# The AAL project: Automated monitoring and intelligent AnaLysis for the ATLAS data taking infrastructure

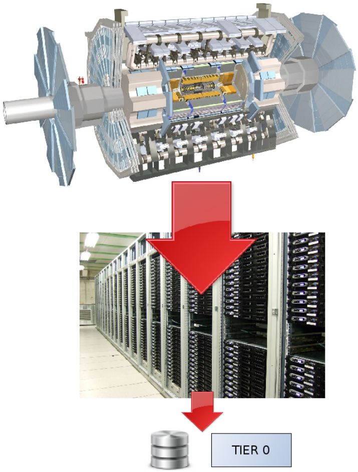
Magnoni Luca
*CERN PH-ADT*

The ACAT2011 conference
September 05th, 2011

# OUTLINE

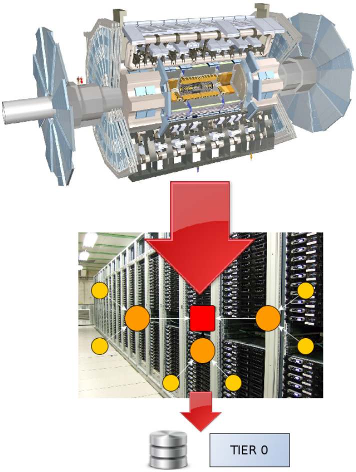
# TRIGGER AND DATA ACQUISITION (TDAQ) SYSTEM

- The TDAQ system is responsible for **filtering** and **transferring** data from detectors to the mass storage system
  - particle interactions at 40 MHz
  - every interaction produces ∼ 1MB of data event
  - most of these millions of generated events are totally uninteresting
  - a filter mechanism is needed to select the more interesting ones



# Trigger and Data Acquisition (TDAQ) system

- The TDAQ system relies on a large computing environment with **thousands of software applications** running concurrently and interacting with each other
  - online computing farm with ∼ 2000 nodes
  - ∼ 20000 running applications
  - highly distributed system
  - 3 independent GbE networks: controls, data collection and event filtering
  - Linux OS and multi-threadedsoftware



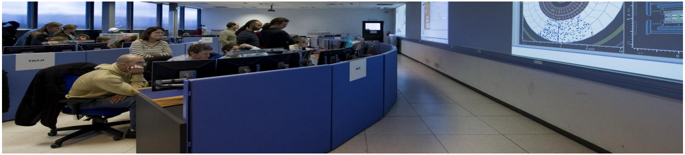
# Running the system

We aim at **maximizing operations efficiency**

- minimizing system down-time
- dealing fast and effectively (and possibly automatically) with errors and failures

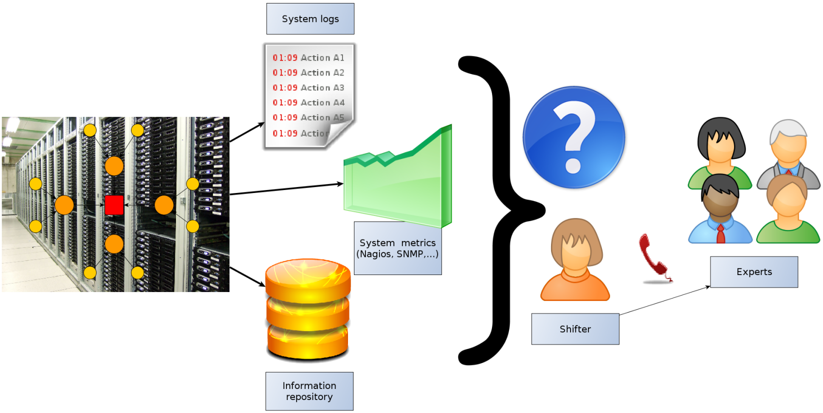
The flow of operational data as well as the status of the infrastructure is constantly monitored by shifters and experts with the help of automated tools

# Shifters and experts



- The system is operated by a non-expert shift crew, assisted by a set of experts providing knowledge for specific components
- Operational tasks can be divided in:
  - Operational procedures to run the system
  - A set of periodic checks and controls
  - Notify experts in case of problems

# System analysis and monitoring



# We can do better - automation!

It's time to have the Robot doing the work for you
*iRoomba vacuum cleaner ad*

Computers are good in automation. Checks and controls can be *easily* automated.

**Aim:**

- reduce and simplify shifters tasks
- help shifters with more detailed and pertinent information
- be more efficient, avoid repetition
- formalize knowledge from experts

# "AAL" THE ASSISTANT

## The Assistant

- A tool meant at guiding the operator in his daily work. It can both help diagnosing problematic situations and suggesting actions to take, as well as remind the operator that he should (not) do something.
- DAQ systems already provide all the information we need, it is a matter of using it effectively.

# WHY INTELLIGENT

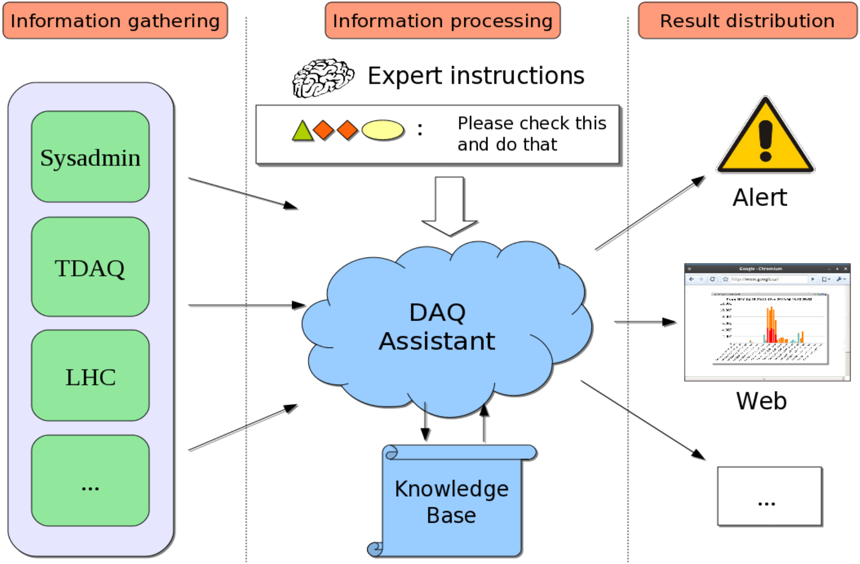
- To process and analyze high rate of different stream of events
- To detect known problem (driven by expert input)
- To automatically detect anomalies on online stream of data
- To support several output handlers

# TDAQ challenges

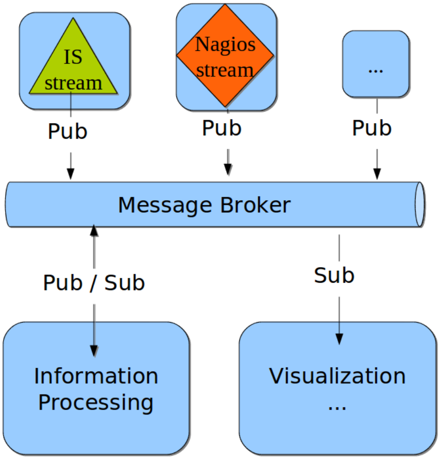
- Many information sources (log streams, information system, Nagios, ...)
- Several technologies to gather data (DB, API library, ...)
- Heterogeneous data (format, publication mechanism)
- Dynamic system conditions (calibration, physics, ...)

## Main challenges

1. **Information gathering** (different streams with thousands of information update per second)
2. **Information processing** (building Knowledge Base, process data streams and discover complex patterns)
3. **Present results** in several way


Information gathering
Information processing
Result distribution
Expert instructions
: Please check this and do that
Sysadmin
TDAQ
LHC
...
DAQ Assistant
Knowledge Base
Alert
Web
...

# 1) Information gathering: Event driven architecture




- Message-oriented communication
  - Unique bus
  - Publish/subscribe
- Loose-coupled components
  - Scalability and Modularity
  - Independent components
- **Apache ActiveMQ** as broker
  - JMS standard
  - Cross language clients
  - Multiple wire protocols
  - Multiple network protocols

# 2) REAL TIME INFORMATION PROCESSING

- **Complex Event Processing** (CEP)
- Technology to **process events and discover complex patterns among streams** of events
- Based on rules or queries (SQL-like)
- Continuous evaluation
- History
  - On 1995 prof. David Luckham from Stanford coined the term CEP
  - Database research topic: Data Stream
  - Management System (DBMS)
  - Used in financial analysis, business process management, etc.

# ESPER FROM ESPERTECH

- Open source event processing engine
  - detects pattern among stream of events
  - Java and GPL
  - strong community and support
  - good documentation
  - the most widely- deployed CEP engine
- Simple API
  - easy to integrate and use
  - strong performance
- Powerful Event Processing Language (EPL)
  - SQL-like
  - Data windows of time
  - Pattern matching
  - Event aggregations

# Hello world with Esper

**Define Event:**

```
public class HelloWorld {}
esper.addEventType(HelloWorld.class.getName())
```

**Define EPL Statement:**

```
EPStatement statement =
esper.create("select * from HelloWorld");
```

**Attach Observer:**

```
public class MyObserver {
   update(Event[] events) {
         System.out.println ("Hello World"); }
   }
statement.setObserver(new MyObserver());
```

**Inject Events:**

```
esper.sendEvent(new HelloWorld());
```

# EPL STATEMENTS

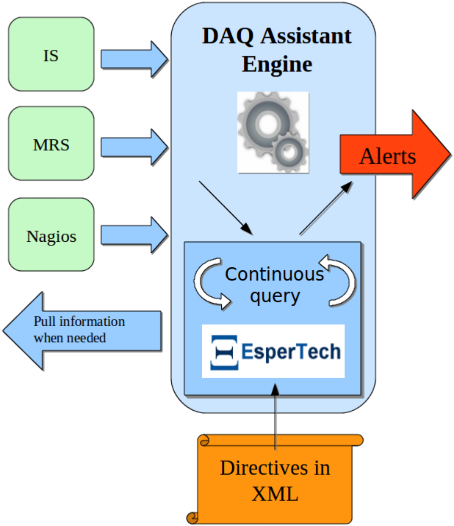
- Single event
  - `every Message(messageID=xyz)`
- Data window and filtering
  - `select * from Message(severity='Error').win:time_batch(30 sec)`
- Aggregation
  - `select messageID, messageTxt, sum(numberOfMessages) from Message.win:time_batch(30 sec) group by messageID, messageTxt`

# EPL statements contd.

- Detect outlier
  - `select applicationID, sum(numOfMessages) from Message.win:time(30 sec) group by applicationID` **`having sum(numOfMessages) > avg(numOfMessages)*0.75`**
- Call Java method
  - `select applicationName, info from` **`method:ISReader.getISInfo(test.RunParams.XYZ)`** `as info, Message.win:time_batch(5 sec) group by applicationName`
- Seamless Database integration
- Much, much more
  - `subquery, -> (followed by), timer actions, ...`

# AAL Engine in action

- Data gathered and feed into the engine
- EPL statements (from KB) are evaluated against data (continuous query)
- Generating alert, notification, statistics as soon as incoming events meet the constraints of the rule



# XML KNOWLEDGE BASE

**DIRECTIVE**

**What to detect**

```
select e.partitionName,e.errorDescription
from ISEvent(partitionName='ATLAS' or 'initial
and fault = 'true') as e
```

**How to react**

**Properties:**

```
ALERT:message = 'Partition in error!'
ALERT:action = 'Please do something!'
```

**Writer <mrs,jms, file>:**
Define the format of the alert

Writer Properties (MRS options, JMS options, etc.)

# XML KNOWLEDGE BASE

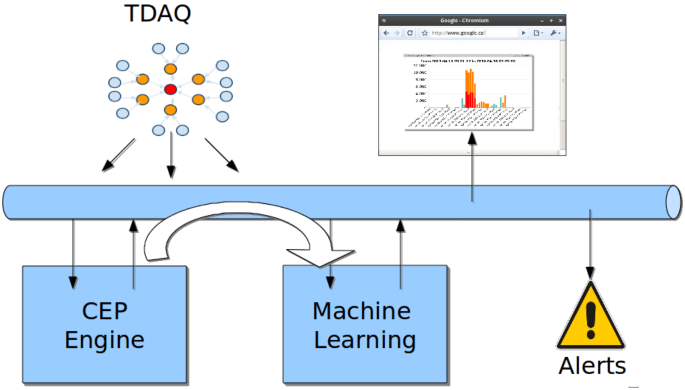

```
-<cassandra>
  -<!--

          ROS Load directive. Active only when ATLAS is running.
          This directive check for every ROS if the
          avg (rosLoad) > 70 and the numberOfQueueElement >20 in the last minute.
          NOTE: using the std:groupwin to look for the 1 min window per ros name, not aggregated

  -->
  -<directive name="ROSLoad">
    -<epl>
       select name, avg(attributes('rosLoad').long) as ROSLoad, avg(attributes('numberofQueueElement').long) as ROSQueue from
       ISEvent(partitionName="ATLAS", name regexp 'ROS.ROS[^.]+', type="ros").std:groupwin(name).win:time(1 min) group by name having
       avg(attributes('rosLoad').long) >70 and avg(attributes('numberofQueueElement').long) > 20
     </epl>
    -<listener type="alert">
       <domain>DAQHLT.CHECKLIST</domain>
       <severity> WARNING </severity>
       <message>This ROS is close to saturation!</message>
       <action>Check the whiteboard and react properly!</action>
       <details>true</details>
      -<writer type="file">
         <partition>assistant</partition>
         <severity>ERROR</severity>
       </writer>
     </listener>
   </directive>
  +<!---->
  +<directive name="SFOThroughput"></directive>
  +<!---->
  +<directive name="badcounter"></directive>
 </cassandra>
```

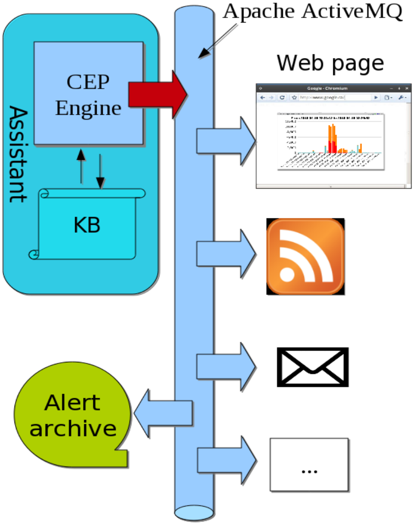
# 3) CEP and machine learning

- the engine can be used to normalize a chaotic flow of data
- joins the multiple (heterogeneous and asynchronous) streams to create an input vector
- feed an intelligent system to detect never seen anomalies



# Results distribution

- Different result types
  - Alert messages
  - Statistics
- Message oriented communication
  - Loose coupled communication among several components
  - Flexible way to create new visualization component
  - Useful built-in facilities (RSS, XMPP)



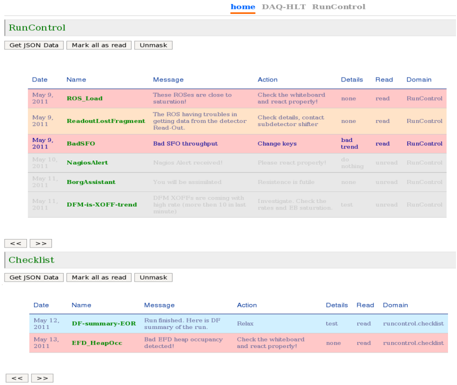
# Alert

Alerts are the main output of the assistant.

- **Problem description**
- **Expected reaction**
- **Severity**
- **Hierarchical domain**
- **Details**: every information collected by the engine that triggered the rules

# WEB PORTAL

- Web page for interactive visualization of alerts
  - Alert grouped per categories/user preferences
  - User interaction:
    - Mark alert as read when problem solved
    - Mask alerts
  - Automated Alert removal
  - Alert history
- Django project, SQLite archive and some jQuery goodies



home DAQ-HLT RunControl

RunControl

Get JSON Data | Mark all as read | Unmask

| Date | Name | Message | Action | Details | Read | Domain |
|---|---|---|---|---|---|---|
| May 9, 2011 | ROS_Load | These ROSes are close to saturation! | Check the whiteboard and react properly! | none | read | RunControl |
| May 9, 2011 | ReadoutLostFragment | The ROS having troubles in getting data from the detector Read-Out. | Check details, contact subdetector shifter | none | read | RunControl |
| May 9, 2011 | BadSFO | Bad SFO throughput | Change keys | bad trend | read | RunControl |
| May 10, 2011 | NagiosAlert | Nagios Alert received! | Please react properly! | do nothing | unread | RunControl |
| May 11, 2011 | BorgAssistant | You will be assimilated | Resistence is futile | none | unread | RunControl |
| May 11, 2011 | DFM-is-XOFF-trend | DFM XOFFs are coming with high rate (more then 10 in last minute) | Investigate. Check the rates and EB saturation. | test | unread | RunControl |

<< >>

Checklist

Get JSON Data | Mark all as read | Unmask

| Date | Name | Message | Action | Details | Read | Domain |
|---|---|---|---|---|---|---|
| May 12, 2011 | DF-summary-EOR | Run finished. Here is DF summary of the run. | Relax | test | read | runcontrol.checklist |
| May 13, 2011 | EFD_HeapOcc | Bad EFD heap occupancy detected! | Check the whiteboard and react properly! | none | read | runcontrol.checklist |

<< >>

# CONCLUSION

- AAL is working since June 2011
- An intelligent monitoring improves operator's tasks and expert's nights
- Gathering/processing/visualization are all main challenges
- Event driven architecture allows for a loose-coupled independent modules
- Doing CEP with Esper is a powerful solution:
  - detect simple and complex known patterns
  - correlation on streams of event with built-in time window support
  - next step: regularize and formalize a chaotic flow of data to feed the learning module

THANK YOU!
ANY QUESTION?