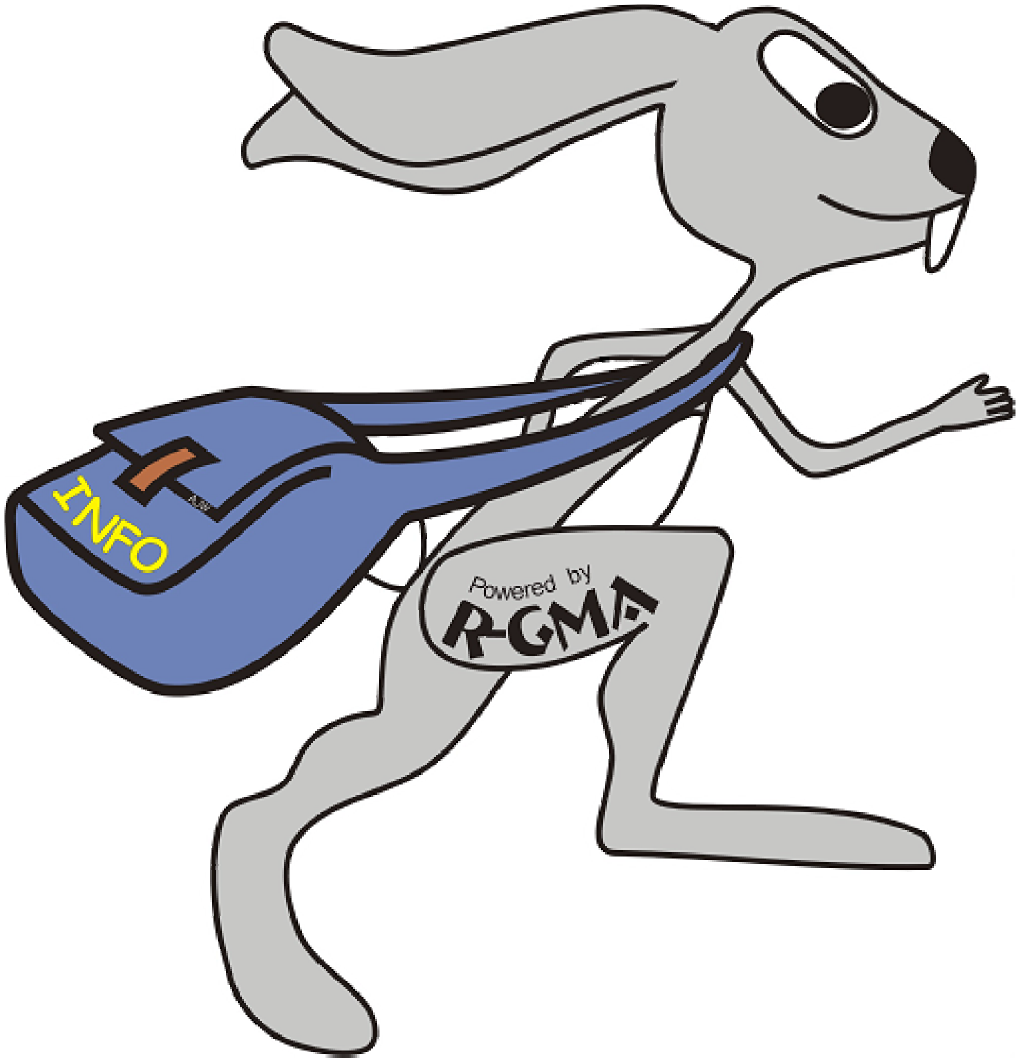
Use of R-GMA in BOSS

Henry Nebrensky (Brunel University)

VRVS 26 April 2004

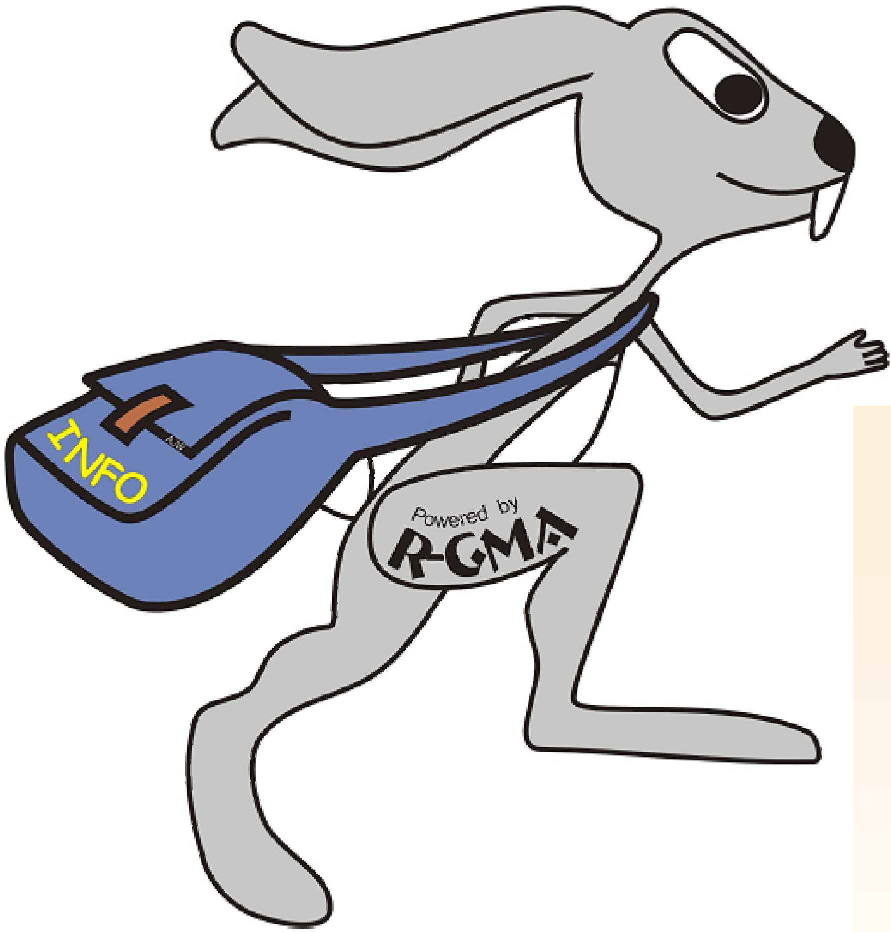
Some slides stolen from various talks at EDG 2nd Review
(http://documents.cern.ch/AGE/current/fullAgenda.php?ida=a021814),
WP3 overview at GridPP middleware mtg. (???),
WP1-WP7, CERN, 18th June 2002
(http://documents.cern.ch/AGE/current/fullAgenda.php?ida=a02943),
and Claudio Grandi's talk at CHEP'03

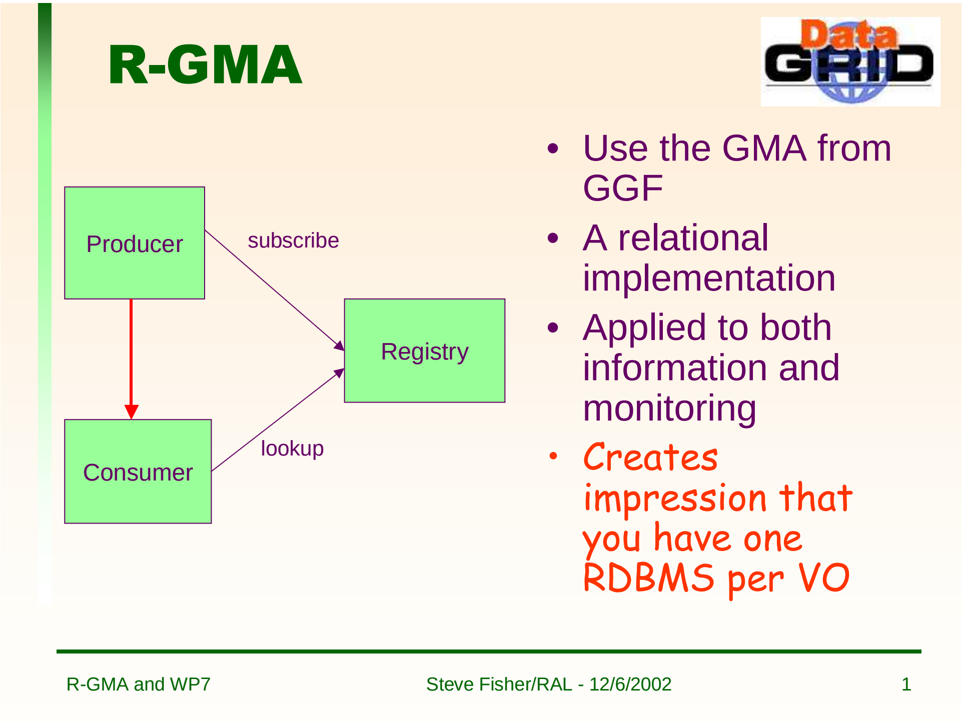


R-GMA

- Grid monitoring infrastructure
- Based on GGF GMA
- Discrete consumers and producers
- Registry acts as matchmaker

R-GMA





R-GMA

Producer

subscribe

Registry

Consumer

lookup

- Use the GMA from GGF
- A relational implementation
- Applied to both information and monitoring
- Creates impression that you have one RDBMS per VO

R-GMA and WP7 Steve Fisher/RAL - 12/6/2002 1

More on R-GMA see e.g.“**RGMA deployment**” at `http://www.gridpp.ac.uk/gridpp7/gridpp7_fisher.ppt`

Basic BOSS components

`boss` executable:

the BOSS interface to the user

MySQL database:

where BOSS stores job information

`jobExecutor` executable:

the BOSS wrapper around the user job

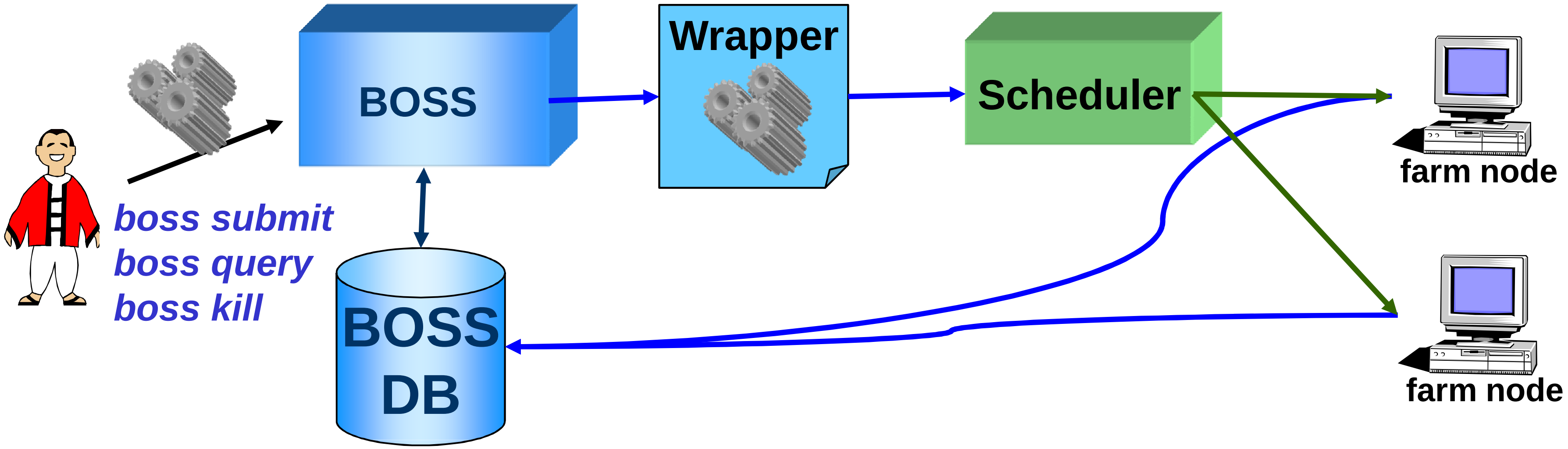
`dbUpdator` executable:

the process that writes to the database while the job is running

Local scheduler

may be a “Grid” scheduler

Basic flow



Accepts job submission from users

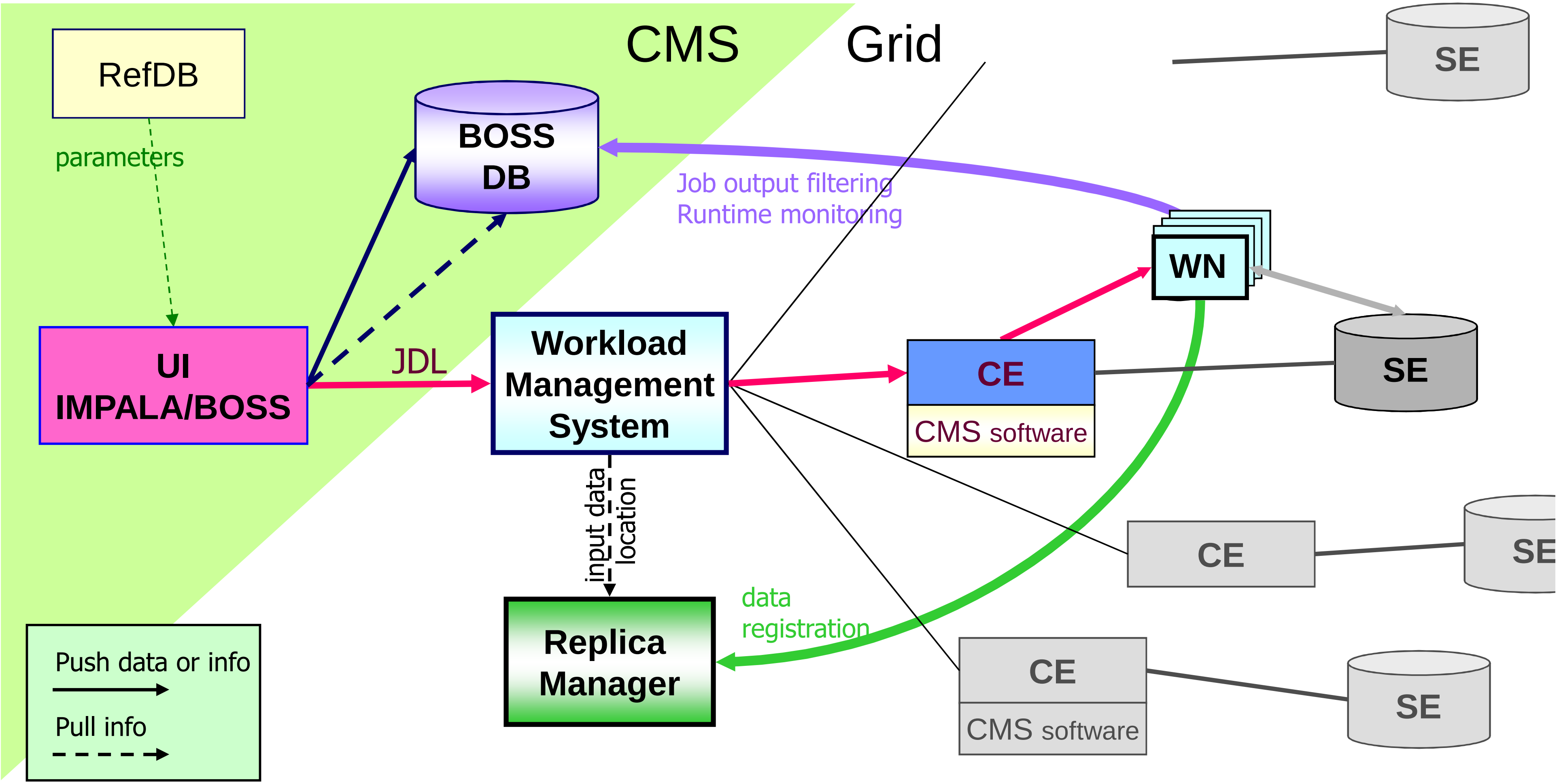
Stores info about job in a DB

Builds a wrapper around the job (*jobExecutor*)

Sends the wrapper to the local scheduler

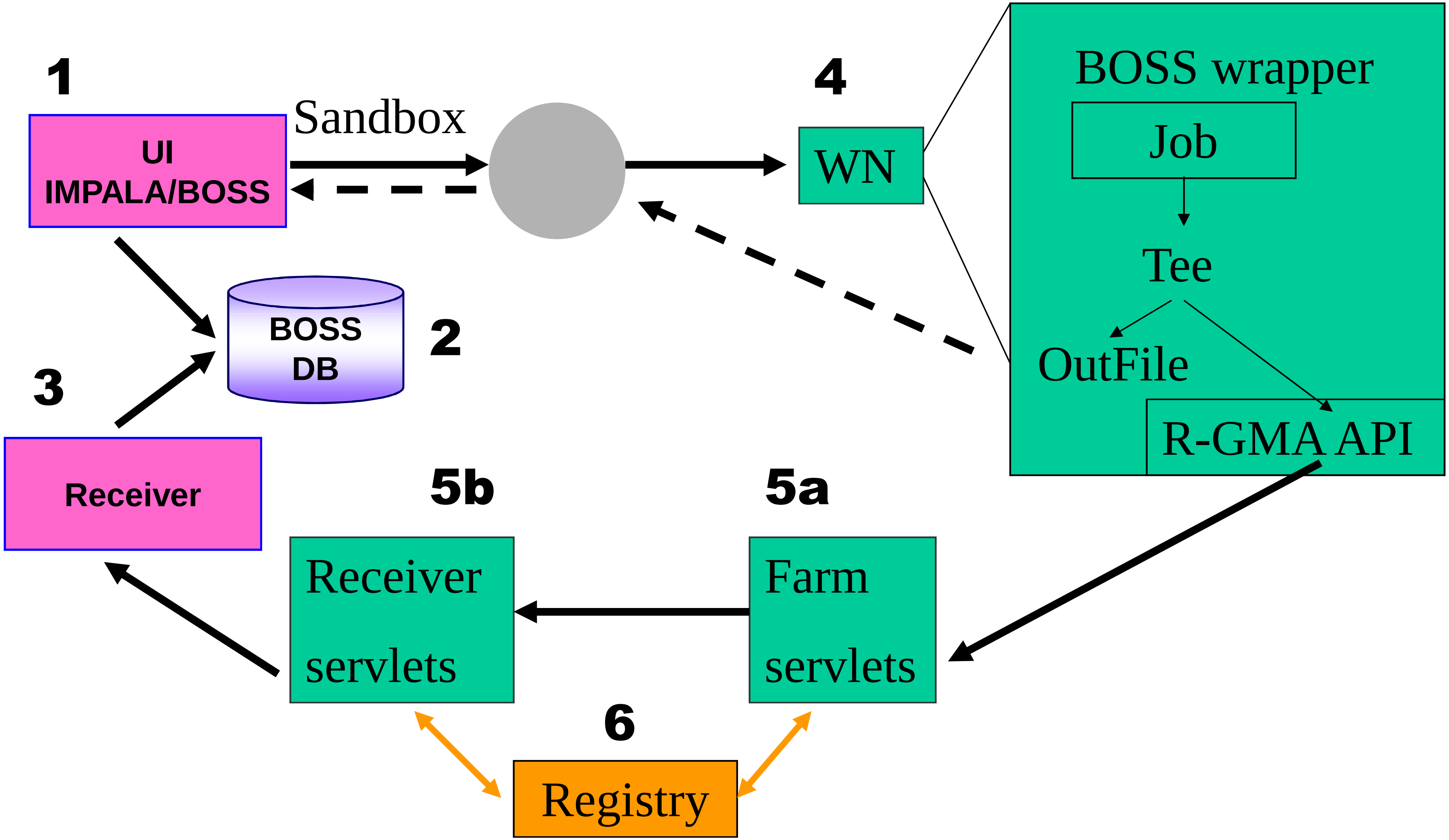
The wrapper sends to the DB info about the job

BOSS



CMS
Grid
RefDB
parameters
BOSS DB
UI IMPALA/BOSS
JDL
Workload Management System
Job output filtering
Runtime monitoring
WN
CE
CMS software
SE
SE
CE
SE
CE
CMS software
SE
input data location
Replica Manager
data registration
Push data or info
Pull info

Use of R-GMA in BOSS



Use of R-GMA in BOSS

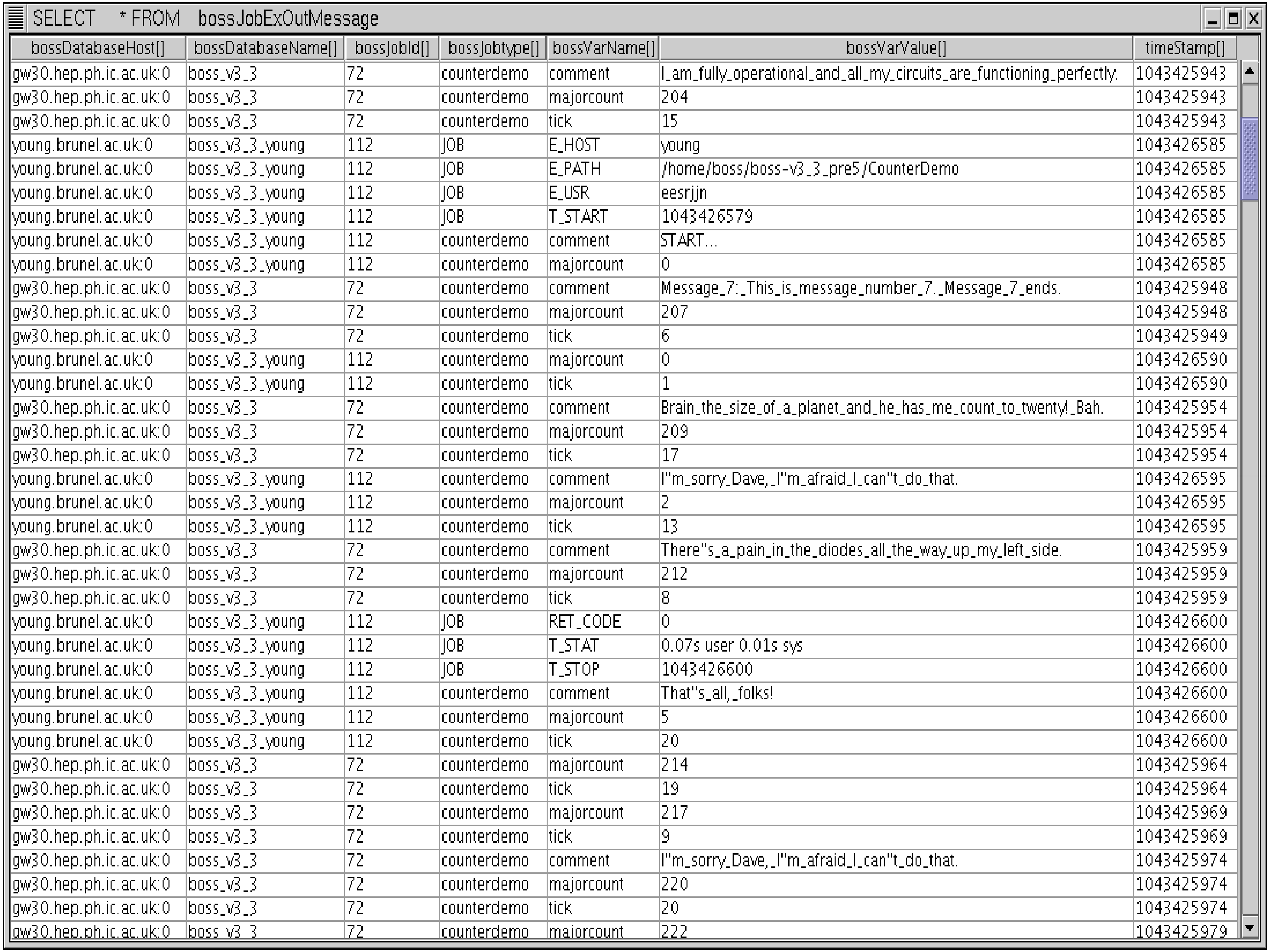
- Publish each update into R-GMA as a separate message – separate row
- Each producer gives host and name of “home” BOSS DB, and jobId; this identifies it uniquely
- Receiver looks for all rows relating to its DB; uses jobId and jobType to do SQL **`UPDATE`**

Use of R-GMA in BOSS

The screenshot below shows the streamed output messages from a Brunel job (ID 112) being sent through R-GMA and displayed using the EDG Pulse tool from WP3. As Pulse can monitor multiple producers, it also shows the output from a longer job already running at Imperial (ID 72).

The receivers that update the BOSS databases use the bossDatabaseHost and bossDatabaseName fields to select only the relevant messages, so that the database at each institute is updated with only the information about its own jobs.

http://www.brunel.ac.uk/~eestprh/GRIDPP/Index.htm



SELECT * FROM bossJobExOutMessage

bossDatabaseHost[]	bossDatabaseName[]	bossJobId[]	bossJobtype[]	bossVarName[]	bossVarValue[]	timeStamp[]
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	comment	I_am_fully_operational_and_all_my_circuits_are_functioning_perfectly.	1043425943
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	204	1043425943
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	tick	15	1043425943
young.brunel.ac.uk:0	boss_v3_3_young	112	JOB	E_HOST	young	1043426585
young.brunel.ac.uk:0	boss_v3_3_young	112	JOB	E_PATH	/home/boss/boss-v3_3_pre5/CounterDemo	1043426585
young.brunel.ac.uk:0	boss_v3_3_young	112	JOB	E_USR	eesrjjn	1043426585
young.brunel.ac.uk:0	boss_v3_3_young	112	JOB	T_START	1043426579	1043426585
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	comment	START...	1043426585
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	majorcount	0	1043426585
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	comment	Message_7:_This_is_message_number_7._Message_7_ends.	1043425948
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	207	1043425948
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	tick	6	1043425949
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	majorcount	0	1043426590
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	tick	1	1043426590
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	comment	Brain_the_size_of_a_planet_and_he_has_me_count_to_twenty!_Bah.	1043425954
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	209	1043425954
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	tick	17	1043425954
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	comment	I"m_sorry_Dave,_I"m_afraid_I_can"t_do_that.	1043426595
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	majorcount	2	1043426595
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	tick	13	1043426595
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	comment	There"s_a_pain_in_the_diodes_all_the_way_up_my_left_side.	1043425959
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	212	1043425959
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	tick	8	1043425959
young.brunel.ac.uk:0	boss_v3_3_young	112	JOB	RET_CODE	0	1043426600
young.brunel.ac.uk:0	boss_v3_3_young	112	JOB	T_STAT	0.07s user 0.01s sys	1043426600
young.brunel.ac.uk:0	boss_v3_3_young	112	JOB	T_STOP	1043426600	1043426600
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	comment	That"s_all,_folks!	1043426600
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	majorcount	5	1043426600
young.brunel.ac.uk:0	boss_v3_3_young	112	counterdemo	tick	20	1043426600
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	214	1043425964
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	tick	19	1043425964
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	217	1043425969
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	tick	9	1043425969
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	comment	I"m_sorry_Dave,_I"m_afraid_I_can"t_do_that.	1043425974
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	220	1043425974
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	tick	20	1043425974
gw30.hep.ph.ic.ac.uk:0	boss_v3_3	72	counterdemo	majorcount	222	1043425979

Use of R-GMA in BOSS (1)

- R-GMA smoothes “firewall” issues
- Consumer can watch many producers; producers can feed multiple consumers.
- Provides uniform access to range of monitoring data (WP7 network, etc.)
- Doesn’t depend on other EDG components

Use of R-GMA in BOSS (2)

- BOSS job wrapper uses an R-GMA StreamProducer and C++ API
  - Can define minimum retention period
  - No guarantees
- BOSS receiver implemented in Java

Scalability Tests With CMS, Boss and R-GMA

Stolen from Rob Byrom’s slides at

http://agenda.cern.ch/fullAgenda.php?ida=a036755

(Presented at 2003 IEEE/NSS mtg, sub. to Trans. Nuc. Sci.)

Test Motivation

- Want to ensure R-GMA can cope with volume of expected traffic and is scalable.
- CMS production load estimated at around 2000 jobs.
- Initial tests with v3-3-28 only managed about 400 - could do better L .

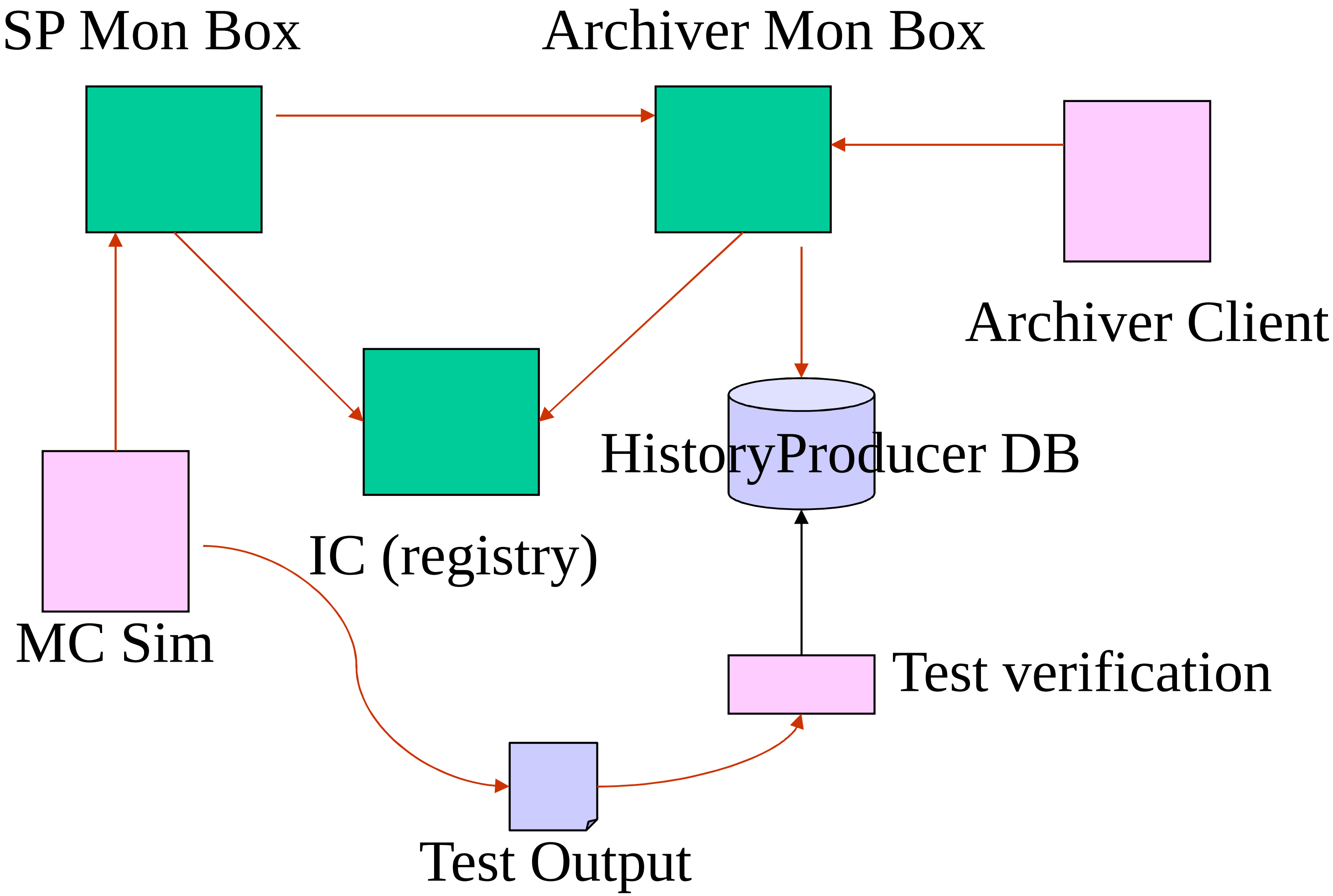
Test Design

- A simulation of the CMS production system was created.
  - A Java MC simulation was designed to represent a typical job.
  - Each job creates a stream producer.
  - Each job publishes a number of tuples depending on the job phase.
  - Each job contains 3 phases with varying time delays.
  - Emulates "CMSIM" message publishing pattern, but so far with 10 hour run time compressed into a minute …
  - … so actually have fewer simultaneous jobs than real case, but overall a much higher rate of message production.

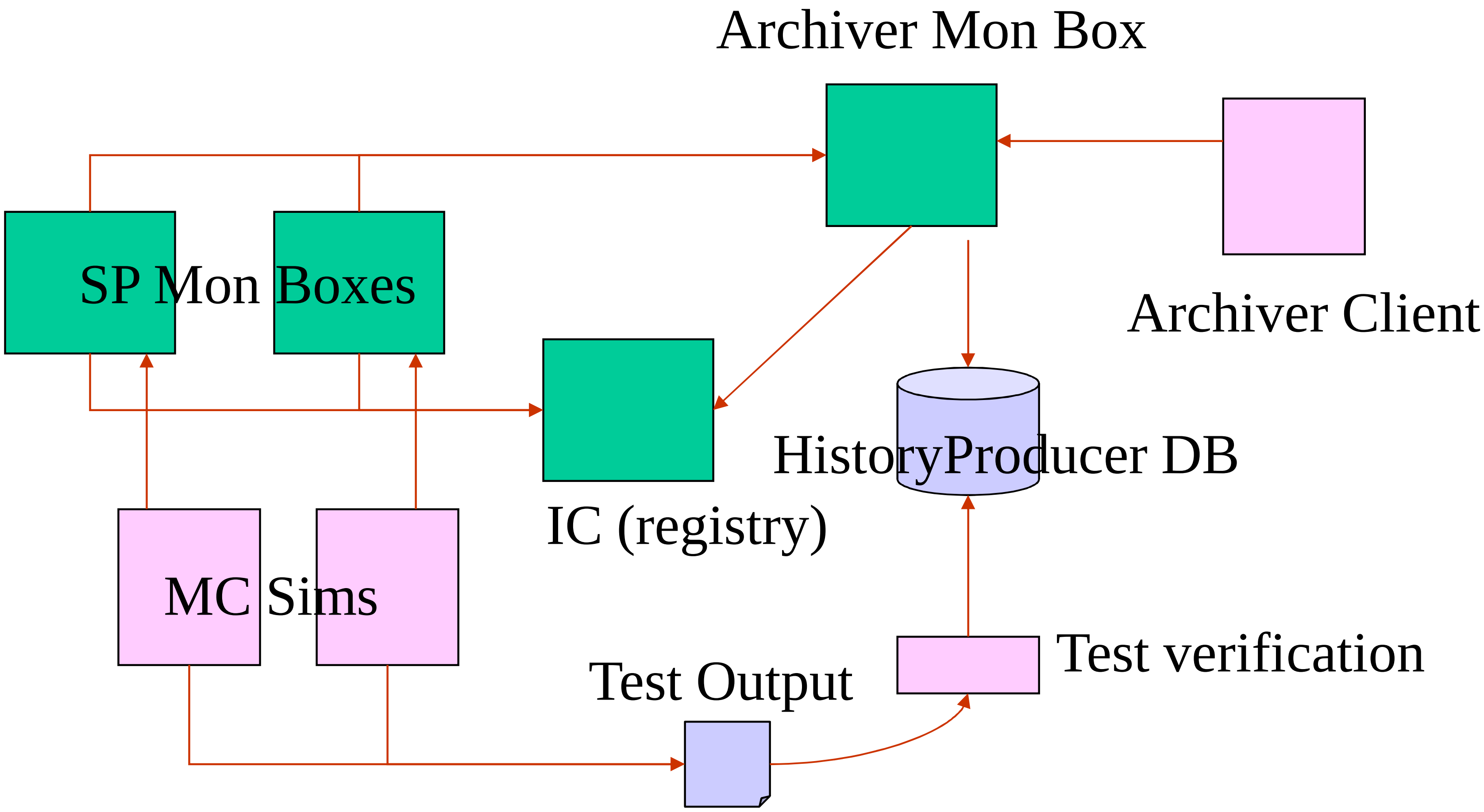
Test Design

- An Archiver scoops up published tuples.
  - The Archiver db used is a representation of the BOSS db, but stores history of received messages, rather than just a cumulative update.
  - Archived tuples are compared with published tuples to verify the test outcome.

Topology


SP Mon Box
Archiver Mon Box
Archiver Client
HistoryProducer DB
IC (registry)
MC Sim
Test verification
Test Output

Topology


Archiver Mon Box
SP Mon Boxes
Archiver Client
IC (registry)
HistoryProducer DB
MC Sims
Test Output
Test verification

Test Setup

- Archiver & SP mon box setup at Imperial.
- SP mon box & IC setup at Brunel.
- Archiver and MC sim clients positioned at various nodes within both sites.
- Tried 1 MC sim and Archiver with variable Job submissions.
- Also setup similar test on WP3 testbed using 2 MC sims and 1 Archiver.

Results

- 1 MC sim creating 2000 jobs and publishing 7600 tuples proven to work without glitch (R-GMA v3.4.13)

- Demonstrated 2 MC sims each running 4000 jobs (with 15200 published tuples) on the WP3 testbed.

Pitfalls Encountered

- Lots of fun integration problems.
  - Firewall access between imperial and Brunel initially blocked for streaming data (port 8088).
  - Limitation on number of open streaming sockets – 1K.
  - Discovered lots of OutOfMemoryErrors.
  - Various configurations problems at both imperial and Brunel sites.
  - Caused many test runs to fail.
- Probably explained poor initial performance.

Weaknesses

- Test is time consuming to set-up.
  - MC and Archiver are are manually started.
  - Analysis bash script takes forever to complete.
- Requires continual attention while running.
  - To ensure the MC simulation is working correctly.
  - Test takes considerable time for large number of jobs (ie > 1000).
- Need to run multiple MC sims.
  - To generate more realistic load.
  - How to we account for ‘other’ R-GMA traffic?

Improving the Design

- Need to automate!
  - Improvements made so far.
    - CMS test comes as part of the ‘performance’ R-GMA RPM.
    - MC simulation can be configured to occur repeatedly with random job ids.
    - Rather than monitoring the STDOUT from the MC Sim, test results are published to an R-GMA table.
    - Analysis script now implemented in java; Verification of data now takes seconds rather than hours!
    - CMS test can now be used as part of the WP3 testing framework and is ready for Nagios integration.

Measuring Performance

- Need a nifty way of measuring performance!
    - Things to measure.
        - Average time taken for a tuple published via the MC simulation to arrive at the Archiver.
        - The tuple throughput of the CMS test (eg how do we accurately measure the rate at which tuples are archived).
        - Would be nice to measure the number of jobs each mon box can carry before hitting memory limits.
    - Need to define a hardware spec that satisfies a level of performance.

Summary

- After initial configuration problems tests were successful - J J .
- But scalability of test is largely dependent on the specs of the Stream Producer/Archiver Mon box.
- Possible to keep increasing number of submitted Jobs but will eventually hit an upper memory limit.
- Need more accurate ways to record performance particularly for data and time related measurements.
- Need to pin down exact performance requirements!