Data Acquisition, Trigger and Control (for large Physics Experiments)

Concepts & Principles

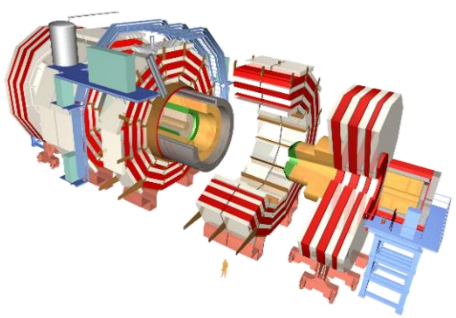
Clara Gaspar, July 2023

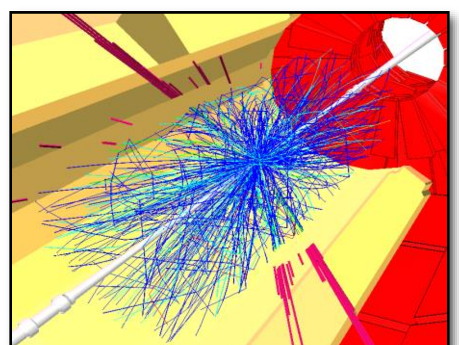
Acknowledgements:
This presentation is based on the work of many people throughout many years, Thanks!

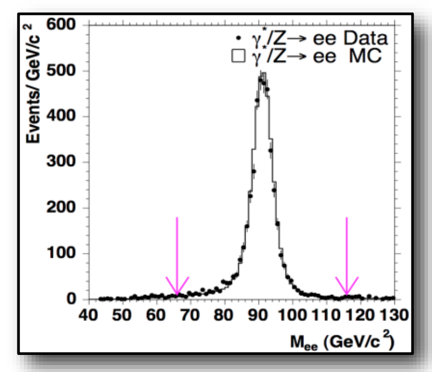
The LHC Experiments

Data Acquisition at the LHC

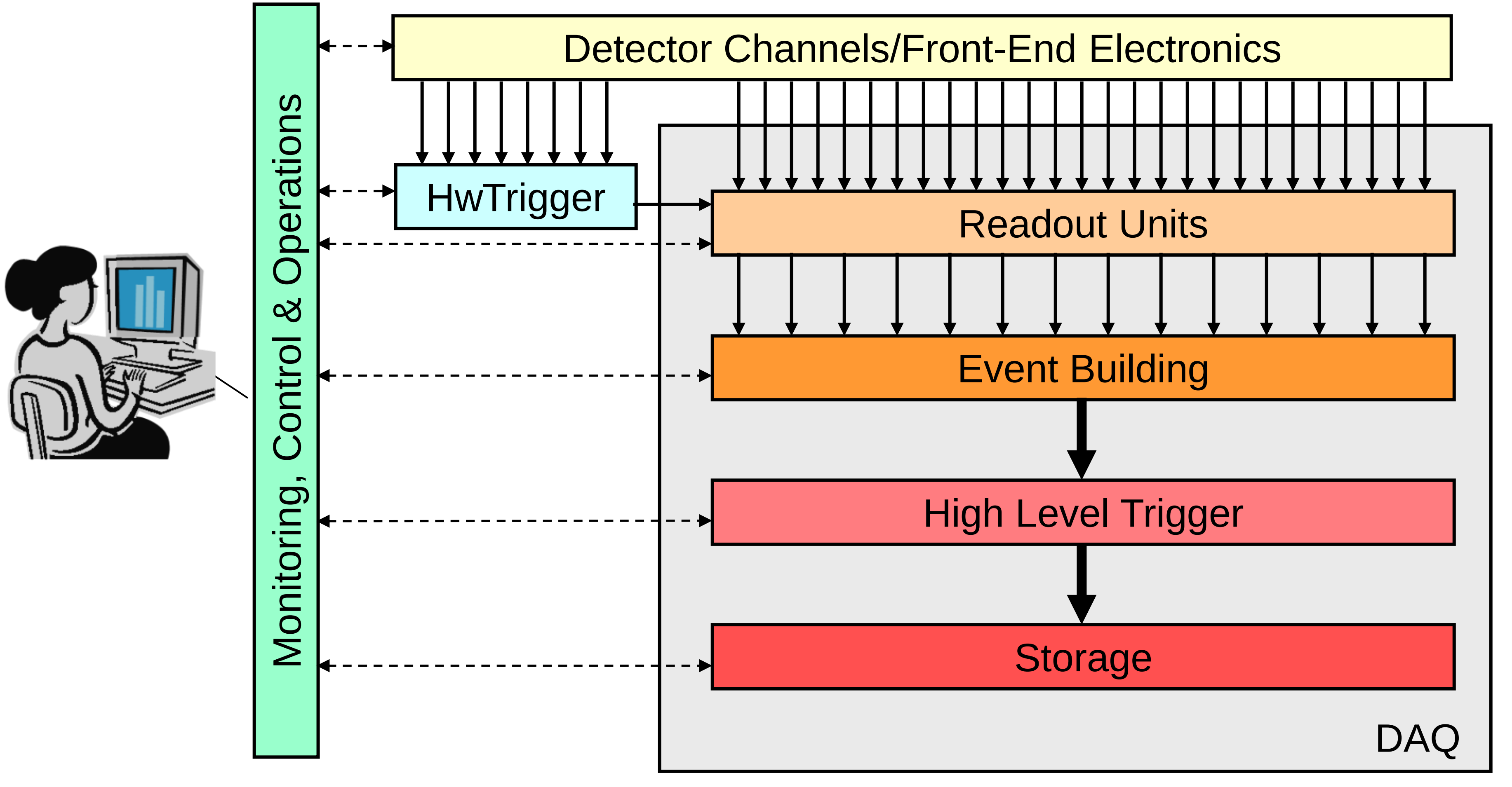
- Data Acquisition enables physics analyses to be performed on the data produced by the detector. Definitions:
  - Trigger System
    - Selects in Real Time “interesting” events from the bulk of collisions. - Decides if YES or NO the event should be read out of the detector and stored
  - Data Acquisition System
    - Gathers the data produced by the detector and stores it (for positive trigger decisions)
  - Control System
    - Provides the overall Operation, Configuration and Monitoring







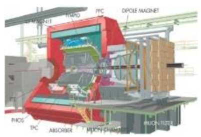
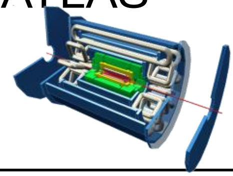
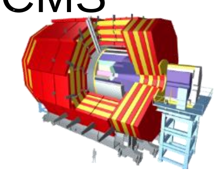
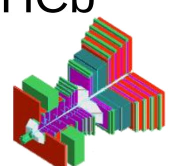
Trigger, DAQ & Control



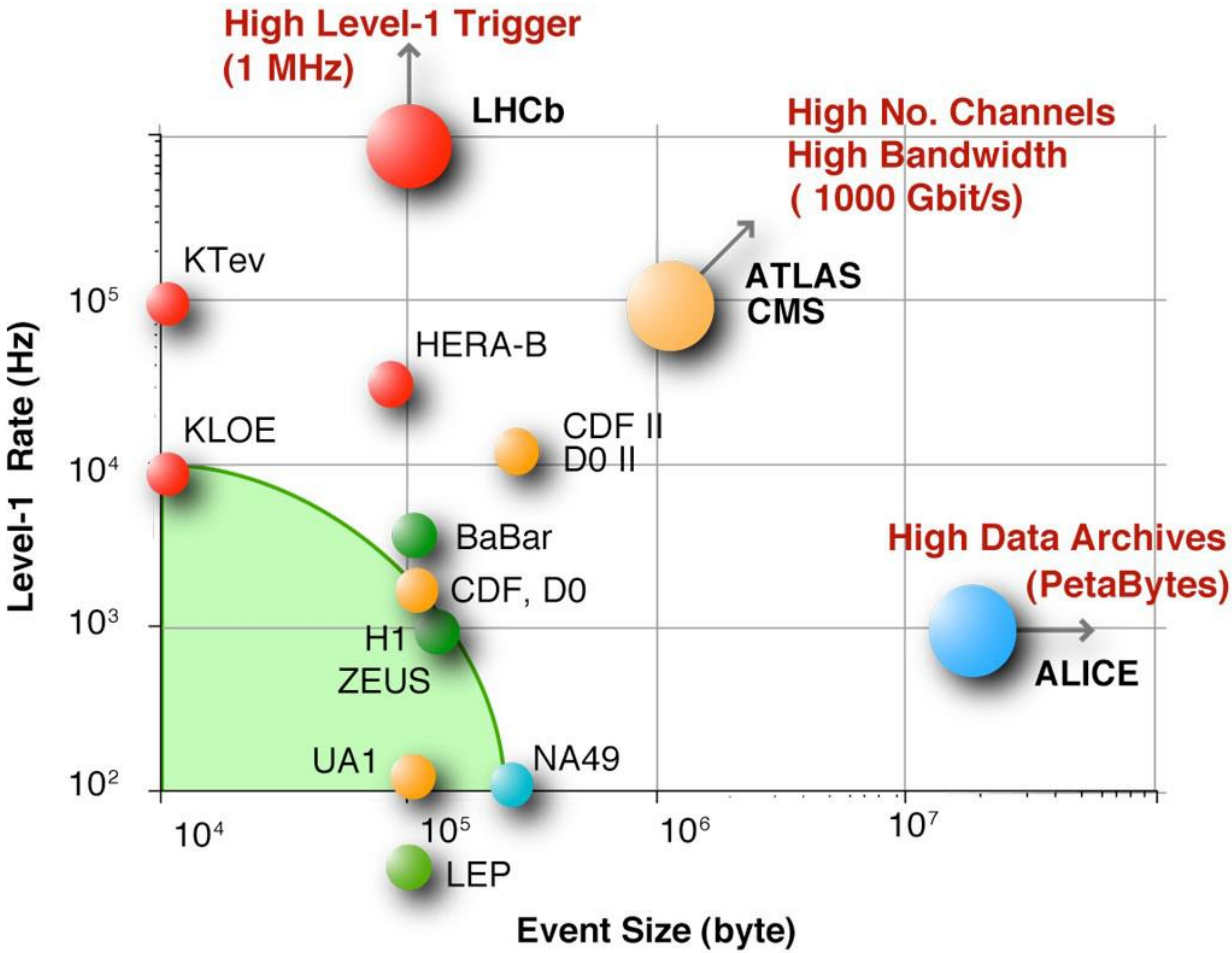
LHC in Numbers

	LHC (2009)
Bunch Crossing Rate	40 MHz
Bunch Separation	25 ns
Nr. Electronic Channels	≈ 10 000 000
Raw data rate	≈ 1 000 TB/s
Data rate on Tape	≈ 100 MB/s
Event size	≈ 1 MB
Rate on Tape	100 Hz
Analysis	10^{-6} Hz (Higgs)

LHC Exp DAQ (until 2020)

	Event Size (MByte)	L1 Rate (KHz)	Bandwidth (GByte/s)	Storage Rate (KHz)	Storage (GBytes/s)
ALICE 	25	1	25	0.050	1.250
ATLAS 	1	100	100	0.200	0.200
CMS 	1	100	100	0.200	0.200
LHCb 	0.05	1000	50	5	0.250

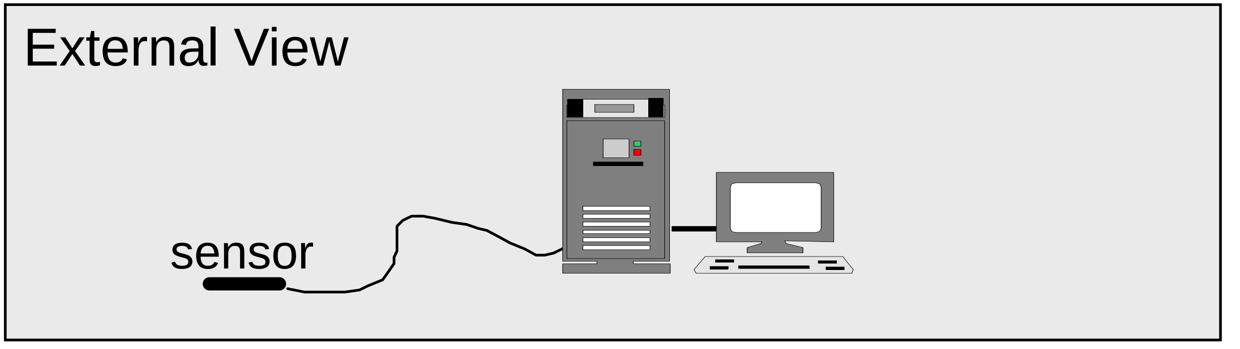
LHC DAQ Phase-space

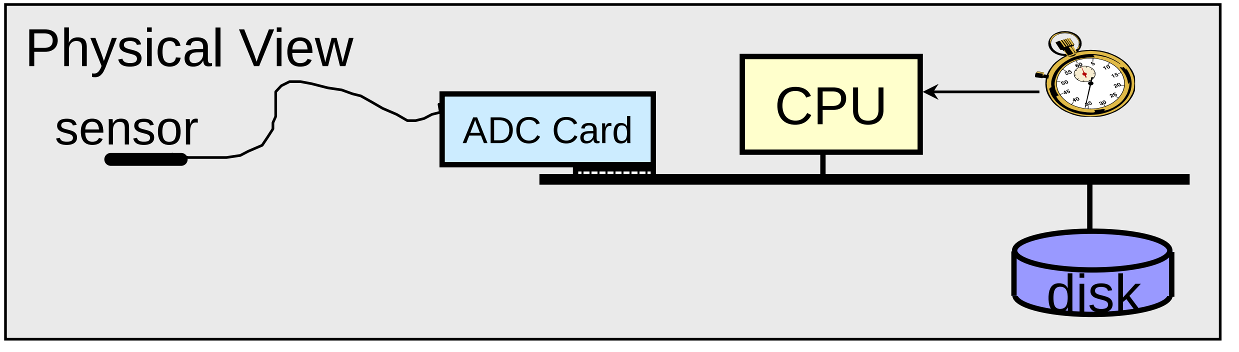


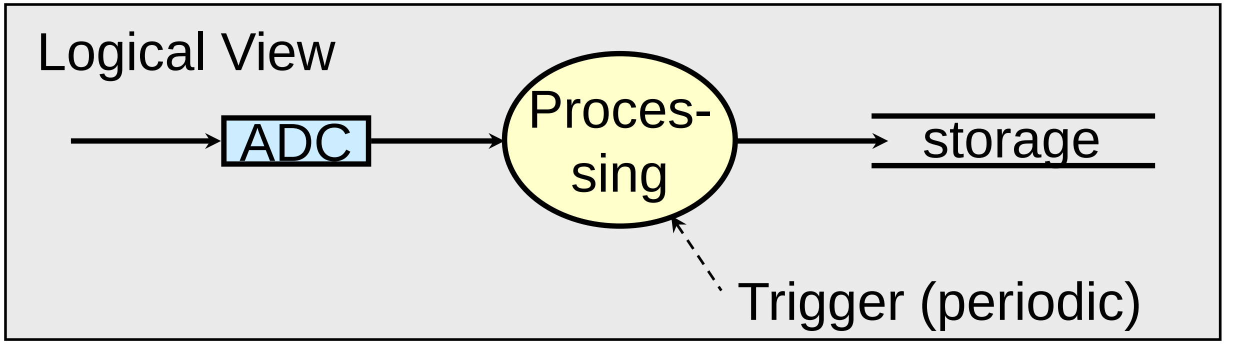
Basic Concepts

Clara Gaspar, July 2023

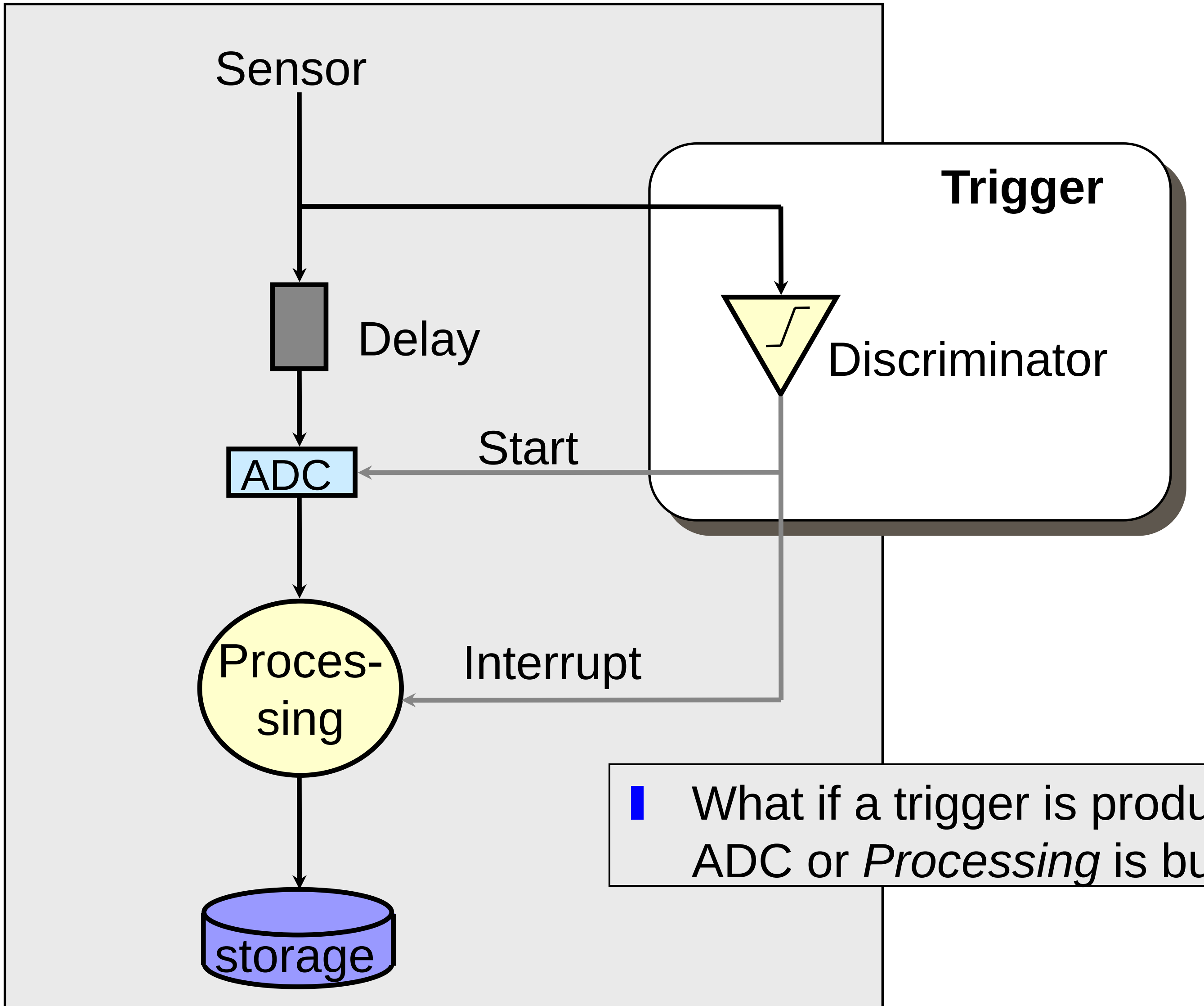
Trivial DAQ


External View
sensor


Physical View
sensor
ADC Card
CPU
disk

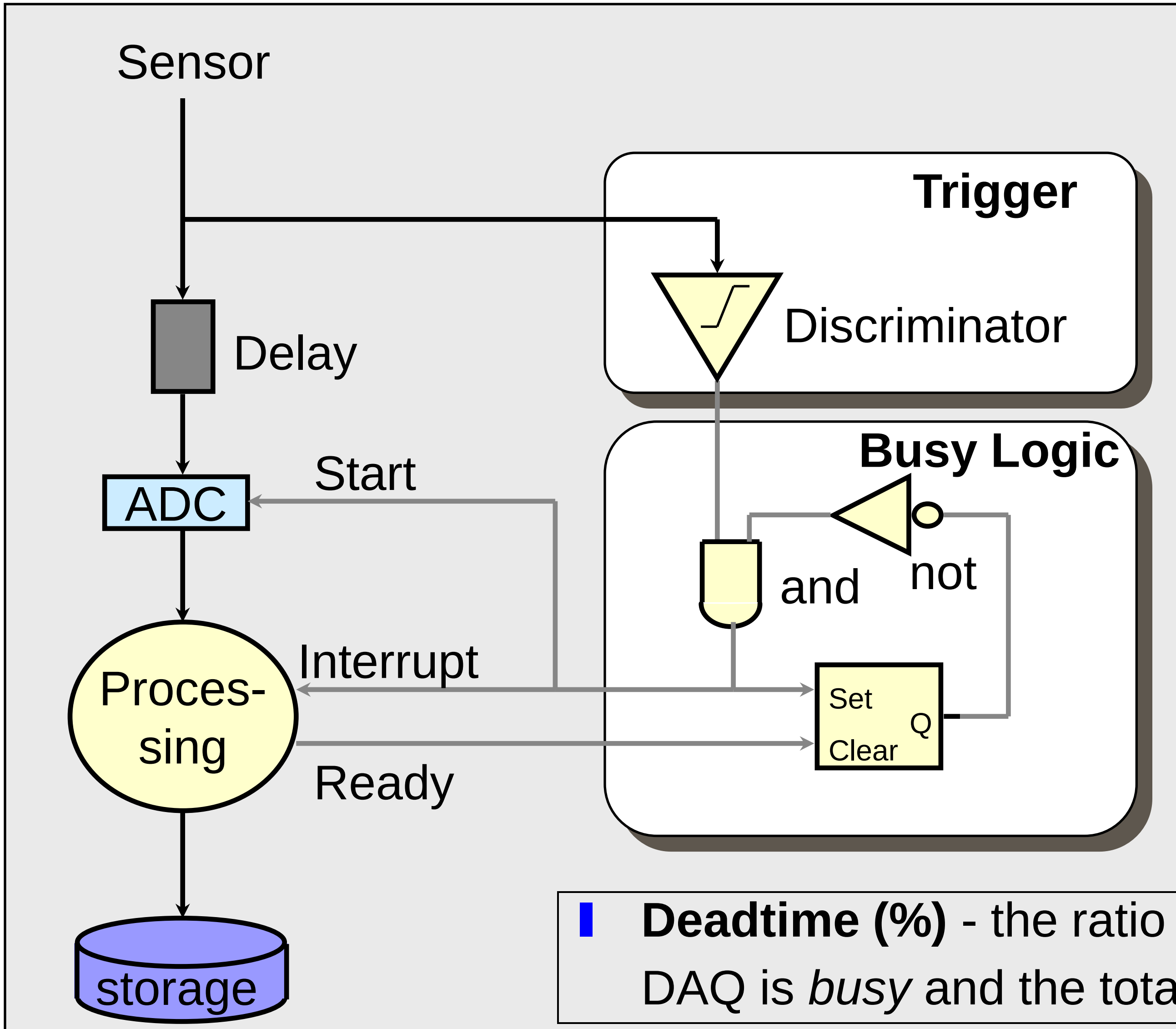

Logical View
ADC
Proces-
sing
storage
Trigger (periodic)

Trivial DAQ with a real trigger



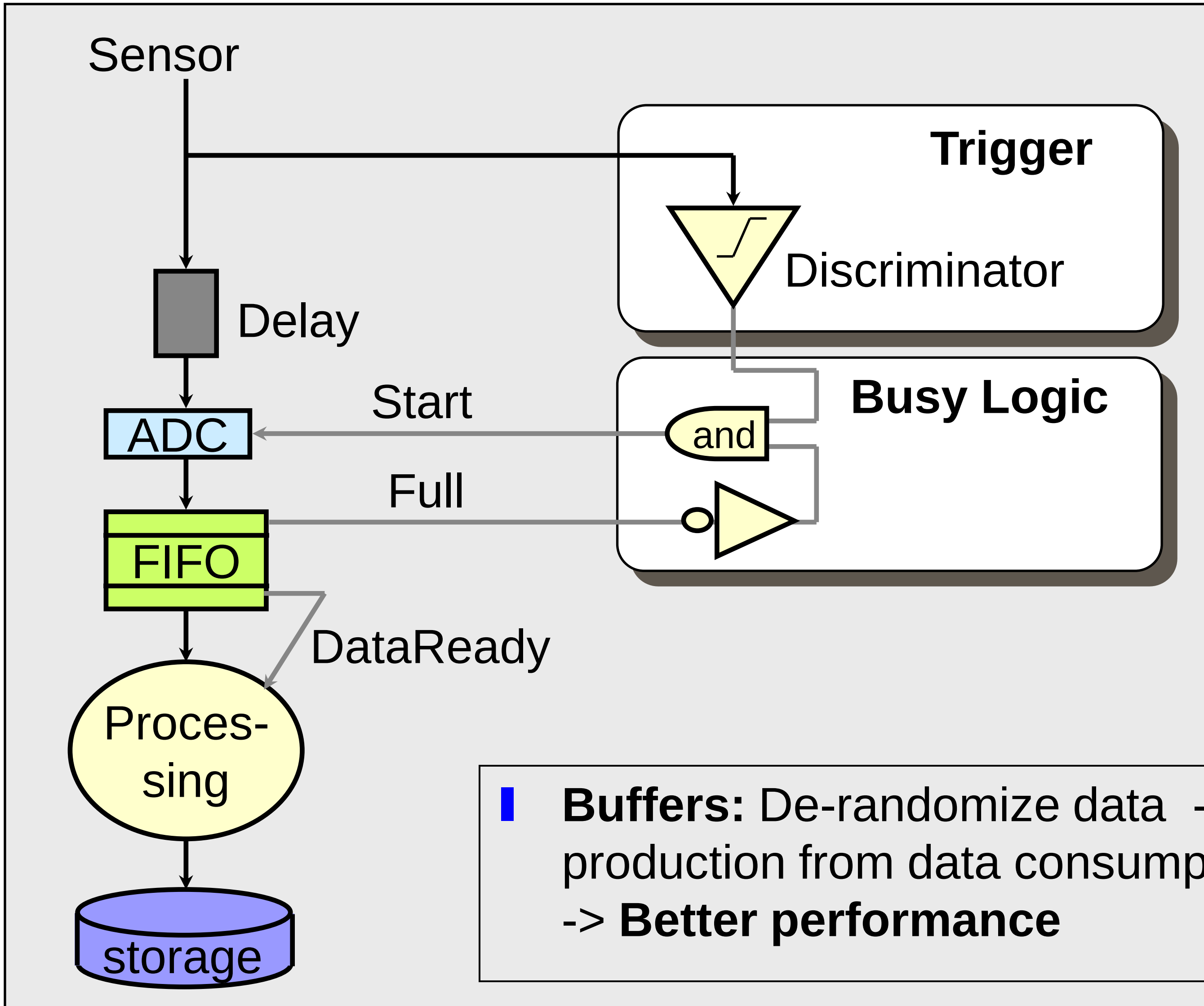
- What if a trigger is produced when the ADC or *Processing* is busy ?

Trivial DAQ with a real trigger (2)



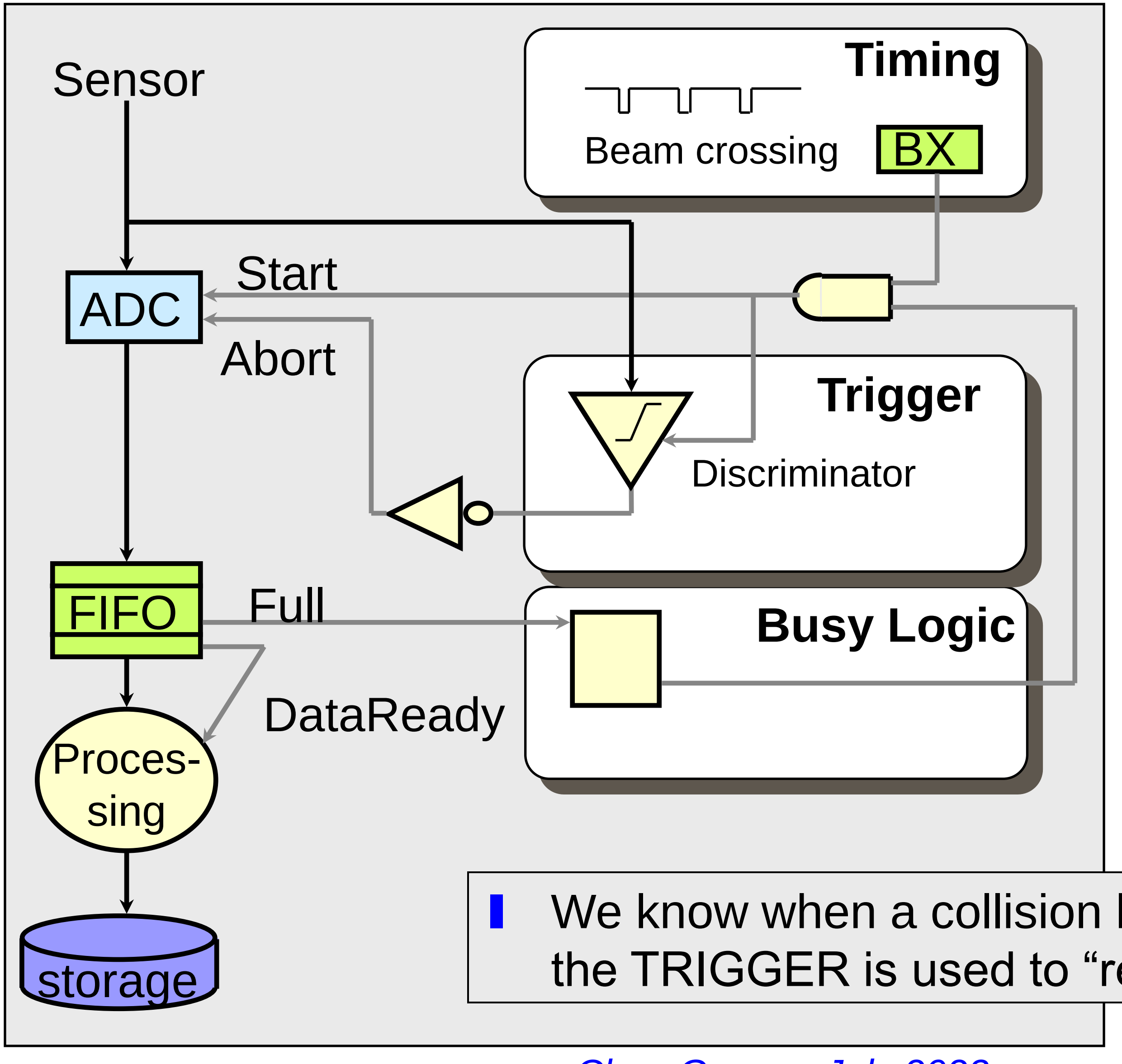
- **Deadtime (%)** - the ratio between the time the DAQ is *busy* and the total time

Trivial DAQ with a real trigger (3)



- **Buffers:** De-randomize data -> decouple data production from data consumption
-> **Better performance**

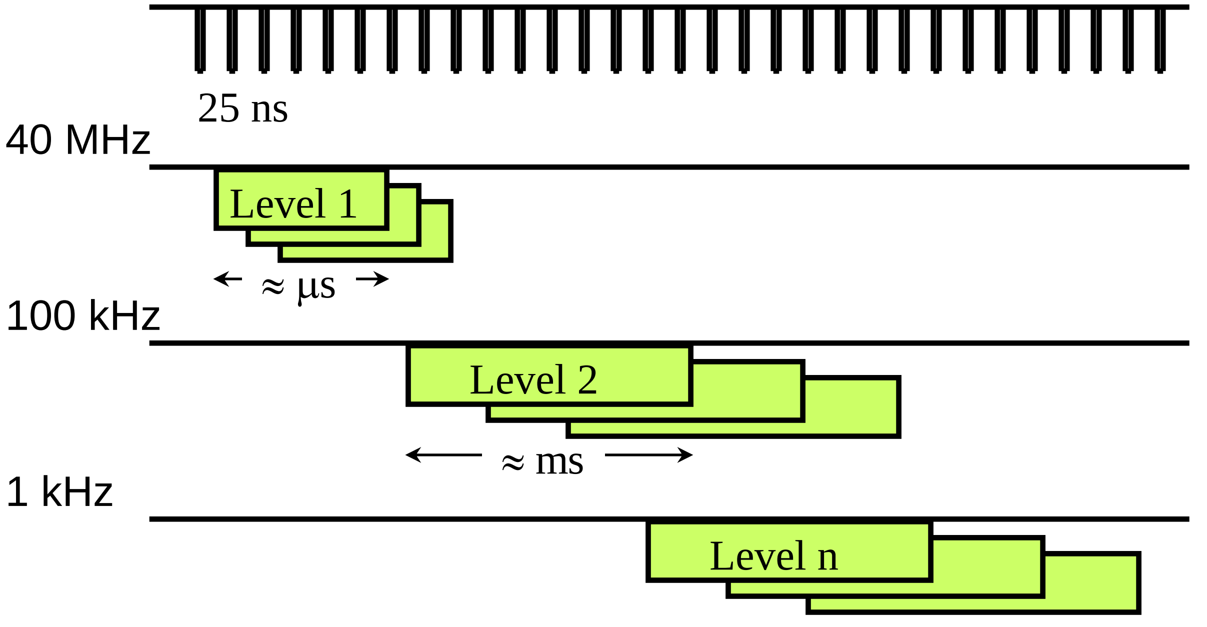
Trivial DAQ in collider mode



- We know when a collision happens
  the TRIGGER is used to "reject" the data

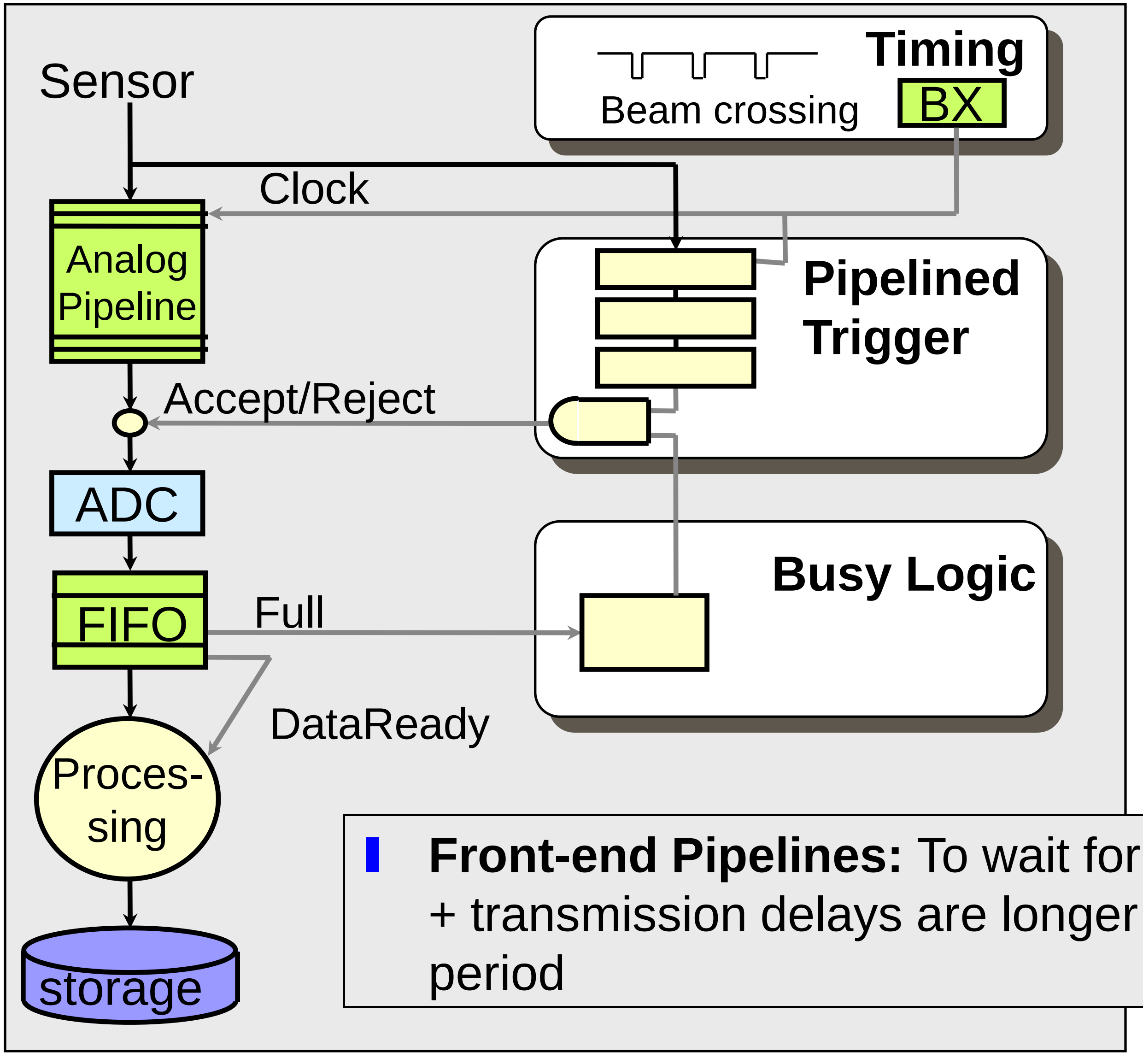
LHC timing

p p crossing rate 40 MHz (L=10^{33}- 4×10^{34}cm^{-2} s^{-1})



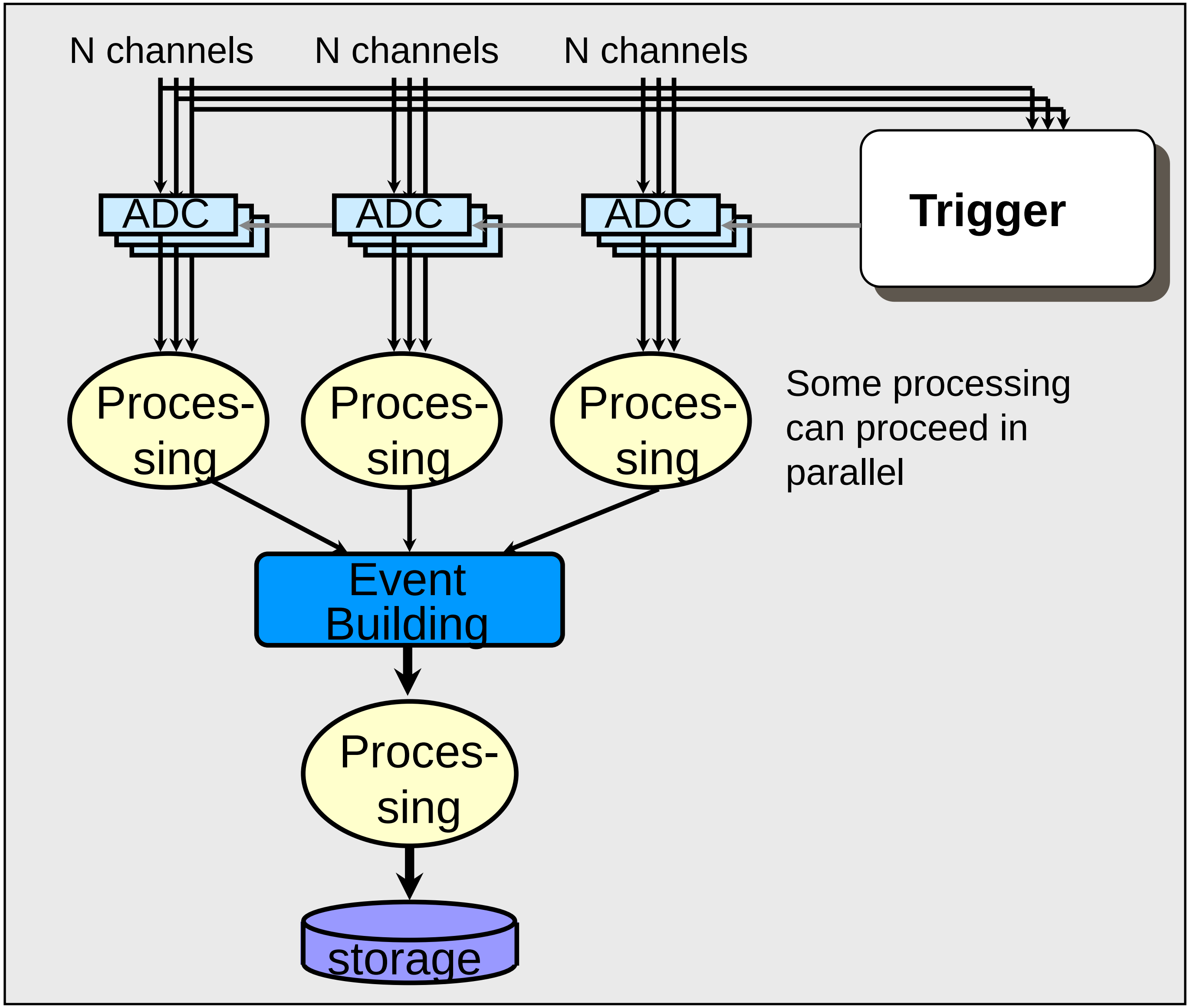
- Level 1 trigger time exceeds bunch interval
- Event overlap & signal pileup (multiple crossings since the detector cell memory greater than 25 ns)
- Very high number of channels

Trivial DAQ in LHC



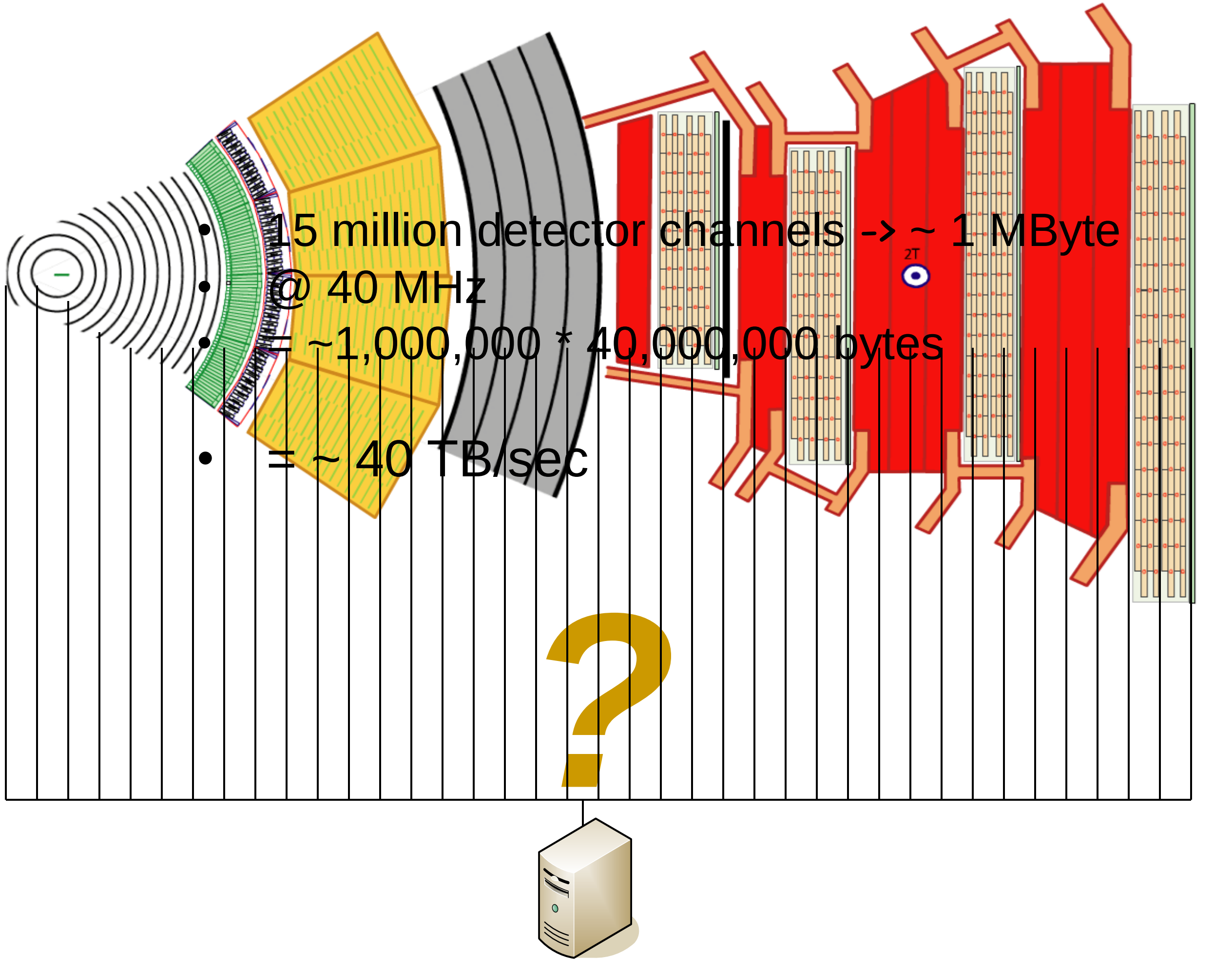
- **Front-end Pipelines:** To wait for the trigger decision + transmission delays are longer than beam crossing period

Less trivial DAQ



The Real Thing

- 15 million detector channels -> ~ 1 MByte
- @ 40 MHz
- = ~1,000,000 * 40,000,000 bytes
- = ~ 40 TB/sec

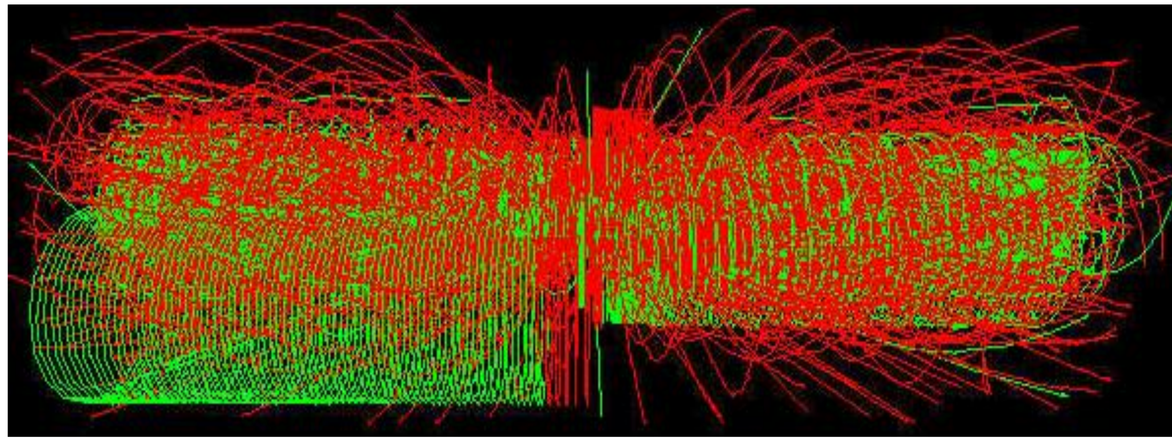


Trigger

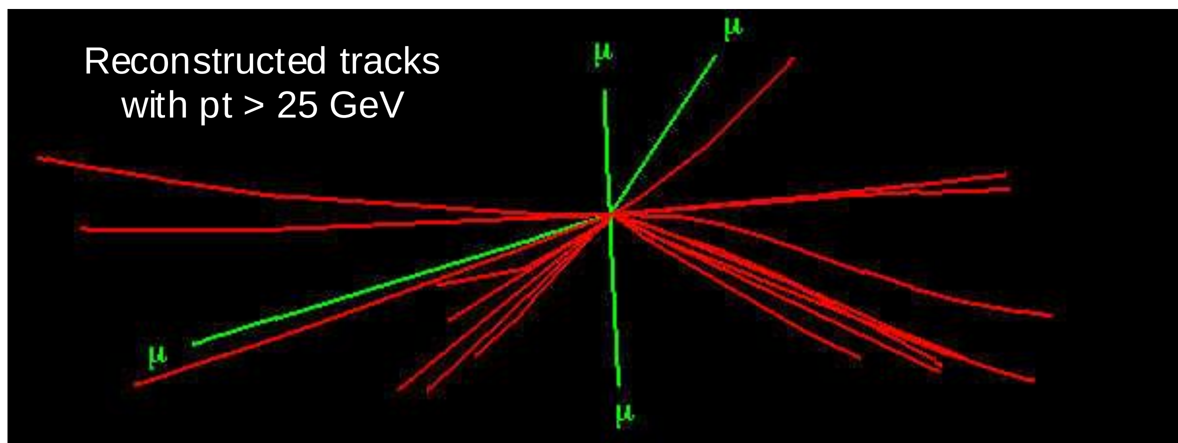
Clara Gaspar, July 2023

Trigger System

- **The Trigger system** detects whether an event is interesting or not
  - Typical ATLAS and CMS* event
  - 20 collisions may overlap
  - This repeats every 25 ns



  - A Higgs event



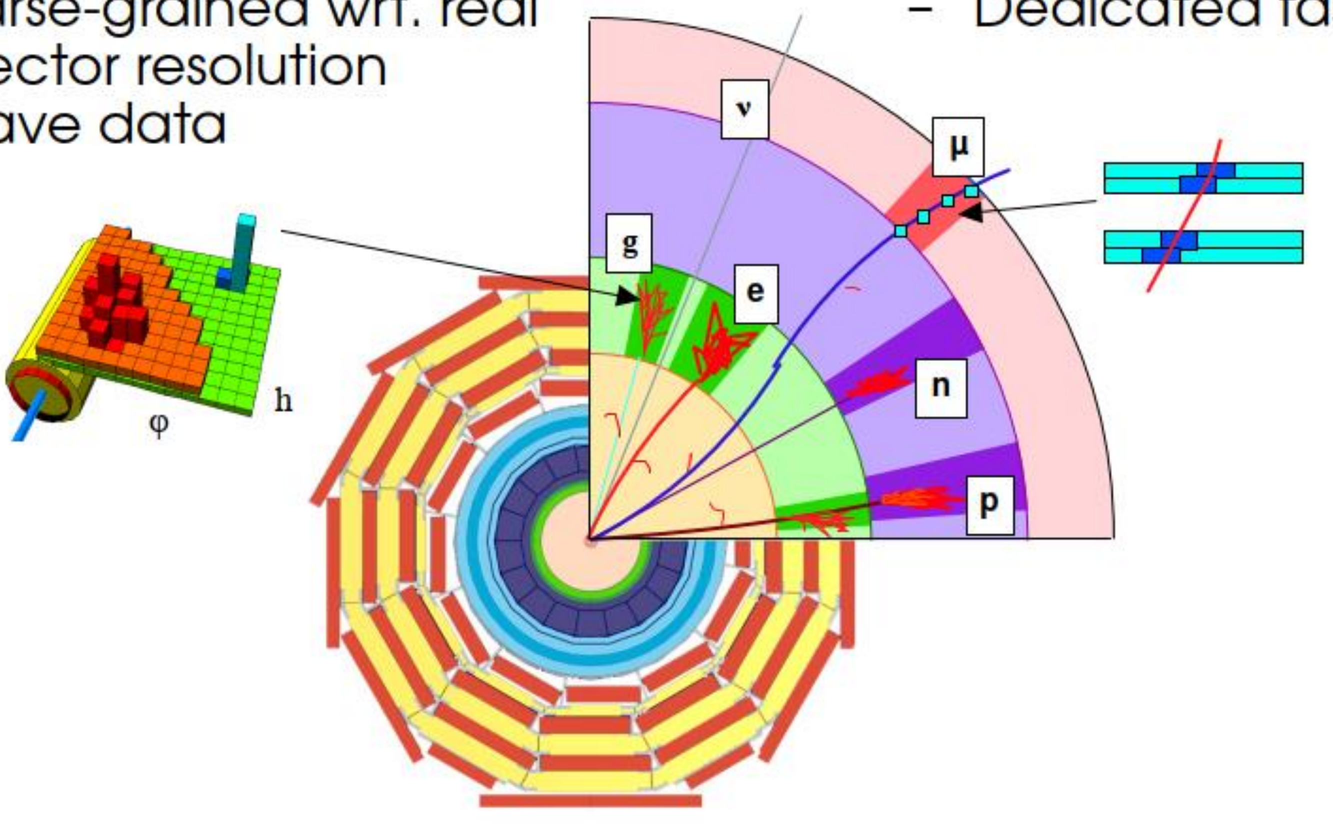
*) LHCb isn't much nicer and in Alice (PbPb) it can be even worse

Trigger Levels

- Since the detector data is not promptly available and the trigger function is highly complex, it is evaluated by successive approximations:
  - Hardware trigger(s):
    - *Fast* trigger, uses data only from few detectors
    - has a limited time budget
    - Normally implemented using custom electronics
    - ➡ Level 1, Sometimes Level 2
  - Software trigger(s):
    - Refines the decisions of the hardware trigger by using more detailed data and more complex algorithms.
    - It is usually implemented using processors running a program.
    - ➡ High Level Triggers (HLT)

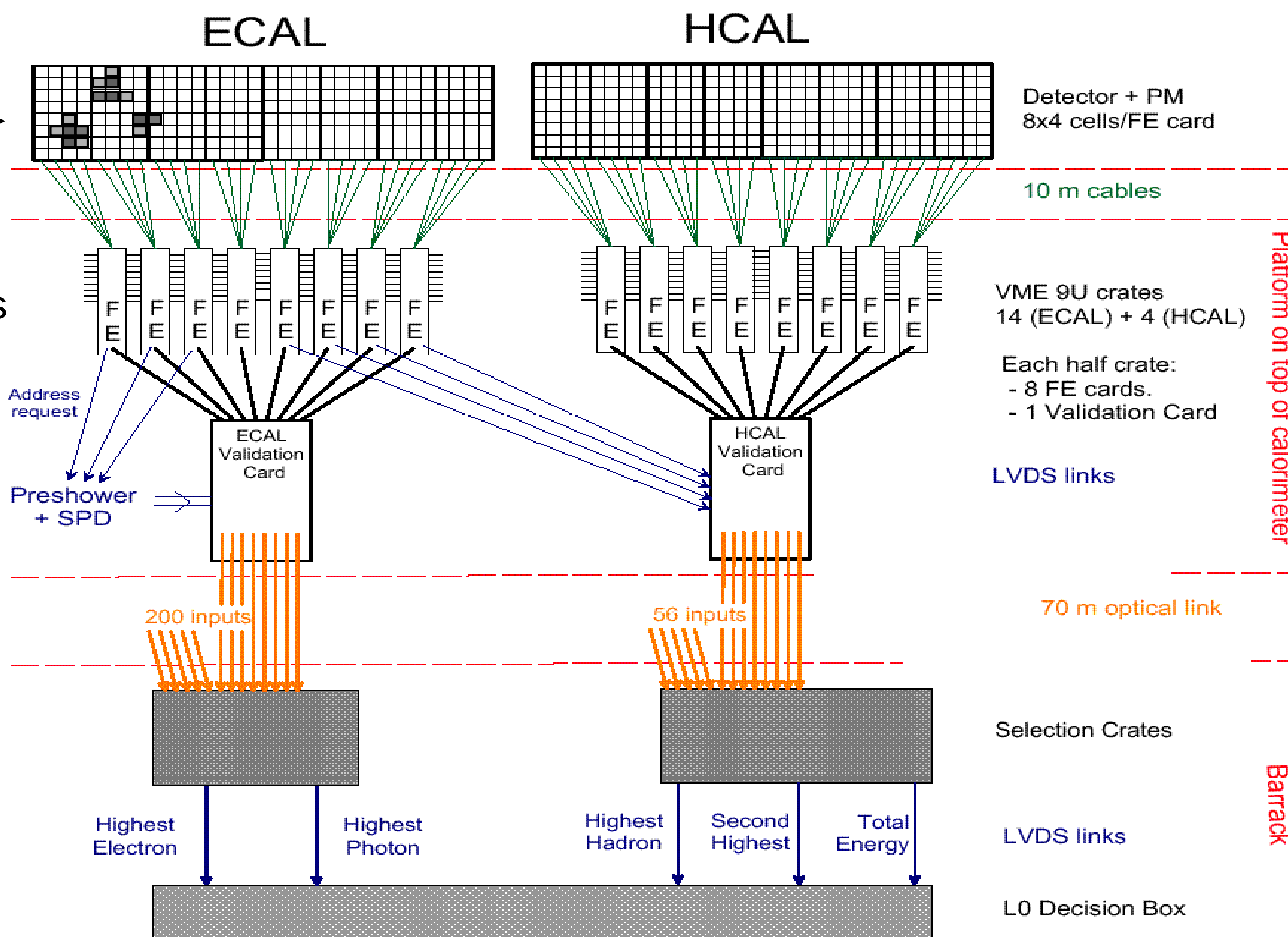
Hardware Trigger

- Calorimeters
  - Cluster finding
  - Energy deposition evaluation
  - Coarse-grained wrt. real detector resolution to save data
- Muon systems
  - Track finding
  - Momentum evaluation
  - Dedicated fast sensors

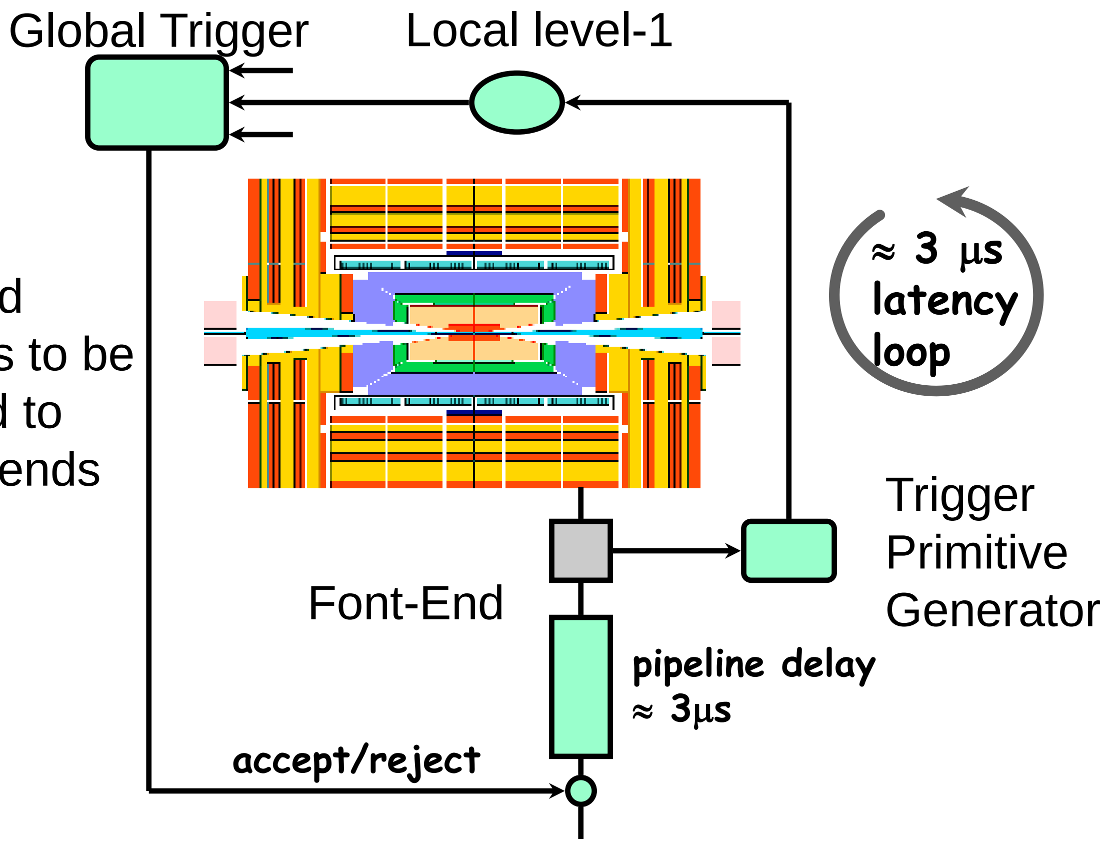


Example: LHCb Calorimeter Trigger

- Detect a high energy in a small surface
- Don't forget the neighbors
- Aggregate information
- Evaluate decision(s)



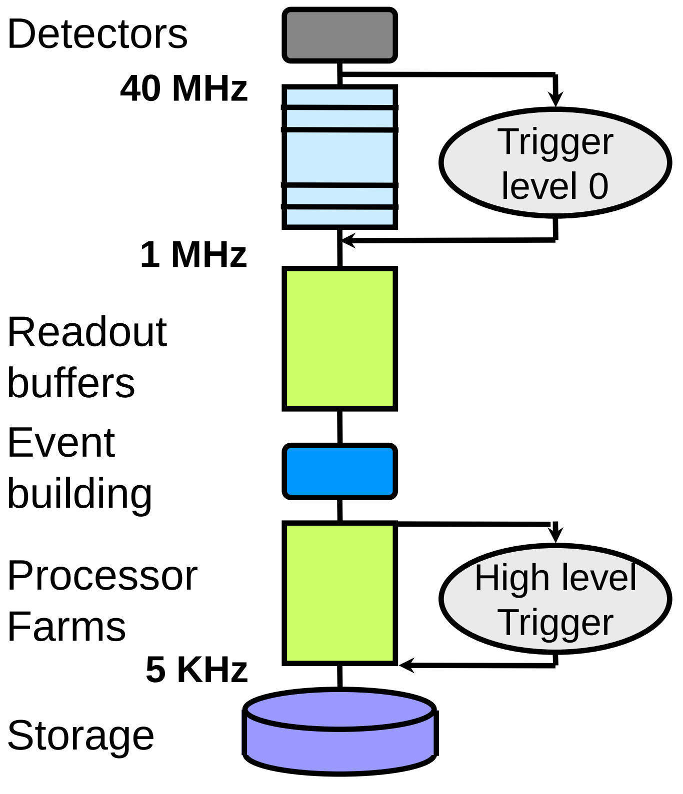
LHC: Trigger communication loop



Timing and trigger has to be distributed to ALL front-ends

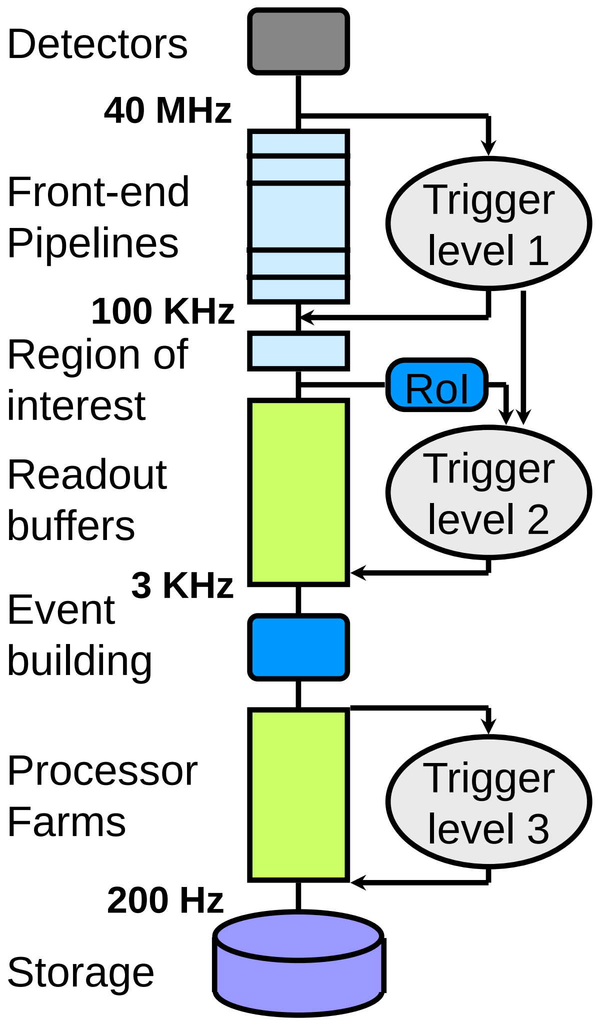
- **40 MHz synchronous digital system**
- **Synchronization at the exit of the pipeline non trivial.**
  ⇒ Timing calibration

Trigger Levels in LHCb (up to 2020)



- Level-0 (4 μs) (custom processors)
  - High p_T for electrons, muons, hadrons
- HLT (≈ms) (commercial processors)
  - Refinement of the Level-1. Background rejection.
  - Event reconstruction. Select physics channels.
  - Needs the full event

Trigger Levels in ATLAS

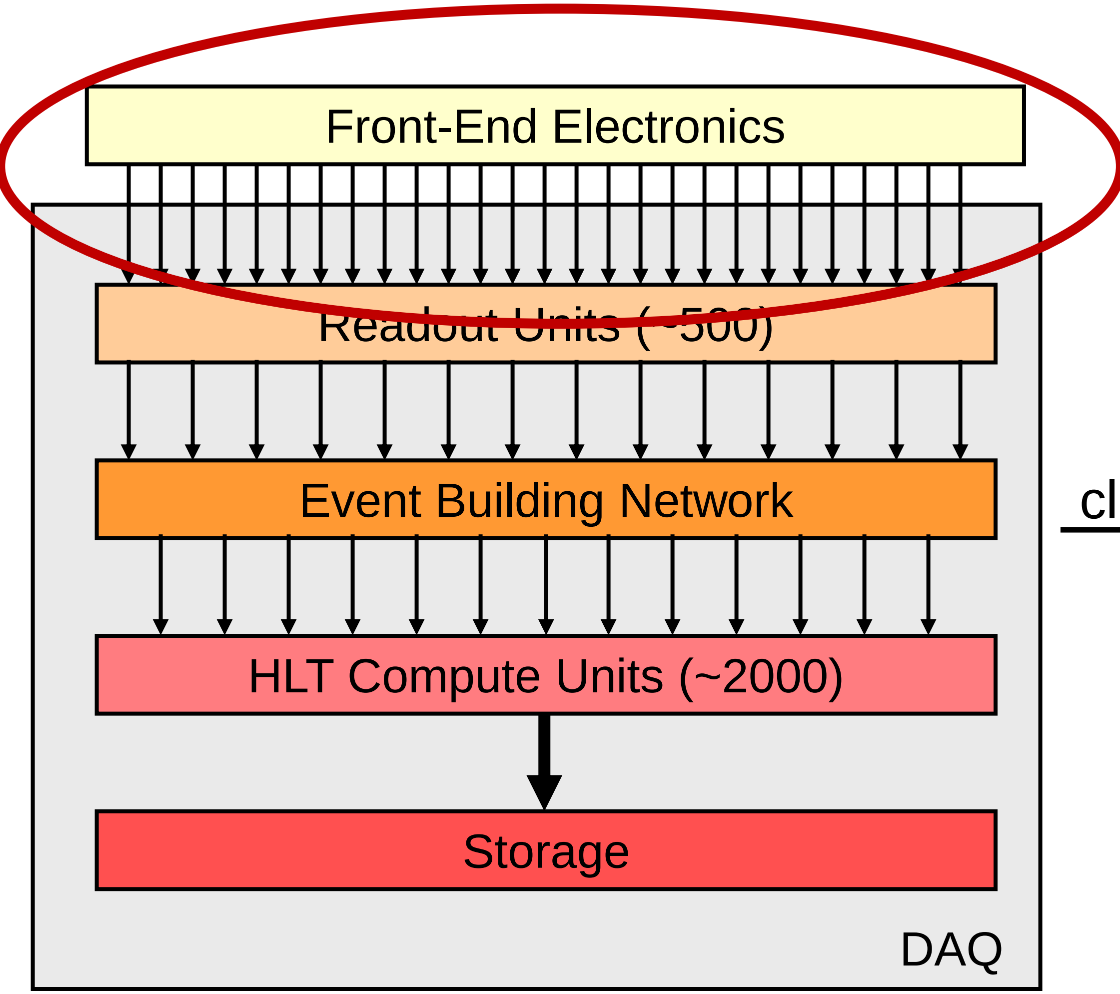


- Level-1 (3 µs) (custom processors)
  - Energy clusters in calorimeters
  - Muon trigger: tracking coincidence matrix.
- Level-2 (≈ms) (~commercial processors)
  - Few Regions Of Interest relevant to trigger decisions
  - Selected information (ROI) by routers and switches
  - More sophisticated algorithms on full-granularity data
- Level-3 (≈s) (commercial processors)
  - Reconstructs the event using all data
  - Selection of interesting physics channels

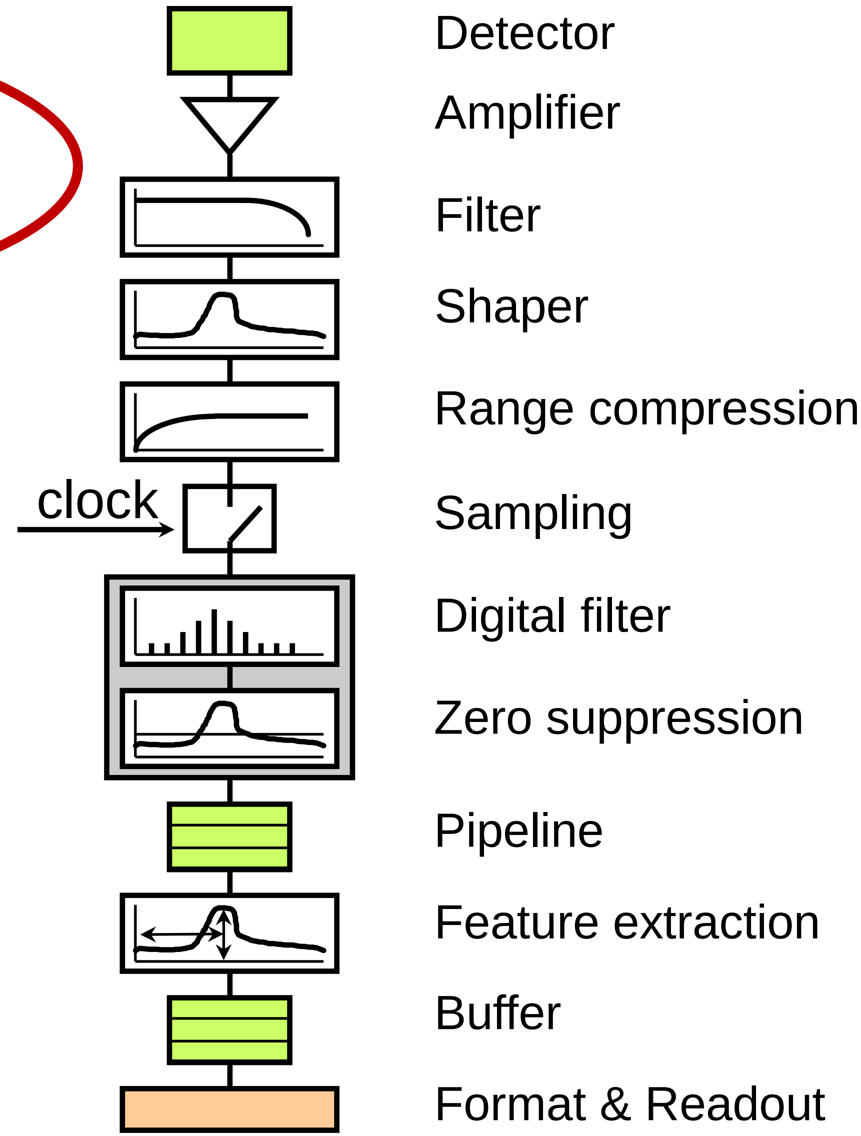
Data Acquisition

Clara Gaspar, July 2023

Front-end electronics



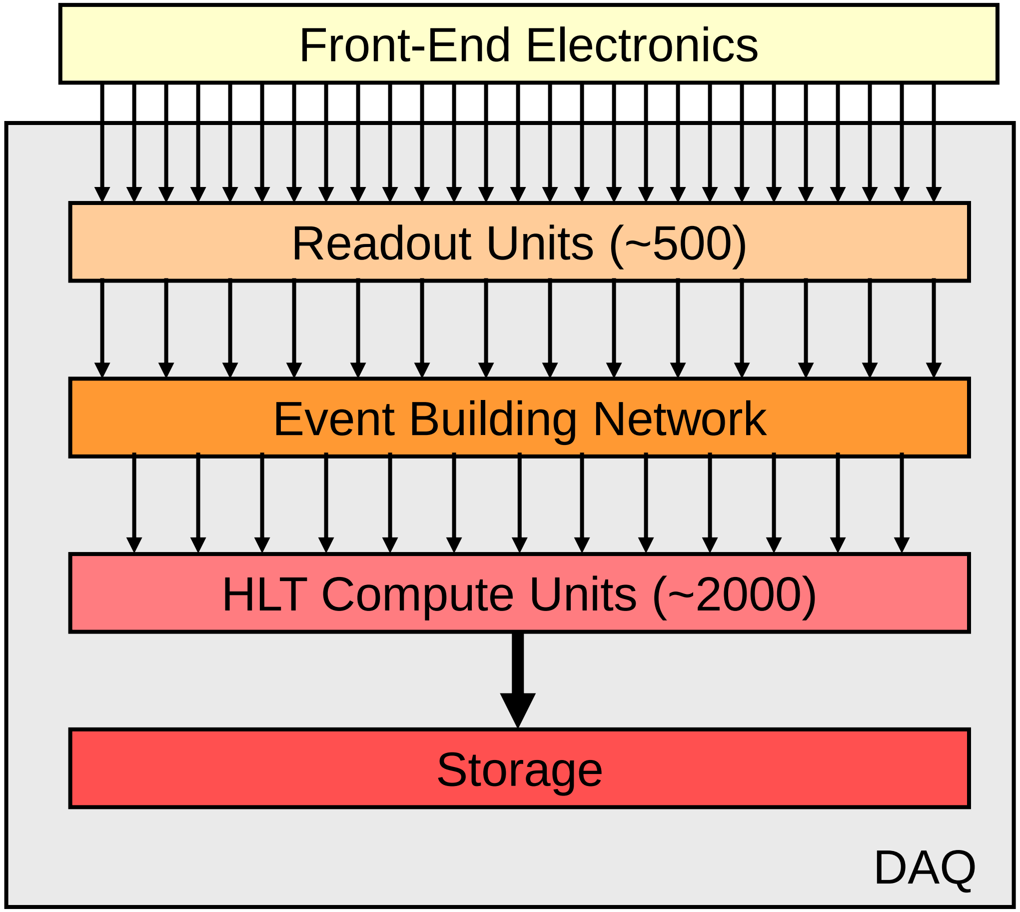
Custom point-to-point radiation-hard links



Front-End Electronics

- **Detector dependent (Home made)**
  - On Detector
    - Pre-amplification, Discrimination, Shaping amplification and Multiplexing of a few channels
    - **Problems: Radiation levels, power consumption**
  - Transmission
    - Long Cables (50-100 m), electrical or fiber-optics
  - In Counting Rooms
    - Hundreds of FE crates :
      Reception, A/D conversion and Buffering

Data Acquisition proper



- Every Readout Unit has a piece of the collision data tagged with a unique id
- All pieces must be brought together into a single Compute unit
- The Compute Unit runs the software filtering (High Level Trigger)

Event Building to a CPU farm

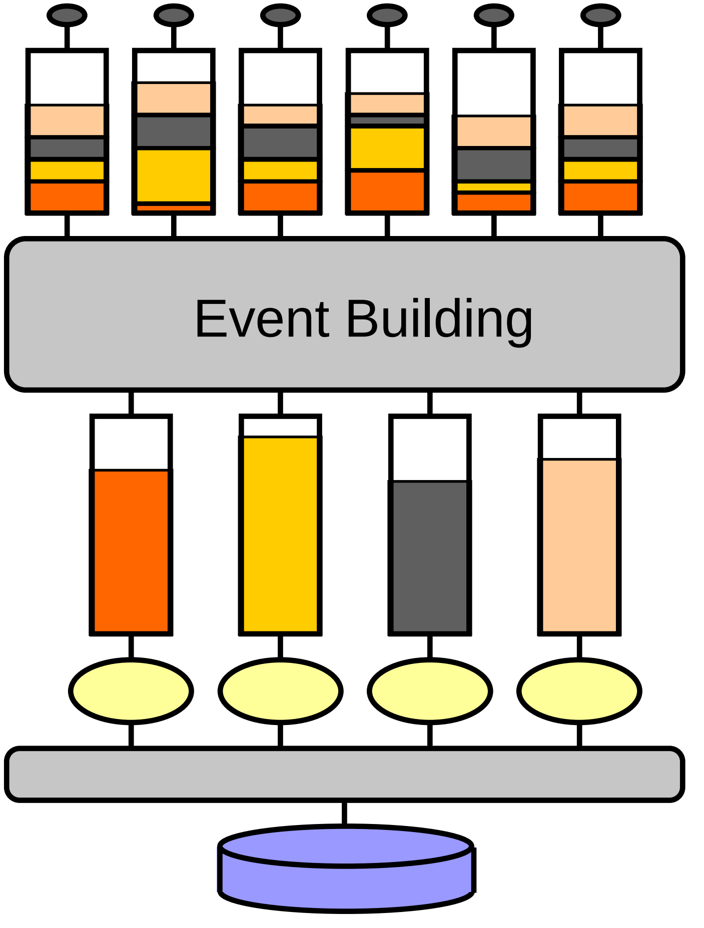
Data sources

Event Fragments

Full Events

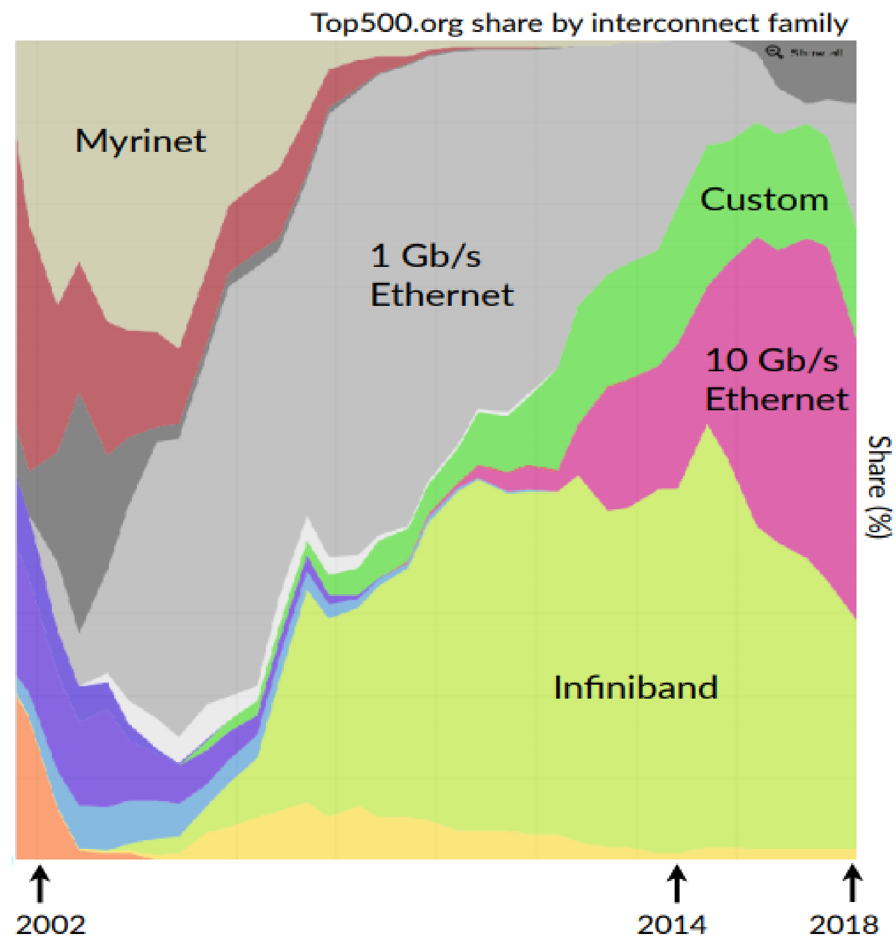
Event filter CPUs

Data storage



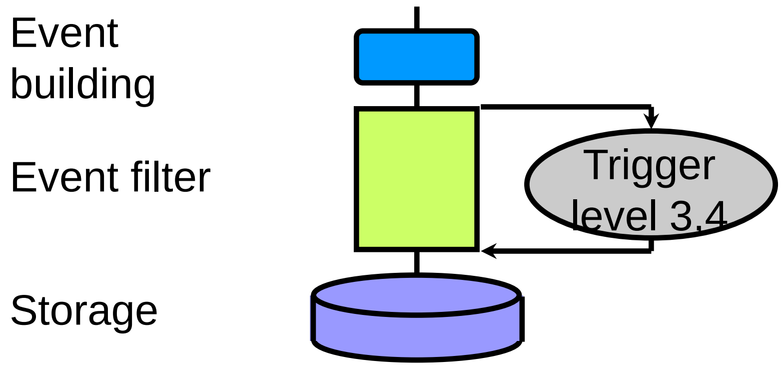
Network Technology

- Need to use what is popular -> best price
- Myrinet very popular when LHC experiments started being designed
  Used by CMS until 2014
- Others could postpone the decision and used Ethernet



Event Filters

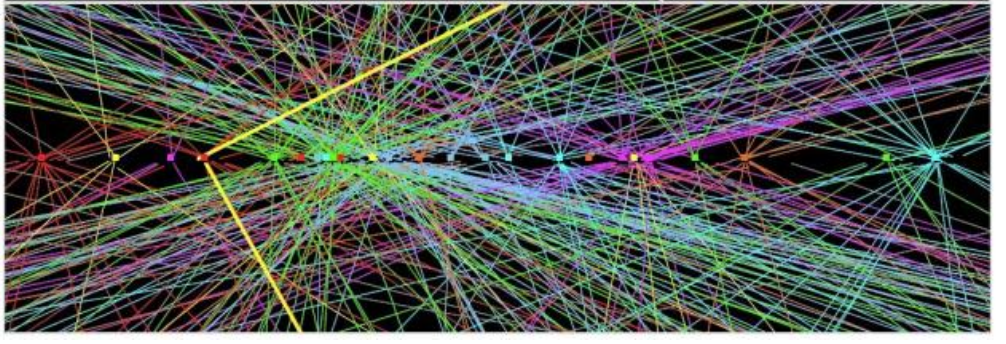
- **Higher level triggers (3, 4, …)**



- LHC experiments can not afford to write all acquired data into mass-storage. -> Only useful events should be written to the storage
- The event filter function selects events that will be used in the data analysis. Selected physics channels.
- Uses commercially available processors (common PCs) But needs thousands of them running in parallel.

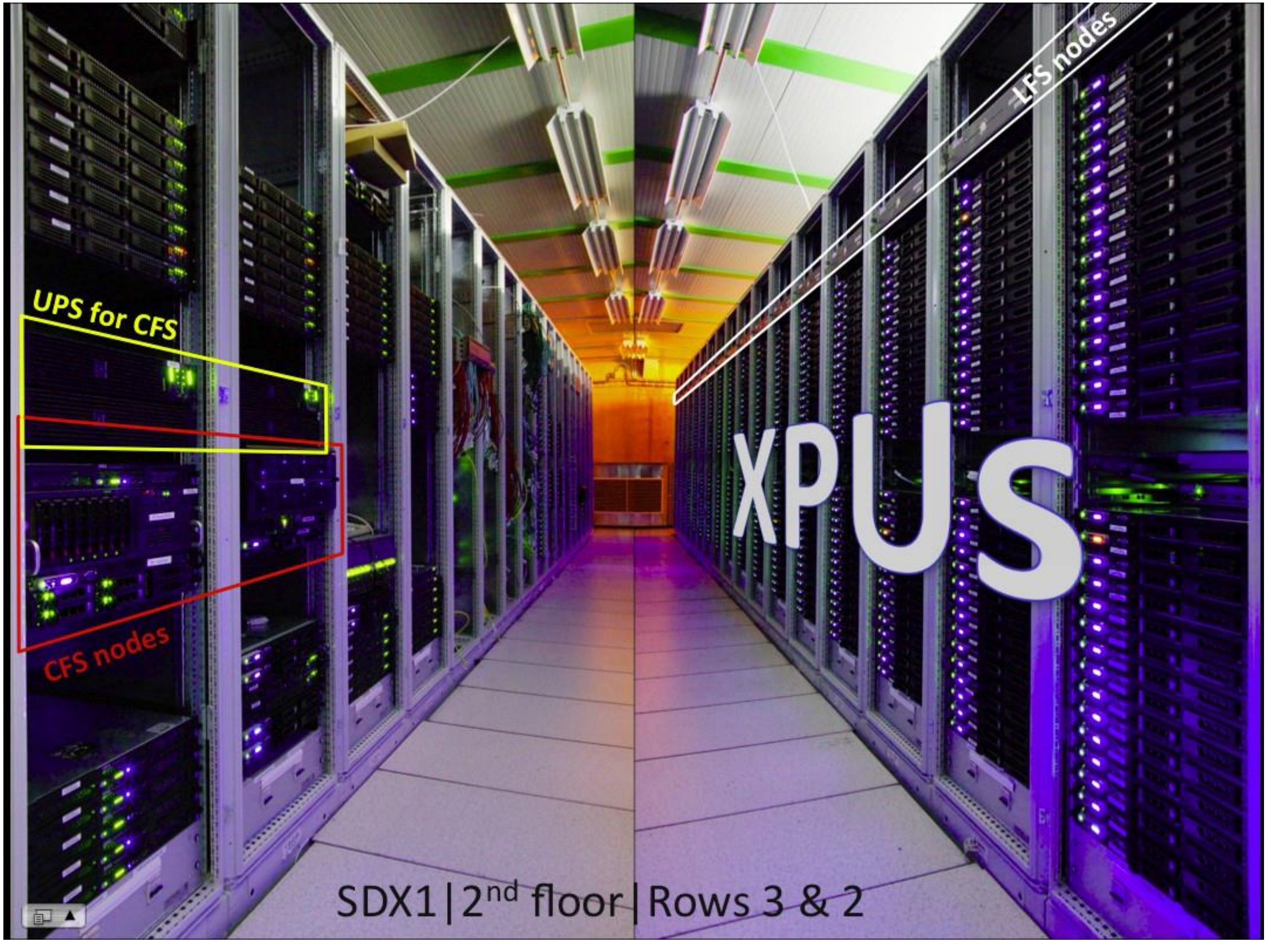
High Level Trigger

- **Full Event “Reconstruction”**
  - Reconstruct all charged particle trajectories
    - Find segments, connect them, re-fit to physical trajectory
  - Associate the particles with the correct p-p collision
    - Multiple interactions in each crossing

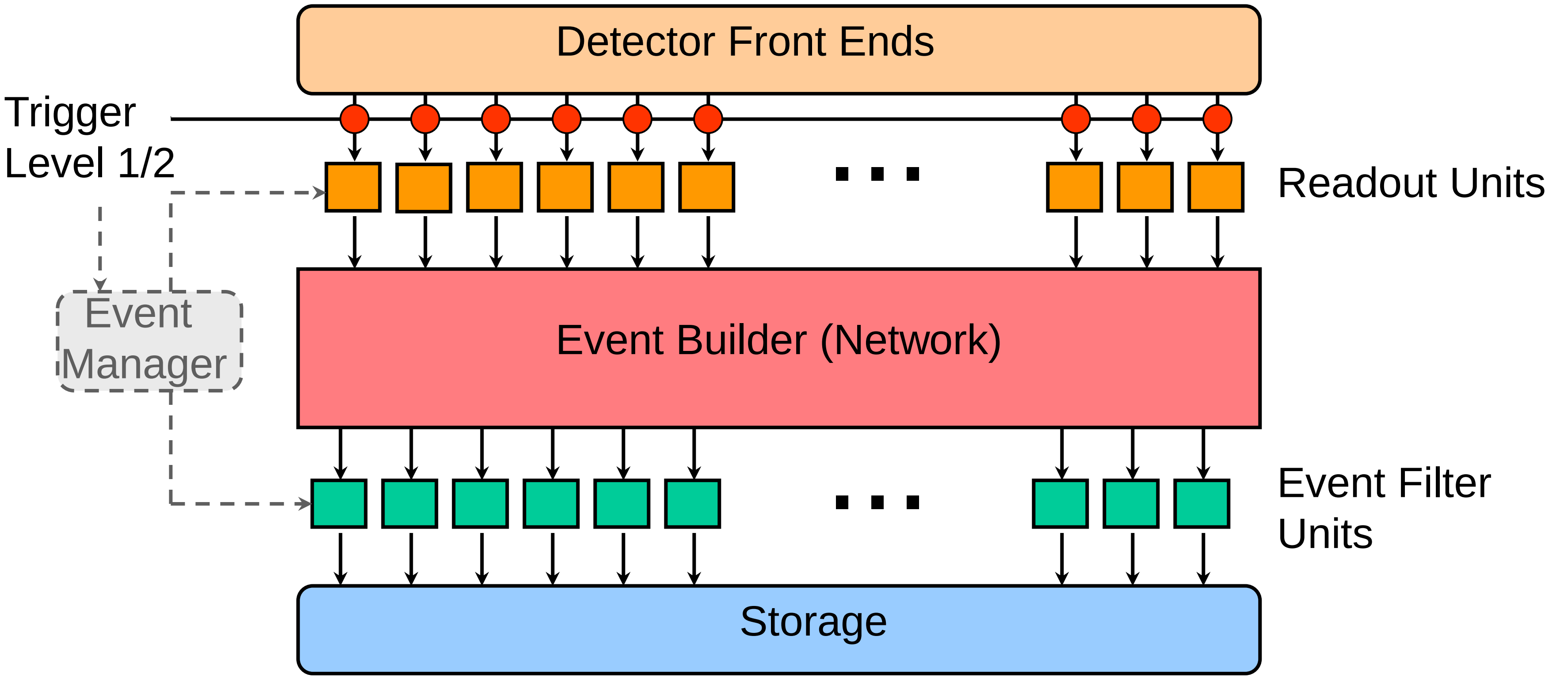


  - Measure all the energy depositions in the calorimeters
    - with fine granularity
  - Associate tracks and energy depositions
- **Decide whether it’s interesting**

ATLAS HLT Farm

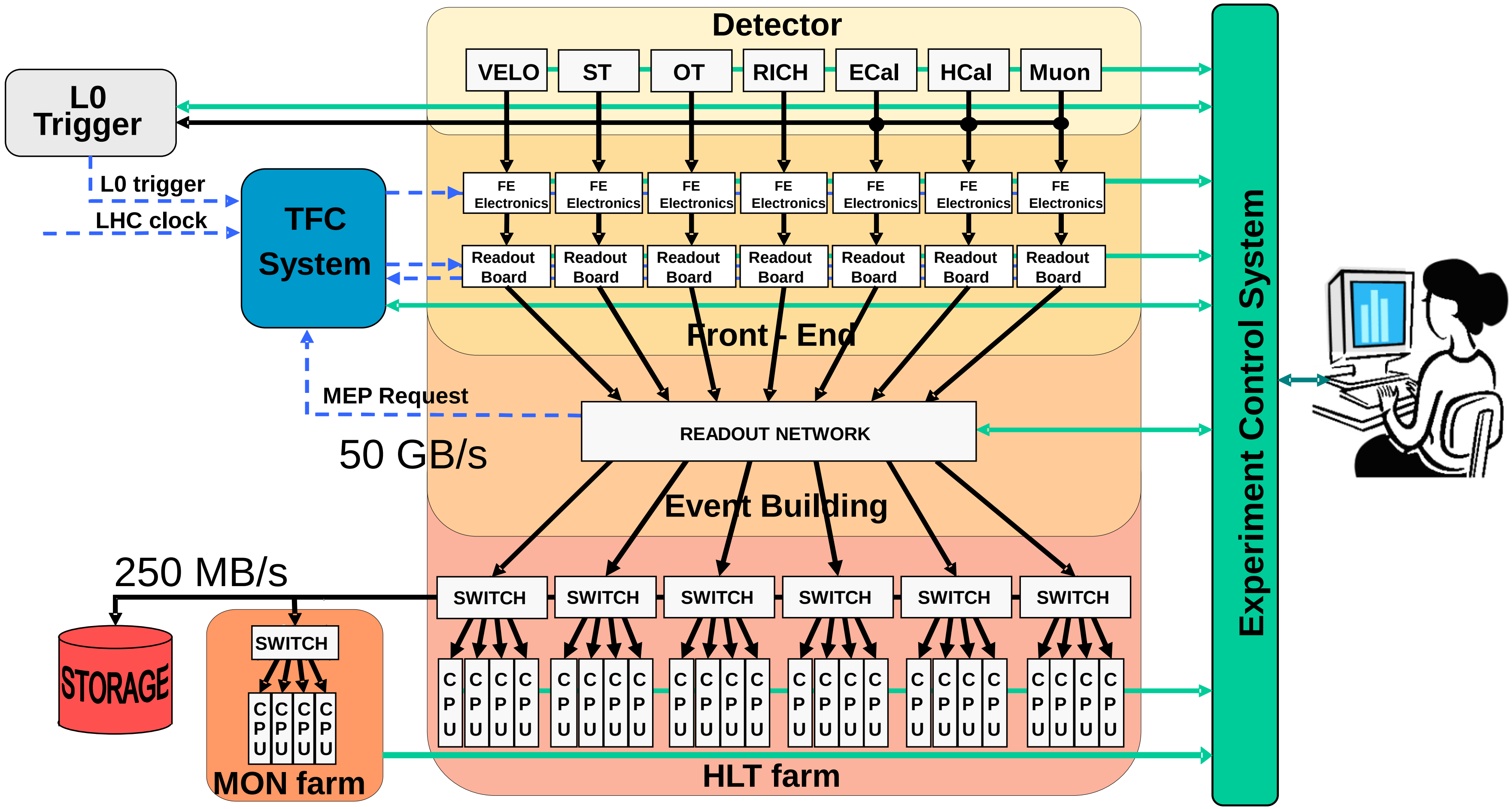


LHC Readout Architecture



- With or Without Event Manager
- Push or Pull Protocols

LHCb DAQ (up to 2020)



- Event data
- Timing and Fast Control Signals
- Control and Monitoring data

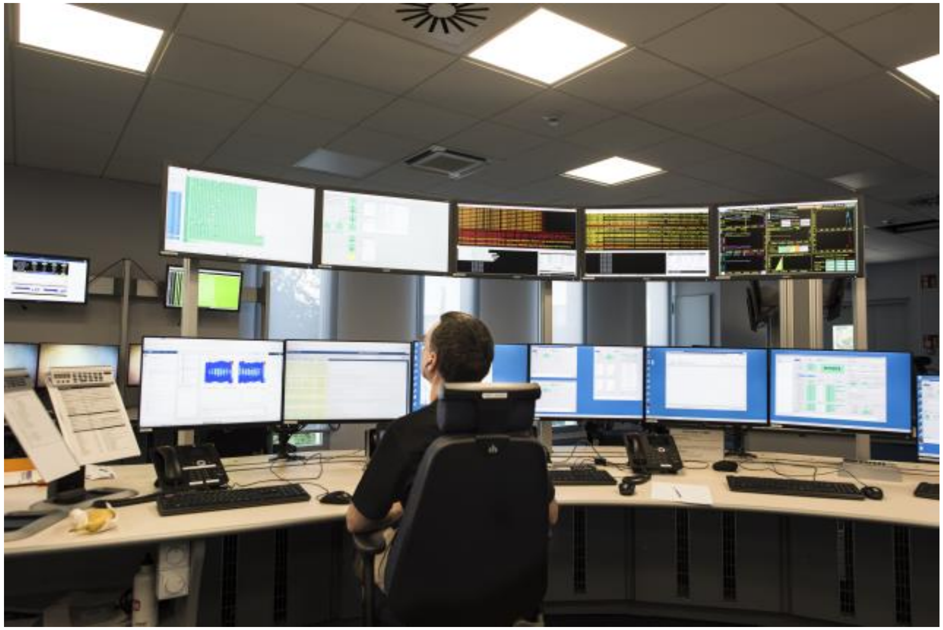
Average event size 50 kB
Average rate into farm 1 MHz
Average rate to tape 5 kHz

Operations

Clara Gaspar, July 2023

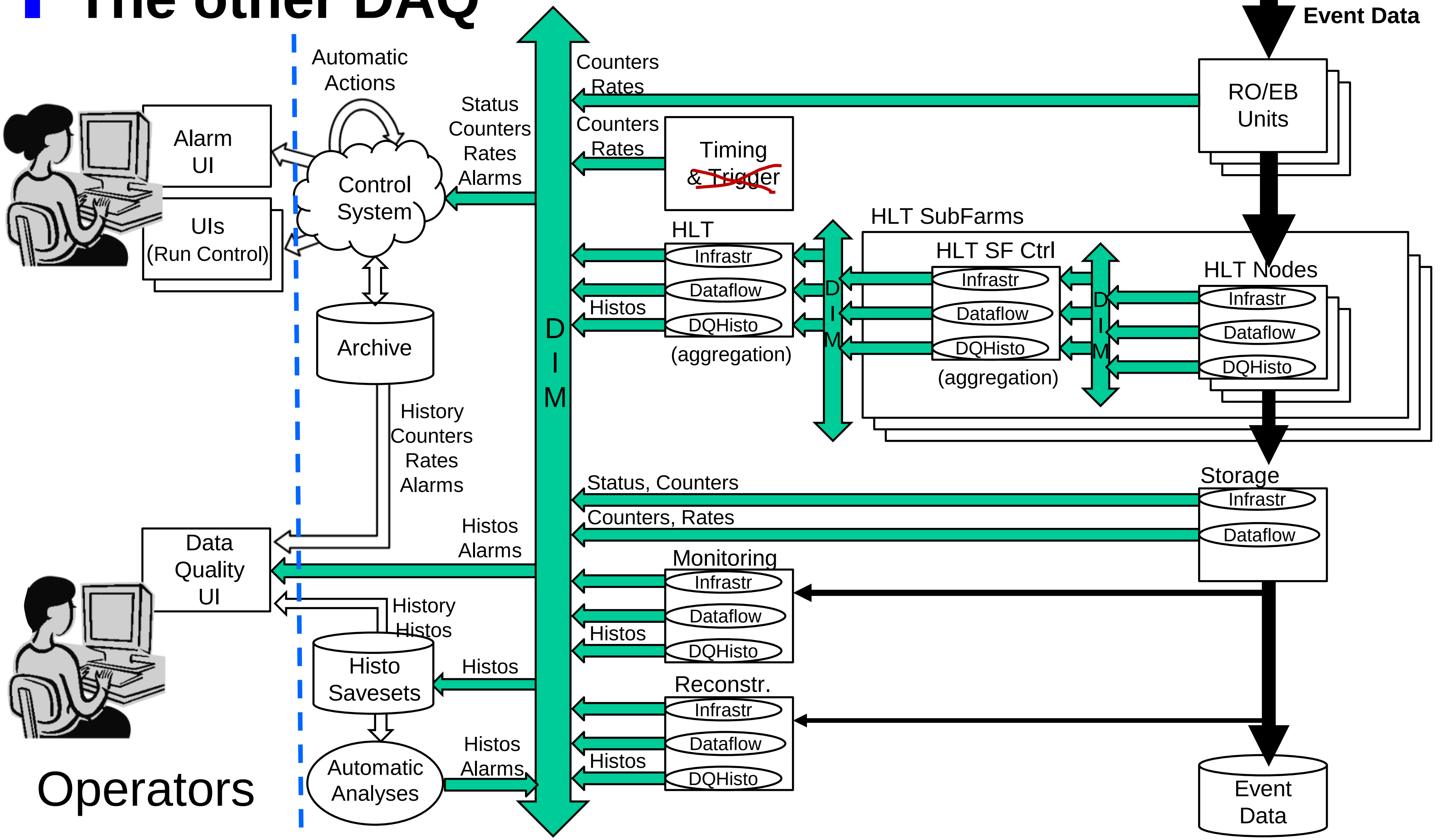
Operations

- **Experiments run 24/24 7/7**
- **In LHCb Only 2 (non-expert) operators**
- **Monitoring**
  - Detect any problems as soon as possible
- **Configuration**
  - Prepare for a particular running mode
- **Automation**
  - Avoid human mistakes
  - Speed up standard procedures

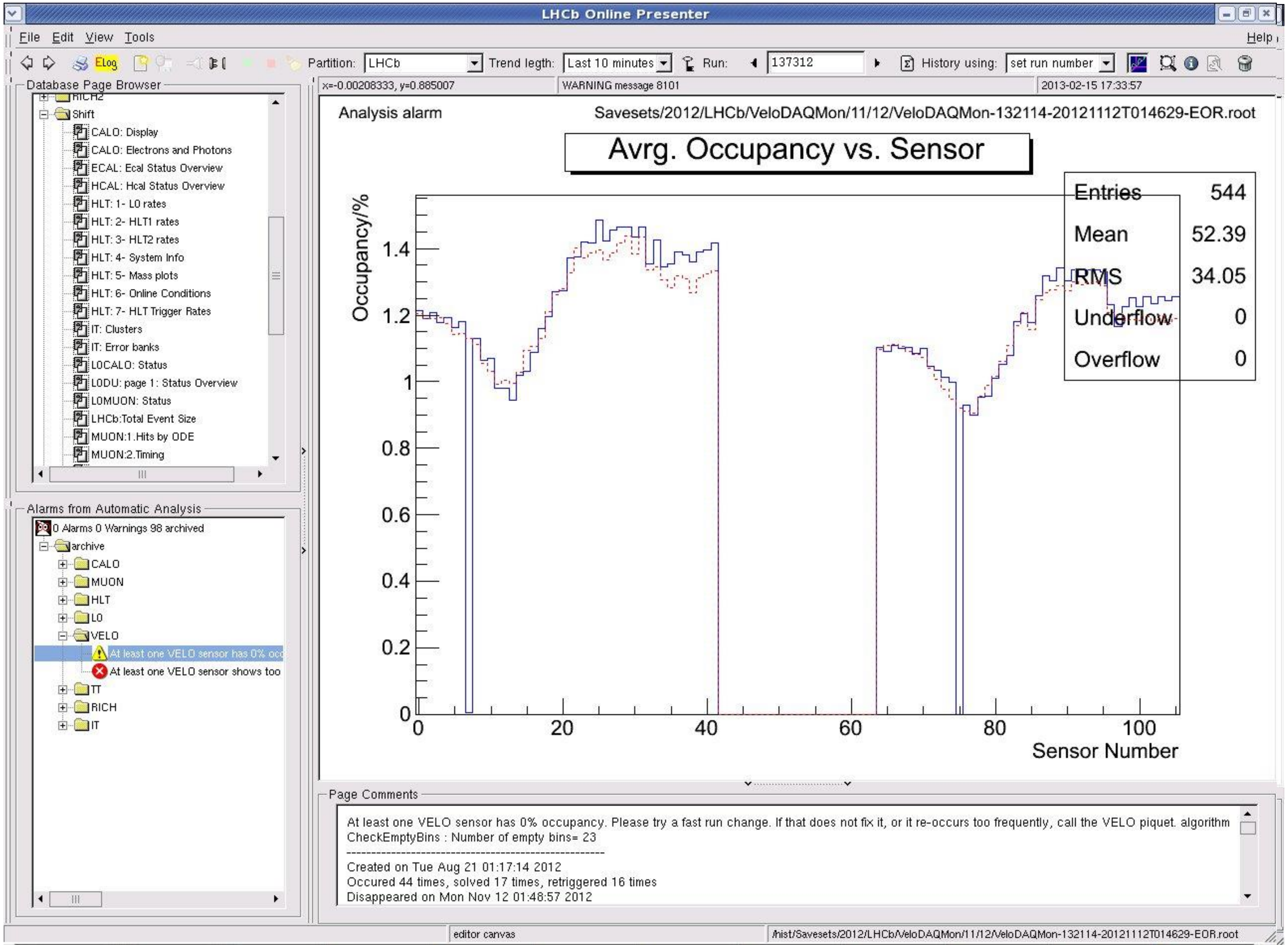


Monitoring Dataflow

- The other DAQ

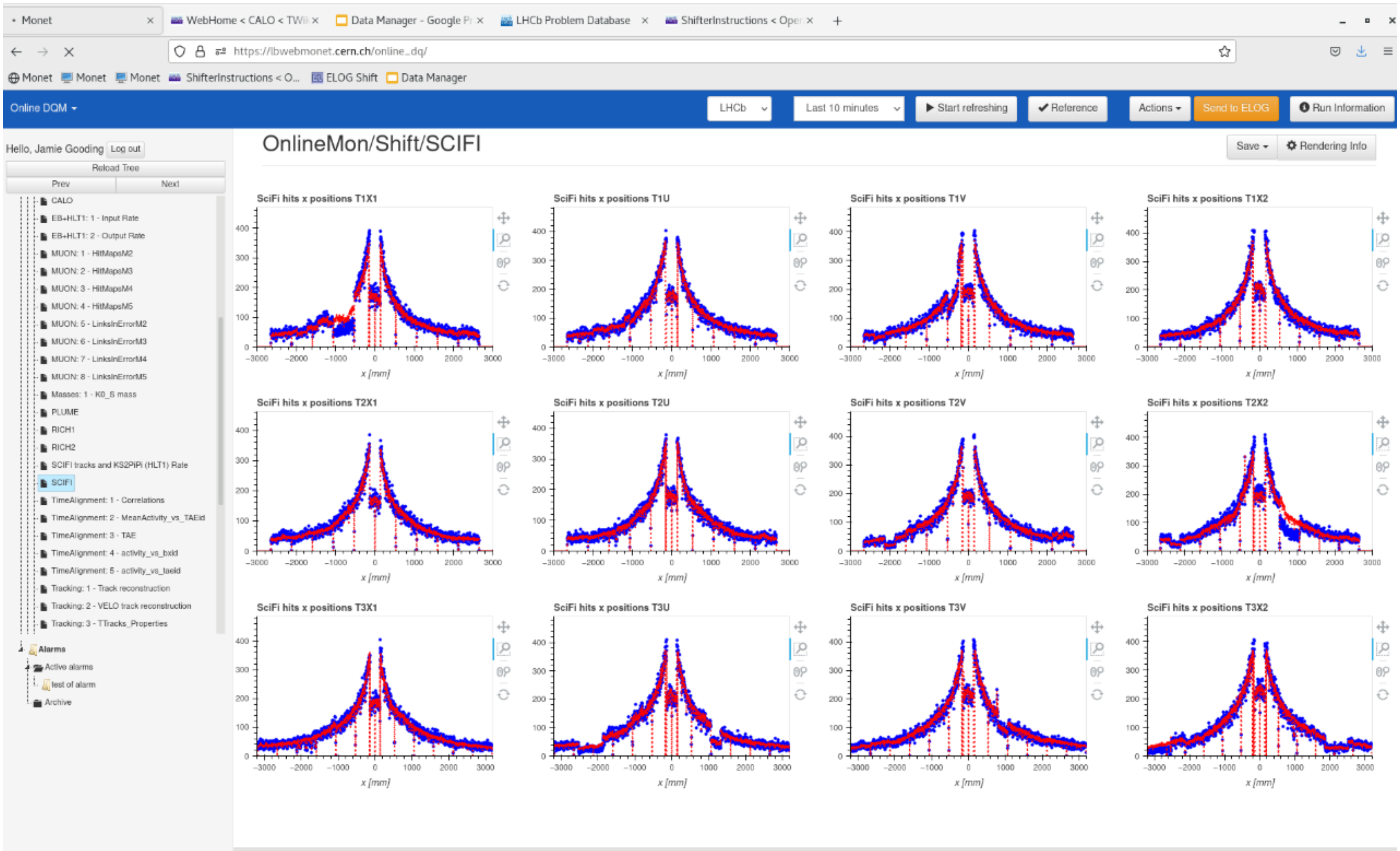


LHCb Data Quality (old)

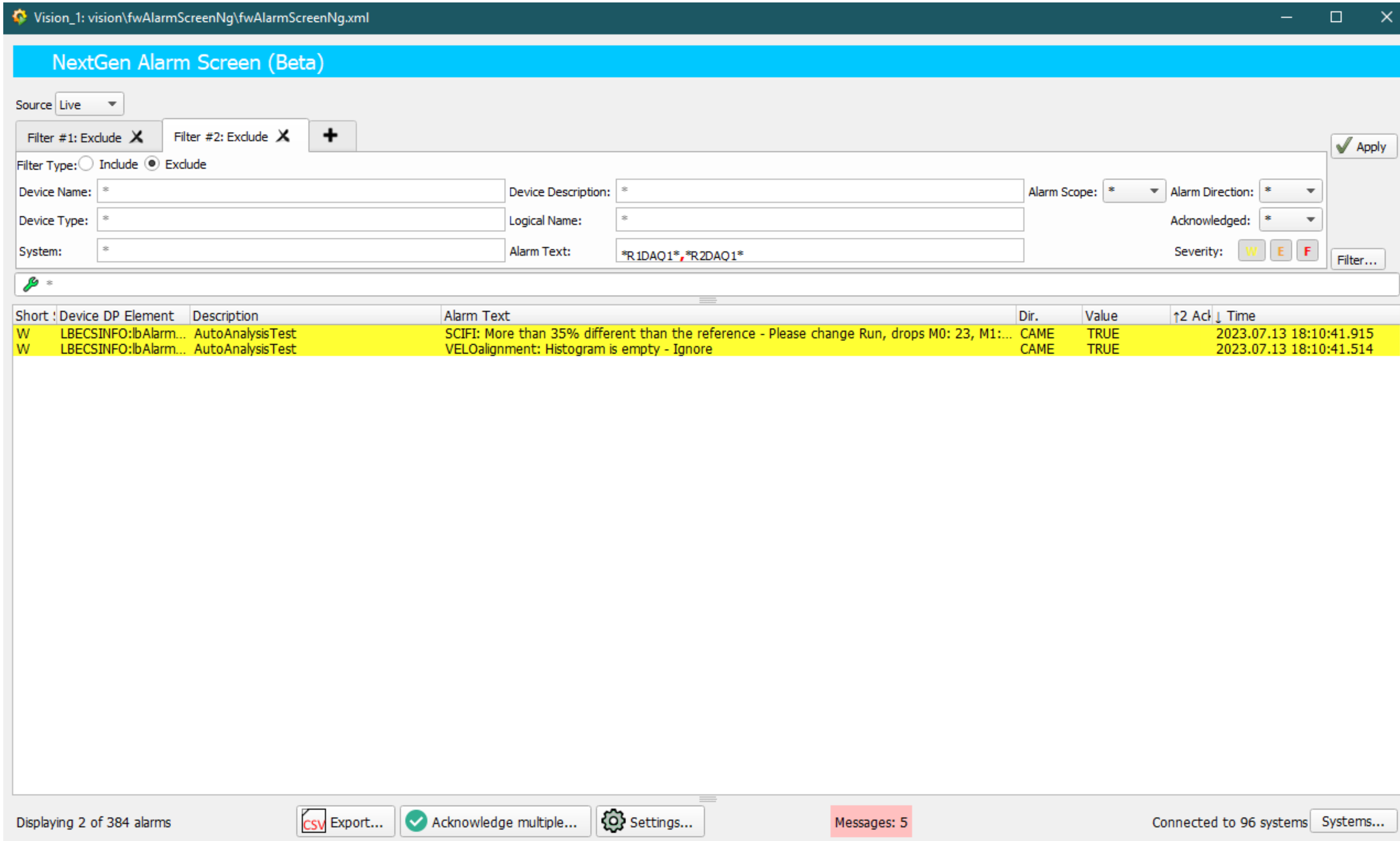


- Monitor the quality of the data being acquired
- Compare with reference
- Automatic analyses

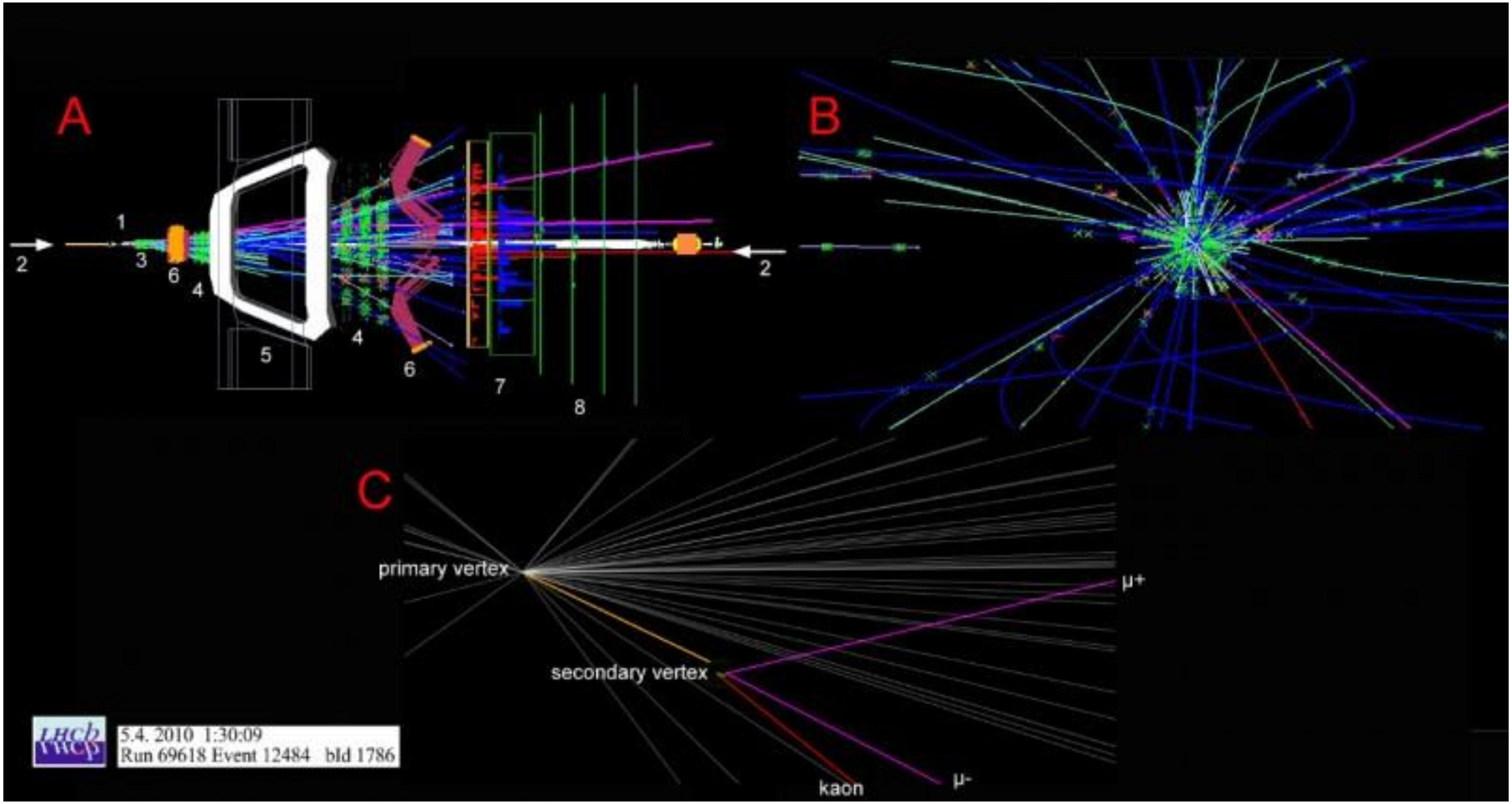
LHCb Data Quality: Monet



LHCb Alarm Screen

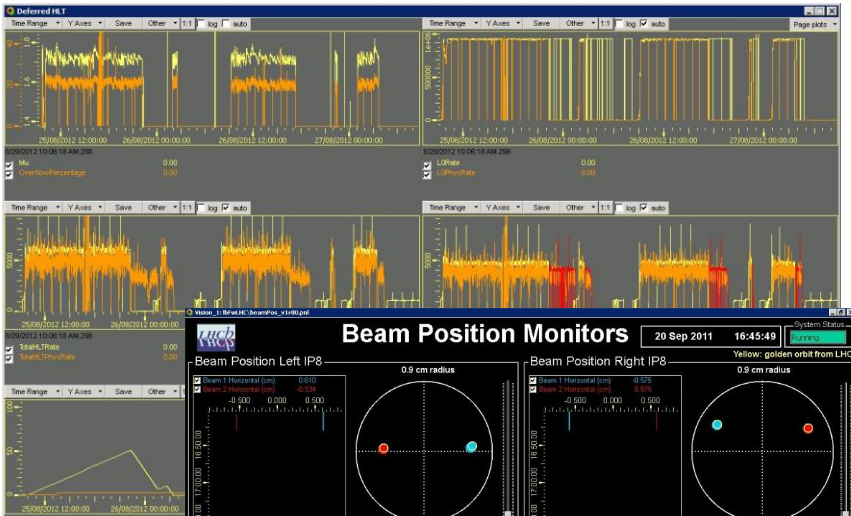


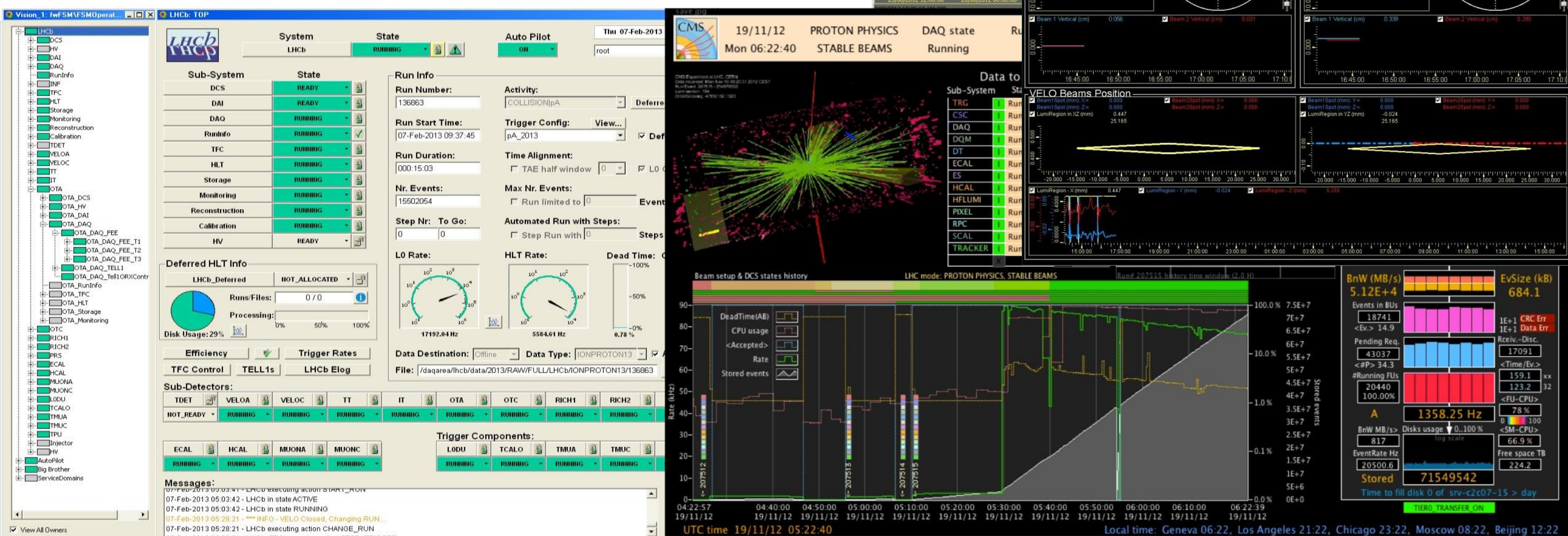
LHCb Online Event Display



Operator Interfaces

- Types of User Interfaces
  - Alarm Screens and/or Message Displays
  - Monitoring Displays
  - Run Control & DCS Control

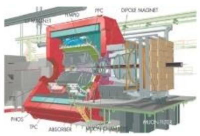
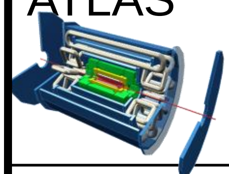
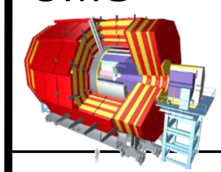
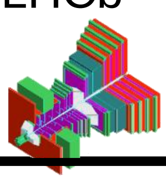




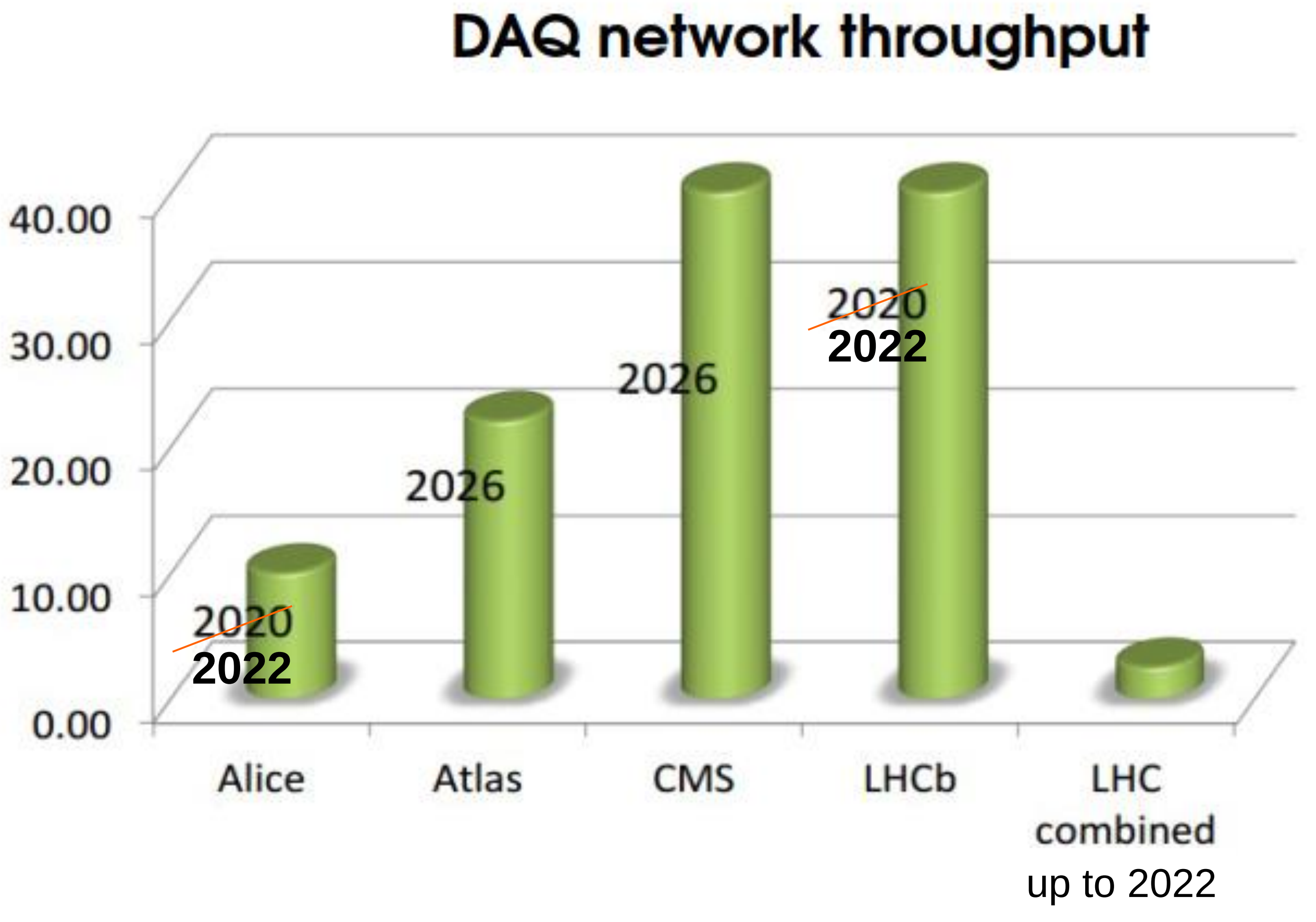
DAQ Upgrades

Clara Gaspar, July 2023

LHC DAQ Upgrades

	Up to Yesterday			Upgrade (2022-2026)		
	L1 Rate (KHz)	Event Size (MByte)	Bandwidth (GByte/s)	L1 Rate (KHz)	Event Size (MByte)	Bandwidth (GByte/s)
ALICE 	1	25	25	50	20	1000
ATLAS 	100	1	100	200	4	800
CMS 	100	1	100	1000	4	4000
LHCb 	1000	0.05	50	40000	0.1	4000

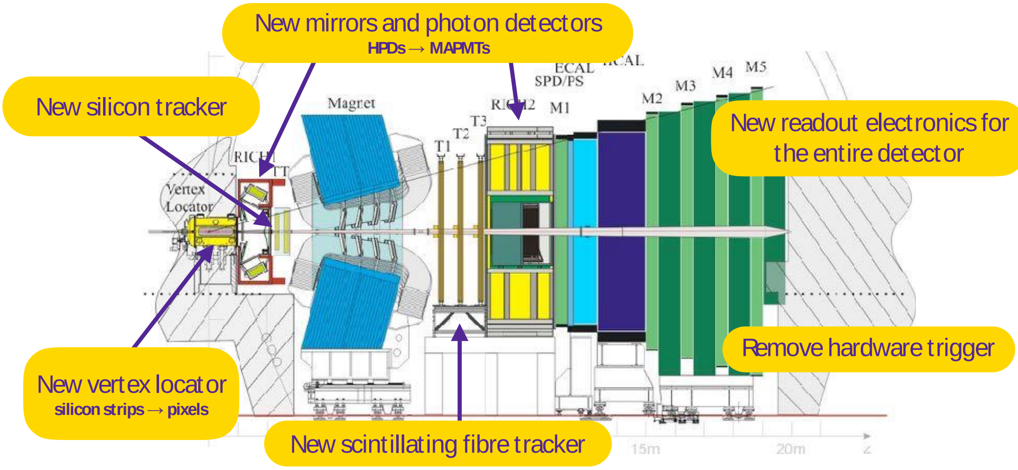
The Upgrades



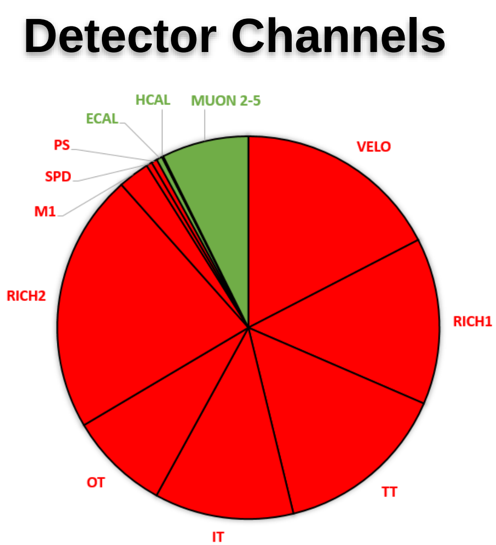
LHC Exp. Upgrades

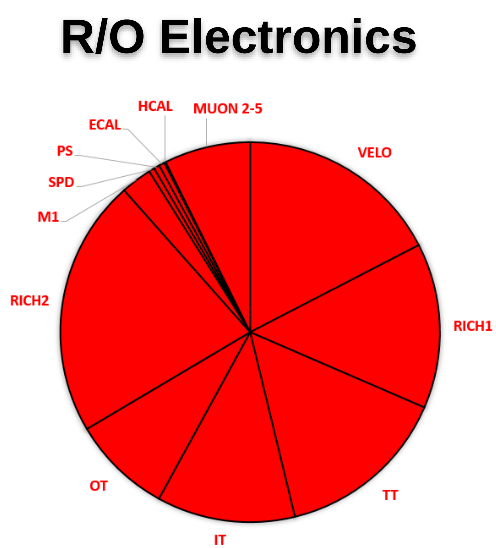
- **Reasons:**
  - More Physics/Faster Physics/Better Physics
  - (Improved Detectors)
  - Reduce impact of Hw Triggers:
    - Working with partial information and with drastic simplifications has a price:
      - Potentially interesting and valuable events are lost
    - Directions:
      - Eliminate / reduce hardware Level-1 (ALICE, LHCb)
      - Substantially upgrade Level-1 (ATLAS, CMS)
  - Emphasize Real-Time Alignment and Calibration

LHCb Upgrade



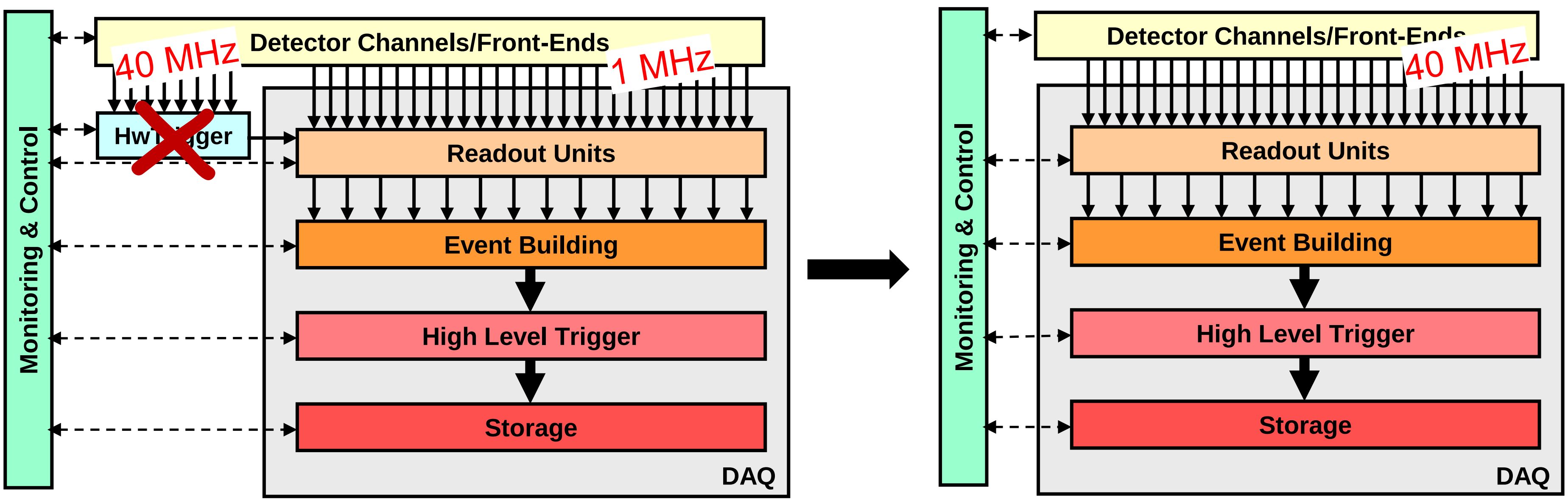
- **40 MHz Trigger-less Readout**
  - **< 10% detector channels kept**
  - **100% or R/O electronics replaced**
  - **New DAQ system**
  - **New Data Center**





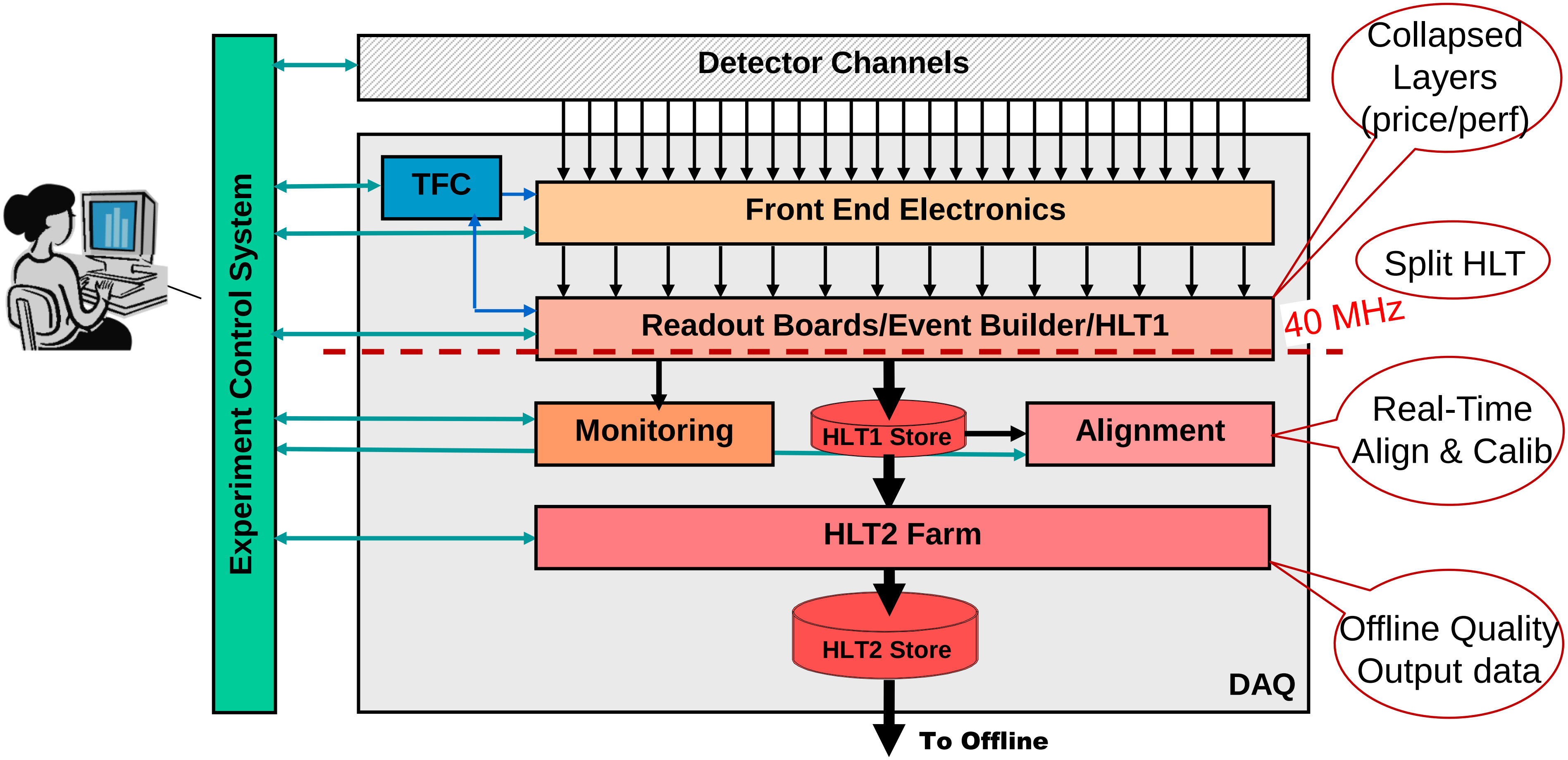
The LHCb Example

- 1st Large Physics Experiment to completely abolish the Hardware Trigger -> Probably the largest Data Acquisition System in the world today

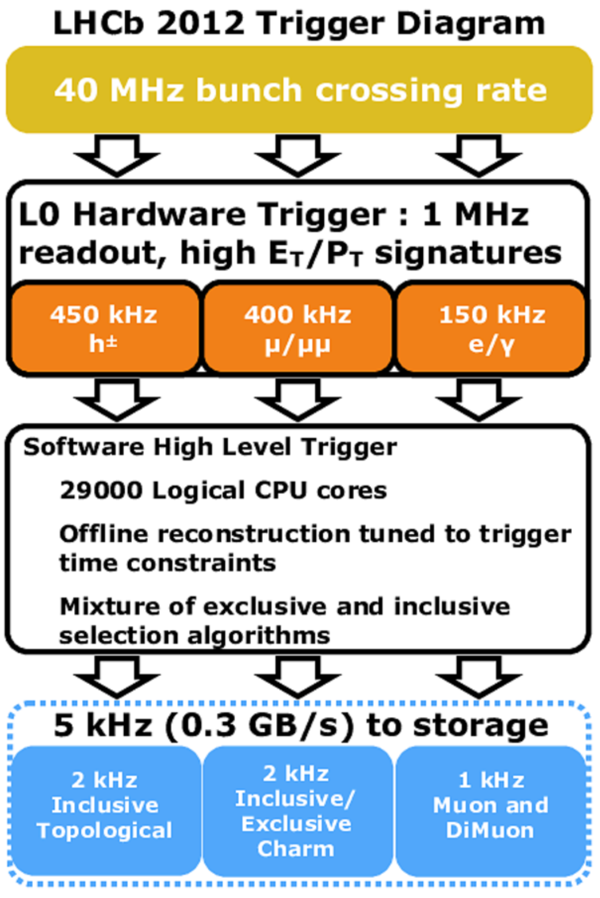


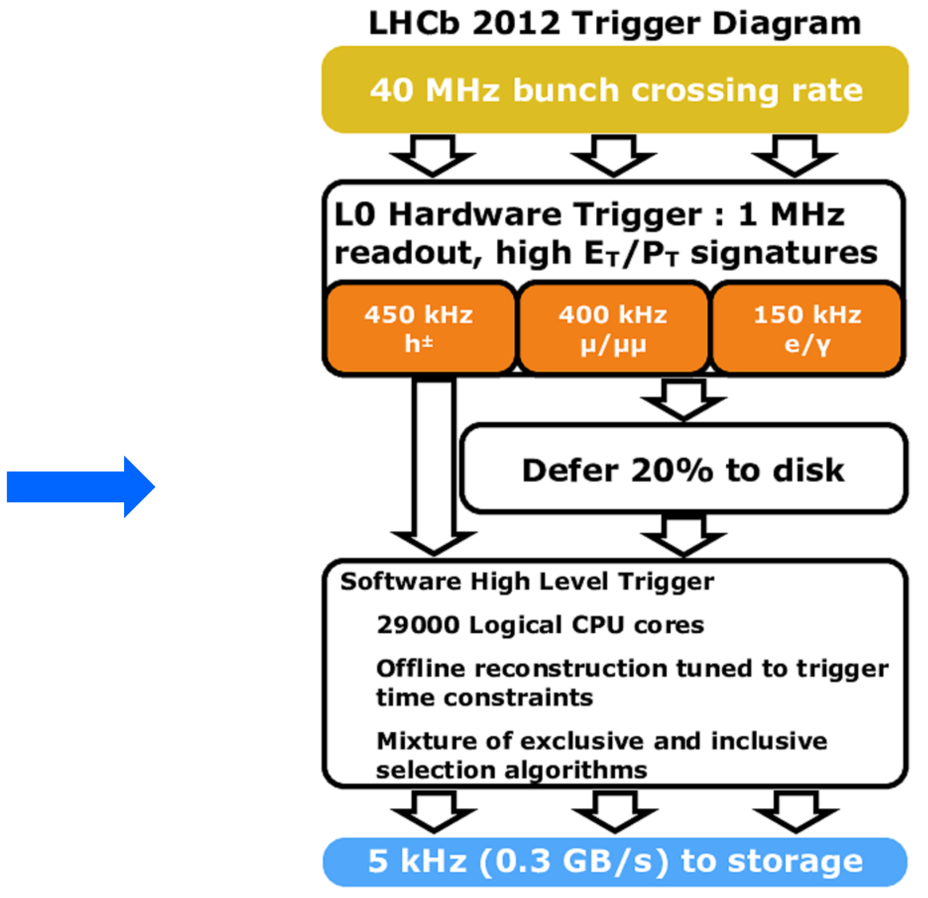
- Spoiler: Commissioned last year -> And it works!!

The New LHCb DAQ



LHCb Dataflow Evolution

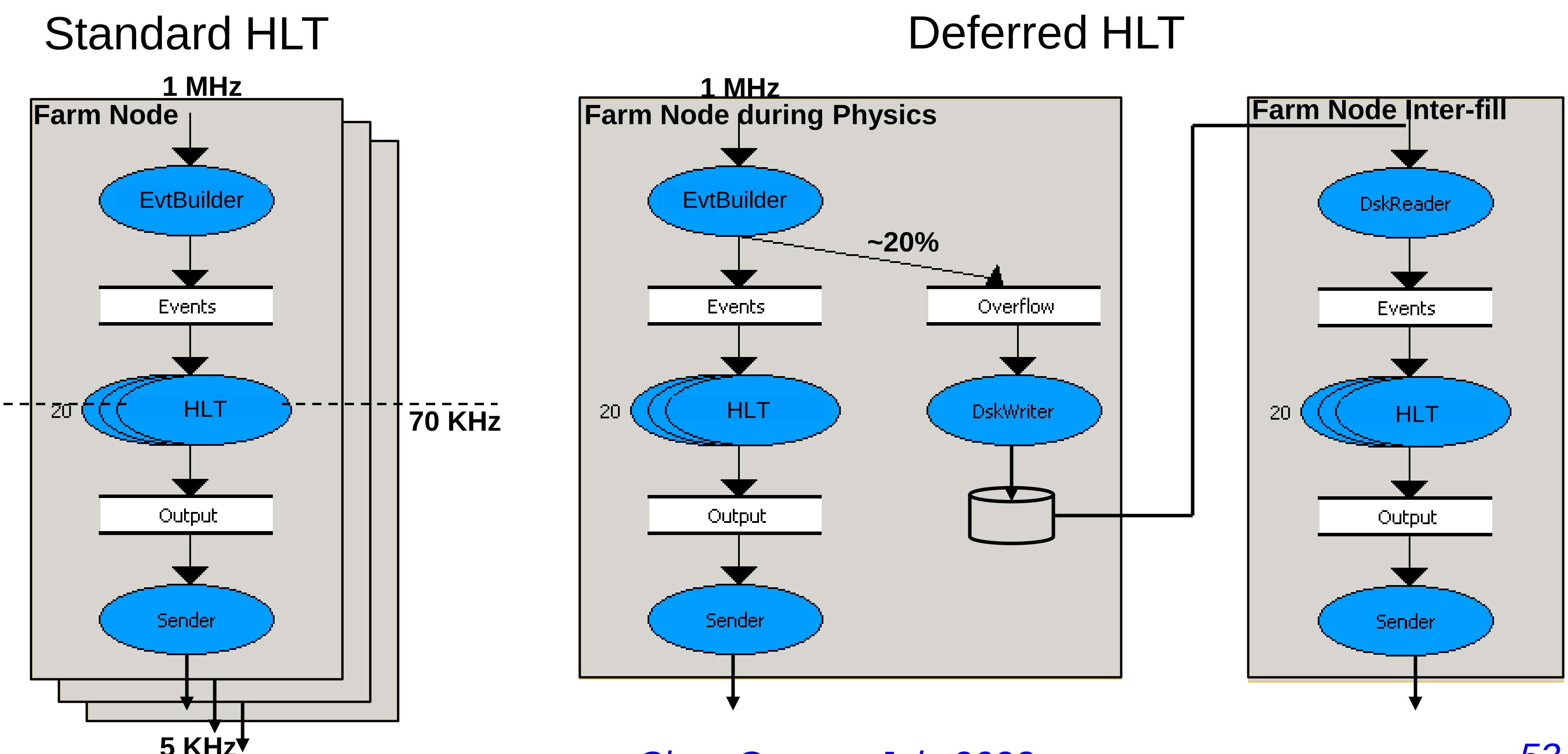




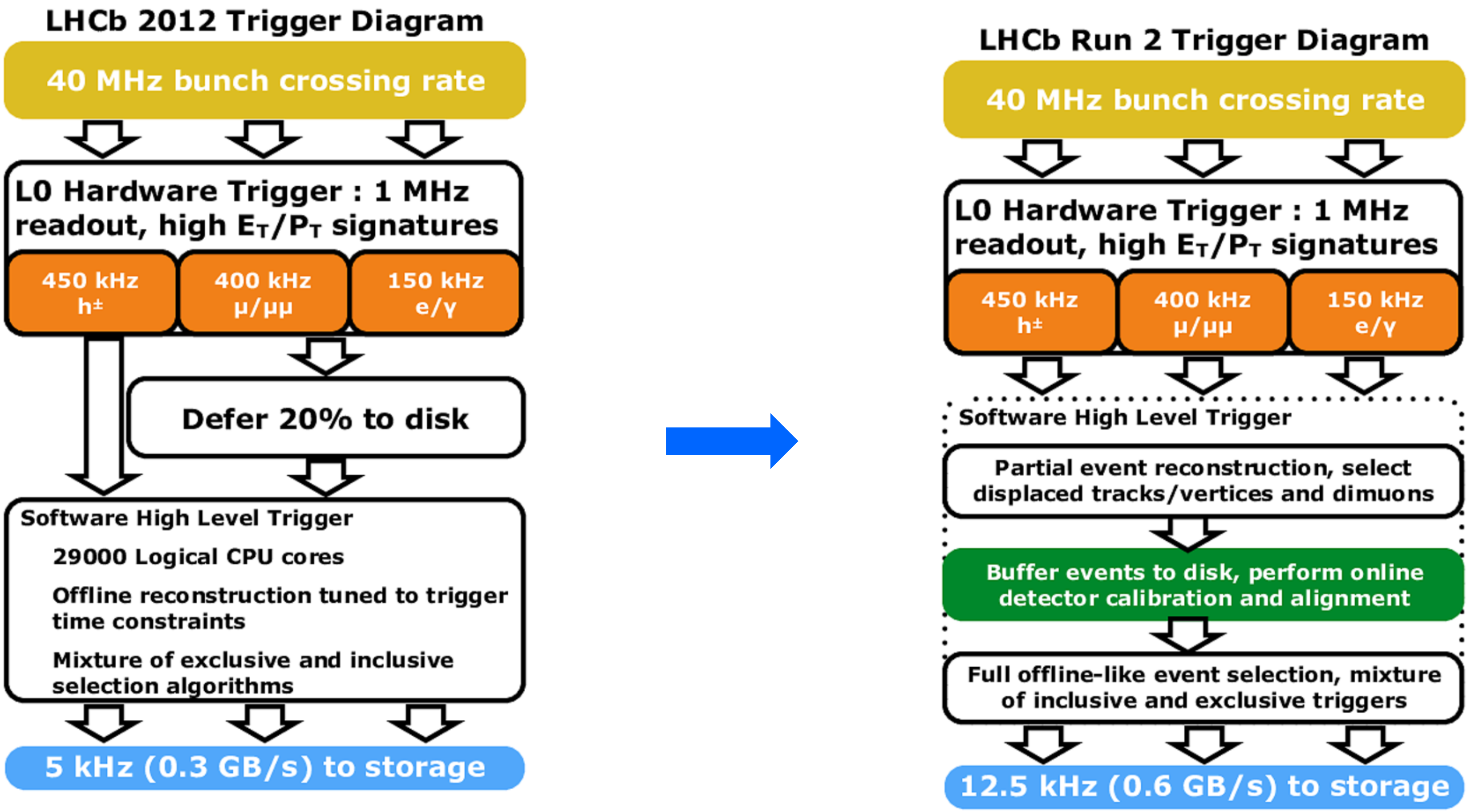
- **In 2012: More CPU needed for better Trigger**
  - Since: unused interfill CPU + free disk space

HLT Farm Usage Evolution

- **Resource Optimization in 2012 -> Deferred HLT**
  - Idea: Buffer data to disk when HLT busy / Process in inter-fill gap
  - ➡ More time for more complex algorithms



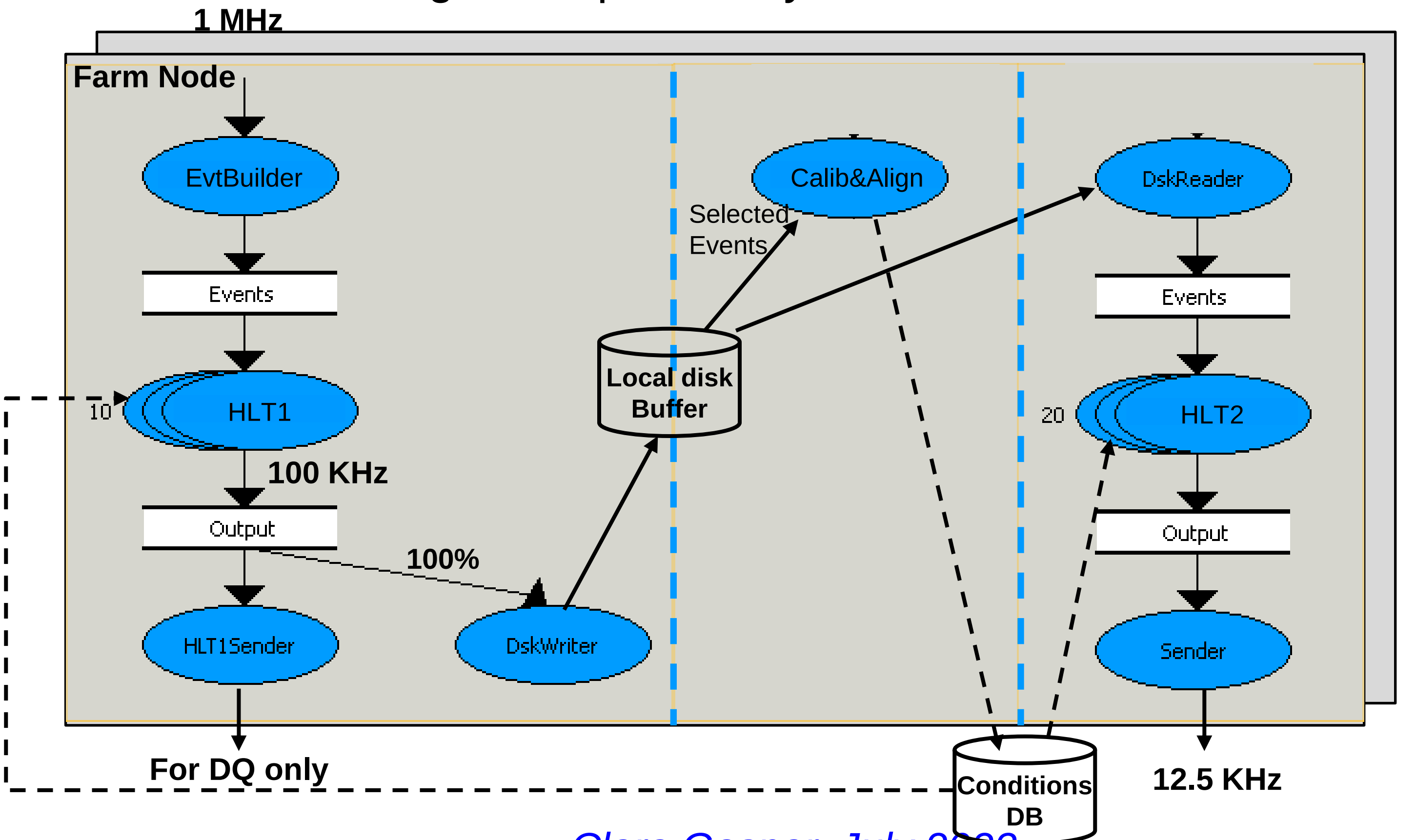
LHCb Dataflow Evolution



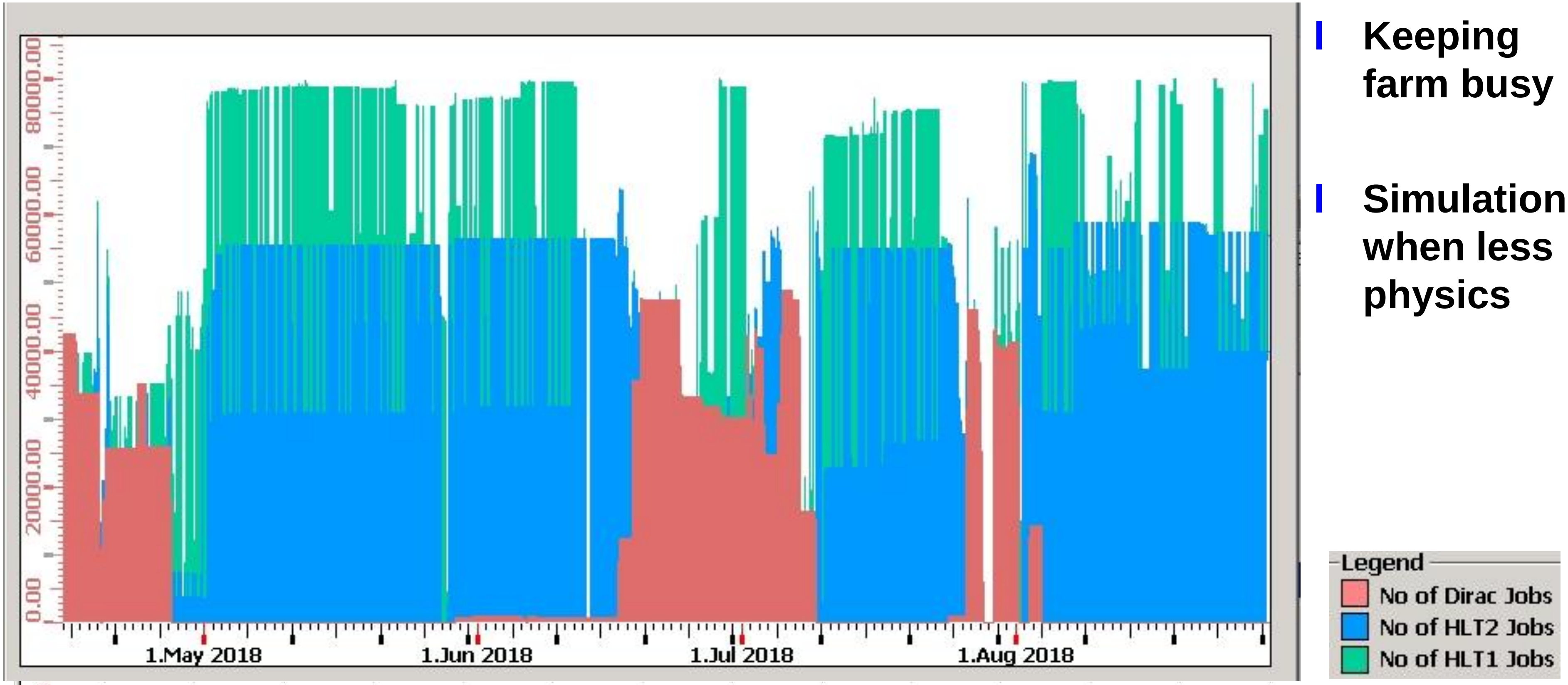
- **In 2015: Better Trigger and Trigger output data**
  - Since: HLT anyway composed of two steps

HLT Farm Usage Evolution

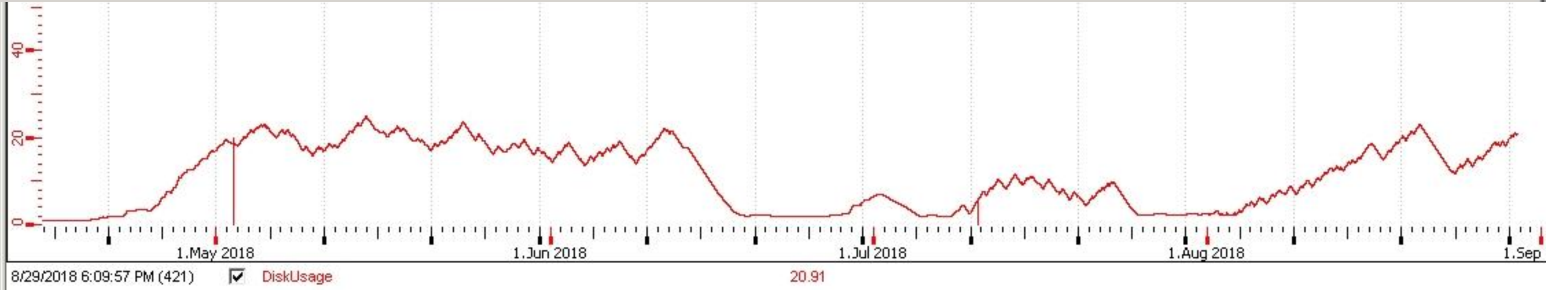
- **Even better Resource Optimization in 2015 -> Split HLT**
  - Idea: Buffer ALL data to disk after HLT1 / Perform Calibration & Alignment / Run HLT2 permanently in background
  - ➡ Calibration and Alignment previously done Offine -> Better data!



Farm Usage in 2018

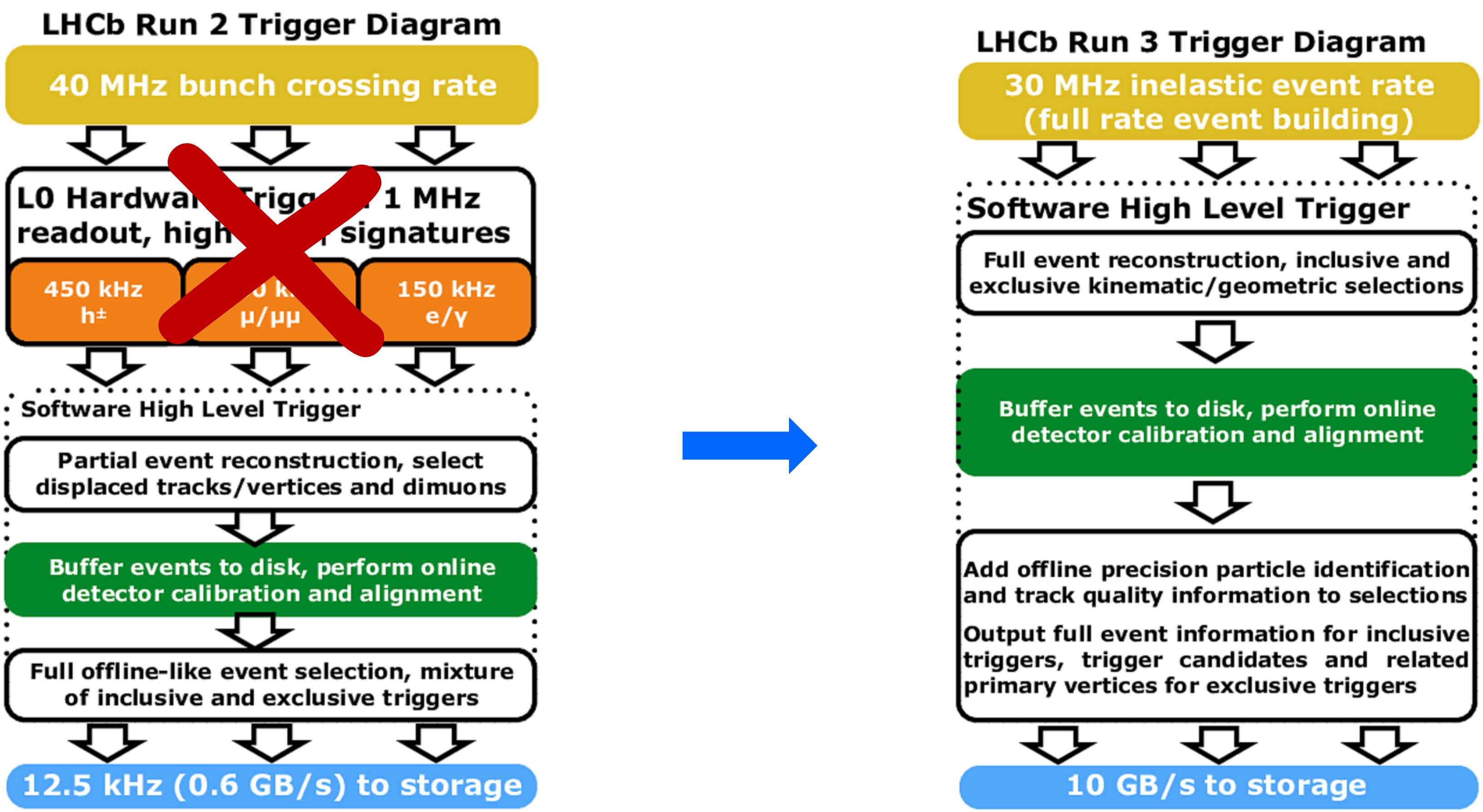


- Keeping farm busy
- Simulation when less physics



- Disk Usage < 25%

LHCb Dataflow Evolution



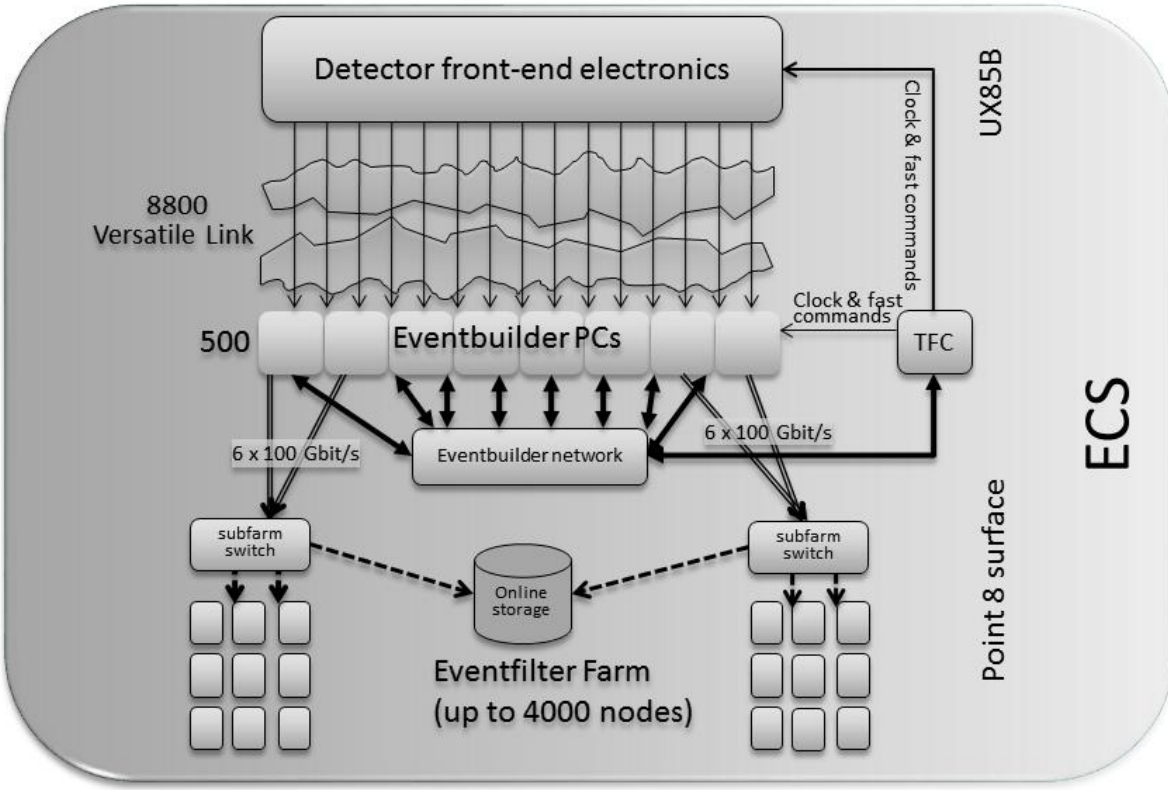
- **In 2022: Even better -> No HW Trigger**
  - Since: Better readout and network technology available

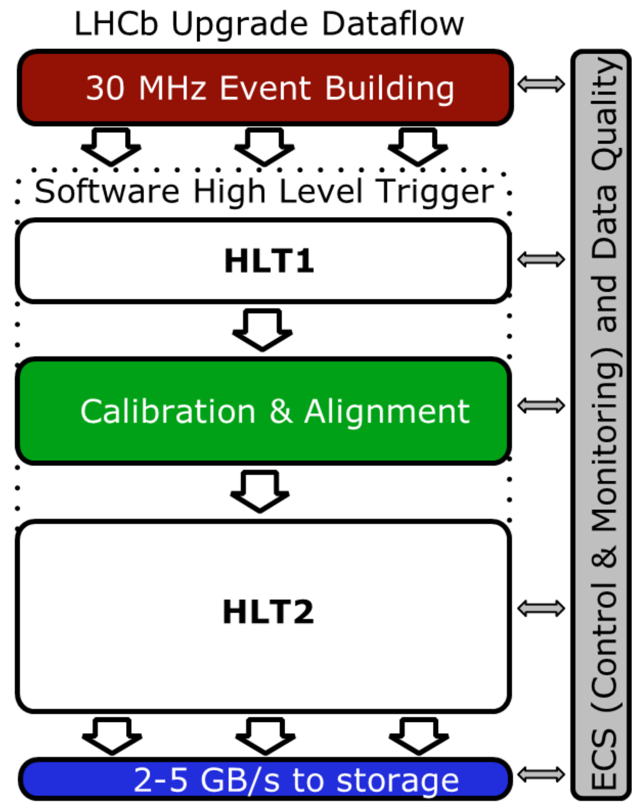
New DAQ Design Principles

- **Selected Concepts:**
  - **Simplicity (whenever possible)**
  - **Integration & Homogeneity**
  - **Promote HW Standardization**
    - Same hardware components for all sub-systems
  - **Promote SW Uniformity**
    - Guidelines, Framework Components, Templates
  - **Separate Data/Control paths**
    - From the Front Ends to the High Level Trigger

Online System Upgrade

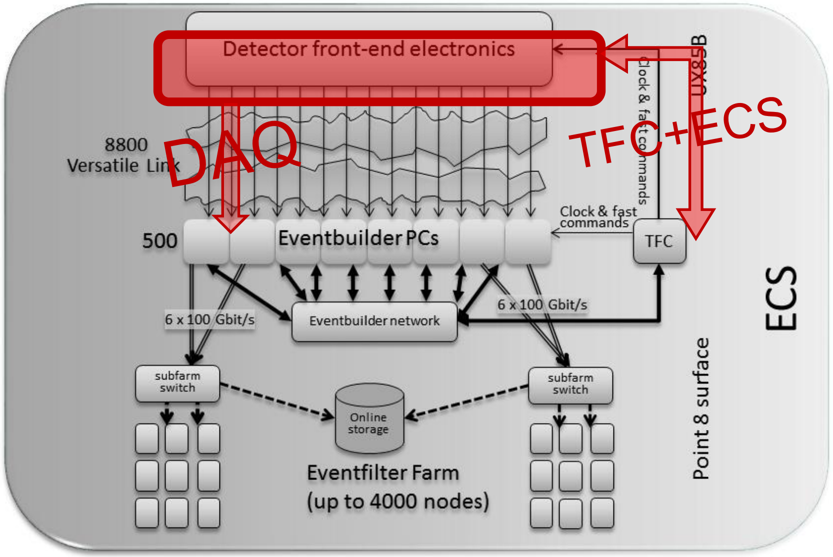
- **New Data Acquisition Hardware and Dataflow Software**



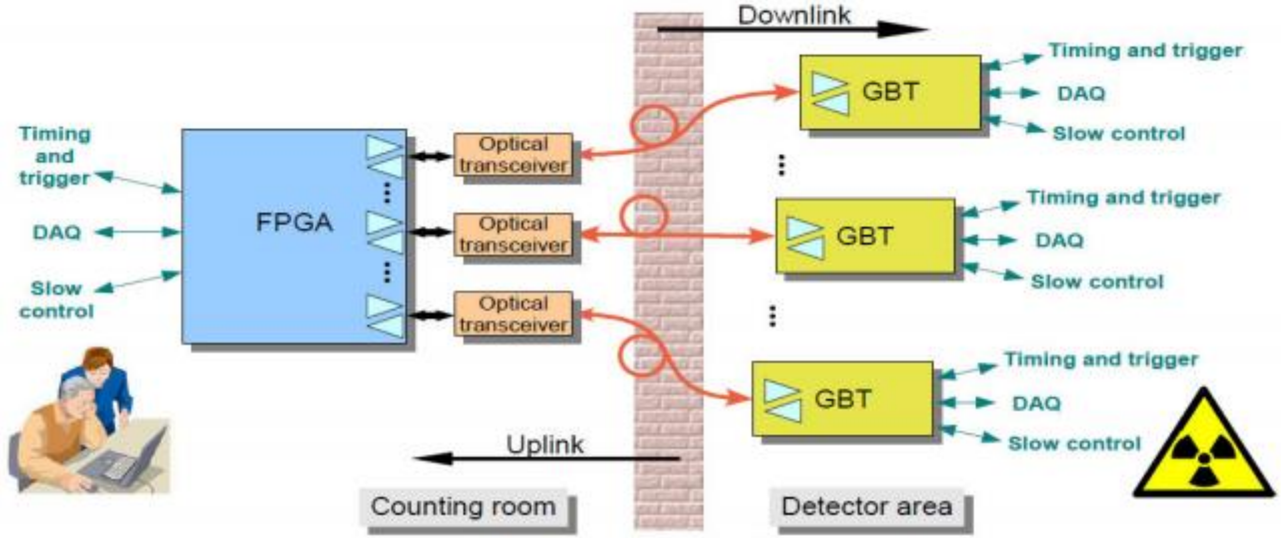


- **New Infrastructure & Sub-Detector Equipment**

FE Electronics



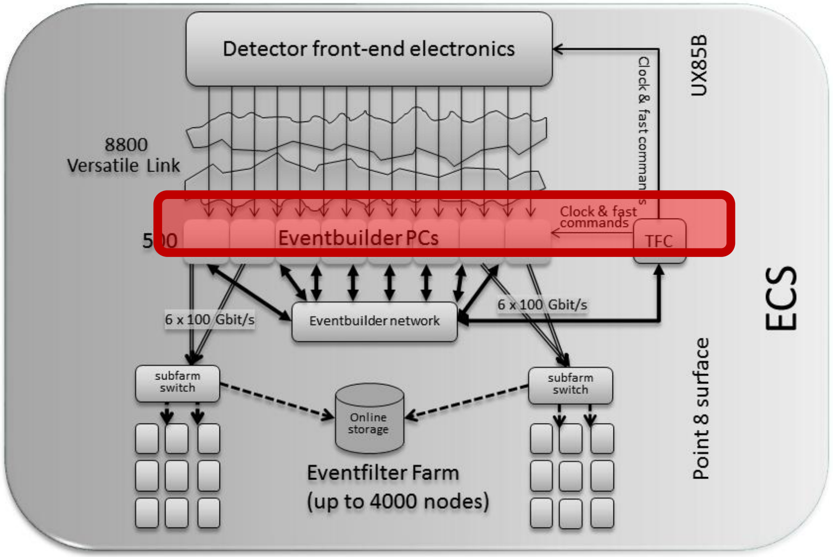
- **Front-end electronics**
  - Use CERN GBT chip (Gigabit Transceiver)
  - A radiation hard ASIC (Application Specific Integrated Circuit)

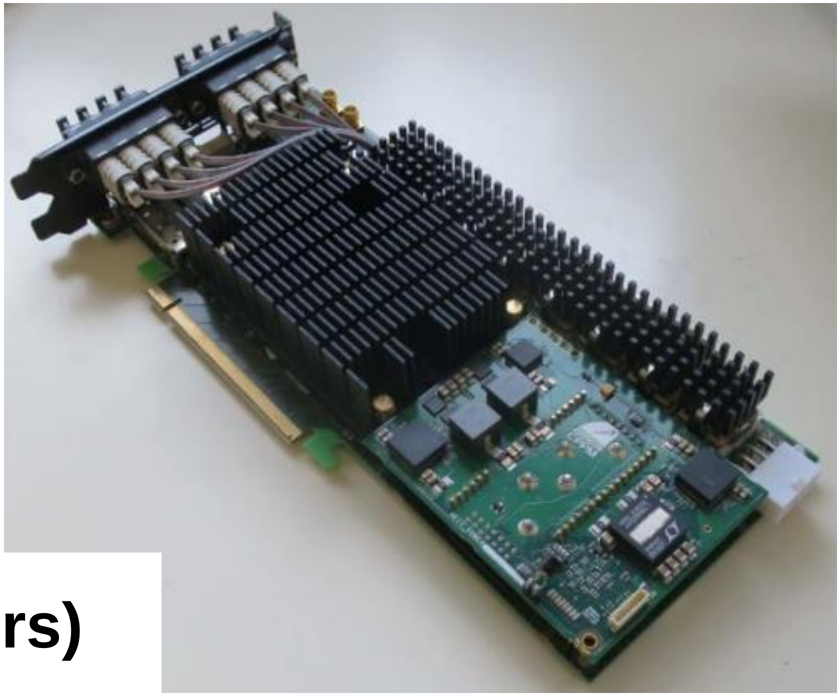


  - But unlike the drawing: separately for DAQ and for Timing & Control

Readout Units

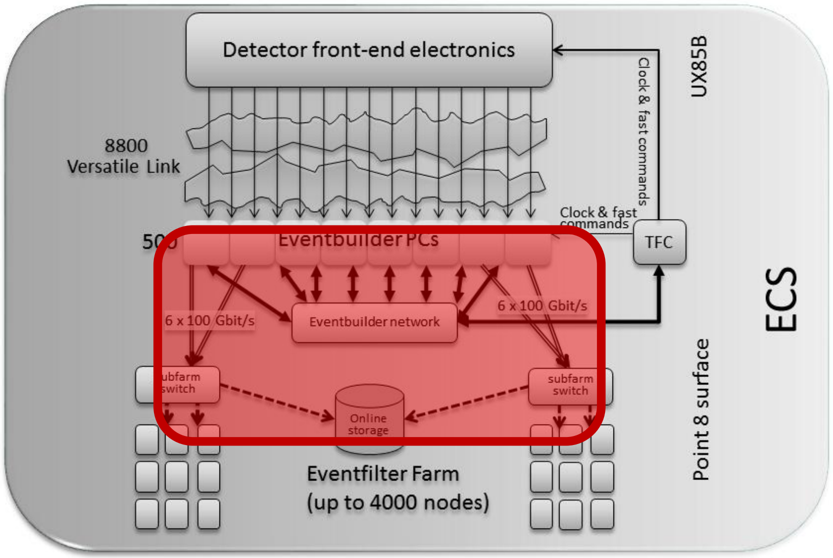
- **PCIe40 cards in EB PCs**
  - PCIexpress card
  - Large FPGA
  - 48 GBT links, 10 Gb/s each
  - Versatile via the firmware
- **Located on surface**
  - Distance from FE: ~350m
- **In total:**
  - ~ 10000 optical links
  - ~ 500 readout boards
  - ~ 100 TFC/ECS cards (separated from data path)
- **Used also by ALICE (and others)**



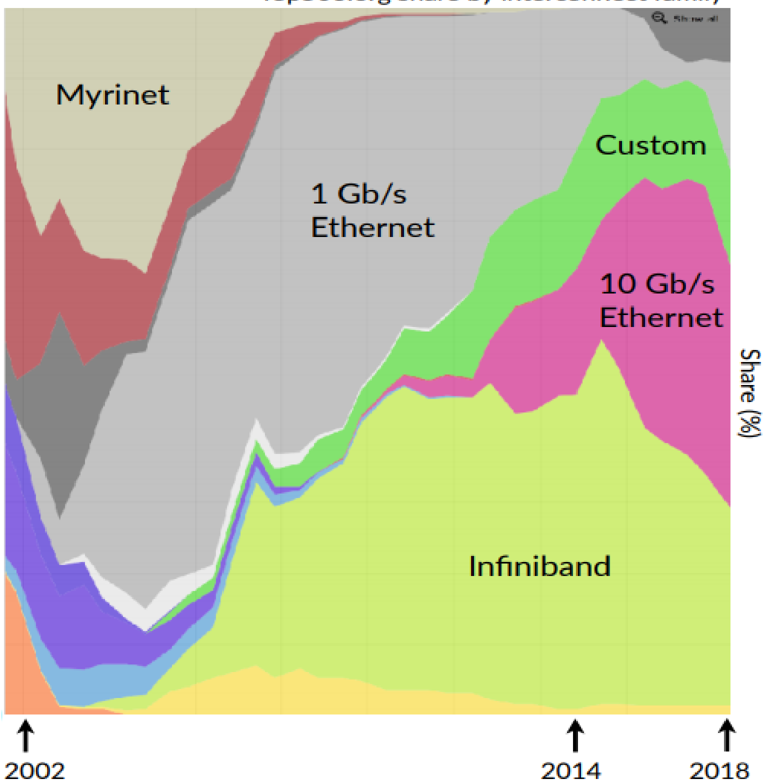


Networking

- Again, use what is popular -> best price
- Basically the choice was Infiniband and/or Ethernet

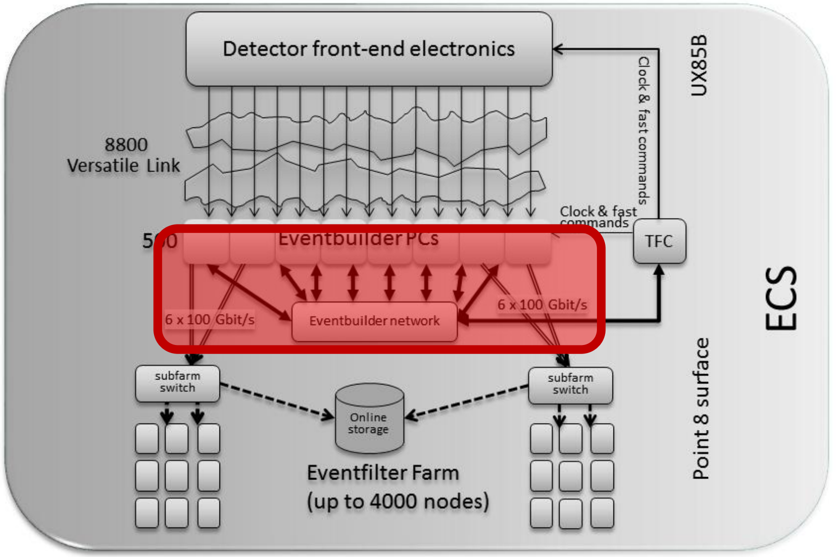


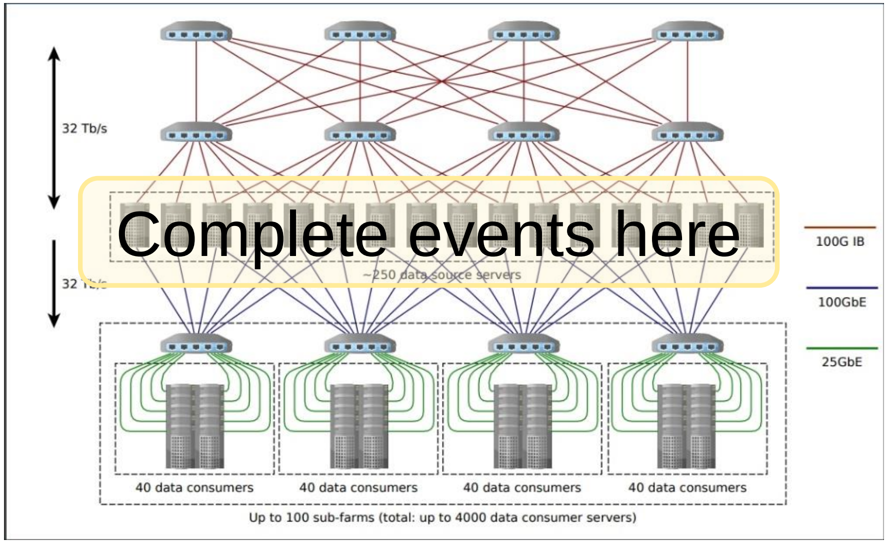
Top500.org share by interconnect family



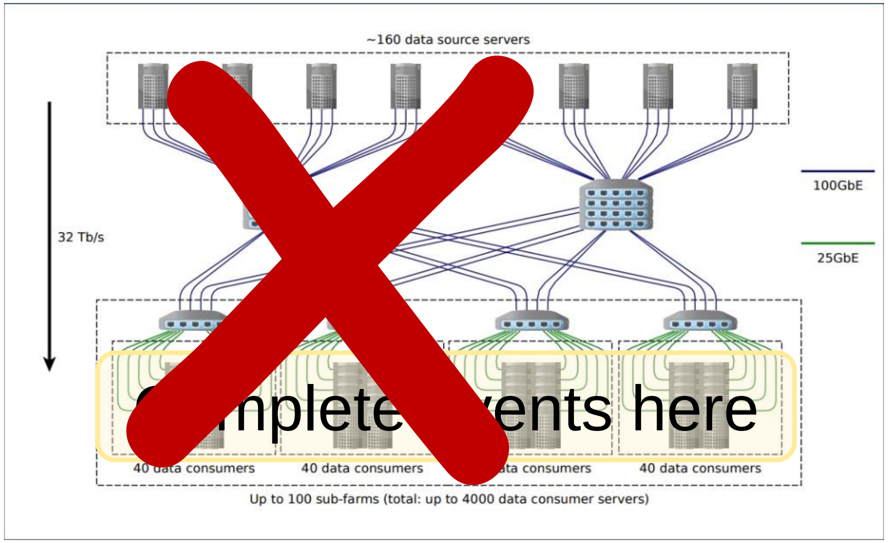
Event Builder

- **Two options tested**
  - Only one worked well
    within our time and budget
    (traffic pattern: All to One/flow control /switch buffering very expensive)



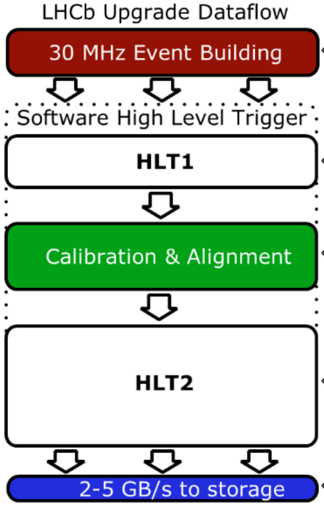


Technology: **Infiniband**
+ Ethernet for Distribution
Dedicated EB (TDR baseline)

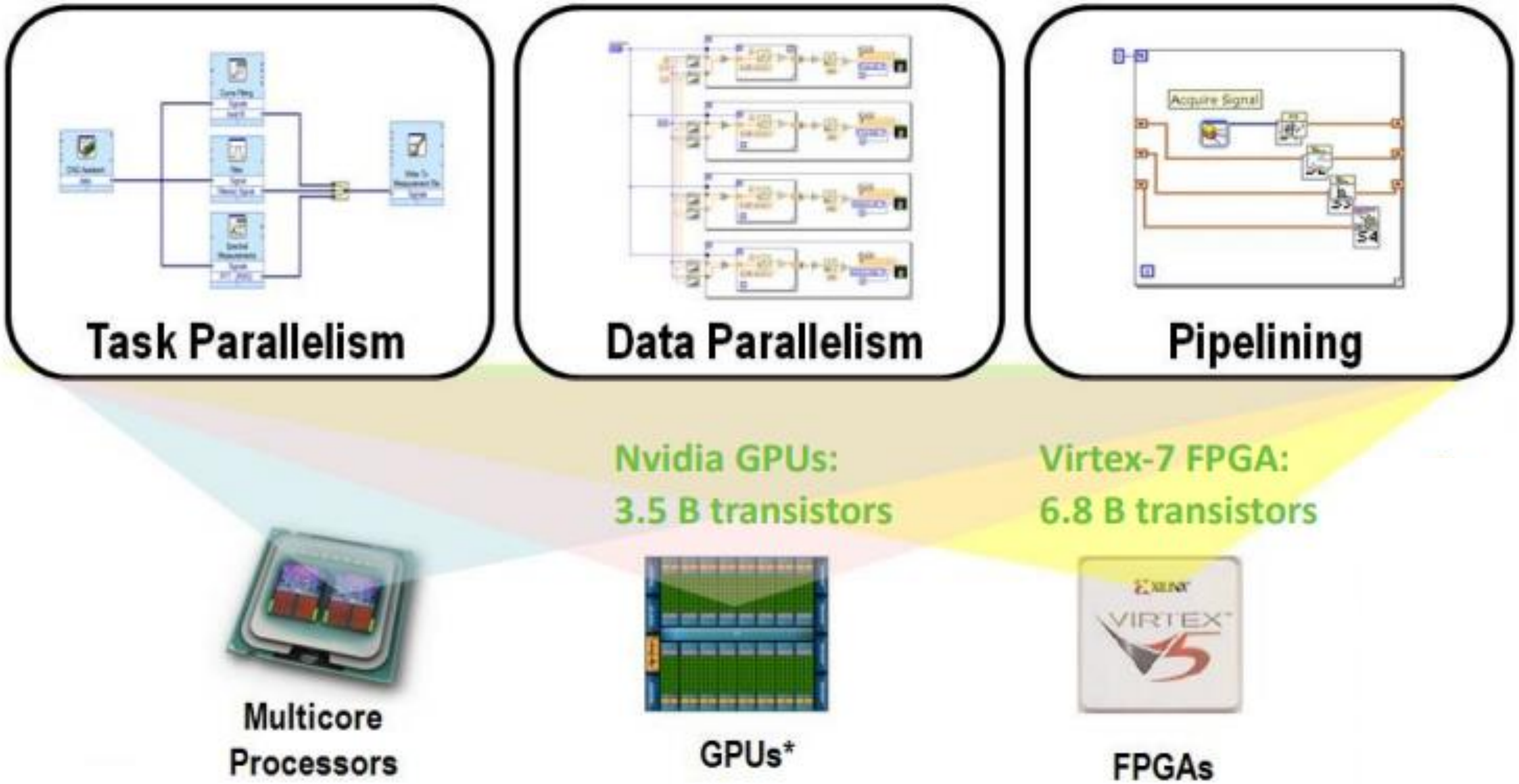


Technology: **Ethernet**
(Would have been simpler and known)
Distributed EB

High Level Trigger

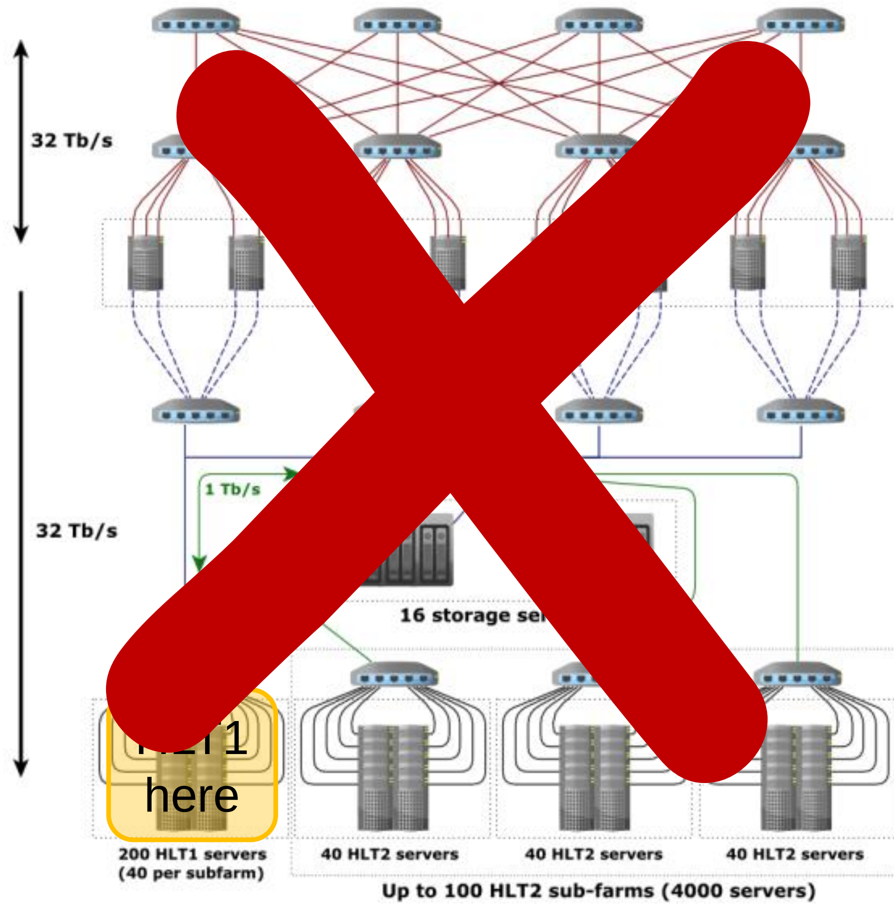


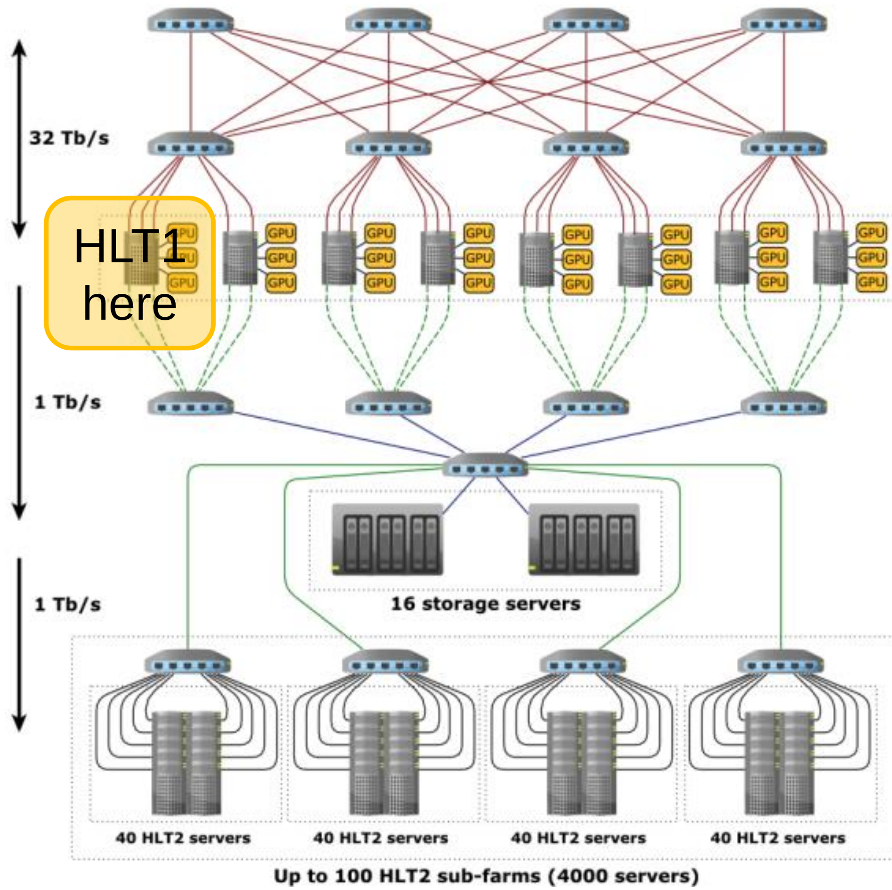
Processing Options (HLT1)



- Several R&D Projects in all these areas

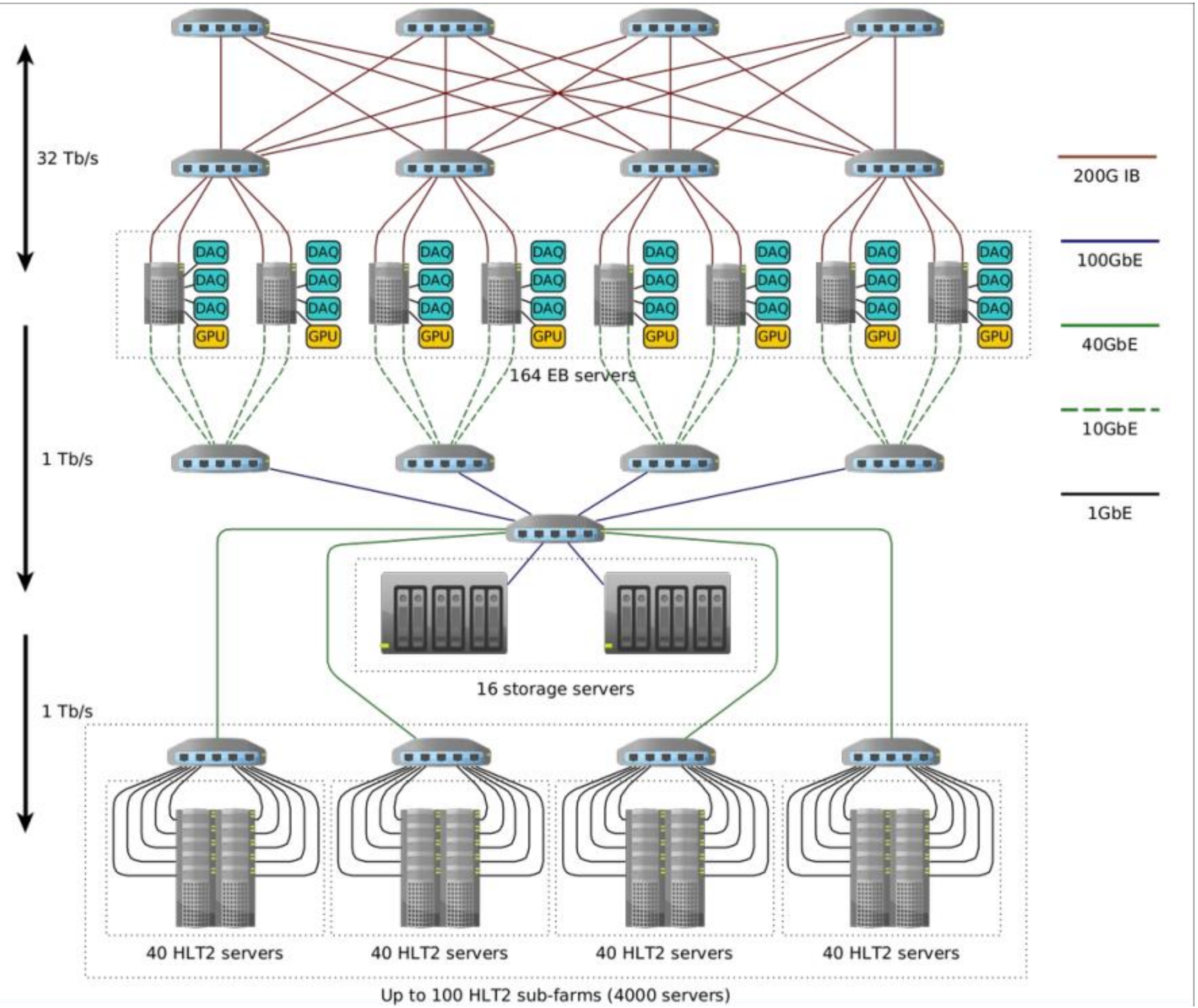
HLT1





- **Two options: CPU vs GPU**
  - Both worked: Cost (and politics/sociology) made the difference

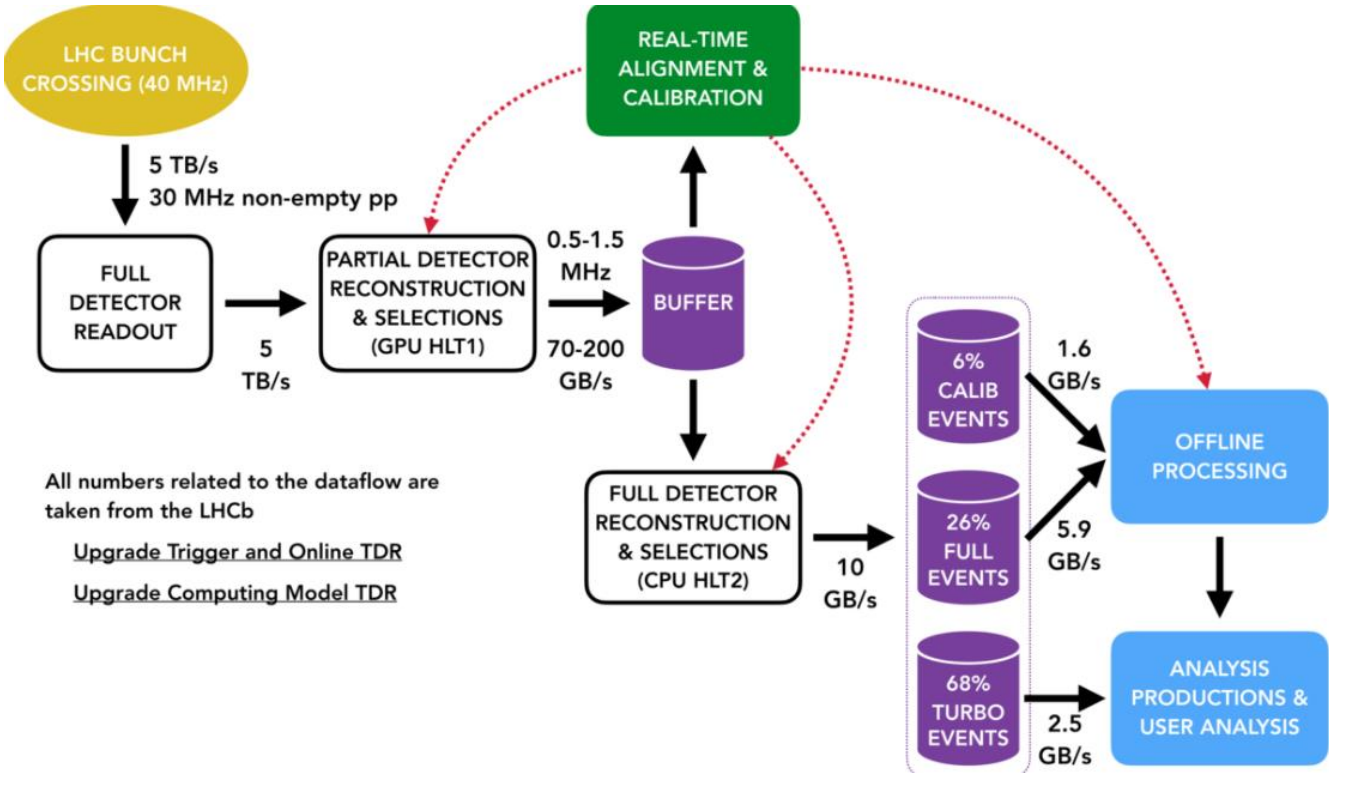
LHCb Current DAQ Core



- **EB Servers**
  - ~160
  - Hosting PCIe cards
- DAQ **Readout Units:**
  - ~500
  - Powerful FPGA
  - 48 FE links (1Gb)
- GPU **GPUs:**
  - ~160 -> ~320
  - Fast HLT1 Pass
- **CPUs**
  - ~4000 (donated)
  - Thorough HLT2

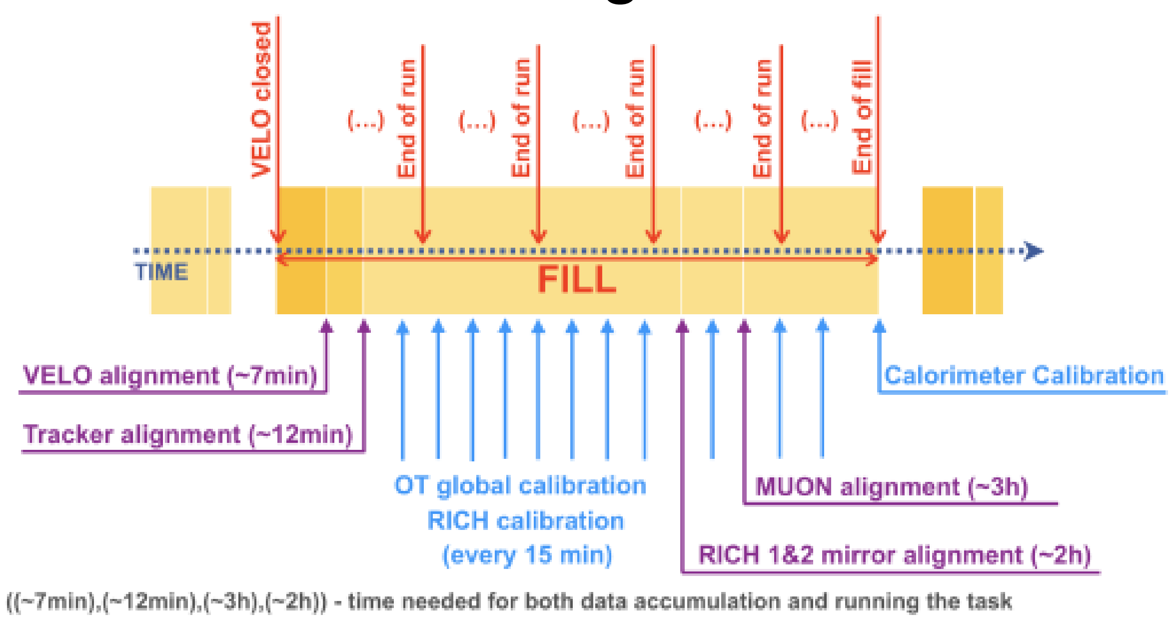
Data Processing

- **Improvements all the way to analysis strategy: Turbo stream**
  - Since we can automatically perform final alignment and calibration online
  - So we can achieve offline quality alignment and calibration in the trigger
  - Store only part of the event → allows for higher output bandwidth

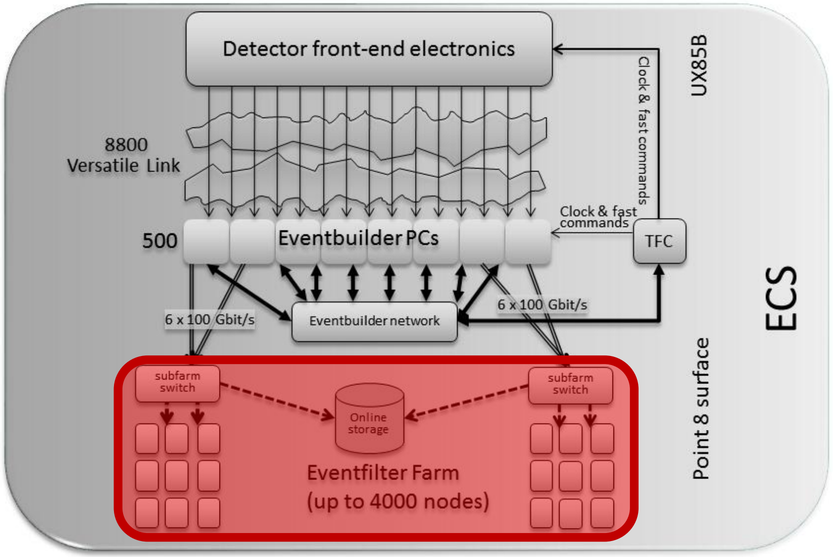


Alignment and Calibration

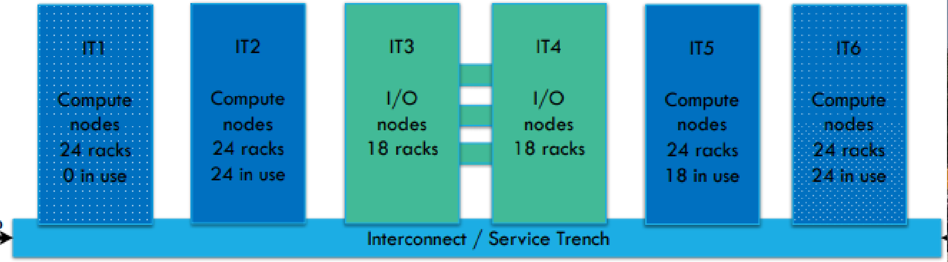
- **Why do we need alignment & Calibration?**
  - Physical movements, magnetic field, temperature, pressure effects
  - Better mass resolution
  - Better particle identification (PID)
  - Store less background → More bandwidth for physics!
- **What and when do we align?**



HLT2



- **A large CPU Farm**
  - Around 4000 PCs
    (many inherited for free)
  - Also runs Calibration and Alignment tasks
- **A brand new Computer Center**



Configuration, Control and Monitoring

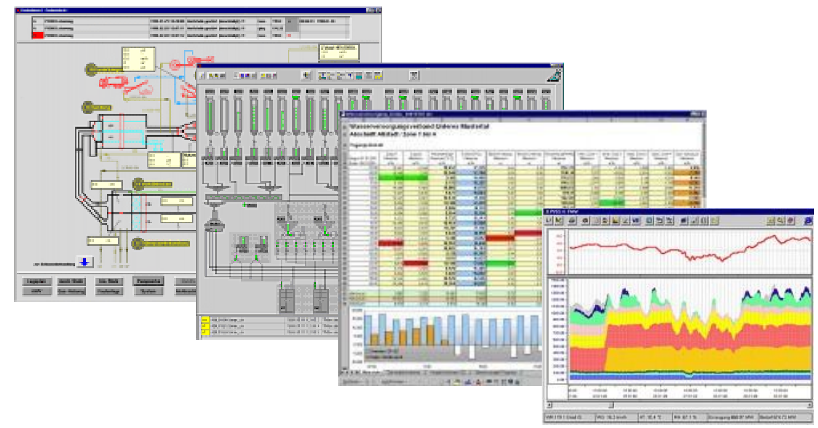
Clara Gaspar, July 2023

Control System Tasks

- **Configuration**
  - Selecting which components take part in a certain "Activity"
  - Loading several millions of parameters (according to the "Activity")
- **Control core**
  - Sequencing and Synchronization of operations across the various components
- **Monitoring, Error Reporting & Recovery**
  - Detect and recover problems as fast as possible
  - Automate Standard Operations
- **User Interfacing**
  - Allow the operator to visualize and interact with the system

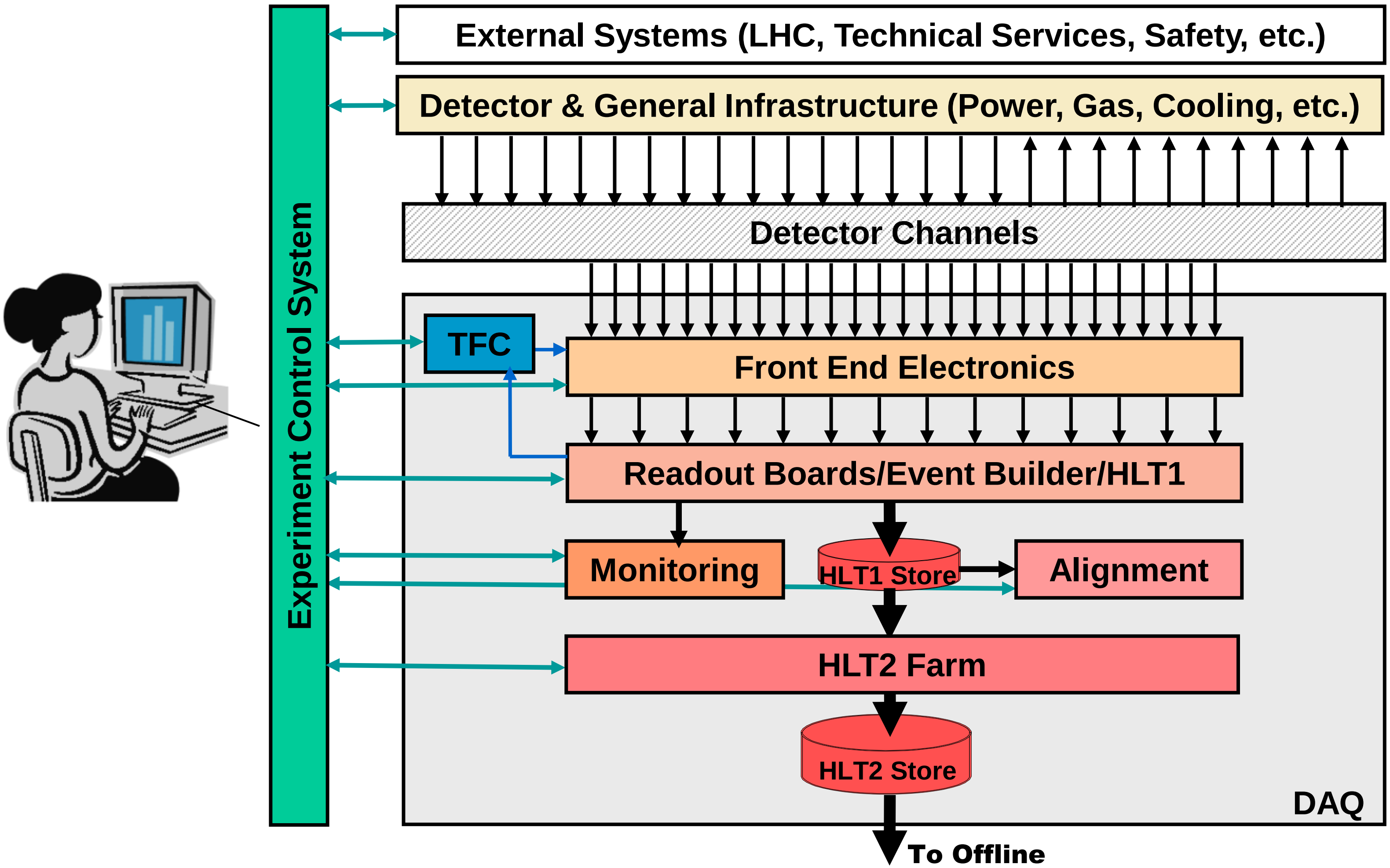
Control Framework

- **Can be based on Commercial SCADA Systems**
  (Supervisory Control and Data Acquisition)
  - Commonly used for:
    - Industrial Automation
    - Control of Power Plants, etc.
  - Providing:
    - Run-time Database and Tools
    - Archiving of Monitoring Data including display and trending Tools.
    - Alarm definition and reporting tools
    - User Interface design tools
  - Used in LHC experiments DCS and LHCb for everything

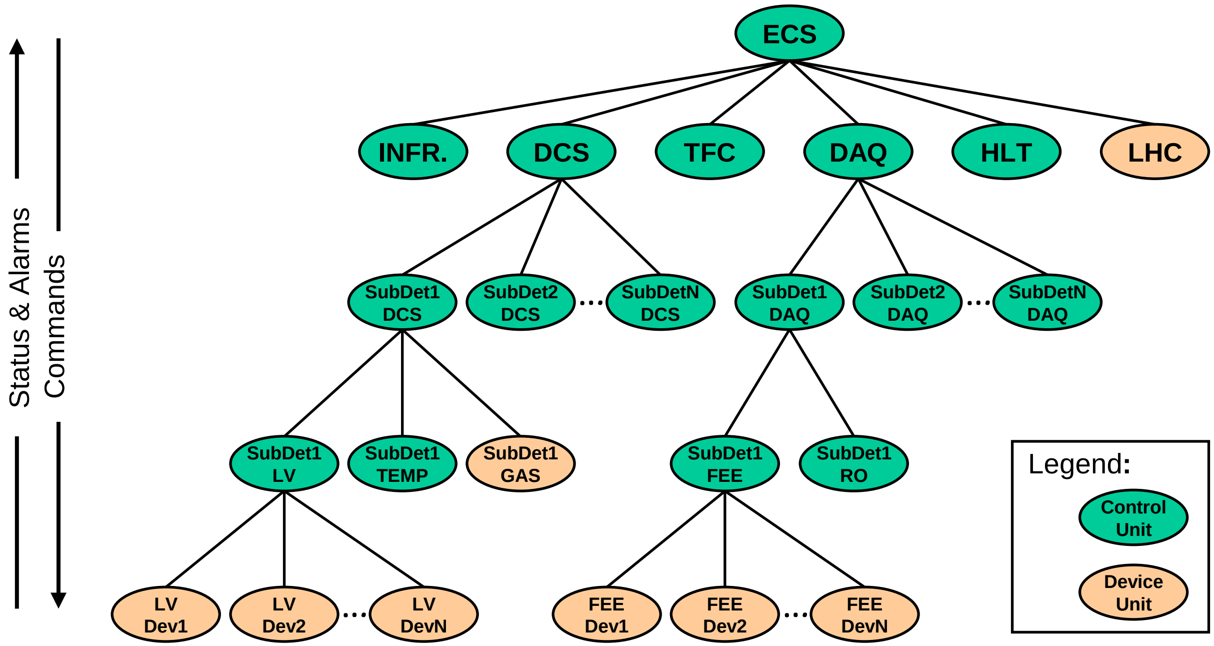


- **Or can be home made**

The Control System



Experiment Control System

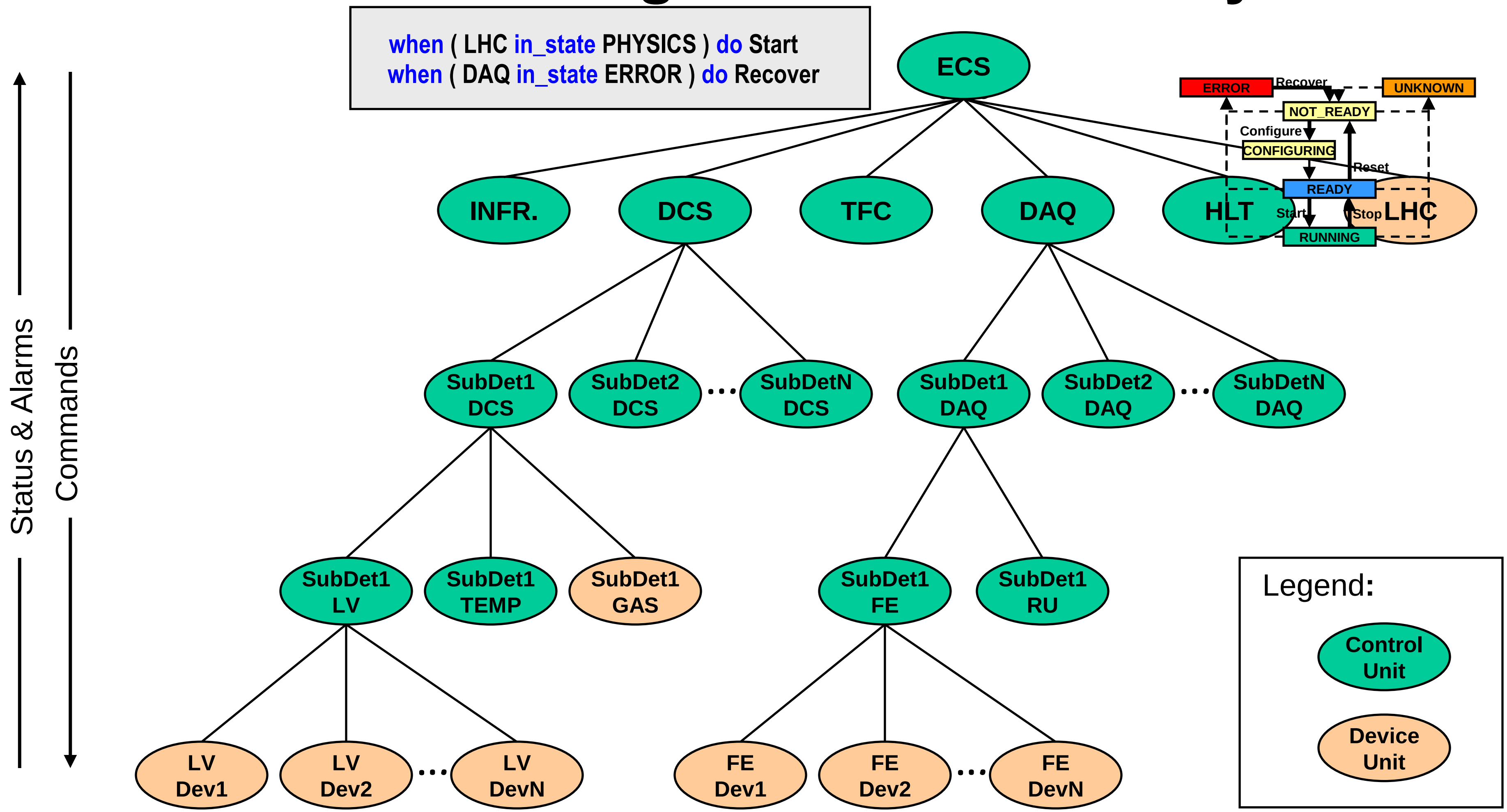


- **Implementation: (JCOP project)**
  - Framework
- **Deployment:**
  - Runs distributed over >100 PCs (Virtual Machines)

- **Control Units are logical entities:**
  - Behave as a Finite State Machine / Rule Based system:
    - Capable of Partitioning: Exclude/Include children
    - Can take local decisions: Sequence & Automate Operations or Recover Errors
- **Device Units**
  - Provide the interface to the device (hardware or software)

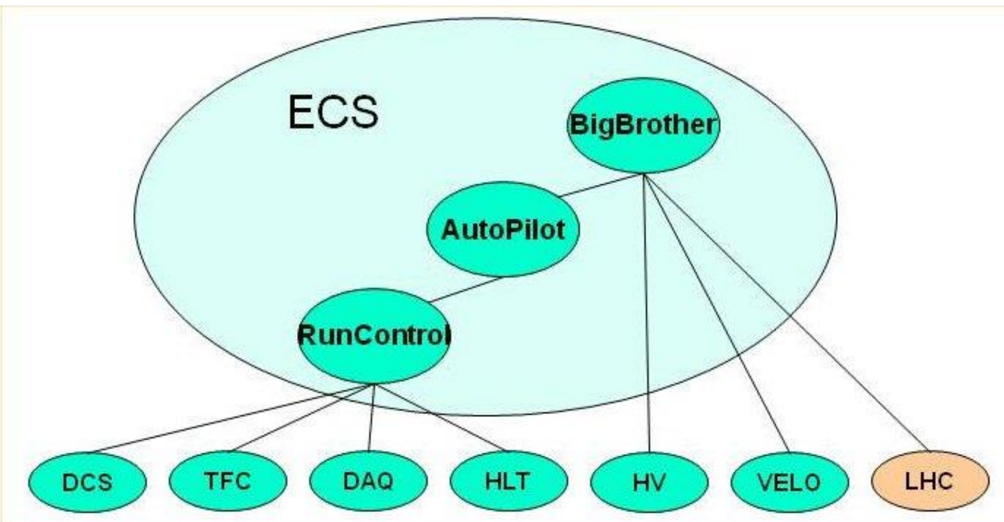
Control System Behaviour

- Distributed "Intelligent" Hierarchical System



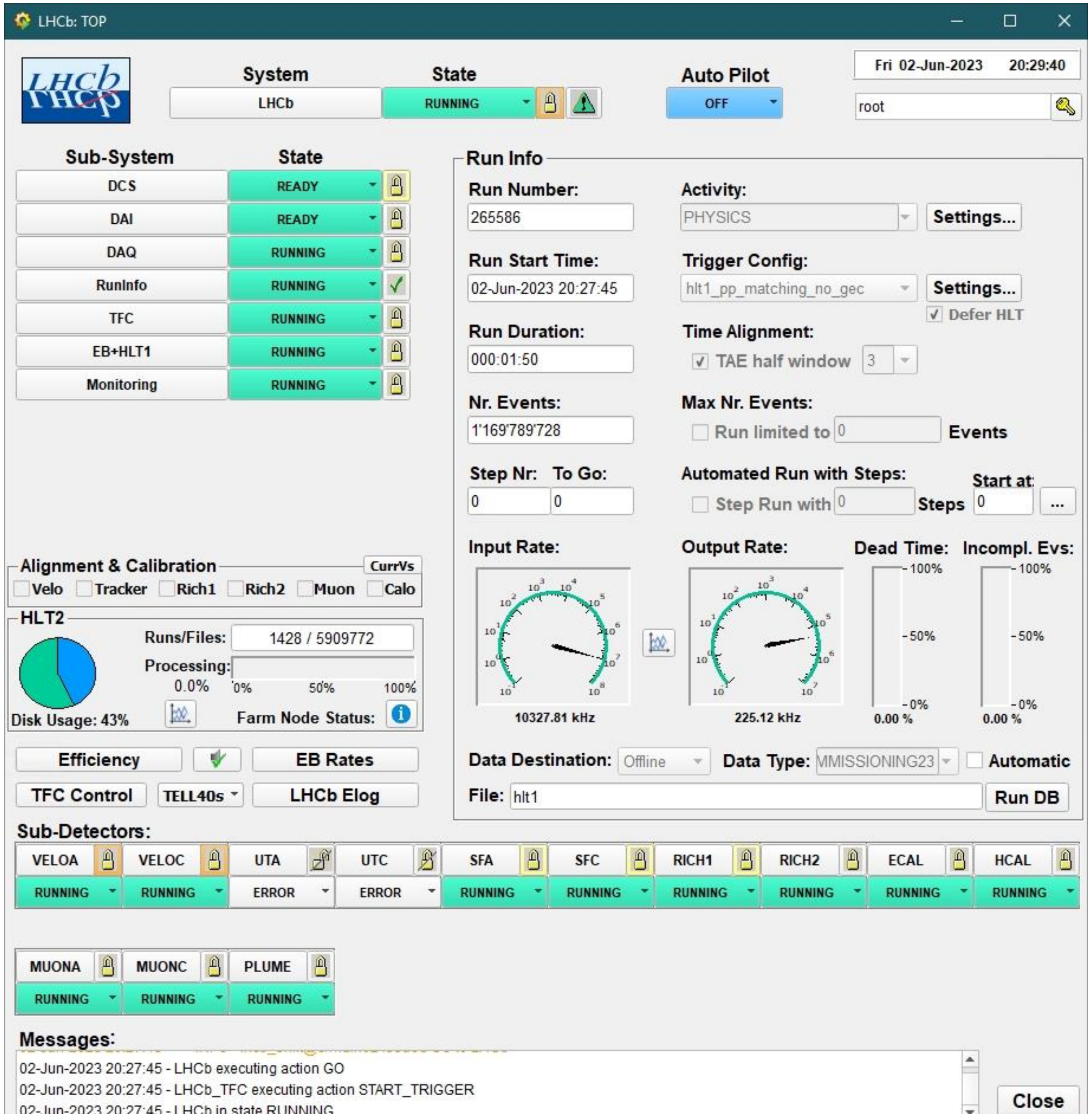
Operations

- **Main Tools:**

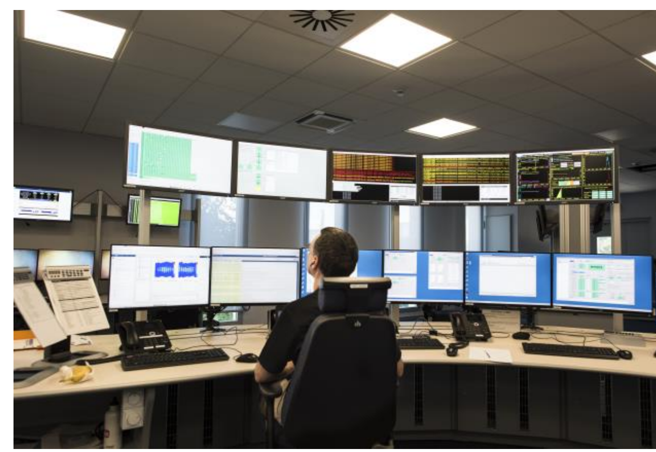


- **RunControl**
  - **Handles the DAQ & Dataflow**
  - **Allows to:**
    - **Configure the system**
    - **Start & Stop runs**
- **AutoPilot**
  - **Knows how to start and keep a run going from any state.**
- **BigBrother**
  - **Based on the LHC state:**
    - **Controls SD Voltages**
    - **VELO Closure**
    - **RunControl**

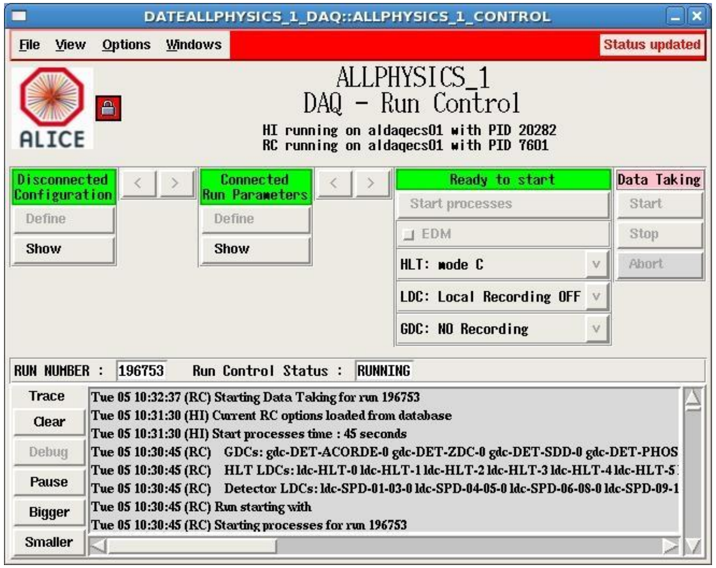
Run Control



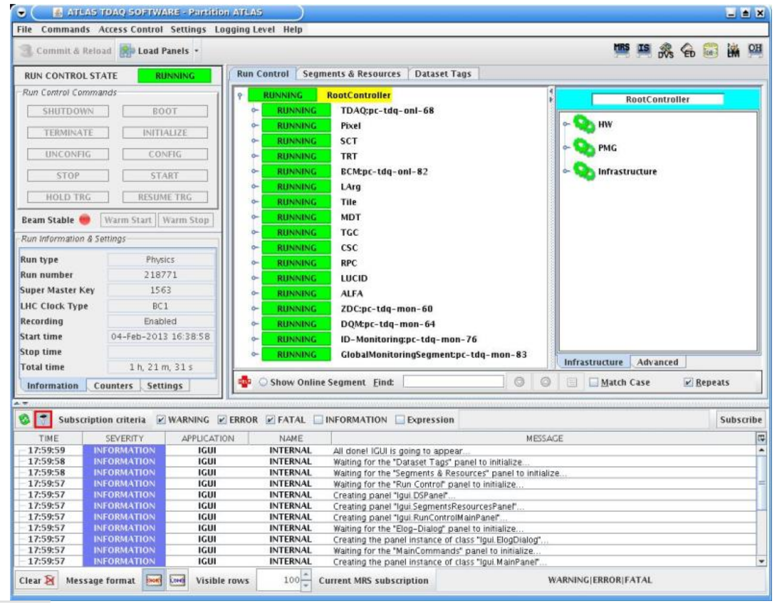
- Run Control & The Autopilot:
  - Configures, Starts and Keeps the RUN going.
  - Configuration Driven by an “Activity”



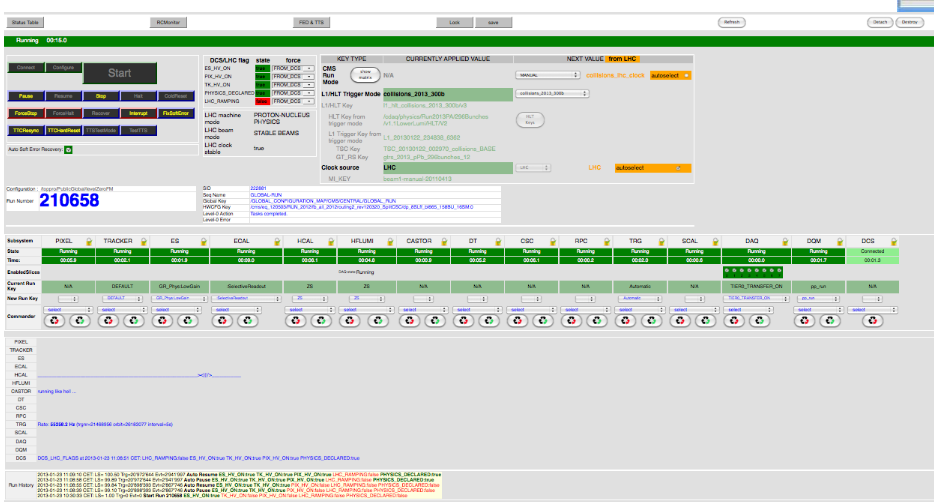
Run Control



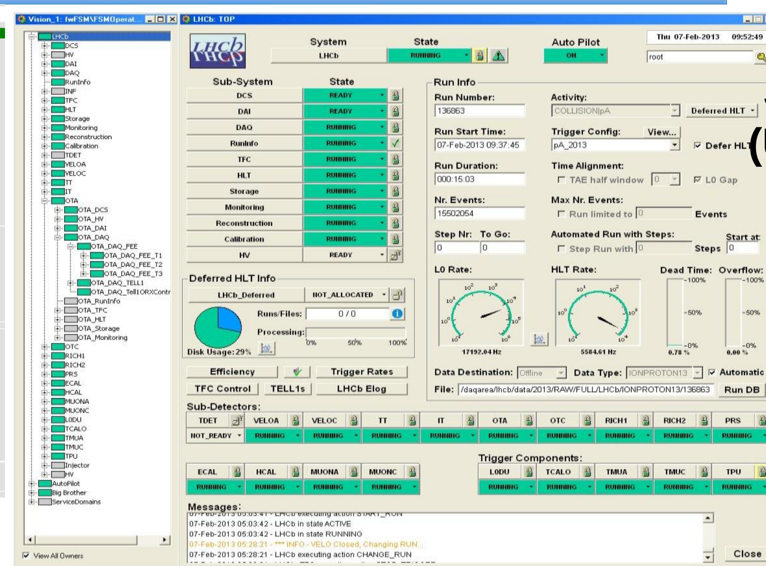
ALICE: Tcl/Tk



ATLAS: Java (modular)

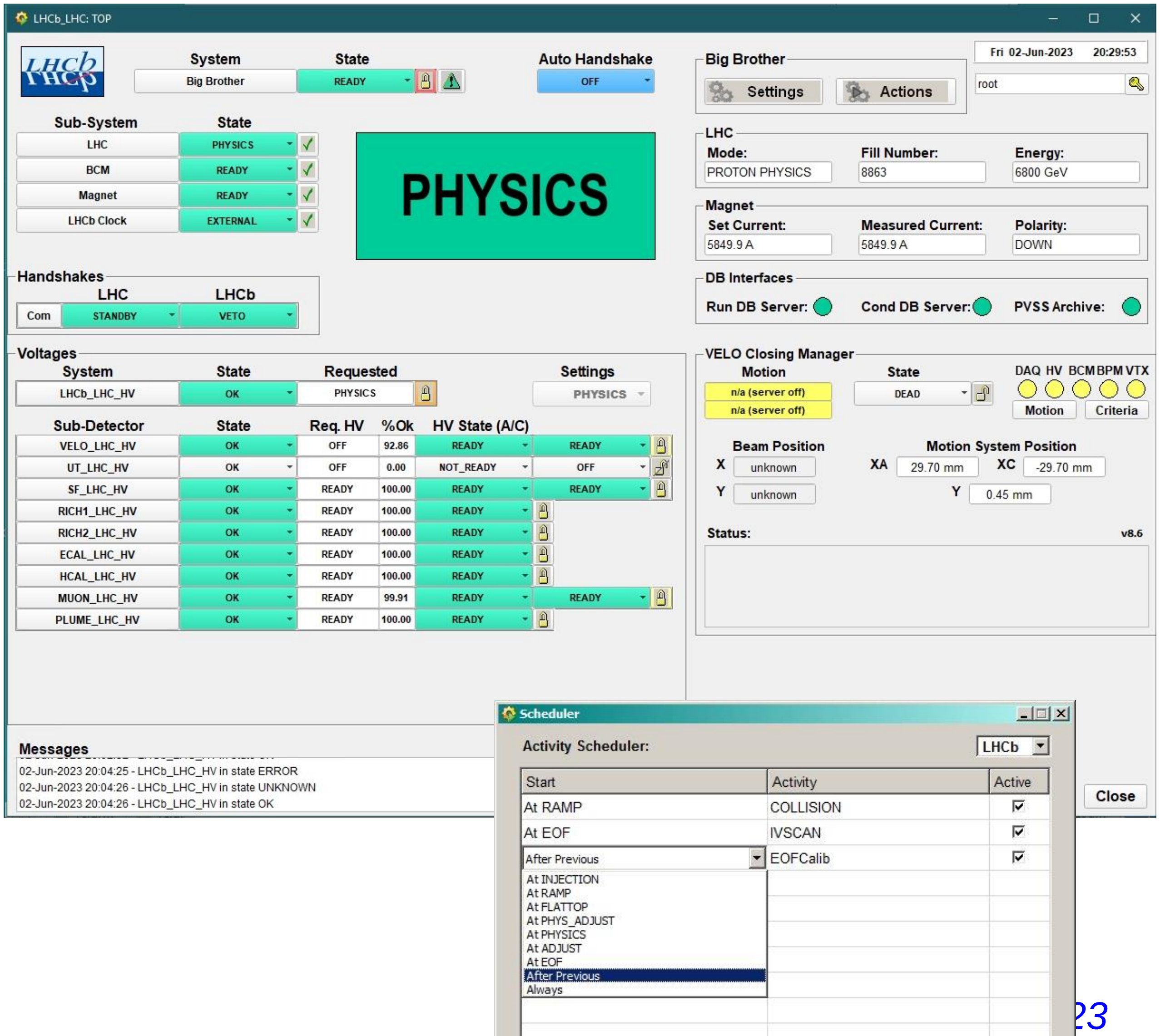


CMS: Web Tools (JavaScript+HTML)



LHCb: JCOP FW (UI builder)

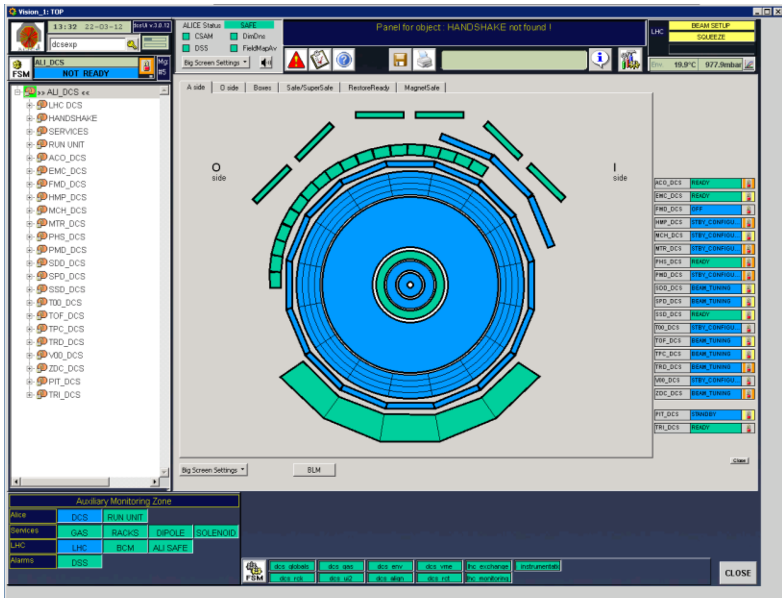
Big Brother



- Based on LHC state, controls:
  - Voltages
  - VELO Closure
  - Run Control
- Can sequence activities, ex.:
  - End-of-fill Calibration
- Confirmation requests and Information
  - Voice Messages

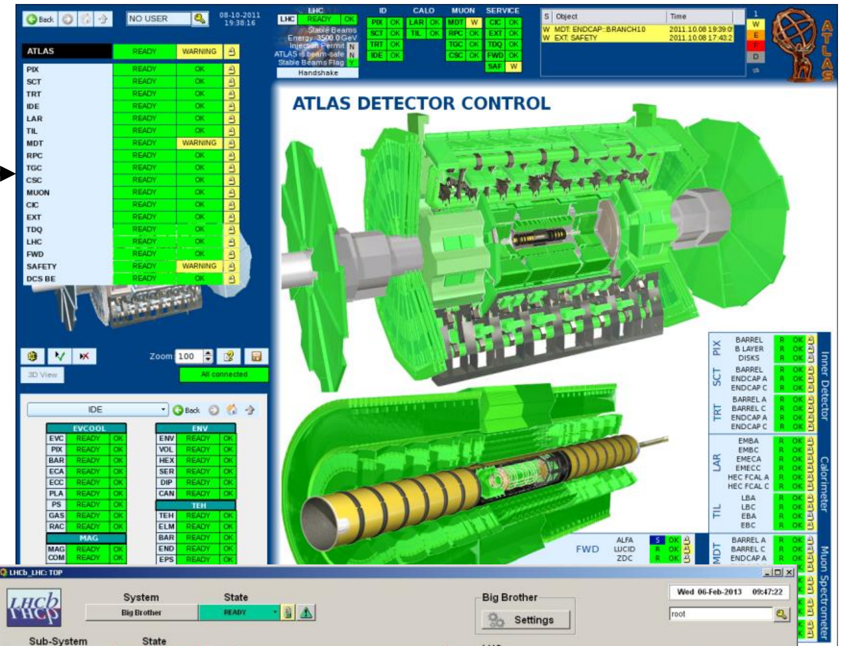
Detector Control System

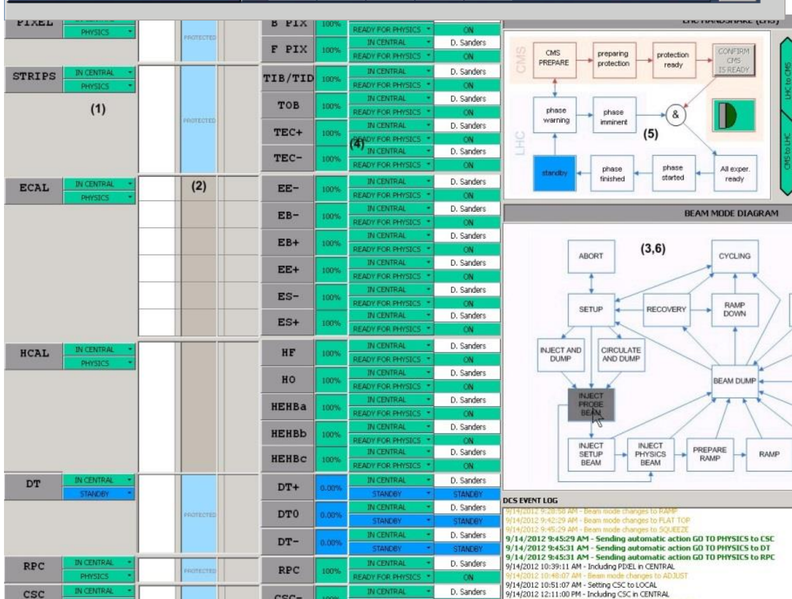
JCOP FW



← **ALICE**

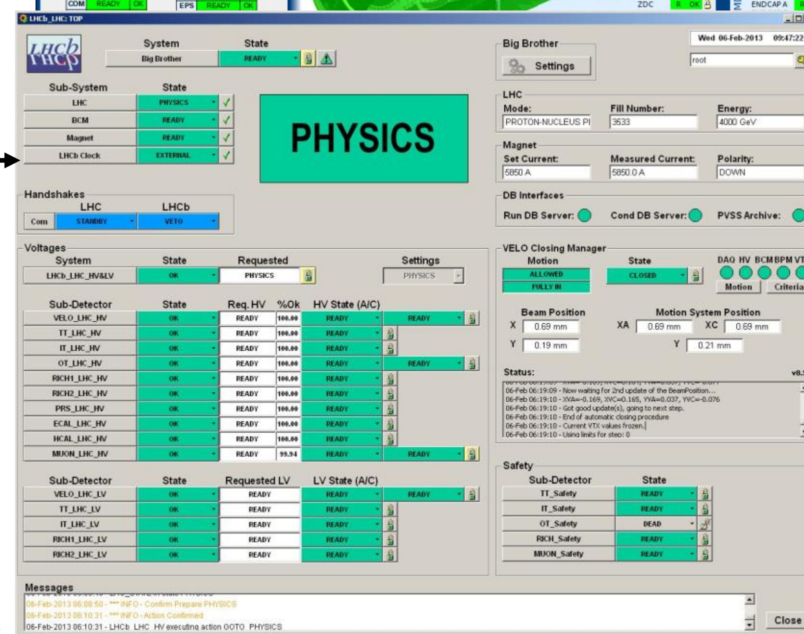
ATLAS →



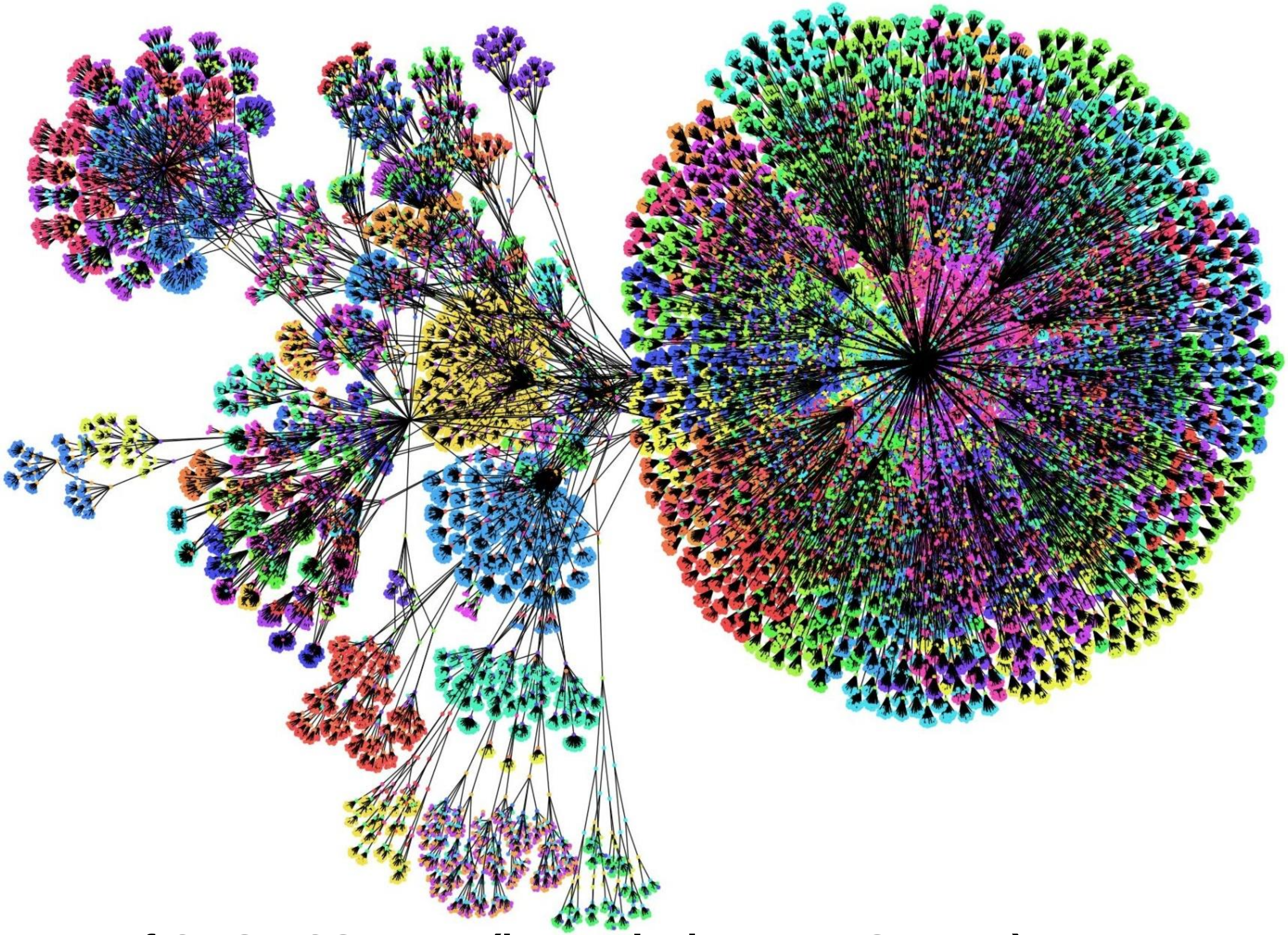


← **CMS**

LHCb →



LHCb Control System



- **Courtesy of CMS DCS Team (by analyzing our FSM tree)**

Concluding Remarks

- Trigger and Data Acquisition systems have become increasingly complex.
- Luckily the requirements of telecommunications and computing in general have strongly contributed to the development of standard technologies:
  - Hardware: Fast ADCs, Field-Programmable Gate Arrays, Analog memories, multi-core PCs, Networks, Helical scan recording, Data compression, Image processing (GPUs), ...
  - Software: Distributed computing, Software development environments, Supervisory systems, Artificial Intelligence tools...
- We can now build a large fraction of our systems using commercial components (customization will still be needed in the front-end).
- It is essential that we keep up-to-date with the progress being made by industry.

LHC/LHCb Fill Sequence

NO_BEAM — **Injection Permit = FALSE, Internal Clock, Any state of LHCb**

↓ *Handshake for Injection(TI8_Setup)* — **Act: Lower Voltages, VELO Out**

INJECTION — **Injection Permit = TRUE**

↓

RAMP — **Injection Permit = FALSE** — **Act: Start Raising Voltages**

↓ **At FLAT_TOP:** — **Act: External Clock, Reset DAQ**

PHYS_ADJUST

↓

PHYSICS — **Act: Raise Voltages, VELO In**

↕ *Handshake for Adjust* — **(Act: Lower Voltages, VELO Out)**

(ADJUST)

↓ (from PHYSICS)

DUMP/EOF — **Act: VELO Out, Internal Clock**
Act: EOF Calibrations

↓ (red) ***Handshake: Confirm Handshake -> Prepare detector -> Confirm Ready***

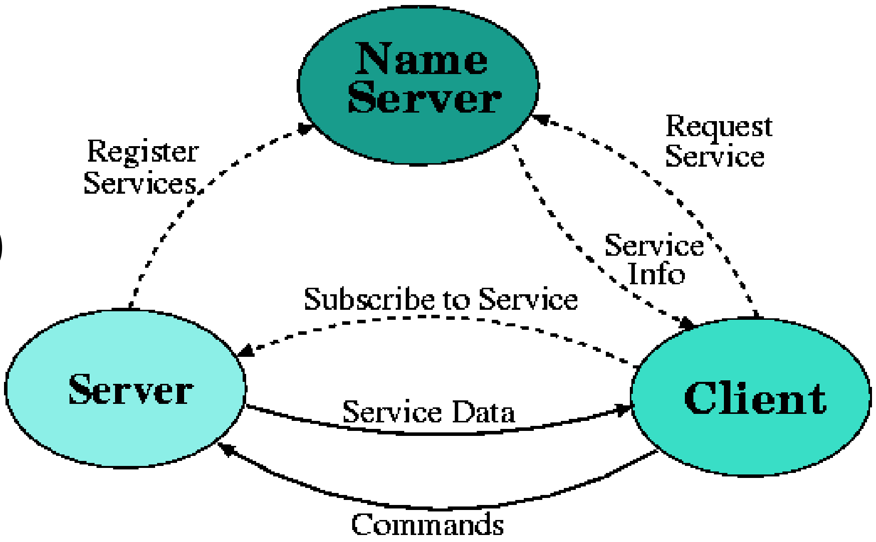
↓ (green) ***Simple Confirmation***

Other Tools & Components

- **Main Framework Components:**
  - Communications
    - Device Access and Message Exchange between processes
  - Intelligence: Finite State Machines/Expert System Functionality
    - System Description, Synchronization and Sequencing
    - Error Recovery, Assistance and Automation
  - Databases
    - Configuration, Archive, Conditions, etc.
  - User Interfaces
    - Visualization and Operation
  - Other Services:
    - Process Management (start/stop processes across machines)
    - Resource Management (allocate/de-allocate common resources)
    - Logging, etc.

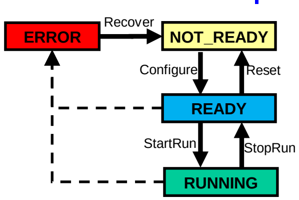
Communications (example)

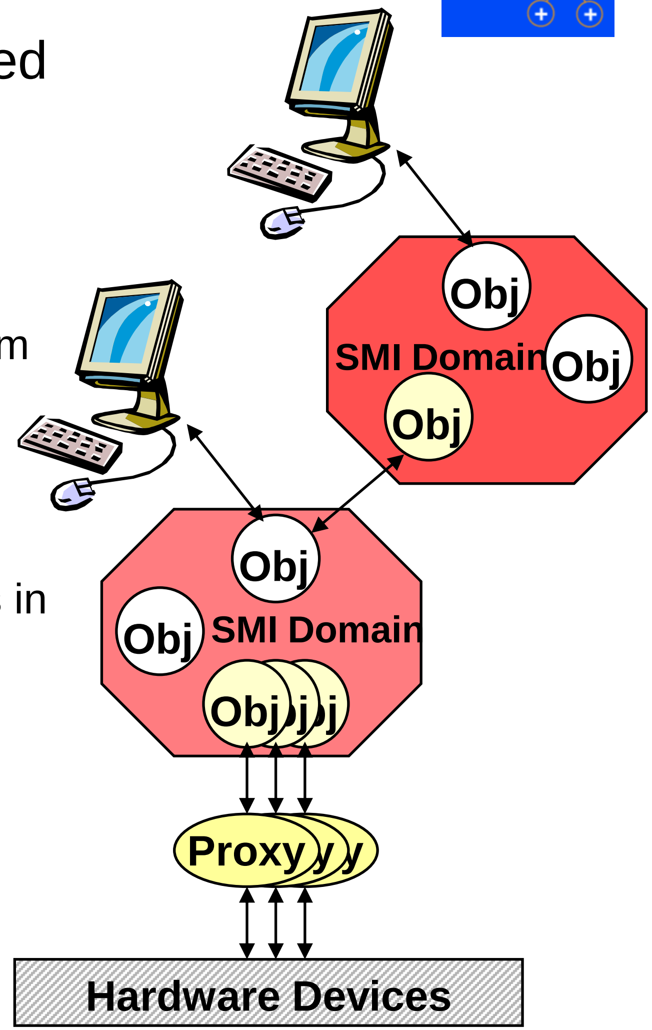
- **DIM – Distributed Information Management System**
  - Efficient and light weight data exchange across processes (and machines)
  - Available for:
    - C, C++, Java, Python
    - Linux, Windows, etc.
  - Client/Server (Publish/Subscribe)
  - Services
    - Set of data, any type or size
      - Single items, arrays or structures
    - Free name space
      - But better use a naming convention
    - Servers publish Services.
    - Clients subscribe to Services.
      - Once, at regular intervals or on change
    - Clients can also send Commands to Servers
  - Name Server
    - Keeps the coordinates of available Services



Automation (example)

- **SMI++ - State Management Interface**
  - A Tool for the Automation of large distributed control systems
  - Method:
    - Classes and Objects
      - Allow the decomposition of a complex system into smaller manageable entities
    - Finite State Machines
      - Allow the modeling of the behavior of each entity and of the interaction between entities in terms of STATES and ACTIONS
    - Rule-based reasoning
      - React to asynchronous events (allow Automation and Error Recovery)





SMI++ - The Language: SML

- Finite State Logic
  - Objects are described as FSMs their main attribute is a STATE
- Parallelism
  - Actions can be sent in parallel to several objects.
- Synchronization and Sequencing
  - The user can also wait until actions finish before sending the next one.
- Asynchronous Rules
  - Actions can be triggered by logical conditions on the state of other objects.

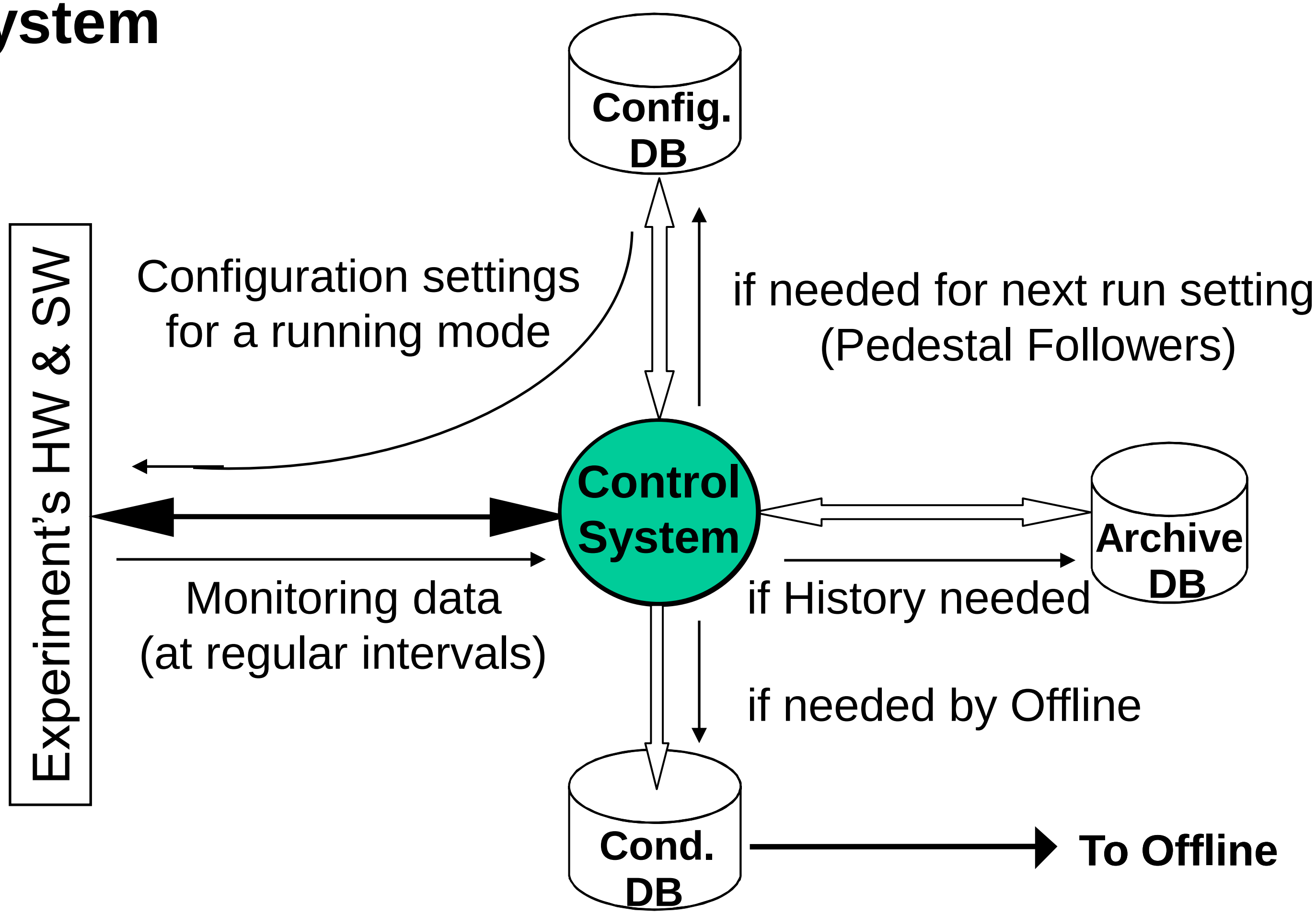
```
class: EventBuilder /associated
   state: UNKNOWN /dead_state
   state: NOT_READY
      action : Configure
   state: READY
      action : Start
      action : Reset
   state: RUNNING
      action : Stop
   state: ERROR
      action : Recover
      ...

object: EB1 is_of_class EventBuilder
object: EB2 is_of_class EventBuilder
object: EB3 is_of_class EventBuilder
...
objectset: EBS {EB1, EB2, EB3, ...}
```

```
class: DAQ
  state: NOT_READY /initial_state
    action: GET_READY
      (do Configure all_in Electronics)
      do Configure all_in EBS
      if (all_in EBS in_state READY) then
        move_to READY
      endif
      move_to ERROR
  state: READY
    when ( any_in EBS in_state ERROR ) do RECOVER
    when ( any_in EBS not_in_state READY ) move_to NOT_READY
    action: RECOVER
      do Recover all_in EBS
    ...
  state: ERROR
    ...
object: SubDetDAQ is_of_class DAQ
```

```
class: TopControl
  state: STANDBY
    when ( LHC in_state PHYSICS ) do STARTUP
    action: STARTUP
      (do GET_READY all_in SubDetDCS)
      do GET_READY all_in SubDetDAQ
      ...
object: BigBrother is_of_class TopControl
```

Online Databases

- Three main logical Database concepts in the Online System



➡ **But naming, grouping and technology can be different in the different experiments…**