

# Applications of Advanced Data Analysis and Expert System Technologies in the ATLAS Trigger-DAQ Controls Framework

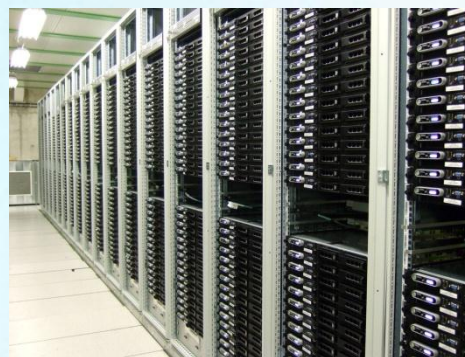
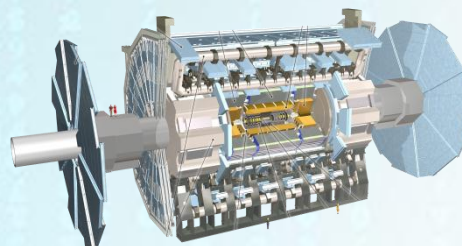
G. Avolio – University of California, Irvine

# Outline

- **The ATLAS Trigger and Data Acquisition (TDAQ) system**
  - General schema
  - Computing infrastructure
- **Intelligent systems and automation**
  - Why in TDAQ?
- **Error management in TDAQ**
  - Error Management System
    - Online Recovery
    - Diagnostic and Verification System (DVS)
  - Complex Event Processing
    - The “DAQ Assistant”
- **Conclusions**

# Trigger and Data Acquisition System

- The **Trigger and Data Acquisition (TDAQ)** system is responsible for filtering and transferring data from the detector to the mass storage
  - **40 millions** particle interactions per second
  - More than **1.5 MB** of data for each event
  - Most of the generated events are **totally uninteresting**
    - A filter mechanism is needed in order to select and collect the more interesting ones



From detector to data storage

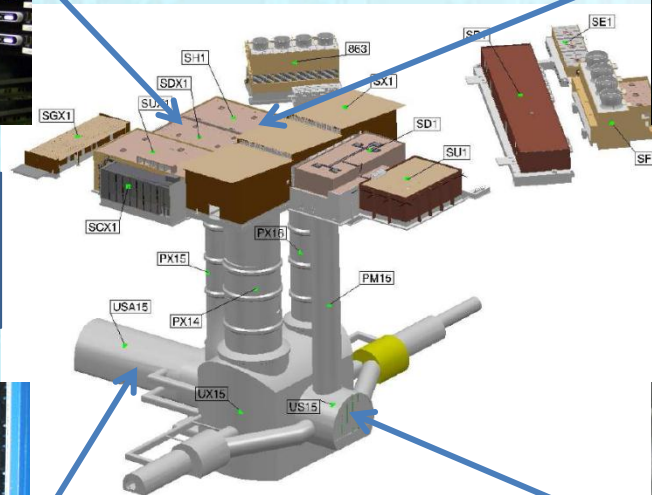
# TDAQ Computing Infrastructure



More than 12k cores  
and 20k applications



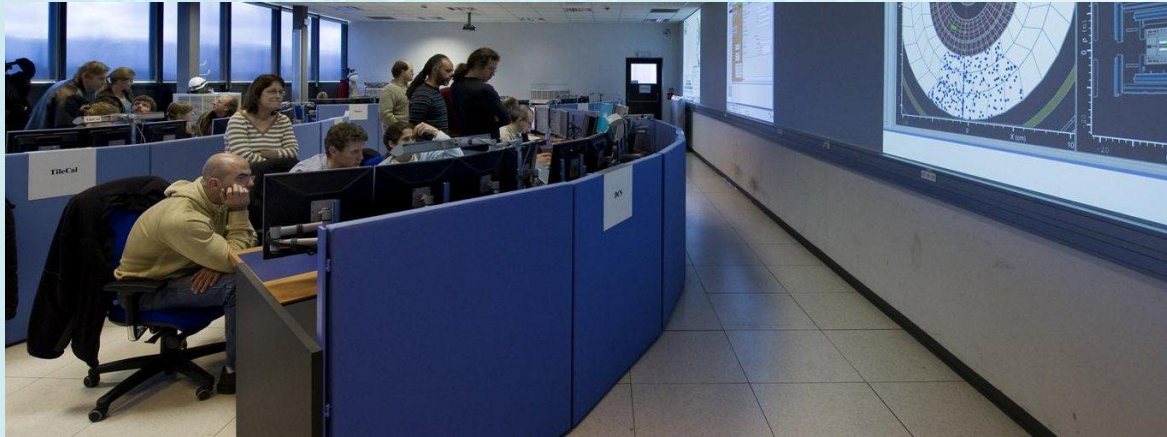
1600 point-to-point  
connections from  
detector to TDAQ



GbE network(s) with more  
than 10k channels

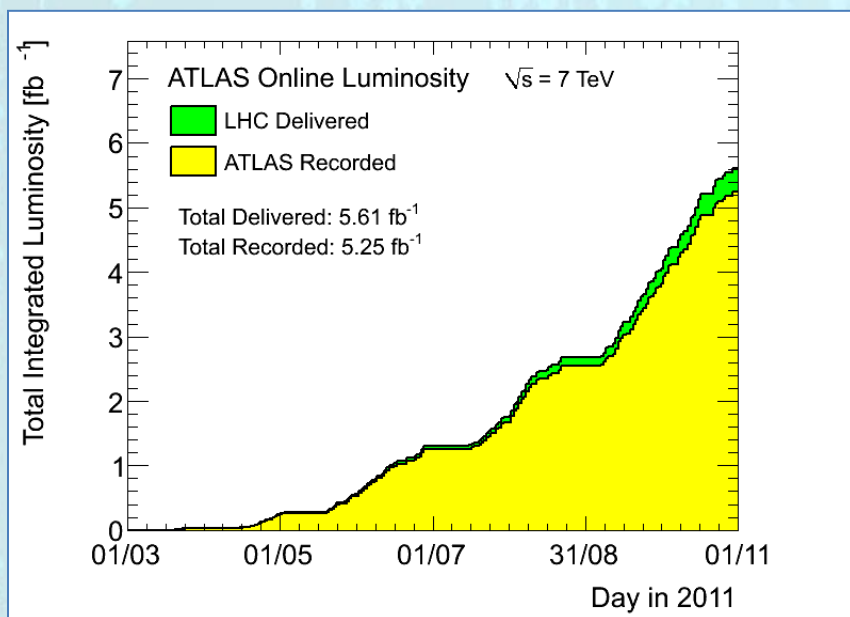


# Why Intelligent Systems in TDAQ?



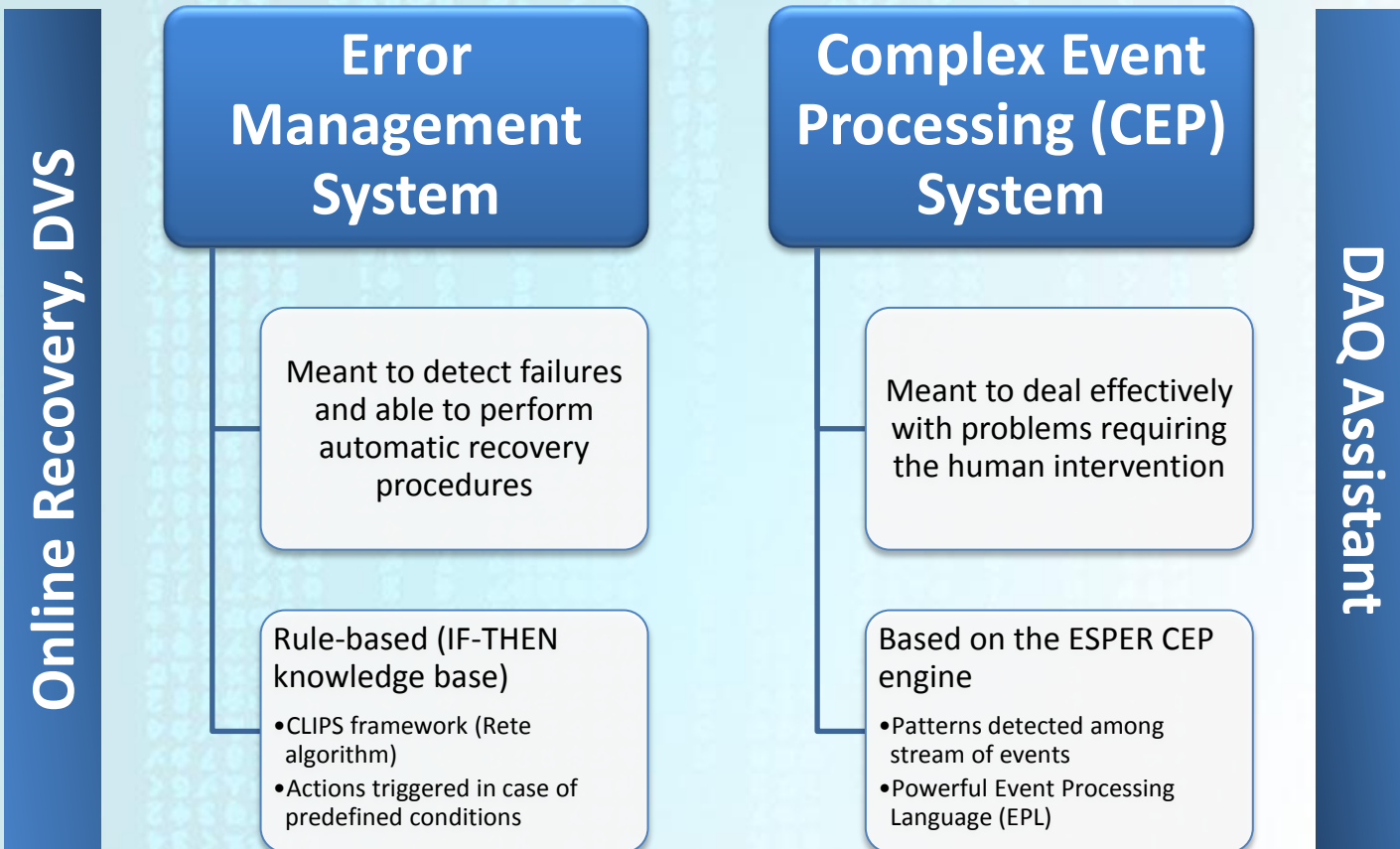
- **The main goal operating the system is to maximize the data taking efficiency**
  - Dealing fast and effectively with errors and failures
- **The system is operated by a non-expert shift crew assisted by experts providing knowledge for specific components**
  - Inefficiency may come from human interventions
- **Automating error detection, diagnosis and recovery is a key feature**
  - Effective analysis and monitoring system

# ATLAS Data Taking Efficiency in 2011



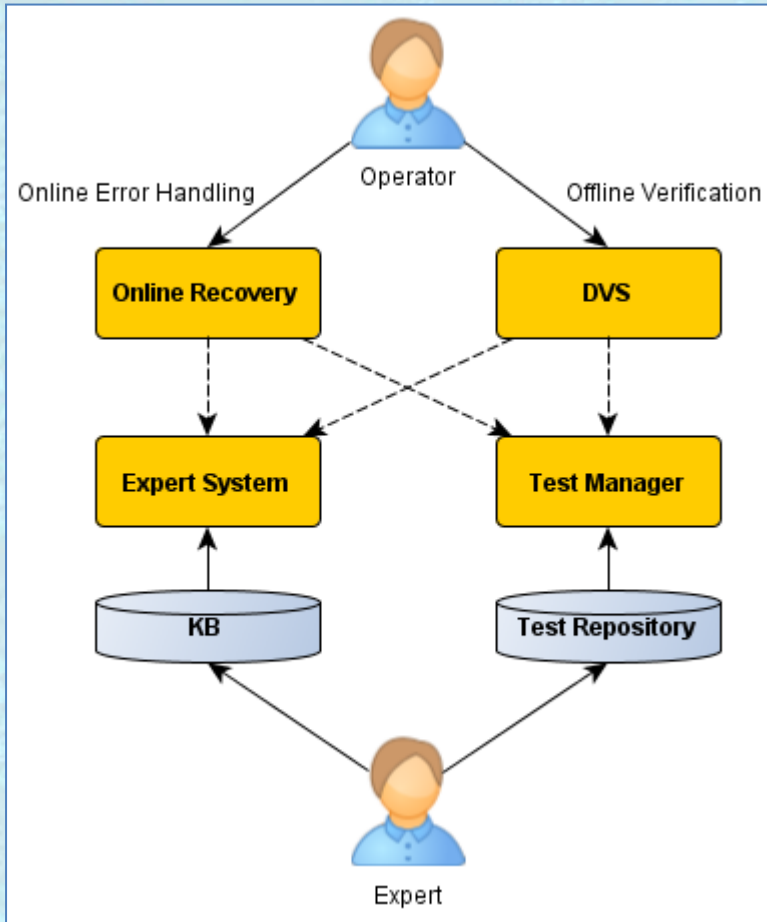
- In 2011 ATLAS was able to record **93.5%** of the total luminosity provided by the accelerator
- Excellent result!
  - But about **50%** of the inefficiency due to situations involving the **human intervention**

# Error Management in TDAQ



# ONLINE RECOVERY & DIAGNOSTIC AND VERIFICATION SYSTEM (DVS)

# Online Recovery and DVS

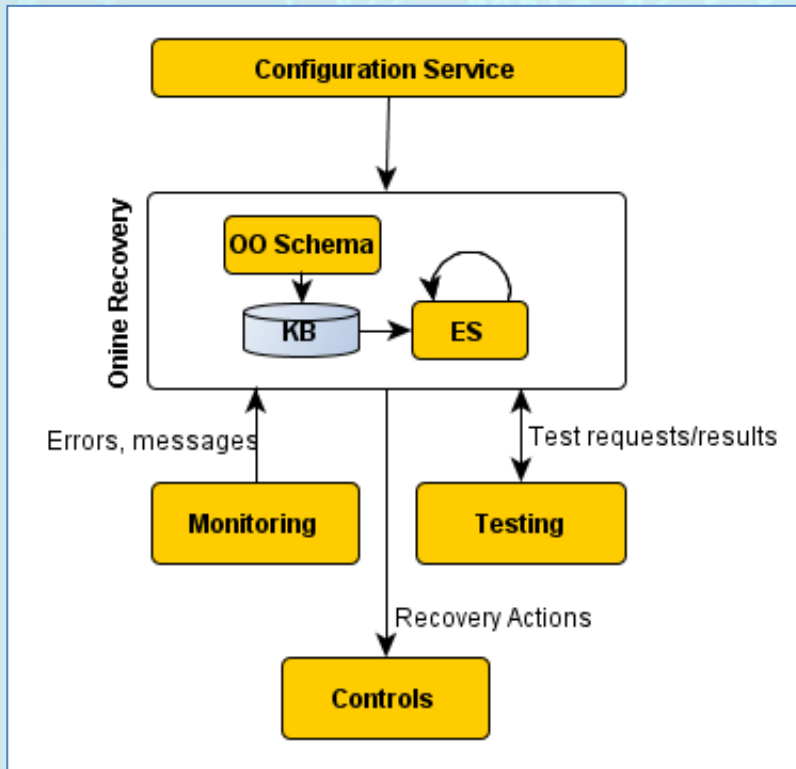


- **Online Recovery**
  - Analyze and recover from errors during the data taking
    - One global server dealing with system-wide errors and procedures
    - Local units handling errors that can be dealt with at a sub-system level
- **Diagnostic and Verification System (DVS)**
  - Asses the correct functionality of the system
  - Detect and diagnose eventual problems
- **Test Manager**
  - Framework allowing to develop and configure tests for any component in the system
- **Expert System**
  - Based on the CLIPS toolkit
  - “if-then” rules

# CLIPS

- **Originally developed by NASA**
- **Available as open-source**
- **Stand-alone application or embeddable as a library**
- **Different programming paradigms**
  - IF-THEN rules and a forward-chaining inference engine
  - Object oriented constructs
    - “COOL” language
  - Traditional algorithmic constructs
- **Rete algorithm**

# Online Recovery in Action



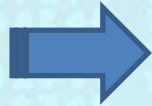
- The system is described and configured via the Configuration Service
  - Object-oriented schema
- The set of objects for the actual configuration is loaded by the Expert System
  - Using the COOL object-oriented language provided by CLIPS
- The Expert System engine uses information coming from errors, messages and tests to match the loaded rules
- The KB is parsed at run-time
  - Easy customization of recovery procedures

# DVS

- **The DVS is a framework that allows to**
  - Configure a test for any component in the system
  - Have a graphical representation of the testable components and of the test results
    - Via an user-friendly GUI
  - Automate testing of the system
  - Provide the operator with diagnosis and recovery advices in case of failures
  - Add knowledge for testing sequences and error diagnostics

# Recovery Scenarios

Simple recovery  
action for a  
dead application

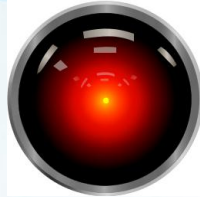


```
if
system state is running, and
application App1 status is absent, and
application App1 has supervisor S1, and
application App1 membership in
then
notify S1 ignore App1
set membership App1 out
```

- **Recoveries cover a wide range of possible scenarios**
  - Simple local actions
    - Restarting a dead application
    - Ignore problems from non-critical applications
  - System wide actions
    - Disable (and eventually re-enable) a busy read-out channel without stopping the run
    - Re-configure a sub-system during the data taking

# THE DAQ ASSISTANT

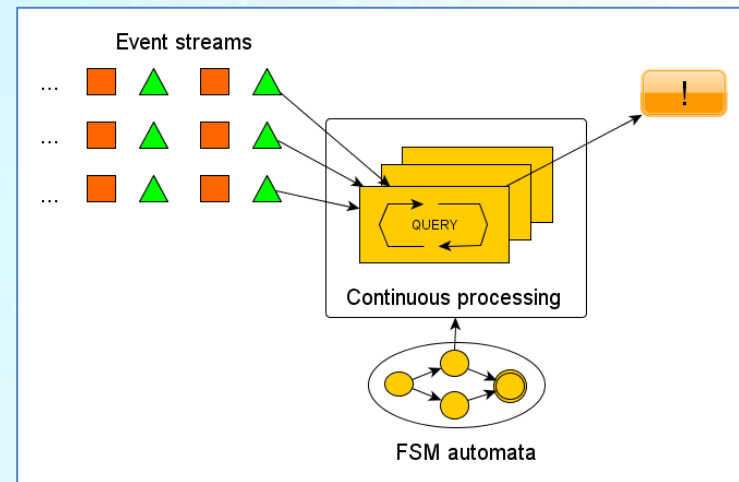
# The TDAQ Assistant



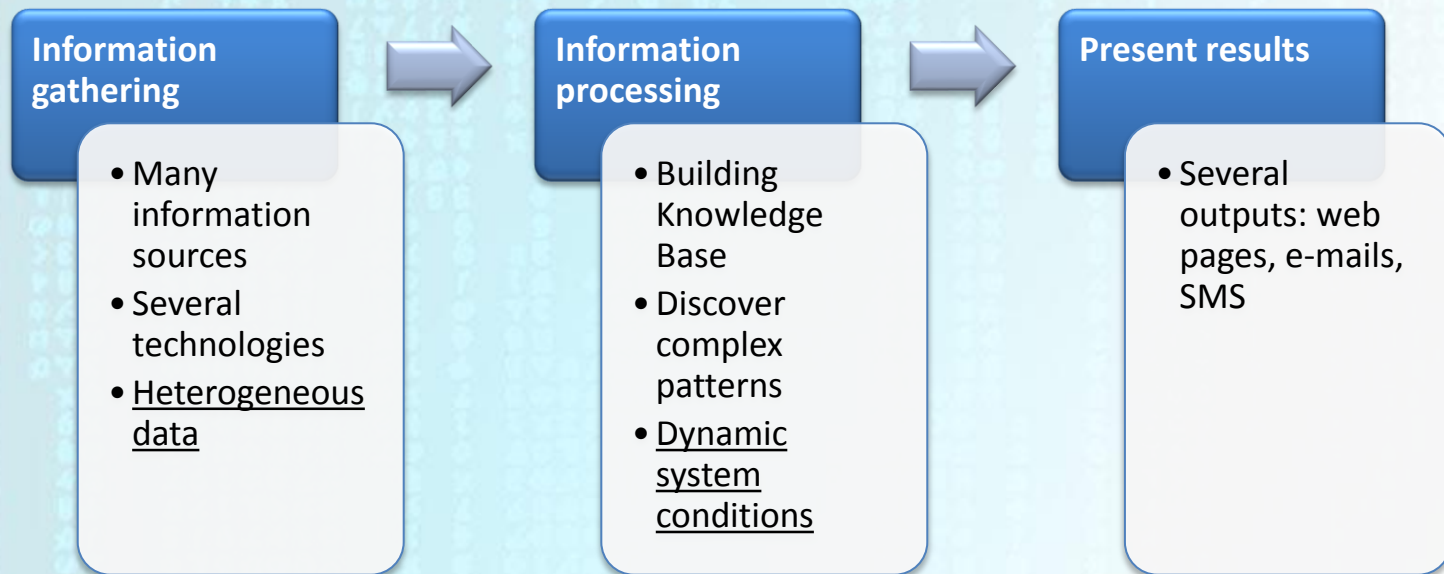
What	Aim	Objectives
<ul style="list-style-type: none"><li>• A tool meant at guiding the operator in his daily work</li><li>• Diagnosing problematic situations and suggesting action to take</li><li>• Remind the operator he should (not) do something</li></ul>	<ul style="list-style-type: none"><li>• Reduce and simplify shifter tasks</li><li>• Help shifters with more detailed and pertinent information</li><li>• Be more efficient, avoid repetition</li><li>• Formalize knowledge from experts</li></ul>	<ul style="list-style-type: none"><li>• Automate checks and controls in real-time</li><li>• Process and analyze heterogeneous streams of information</li><li>• Receive instructions from TDAQ experts on what to do and how to react</li><li>• Promptly notify operators of problems and failures</li></ul>

# Complex Event Processing

- **A set of technologies to process events and discover complex patterns among streams of events**
  - Used in financial analysis, wireless sensor networks, business process management
- **A cross between Data Base Management System and Rule Engines**
- **Main characteristics**
  - Continuous stream processing
  - Support for time/size windows, aggregation and grouping events
  - SQL-like pattern languages
    - Augmented with constructs to express event relationships (time, cause and aggregation)
    - Streams replacing tables in a continuous evaluation model

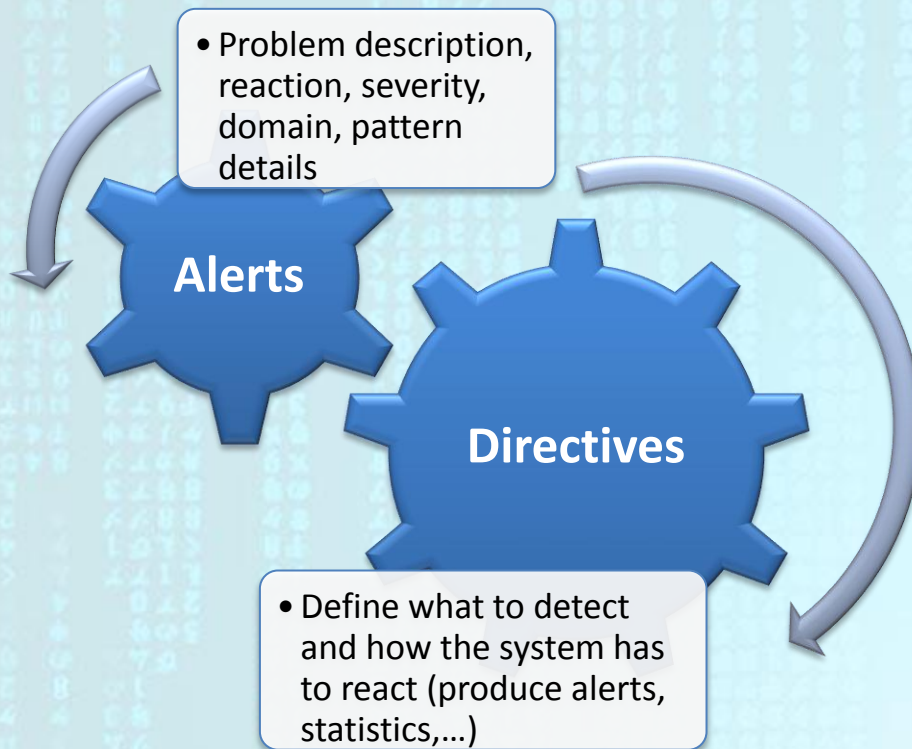


# Challenges in TDAQ



**... and all with thousands of information updates per second!**

# Directives and Alerts



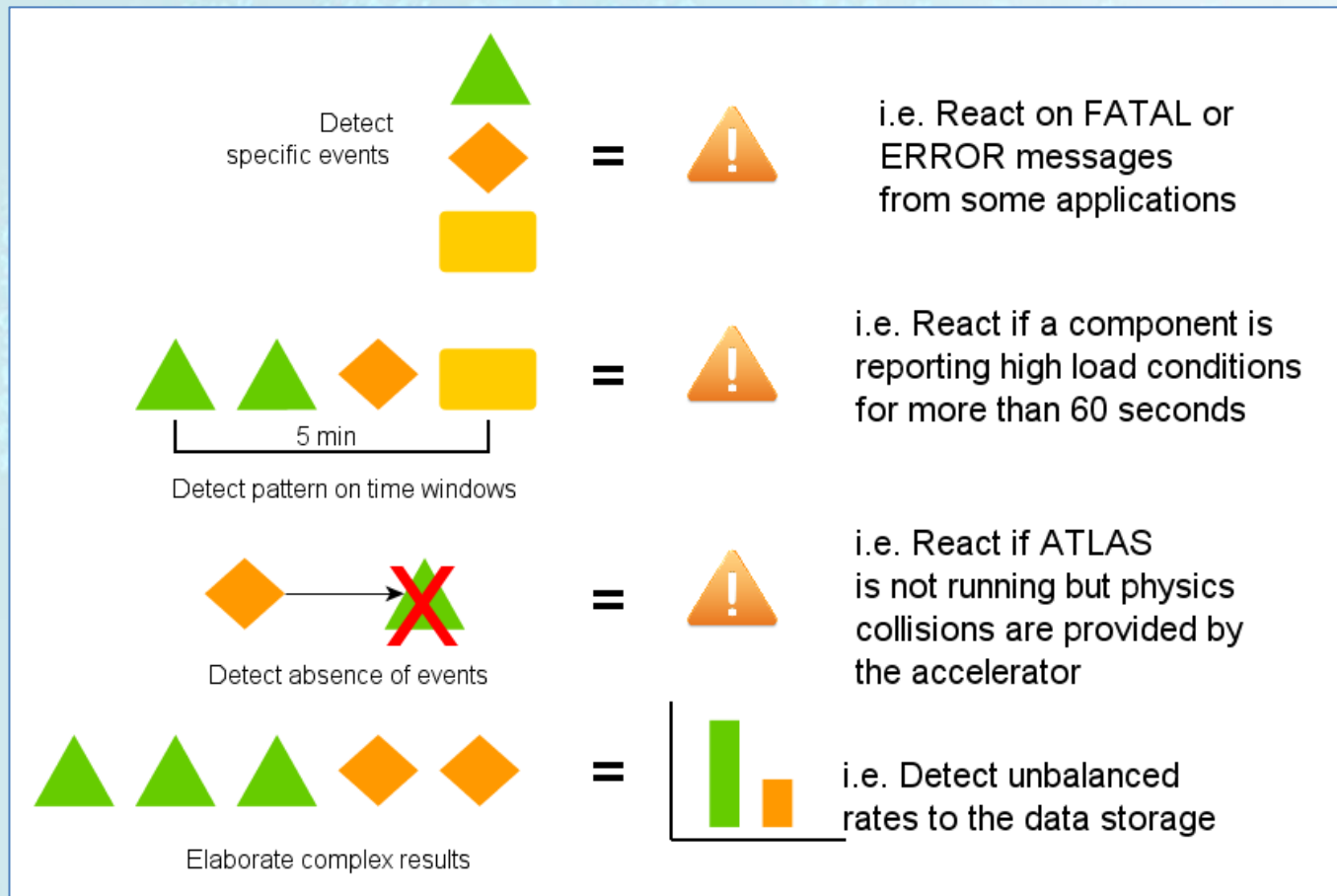
- **Directives**

- Encode the knowledge from experts
- XML structured KB
- Can be modified at run time via a web-based admin interface

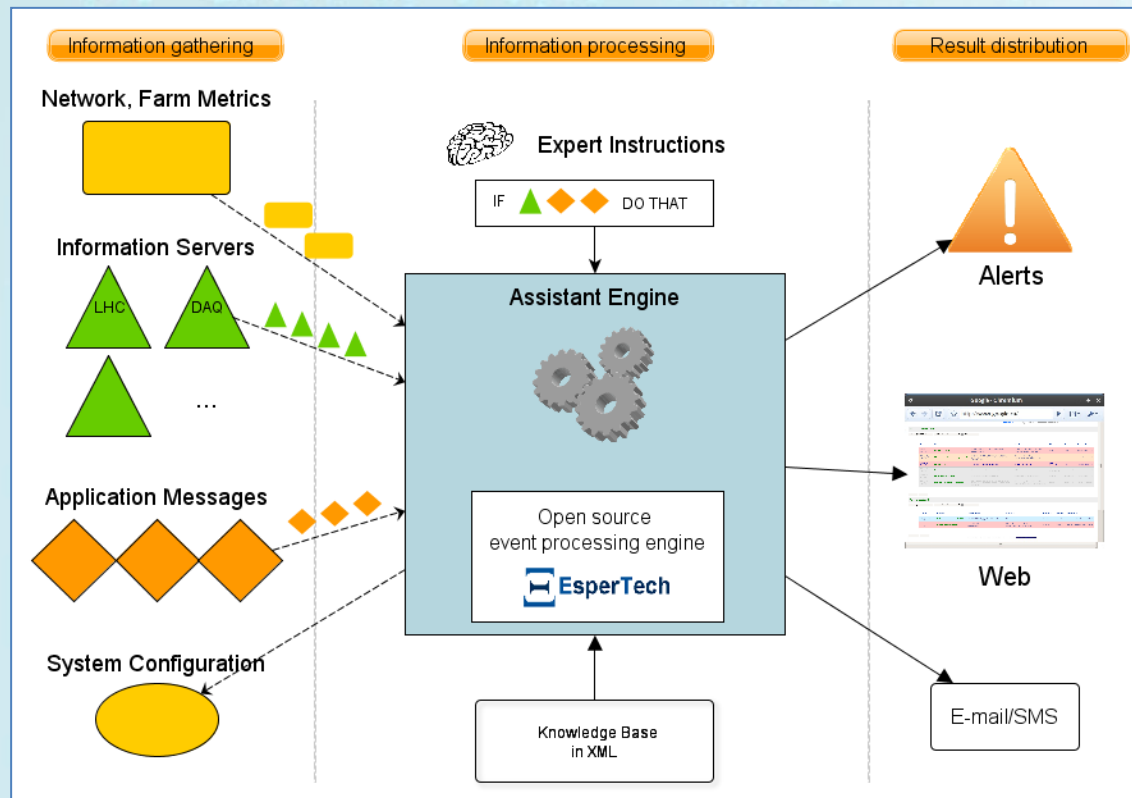
- **Alerts**

- Effective and timeliness notification
- Intelligent processing
  - Thanks to CEP the number of false-positive situations is drastically reduced
- Carry all the information needed for debug and fault diagnosis

# Detecting Patterns

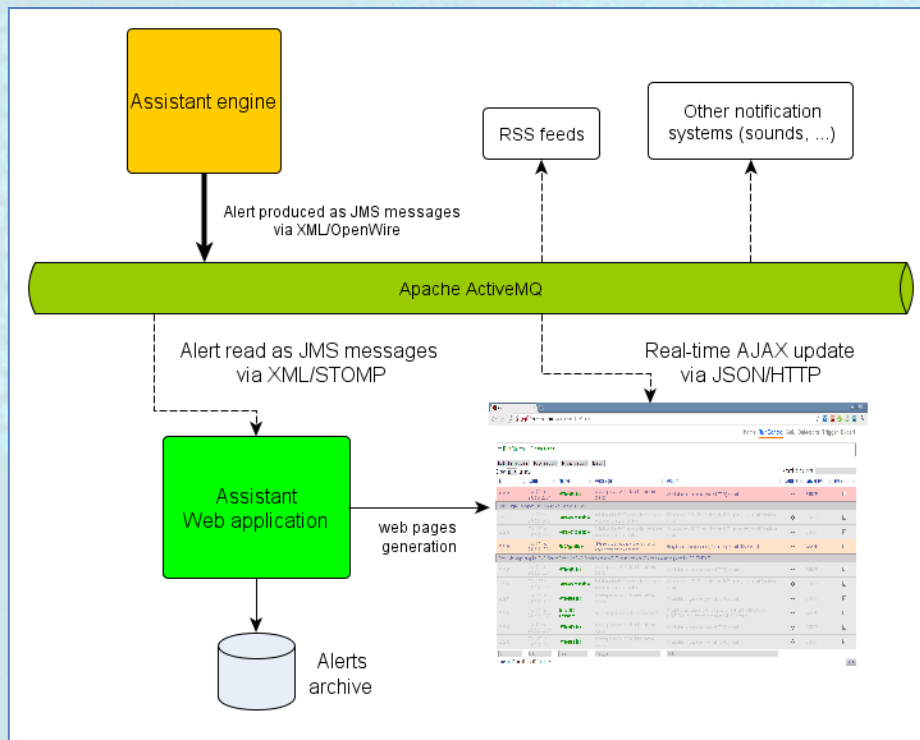


# Architecture



- Data gathered and feed into the **engine**
- EPL statements (**from directives**) are evaluated against data (**continuous query**)
- Generating alerts, notifications, statistics as soon as incoming events meet the constraints of the **pattern**

# Web-Based Visualization



- **Message driven alert distribution**
  - Based on Apache ActiveMQ
- **Web page for interactive visualization of alerts**
  - Alerts grouped per categories/user preferences
  - User interaction
    - Mark alerts as read when the problem is solved
    - Mask alerts
  - Alert history
- **Django project with some SQLite and jQuery goodies**

# Conclusions

- **Effective monitoring, fault diagnosis and automation of recovery procedures have shown to really help improving the ATLAS data taking efficiency**
- **The DAQ Assistant is in production since June 2011**
  - A message-driven architecture and CEP techniques allowed to build an intelligent and automated monitoring tool
  - Used to assist the data acquisition operators
    - From simple reminders to the detection of complex error conditions
    - Shift crew reduced by one unit (DAQ shifter)
  - Integrated with the EMS system in order to trigger automated recovery actions
- **Looking forward to a successful 2012 for ATLAS**
  - And the goal is to always improve the data taking efficiency