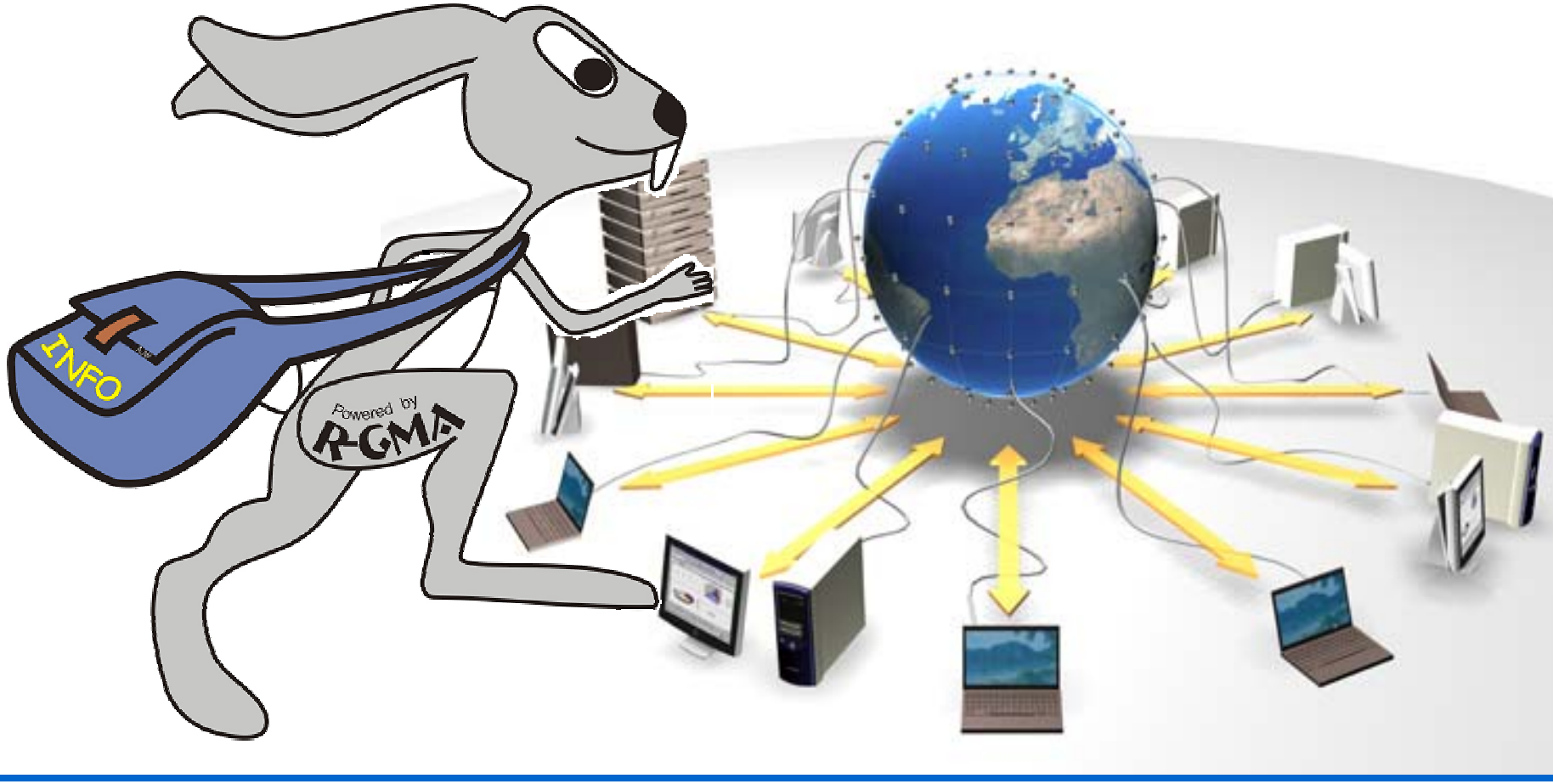
Information and Monitoring

The European DataGrid Project Team

http://www.eu-datagrid.org



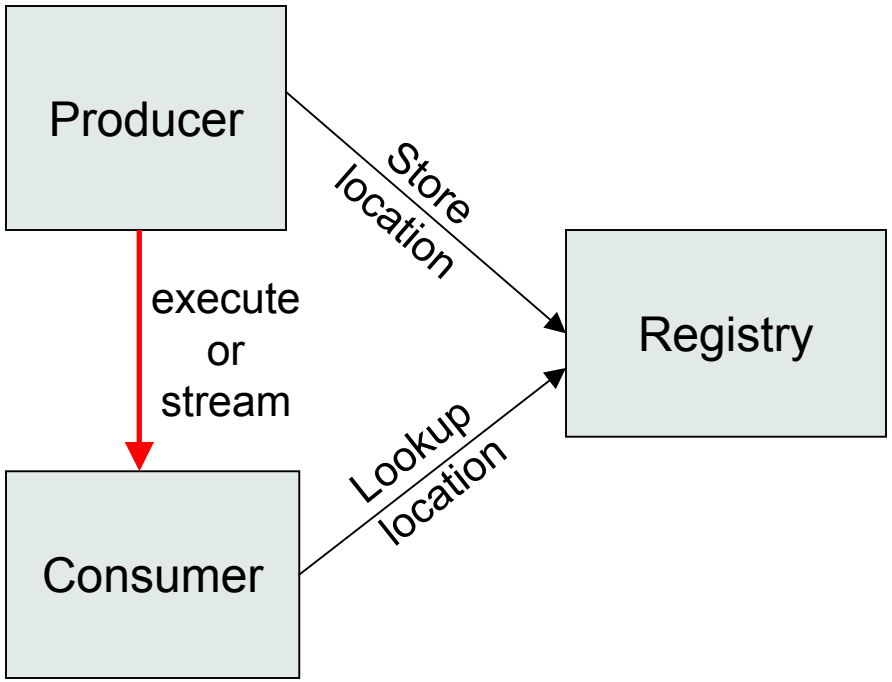
Contents

- Grid Information Systems
- GMA and R-GMA
- Tools and APIs

Features of a grid information system

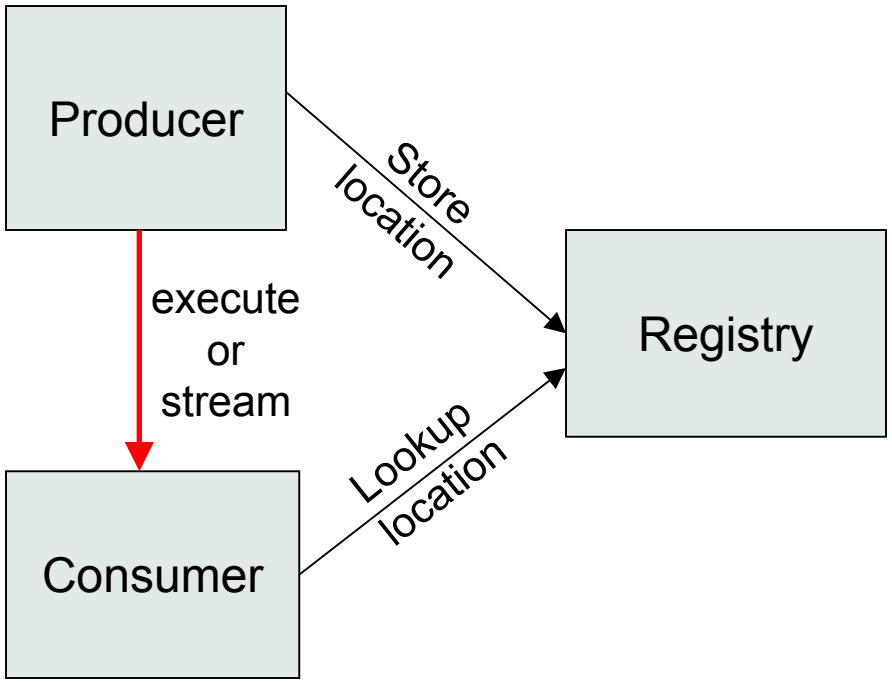
- Provides information on both:
  - The Grid itself
    - Mainly for the middleware packages
    - The user may query it to understand the status of the Grid
  - Grid applications
    - For users
- Flexible infrastructure
  - Able to cope with nodes in a distributed environment with an unreliable network
  - 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
- All R-GMA tuples are time-stamped

Example: 2 tables

- Service

URI	VARCHAR(255)	URI to contact the service
VO	VARCHAR(50)	Where info should be published – or an empty string to indicate all
type	VARCHAR(50)	Type of service
emailContact	VARCHAR(50)	The e-mail of a human being to complain to
site	VARCHAR(50)	Domain name of site hosting the service
secure	VARCHAR(1)	'y' or 'n' - indicates whether or not this is a secure service
majorVersion	INT	Version of protocol not implementation
minorVersion	INT	Version of protocol not implementation
patchVersion	INT	Version of protocol not implementation

- ServiceStatus

URI	VARCHAR(255)	URI to contact the service
status	INT	status code. 0 means the service is up.
message	VARCHAR(255)	Message corresponding to status code

SQL example 1

- SELECT DISTINCT type FROM Service

```
+--------------------------------------+
| type                                 |
+--------------------------------------+
| GridFTP                              |
| GRIS                                 |
| RFIO                                 |
----------------------------------------

| R-GMA.ResilientStreamProducerService |
| R-GMA.ArchiverService                |
| R-GMA.StreamProducerService          |
| R-GMA.CanonicalProducerService       |
| R-GMA.DBProducerService              |
| R-GMA.LatestProducerService          |
| GIN                                  |
| R-GMA.RegistryService                |
| R-GMA.SchemaService                  |
| R-GMA.BrowserService                 |
| GOUT                                 |
| edg-netmon                           |
| edg-iperf                            |
| edg-udpmon                           |
| myproxy                              |
| edg-pinger                           |
+--------------------------------------+
25 Rows in set
```

SQL Example 2

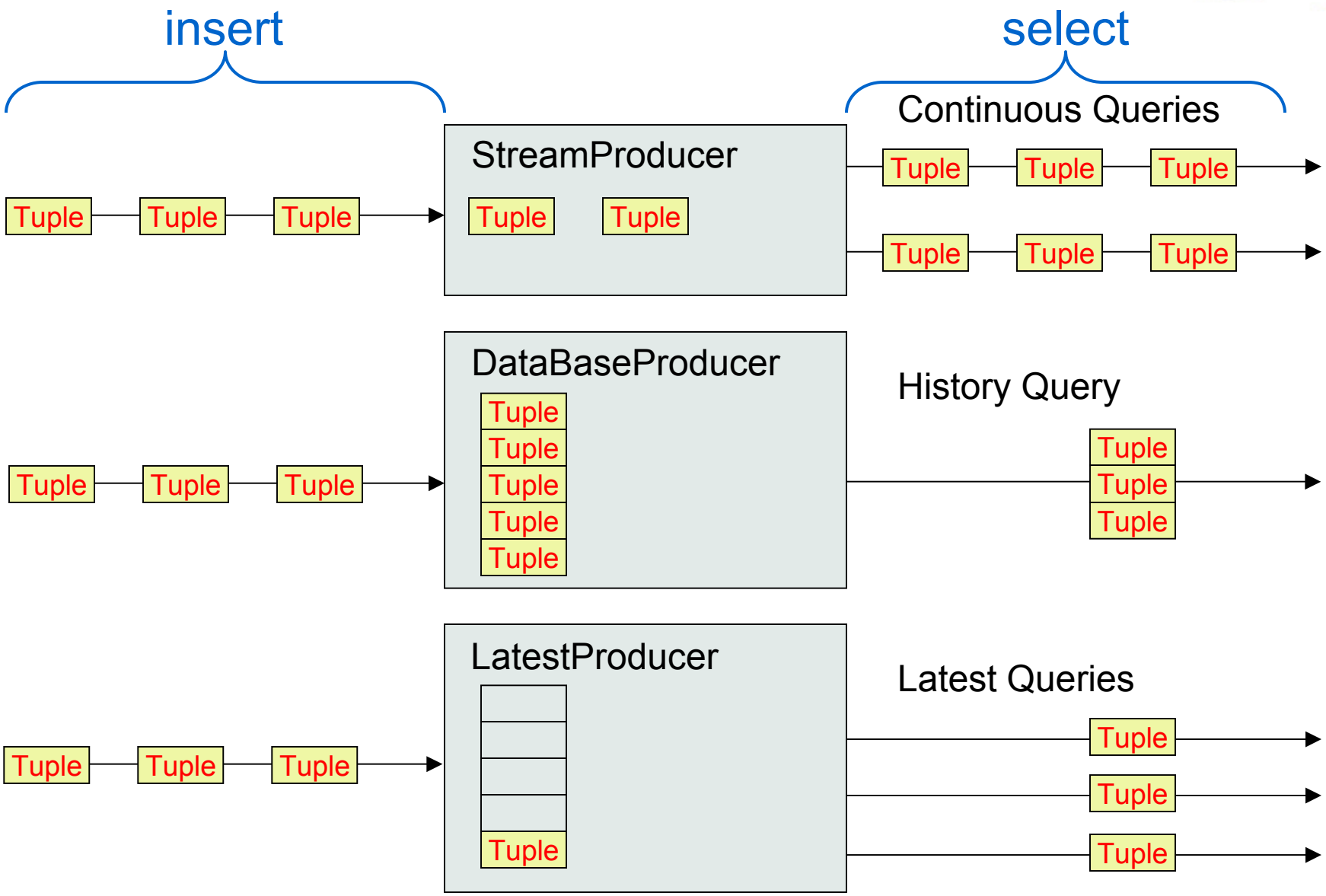
```
SELECT Service.site, ServiceStatus.status,
ServiceStatus.message, Service.URI
FROM Service,ServiceStatus
WHERE Service.URI = ServiceStatus.URI
AND ServiceStatus.status <> 0
AND Service.Type = 'GIN'
```

```
+-----------+--------+----------------+----------------------------+
| site      | status | message        | URI                        |
+-----------+--------+----------------+----------------------------+
| nikhef.nl | 2      | Gin is stopped | http://tbn03.nikhef.nl/GIN |
| nikhef.nl | 2      | Gin is stopped | http://tbn09.nikhef.nl/GIN |
| nikhef.nl | 2      | Gin is stopped | http://tbn16.nikhef.nl/GIN |
+-----------+--------+----------------+----------------------------+
3 Rows in set
```

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 new data matching the query is produced it is streamed to the Consumer
    - Can be seen as an alert mechanism
    - Remember that all tuples carry a time-stamp

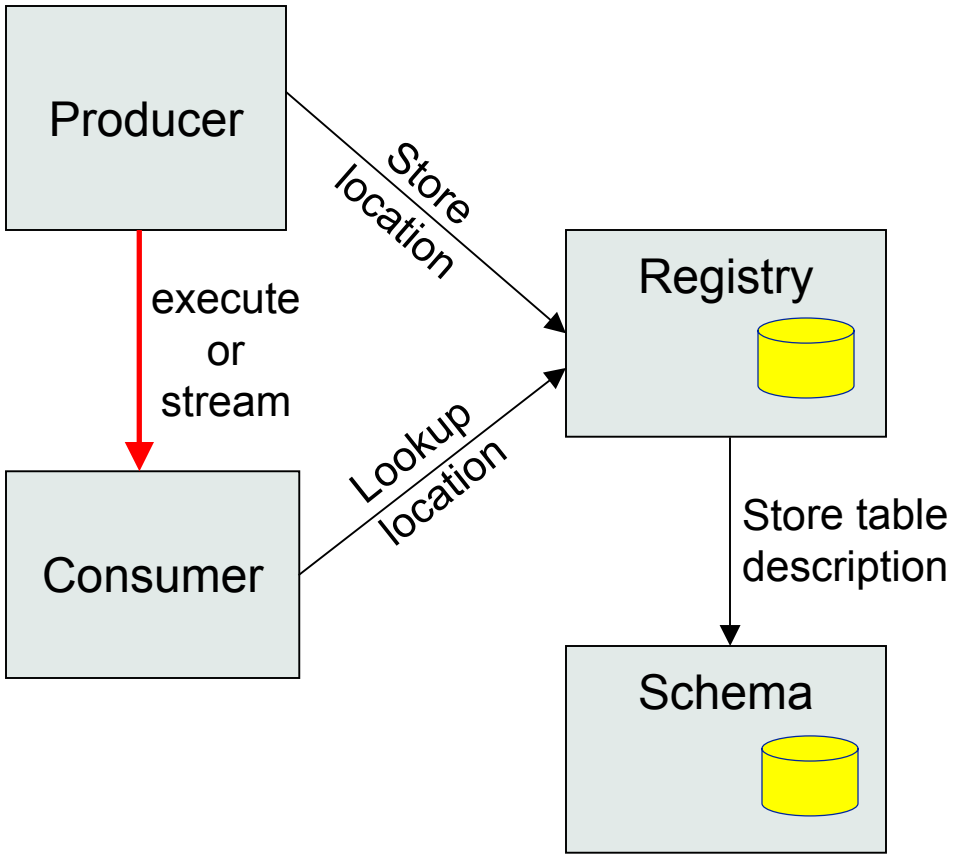
3 Kinds of Query



Producers

- StreamProducer – Supports Continuous Queries
  - In memory data structure
  - Can define minimum retention period
- DataBaseProducer – Supports History Queries
  - Information not lost
  - Supports joins
  - Clean up strategy
- LatestProducer – Supports Latest Queries
  - As DataBaseProducer but
  - Just holds the latest information for any “primaryish” key
- 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 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

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

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

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

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	…	…		

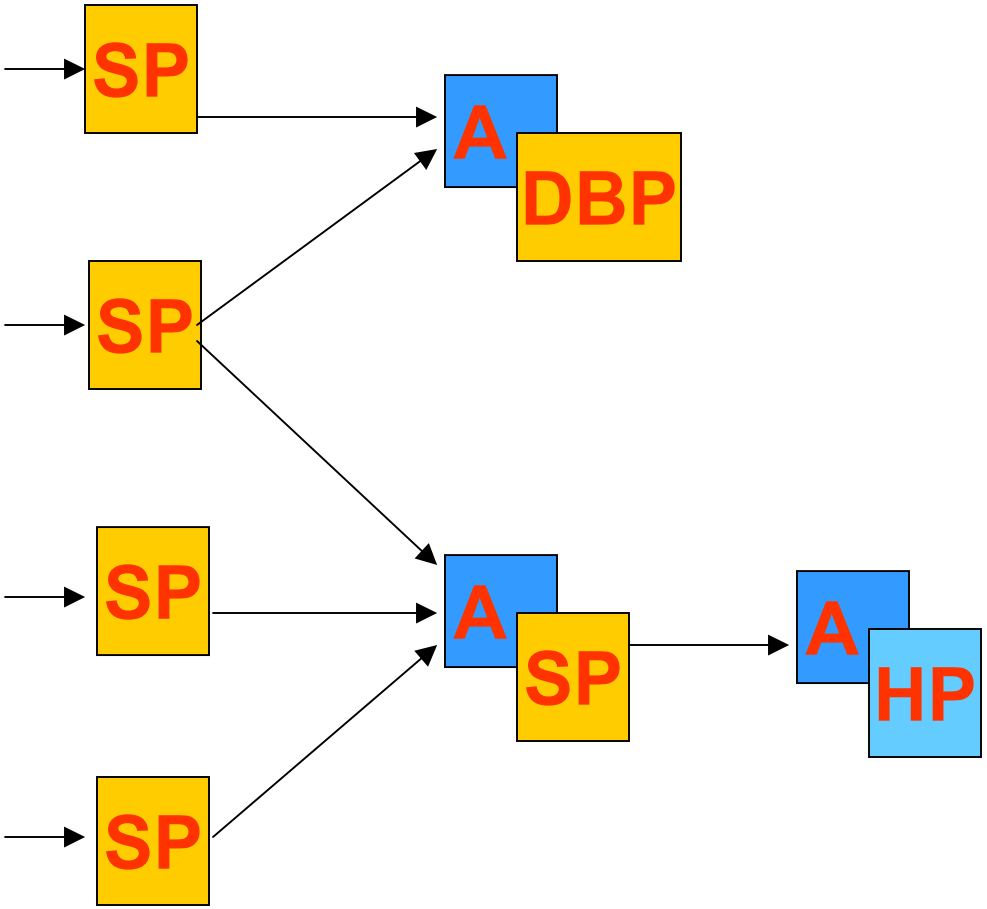
ServiceStatus

URI			up	message
gppse01			y	SE is running
gppse02			n	SE ERROR 101
lxshare0404			y	SE is running

Archiver (Re-publisher)

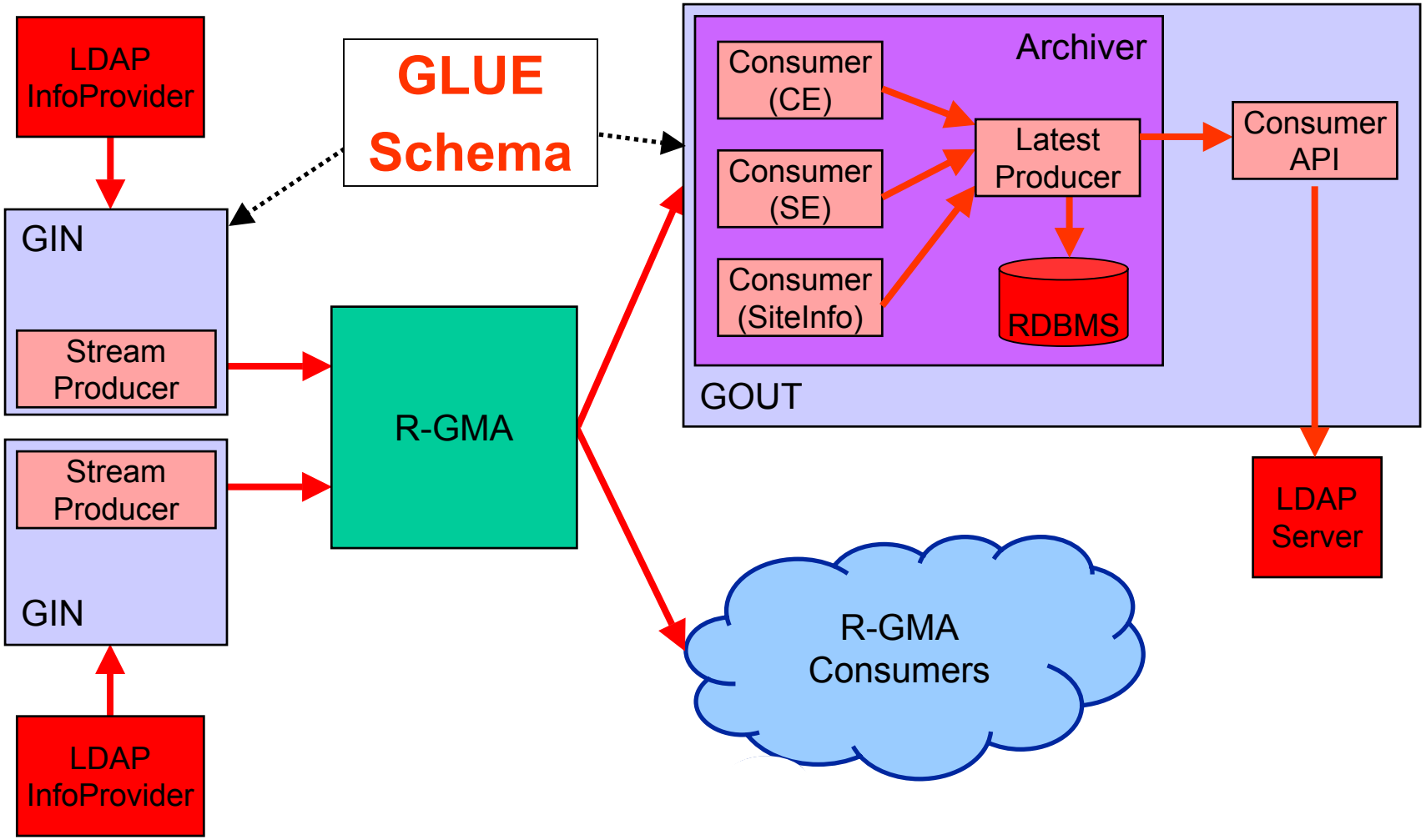
- It is a combined Consumer-Producer
  - Follows the GMA concept but packaged for ease of use
- You just have to tell it what to collect and it does so on your behalf
- Re-publishes to any kind of "Insertable" (i.e. not to the CanonicalProducer)
  - Can support joins if archiving to a DataBaseProducer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

Topologies



- Normally publish via SP
- Archivers instantiated with a Producer and a Predicate
  - Often no predicate
- Must avoid cycles in the graph

GIN and GOUT
(Gadget IN and Gadget OUT)



Ranglia

- R-GMA meets Ganglia
- A CanonicalProducer is used to interface Ganglia
- Allows R-GMA queries to be made to Ganglia
  - Not yet released

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 packaged SQL (edg-rgma-util)
  - e.g. edg-rgma-util contacts:
    - Command: SELECT siteName, sysAdminContact, userSupportContact, siteSecurityContact FROM SiteInfo

R-GMA Browser

All tables
EDG Info Providers
Network Monitoring
CMS

Home
Predefined Queries
Service Status
Site Info
Table Sets

EDG Info Providers

GlueCE
GlueCEAccessControlBaseRule
GlueCESEBind
GlueCluster
GlueHostRemoteFileSystem
GlueSA
GlueSAAccessControlBaseRule
GlueSE
GlueSEAccessProtocol
GlueSEAccessProtocolSupportedSecur
GlueSL
GlueSubCluster
GlueSubClusterSoftwareRunTimeEnvirc
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 ServiceStatus
- show producers of ServiceStatus
- latest select * from ServiceStatus
- old continuous select * from ServiceStatus

edg-rgma – Example

```
$> edg-rgma

rgma> stream declare userTable

rgma> stream minret 0.2

rgma> stream INSERT into userTable (userId, aString,
        aReal, anInt) values ('fisher', 'hello', 3.162, 21)

rgma> timeout 0.3

rgma> old continuous SELECT * from userTable
```

```
+--------+---------+-------+-------+-----------------+-----------------+
| userId | aString | aReal | anInt | MeasurementDate | MeasurementTime |
+--------+---------+-------+-------+-----------------+-----------------+
| fisher | hello   | 3.162 | 21    | 2003-11-11      | 11:06:01        |
+--------+---------+-------+-------+-----------------+-----------------+
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 = edg_rgma_perl::StreamProducer_new();
```

Python
```
myProducer = edg_rgma_python.StreamProducer_new()    or
myProducer = rgma.StreamProducer()
```

Some Times…

- 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 short
    - Don’t set it too long
- RetentionPeriod
  - Period for which the published data will remain available, even after the Producer has been closed
    - Default is 0

C++ Producer - Example

```
#include …
#include "info/StreamProducer.hh"
int main(int argc, char* args[]) {
  if (argc != 2) {
    std::cout << "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++ Consumer - Example

```
#include …
#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(1);
     }
     if(myConsumer.hasAborted()){
        std::printf("Consumer query timed-out\n");
     }
     edg::info::ResultSet resultSet = myConsumer.popIfPossible();
     if (resultSet) {
       std::printf("ResultSet: %s\n", resultSet.toString().c_str());
      }
   } catch (edg::info::RGMAException& e) {
      std::printf("Exception: %s\n", e.what());
    }
}
```

Summary

- R-GMA
  - is suitable for Information **and** Monitoring
  - is a relational implementation of the GGF's GMA
  - has different Producer types
  - mediator creates the impression of a single RDBMS
  - has authentication using grid certificates
  - has been integrated with Ganglia
  - 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/