



Dataflow Monitoring

Nicoletta Garelli

ALICE, ATLAS, CMS & LHCb joint
workshop on *DAQ@LHC*

12-14 March 2013
Château de Bossey



What Dataflow Monitoring Means

- Monitoring in real time **the flow of data** to ensure optimal data taking
 - from detector readout to permanent storage
 - trigger & DAQ quantities (counters, data rate, buffer occupancy, etc.)
- Avoid **dead-time**
 - and eventually allow to fix problems
- Each experiment uses its own jargon to indicate the same thing

Requirements

Basics

- Access any relevant information in real time to follow data taking
- Online aggregation & data correlation
- Online problem detection: dead-time, data losses, etc.

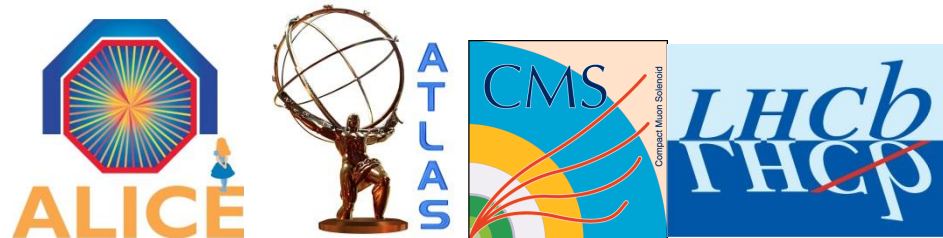
Added later

- Archive: access historical data for diagnostics, statistics, post-mortem
- Use monitoring data to trigger alarms and/or automatic actions to recover problems

Evolution

- Users: **shifters & experts**
- **LHC operations ...at the beginning**
 - scattered information and rudimentary tools
 - shifters: intense monitoring activity
 - experts: high presence in control room + ringing on-call phone
- **LHC operations ...routine**
 - coherent information and optimized tools
 - **automate** as much as possible to reduce shifter's tasks
 - see Luca's talk of this morning
 - move from custom GUI to ubiquitous **web based tools**
 - let's do all with a smartphone



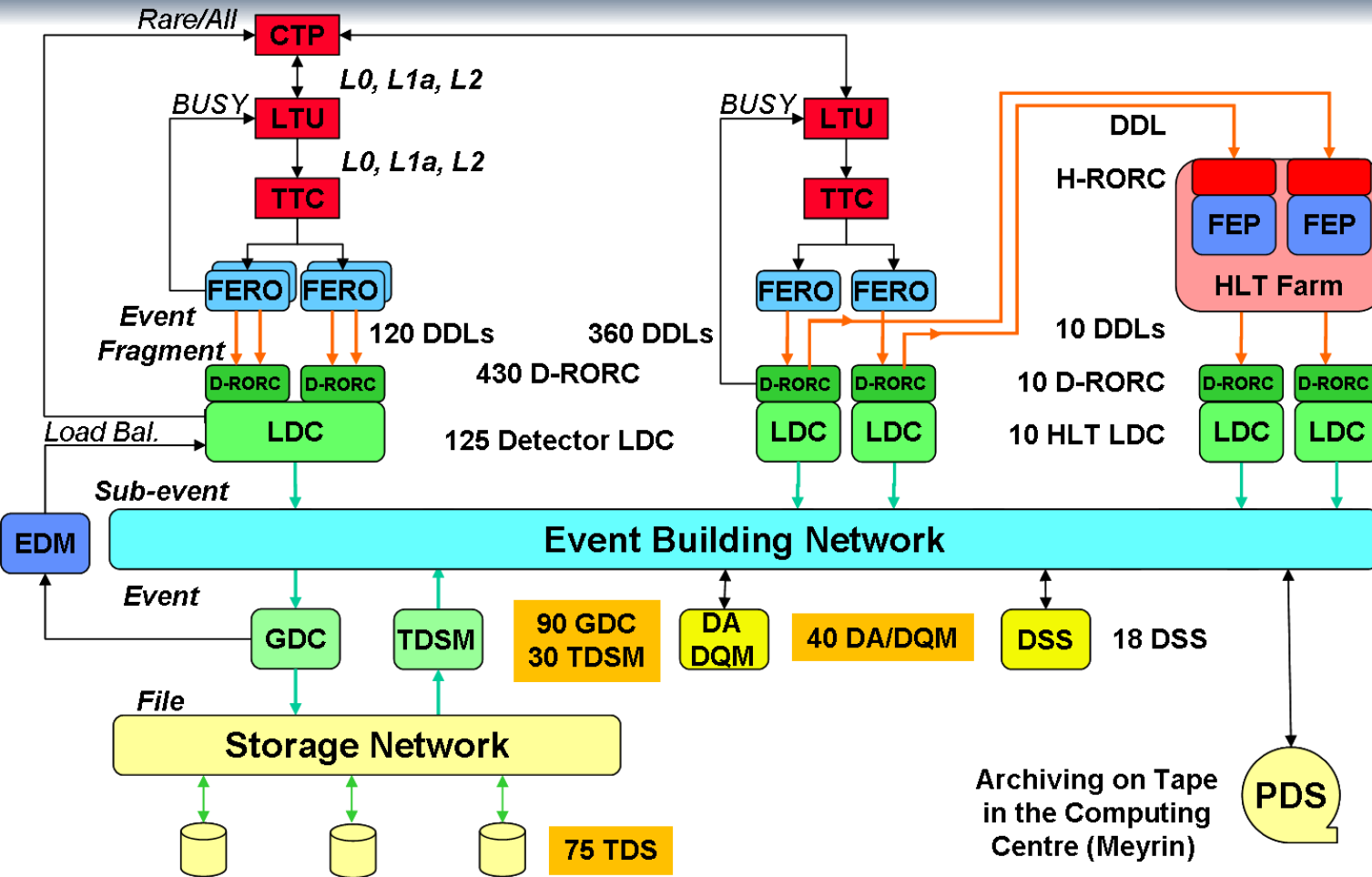


THE 4 ARCHITECTURES

Middleware

- Each experiment developed **4 different DAQ systems** using different technologies
- Variety reflected in dataflow monitoring middleware
 - LHCb & ALICE: Distributed Information Management (**DIM**)
 - client/server paradigm, light weight
 - ATLAS: Information Service (**IS**)
 - custom library on top of CORBA
 - client-server communication model where information is stored in memory by so called IS servers
 - CMS: **Web Service**
 - Cross-DAQ (**XDAQ**) framework

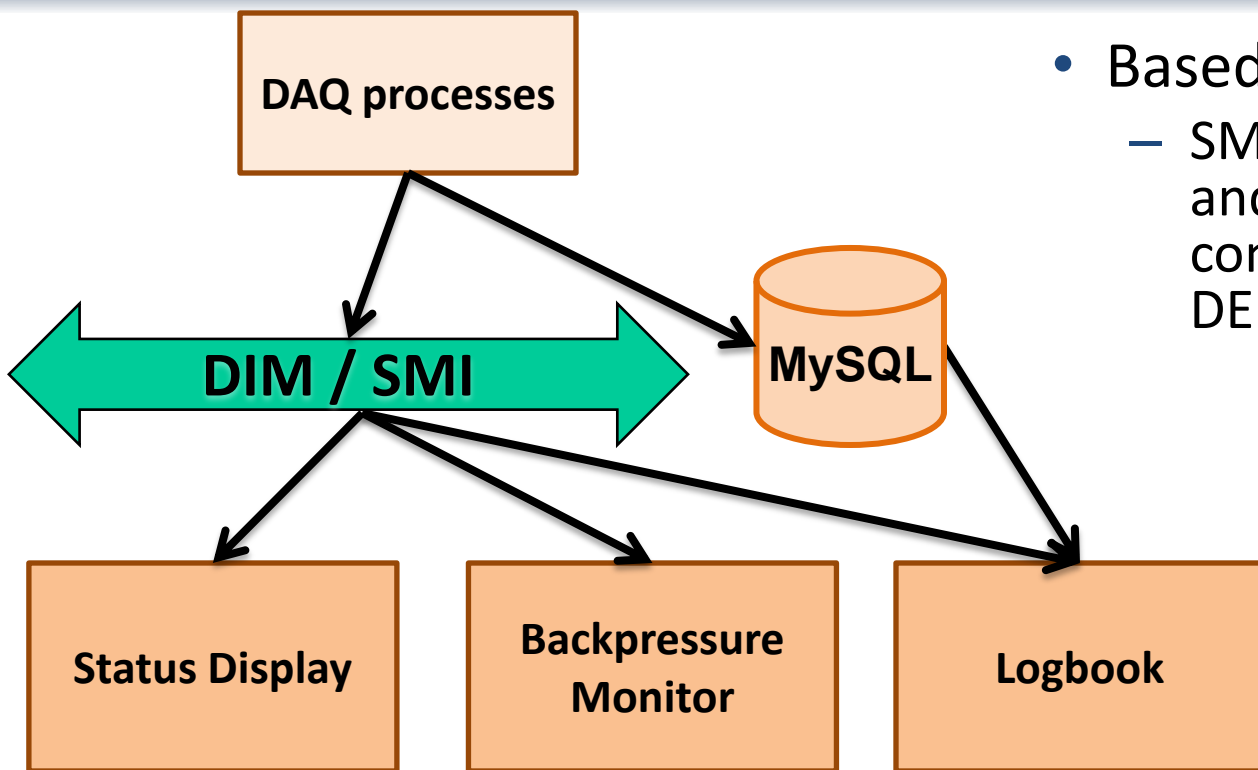
ALICE DAQ



NOTE:
HLT monitoring & dedicated expert storage monitoring not discussed here

- ~300 processes on ~300 machines
- 100k dataflow information published every 5 s → ~3 GB/h

ALICE Dataflow Monitoring Architecture



- Based on **DIM/SMI**
 - SMI: framework for designing and implementing distributed control systems developed by DELPHI

- **MySQL:**
 - store system configuration
 - LDC&GDC write run info
 - Archive.
 - Logbook as visualization

- **2 monitoring applications**
- Tcl/Tk

- **“Logbook”**
 - much more than what you think
 - PHP

ALICE Visualization

195531

General Info | Trigger Info | DAQ Info | HLT Info | DQM Info | Migration & Offline | Logs

Run Conditions | Run Statistics | EOR Reasons

General

Run #: 195531
 Period: LH13c
 Partition: PHYSICS_1
 Readout Detectors: ACORDE
 EMCAL
 FMD
 HLT
 HMPID
 MUON_TRG
 MUON_TRK
 PHOS
 PMD
 SDD
 SPD
 SSD
 TO
 TOF
 TPC
 TRD
 TRIGGER
 VO
 ZDC

Configuration

Run Type: PHYSICS
 HLT Mode: C
 # of LDCs: 170
 # of GDCs: 83
 (Old) EOR Reason: Operator_Request
 ECS Success: Yes
 DAQ Success: No

Date/Time

ECS Start Time: 22/01/2013 12:26:26
 DAQ Start Time: 22/01/2013 12:23:41
 CTP Start Time: 22/01/2013 12:23:44
 CTP End Time: 22/01/2013 23:36:51
 DAQ End Time: 22/01/2013 23:40:07
 ECS End Time: 22/01/2013 23:40:18
 Duration: 11:16:26
 Pause Duration: 00:03:50

Data Taking - Readout

Cluster	L2a	Events								
		Total			Physics			Calibration		
		#	MB	#	Hz	MB	MB/s	#	MB	
1	34 490 677	35 057 859	82 069 676	34 490 678	849.82	82 056 933	2 021.8	567 178	12 742	
2	33 996 004	34 555 242	86 666 195	33 996 004	837.63	86 653 623	2 135.1	559 235	12 571	
3	1 200	3 189	22 350	1 200	0.03	22 350	0.6	1 989	0	

Data Taking - Event Building

Total Events: 35 014 826
 Event Rate: 862.73
 Total Data: 29 614 294 MB
 Data Rate: 729.67 MB/s

Data Taking - Recording

Total Data: 22 847 661 MB
 Data Rate: 562.94 MB/s

DAQ_TEST

LDC status display

LDC name	aloneldc
host	trg-test
Current Trigger rate	5049.500
Average Trigger rate	4399.143
Number of sub-events	736887314
Sub-event rate	5049
Sub-events recorded	736887316
Sub-event recorded rate	5050
Bytes injected	579831198168864
Byte injected rate	4.003 GB/s
Bytes recorded	579831198168864
Byte recorded rate	4.003 GB/s
Nb. evt's w/o HLT decision	0
mem allocation failed	0
average time brAllocate	0.000000

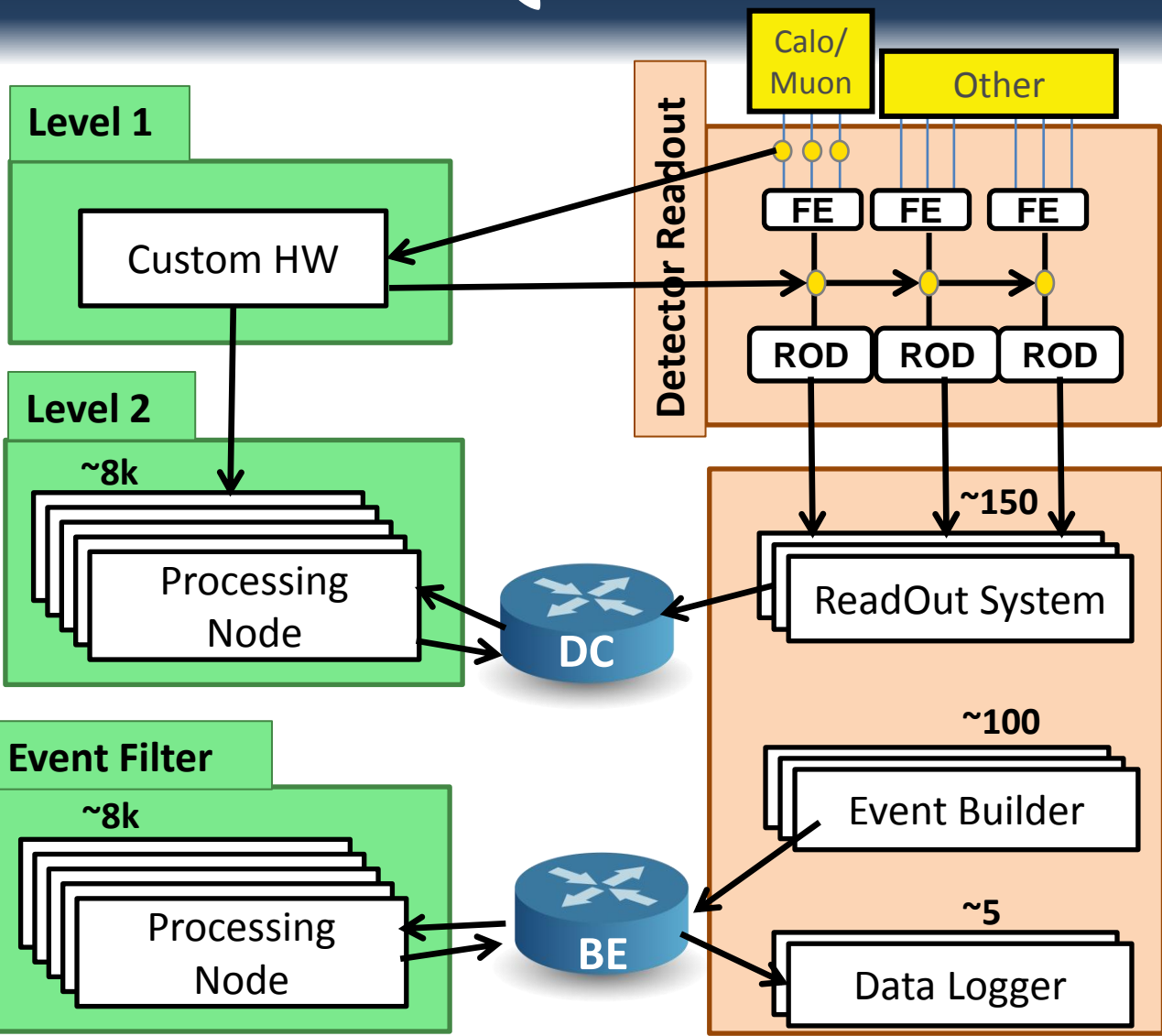
GDC status display

GDC name	alongdc
host	trg-test
Number of sub-events	0
Sub-event rate	0
Events recorded	0
Event recorded rate	0
Bytes recorded	0
Byte recorded rate	0 B/s
File count	0
Nb. incomplete events	0

ALICE & Android

The image displays three overlapping screenshots of the ALICE&Droid application on an Android smartphone. The top screenshot shows the home screen with the ALICE&Droid icon circled in blue. The middle screenshot shows the main interface with a circular detector diagram and buttons for 'Now @ P2', 'Saved Runs', 'Summary', and 'Search'. The bottom screenshot shows a 'Select period' dialog with options for 'Current shift', 'Previous shift', and 'Today', and a 'Select focus' dialog with options for 'ALICE', 'ACORDE', and 'CPV'.

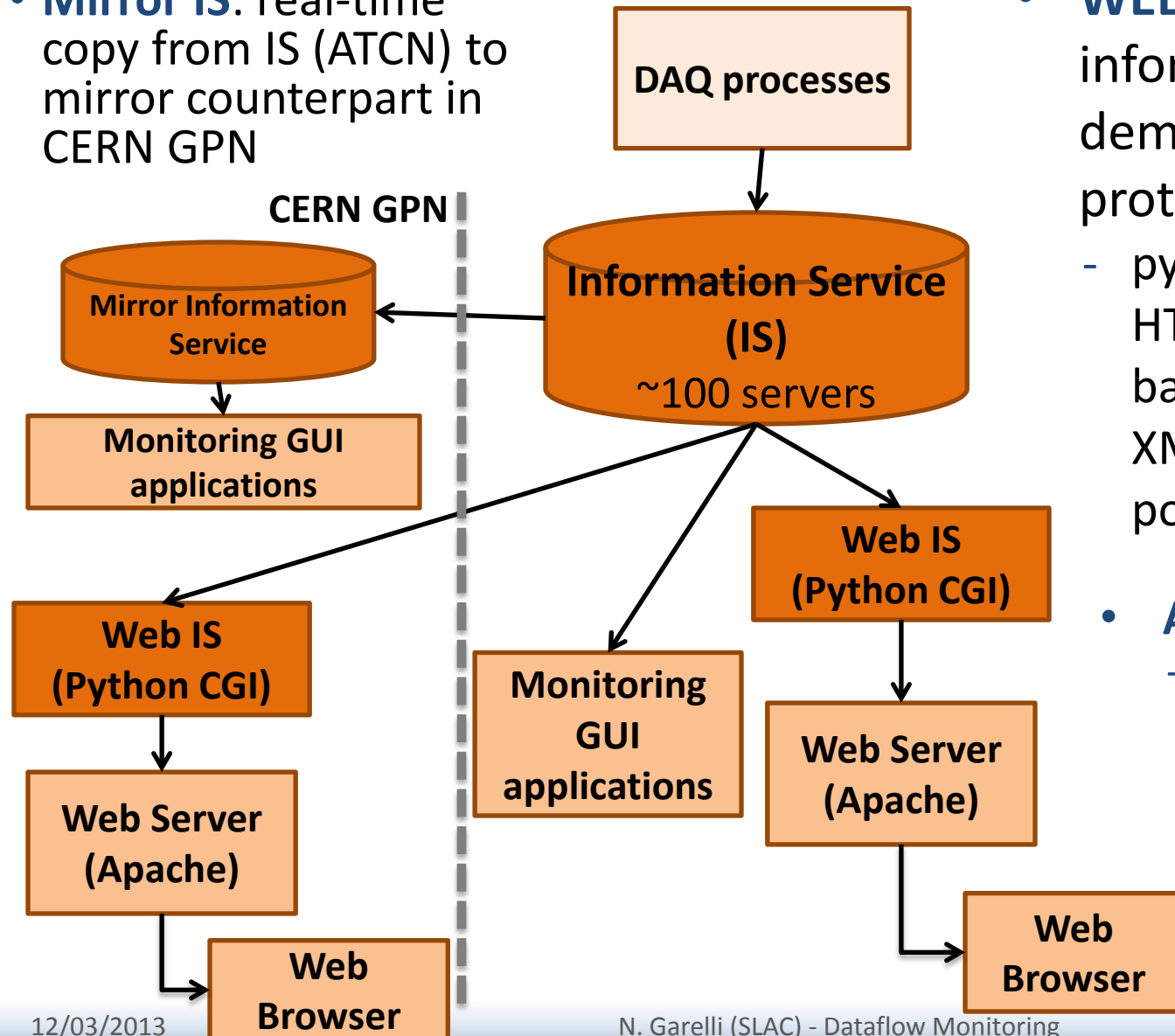
ATLAS DAQ



- $O(20k)$ processes on $\sim 2k$ machines
- 1M dataflow information published every 5-10 s
→ **~ 4 GB/h**

ATLAS Dataflow Monitoring Architecture

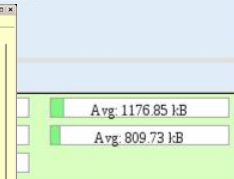
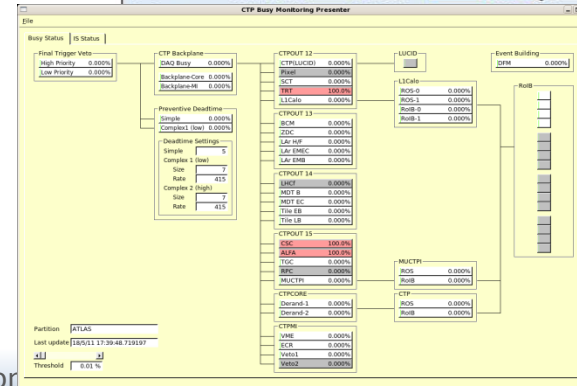
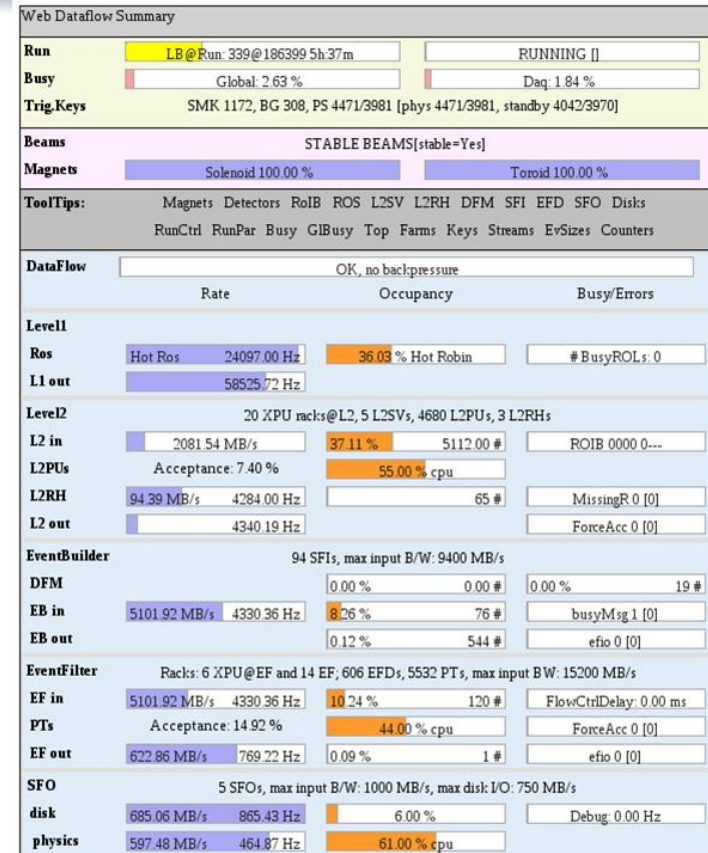
- **Mirror IS:** real-time copy from IS (ATCN) to mirror counterpart in CERN GPN



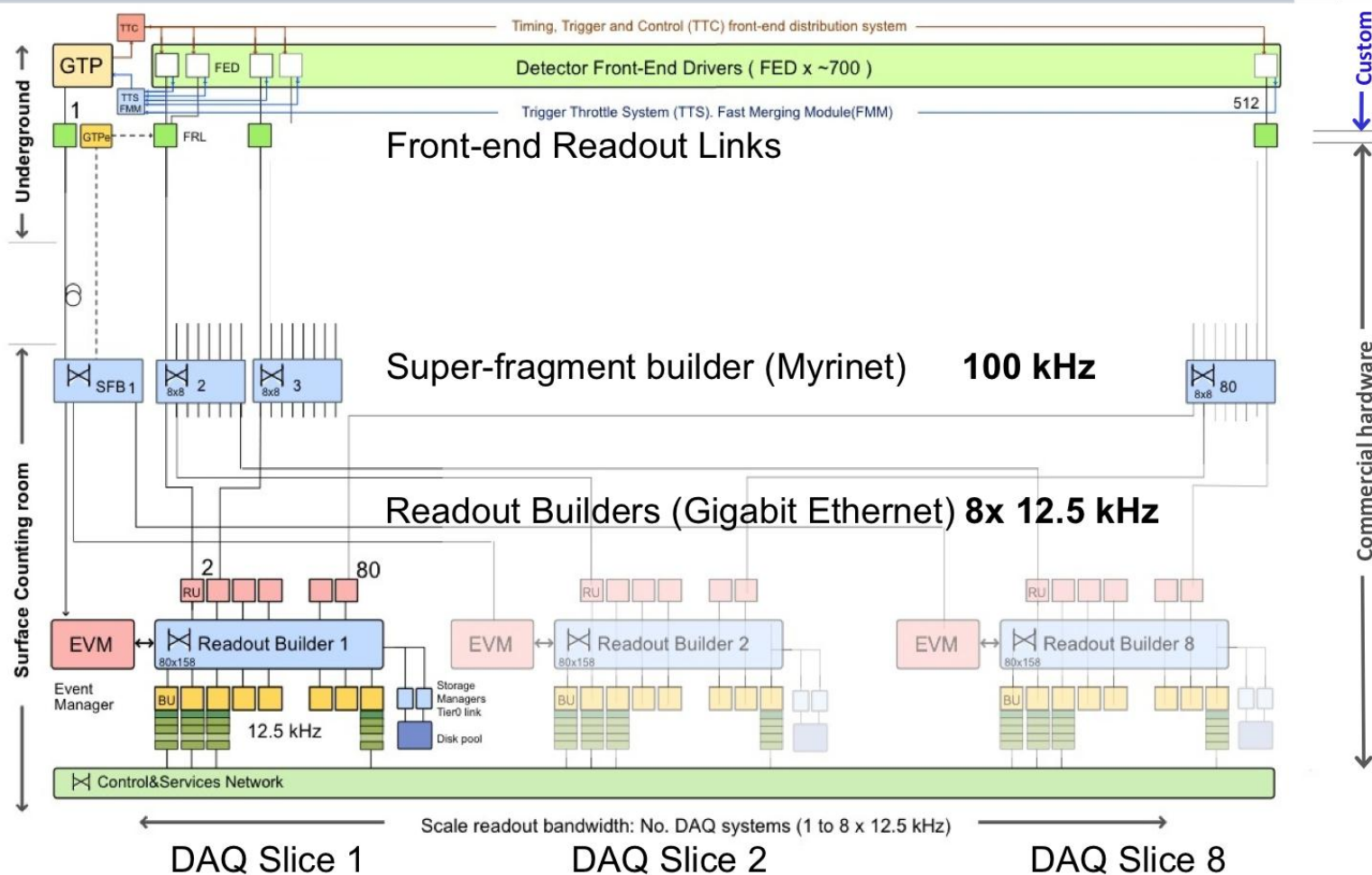
- **WEB IS:** IS gives information access on demand via HTTP protocol
 - python wrapper accepts HTTP requests & sends back dynamically formed XML text (value of IS obj pointed by given URL)
- **Archive:** None.
 - information stored & accessed for ~2 month in RDD files each ~30 s via network monitoring system

Shifter's Tools in 2012

- **DAQ Panel:** tool portal for shifters
- **DFSsummary**
 - dynamically constructed web page which computes & displays most important dataflow parameters (~200 variables)
 - ~30 s update rate
- **Busy Panel:** Qt application for monitoring dead-time
- **Shifter Assistant**
 - see Luca's talk of this morning

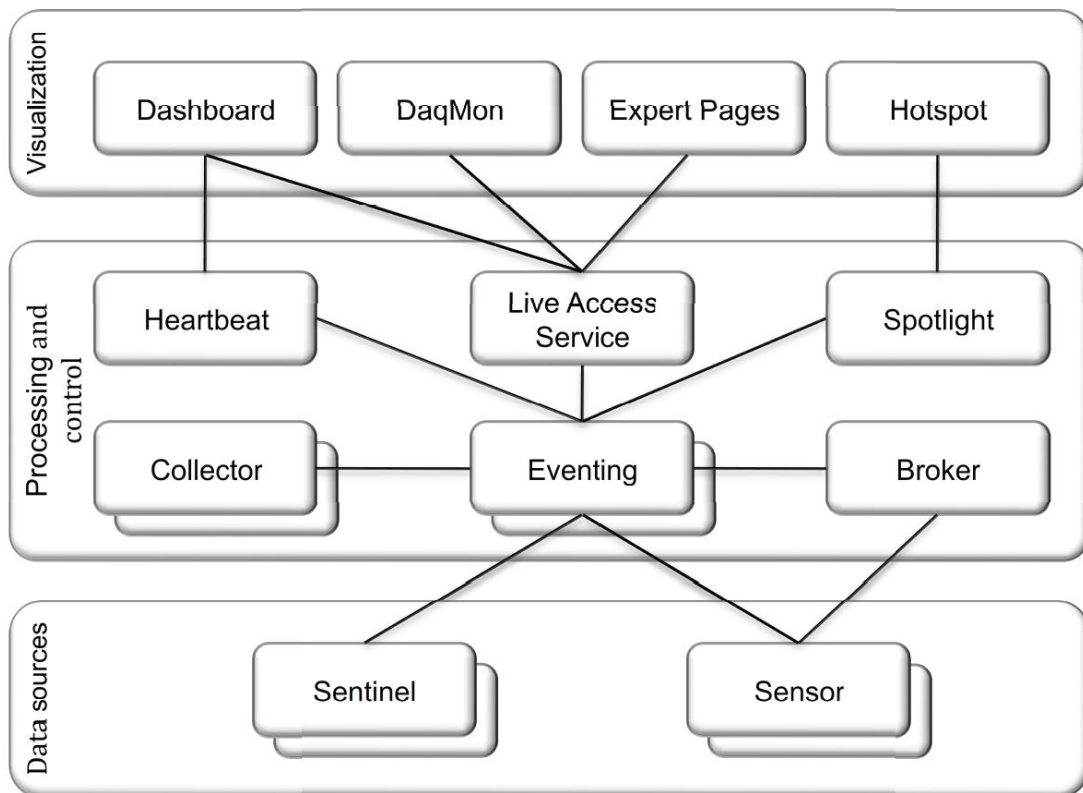


CMS DAQ



- $O(20k)$ processes on $\sim 2k$ machines
- $O(600k)$ dataflow information published every 1-5 s $\rightarrow \sim 8$ GB/h

XDAQ Monitoring & Alarming Service

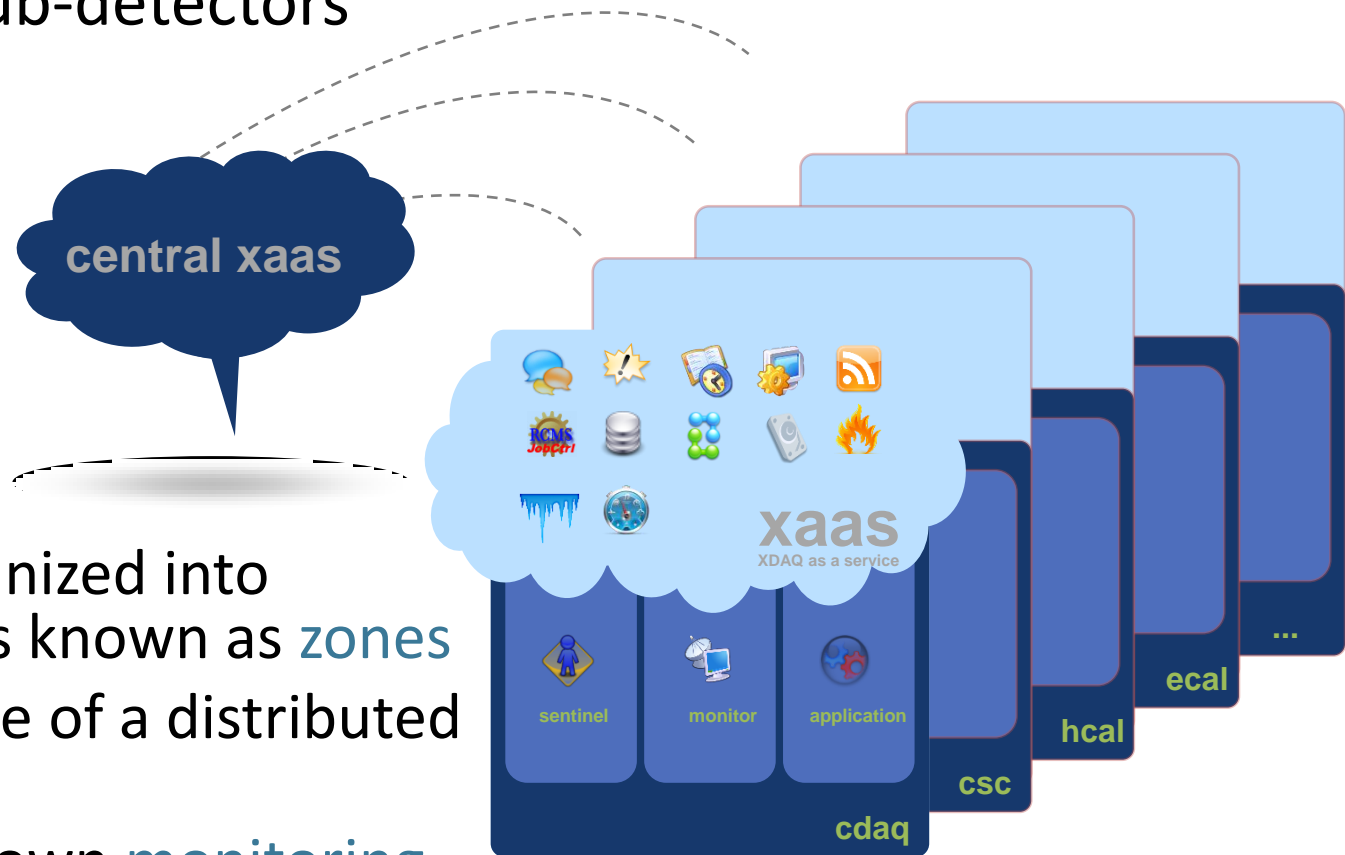


- **Archive:** automatic persistency of collected tables in ORACLE according to configuration
 - Subset of information stored: ~30 GB/y

- Fully scalable distributed monitoring & alarming system
- Service-oriented architecture organized in 3-tier structured collection of communicating components:
 - **Sensor:** report monitoring data
 - **Eventing:** scalable publisher-subscriber service orchestrated by a load balancer application (**Broker**)
 - **Collector:** build relational tables
 - **Live Access Service:** presentation of raw data (**Web Service**)

Monitoring as a Service

- XDAQ as a Service (**XaaS**): common infrastructure for both central DAQ & sub-detectors
- interoperable services providing standard functionalities for use in XDAQ environment
- All processes organized into searchable groups known as **zones**
- **zone** defines scope of a distributed XDAQ application
- Each zone has its own **monitoring data types** (flashlists)

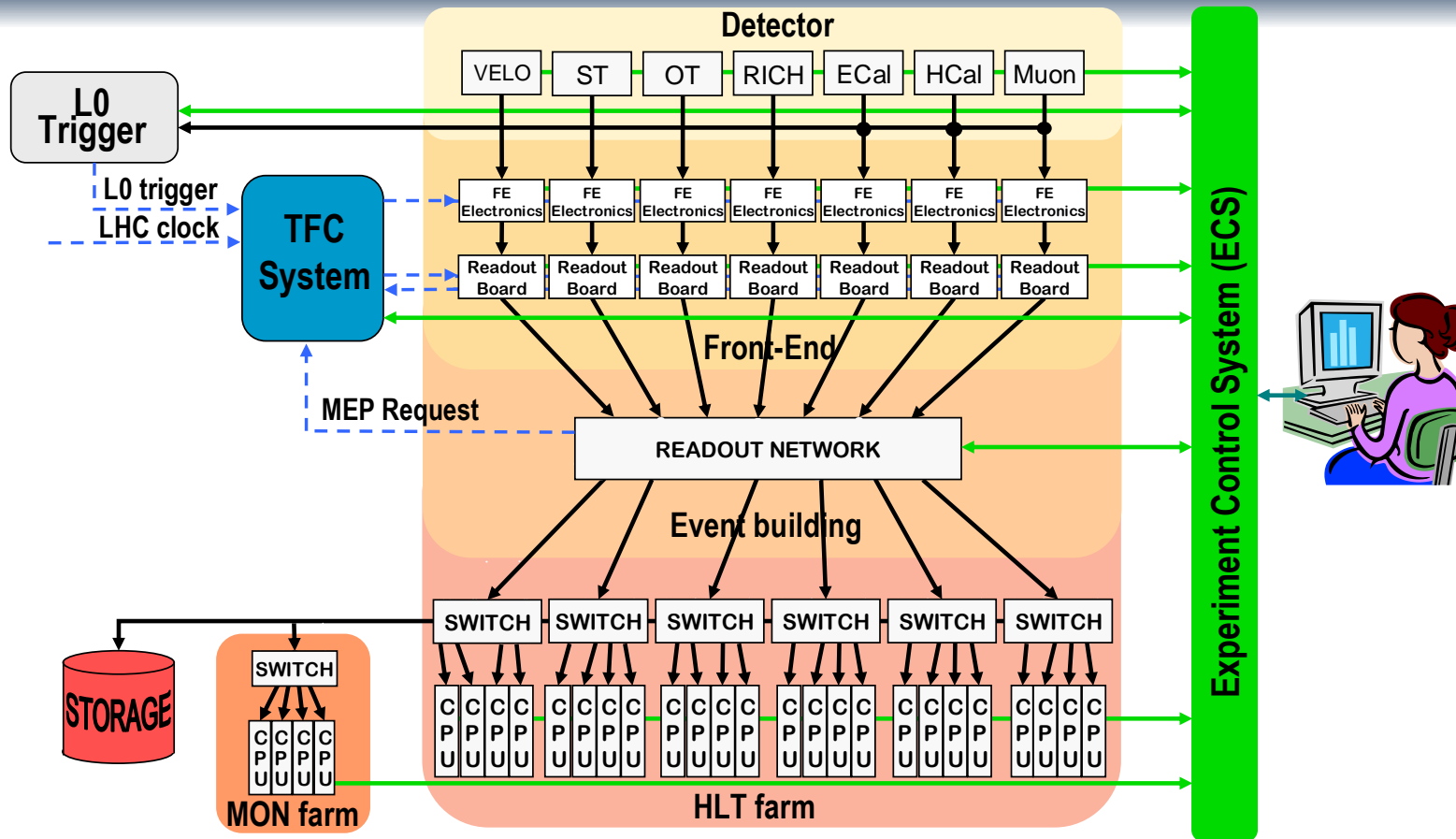


CMS Visualization

- LabView DAQMon

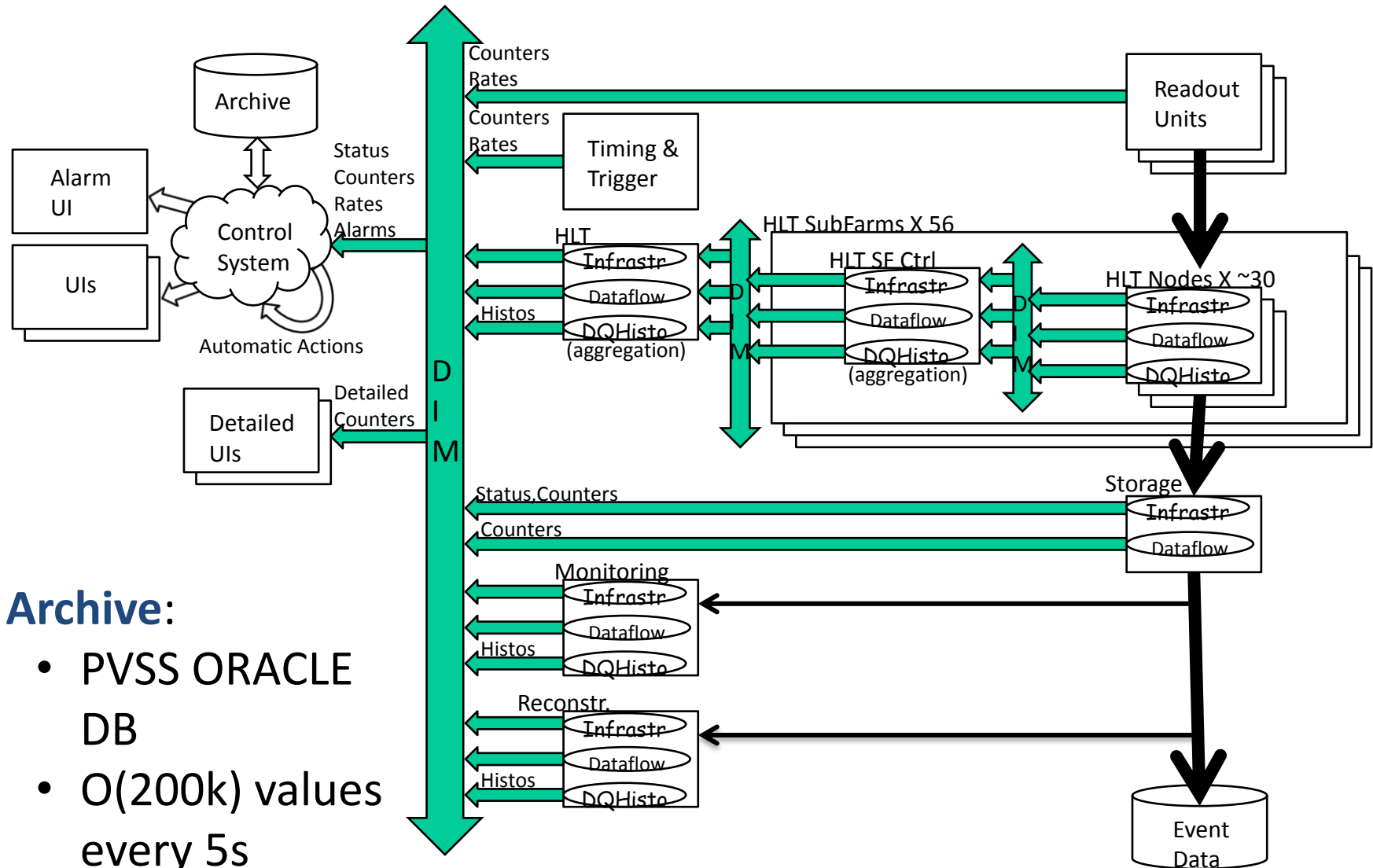


LHCb DAQ



- O(40 k) processes on 2k machines
- 4M dataflow information published every 5s → ~ 11.5 GB/h

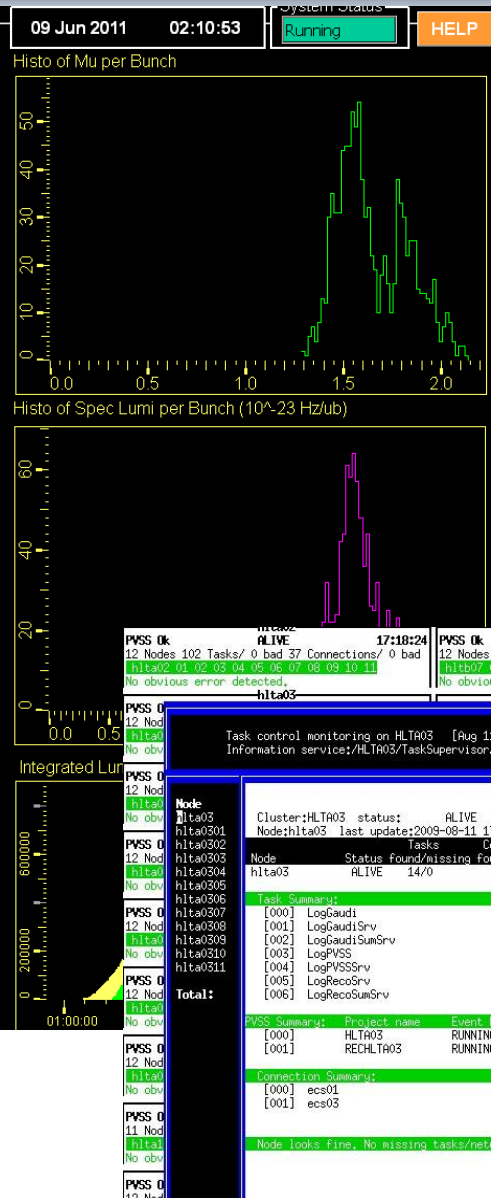
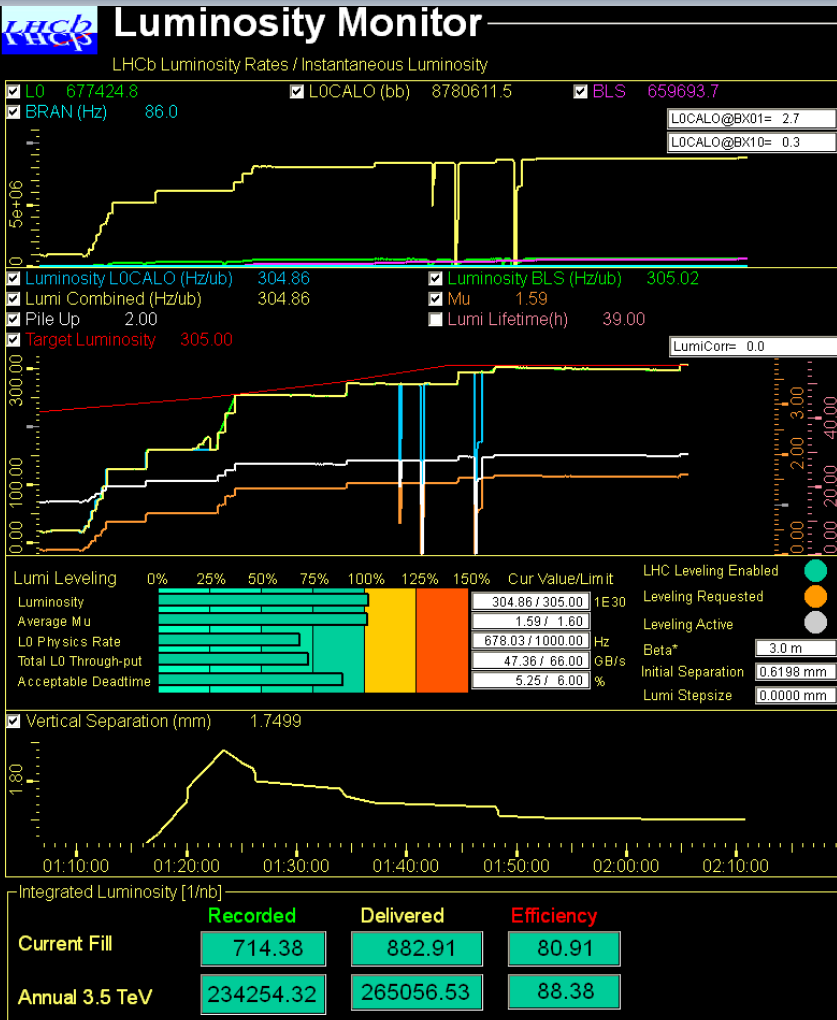
LHCb Dataflow Monitoring Architecture



Archive:

- PVSS ORACLE DB
- O(200k) values every 5s

LHCb Visualization



PVSS GUI

VT100 graphics
detailed UI

CONCLUSIONS

Satisfied?

“**YES**, it does the job”

“ ... **BUT** ... ”

- 4 different solutions for the same problem ...
- sharing experience and maybe even future common solutions?

→ **Luciano's talk on Thursday**

