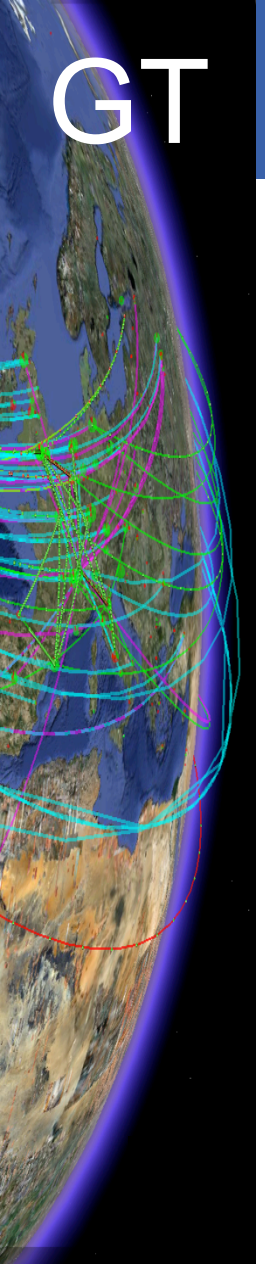


Messaging Services @ CERN

HEPiX Spring 2013 Workshop

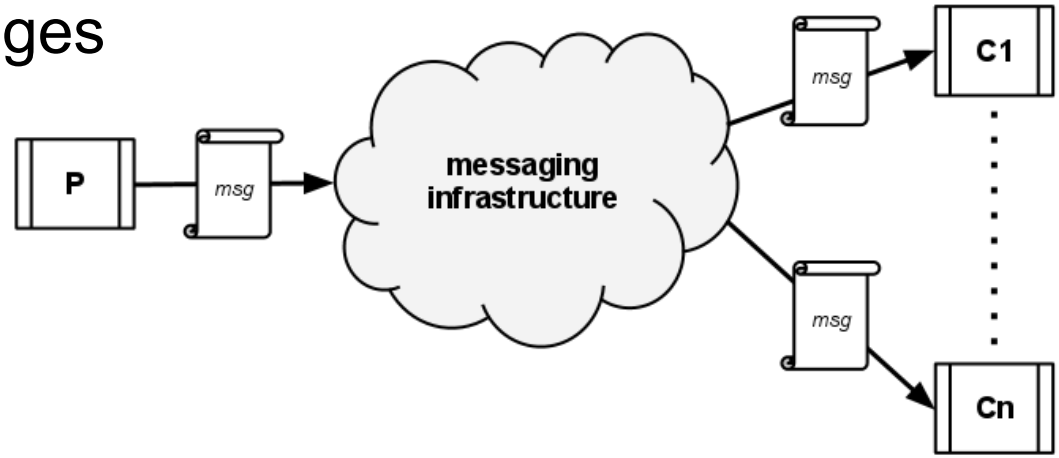
Lionel Cons, Massimo Paladin - IT/GT/TOM

CERN IT Department
CH-1211 Geneva 23
Switzerland
www.cern.ch/it

Messaging

What messaging is

- Messaging is for software components what electronic mail is for people
- Information production and consumption can be decoupled (asynchronous) and use different:
  - programming languages
  - operating systems
  - network protocols
  - hosts
  - time



- Only a common understanding of the message format (JSON, XML, **`protobuf`**…) is required

What messaging is not

- Messaging is not suited to:
  - large data transfers
  - low latency communication
- Message brokers are not message stores
  - better think of them as higher-level routers
- Messaging is not a magic security solution
  - you still need to define who is allowed to do what

☞ Messaging is not a magic bullet

Messaging buzzwords

- API: JMS
- Concepts: acknowledgement, binding, body, channel, credit, encoding, exchange, header, link, message, node, persistence, queue, selector, session, subscription, topic, transaction…
- Protocols: AMQP, MQTT, OpenWire, STOMP, XMPP…
- Software: ActiveMQ, Apollo, HornetQ, Qpid, MRG, RabbitMQ…

Messaging protocols

- STOMP = *Streaming Text Orientated Messaging Protocol*
  - very simple and text based
  - implemented in many languages
  - supporting basic messaging features (e.g. transaction)
- AMQP = *Advanced Message Queuing Protocol*
  - major contributors: Cisco, Microsoft, Red Hat, banks…
  - complete but complex, limited native language support
  - binary protocol so in theory faster and more compact
  - major transition: from AMQP 0.x to AMQP 1.0
    - very new, no backward compatibility, future still unclear

☞ "Best protocol" depends on use cases and may change

ActiveMQ vs. Apollo vs. RabbitMQ

- ActiveMQ
  - created in 2004, 5.0 released in 2007, 5.8 Feb 2013
  - solid but bloated, mature but reaching its end of life
  - some missing management and monitoring features
- Apollo
  - created in 2011, 1.0 released Feb 2012, 1.6 Feb 2013
  - defined as “ActiveMQ's next generation of messaging”
  - feature complete but still missing big scale deployments
- RabbitMQ
  - created in 2007, 3.0.4 released on Mar 2013
  - small and neat but missing several features
  - impact of AMQP 1.0’s wider acceptance still unclear

Messaging software

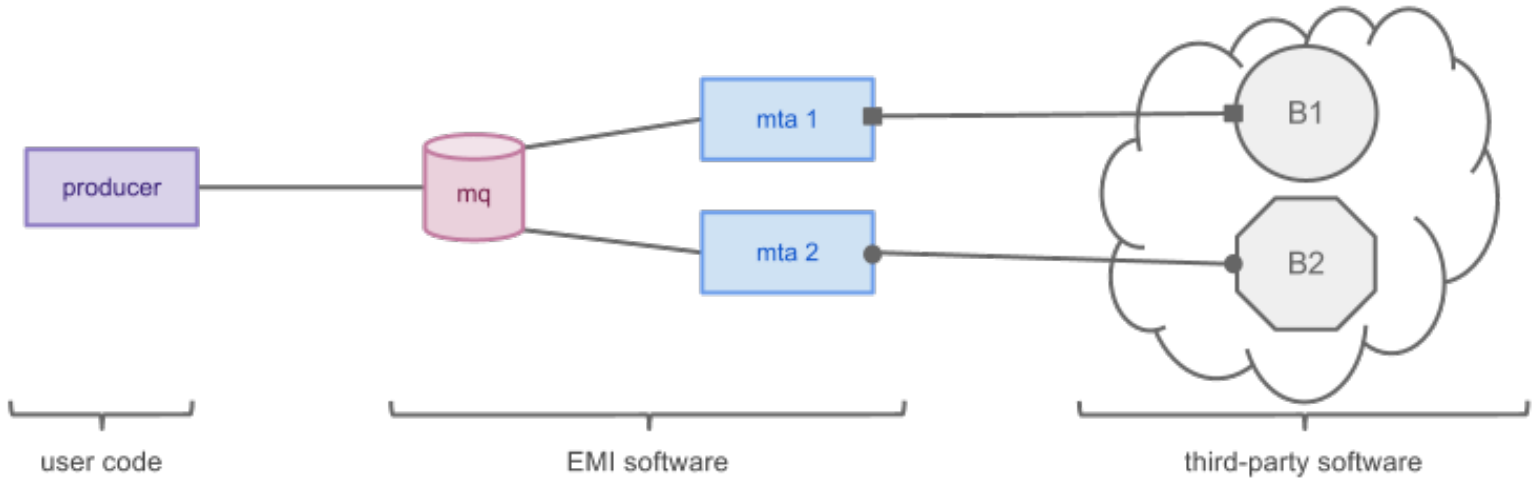
Broker	*Qpid*	*MRG*	*HornetQ*	*ActiveMQ*	*Apollo*	*RabbitMQ*
Language	Java	C++	Java	Java	Scala	Erlang
Main Protocols	AMQP	AMQP	proprietary STOMP	OpenWire STOMP AMQP MQTT	OpenWire STOMP AMQP MQTT	AMQP STOMP (MQTT)
Owner (*)	Red Hat			FuseSource (Progress)		VMware

7 September 2012 : Red Hat completed its acquisition of FuseSource
http://fusesource.com/redhat/

- No clear "best of breed" (yet) but this may change

Messaging clients (EMI)

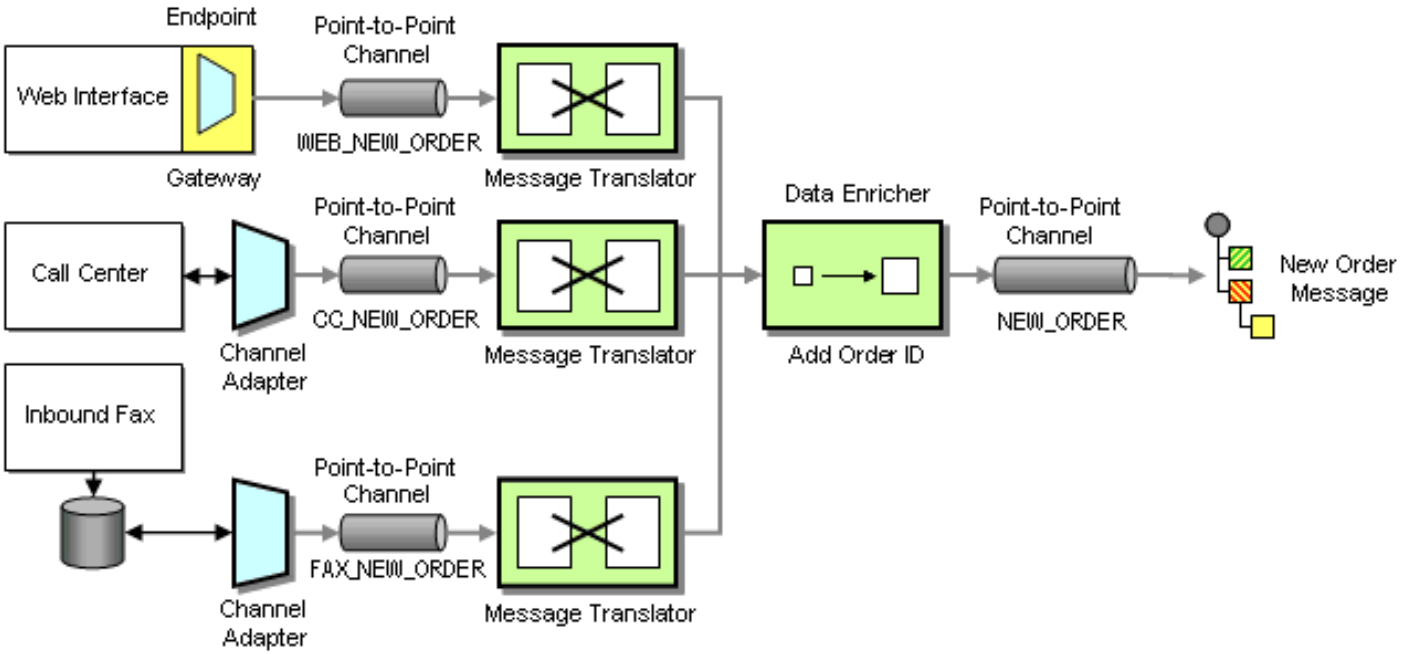
- EMI recommendation: reuse messaging blocks

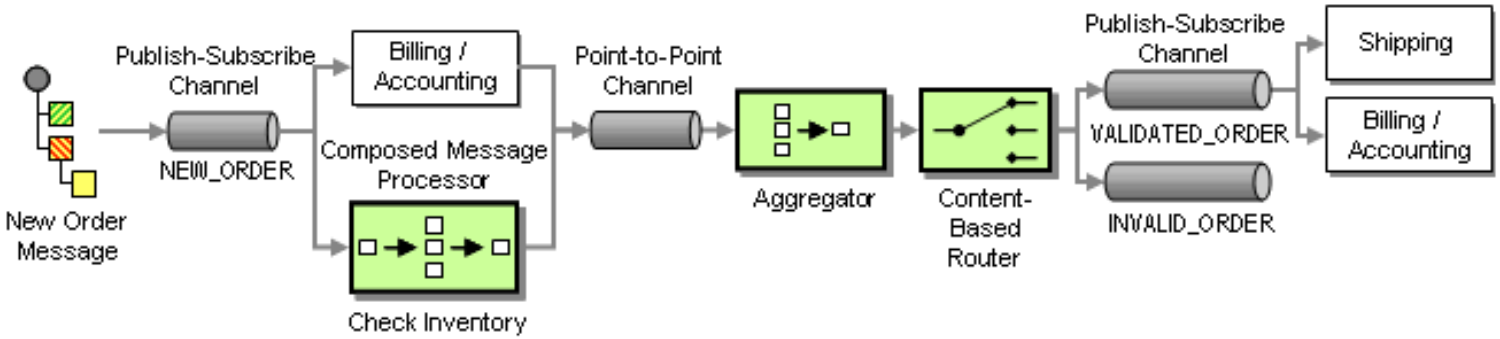


- See: http://cern.ch/messaging-chep2012
- So user code can be independent from brokers
  - no vaporware: this is being used today!

Enterprise Integration Patterns

http://www.eaipatterns.com/





Messaging Services

Messaging services design

- The chosen broker software is not very important
  … and will change with time anyway
- Key questions:
  - how many different applications per service?
  - how to best tune the broker software?
  - which resources (RAM, CPU…) are currently needed?
  - how to make sure the service can scale as needed?
  - prefer real hardware or virtual machines?
  - use a controlled or lax configuration?
- One size (probably) does not fit all

Messaging performance

- The number of messages per second is important
  - … but this is not the only parameter to consider
- Things to look at:
  - persistence
  - message size
  - amplification factor
  - number of concurrent connections
  - number of new connections per second
  - number of messages to keep in store
  - rates, sizes and counts distributions

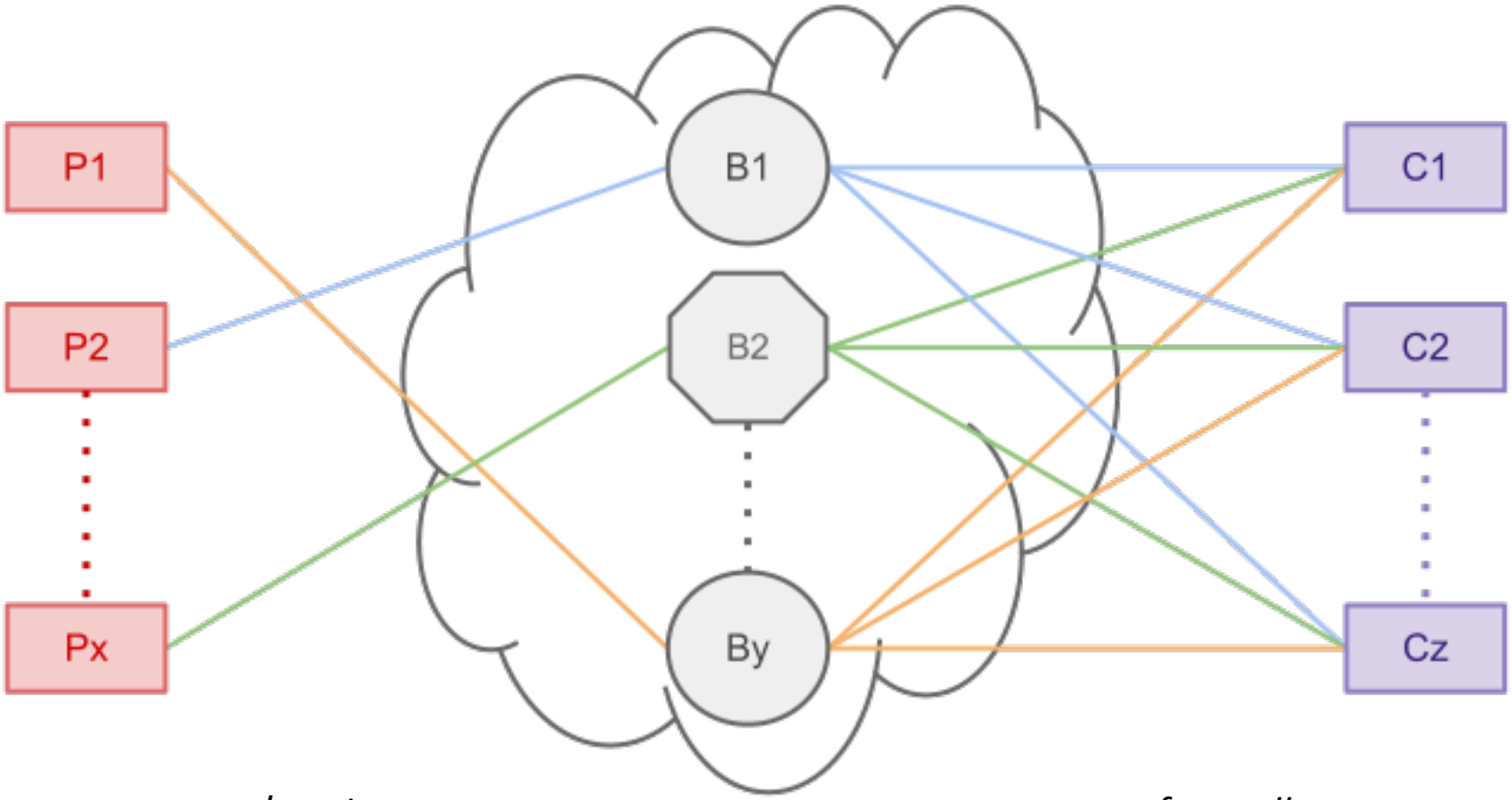
☞ Do not (blindly) believe what you can read but benchmark to find the best service setup for you

Trade-offs

- Computer center monitoring use case:
  - 120k clients sending 100 metrics every minute
- 1 metric per message → 200k msg/sec *versus*
  100 metrics per message → 2k msg/sec
  - drawbacks: bigger messages and harder to filter them with selectors
- Long-lived clients → 120k concurrent connections *vs.*
  1 flush every 5 minutes → ~400 concurrent connections
  - drawbacks: higher latency and 400 new connections per second
- Message compression:
  - CPU time *versus* network bandwidth and broker storage

Messaging services (EMI)

- dedicated messaging services made of independent brokers



Service quality

- The “RAID approach” is possible with messaging
  - however clients must be able to handle errors and retries
- High quality is also possible
  - messaging clients must be mature
    - buggy clients will always experience a buggy service
  - stability is a pre-requisite
    - test new broker software/configuration before deploying
    - more important: test changes of messaging clients too!
  - monitoring must be at the same quality level
    - monitor closely and react quickly

☞ Expertise is key to get the most out of messaging

Messaging Services @ CERN

Presented by FuseSource integration everywhere

CamelOne 2012 - Felix Ehm Video

Large Scale Messaging with ActiveMQ for Particle Accelerators at CERN

Felix Nikolaus Ehm, CERN

CERN, the international organization for nuclear research based in Geneva runs one of the largest particle accelerator infrastructure in the world including the youngest and most known Large Hadron Collider (LHC). This presentation will be on how ActiveMQ is used since 2005 for the accelerator Controls System to transport mission critical data reliably to high-level control/monitoring/alarm applications enabling a 24×7×365 operation for these machines. Currently single and clustered brokers are deployed to satisfy the very broad diversity of messaging patterns, e.g. large messages low rate (2MBytes/sec) to small messages high frequency (345M messages/day).

CamelOne 2012 Presentations and Videos are >> here

Videos Presentations from **CamelOne 2011** including Gregor Hohpe's keynote are >> here

Additional links:

- Fuse Enterprise 7.0 Demo
- Fuse ESB Enterprise
- Fuse MQ Enterprise
- Fuse IDE Developer Tooling
- Fuse Management Console for DevOps

FuseSource Subscription Center

Watch this short demo to gain more insight into all the great benefits of the FuseSource Subscription Center!

FuseSource Training

Existing messaging services

- EGI: SAM, APEL, dashboard notifications…
- ATLAS/DDM: tracer, notifications
- Dashboard: FTS monitoring, Xrootd (monitoring & popularity), Diane/Ganga monitoring, T3 job monitoring, ATLAS FAX monitoring…
- Agile Infrastructure: Lemon (metrics & notifications), Castor logs, Puppet's Mcollective, OpenStack's Nova
- Coming soon: ATLAS offline DQ, CORAL monitoring, SAM (new), BDII synchronization…

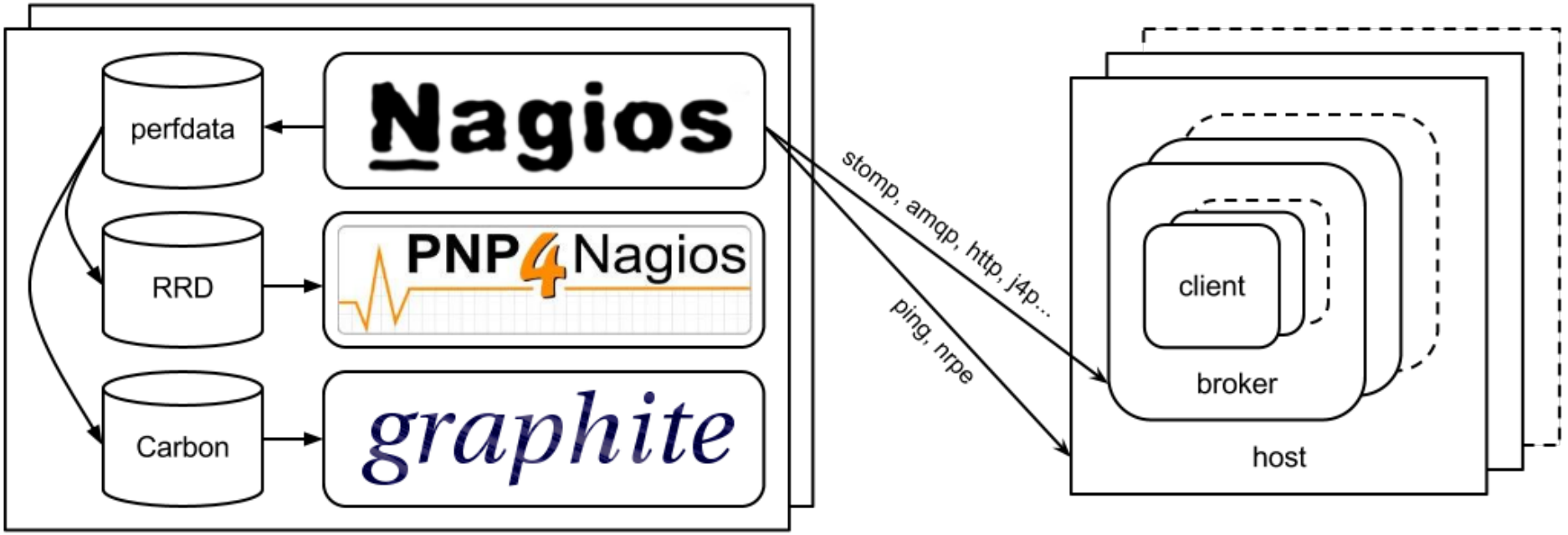
Service configuration

- Started with **`vi`** and YAIM (EGEE era)
- Now using:
  - high-level configuration files (mostly broker independent)
  - ActiveMQ, Apollo & RabbitMQ are supported
  - ad-hoc tools and scripts to generate files and/or rpms
- Current solution is fully compatible with YAIM, Quattor, Chef, Puppet, YUM, **`cfengine`**…
- Complete integration with Puppet is possible
  - different technical solutions exist

Service monitoring

- Started with Nagios (EGEE era)
- Current monitoring includes:
  - ~10 probes per broker instance
  - ~2 probes per messaging “client”
  - detailed metrics (broker specific)
  - trending, predictions, alerts…
    - including for client metrics
- Nagios configuration is large but generated from high-level configuration files
- Watch out for circular dependencies since Agile monitoring will use messaging!

Monitoring architecture



- 2 Nagios instances: production and development
- 24 hosts and 35 brokers
- ~600 “services” known by Nagios
- ~2500 metrics in Carbon

Nagios user interface

Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports
- Availability
- Trends
- Alerts
  - History
  - Summary
  - Histogram
- Notifications
- Event Log

System
- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Current Network Status
Last Updated: Tue Apr 2 09:41:56 CEST 2013
Updated every 601 seconds
Nagios® Core™ 3.4.4 - www.nagios.org
Logged in as *lionel.cons@cern.ch*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
60	0	0	0

All Problems	All Types
0	60

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
587	3	0	7	0

All Problems	All Types
10	597

Service Status Details For All Hosts

Limit Results: 100

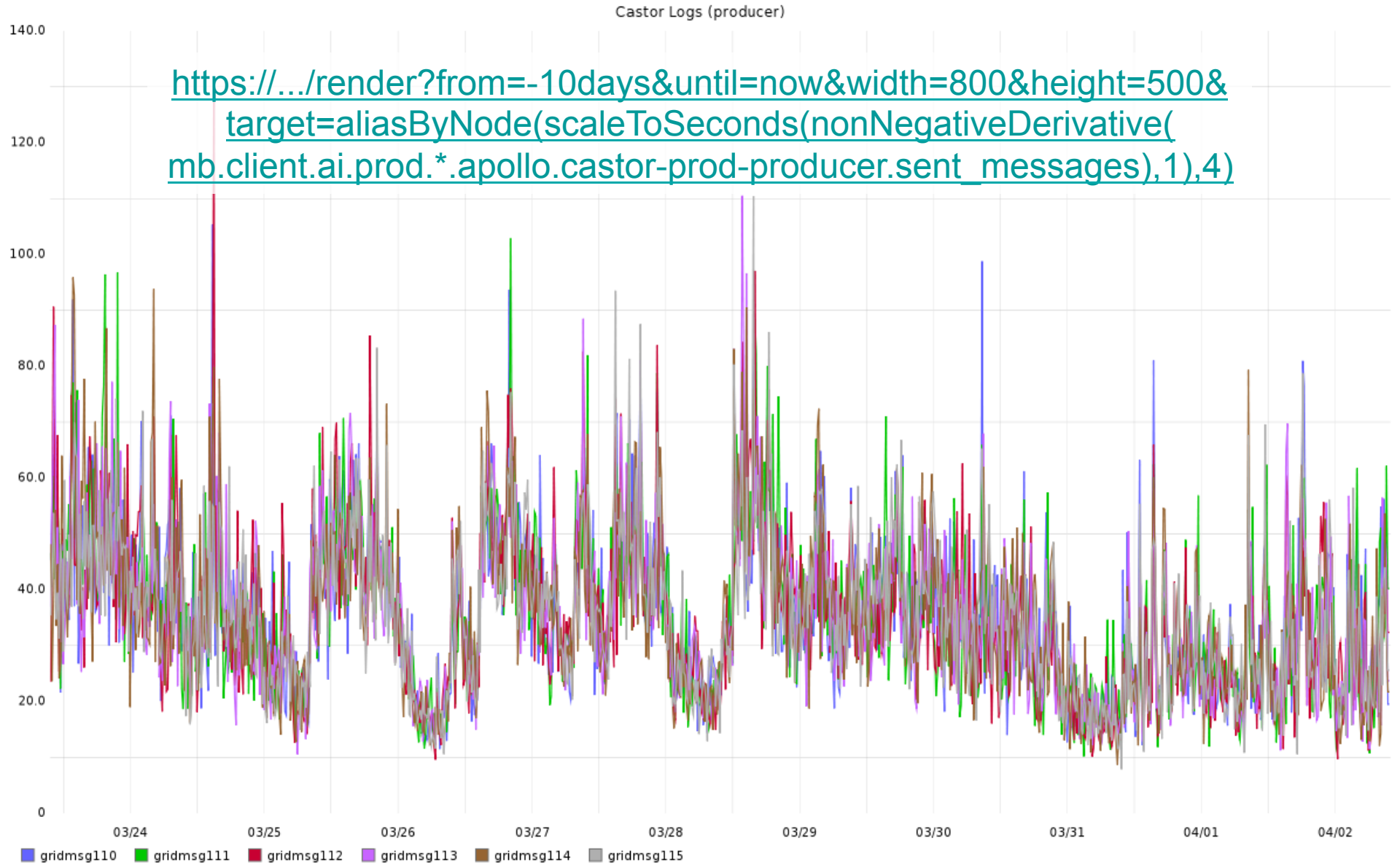
1 2 3 4 5 6

Results 0 - 100 of 597 Matching Services

Host	Service	Status	Last Check	Duration	Attempt	Status Information
activemq.egi-auth.msg.cern.ch	ActiveMQ.Health	OK	04-02-2013 09:41:16	10d 18h 48m 23s	1/3	ACTIVEMQ_HEALTH OK - jmx4perl OK, consumers OK, disk_store OK, disk_temp OK, heap OK, threads OK
	ActiveMQ.Processes	OK	04-02-2013 09:36:57	10d 18h 52m 42s	1/3	check_multi OK - 2 plugins checked, 2 ok
	ActiveMQ.Services	OK	04-02-2013 09:38:37	0d 17h 3m 19s	1/3	check_multi OK - 12 plugins checked, 12 ok
	ActiveMQ.Trend	OK	04-02-2013 09:38:24	25d 15h 22m 11s	1/3	check_multi OK - 4 plugins checked, 4 ok
	ActiveMQ.log_files	OK	04-02-2013 09:35:18	0d 16h 36m 38s	1/1	OK - no errors or warnings
	ActiveMQ.proc_tcp	OK	04-02-2013 09:36:56	0d 17h 10m 0s	1/3	PROC_TCP OK - connections ok for ports: 6162 6163 6166 6167 8161 62001
	client_brkpurge-consumer	OK	04-02-2013 09:37:39	0d 17h 9m 17s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_catchall	OK	04-02-2013 09:41:17	6d 13h 10m 52s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_egi-monitor-consumer	OK	04-02-2013 09:38:59	0d 17h 2m 57s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_egi-monitor-producer	OK	04-02-2013 09:41:40	10d 18h 46m 45s	1/3	check_multi OK - 1 plugins checked, 1 ok
	client_mb-monitor-queue-consumer	OK	04-02-2013 09:37:26	10d 18h 47m 19s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_mb-monitor-queue-producer	OK	04-02-2013 09:41:16	10d 18h 46m 36s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_mb-monitor-topic-consumer	OK	04-02-2013 09:36:57	10d 18h 50m 57s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_mb-monitor-topic-producer	OK	04-02-2013 09:37:38	10d 18h 50m 17s	1/3	check_multi OK - 2 plugins checked, 2 ok
activemq.egi-srce.msg.cern.ch	ActiveMQ.Health	OK	04-02-2013 09:41:16	4d 13h 25m 40s	1/3	ACTIVEMQ_HEALTH OK - jmx4perl OK, consumers OK, disk_store OK, disk_temp OK, heap OK, threads OK
	ActiveMQ.Processes	OK	04-02-2013 09:36:57	4d 13h 24m 59s	1/3	check_multi OK - 2 plugins checked, 2 ok
	ActiveMQ.Services	OK	04-02-2013 09:38:38	0d 0h 43m 18s	1/3	check_multi OK - 12 plugins checked, 12 ok
	ActiveMQ.Trend	OK	04-02-2013 09:41:47	25d 8h 47m 2s	1/3	check_multi OK - 4 plugins checked, 4 ok
	ActiveMQ.log_files	WARNING	04-02-2013 09:38:03	13d 23h 19m 42s	1/1	WARNING - 4 warnings
	ActiveMQ.proc_tcp	OK	04-02-2013 09:40:58	4d 13h 25m 58s	1/3	PROC_TCP OK - connections ok for ports: 6162 6163 6166 6167 8161 62001
	client_brkpurge-consumer	OK	04-02-2013 09:41:40	4d 13h 25m 17s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_catchall	OK	04-02-2013 09:37:26	4d 13h 29m 30s	1/3	check_multi OK - 2 plugins checked, 2 ok
	client_egi-monitor-consumer	OK	04-02-2013 09:37:16	3d 0h 29m 40s	1/3	check_multi OK - 2 plugins checked, 2 ok

Graphite user interface

https://.../render?from=-10days&until=now&width=800&height=500&target=aliasByNode(scaleToSeconds(nonNegativeDerivative(mb.client.ai.prod.*.apollo.castor-prod-producer.sent_messages),1),4)



Higher level abstraction

- Duplication of information exists between:
  - high-level configuration → broker configuration files
  - high-level configuration → monitoring configuration files
- A higher-level abstraction:
  - can define clusters, hosts, brokers, services, clients…
  - can be the "source of truth" for:
    - broker and monitoring high-level configurations
    - system management: Quattor, Puppet…
    - also: DynDNS, documentation…
  - could be for messaging what LANDB is for networking
  - has been prototyped using Django + MySQL

Messaging as a service

http://www.cloudamqp.com/

CloudAMQP | Plans | Documentation | Support | Control panel

Plans & Prices

Little Lemur

- 30 MB of messages
- 3 connections

Free

Tough Tiger

- 2 GB of messages
- 6 connections

$ 19 per month

Big Bunny

- 20 GB of messages
- 20 connections

$ 99 per month

Roaring Rabbit

- 100 GB of messages
- 100 connections

$ 299 per month

Power Panda

- Unlimited traffic
- Unlimited connections
- Custom plugins
- Phone support

$ 499 per month

Create a fully managed RabbitMQ instance now ▶

Now available in both US and EU!

Or add it to your existing applications at:

heroku | AppHarbor | cloudControl

Always included

- High availability clusters
- Mirrored queues
- Unlimited number of queues
- Unlimited number of exchanges
- Full access to your virtual host
- Access to management interface
  - Administration
  - Statistics
  - Inspection
- 24/7 monitoring
- Fully automated platform

Messaging future at CERN

- Messaging has proven to be a useful paradigm
- Use of messaging will continue to grow, including more diverse and demanding use cases
- Messaging technology may change drastically:
  - convergence of all Red Hat products?
  - widespread adoption of AMQP 1.0?

☞ CERN messaging services will grow and evolve to match the needs

http://cern.ch/mig