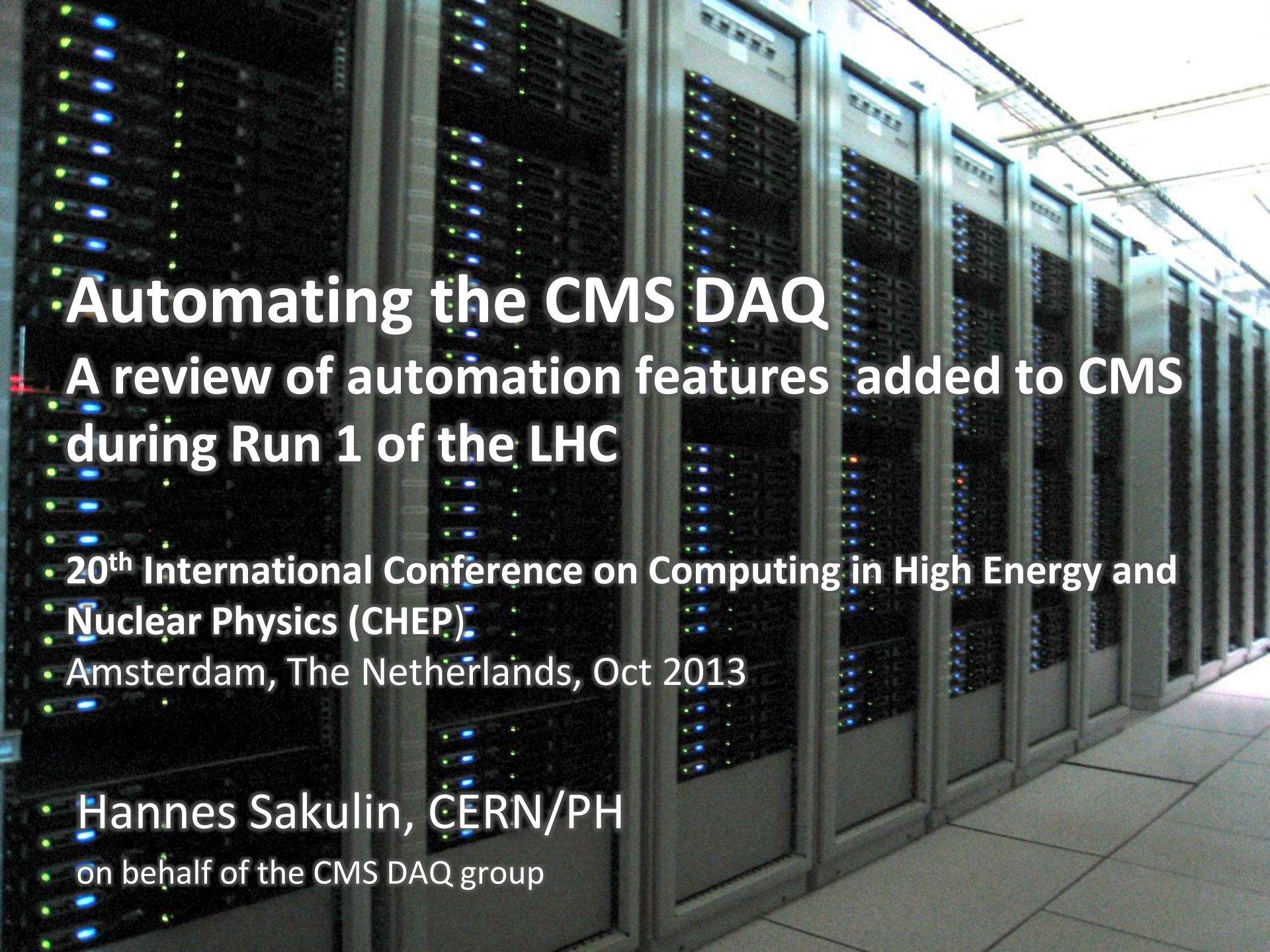


Automating the CMS DAQ

A review of automation features added to CMS during Run 1 of the LHC

20th International Conference on Computing in High Energy and Nuclear Physics (CHEP)
Amsterdam, The Netherlands, Oct 2013

Hannes Sakulin, CERN/PH
on behalf of the CMS DAQ group

Overview

- Automation Features added to CMS DAQ over Run 1 of the LHC added gradually
  - as we learned from operation
  - as new requirements emerged
  - as new failure scenarios became apparent
- These features were needed
  - to operate CMS with high data taking efficiency
  - to operate CMS with a non-expert crew
  - to reduce the work load for on-call experts

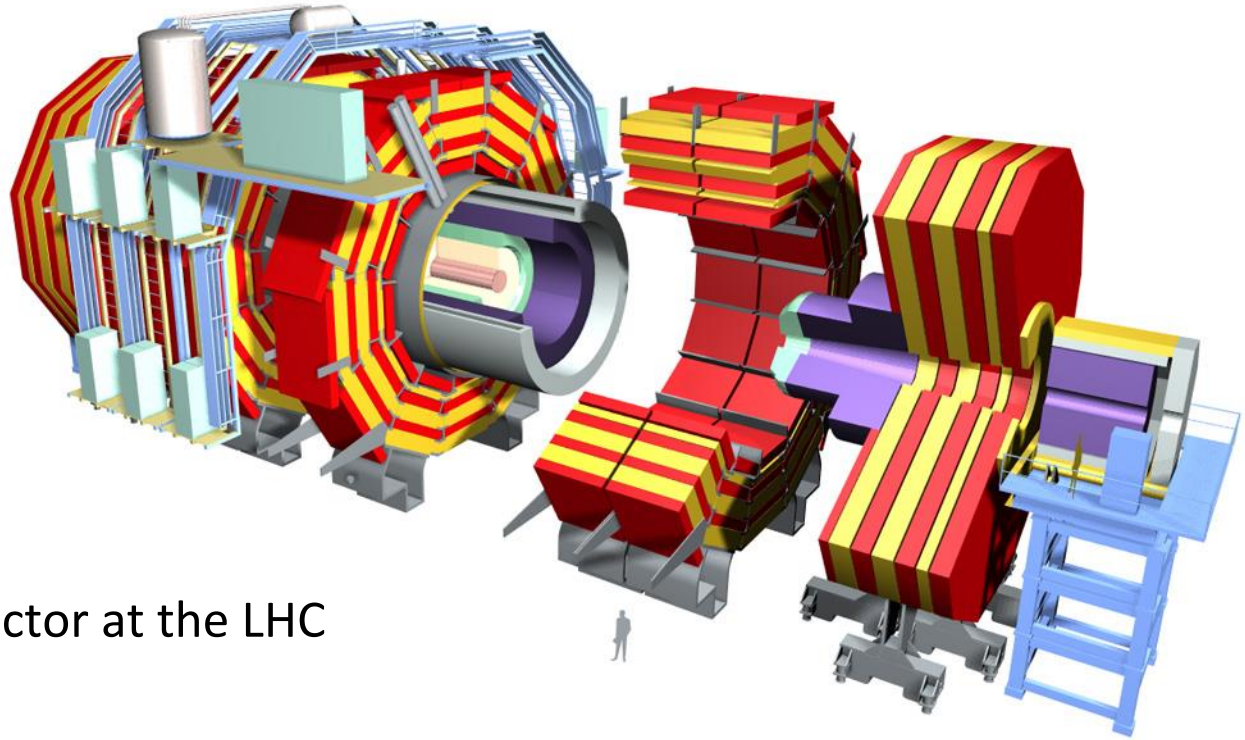
Definition: Automation

- In principle, anything done by the experiment's online software ultimately falls into the domain of automation

- For this talk we define **automation** to be
  - The automation of routine tasks that otherwise would be performed by the operator or on-call expert (or which were initially performed by the operator)
  - Automatic diagnosis
  - Automatic recovery from known error conditions

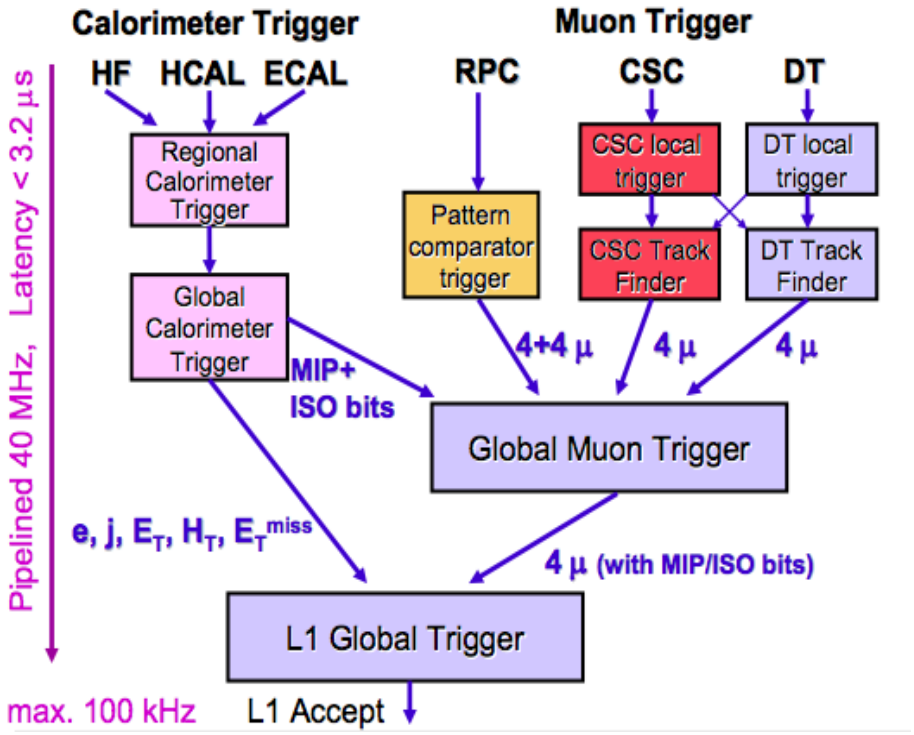
System Overview

The Compact Muon Solenoid Experiment

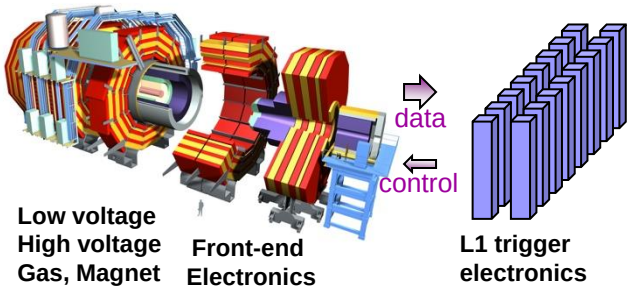


- General-purpose detector at the LHC
- 12 sub-systems
- 55 million readout channels

The first-level trigger

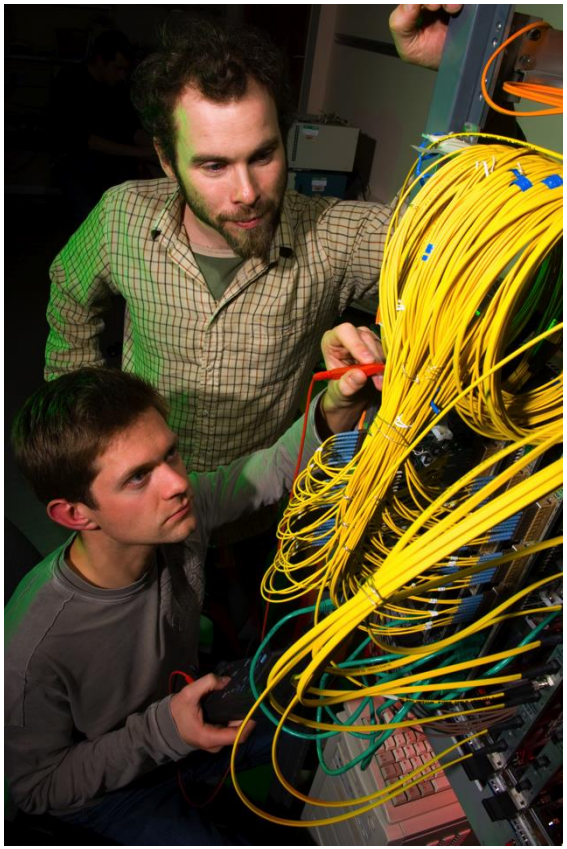


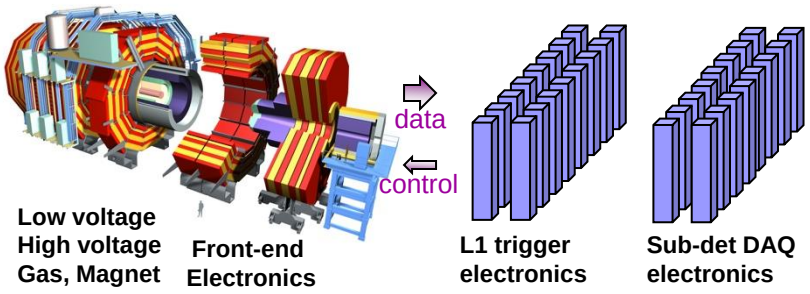
40 racks of custom electronics



Sub-detector DAQ electronics

- Front-end Controllers
- Front-end Drivers
  - Delivering data to the central DAQ
  - 700 links

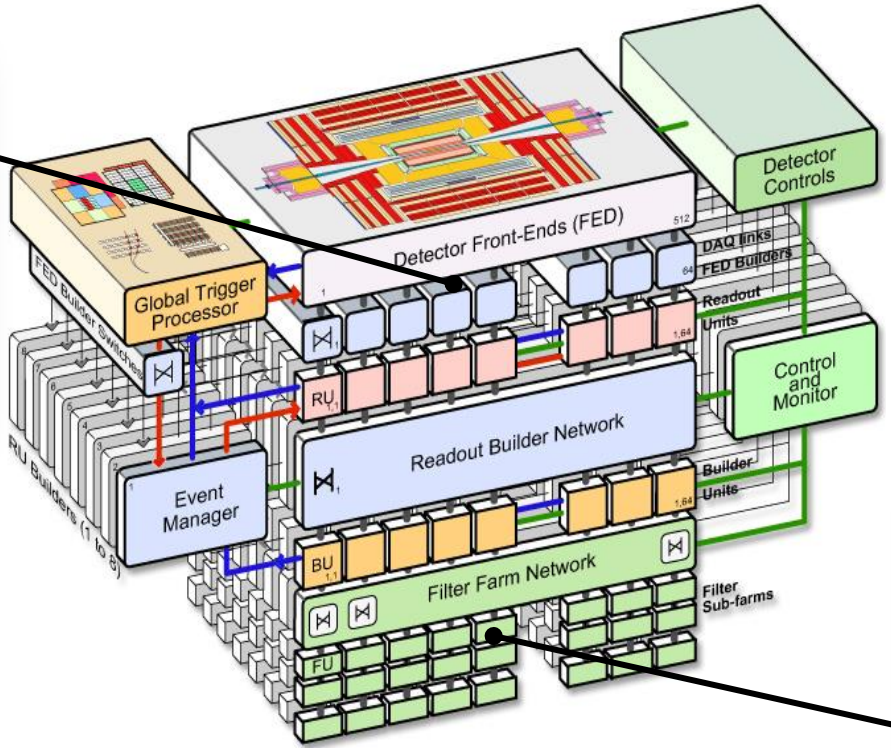


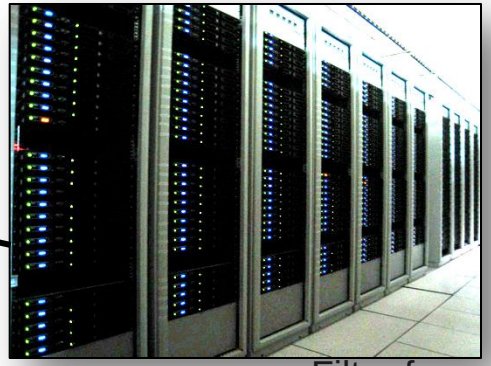


The central DAQ

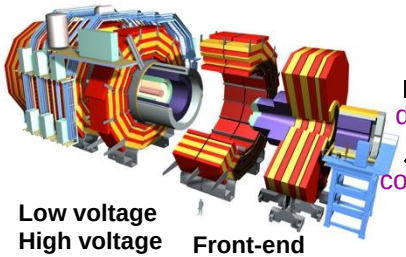
- Builds events at 100 kHz, 100 GB/s
  - 2 stages: Myrinet Gigabit Ethernet
  - 8 independent event builder / filter slices
- High level trigger running on filter farm
  - ~1200 PCs
  - ~13000 cores (2012)

Frontend Readout Links





Filter farm



Low voltage
High voltage
Gas, Magnet

Front-end Electronics

data

control

L1 trigger electronics

Sub-det DAQ electronics

Central DAQ electronics

data

Central DAQ farm, High Level Trigger & storage

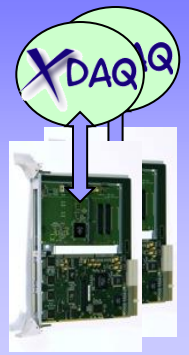
The online software

XDAQ Framework – C++

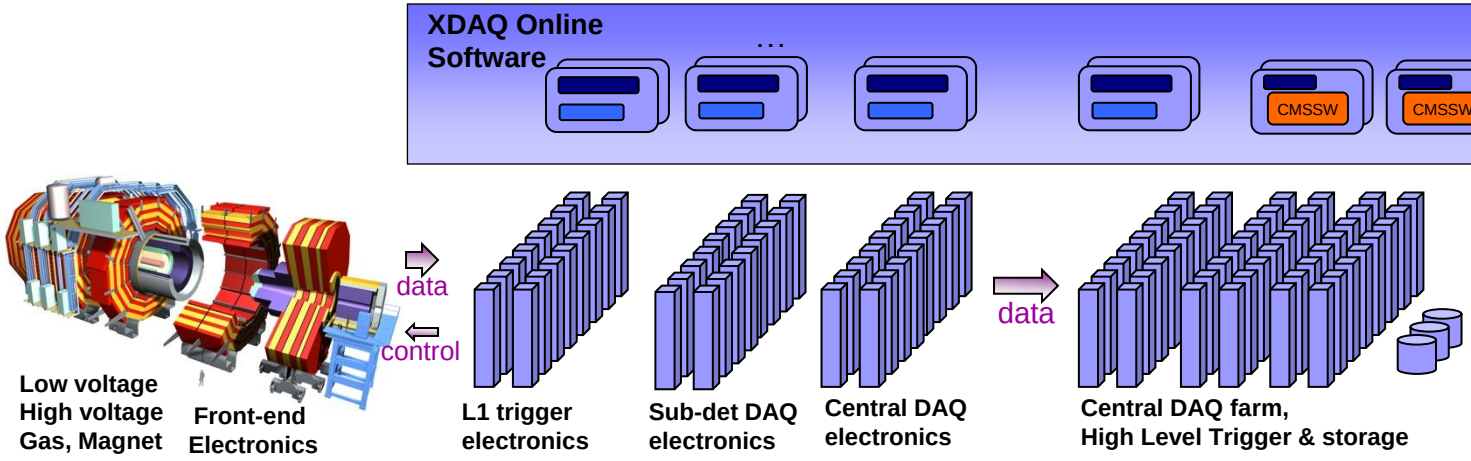
XDAQ applications control **hardware** and handle **data flow**

Hardware Access, Transport Protocols,
XML configuration, SOAP communication,
HyperDAQ web server

XDAQ Application



~20000 applications to control



The online software

Run Control System – Java, Web Technologies

Defines the control structure

GUI in a web browser

HTML, CSS, JavaScript, AJAX

Run Control Web Application

Apache Tomcat Servlet Container

Java Server Pages, Tag Libraries,

Communication

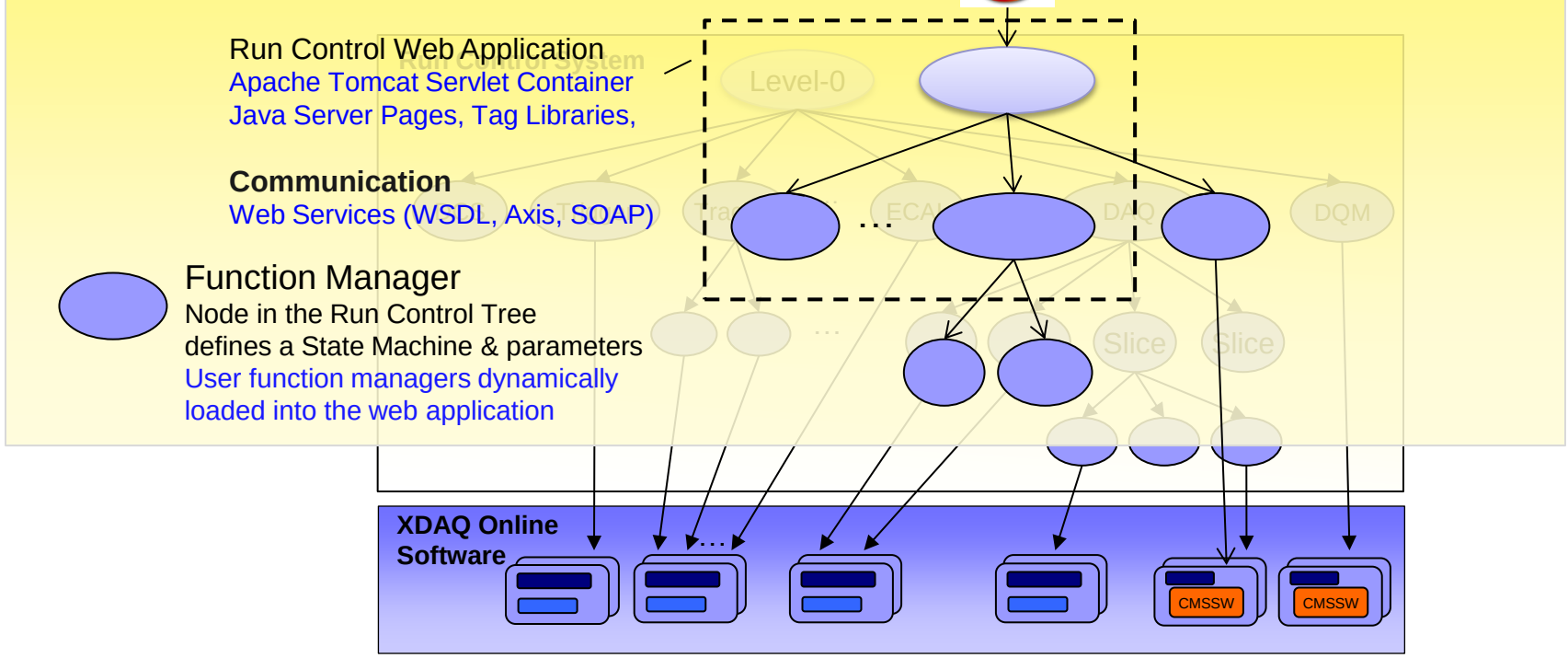
Web Services (WSDL, Axis, SOAP)

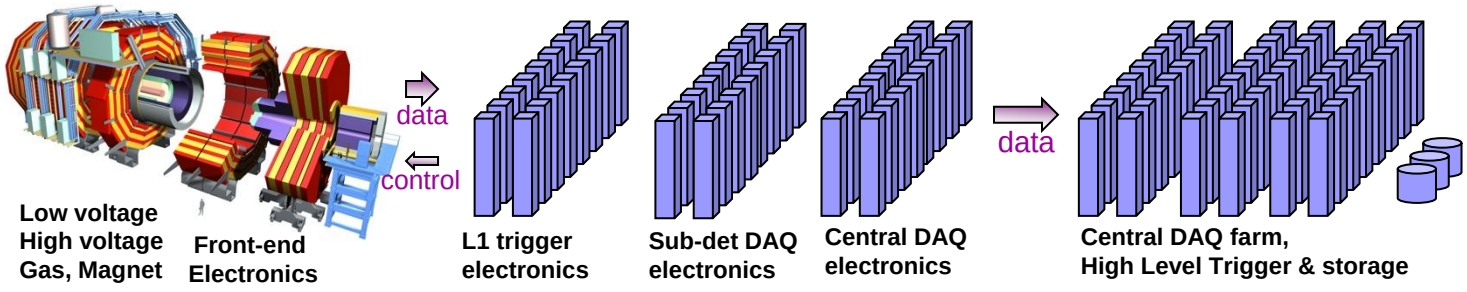
Function Manager

Node in the Run Control Tree

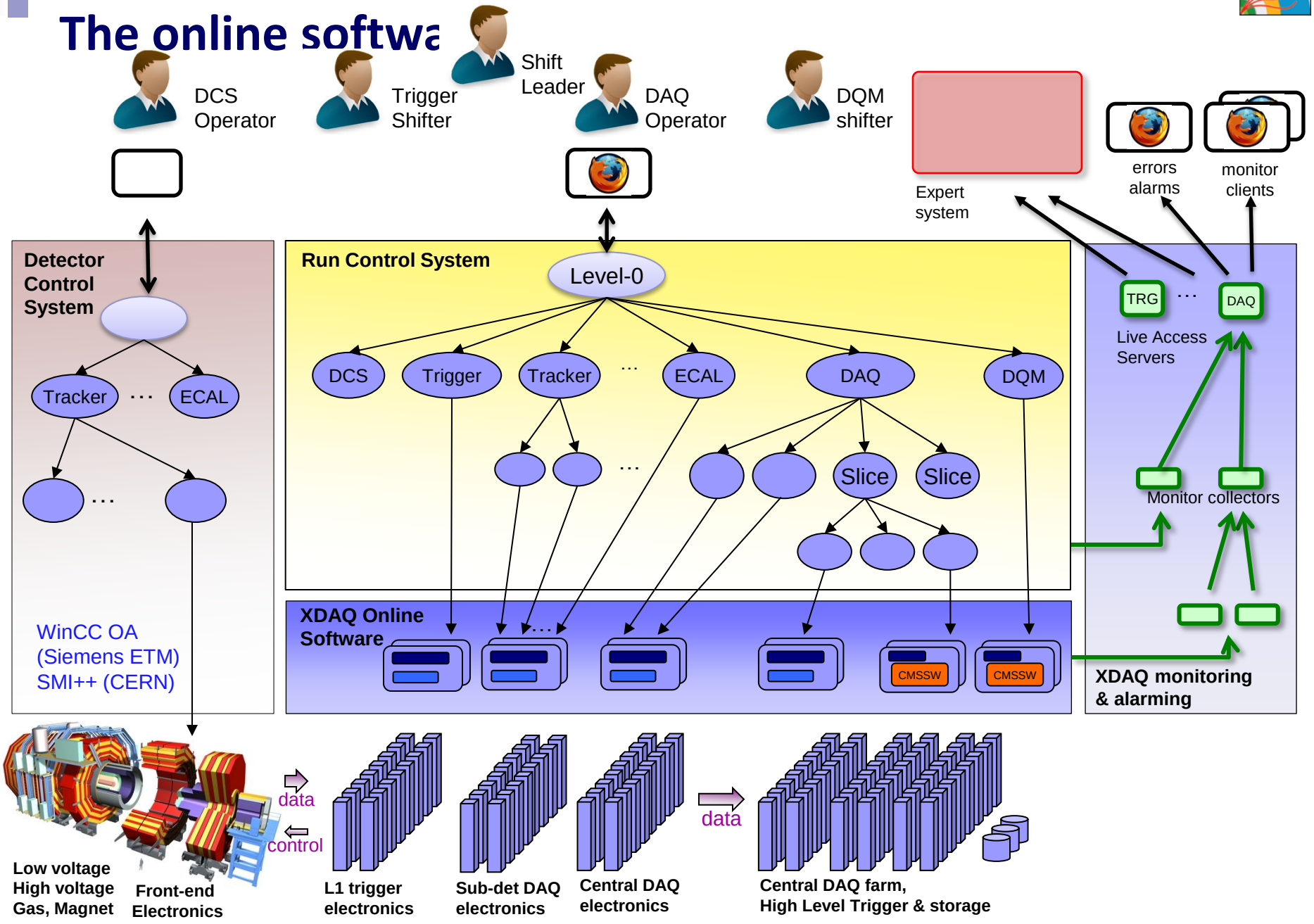
defines a State Machine & parameters

User function managers dynamically loaded into the web application





The online softwa



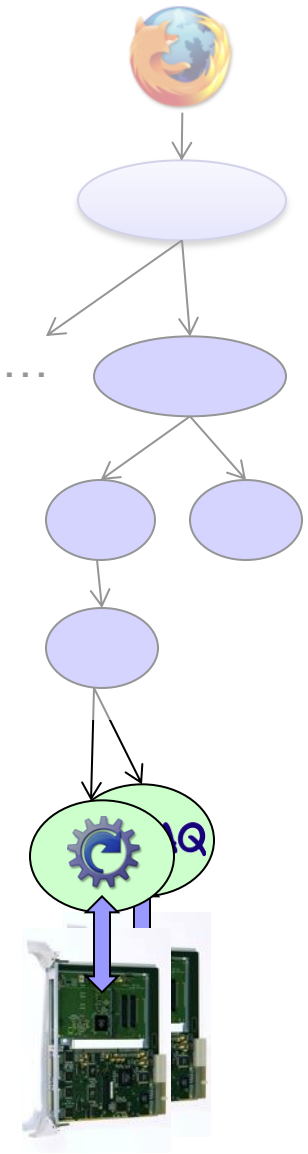
Adding automation

Principles of Automation in CMS

- Integrated into the control tree
  - As **local** as possible
    - Control nodes react to state of their child nodes
    - Detailed knowledge stays encapsulated
    - Propagate information only as far up as necessary
  - Implemented in the Run Control & XDAQ frameworks
  - Distributed
  - Parallel
- In addition – external automation
  - For recovery that changes the system structure
    - E.g. re-computation of the configuration after hardware failure

Integrated Automation
a) at the XDAQ layer

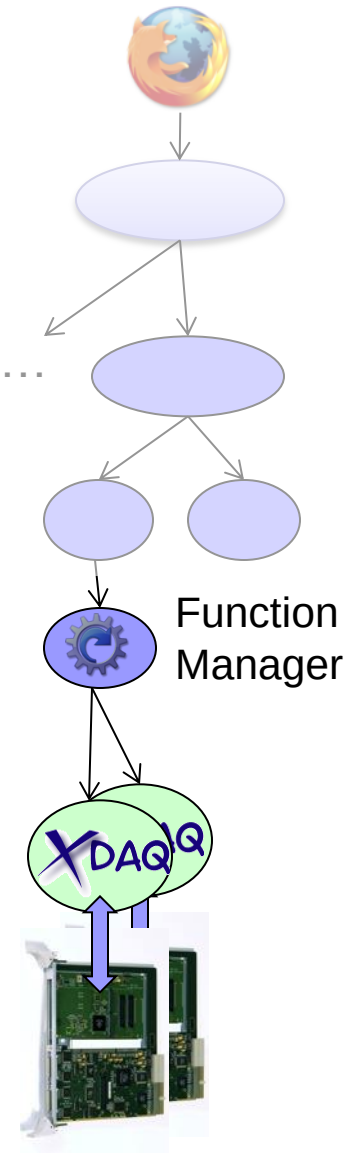
Automation in the XDAQ layer



- XDAQ application detects problem and takes corrective action
- How
  - Using framework functionality (work loops …)
  - Full flexibility of C++ (e.g. Unix signals)
- Example
  - Event Processors (High-Level Trigger)
    - Child processes are forked to process in parallel
      - reduces memory footprint (Copy-On-Write)
      - Reduces configuration time
    - In case a child processes crashes, a new one is created

Integrated Automation
b) at the Run Control layer

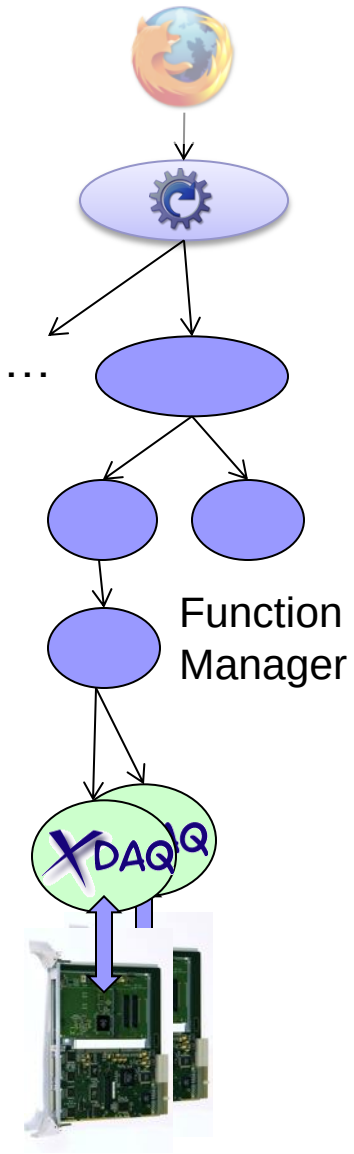
Automation in Run Control



- How
  - Function Managers receive **asynchronous updates** from their children
    - State updates
    - Monitoring data
  - Function Managers define **event handlers** (Java) reacting to these notifications
  - Full flexibility of Java

- Examples
  - Exclude crashed resources from control
  - Recover flush failure
    - When stopping a run, sometimes a DAQ slice cannot be flushed
    - The corresponding function manager detects this and locally recovers this slice

Automation in the Run Control Top-level Node



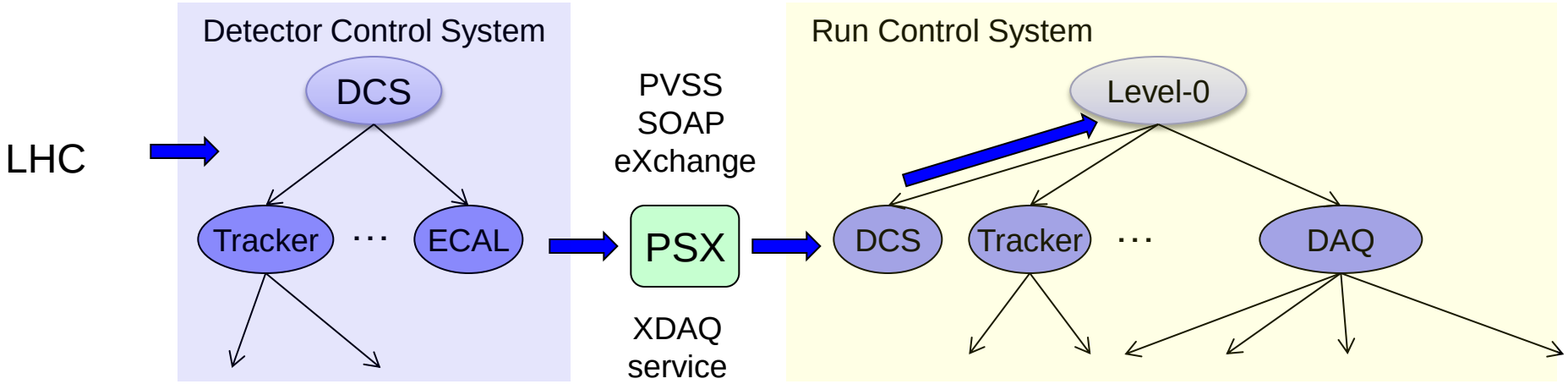
- For actions that need coordination between multiple sub-systems
  - Example 1: Automatic actions in the DAQ in response to condition changes in the DCS and the LHC
  - Example 2: Synchronized error recovery
  - Example 3: Automatic configuration selection based on LHC mode

see following slides …

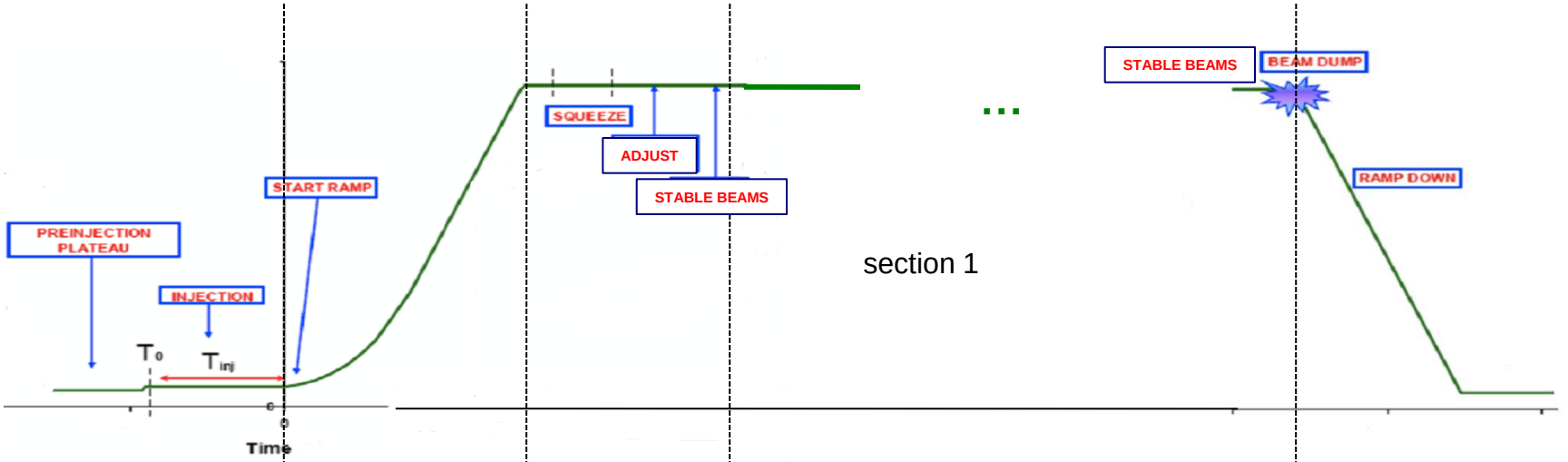
Automation Example 1: Automatic reaction to LHC and DCS

DAQ actions on LHC and DCS state changes

- Extensive automation has been added to DCS
  - Automatic handshake with the LHC
  - Automatic ramping of high voltages (HV) driven by LHC machine and beam mode
- Some DAQ settings depend on the LHC and DCS states
  - Suppress payload while HV is off (noise)
  - Reduce gain while HV is off
  - Mask sensitive channels while LHC ramps …
- Initially needed to start new run with new settings – *very error prone*
- Since 2010: automatic new **run sections** driven by asynchronous state notifications from DCS/LHC



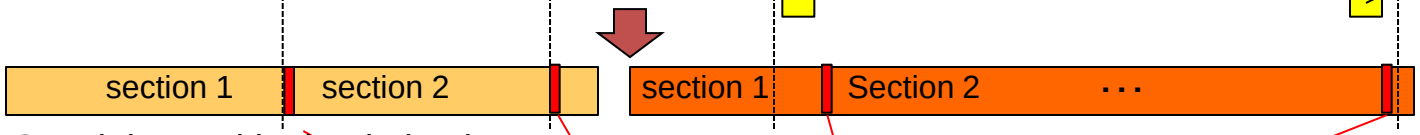
Now everything is driven by the LHC …



Start run at FLAT TOP.
The rest is automatic.

DCS ramps up tracker HV

DCS ramps down tracker HV



Special run with circulating beam

Automatic actions in DAQ :

ramp start	**ramp done**	**Tracker HV on**	**Tracker HV off**
Mask sensitive trigger channels	Unmask sensitive trigger channels	Enable payload (Tk) raise gains (Pixel)	Disable payload (Tk) reduce gains (Pixel)

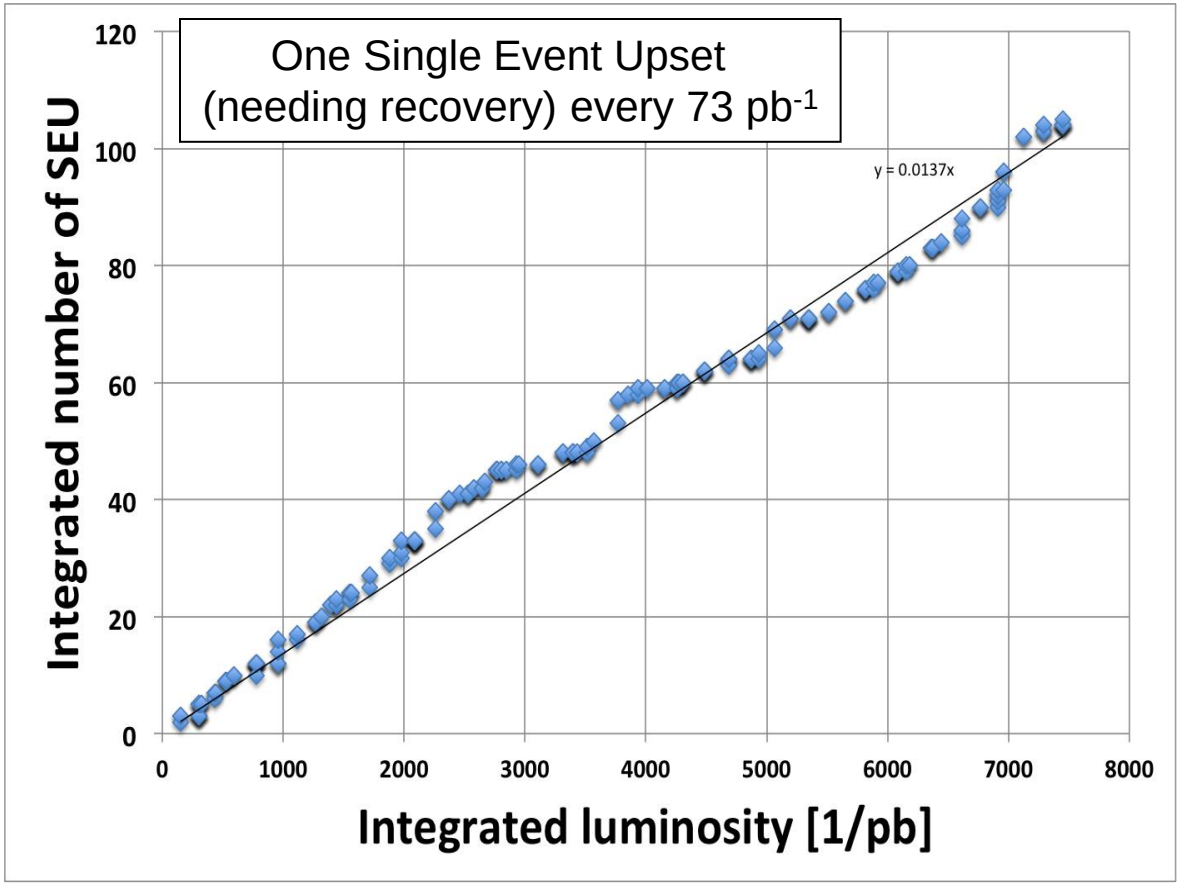
Example 2:
Automatic recovery from Soft Errors

Automatic soft error recovery

- With higher instantaneous luminosity in 2011 more and more frequent "soft errors" causing the run to get stuck
  - Proportional to integrated luminosity
  - Believed to be due to single event upsets

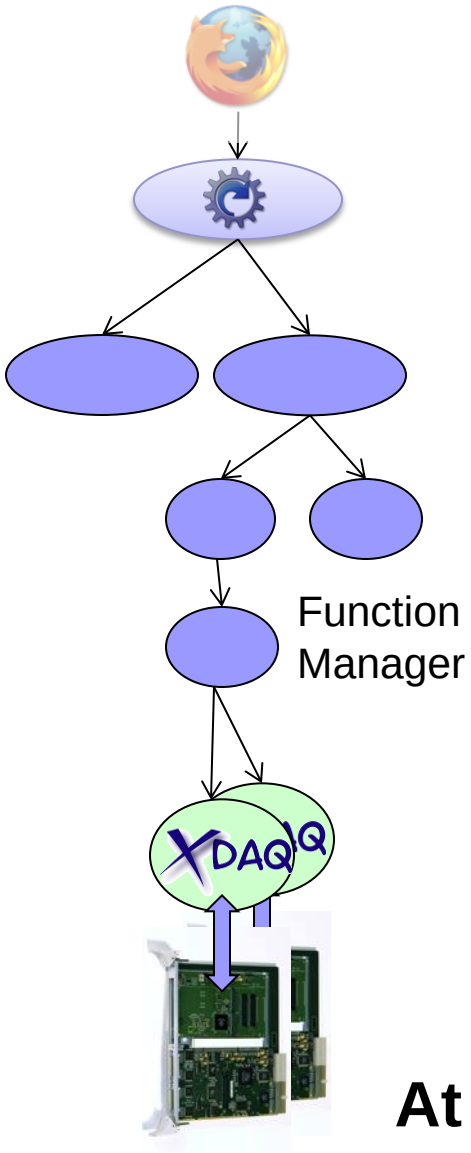
- Recovery procedure
  - Stop run (30 sec)
  - Re-configure a sub-detector (2-3 min)
  - Start new run (20 sec)

3-10 min down-time



Single-event upsets in the electronics of the Si-Pixel detector. Proportional to integrated luminosity.

Automatic soft error recovery



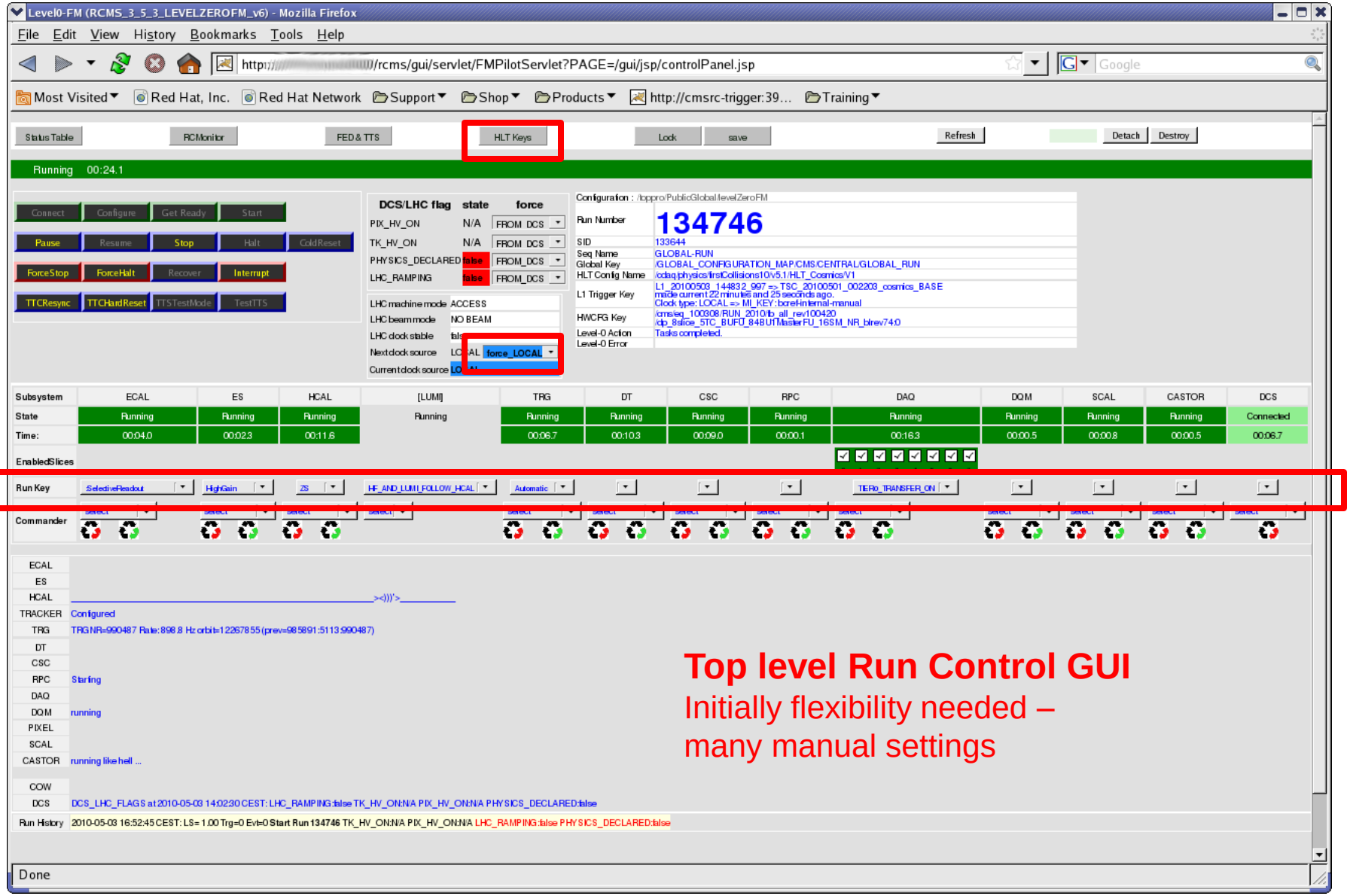
- From 2012, new automatic recovery procedure in top-level control node
  1. Sub-system detects soft error and signals by changing its state
  2. Top-level control node invokes recovery procedure
     a) Pause Triggers
     b) Invoke newly defined selective recovery transition on requesting detector
     c) In parallel perform preventive recovery of other detectors
     d) Resynchronize
     e) Resume

12 seconds down-time

At least 46 hours of down-time avoided in 2012

Example 3: Automatic Configuration Handling

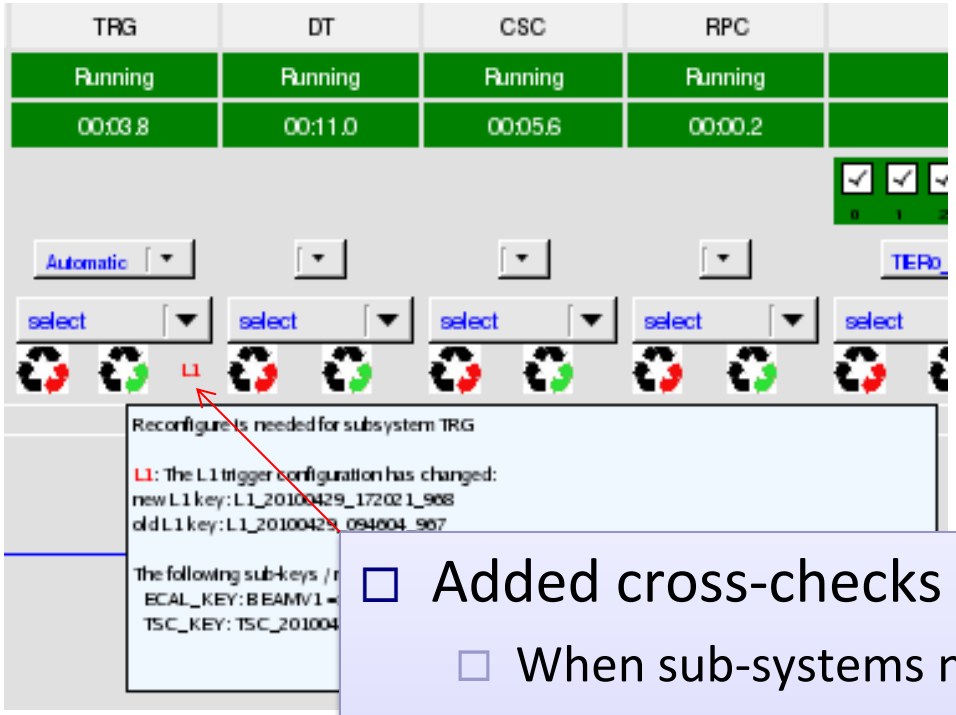
Lots of options to parameterize the configuration



Top level Run Control GUI

Initially flexibility needed –
many manual settings

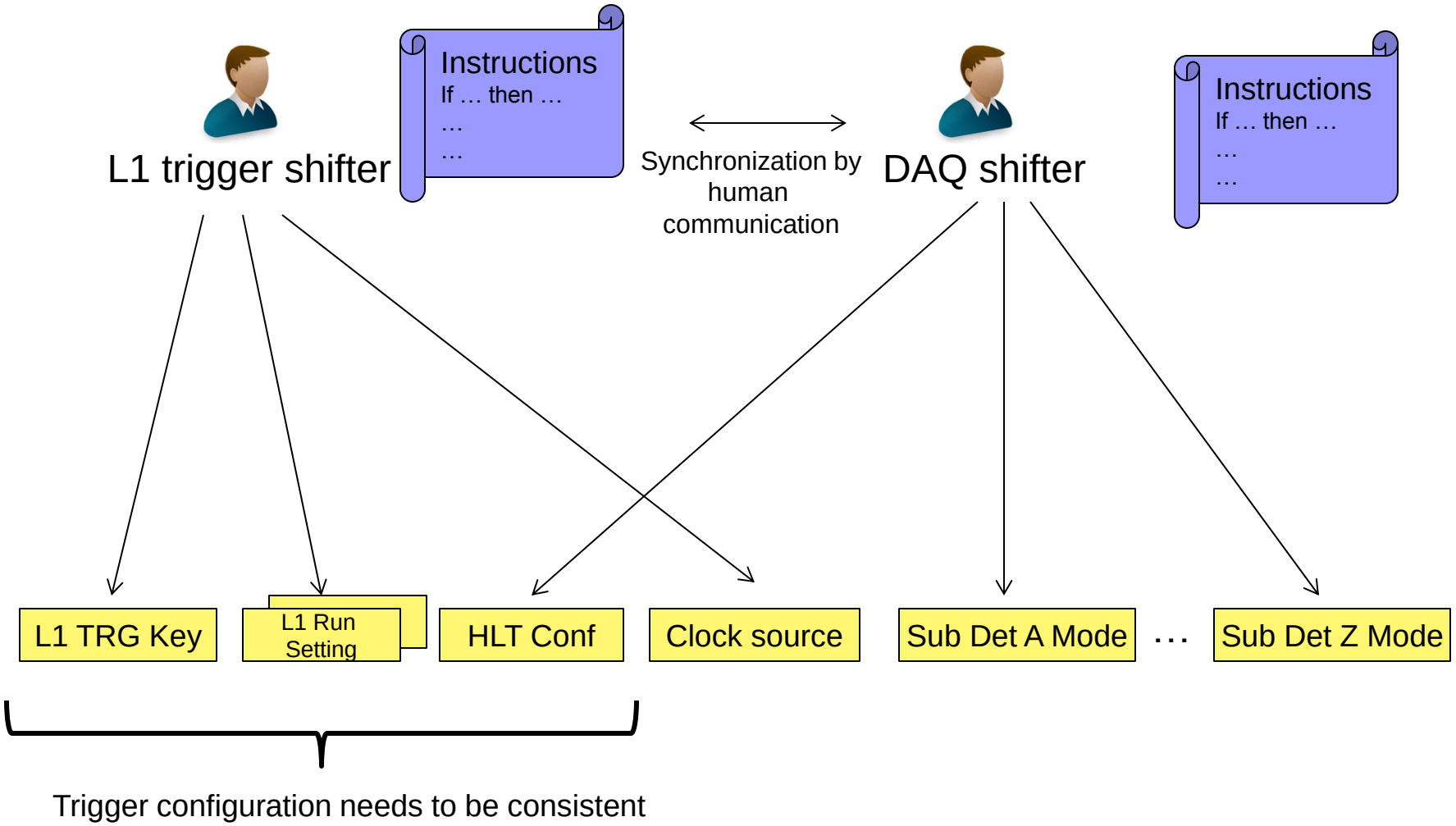
First added shifter guidance



- Added cross-checks and reminders
  - When sub-systems need to be re-configured
    - Because of clock instabilities
    - Because (External) configuration changes ..
    - Correct order of operations enforced
- Then simplified and automated …

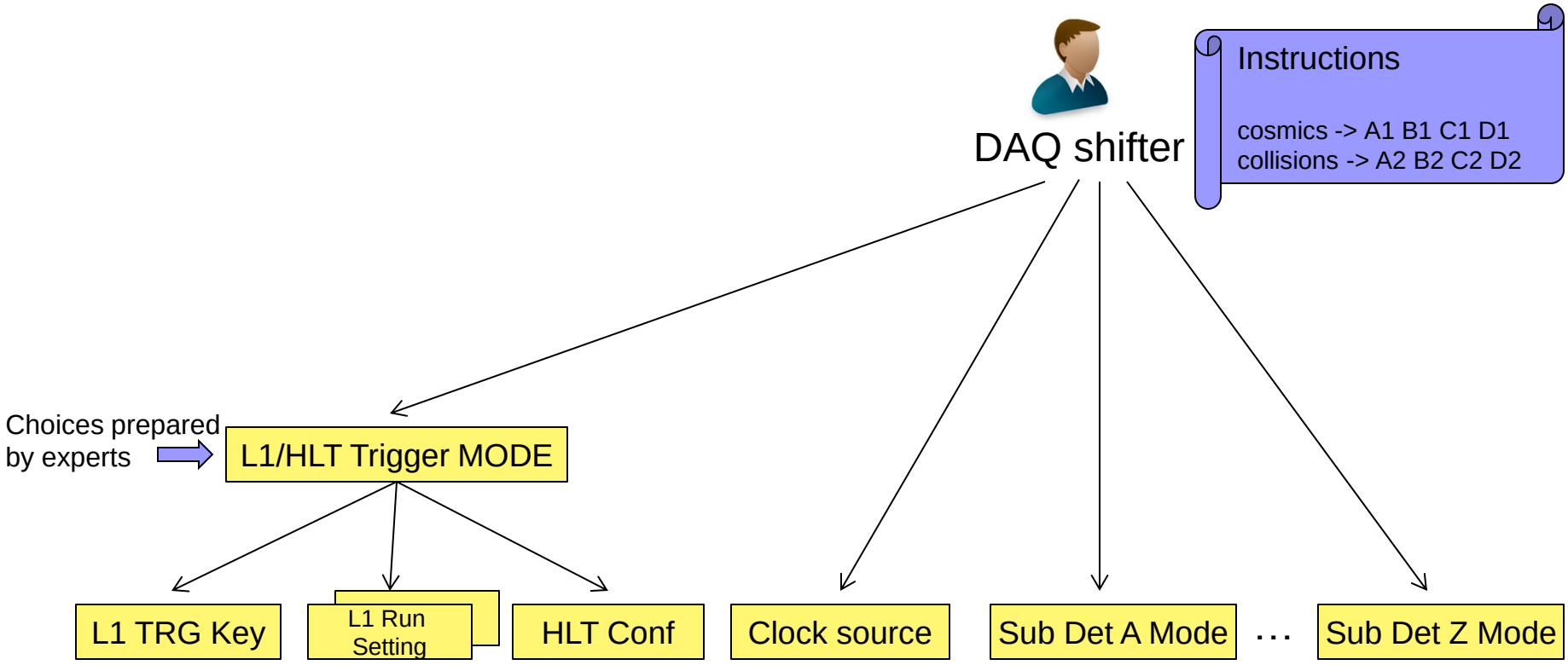
Configuration Handling (initial)

- 2009/2010



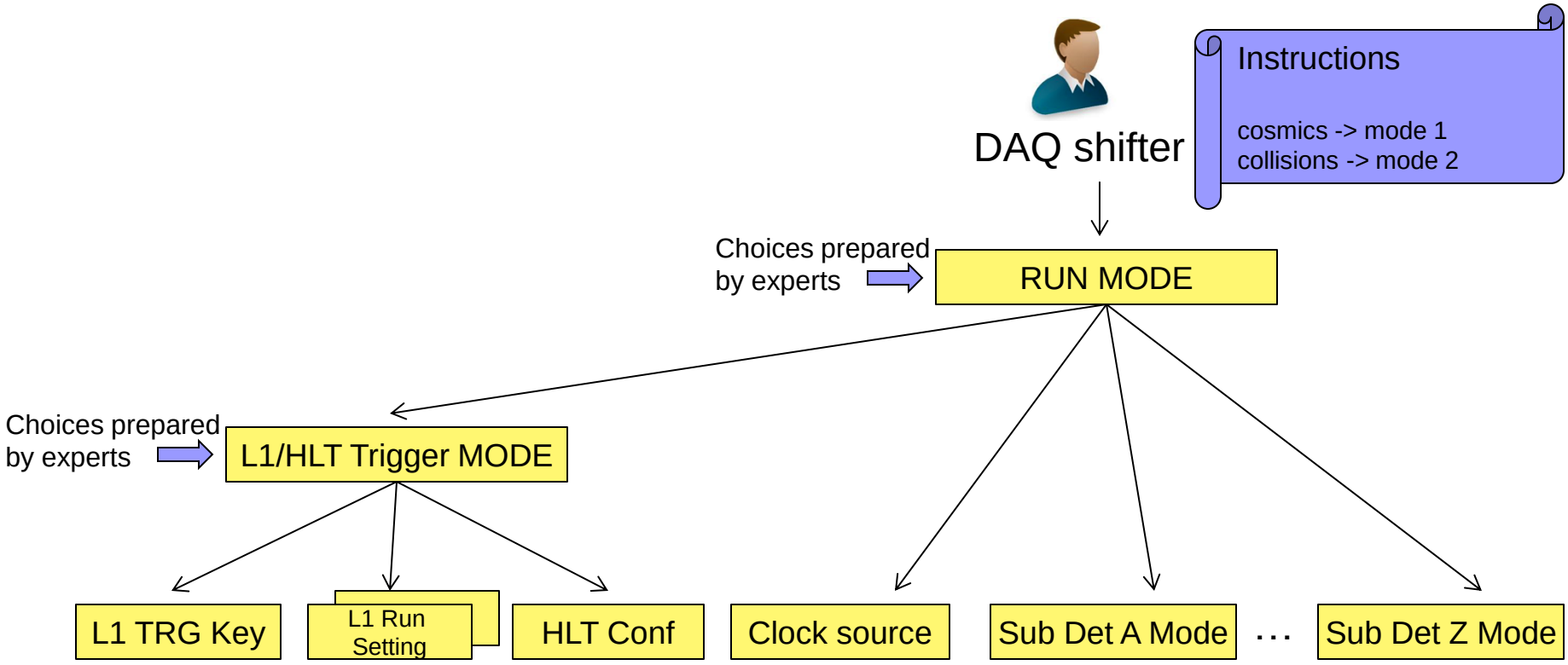
Configuration Handling (simplified)

- 2010



Configuration Handling (Run Mode)

- 2012

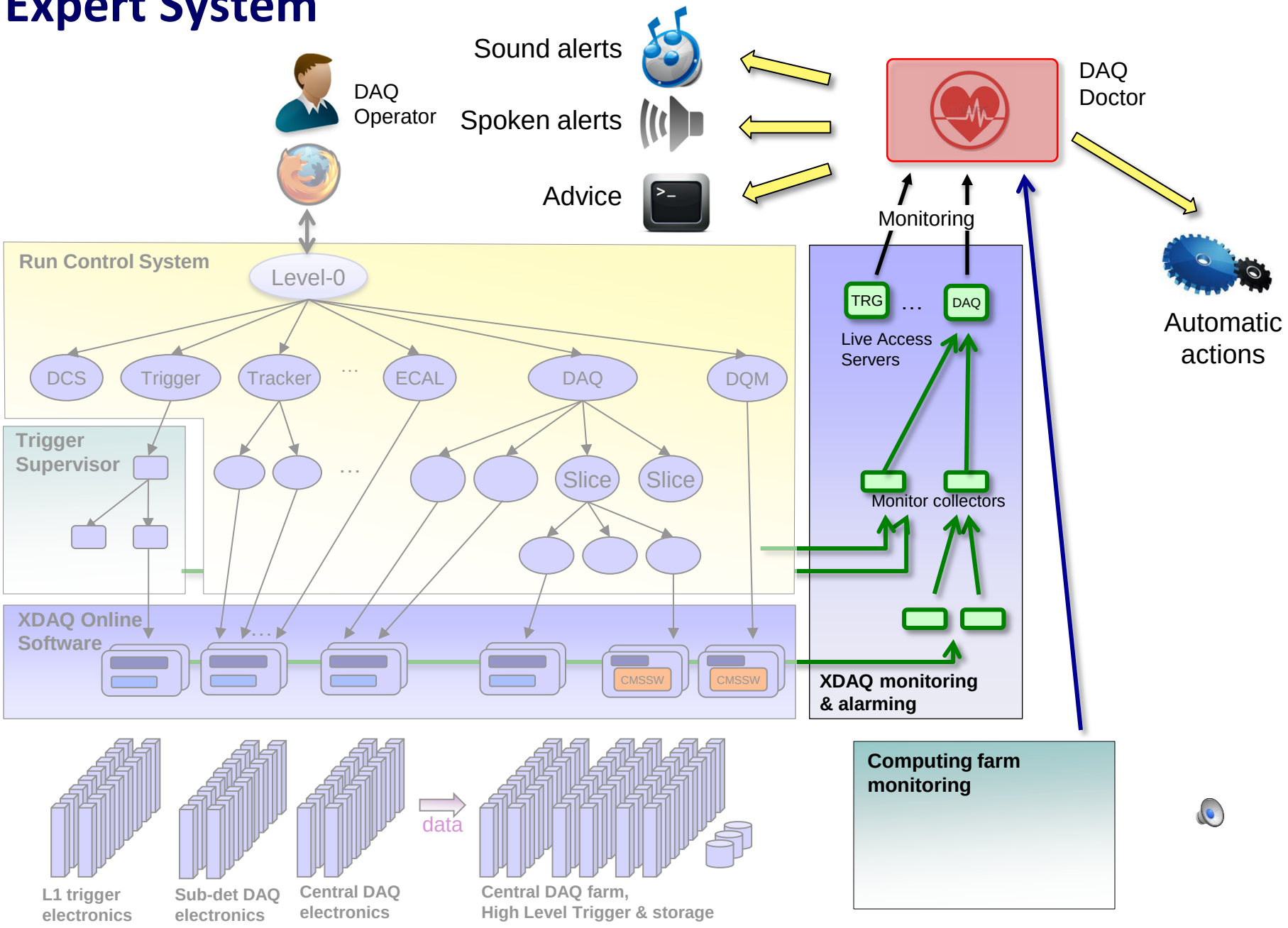


Configuration Handling (automatic)

- 2012

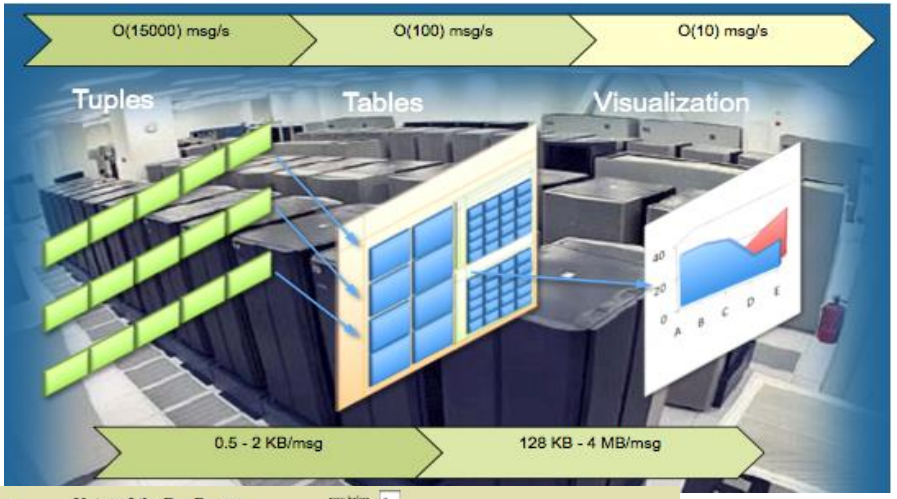
Central Expert System

Expert System



The DAQ Doctor

- Expert tool based on the same technology as Booking.com
  - High level scripting language (Perl)
- Generic framework & pluggable modules
- Detection of high level anomaly triggers further investigation
- Archive (web based)
  - All Notes
  - Sub-system errors
  - CRC errors
- Dumps (of all monitoring data) for expert analysis in case of anomalies



The DAQ Doctor

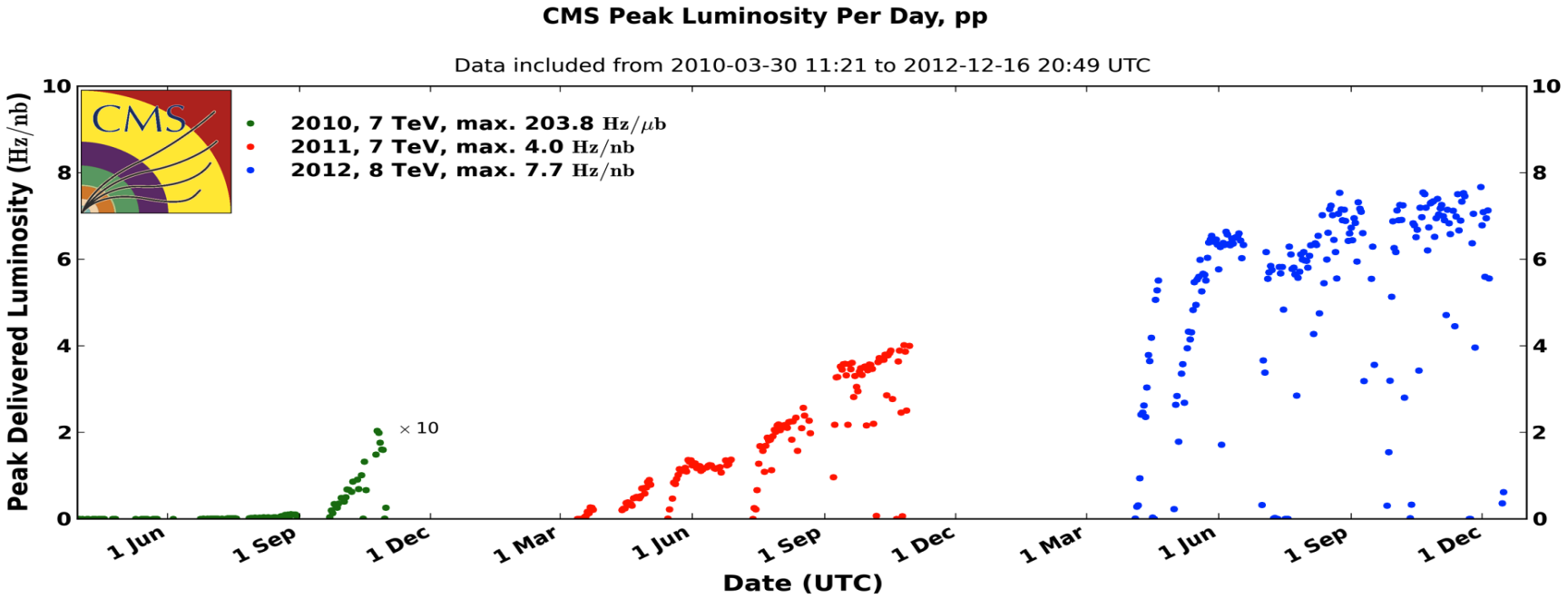
- Diagnoses Anomalies in
  - L1 rate
  - HLT physics stream rate
  - Dead time
  - Backpressure
  - Resynchronization rate
  - Farm health
  - Event builder and HLT farm data flow
  - HLT farm CPU utilization
  - …

2013-02-02 00:02:31	ADJUST	The average event size is suspiciously SMALL: 237 kB. Normally around 700kB are expected during a Physics run with all subdetectors in and at the beginning of a fill. Check with the shiftleader and the DQM shifter if something is going wrong. (Check the Radar plot of event sizes, DQM, Detectors which are missing in the readout, ...) (...I could do this also ... but I am too lazy and something also YOU have to do...)
2013-02-02 00:05:27	ADJUST	The total datarate written by the StorageManagers to disk has changed by 932.471560311376%. It is now 114.9 MB/s.
2013-02-02 00:05:35	ADJUST	sTTS of TIBTID is in Out-Of-Sync. Please follow the instructions in the DAQ Shifter Action Matrix (in the bulletin board)!
2013-02-02 00:05:41	ADJUST	Number of resyncs in this run now: 4
2013-02-02 00:05:43	ADJUST	sTTS of TIBTID is not anymore in Out-Of-Sync.
2013-02-02 00:05:51	ADJUST	TTS Alarm for partition CSC+: FMM fmmpc-s1d12-07.cms slot 3 is 9.25% in busy!

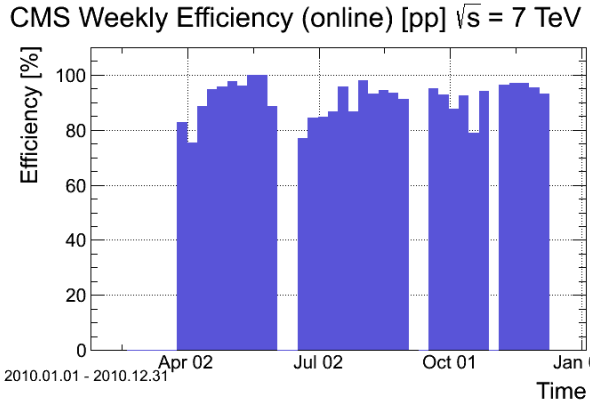
- Automatic actions
  - Triggers computation of a new central DAQ configuration in case of PC hardware failure(great help for on-call experts since 2012)

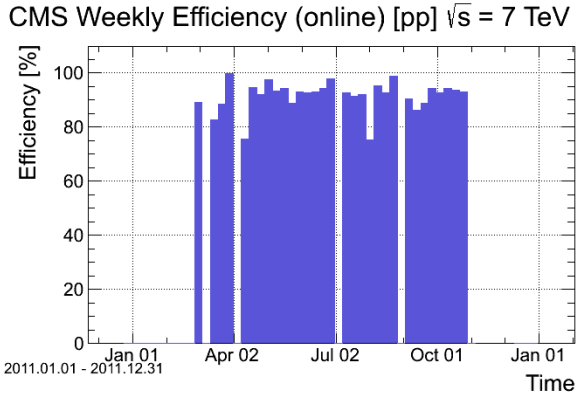
2013-02-02 01:48:32	INJ PROBE BEAM	I did not find a reason yet why the DAQ is in ERROR... may be the monitoring system is slow and I need more up to date data. I try again in 10 seconds. (try no: 2/3)
2013-02-02 01:48:47	INJ PROBE BEAM	====> !!! Trouble ahead !!! <==== The machine ru-c2a05-15.cms does not respond to ssh connections. It is probably crashed.
2013-02-02 01:48:47	INJ PROBE BEAM	I am now trying to generate and register a new configuration without the offending host ru-c2a05-15.cms in slice 1... some patience please...
2013-02-02 01:49:27	INJ PROBE BEAM	The new configuration has been generated and created. Probably stopping the run will fail. When you recylce the DAQ the new configuration without the broken computer will be picked up and you should be able to continue. But remember: if you are happily taking data at the moment defer the recycling of the DAQ as much as possible! Only interrupt this run when really necessary!

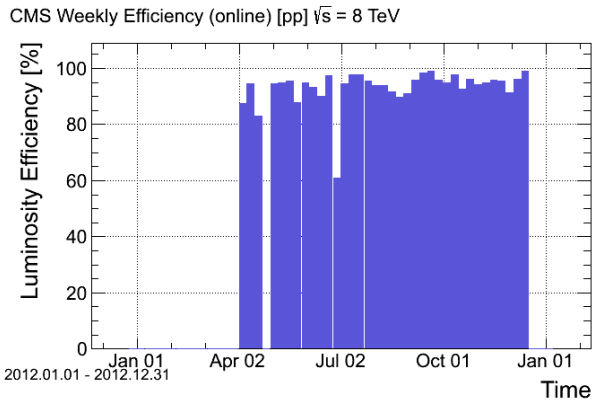

Operational Efficiency
CMS control room, Cessy, France …



CMS data taking efficiency:







CMS data taking efficiency (stable beams):

Central DAQ availability > 99.6% in all three years

2010: **90.7%** 2011: **93.1%** 2012 : **94.8%**

Summary & Outlook

Summary

- During Run 1 of the LHC, routine actions and error recovery have largely been automated in the CMS DAQ

- Two approaches were followed
  - Integrated into the control tree
    - **Local**, distributed
  - Central expert system
    - For diagnosis, advice &
    - automation that changes the system structure

- Automation allowed us to
  - Run CMS with a non-expert crew
  - Increase data taking efficiency despite frequent single-event upsets at high luminosity
  - Significantly ease the load on the on-call experts

Outlook

- Automation integrated into the DAQ control tree served us well
  - Knowledge encapsulated (rather than centralized)
  - Fast & reliable
  - Plan to cover more error scenarios using this approach

- Expert analysis of the central DAQ system will change significantly due to the DAQ & Filter Farm upgrade
  - Planning to include more information sources in analysis: errors, alarms …
  - We are currently investigating the use of a complex event processing engine (Esper)
    - Powerful query language
    - Rather steep learning curve

#87: P. Zejdl: 10 Gbps TCP/IP streams from FPGA for HEP

#72: R. Mommsen: Prototype of a file-based high level trigger in CMS

#139: A. Holzner: The new CMS DAQ system for LHC operation after 2014

- On our wish list
  - Completely automated operation: start of collision, cosmic and calibration runs
  - Correlation of errors: root cause finding

… stay tuned for CHEP 2015

Thank You