

RGMA: Architecture and use

Antonio Delgado Peris

LCG Experiment Integration and Support. CERN IT.

First gLite tutorial on GILDA, Catania, 13-15.06.2005

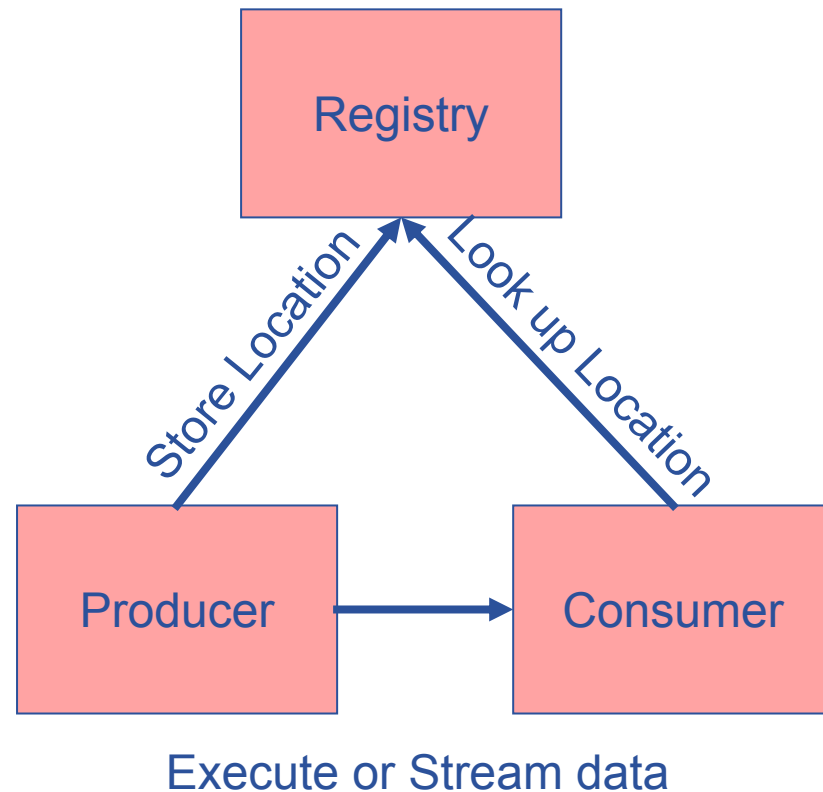
This tutorial covers the following:

- Overview.
- R-GMA in-depth.
- Installation and testing.
- Using R-GMA.

Overview.

- **Relational Grid Monitoring Architecture (R-GMA)**
 - Developed as part of the EuropeanDataGrid Project (EDG)
 - Now as part of the Enabling Grids for E-science (EGEE) project.
 - Based the Grid Monitoring Architecture (GMA) from the Global Grid Forum (GGF).
- **Uses a relational data model.**
 - Data is viewed as a table.
 - Data structure defined by the columns.
 - Each entry is a row (tuple).
 - Queried using Structured Query Language (SQL).
- **A prerequisite to understating R-GMA**
 - Understand the relational data model and SQL.

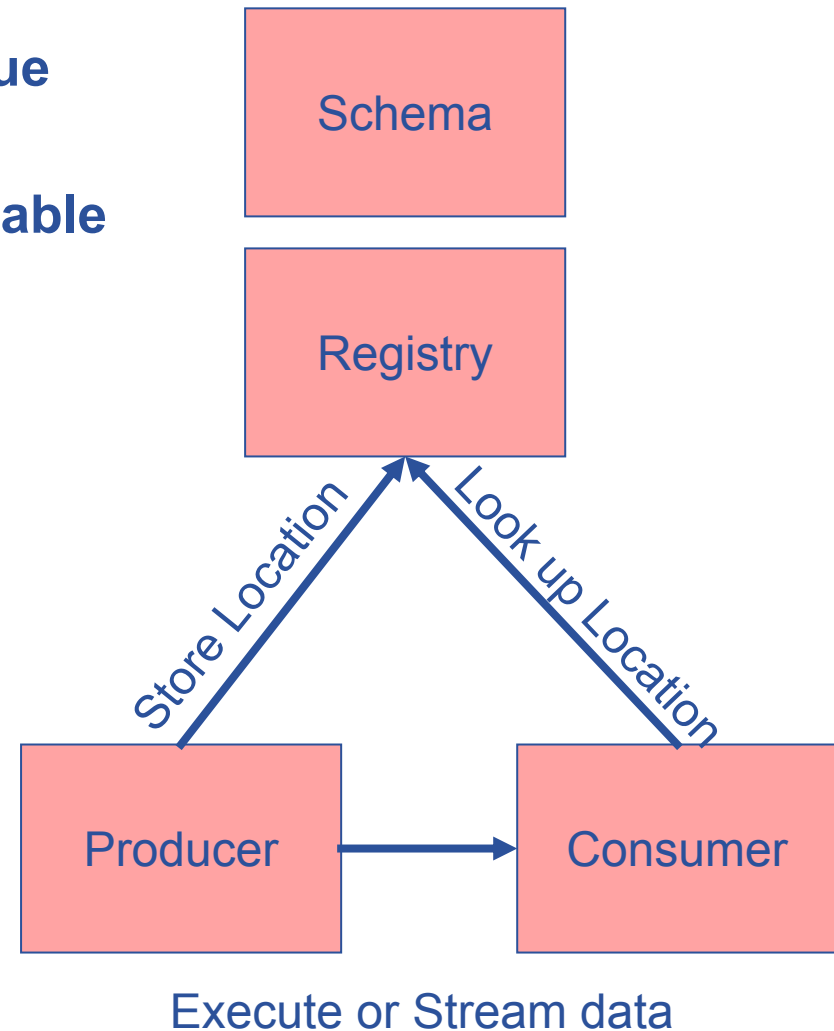
- The Producer stores its location (URL) in the Registry.
- The Consumer looks up producer URLs in the Registry.
- The Consumer contacts the Producer to get all the data.
- Or the Consumer can listen to the Producer for new data.



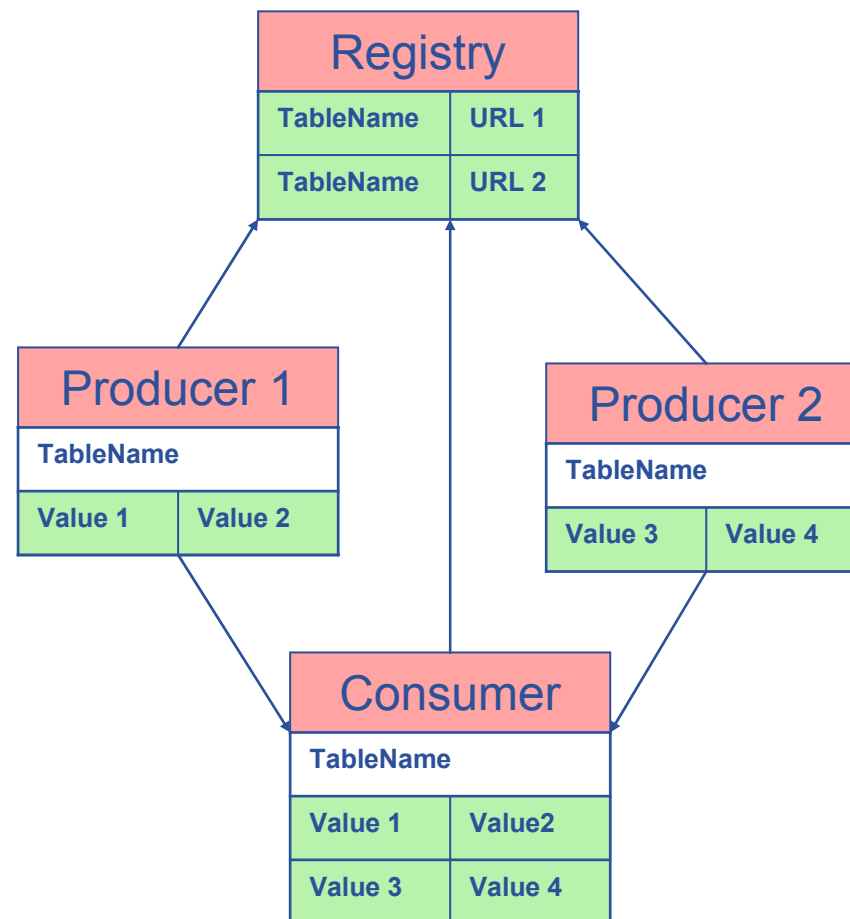
name	ID	birth	Group
Tom	4	1977-08-20	HR

```
SELECT * FROM people WHERE group='HR'
```

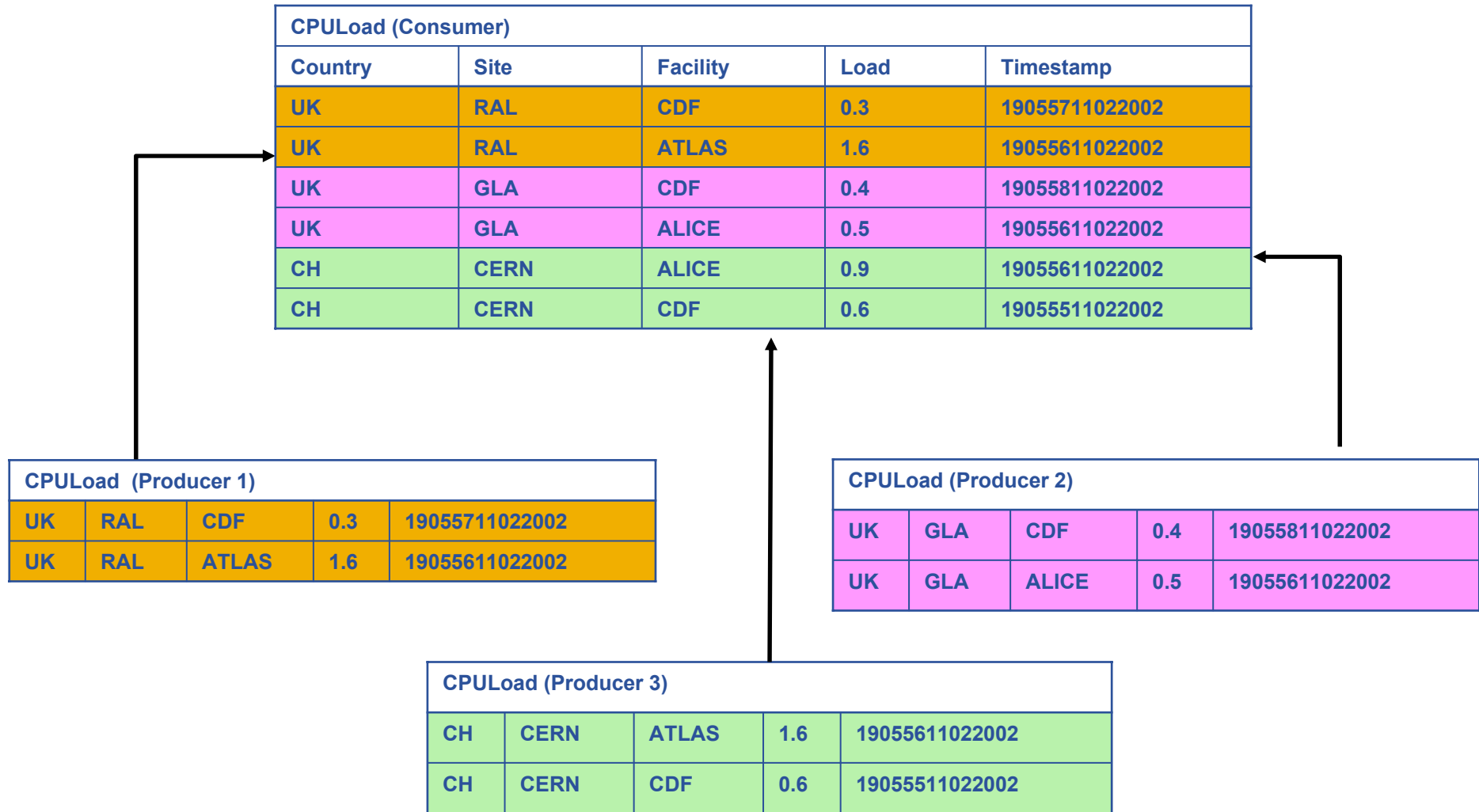
- The data model is relational.
- The table definition is globally unique and is stored in the Schema.
- The Registry stores the Producers table name as well as the URL.
- The data is inserted in the form of a tuple.
- The Consumer gets the tuple from Producer.
- **Producers**
 - publish: SQL “INSERT”
- **Consumers**
 - collect: SQL “SELECT”



- The Consumer will get all the URLs that could satisfy the query.
- The Consumer will connect to all the Producers.
- Producers that can satisfy the query will send the tuples to the Consumer.
- The Consumer will merge these tuples to form one result set.



Select * from CPUload



- **The Mediator is the intelligence of R-GMA**
 - Not a single component, but distributed.
 - Enables queries to be accurately and efficiently returned.
- **The table name is stored next to the URL in the Registry.**
 - For simple queries, only the URLs that can answer query are passed to the Consumer.
 - If the query has a predicate, only the URLs that could satisfy the query will be passed to the Consumer.
- **The Mediator will also try to do joins.**
 - For complex queries the query must use a Producer with a database backend (secondary producer).
 - Merges and produces the resulting result set.
- **The Consumers URL and query is also stored in the Registry.**
 - Enables the Registry to notify listening Consumers about new Producers.

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

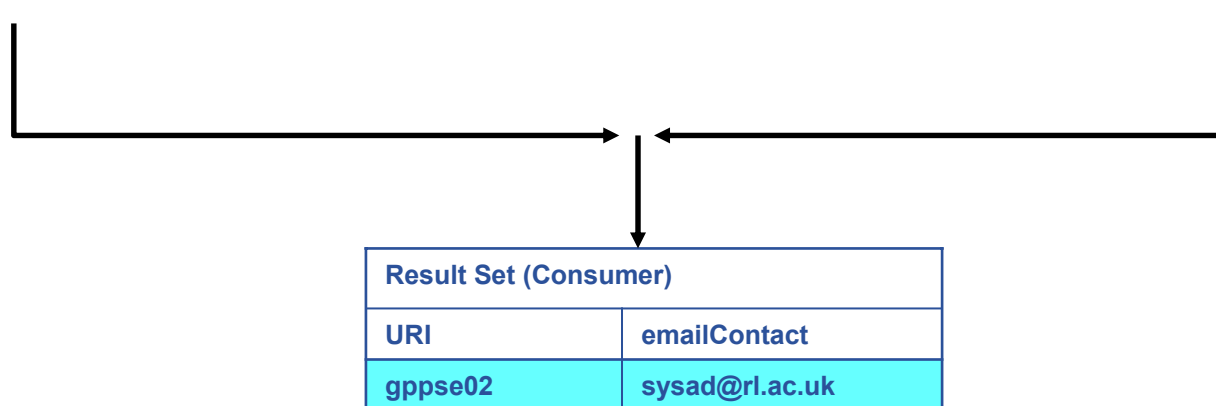
CPULoad (Producer 1)				
UK	RAL	CDF	0.3	19055711022002
UK	RAL	ATLAS	1.6	19055611022002

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

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

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



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

- **There are two different types of producers.**
 - Primary Producer.
 - In memory or database.
 - Publish data.
 - Secondary Producers.
 - Republish data.
- **Producers can have different properties.**
 - Stream.
 - Sends the tuple directly to the Consumer.
 - Latest.
 - Only stores that latest tuple for the primary key.
 - Latest Retention Period, property of tuple.
 - History.
 - Stores all tuples.
 - History Retention Period, property of the producer.

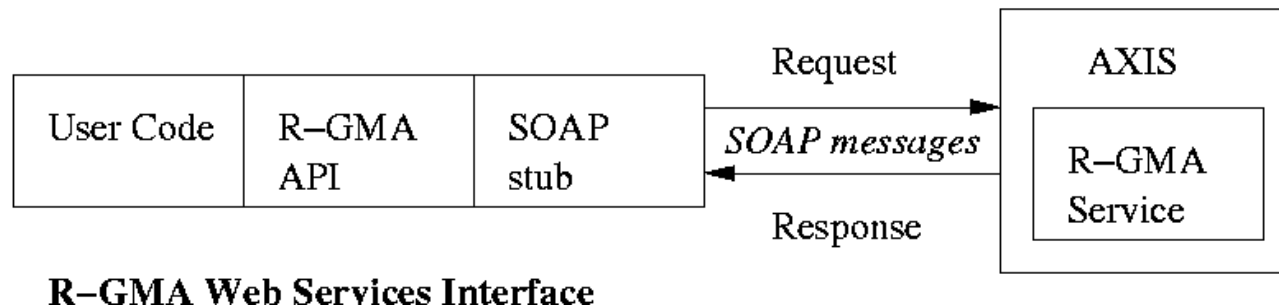
- **A Secondary Producer is a joint Consumer/Producer.**
 - Also known as an *Archiver*.
 - The Consumer listens to the streaming Primary Producers.
 - Continuous query to the streaming Primary Producer.
 - New tuples are inserted into a database or held in memory.
 - The Producer re-publishes these tuples.
 - Currently as History or Latest.

- **Most information is published using streaming Primary Producers.**
 - Secondary Producers can catch these tuples and re-publish them.
 - Necessary for joins in queries

- **A Secondary Producer can be used to populate a DB.**
 - Can be accessed directly given correct permissions.
 - Use this database for another purpose.
 - Eg. custom visualisation.

R-GMA In-depth

- **Consumers and producers are composed of:**
 - Java servlet running as a resource of a R-GMA service
 - The R-GMA service sits in an application server (in a *MON box*)
 - It creates the servlets upon request
 - Client running on the same LAN (UI, WNs, SE, RB)
 - They use SOAP to exchange messages

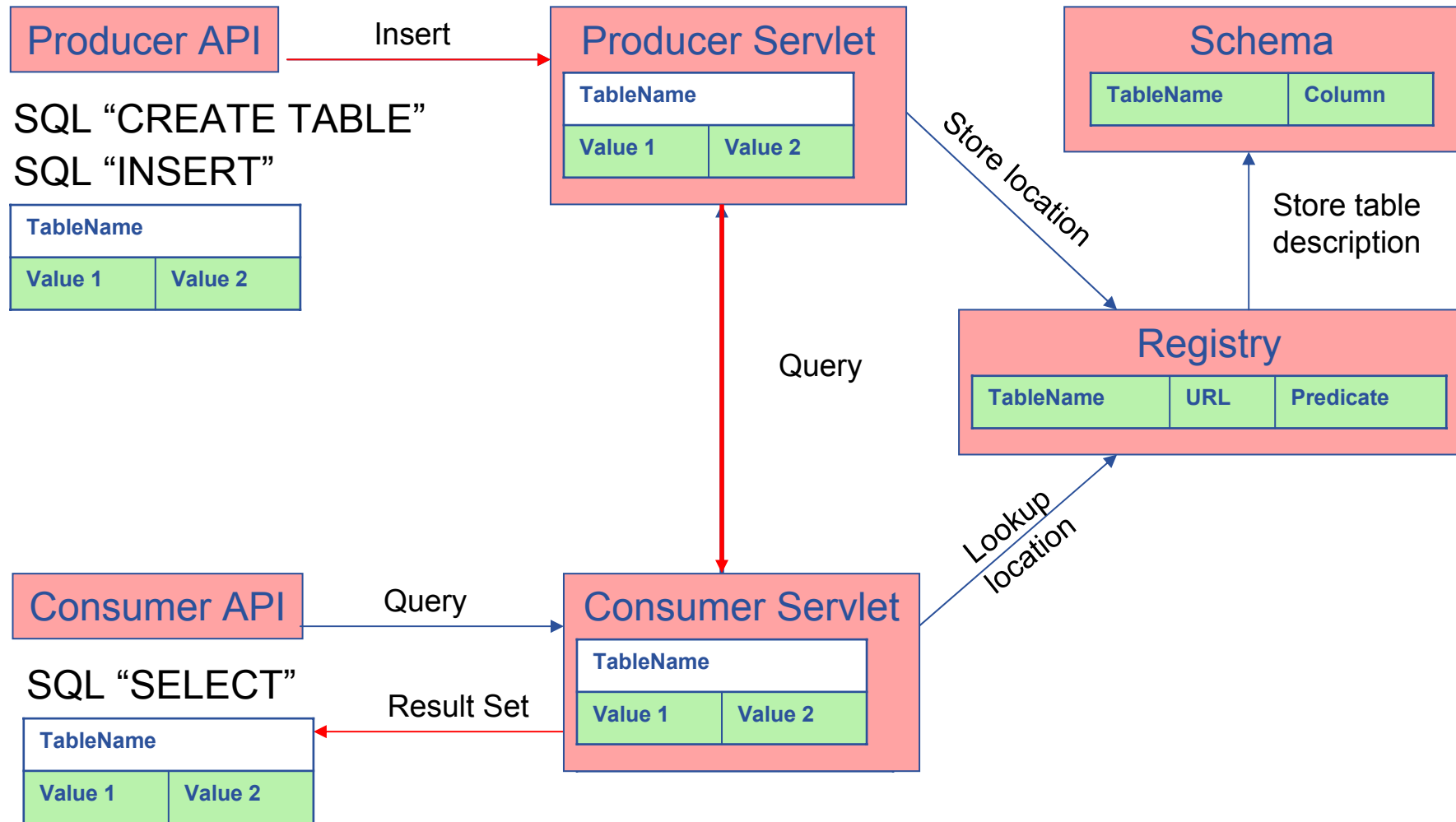


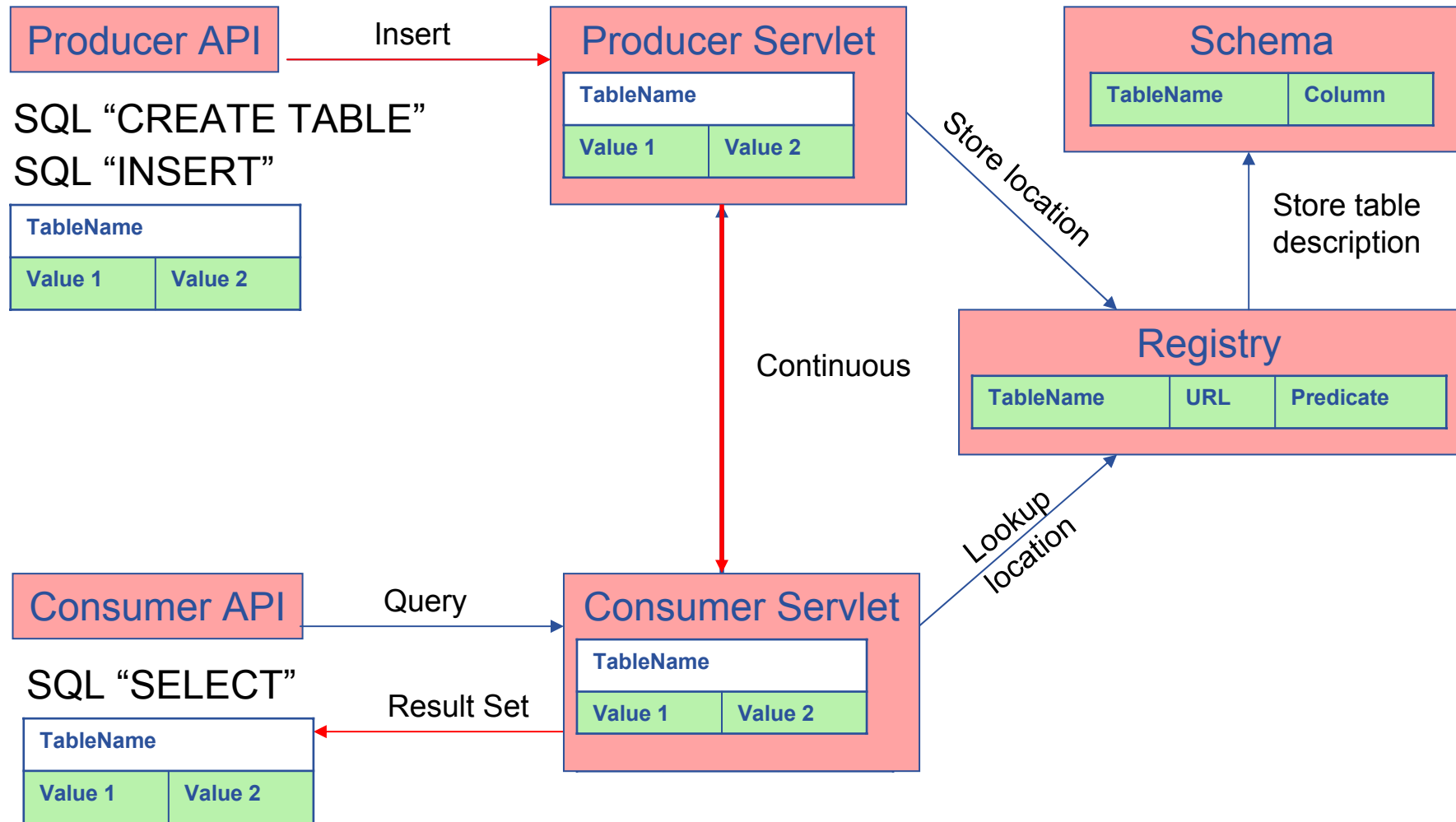
- **The servlet does most of the work.**
 - Executes actions of the client (SOAP message)
 - Communicates with the Consumer and Registry
 - The client inserts the data and controls the servlet

- **The Schema servlet stores the table definitions.**
 - Ensures they are globally unique.
 - Stores them in a database.
- **The Registry servlet stores the Producer URLs.**
 - It also stores the table name and the predicate.
 - Stores them in a database.
 - Consumer URLs and queries are also stored.
 - This is so that the Registry can notify continuous Consumers of new Producers.

- **There is only one registry and only one schema**
 - Currently there is no replication of the Schema
 - But it is not a performance issue
- **The Registry can be replicated.**
 - The Registries swap data between them
 - All have the global view
 - But only manage a subset of Producers
 - All producers can be found by going to only one Registry
 - This is currently not used!

- **Security is available in R-GMA**
 - Uses https instead of http.
 - Authentication via Grid Certificates.
 - Authorization will be coming soon.
 - ... But not currently used in LCG!
- **Soft registration:**
 - For producer and consumer servlets
 - They will close after the termination interval
 - The client needs periodically to show a sign of life
 - For entries in the registry
 - Producers must contact periodically (automatically done by R-GMA)





Installation and testing.

- **Deployment**
 - What Registry and Schema will you use
 - This defines the scope of the system.
 - Will you run your own?
 - All nodes should have the R-GMA client.
 - You will need a local R-GMA server.
 - Producer and Consumer servlets.
- **Read the R-GMA installation documentation!!!**
 - Or you could use YAIM ☺.
 - Install the R-GMA RPMs and dependencies.
 - Run the configure R-GMA server script.
 - This can also configure a Schema and Registry.
 - Run the configure R-GMA client script.

- **R-GMA client configuration**
 - File: `$RGMA_HOME/etc/rgma/rgma.conf`
 - Simple properties file: `param=value`
 - Indicates where the servlets are located
 - Indicates where the registry and the schema are located

- **R-GMA server configuration**
 - File: `$RGMA_HOME/etc/rgma-server/rgma-server.conf`
 - XML parameters file
 - Created automatically by the RGMA server setup script

- **Server logging file**
 - Logging file: `/tmp/rgma-server.log`
 - Customization of logging: `$RGMA_HOME/etc/rgma-server/log4j.properties`

- **R-GMA comes with two testing scripts.**
 - /opt/edg/bin/rgma-client-check
 - Checks the client.
 - /opt/edg/bin/rgma-server-check
 - Checks the server.
- **The client check tries to publish data using various APIs**
 - Verifies that the data is available via R-GMA.
- **The server check is for the node hosting the servlets.**
 - Tries to connect to the servlets.
 - Checks the communication with the Registry.
 - Checks that the correct ports are open.
- **If any of these scripts fail you have a problem!**
 - Time to troubleshoot!!!

- **R-GMA is a complex distributed system.**
- **Troubleshooting is not that difficult.**
 - Take a simple, systematic approach.
- **Is the Schema and Registry okay?**
- **Are the servlets okay?**
- **Is the client okay?**
- **Each Servlet has a getStatus method to check it is OK.**
 - `http://<host>:8080/R-GMA/<Servlet>/getStatus`
- **Not a supported method, but useful if you know it ;)**

Using R-GMA.


- **The easiest way to try out R-GMA.**
 - It is installed on the machine running the Registry and Schema:
<http://lcgic01.gridpp.rl.ac.uk:8080/R-GMA/index.html>
 - You can also install it along with the Producer and Consumer Servlets.
- **Using the Browser you can do the following.**
 - Browse the tables in the schema.
 - Look at the table definitions.
 - See all the available producers for a table.
 - Query a table.
 - Query only selected producers.

**R-GMA
Browser**

[Home](#)

Predefined:

- [Services](#)
- [Site](#)
- [Table Sets](#)



Enabling Grids
For E-science

[All tables](#)

[EDG Info Providers](#)

[Network Monitoring](#)

[CMS](#)

[ArchiverTestTable](#)

[ComputingElementQueue](#)

[DeclarableTestTable](#)

[GAMIAppStart](#)

[GkRecords](#)

[GlueBatchJob](#)

[GlueBatchQueue](#)

[GlueBatchSystem](#)

[GlueCE](#)

[GlueCEAccessControlBaseRule](#)

[GlueCESEBind](#)

[GlueCluster](#)

[GlueHost](#)

[GlueHostLocalFileSystem](#)

[GlueHostNetworkAdapter](#)

[GlueHostPoolAccount](#)

[GlueHostProcess](#)

[GlueHostRemoteFileSystem](#)

[GlueHostRole](#)

[GlueSA](#)

[GlueSAAccessControlBaseRule](#)

[GlueSE](#)

[GlueSEAccessProtocol](#)

[GlueSEAccessProtocolSupportedSe](#)

[GlueSL](#)

[GlueService](#)

[GlueServiceAccessControlRule](#)

[GlueSubCluster](#)

[GlueSubClusterSoftwareRunTimeE](#)

[GlueVO](#)

[GocLookupCountries_v0_1](#)

[GocMaintenance_v0_1](#)

[GocNode_v0_1](#)

[GocSite_v0_1](#)

```

SELECT Status
       WorstResponseTime
       EstimatedResponseTime
       FreeCpus
       Priority
FROM   GlueCE
WHERE

```

Description of table

Type of query:

History Latest Continuous Continuous & old

Queries wait for seconds

Use Mediator

Select producers you want to query:

History

http://lxn1191.cern.ch:8080/R-GMA/DBProducerServlet_1368781140

Latest

http://mon001.m45.ihep.su:8080/R-GMA/LatestProducerServlet_2101942584

http://lxn1191.cern.ch:8080/R-GMA/LatestProducerServlet_1060597273

http://lcmgon01.gridpp.rl.ac.uk:8080/R-GMA/LatestProducerServlet_1372550532

Continuous

http://t2mon02.physics.ox.ac.uk:8080/R-GMA/StreamProducerServlet_744237223

http://lcm-se.ecm.ub.es:8080/R-GMA/StreamProducerServlet_138509328

http://dgbdi0.icepp.jp:8080/R-GMA/StreamProducerServlet_1686274683

http://lxb2059.cern.ch:8080/R-GMA/StreamProducerServlet_1287158431

http://se.ui.savba.sk:8080/R-GMA/StreamProducerServlet_1637339420

http://lcm001.cern.ch:8080/R-GMA/StreamProducerServlet_666003552

R-GMA
Browser

[Home](#)
[Predefined:](#)
[Services](#)
[Site](#)
[Table Sets](#)

[All tables](#)
[EDG Info Providers](#)
[Network Monitoring](#)
[CMS](#)

[ArchiverTestTable](#)
[ComputingElementQueue](#)
[DeclarableTestTable](#)
[GAMIAppStart](#)
[GkRecords](#)
[GlueBatchJob](#)
[GlueBatchQueue](#)
[GlueBatchSystem](#)
[GlueCE](#)
[GlueCEAccessControlBaseRule](#)
[GlueCESEBind](#)
[GlueCluster](#)
[GlueHost](#)
[GlueHostLocalFileSystem](#)
[GlueHostNetworkAdapter](#)
[GlueHostPoolAccount](#)
[GlueHostProcess](#)
[GlueHostRemoteFileSystem](#)
[GlueHostRole](#)
[GlueSA](#)
[GlueSAAccessControlBaseRule](#)
[GlueSE](#)
[GlueSEAccessProtocol](#)
[GlueSEAccessProtocolSupportedSe](#)
[GlueSL](#)
[GlueService](#)
[GlueServiceAccessControlRule](#)
[GlueSubCluster](#)
[GlueSubClusterSoftwareRunTimeEi](#)
[GlueVO](#)
[GocLookupCountries_v0_1](#)
[GocMaintenance_v0_1](#)
[GocNode_v0_1](#)
[GocSite_v0_1](#)

Query: `SELECT UniqueID, FreeCpus FROM GlueCE`

UniqueID	FreeCpus
ce32.hep.ntua.gr:2119/jobmanager-lcgpbs-see	8
ce32.hep.ntua.gr:2119/jobmanager-lcgpbs-esr	8
mu6.matrix.sara.nl:2119/jobmanager-pbs-astrop	42
epgce1.ph.bham.ac.uk:2119/jobmanager-lcgpbs-alice	26
epgce1.ph.bham.ac.uk:2119/jobmanager-lcgpbs-lhcb	26
marseillece01.mrs.grid.cnrs.fr:2119/jobmanager-pbs-esr	24
mu6.matrix.sara.nl:2119/jobmanager-pbs-emutd	42
mu6.matrix.sara.nl:2119/jobmanager-pbs-esr	42
mu6.matrix.sara.nl:2119/jobmanager-pbs-nadc	42
mu6.matrix.sara.nl:2119/jobmanager-pbs-ncf	42
t2ce02.physics.ox.ac.uk:2119/jobmanager-lcgpbs-cdf	42
lxb2018.cern.ch:2119/jobmanager-lcgpbs-dteam	0
ce32.hep.ntua.gr:2119/jobmanager-lcgpbs-cms	8
cmslcgce.fnal.gov:2119/jobmanager-lcgcondor-atlas	127
cmslcgce.fnal.gov:2119/jobmanager-lcgcondor-cms	127
cmslcgce.fnal.gov:2119/jobmanager-lcgcondor-dteam	128
ce.phy.bg.ac.yu:2119/jobmanager-lcgpbs-dteam	23
marseillece01.mrs.grid.cnrs.fr:2119/jobmanager-pbs-dteam	24
marseillece01.mrs.grid.cnrs.fr:2119/jobmanager-pbs-biomed	24
lcg-ce.ecm.ub.es:2119/jobmanager-pbs-dteam	35
ce.ui.savba.sk:2119/jobmanager-pbs-atlas	22
ce.ui.savba.sk:2119/jobmanager-pbs-alice	22
ce.ui.savba.sk:2119/jobmanager-pbs-dteam	22
ce.ui.savba.sk:2119/jobmanager-pbs-esr	22
lcg-gridka-ce.fzk.de:2119/jobmanager-pbspro-short	29
lcg-gridka-ce.fzk.de:2119/jobmanager-pbspro-long	29
lcg-gridka-ce.fzk.de:2119/jobmanager-pbspro-default	29
lcg-gridka-ce.fzk.de:2119/jobmanager-pbspro-opt32	29
lcg-gridka-ce.fzk.de:2119/jobmanager-pbspro-magic	29

- **R-GMA has a command line interface.**
 - This interface has a similar look an feel to the MySQL DB.
- **Can be run interactively or on the command line.**
 - Useful for debugging SQL statements!

```
rgma -c "select Name from Site where Name='RALPP-LCG'"
+-----+
| Name      |
+-----+
| RALPP-LCG |
+-----+
1 Rows in set
```

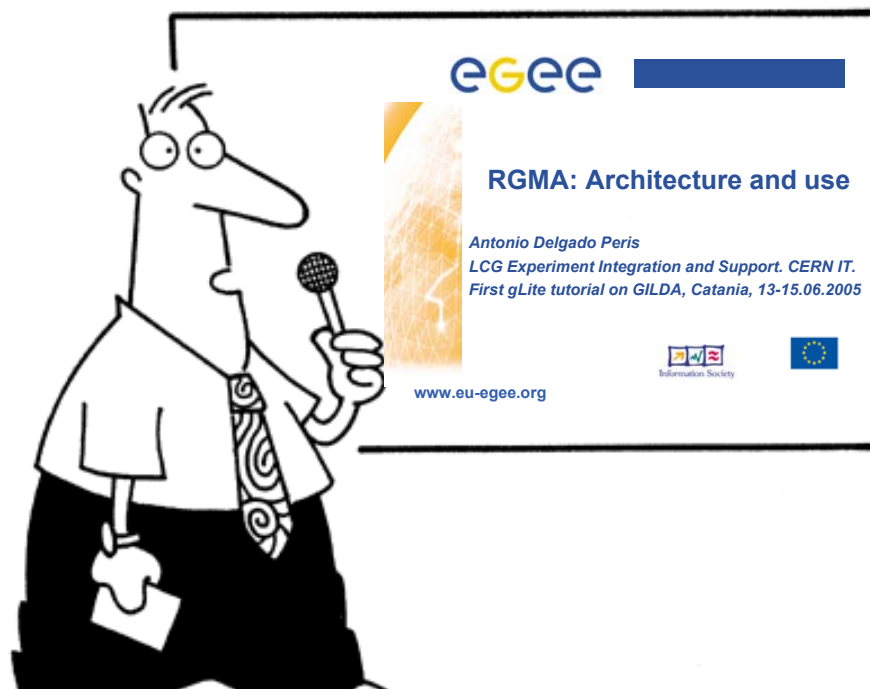
```
rgma> select Name from Site where Name='RALPP-LCG'
+-----+
| Name      |
+-----+
| RALPP-LCG |
+-----+
1 Rows in set
```

- Summary of commands:**

help [<command>]	Information (general or about command)
exit / quit	Exit the R-GMA command line
show [tables producers of <table> ...]	Show the tables in the schema, the producers of a given table...
describe <table>	Show column names and types for <table>
SQL select	Query R-GMA
set query latest continuous historical	Set type of query
SQL insert	Insert tuple into the primary producer
Secondaryproducer <table>	Declare table to be consumed and republished by secondary producer
set [secondary]producer latest continuous historical	Set supported type for the producer or the secondary producer
set [timeout maxage] <timeout> [<units>]	Set timeout for queries or maximum age of tuples to return

- **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.**
- **Documentation exists for all APIs.**
 - Read the documentation!!
 - Example code is in the documentation.

- **R-GMA overview page.**
 - <http://www.r-gma.org/>
- **R-GMA in EGEE**
 - <http://hepunx.rl.ac.uk/egee/jra1-uk/>
- **R-GMA Documentation**
 - <http://hepunx.rl.ac.uk/egee/jra1-uk/LCG/doc/>



Hope you enjoy this lecture.
Thank you for attending !