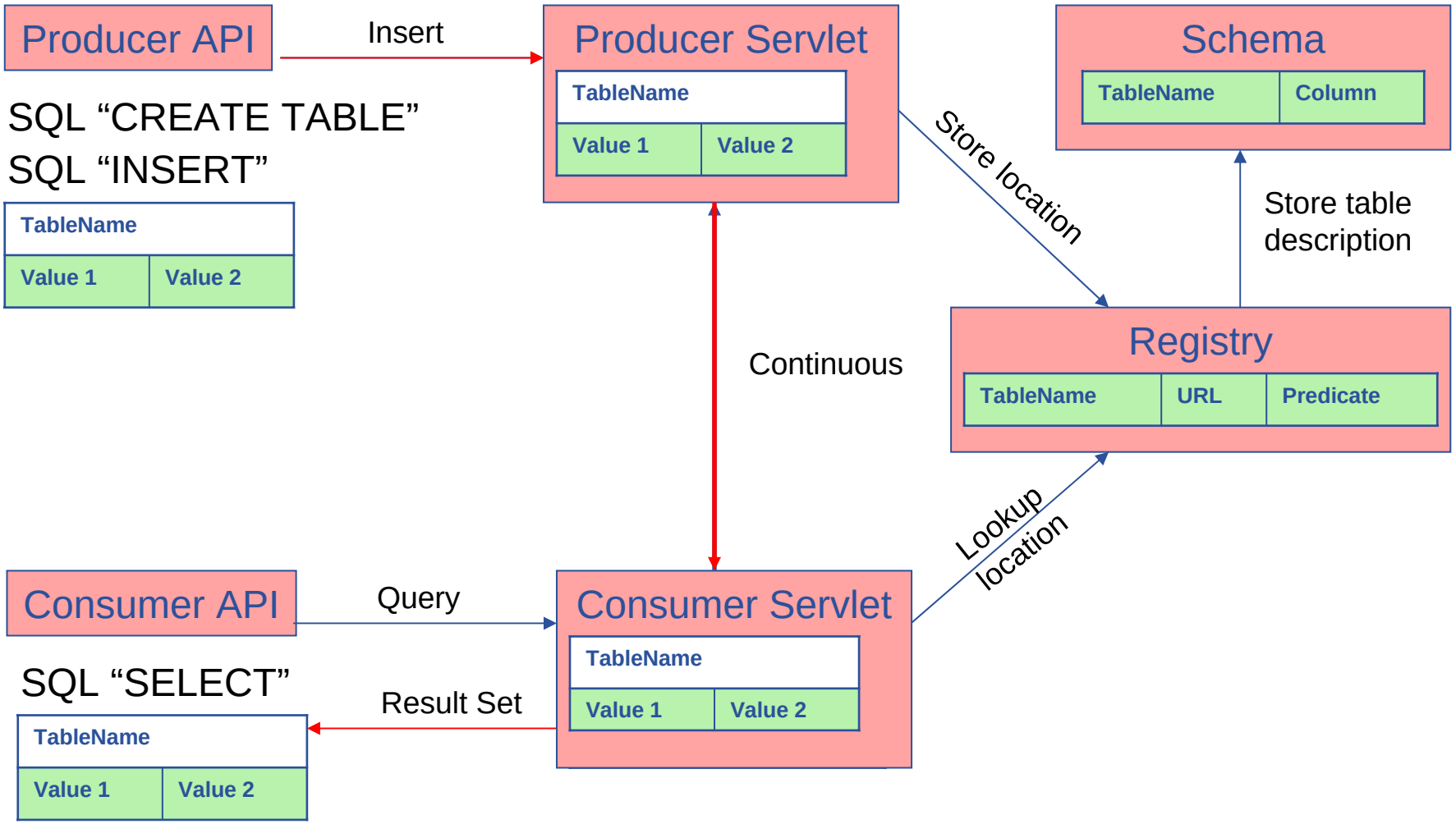
Query Types

Continuous

- **Latest**
- **History**
- **Static**

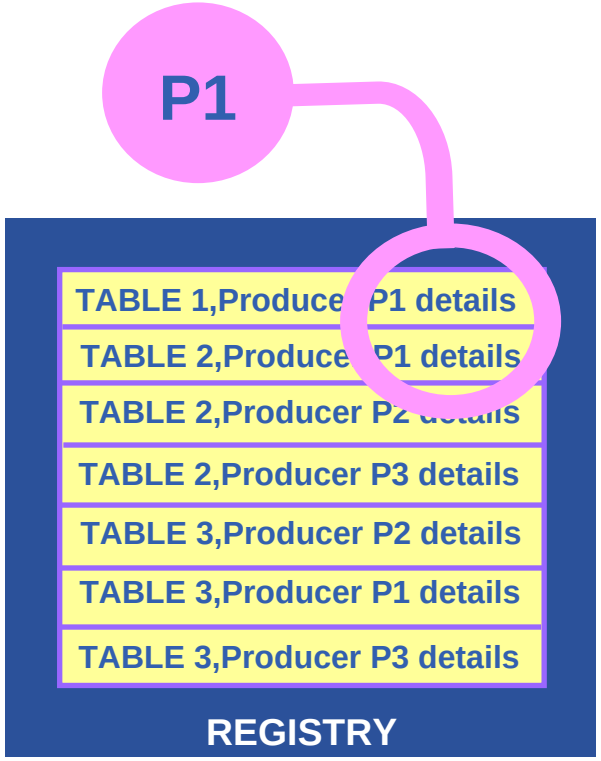


Continuous

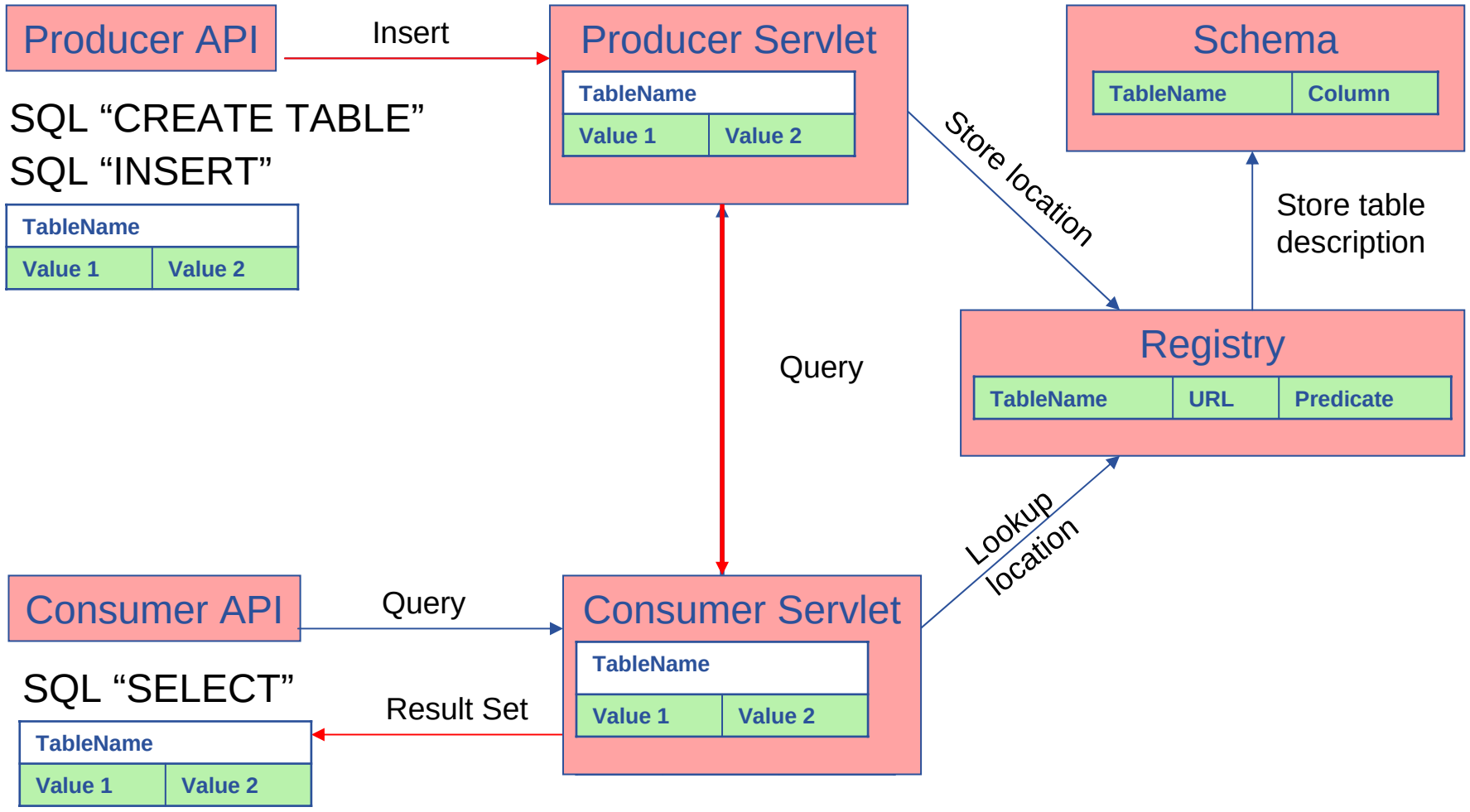

Producer API
Insert
Producer Servlet
TableName
Value 1
Value 2
Schema
TableName
Column
SQL “CREATE TABLE”
SQL “INSERT”
TableName
Value 1
Value 2
Store location
Store table description
Continuous
Registry
TableName
URL
Predicate
Lookup location
Consumer API
Query
Consumer Servlet
TableName
Value 1
Value 2
SQL “SELECT”
Result Set
TableName
Value 1
Value 2

Query Types

- Continuous
- Latest
- History
- Static

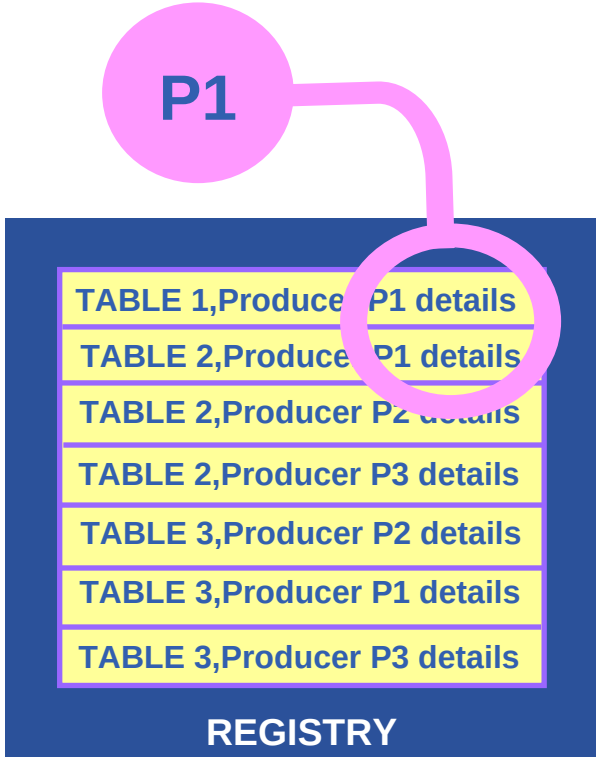


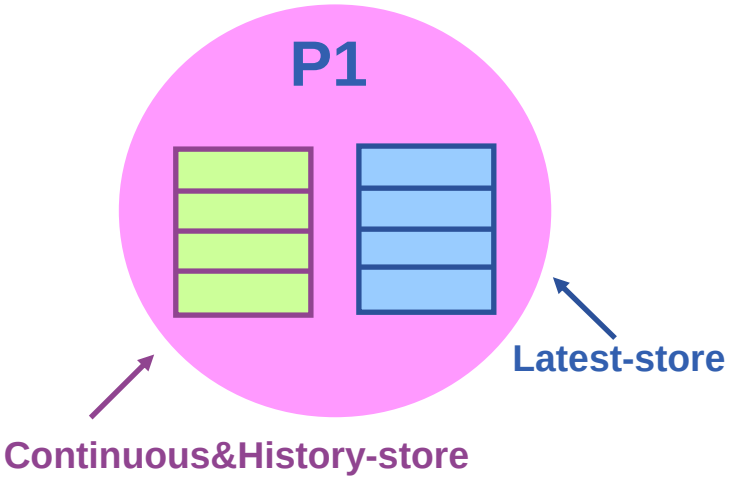
History or Latest



Query Types

- Continuous
- Latest
- History
- Static



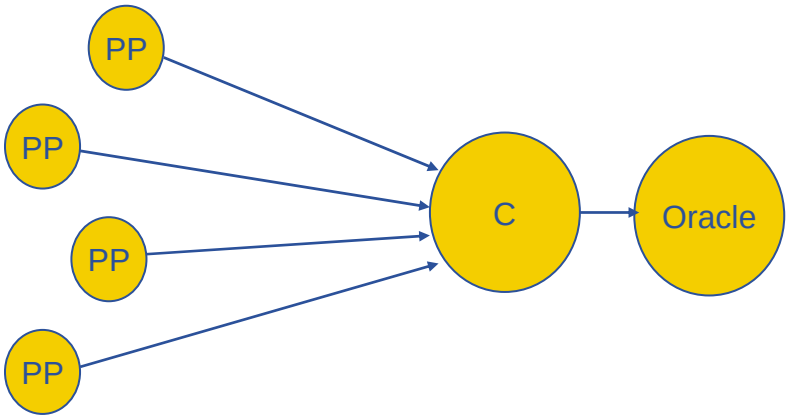


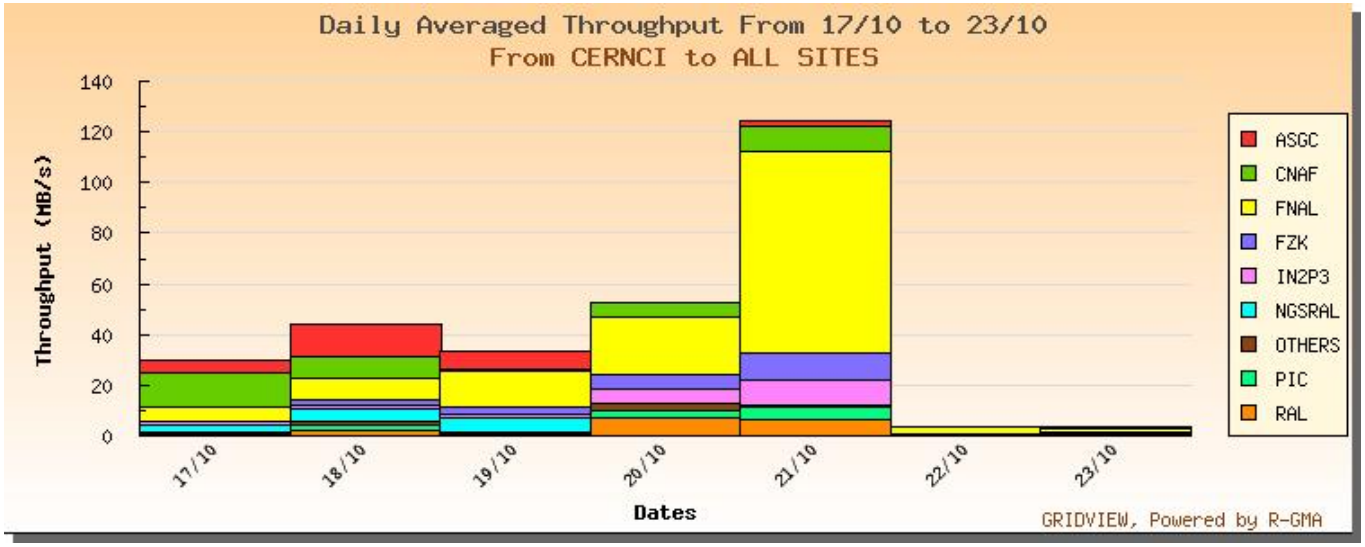
Latest Retention Period

History Retention Period

GridFTP Monitoring (gridView)

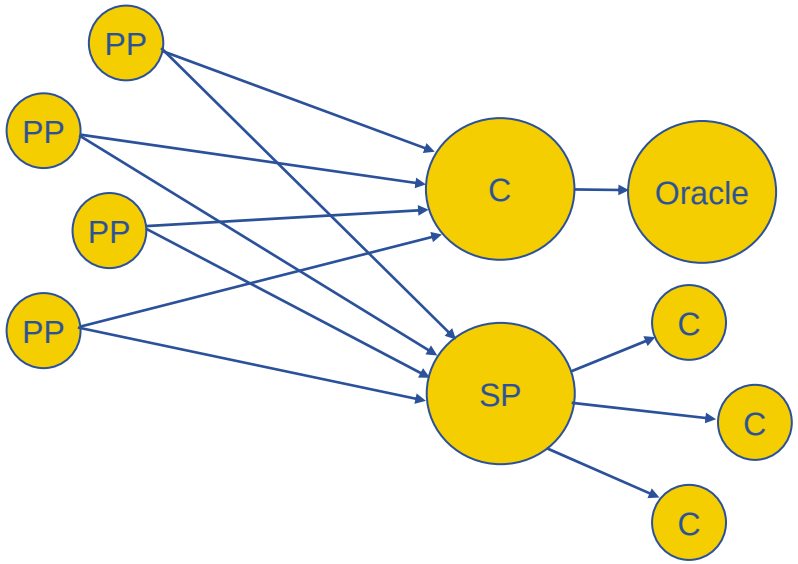
- SA1 have written script to “tail” FTP logs and publish via PP on gridFTP server nodes
- Continuous query pulls all the data to a central location and writes to an Oracle database for analysis
- Used for Service Challenge 3
- http://gridview.cern.ch/GRIDVIEW/





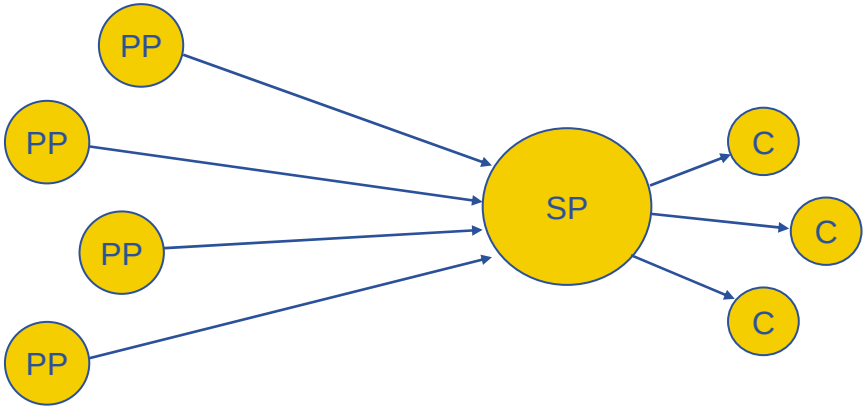
Job Monitoring (L&B)

- Reads L&B logs on the resource broker nodes.
- Publishes data on state of jobs
- A database secondary producer is used to aggregate the data as well as a gridView consumer.
- CMS dashboard
  - http://lxarda09.cern.ch/dashboard/request.py/jobsummary?



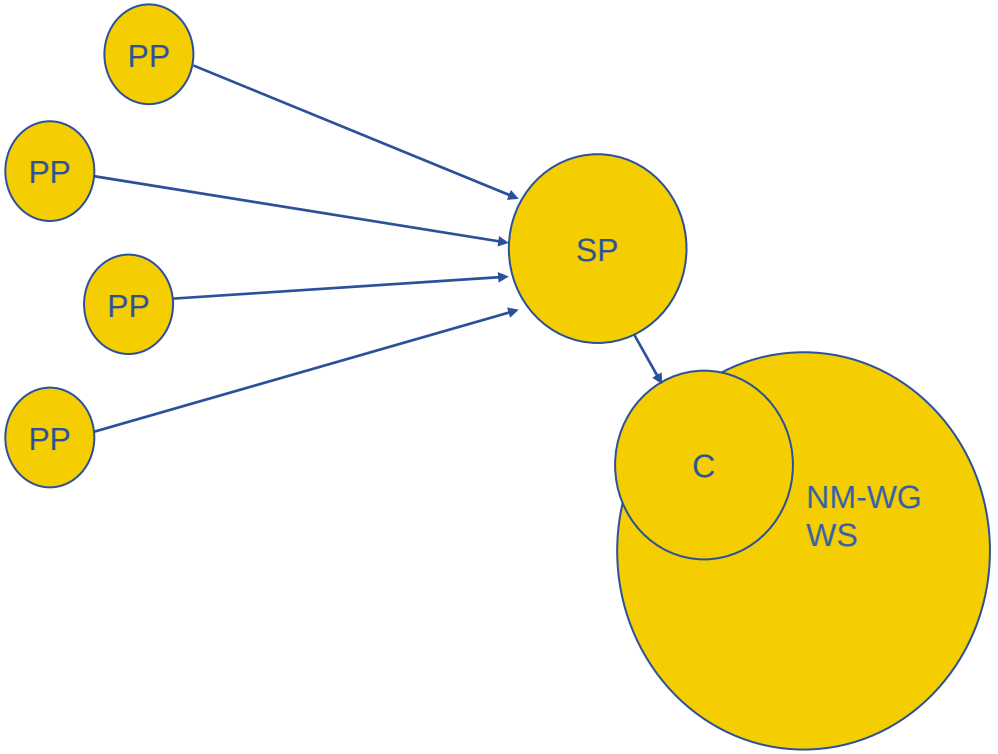
Job Monitoring (WN)

- **On the WNs, the Job Wrapper (if enabled by JDL) periodically publishes information about the state of the process running the job and its environment.**
- **A database secondary producer is used to aggregate the data.**
  - https://rgma13.pp.rl.ac.uk:8443/R-GMABrowser/Browser.do/queryTable?selectQueryType=latest&duration=20&tableName=JobMonitor



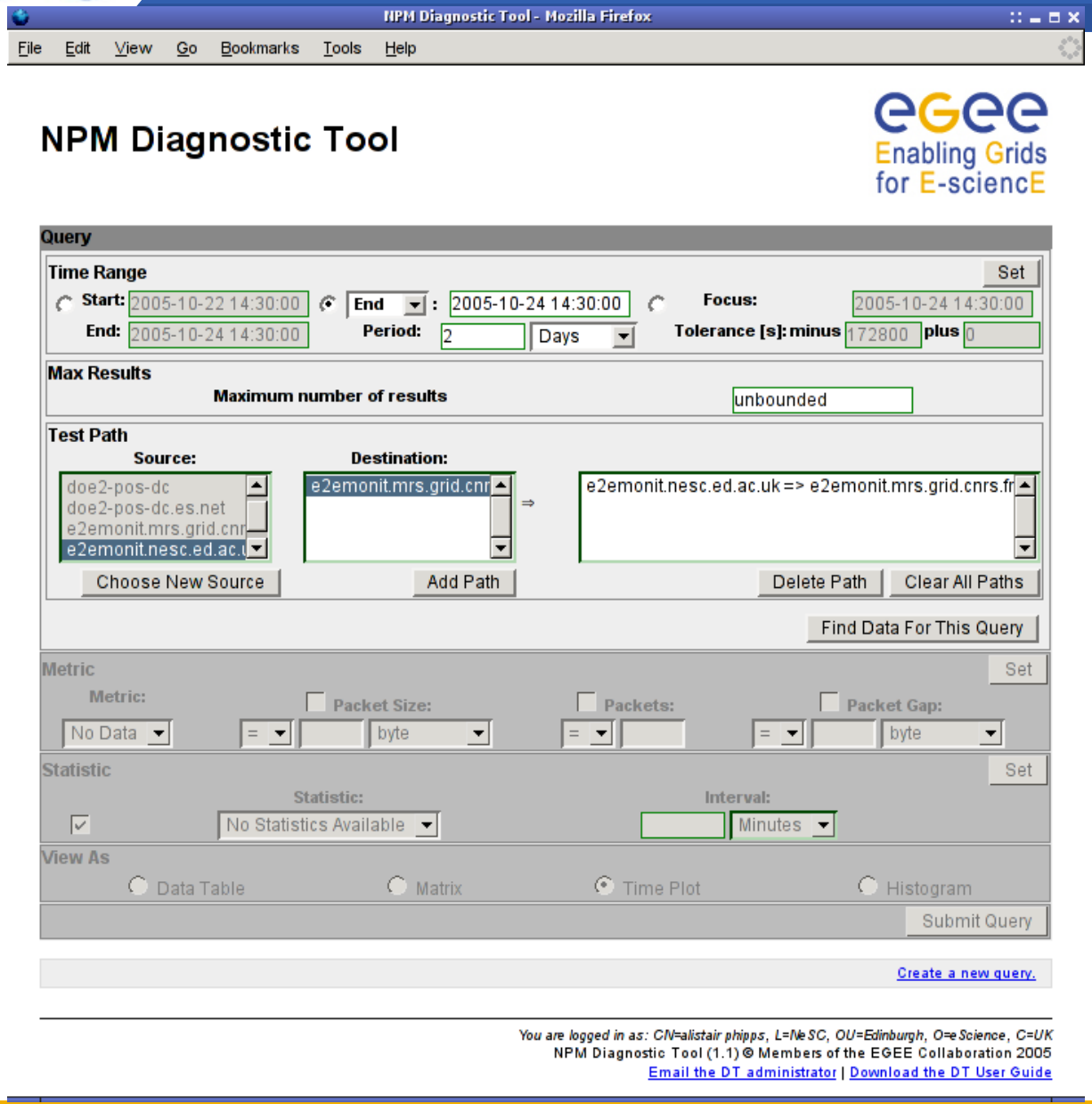
NPM Frameworks: e2emonit

- **Network performance data important:**
  - to detect and resolve network problems.
  - to intelligently schedule jobs based on network load and reliability.
- **active measurements between end-sites, using tools such as**
  - iperf,
  - udpmon
  - ping.

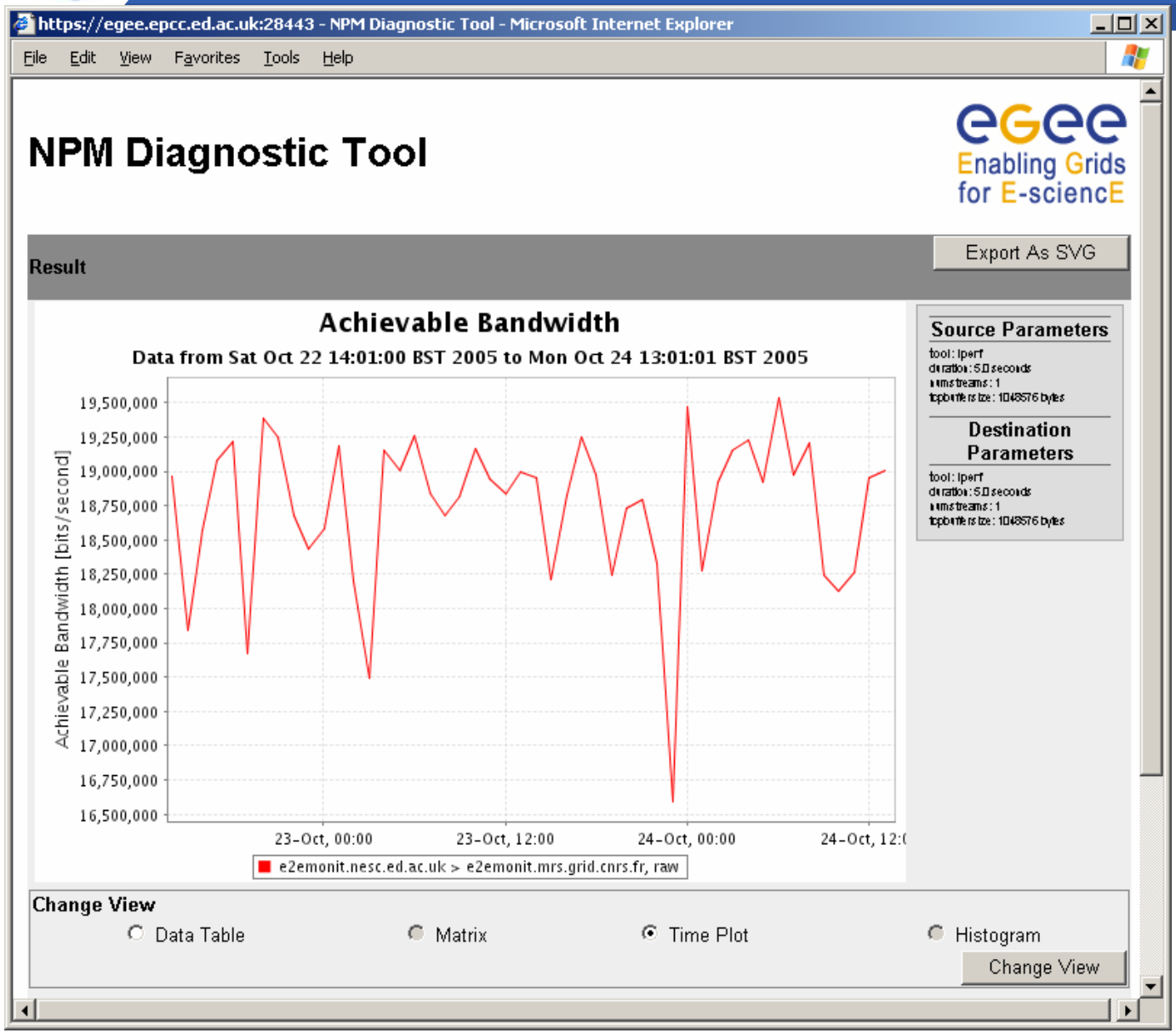


- https://egee.epcc.ed.ac.uk:28443/npm-dt/query.jsp

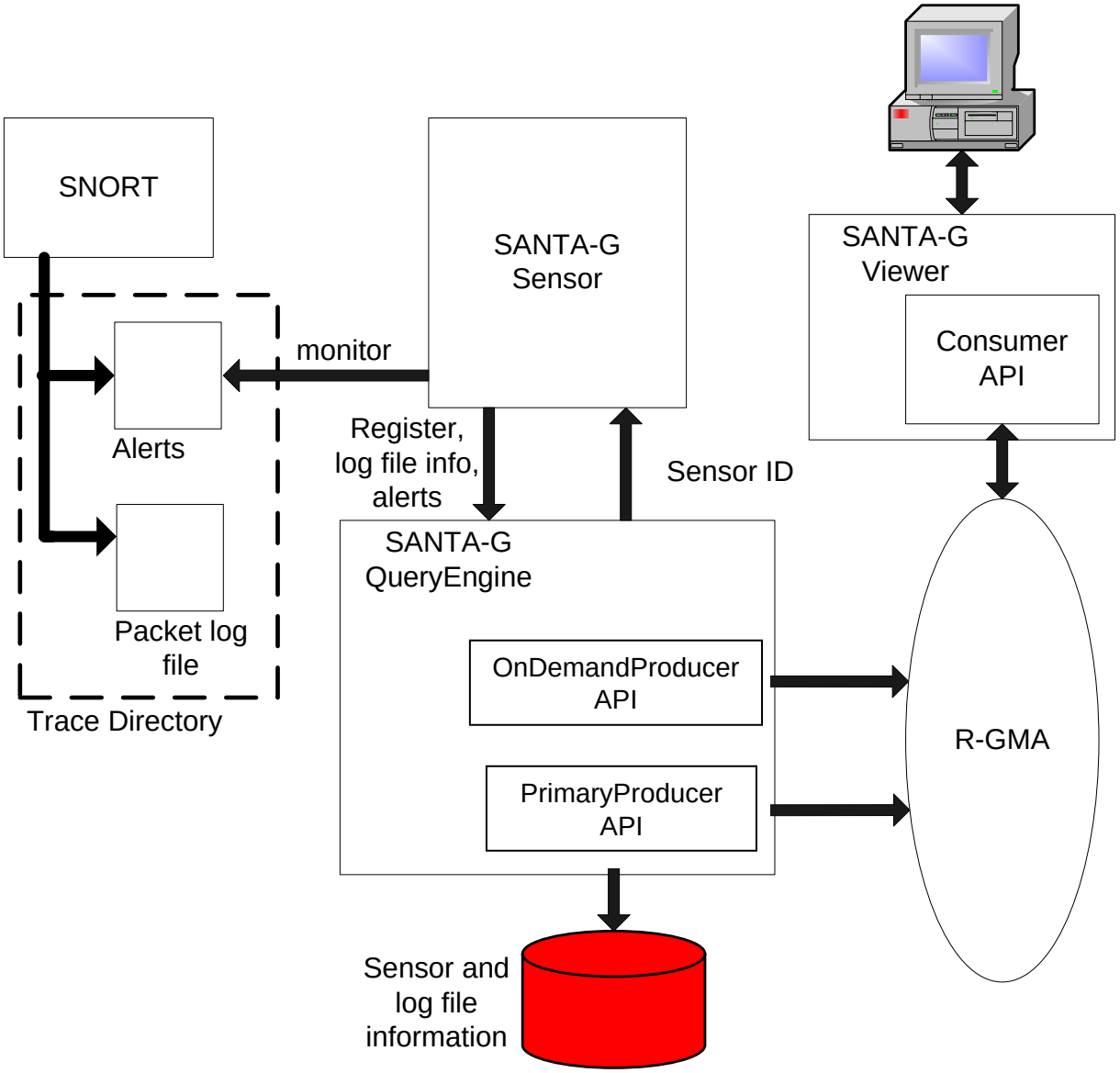
NPM Diagnostic Tool



NPM DT Scenario - results



Intrusion Detection

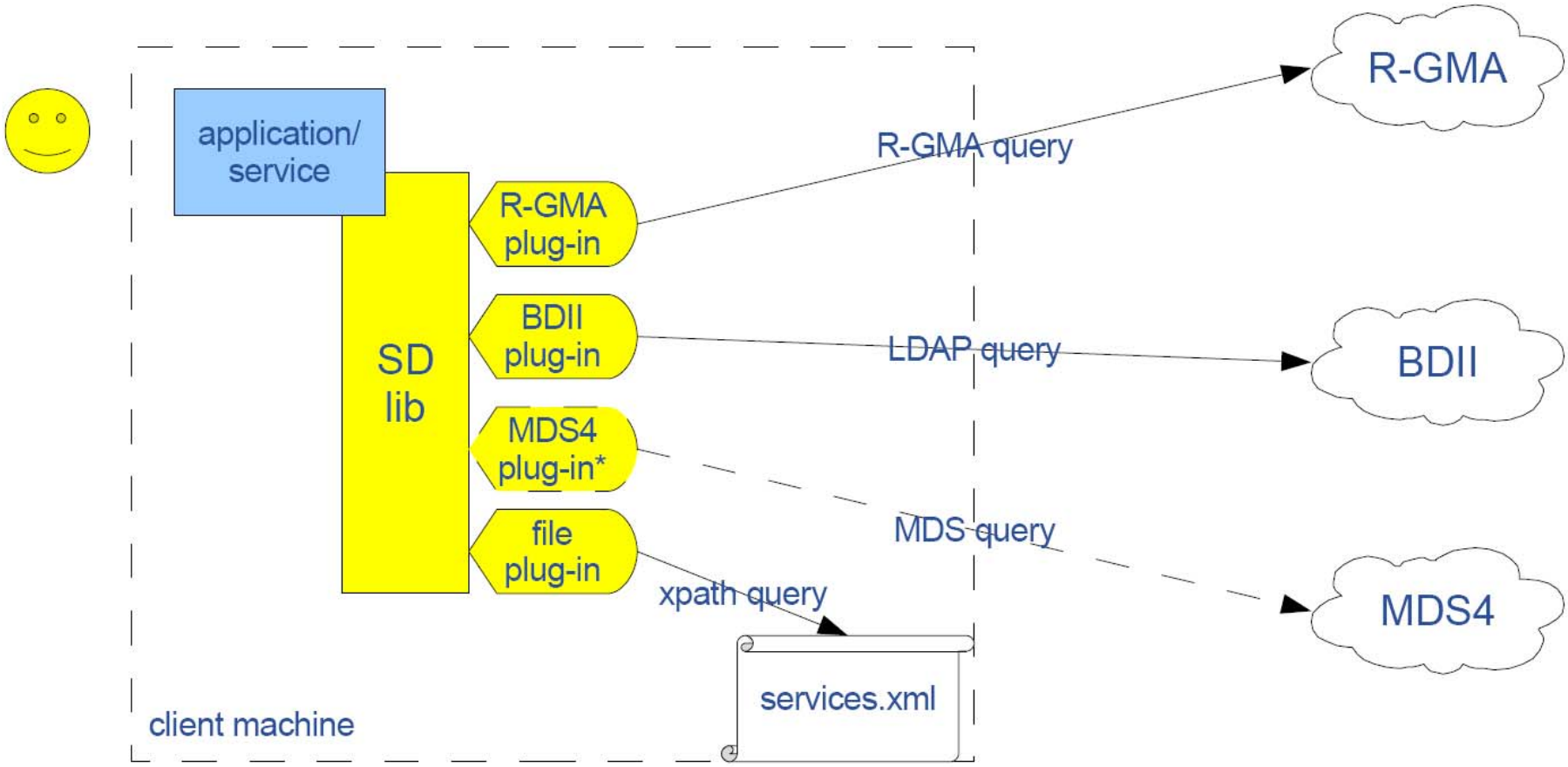


The Grid intrusion detection work is now within the Interactive European Grid (http://www.interactive-grid.eu) project, as part of the JRA workpackage, and is known as Active Security (http://www.grid.ie/i2g)

Service Discovery

- **Questions to answer:**
  - “I am at CERN, in 'dteam' VO. Where is a MyProxy server?”
  - glite-sd-query -t myproxy -s CERN-PROD
- **Service Discovery offers:**
  - client API (library) to hide the differences
  - plug-in architecture to simplify dependencies
  - uses the subset of Glue schema as data model
  - simple API, no complex queries
  - CLI for other tools and testing
- **Plug-ins for:**
  - BDII
  - R-GMA
  - MDS4 (not yet)
  - File (only for testing)

Service Discovery



TCD R-GMA related projects

TCD: Trinity College Dublin

- gridFS: a grid filesystem
- InfoGrid: a grid using an information model
- Keith Rochford's work on grid service monitoring
- Adaptive eLearning: R-GMA is the first course
- Shared memory for grids (SMG)

R-GMA APIs

- **APIs exist in Java, C, C++, Python.**
  - For clients (servlets contacted behind the scenes)
- **They include methods for…**
  - Creating consumers
  - Creating primary and secondary producers
  - Setting type of queries, type of produces, retention periods, time outs…
  - Retrieving tuples, inserting data
  - …
- **You can create your own Producer or Consumer.**

Overview of practical

- We will use a client that gives command-line interfaces to both consumers and producers
- We will explore the tables on the R-GMA service provided on GILDA
- Use a table that is set up for training purposes to produce and consume data

Now please follow the “more information” link

R-GMA practical
html page

Batch Mode

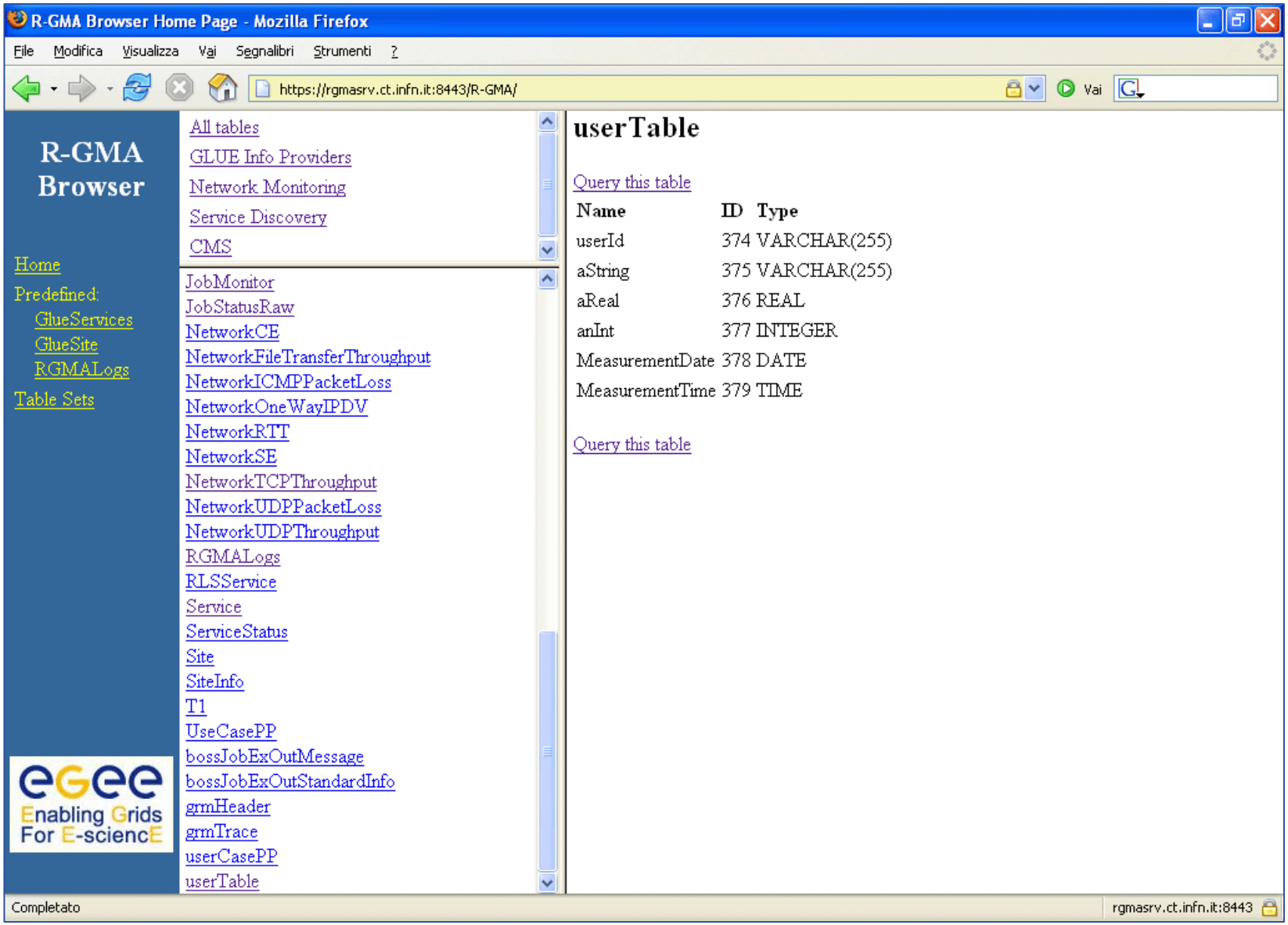
- The command line tool can be used in batch mode in three ways:
  - **`rgma -c <command>`**
    Executes <command> and exits.
    The –c option may be specified more than once.
  - **`rgma -f <file>`**
    Executes commands in <file> sequentially then exits.
    Each line should contain one command.
  - Embedded in a shell script

R-GMA Browser

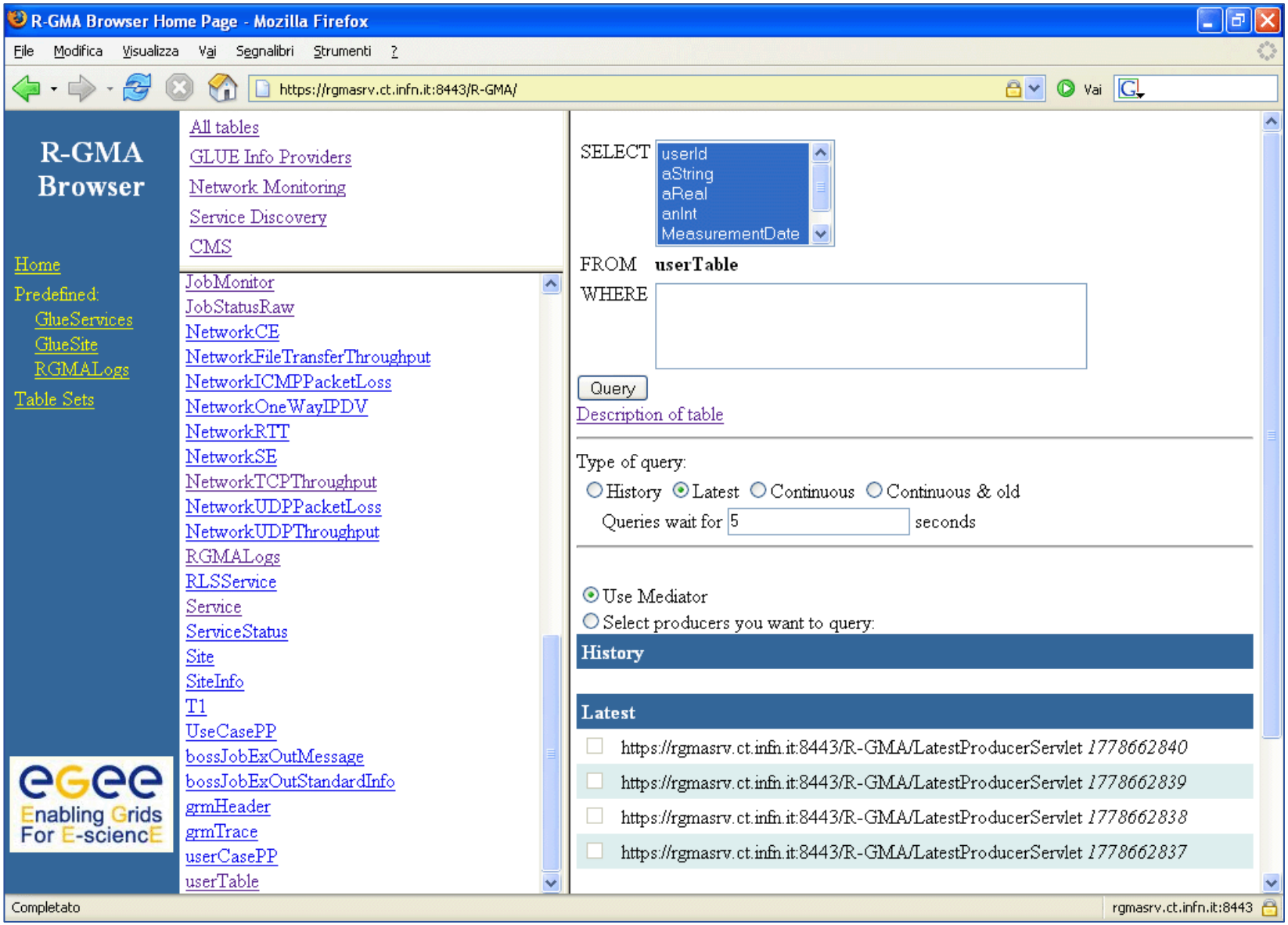
www.eu-egee.org

Information Society

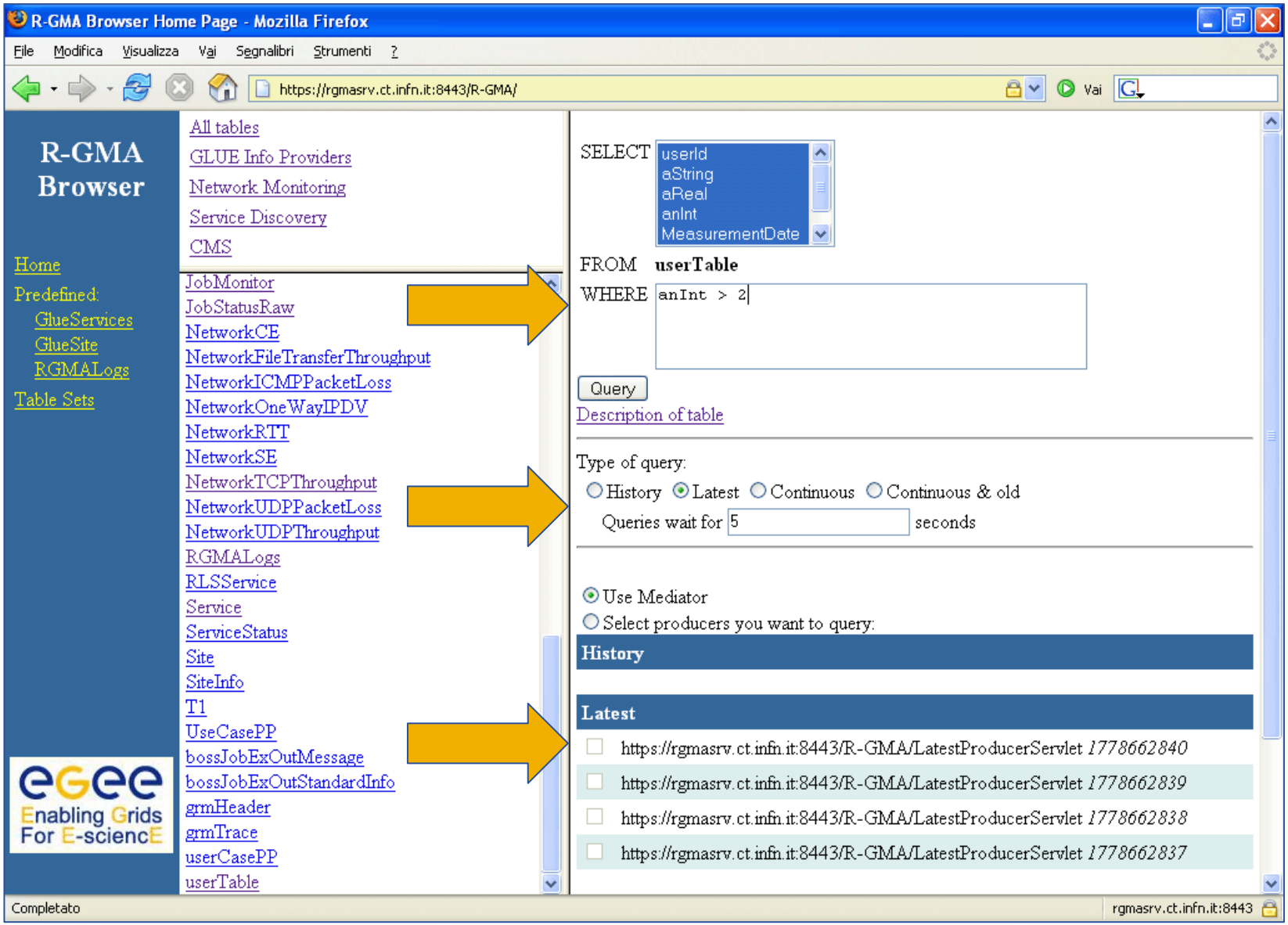
Table description



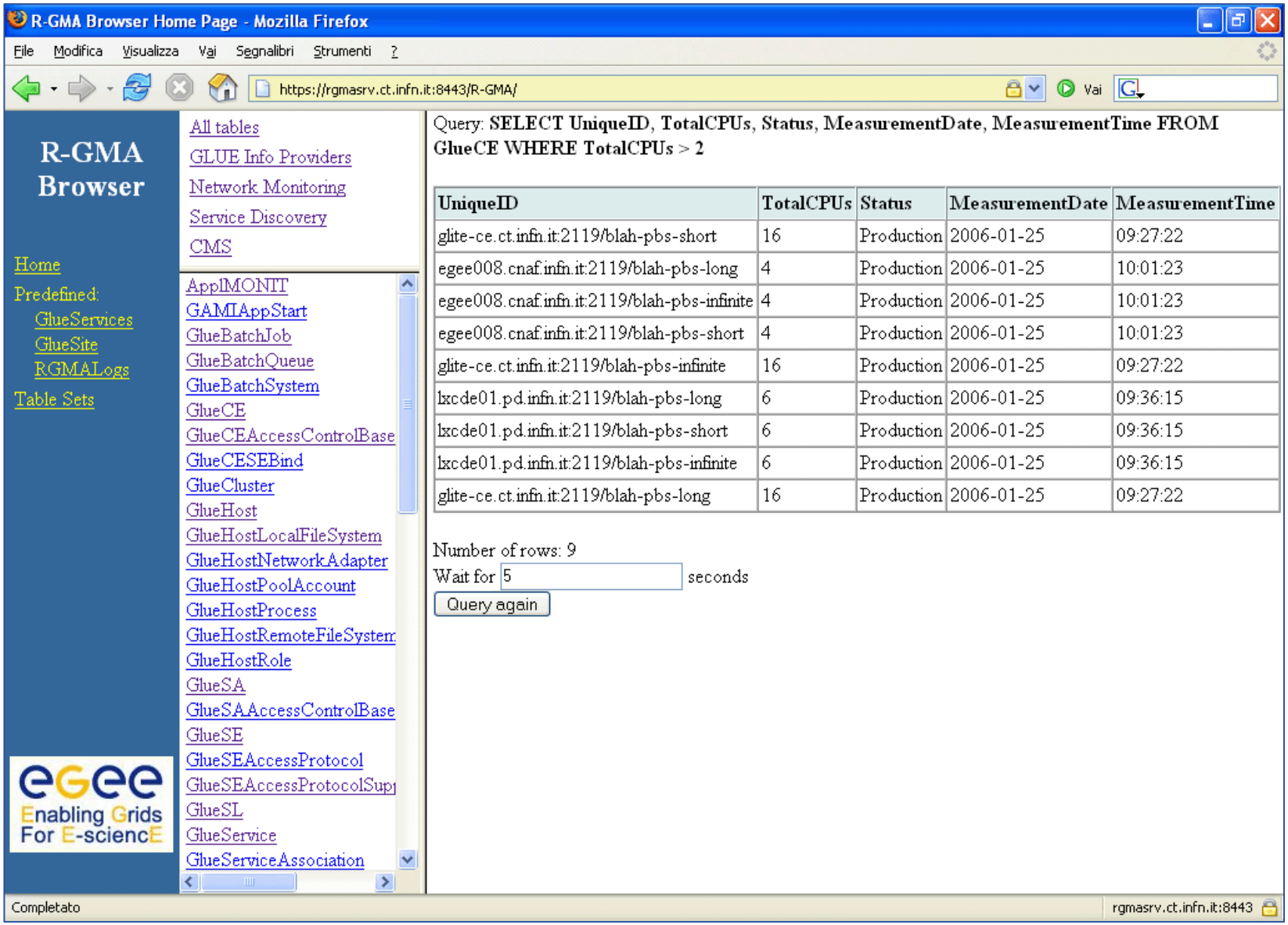
R-GMA Browser as Consumer



Query from R-GMA Browser



Query Results



More information

- **R-GMA overview page.**
  - http://www.r-gma.org/
- **R-GMA in EGEE**
  - http://hepunx.rl.ac.uk/egee/jra1-uk/
- **R-GMA command line tool**
  - http://hepunx.rl.ac.uk/egee/jra1-uk/glite-r1/command-line.pdf
- **R-GMA Browser Home Page**
  - https://rgmasrv.ct.infn.it:8443/R-GMA/