WEPKN006

ICALEPCS'11

icalepcs 2011

13th International Conference on Accelerator and Large Experimental Physics Control Systems
October 10-14, 2011, WTC Grenoble, France

Running a Reliable Messaging Infrastructure for CERN's Control System

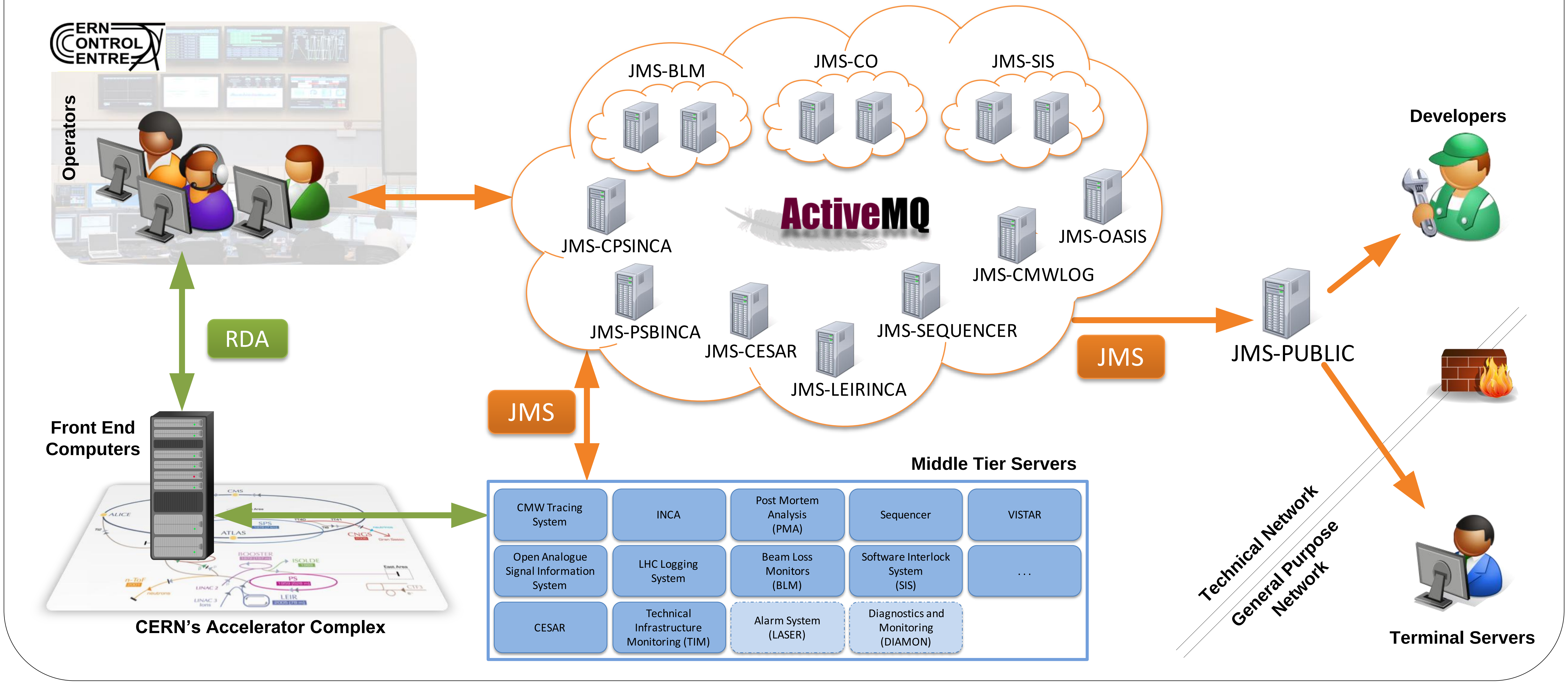
F. Ehm, CERN, Geneva, Switzerland

Abstract

The current middleware for CERN's Control System is based on two implementations: CORBA-based *Controls MiddleWare* (CMW) and *Java Messaging Service* (JMS). The JMS service is realized using the open source messaging product *ActiveMQ* and had became an increasingly vital part of beam operations as data need to be transported reliably for various areas such as the beam protection system, post mortem analysis, beam commissioning or the alarm system. The current JMS service is made of 14 brokers running either in clusters or as single nodes. The main service is deployed as a two node cluster providing failover and load balancing capabilities for high availability. Non-critical applications running on virtual machines or desktop machines read data via a third broker to decouple the load from the operational main cluster. This scenario has been introduced In 2010 and the statistics showed an uptime of 99.998% and an average data serving rate of 1.6GByte per minute represented by around 150 messages per second.

Deploying, running, maintaining and protecting such messaging infrastructure is not trivial and includes setting up of careful monitoring and failure pre-recognition. Naturally, lessons have been learnt and their outcome is very important for the current and future operation of such service.

Messaging in Controls Architecture

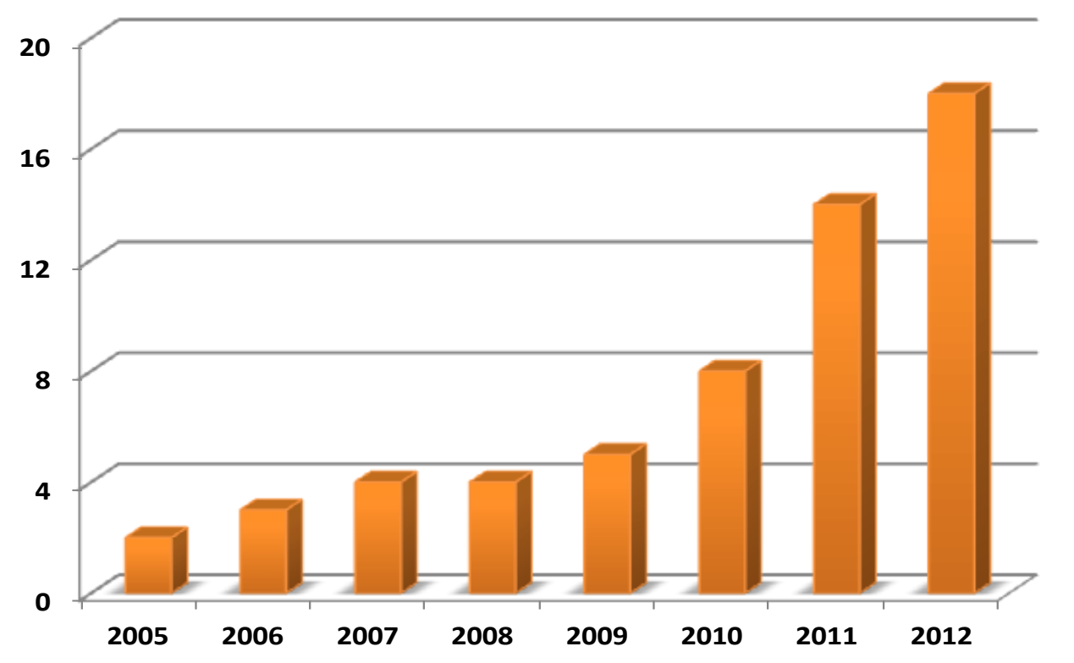


CMW Tracing System	INCA	Post Mortem Analysis (PMA)	Sequencer	VISTAR
Open Analogue Signal Information System	LHC Logging System	Beam Loss Monitors (BLM)	Software Interlock System (SIS)	. . .
CESAR	Technical Infrastructure Monitoring (TIM)	Alarm System (LASER)	Diagnostics and Monitoring (DIAMON)	

Operation

Infrastructure Architecture Key Points

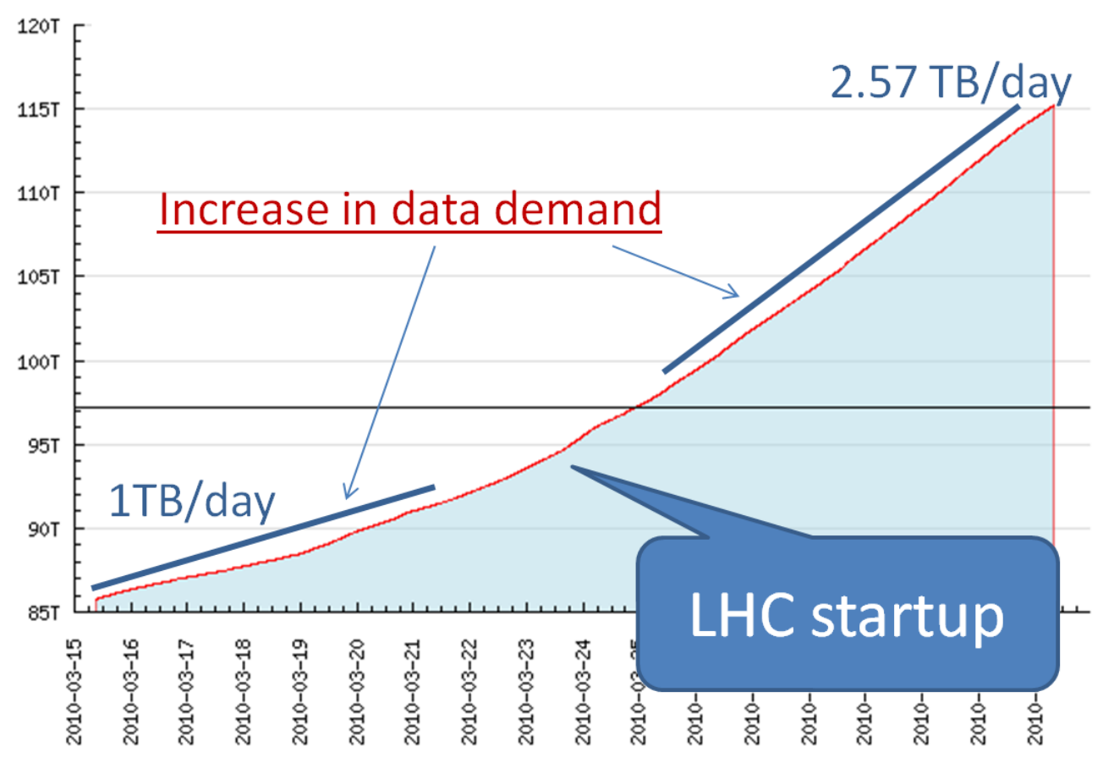
1. Usage of **Service Level Agreements**.
2. Separation of **non-operational** clients.
3. New broker version is **tested in CO Testbed** before going to production.
4. Monitoring for **performance and availability**.
5. Cluster setup for selected services for **load balancing** and failover.
6. Project dedicated brokers avoid **interference**.



Growing Messaging Infrastructure

Performance and Numbers

Overall
- 81 different applications request data
- 1400 concurrent connections
- 6 Million messages per day
- 20.000 subscriptions
- Broad range of message usage patterns (2MByte @ 0.5Hz - 1KByte @ 10kHz)



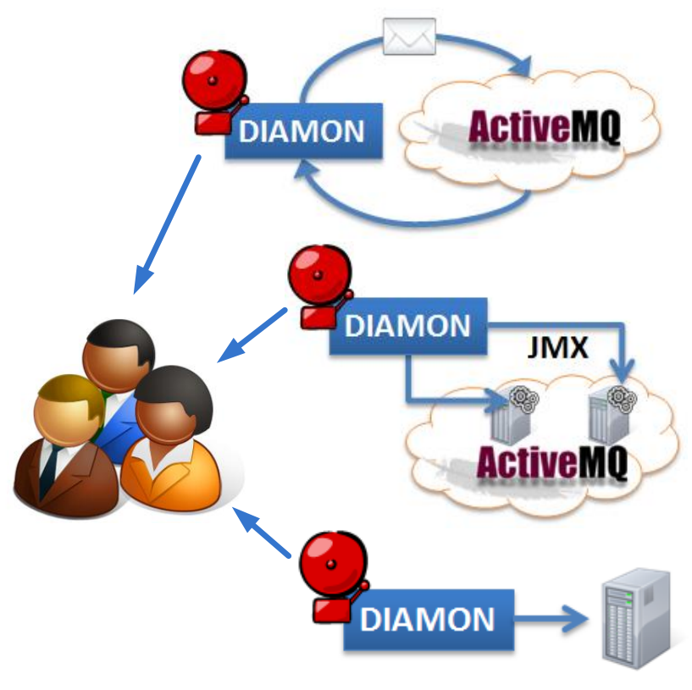
Example of data handling for JMS-CO

Uptimes
- Single brokers reach uptimes of **> 220 days** (kernel update force restart).
- Clusters reached **99.998 %** service availability in 2010.

Monitoring

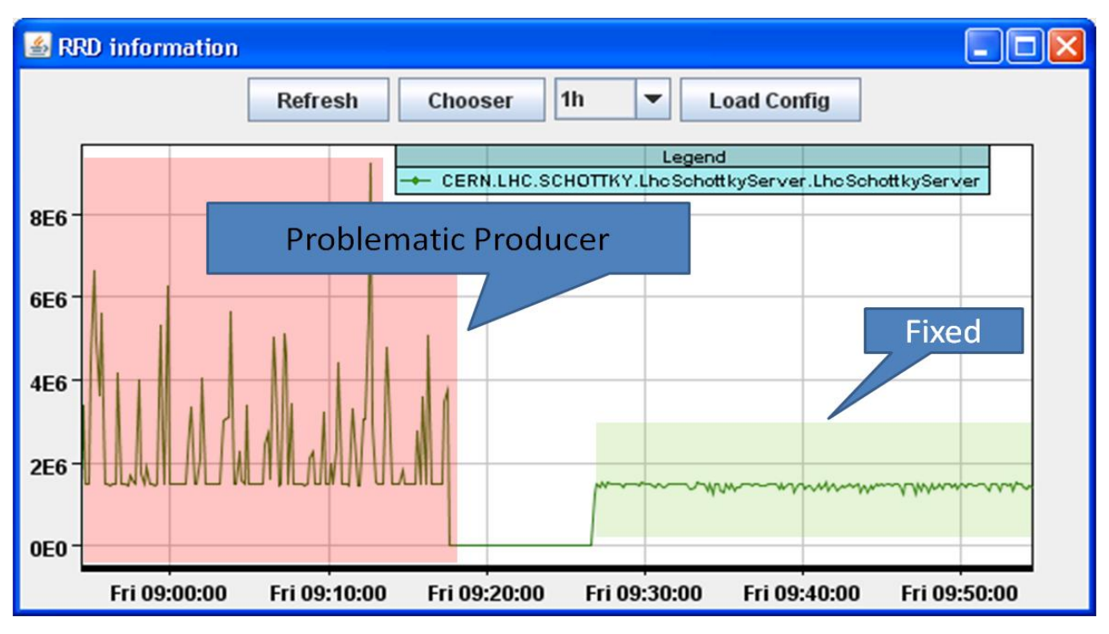
Monitoring Hardware and Broker Software

A broker is monitored for **availability and performance** by CERN's Diagnostic and Monitoring (DIAMON) tool. To determine the health of the broker it uses **metrics** available via the Java Management Interface (JMX) and it sends a **test message**. For the host machine the cpu, memory and disk are surveilled.
In case of a failure or breaching of a **warning or error threshold** a notification is sent via eMail and SMS to the Middleware team.



In-house developed Tools

Topic Monitoring Tool
- Subscribes to Topics on a broker and reads messages to store their **size** and their **sending speed**.
- Viewer program allows a history view on the collected data.
- Irregular or unusual messages patterns can be recognized easily.



Example of detecting a non-functional process

JMXDumper
- Dumps all information available via JMX into a SQLite database
- Analyses the data to print summary on producers / consumers / connection

Conclusions

ActiveMQ was and is a **good choice** for decoupled messaging for the CERN's Control System and has proven to be very stable. Its capability of **scaling linearly** makes it an obvious choice for an environment where the number of reading applications is very dynamic and data demand is growing at the same time. Because unexpected high load is possible it is important to dimension machine resources sufficiently. **Monitoring** is a vital part of operation and the evaluation of the recordings must take influence on future deployment decisions. Effective service downtime is reduced by deploying a **messaging service per usage domain** or project. Service Level Agreements track the (growing) user requirements, help to adapt to usage scenarios and support setting up monitoring thresholds.