MSG - A messaging system for efficient and scalable grid monitoring

James Casey,Operations Automation Team

EGEE User Forum, Catania

www.eu-egee.org

Information Society and Media

Problem statement

- **Grid monitoring should allow users to see various views of the grid infrastructure**
  - Site Administrators
  - Project level managers
  - VO users
- **All users should be able to base their views of a consistent set of information**
  - Gather the information once
  - Pass it to many consumers

Technical Issues

- **The infrastructure is :**
  - Heterogeneous
  - Highly distributed
  - Spanning multiple administrative domains
  - Consisting of unreliable services, sites and network links

- **You don’t know a priori who might be consuming your information**
  - The most interesting results usually come via aggregation of various information sources

Required Features…

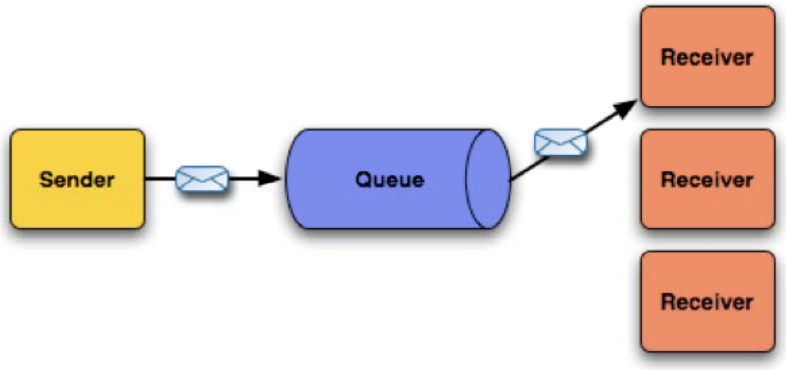
- **We need underlying middleware which handles these issues**
  - Loosely coupled
  - Supports broadcast/multicast
  - Works with clients in multiple languages
  - Resilient to failure of independent components
  - Scales to grid size

Messaging systems as a solution ...

- **Flexible architecture:**
  - Deliver messages using different models
  - Support synchronous or asynchronous communication
  - Durable or transient messaging
  - All communication mediated by a broker
- **Highly Scalable:**
  - Resilient ‘backbone’ of brokers can be created to give scalability and protection against network issues
  - Individual brokers can be failover clusters for reliability
- **Reliability guaranteed to clients by the middleware**
  - Information producers can work in a ‘fire and forget’ mode

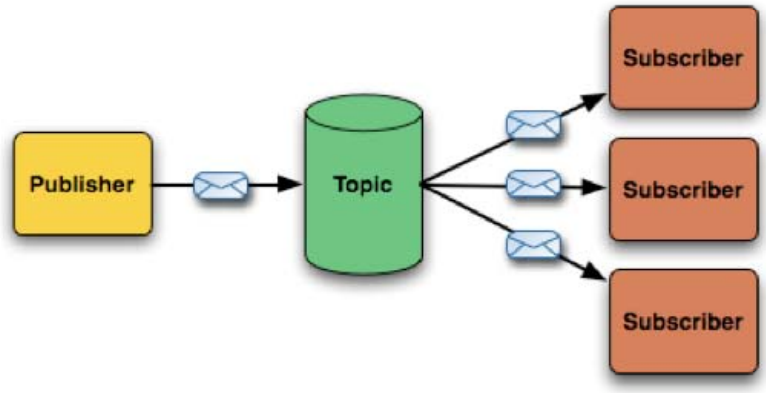
Models of messaging

Point-2-point



- Analogy – **person-to-person** mailing
- One consumer per queue
- Once and only once delivery
- Broker retains messages until consumed

Publish-Subscribe



- Analogy – **Mailing** List
- Producers and consumers are not aware of each other
- All messages delivered to all subscribers

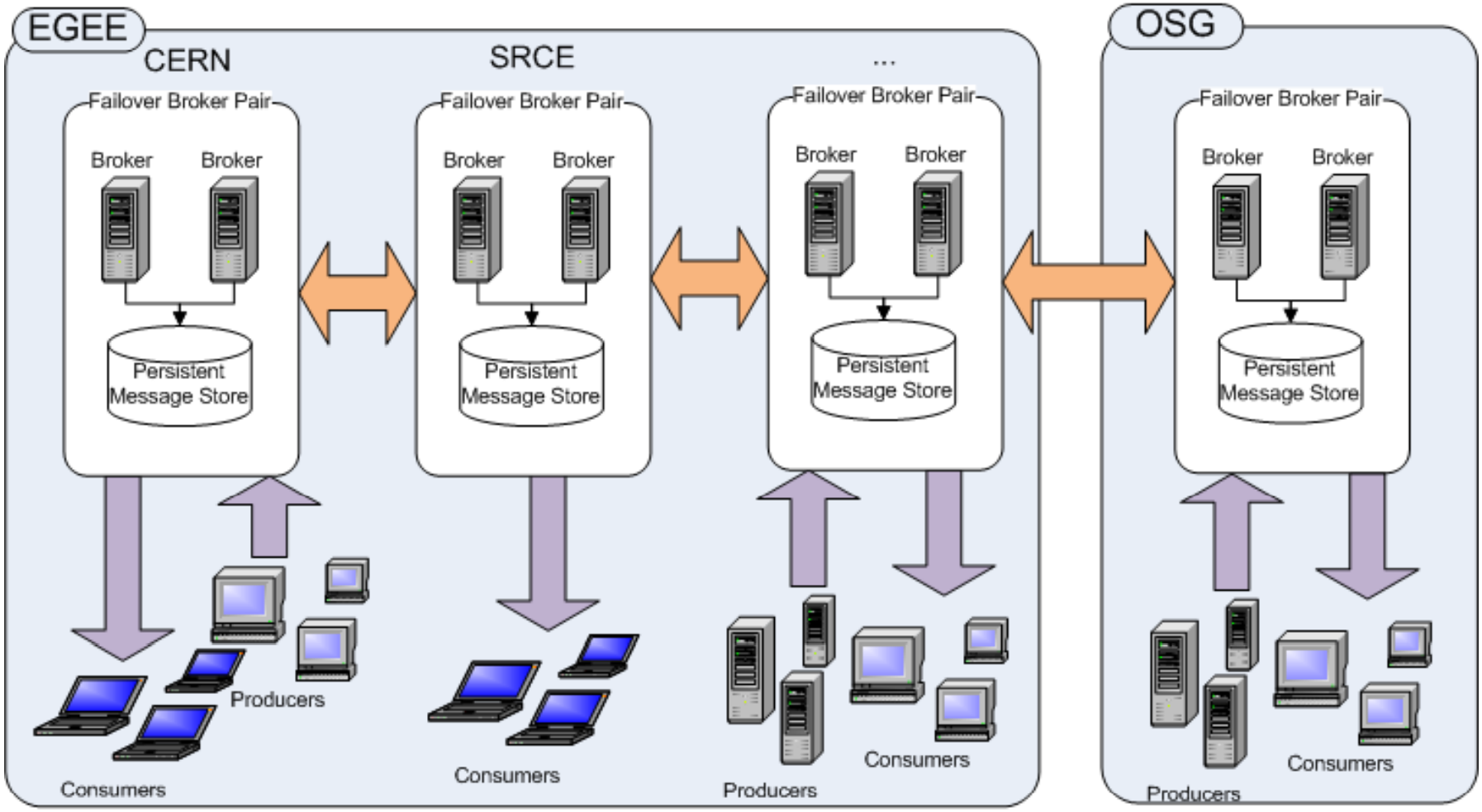
Apache ActiveMQ

- **Mature open-source implementation of these ideas**
  - Top-level Apache project
  - Commercial support available from Progess Software
  - Widely-used commodity software
- **Easy to integrate into your code**
  - Multiple language + transport protocol support
- **Good performance characteristics**
  - 1K – 20K messages per second depending on features used

- **MSG – Messaging System for Grids**
  - Integration of ActiveMQ into our environment
    - RPMs, YAIM configuration, monitoring and alarms
  - Production grade messaging backbone deployed by SA1

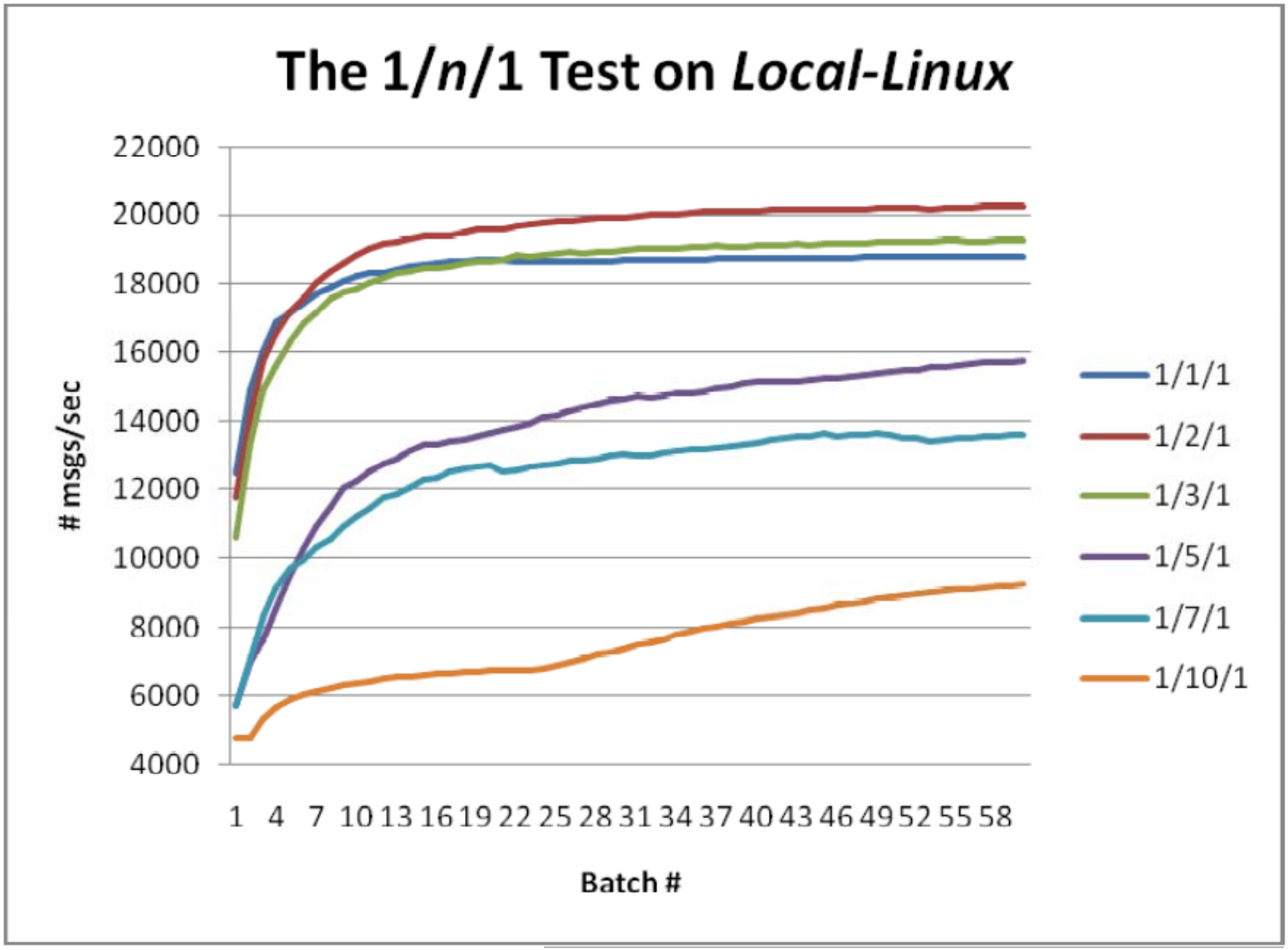
Network of Brokers

Cross-grid messaging backbone consisting of reliable broker clusters



Version 1.0, 19th November, 2008

Vendor tests



1 producer/n consumers/1 broker

From “Optimizing FUSE Message Broker” – http://open.iona.com/resources/collateral/#whitepapers

Talking to a broker - STOMP

Streaming Text Oriented Messaging Protocol

- **Simple, interoperable wire format**
- **Text based, similar in philosophy to HTTP**
- **Multiple Languages**
  - C, C++, Java, C#, Perl, PHP, Python, Ruby, Flash, …
- **Used for producers and consumers**

You can write your first client in 30 minutes – guaranteed !

http://stomp.codehaus.org/

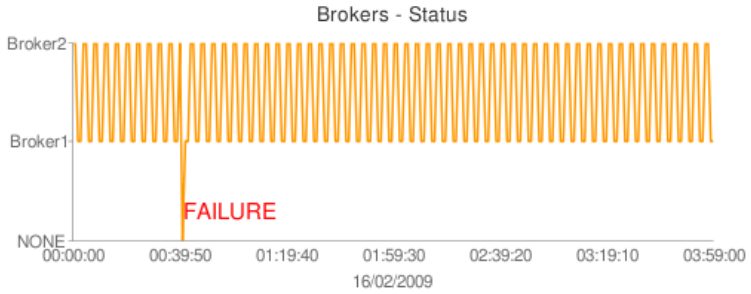
An example…

```
use Net::Stomp ;
my $stomp = Net::Stomp->new ({hostname => 'prod-grid-msg.cern.ch',
                              port => '6163' }) ;
$stomp->connect () ;
$stomp->subscribe ( {
                'destination' => '/topic/grid.probe.results.MySite',
                'ack'         => 'client',
                'activemq.prefetchSize' => 1});
while($should_receive) {
    my $frame = $stomp->receive_frame;
    print $frame->as_string ;
    $stomp->ack({frame => $frame} ) ;
}
$stomp->disconnect ;
```

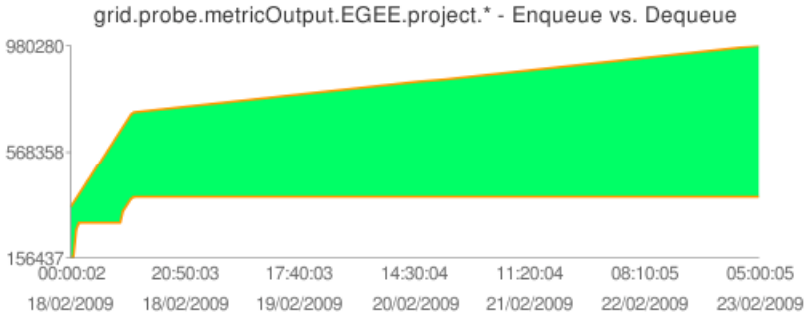
System Management & Monitoring

NAGIOS Application-Level Monitoring

265	Topics Project level Nagios
128	Topics Nagios ROC
126	Topics Nagios sites
570	Queues Nagios sites
4	Durable susbcriptions



List of Topics (Project Level Nagios)



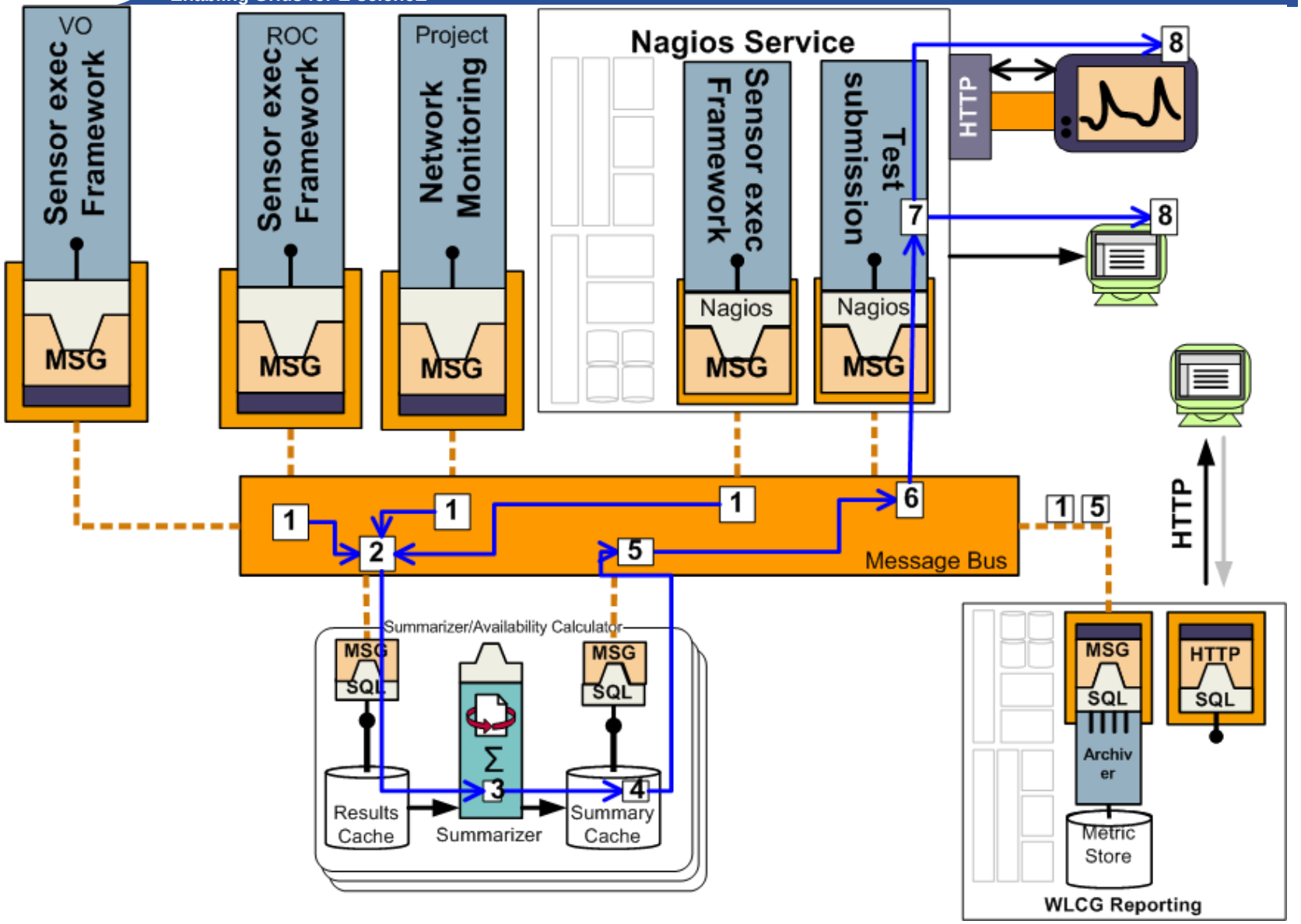
- grid.probe.metricOutput.EGEE.project.AEGIS01-PHY-SCL
- grid.probe.metricOutput.EGEE.project.AEGIS07-PHY-ATLAS
- grid.probe.metricOutput.EGEE.project.ALBERTA-LCG2
- grid.probe.metricOutput.EGEE.project.AUVERGRID
- grid.probe.metricOutput.EGEE.project.Australia-ATLAS
- grid.probe.metricOutput.EGEE.project.BEIJING-LCG2
- grid.probe.metricOutput.EGEE.project.BEgrid-KULeuven
- grid.probe.metricOutput.EGEE.project.BEgrid-UGent
- grid.probe.metricOutput.EGEE.project.BEgrid-ULB-VUB
- grid.probe.metricOutput.EGEE.project.BG-INRNE
- grid.probe.metricOutput.EGEE.project.BG01-IPP

Example screen shots from system management application

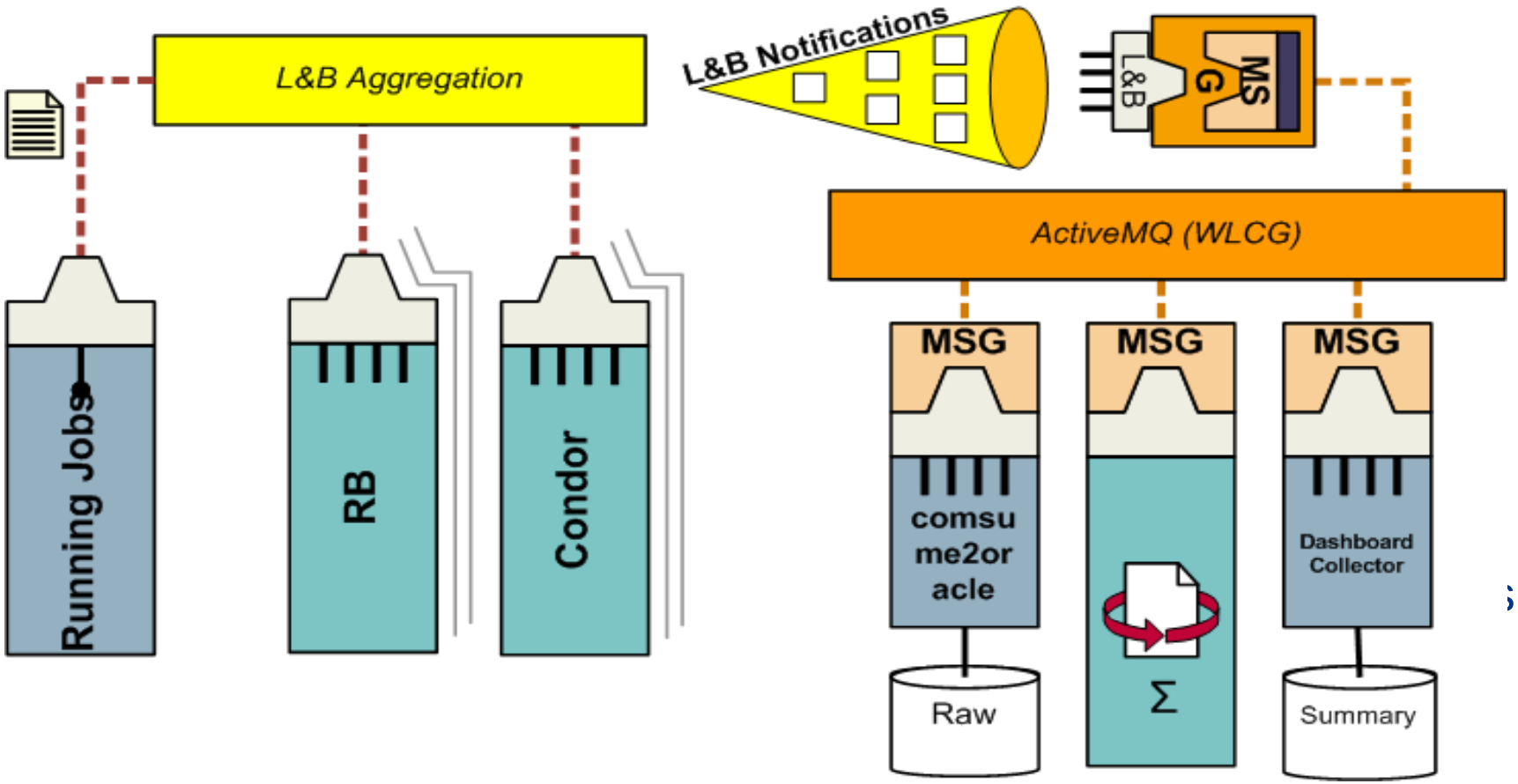
Typical applications of messaging

- **Many applications in grid monitoring follow a simplar pattern of interactions:**
  - Gather results at many points
  - Collect the raw results and store in a database
  - Perform some operation on the raw results
    - Summarisation, availability calculation, …
  - Publish the summarised results to many clients
    - E.g. site monitoring, dashboards, …
  - Store historical data in a database and visualize via web client
- **We provide ‘standard’ components to make this plug and play for many workflows**

VO, ROC, Project & Local monitoring



Another application - Usage Reporting



- **Other integration examples include**
  - Accounting
  - GOCDB synchronization

Currently available data feeds

- **Testing results from sites (a.la. SAM/Nagios)**
- **Job state changes**
  - L&B v2.0
- **Gridftp transfers (a.la. Gridview)**
- **Downtime publications**
  - Only OSG currently
- **Worker node configuration**
  - Software versions, hardware configurations,
  - http://gridops.cern.ch/gcm/tests/

- **All feeds will be documented at**

  https://twiki.cern.ch/twiki/bin/view/EGEE/MsgNamespace

Future and Summary

- **MSG is the solution used for the new monitoring architecture**
  - E.g Talks by John Shade, Emir Imamagic earlier
- **Adopted as standard monitoring communication system for EGEE**
  - Replacing R-GMA
- **We will start working with VO communities to architecture their MSG-based solution**
  - Some work started with ATLAS/Ganga already
- **More infrastructure monitoring data will be integrated in the coming months**

For more information …

- **MSG**
  - https://twiki.cern.ch/twiki/bin/view/EGEE/MsgArchitecture
  - https://twiki.cern.ch/twiki/bin/view/EGEE/MsgNamespace
- **Software used:**
  - http://activemq.apache.org
  - http://stomp.codehaus.org
- **Wikipedia links:**
  - http://en.wikipedia.org/wiki/Enterprise_messaging_system
  - http://en.wikipedia.org/wiki/Message_Oriented_Middleware
  - http://en.wikipedia.org/wiki/Publish/subscribe
  - http://en.wikipedia.org/wiki/Message_queue
- **Patterns for messaging systems →→→**
  - http://www.enterpriseintegrationpatterns.com/index.html