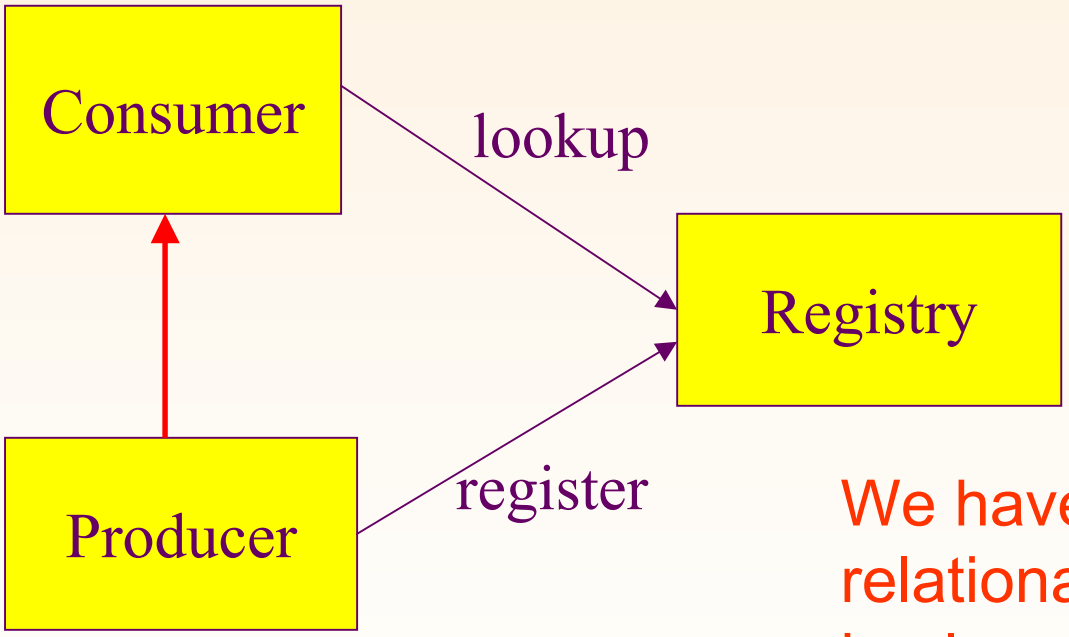
The impact of R-GMA

(upon WP1 and WP4)

Grid Monitoring Architecture (GMA)

- We use it not only for monitoring but also as the basis of an information system



We have chosen a relational implementation

R-GMA

- Not a general distributed RDBMS system, but a way to use the relational model in a distributed environment where ACID properties are not generally important.
- Producers announce: SQL “CREATE TABLE”
  publish: SQL “INSERT”
- Consumers collect: SQL “SELECT”

Table Example

The round trip time from RAL to Lyon as measured pinger was 500 ms on the 9th November at 11:00. :

(RAL, Lyon, Pinger, 1500, 001-11-09T11:00Z)

A set of such tuples could be stored in a table:

NetworkMonitor(NM1, NM2, Tool, Measurement, Time/Date)

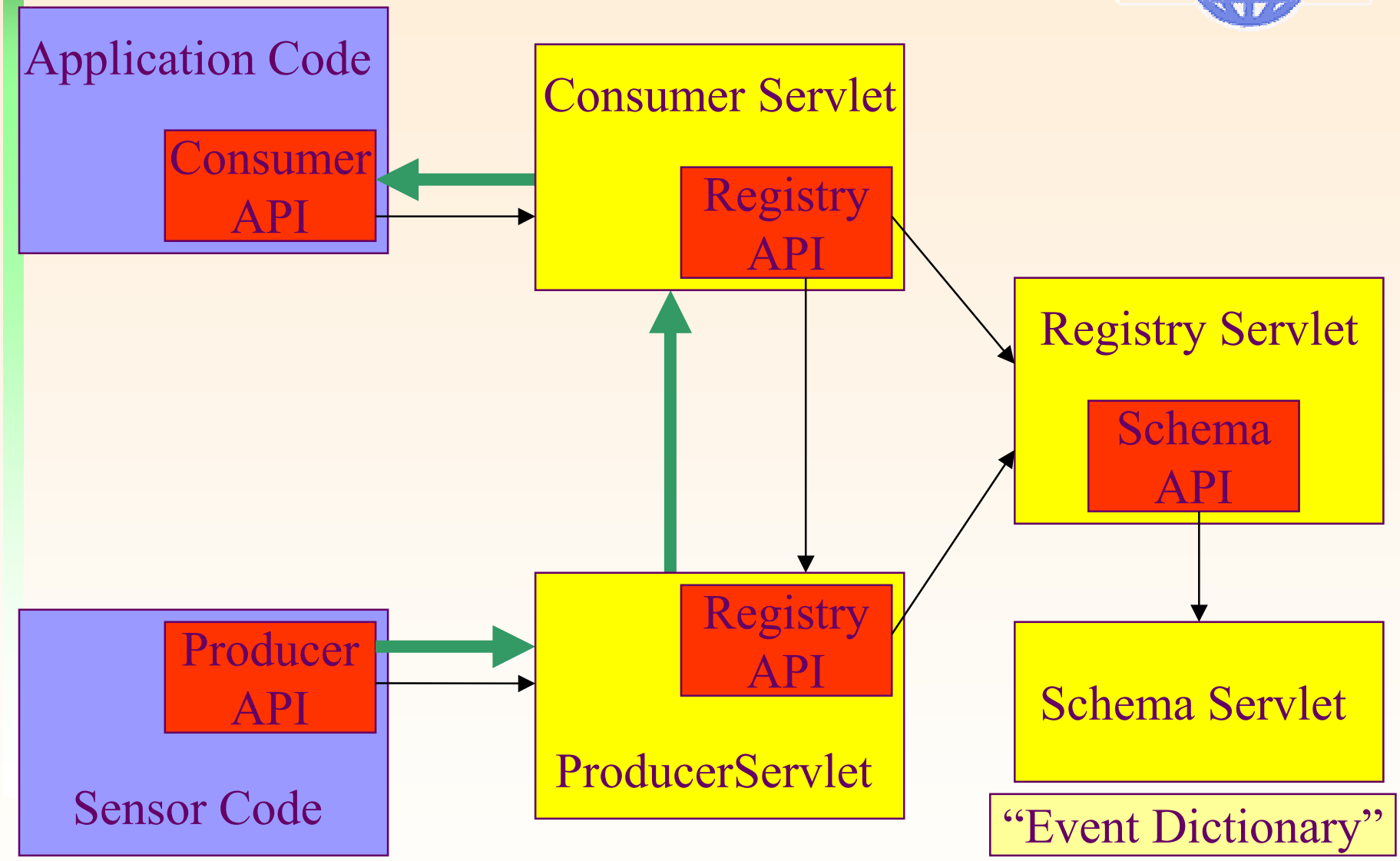
Any structured data can be represented in tables in this manner.

Complex queries can be formulated with SQL.

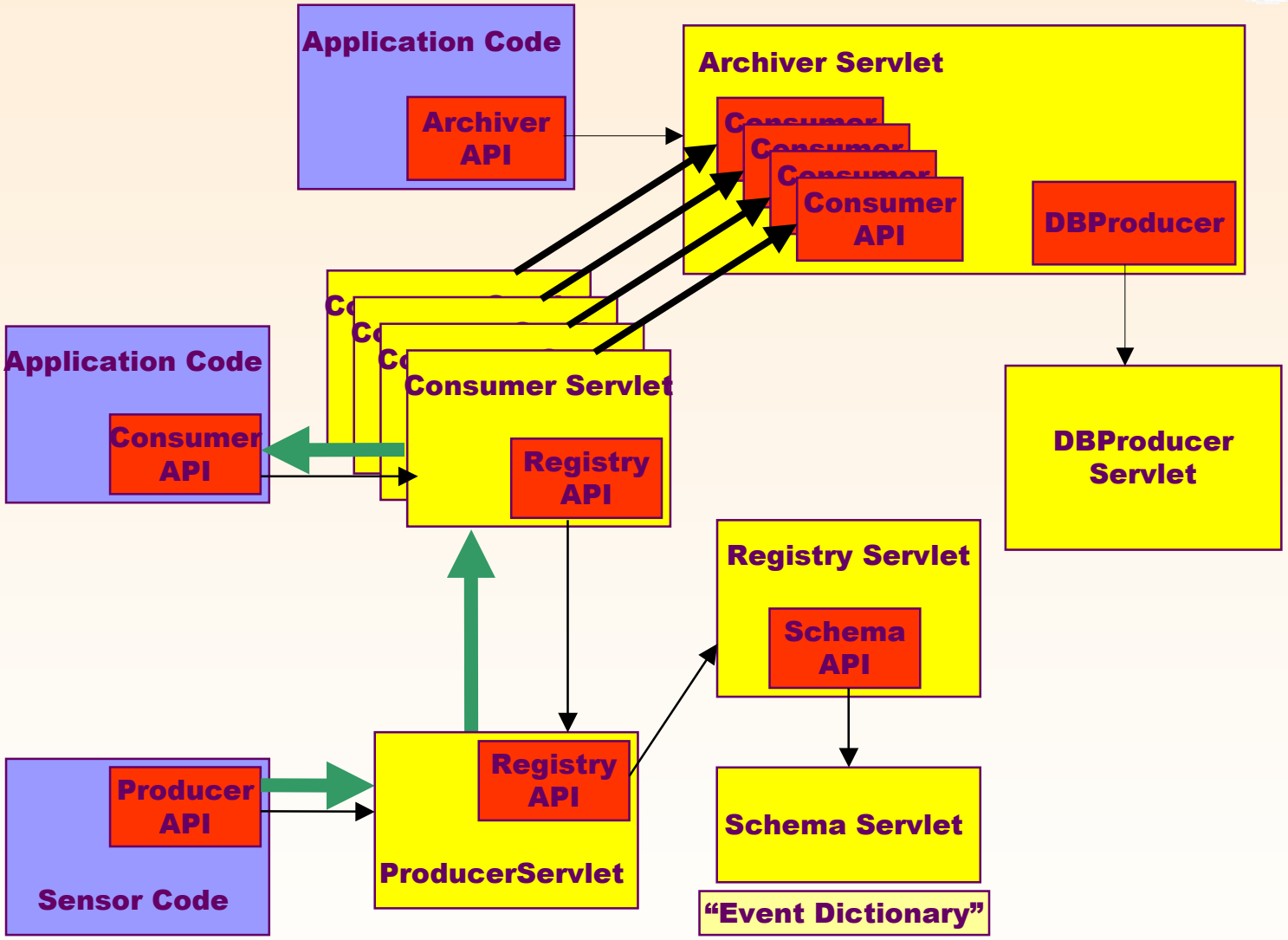
OGSA - Future

- R-GMA is conceptually close to OGSA
- OGSA - replaces resources with services
- MDS was for locating resources
- OGSA has an index for locating services
- MDS does not fit naturally though it will no doubt be kept some time for backwards compatibility
- R-GMA might (eventually) use OGSA index to provide the functionality needed for the Schema/Registry
- In the short term the R-GMA Registry and Schema will instead be OGSA services
- We plan to be quick off the mark with this work

R-GMA Detail


Application Code
Consumer API
Consumer Servlet
Registry API
Registry Servlet
Schema API
Schema Servlet
"Event Dictionary"
Producer API
Sensor Code
Registry API
ProducerServlet

R-GMA Architecture



A Mediator?

- The set of tables producers publish constitute a global relational schema
- Individual producers contribute “views” on this global virtual database
- Views can themselves be expressed in SQL
- Queries against the global schema have to be matched with suitable producers
- This soon becomes complicated due to issues of soundness and completeness of producers

Schema and views I

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

CPULoad (Producer3)				
CH	CERN	ATLAS	1.6	19055611022002
CH	CERN	CDF	0.6	19055511022002

Schema and Views II

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

WHERE Country = ‘UK’ AND Site = ‘RAL’

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

WHERE Country = ‘UK’ AND Site = ‘GLA’

SELECT load FROM CPULoad WHERE Site=‘GLA’

SELECT load FROM CPULoad WHERE Facility=‘CDF’

Example

NetworkMonitor			
NM1	**NM2**	**Tool**	**Measurement**
RAL	Lyon	Pinger	1500
RAL	CERN	Pinger	800

StorageElement		
Name	**NM**	**…**
RALS1	RAL	…
CERNS1	CERN	…
LyonS1	Lyon	…

ComputingElement		
Name	**NM**	**…**
RALQ1	RAL	…
CERNQ1	CERN	…
LyonQ1	Lyon	…

```
SELECT S.Name, C.Name, N.Measurement
FROM NetworkMonitor N, ComputingElement C, StorageElement S
WHERE S.NM = N.NM1 AND C.NM = N.NM2 AND MEASUREMENT < 1000
```

S.Name	C.Name	Measurement
RALS1	CERNQ1	800

```
SELECT S.Name, C.Name
FROM ComputingElement C, StorageElement S
WHERE S.NM = C.NM
```

Today

- Without Mediator use Archiver to assemble information from various tables into one location
- Consumer/Producer Combinations
  - Write Logic to manipulate information
    - eg averages for sites etc.
- Requires some advance knowledge of Producers which exist
  - need a direct match.
- Simple Java and C++ API's
  - insert
  - execute
  - setBufferSize (enables streaming)

Way Forward

- Security (WP2)
- Shift towards Web Services
- Mediator
  - Simple Mediator for producers that register predicates of the form
    - WHERE attr1=val1 AND attr2=val2 …
  - eventually solving more complex queries
- C, Python and Perl APIs
- GRM using R-GMA as transport

Way Forward

- Much improved documentation for
  - Users
  - Developers
  - Installers
- Plans to expand R-GMA beyond Circular Buffer and DataBase producers.
  - Files, existing trace programs e.g. netlogger
  - Working with CrossGrid
- Testing, Testing and more Testing.

Summary

- For TB2 we plan that the main info services will be R-GMA with MDS as a fall back solution if we can't get everything ready on time.
- Assistance to define Schema (Tables)
- Take existing Scripts to populate MDS
  - Cron job
  - Publish to R-GMA
- Future
  - Tailored Sensor using R-GMA Producer API