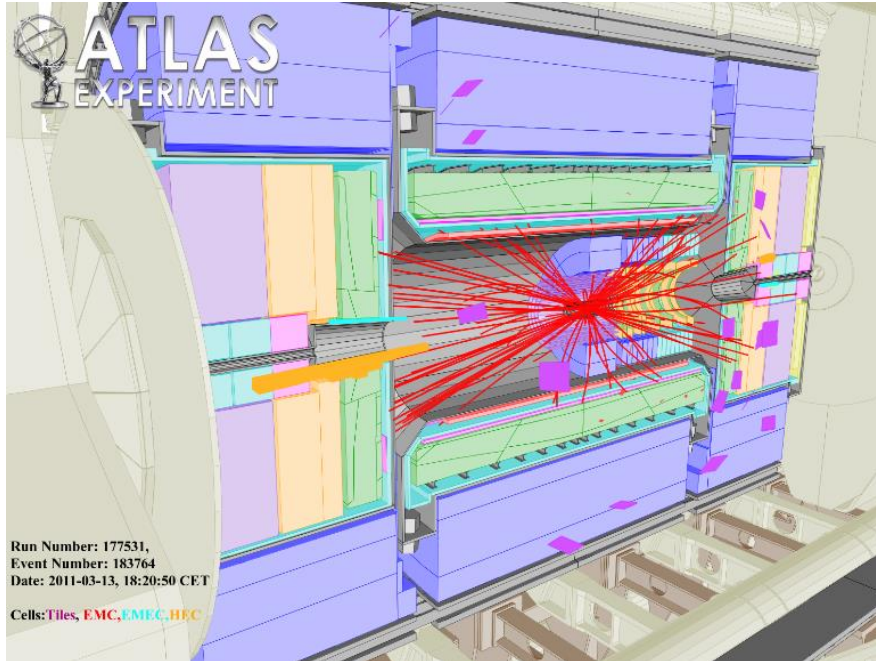


Introduction to Data Acquisition

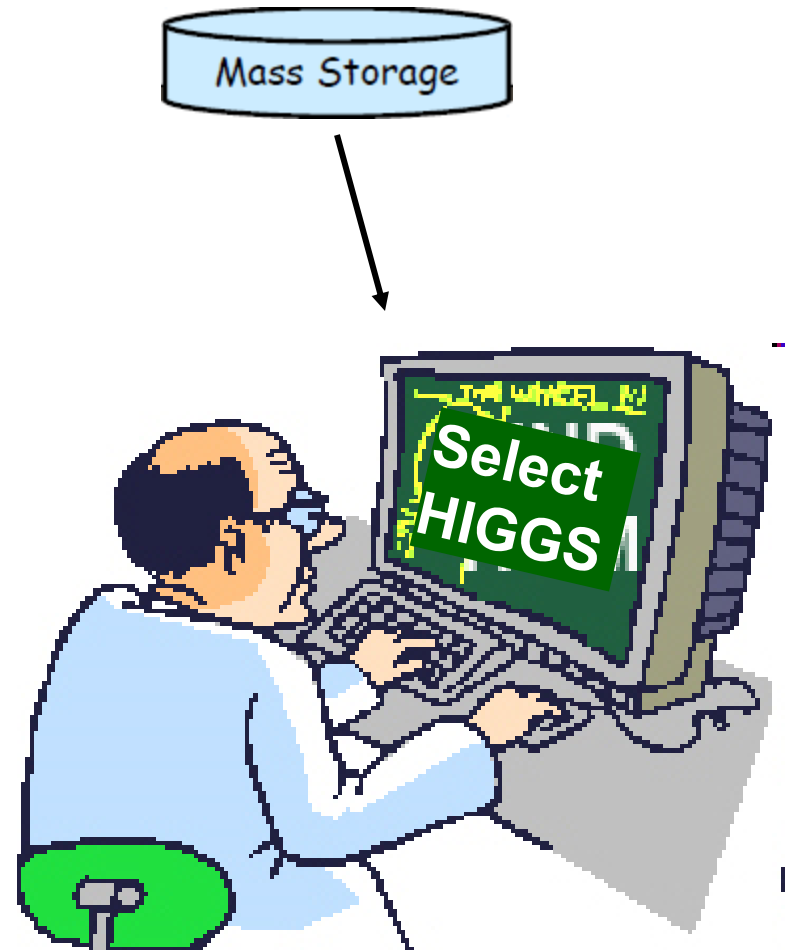
Monika Wielers
Rutherford Appleton
Laboratory

What is it about...

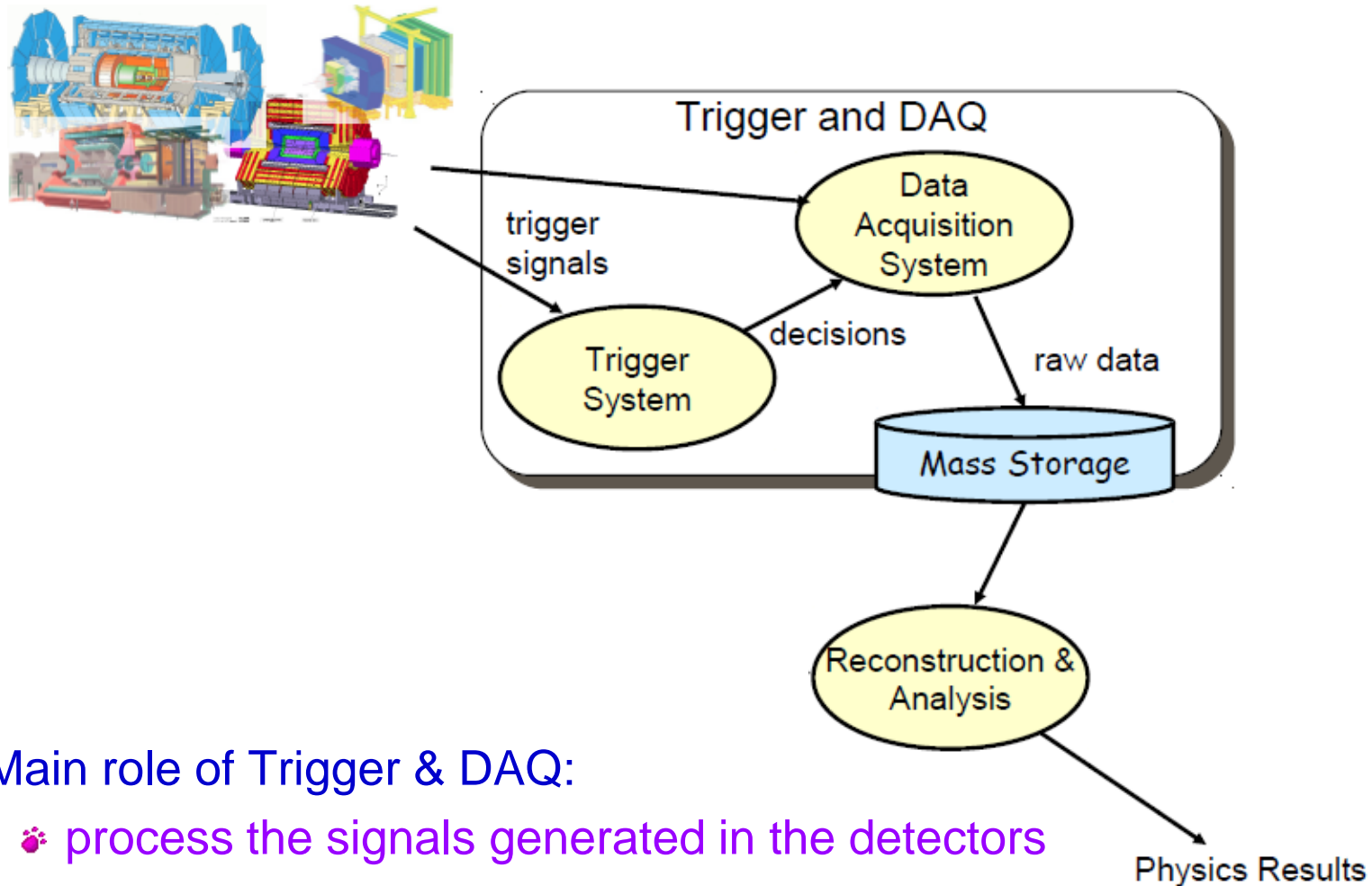
How to get from



to



Or



• Main role of Trigger & DAQ:

- process the signals generated in the detectors
- Select the 'interesting' events and reject the 'boring' ones
- save interesting ones on mass storage for physics analysis

👤 **Heartbeat of the experiment!**

DAQ

- Abbreviation for Data Acquisition System

- Wikipedia:

- process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer.

- In HEP it consists mainly of electronics, computer science, networking and quite some physics

- Tasks

- Gathers the data produced by the detectors (Readout)

- Possibly feeds (several) levels of deciding to keep the collision (called typically **event** in the following)

- Forms complete events (Event Building)

- Stores event data (Data logging)

- Provides control, configuration and monitoring facilities

Trigger

- That's one

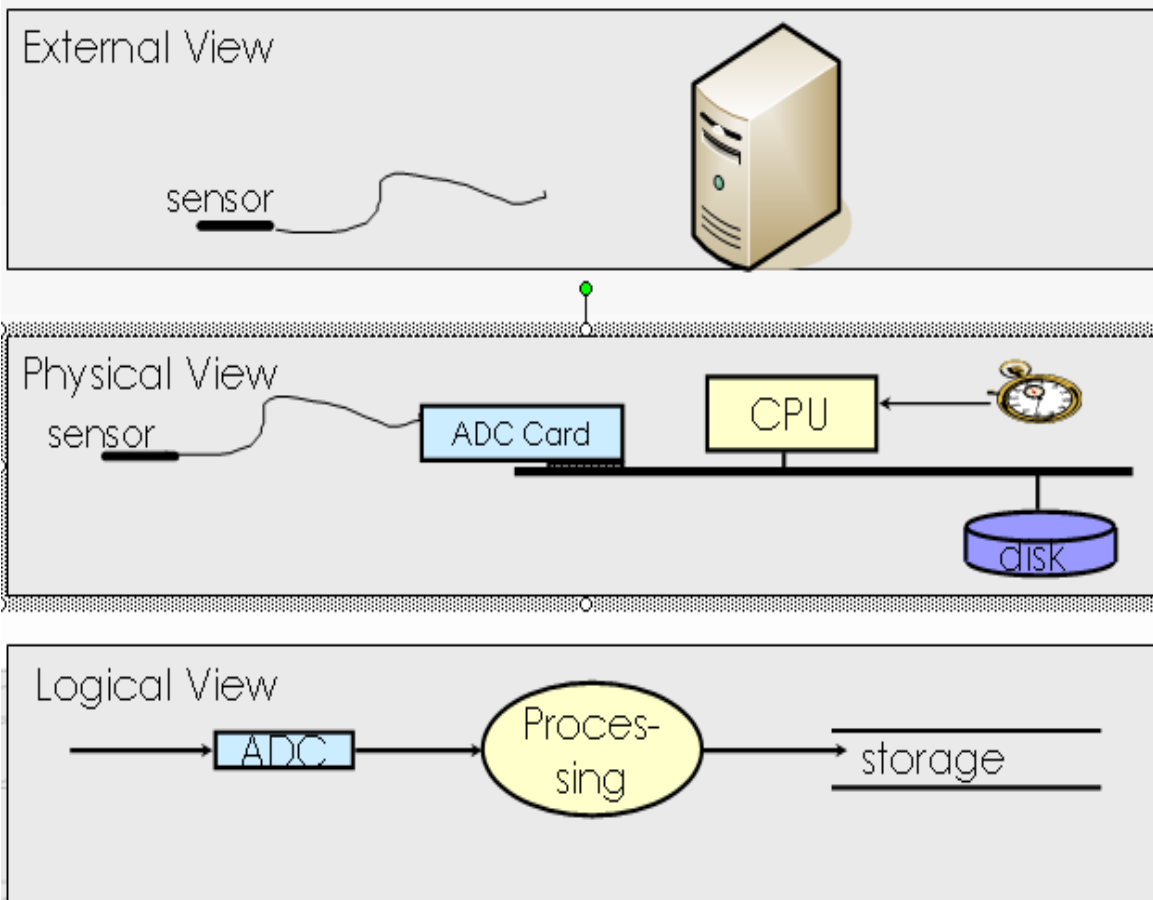


- but that's not what we want to discuss here
- Trigger = in general something which tells you when is the “right” moment to take your data
- Trigger = process to very rapidly decide if you want to keep the data if you can't keep all of them. The decision is based on some ‘simple’ criteria
- This can happen in several stages, if needed
- Note, DAQ and Trigger often are not two separate issues, but are ‘interwoven’

Goals of this lecture

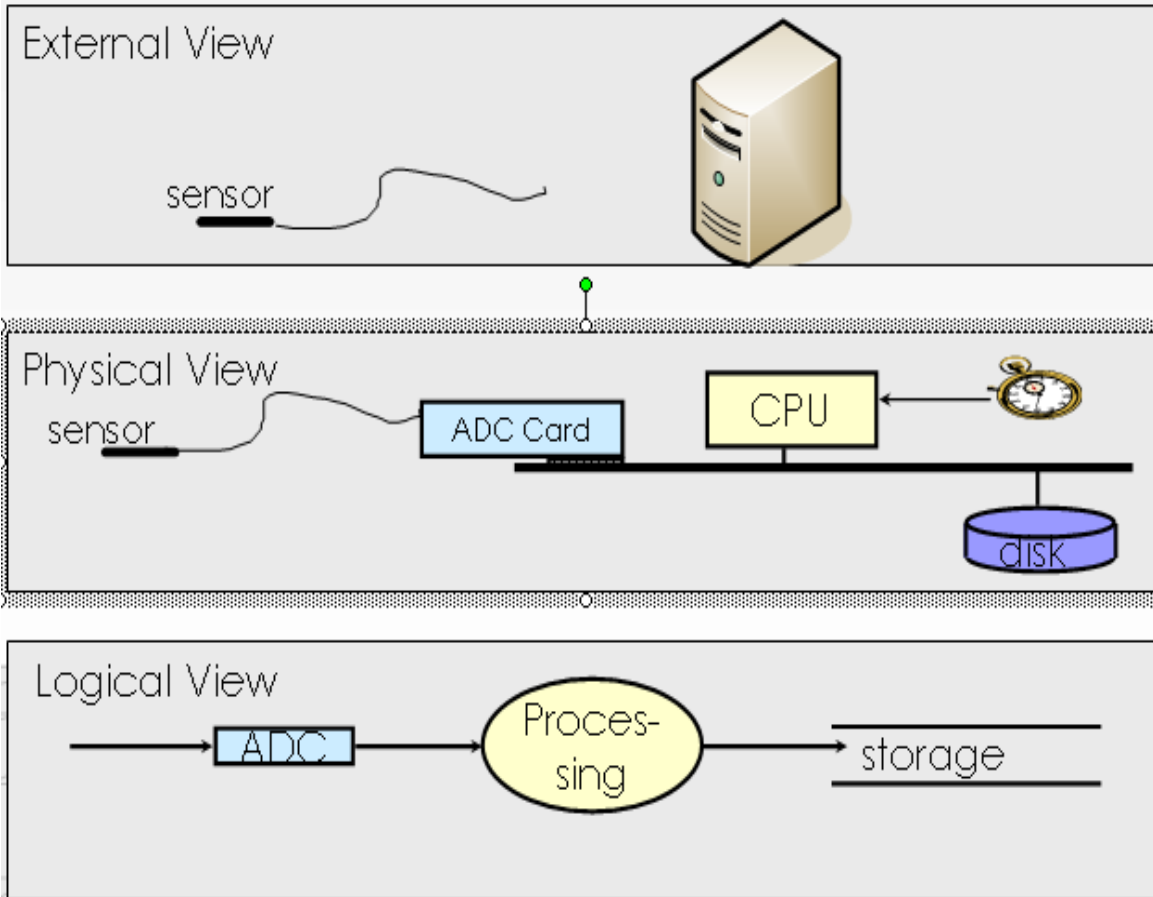
- Understand how data acquisition is devised
 - Start with simple example and then get more complex
- Introduce the terms you will hear when you hear about data acquisition in a particle physics experiment
- Hope this is not too technical and you get an idea what it is about

Trivial DAQ (periodic trigger)



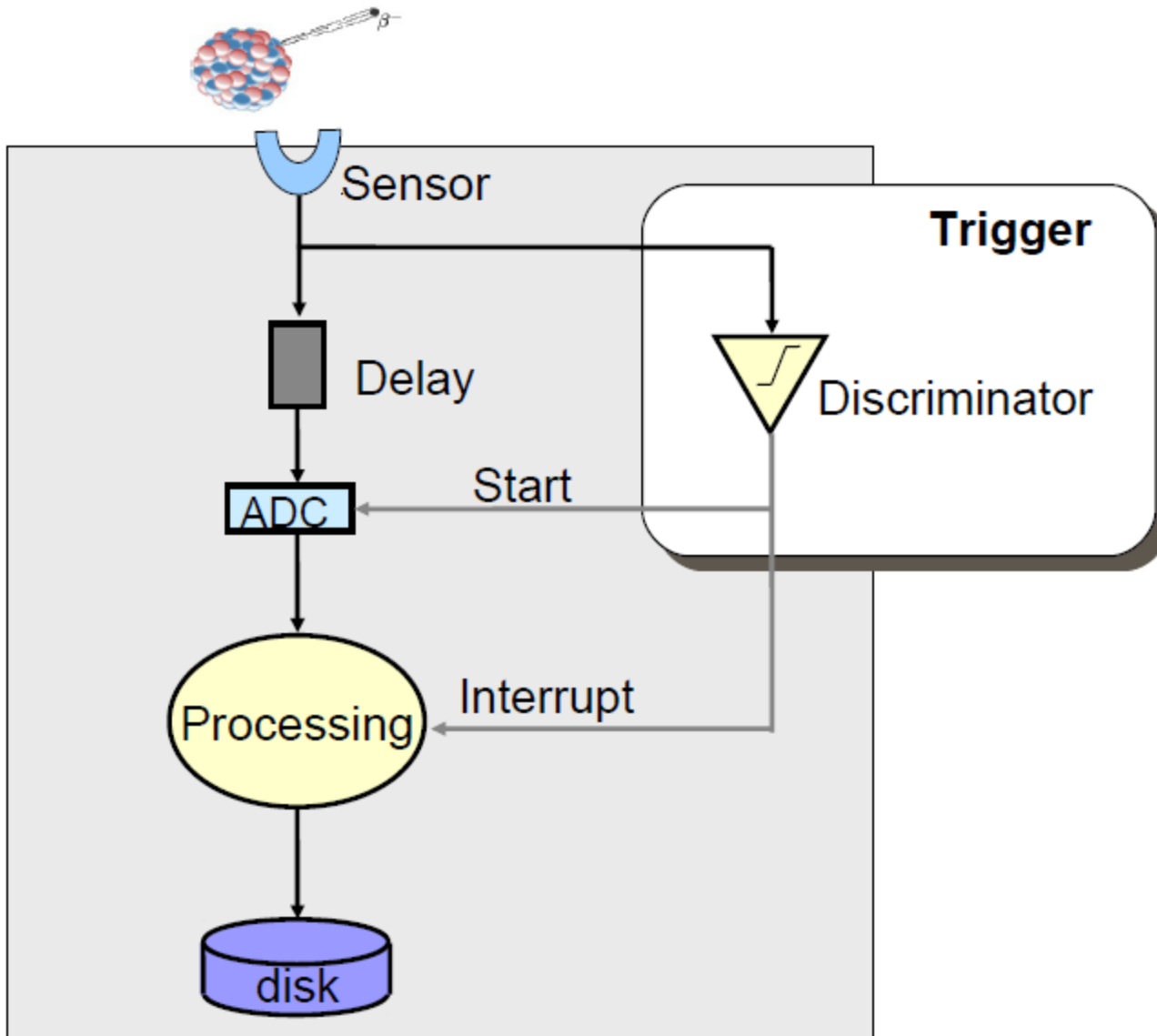
- Measure temperature at a fixed frequency
- ADC performs analog to digital conversion (digitisation)
 - Our frontend electronics
- CPU does readout and processing

Trivial DAQ (periodic trigger)



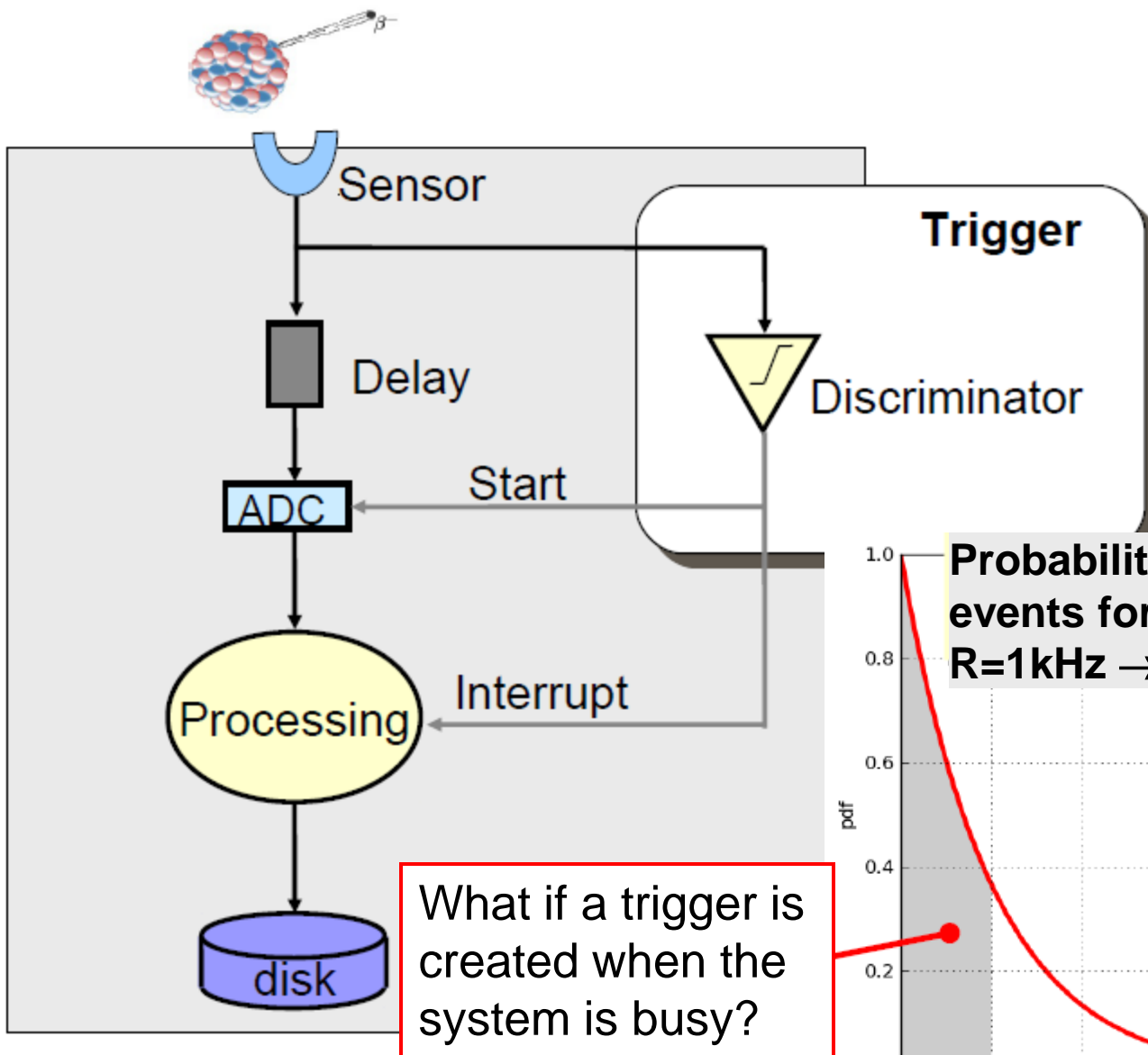
- Measure temperature at a fixed frequency
- The system is clearly limited by the time to process a measurement (or event)
- Example $\tau=1\text{ ms}$ to
 - ADC conversion
 - +CPU processing
 - +Storage
- Sustain $\sim 1/1\text{ ms}=1\text{ kHz}$
periodic trigger rate

Trivial DAQ with a trigger

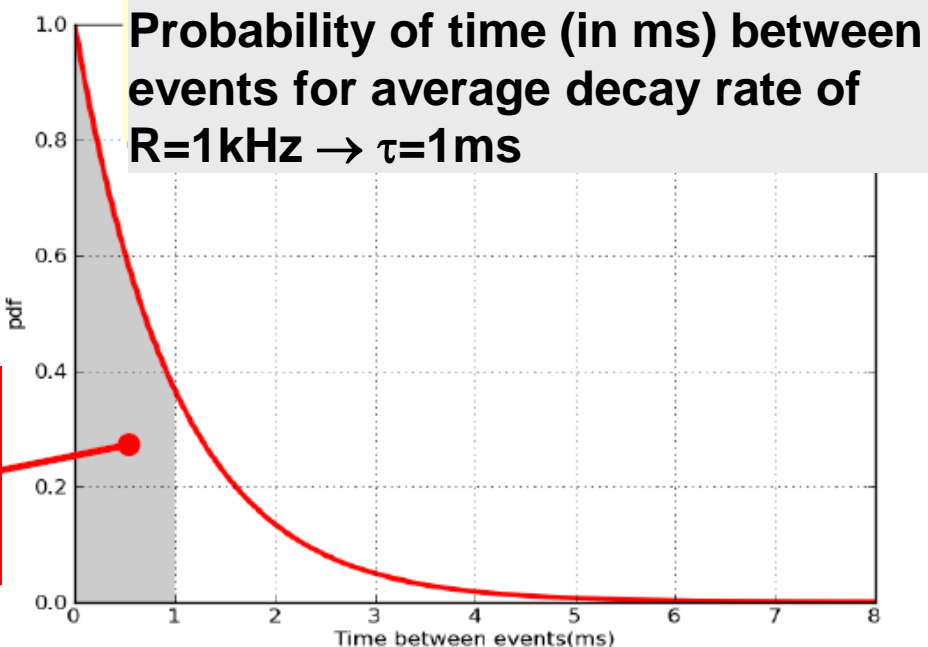


- Example: measure β decay properties
- Our events are asynchronous and unpredictable
 - Need a **physics** trigger
- Delay compensates for the trigger latency

Trivial DAQ with a trigger



- Example: measure β decay properties
- Need a **physics** trigger



What if a trigger is created when the system is busy?

Deadtime and Efficiency

- **Deadtime** (%): ratio between the time the DAQ is busy and the total time (Dead time / trigger = τ sec.)

- $\nu \cdot \tau \rightarrow$ DAQ is busy — $(1 - \nu \cdot \tau) \rightarrow$ DAQ is free, $\nu =$ output rate

$$\nu = f(1 - \nu\tau) \Rightarrow \nu = \frac{f}{1 + f\tau} < f$$

f input rate
 τ readout time

- **Efficiency:** $N_{\text{saved}}/N_{\text{tot}} = 1/(1 + f \cdot \tau)$

- Note, due to the fluctuations introduced by the stochastic process the efficiency will always be less 100%

- In our specific example, $d=0.1\%/Hz$, $f=1kHz \rightarrow \nu=500Hz$, $\varepsilon=50\%$

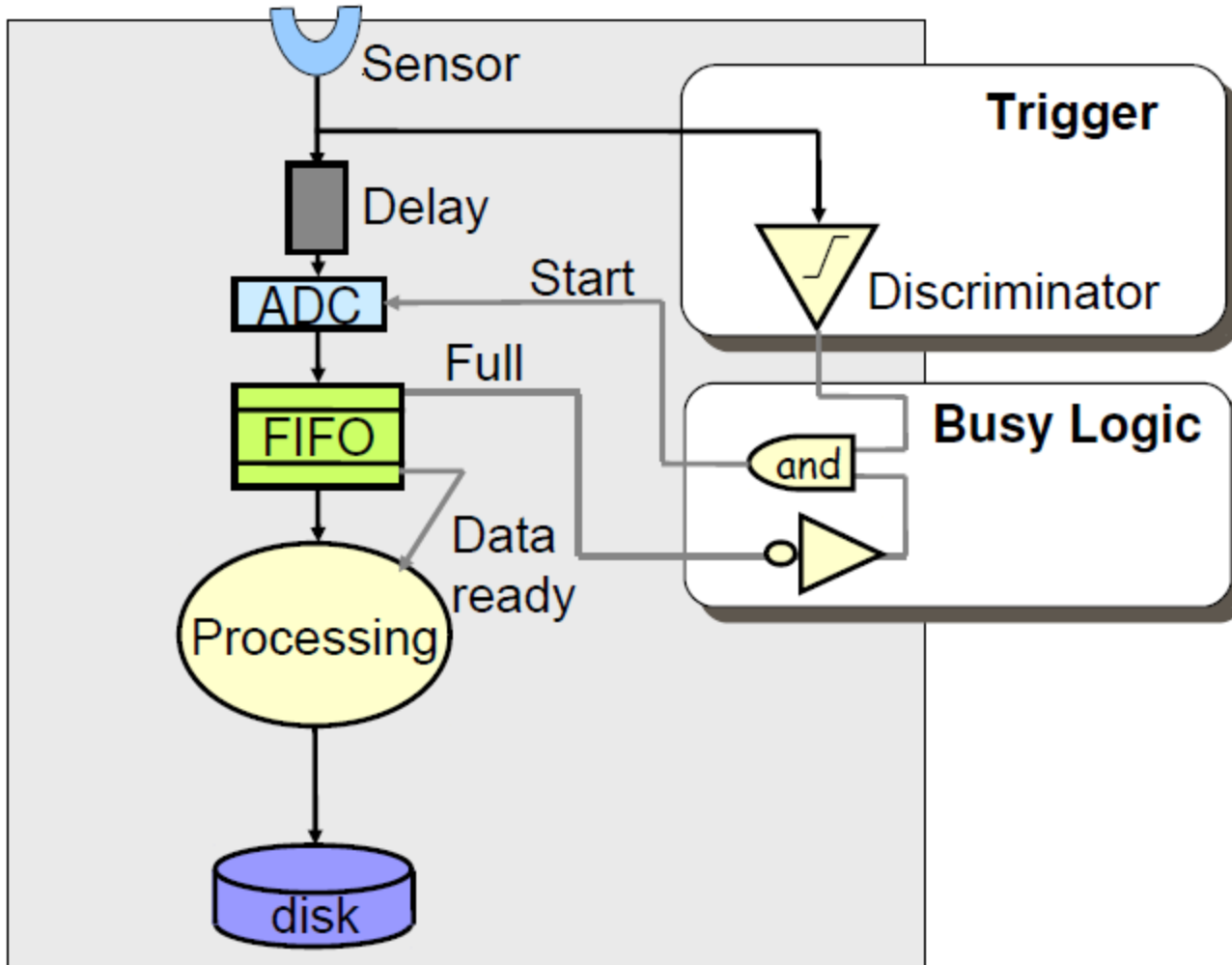
- If we want to obtain $\varepsilon \sim 100\% \rightarrow f \cdot \tau \ll 1$

- $f=1kHz$, $\varepsilon=99\% \rightarrow \tau < 0.1ms \rightarrow 1/\tau > 10kHz$

- In order to cope with the input signal fluctuations, we would need to overdesign our DAQ system by a factor 10. hmmm...

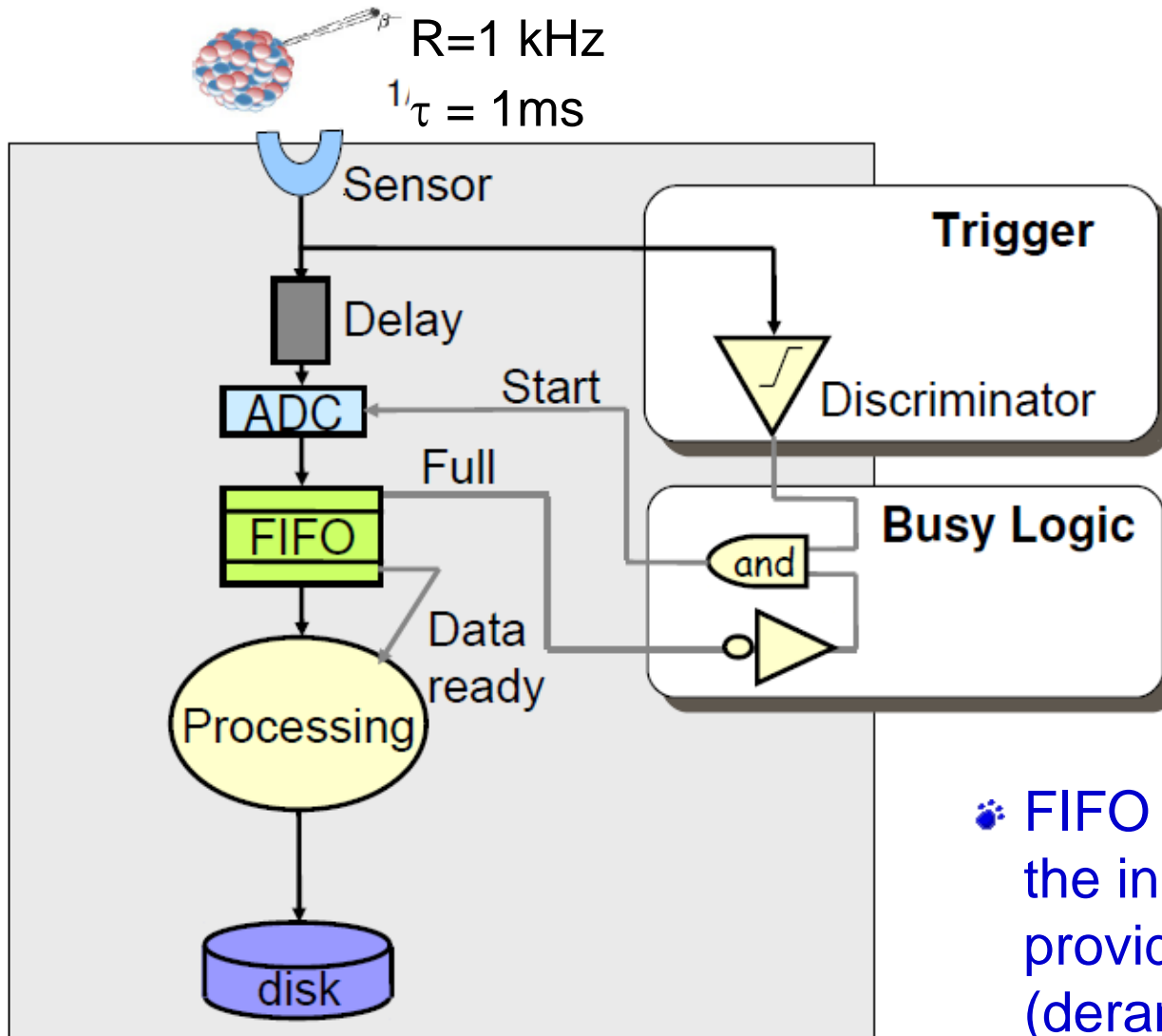
Trivial DAQ with Derandomisation

$R=1\text{ kHz}$
 $\tau = 1\text{ ms}$



- Buffers are introduced which hold temporarily the data.
- They decouple the data production from the data processing → Better performance

Trivial DAQ with Derandomisation



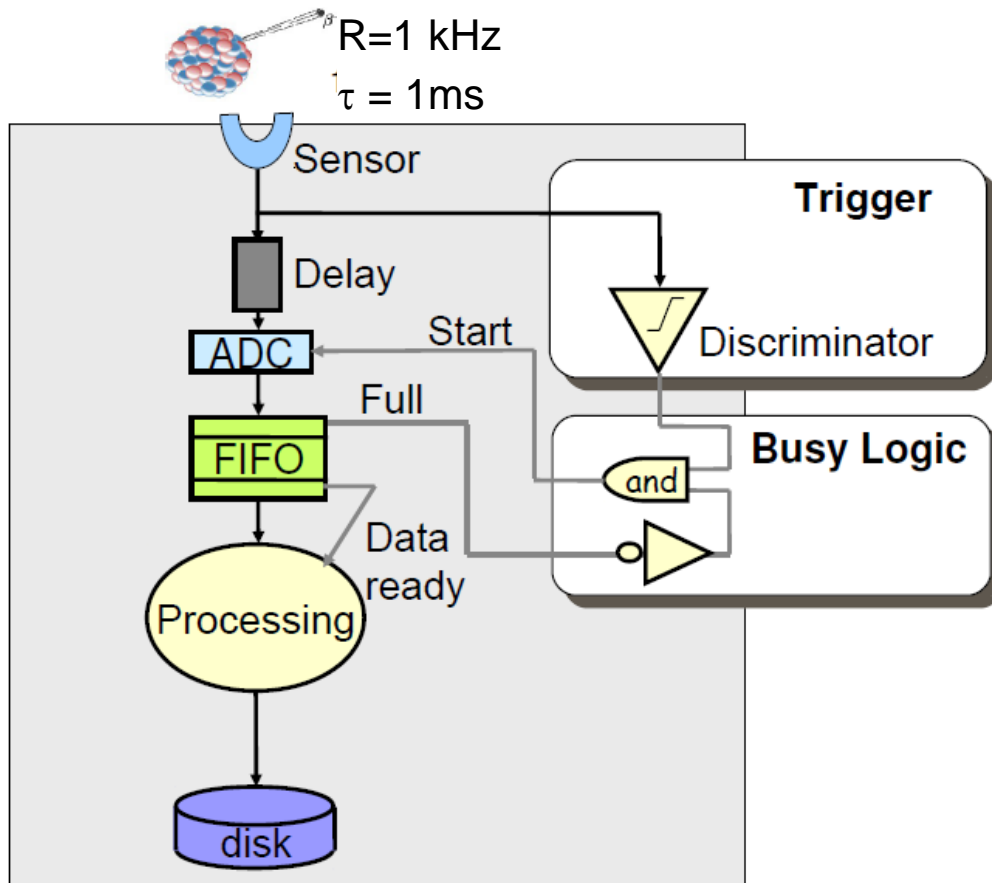
• First In First Out

- Buffer area organized as a queue
- Depth: number of cells
- Implemented in HW and SW



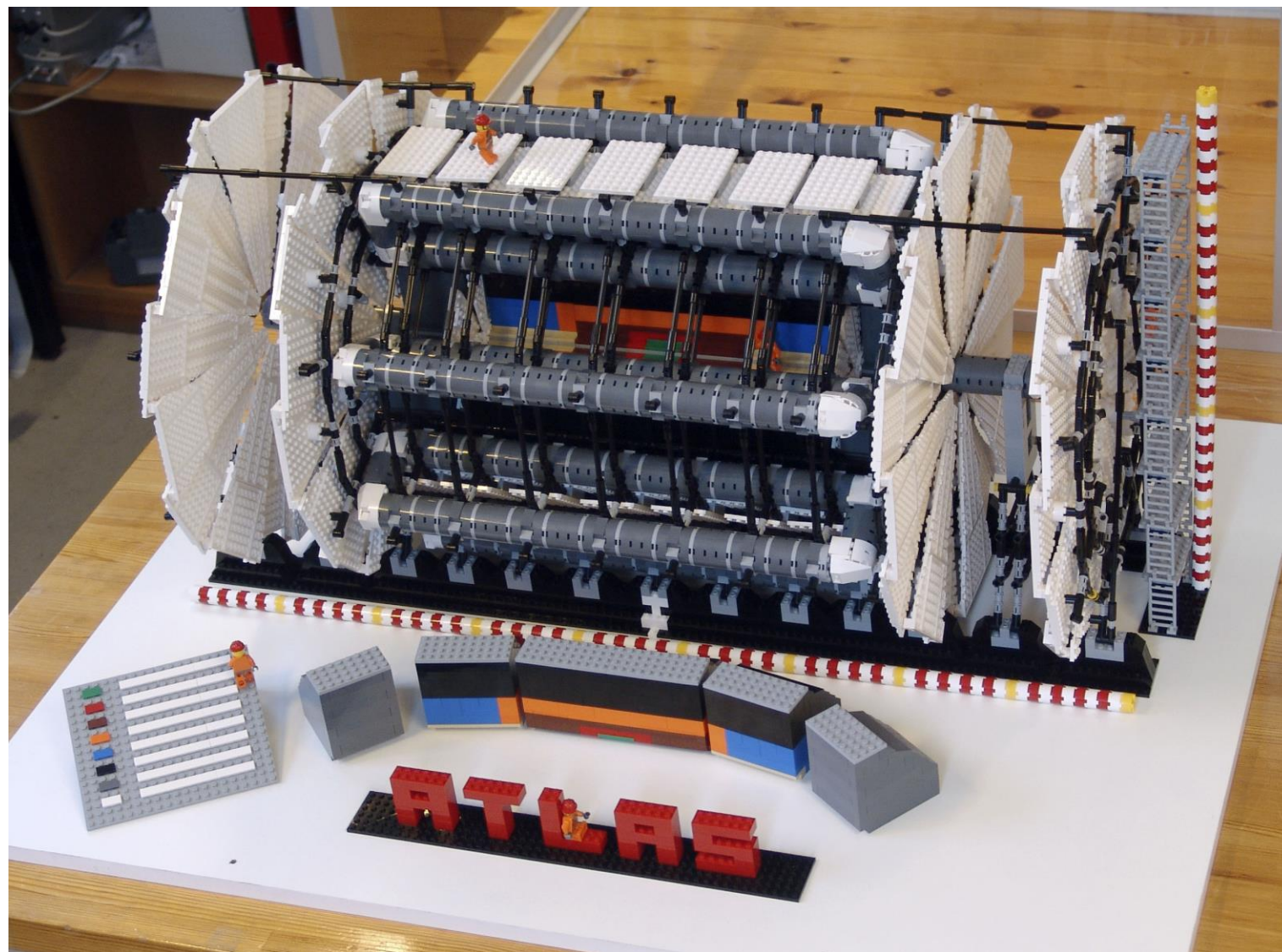
- FIFO absorbs and smoothes the input fluctuations, providing a ~steady (derandomized) output rate
- introduces an additional latency on the data path

Trivial DAQ with Derandomisation

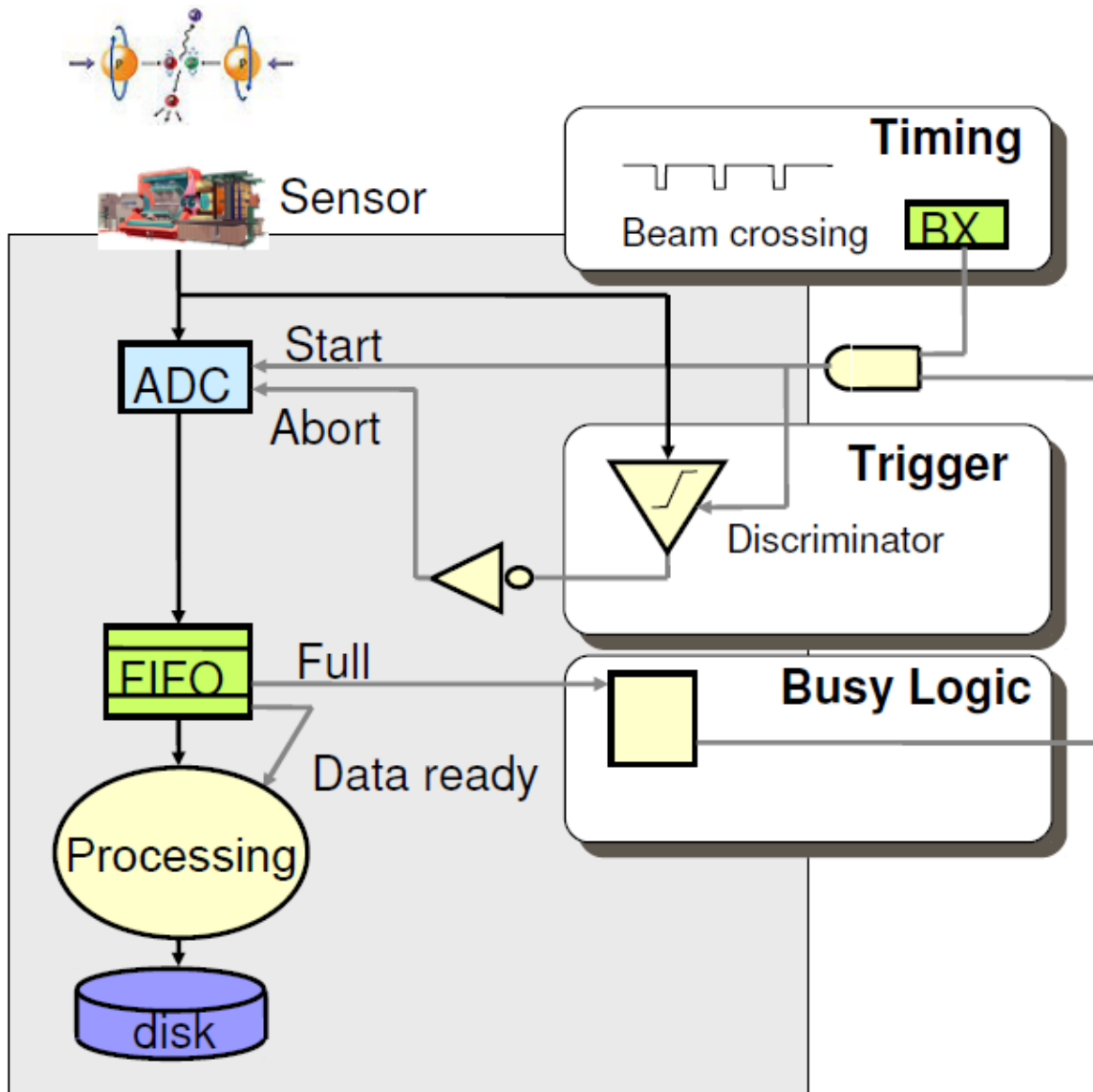


- Almost 100% efficiency and minimal deadtime if
 - ADC is able to operate at rate $\gg f$
 - Data processing and storing operates at $\sim f$
- Minimises the amount of “unnecessary” fast components
- Could the delay be replaced with a “FIFO”?
 - Analog pipelines \rightarrow Heavily used in LHC DAQs

Let's have a closer look at DAQ at a collider

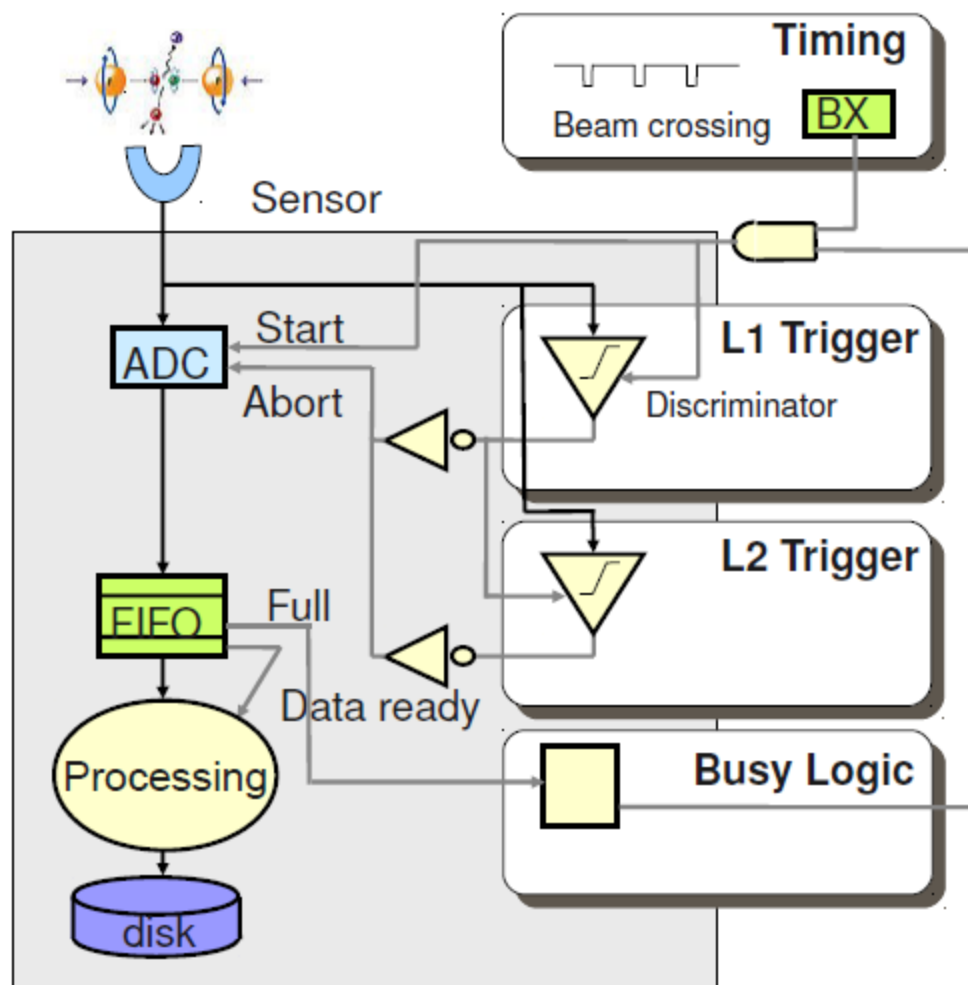


DAQ: Collider mode



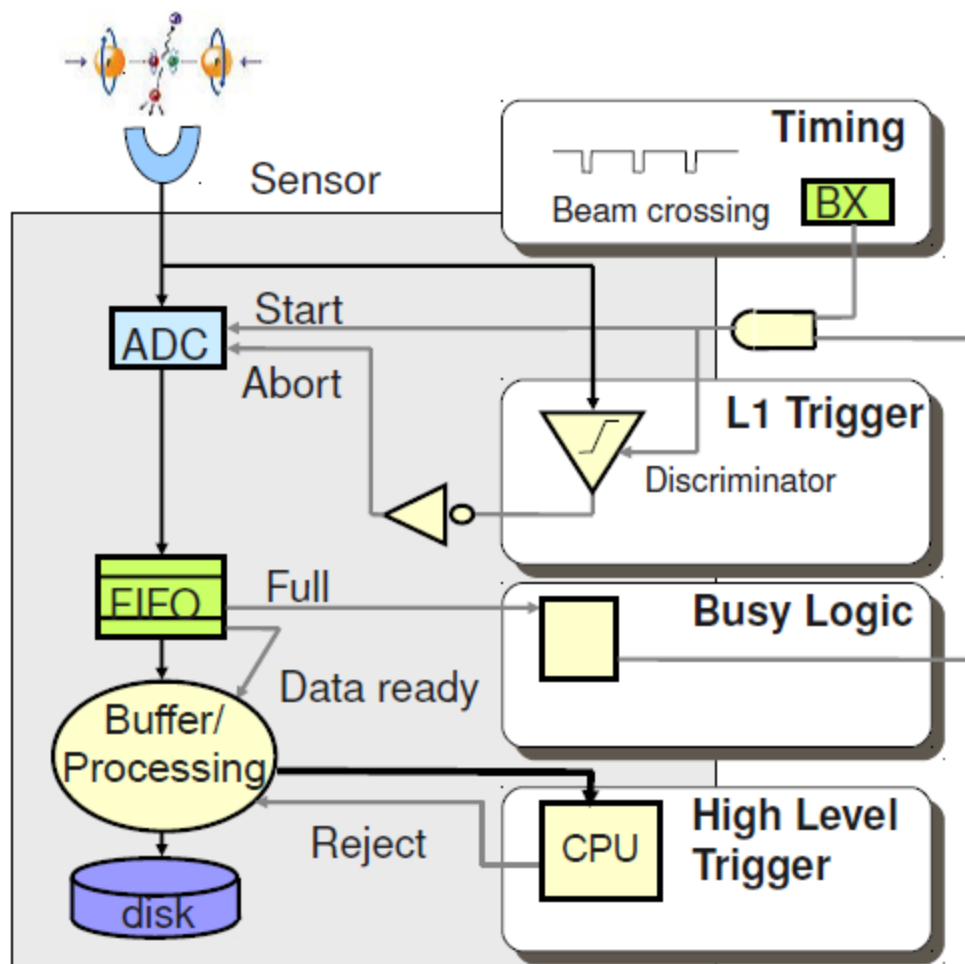
- Particle collisions are synchronous
- Trigger rejects uninteresting events
- Even if collisions are synchronous, the triggers (interesting events) are unpredictable
- Derandomisation is still needed
- No trigger deadtime if trigger latency below time between two beam crossings

Multi-Level Trigger



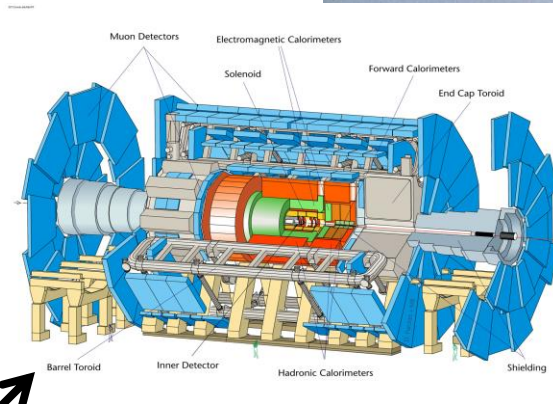
- For complicated triggers latency can be long
 - if $\tau_{\text{trig}} > \tau_{\text{BX}}$,
deadtime > 50%
- Split trigger in several levels with increasing complexity and latency
- All levels can reject events
 - with $\tau_{\text{L1}} < \tau_{\text{BX}}$, trigger deadtime only
 $\nu_{\text{L1}} \cdot \tau_{\text{L2}}$

Multi-Level Trigger

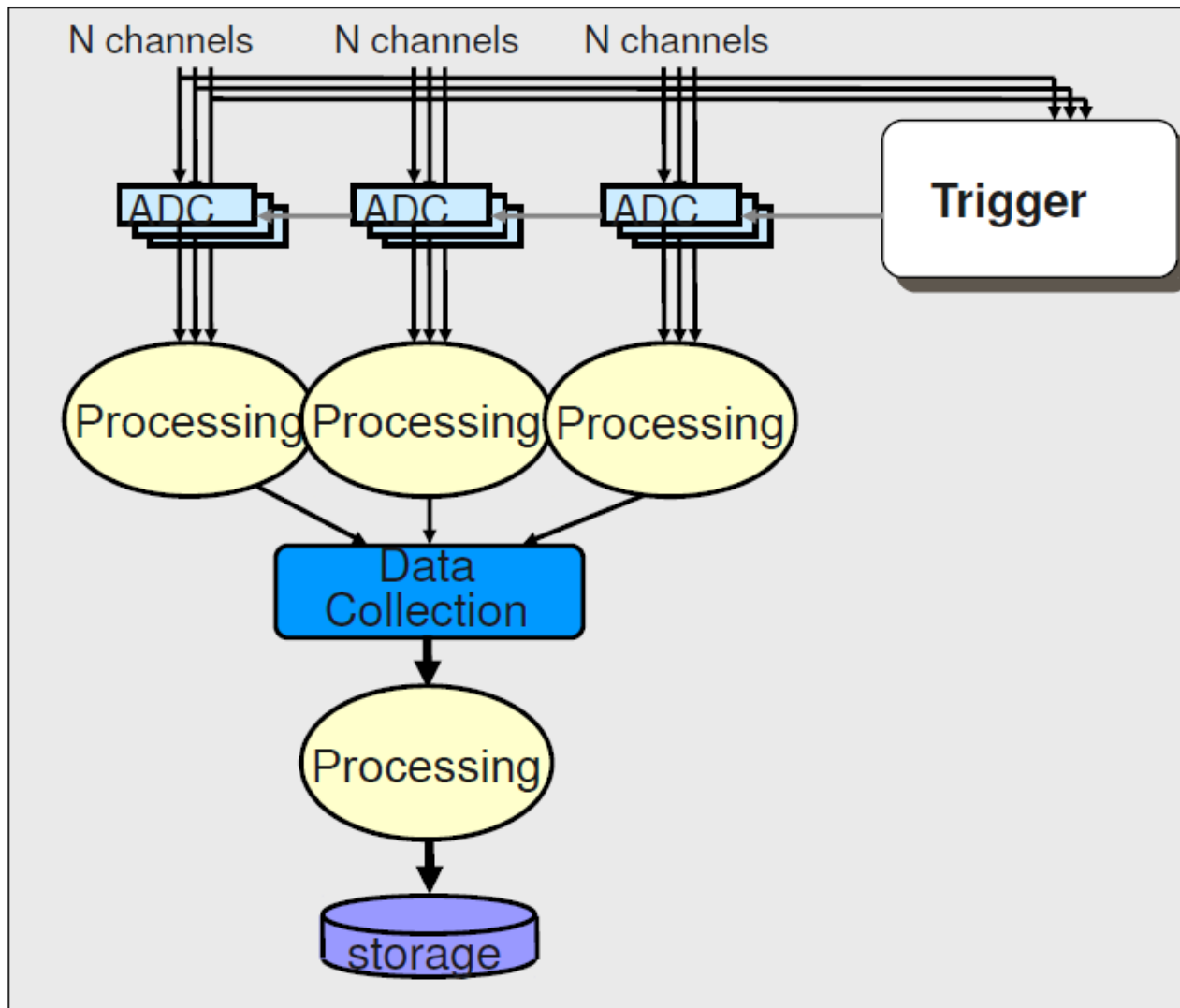


- For optimal data reduction can add trigger level between readout and storage (High-level trigger)
- Has access to some/all processed data

Scaling up

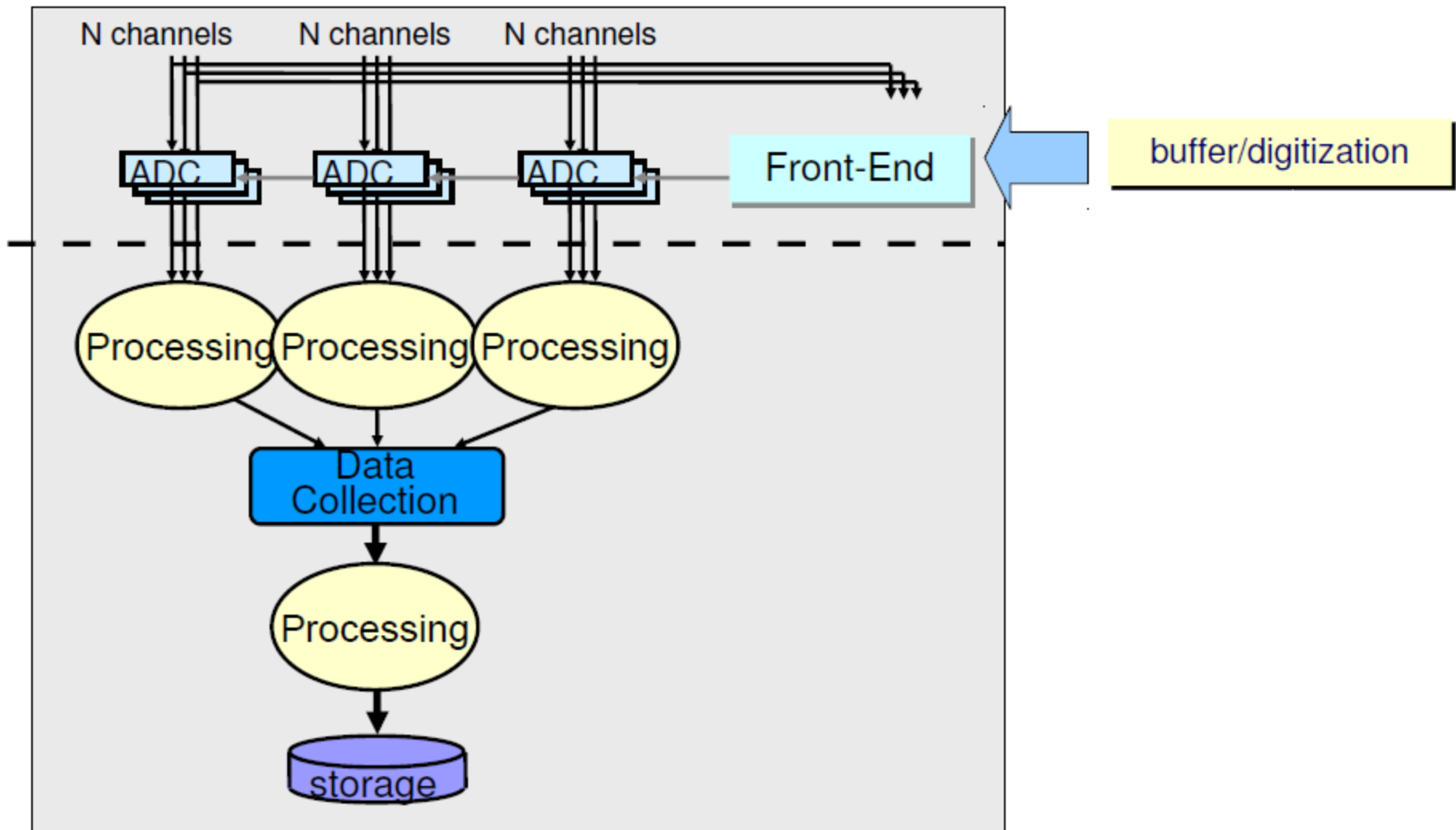


A bit more complicated....

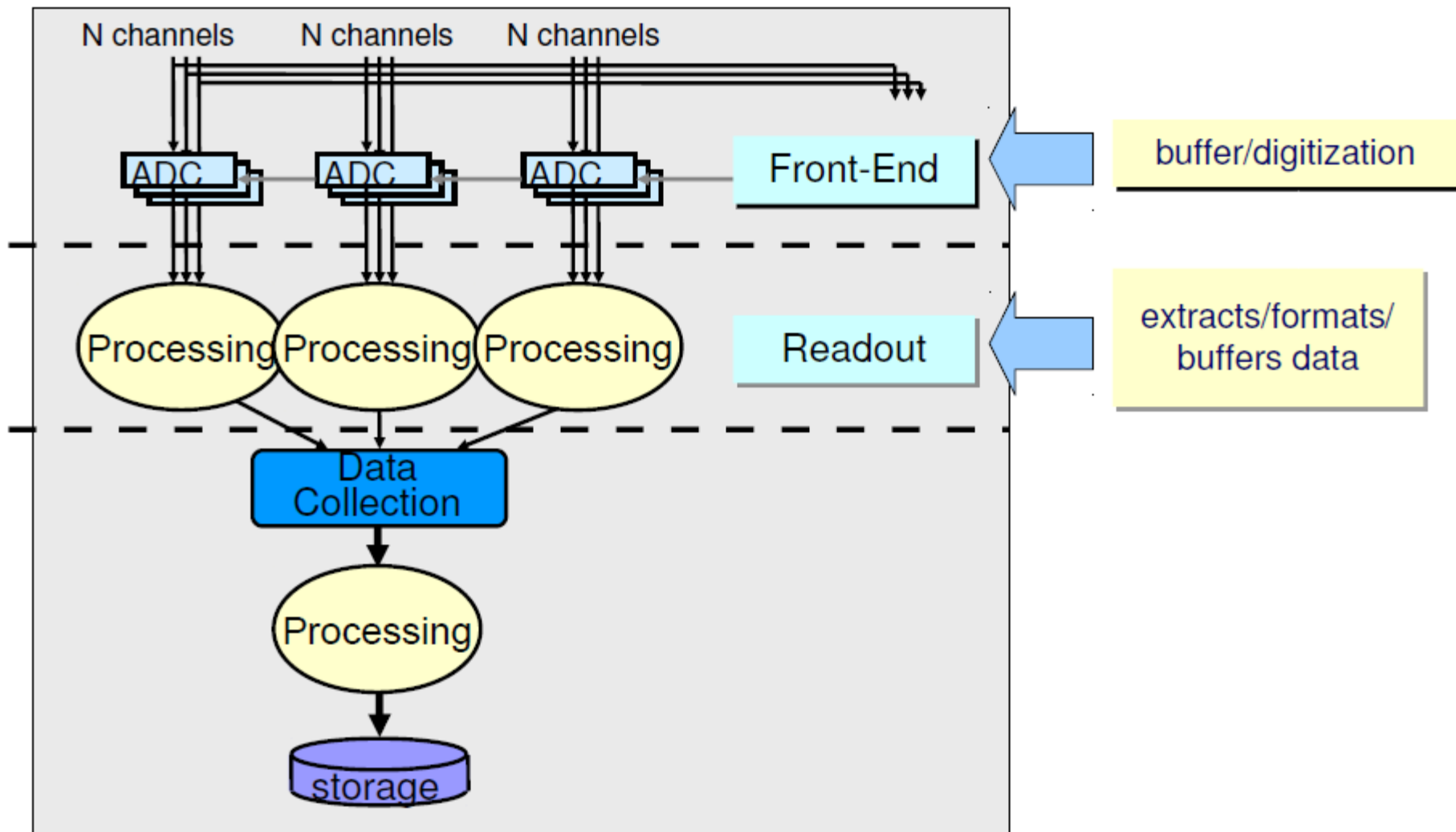


- The increased number of channels require hierarchical structure with well defined interfaces between components

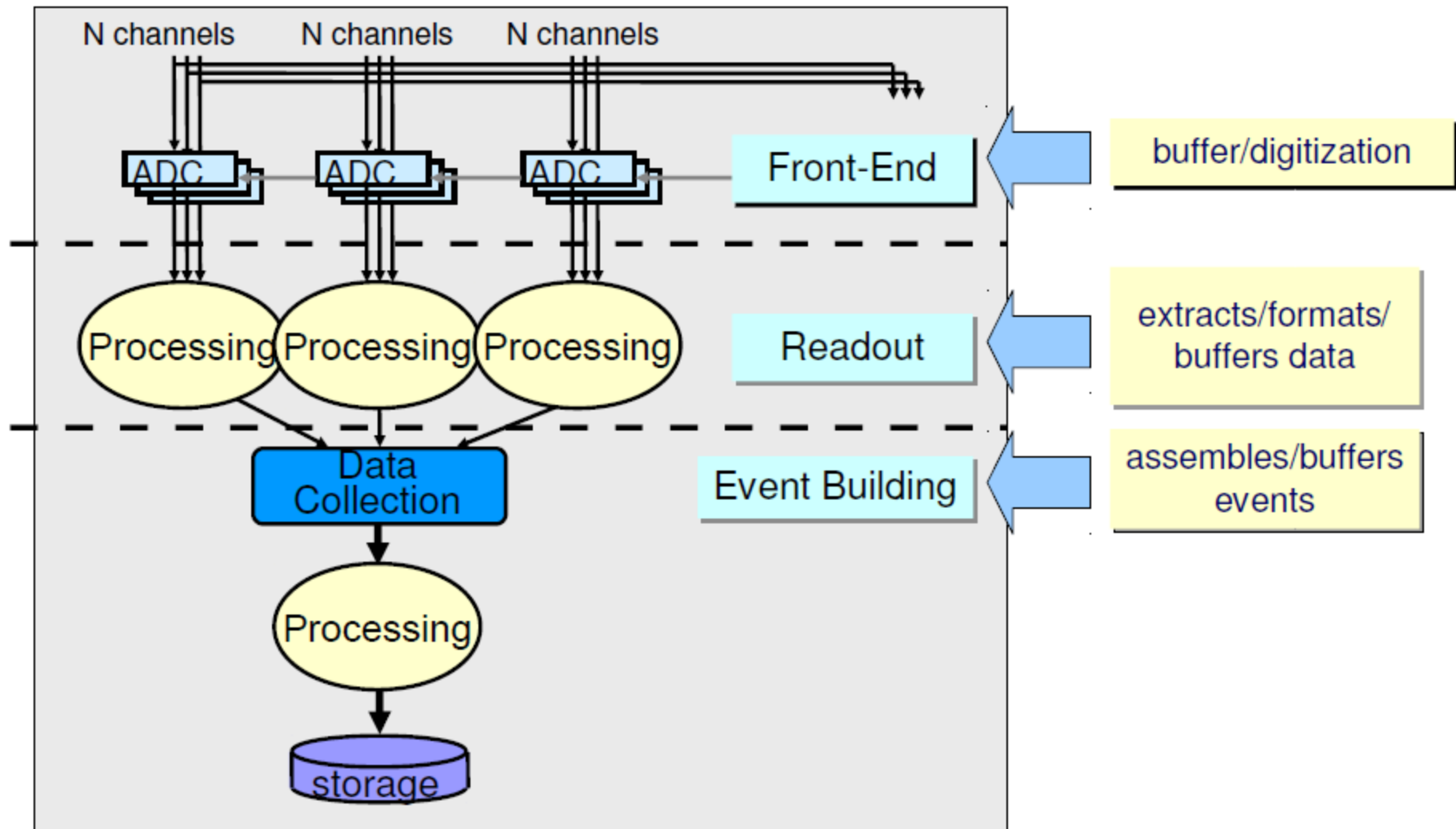
Large DAQ: constituents



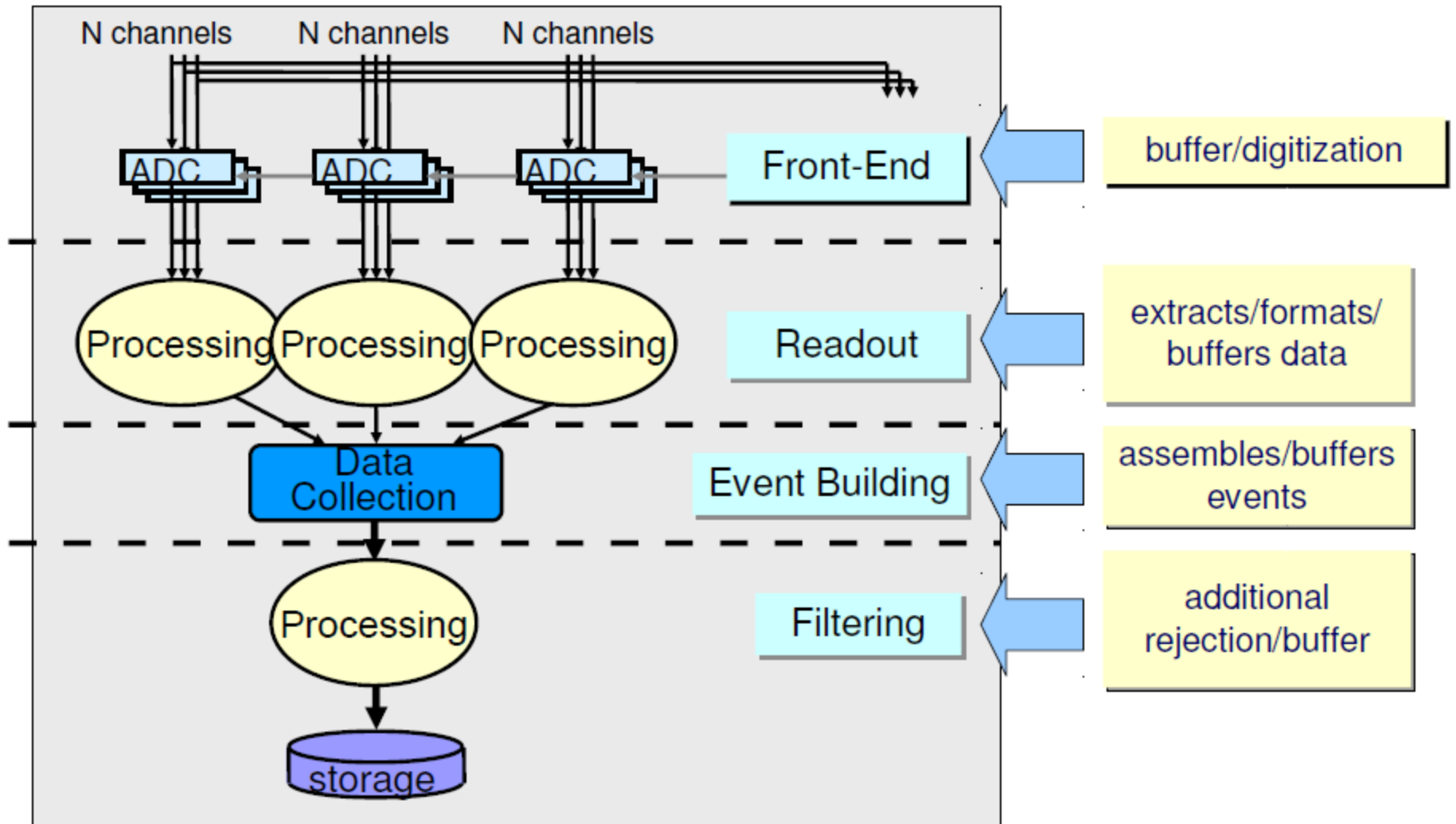
Large DAQ: constituents



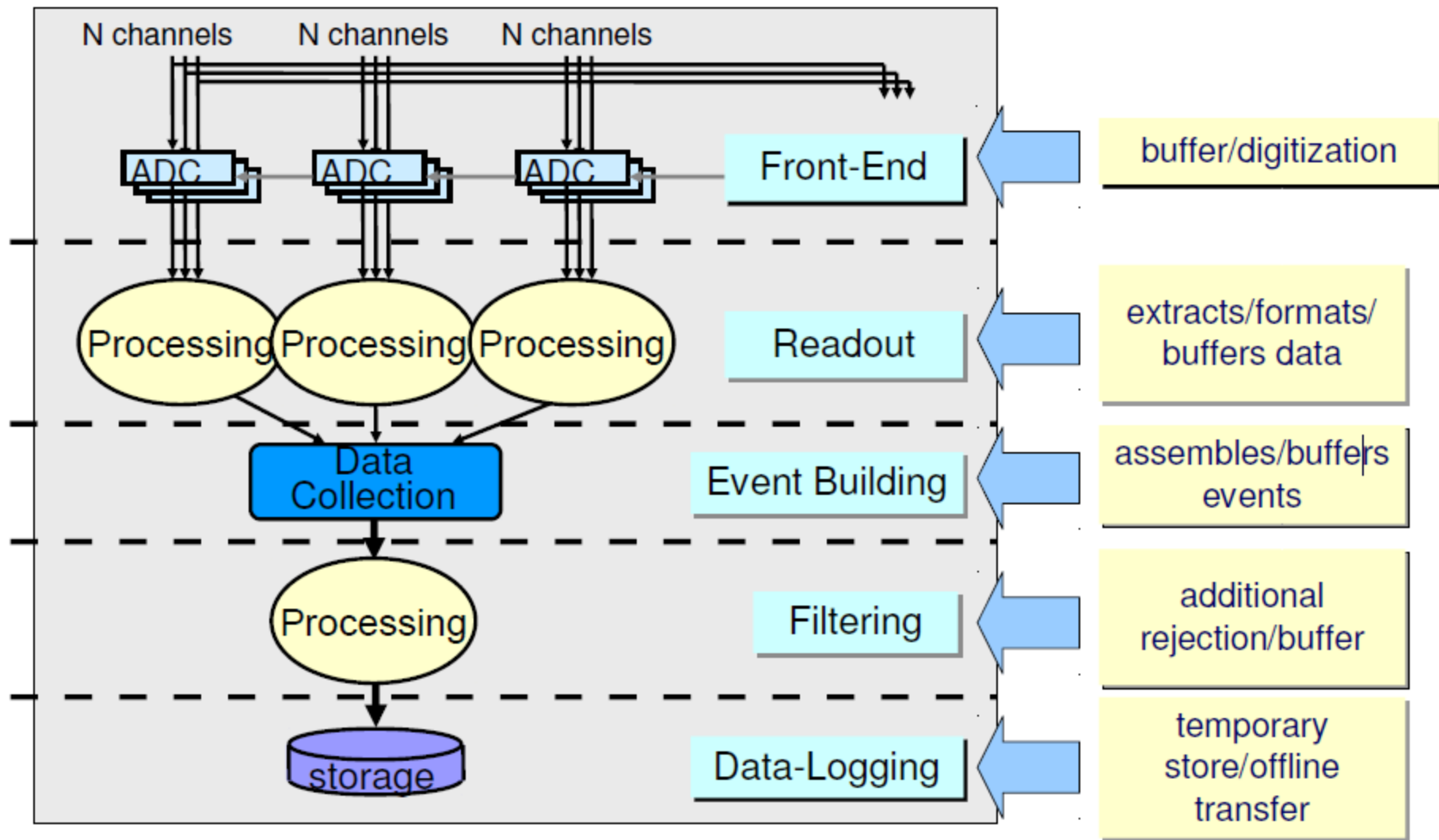
Large DAQ: constituents



Large DAQ: constituents

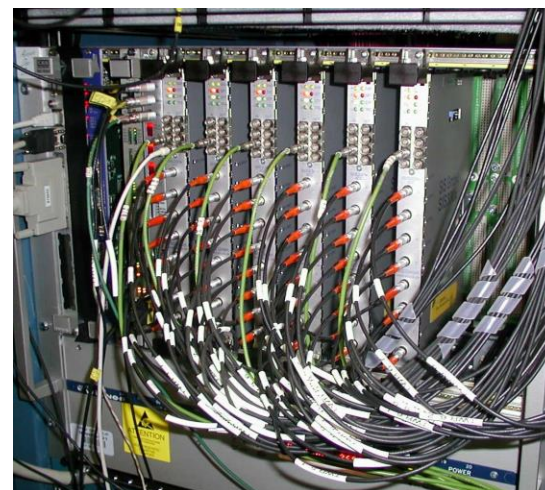
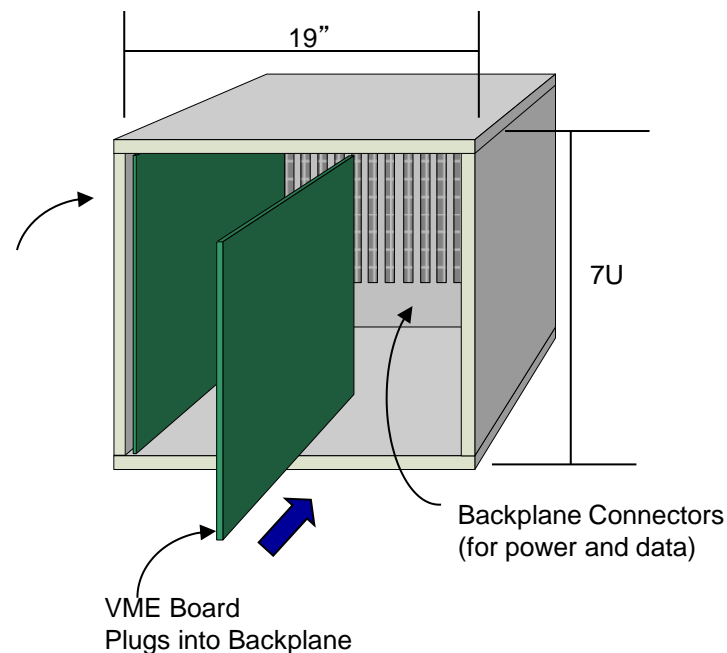


Large DAQ: constituents



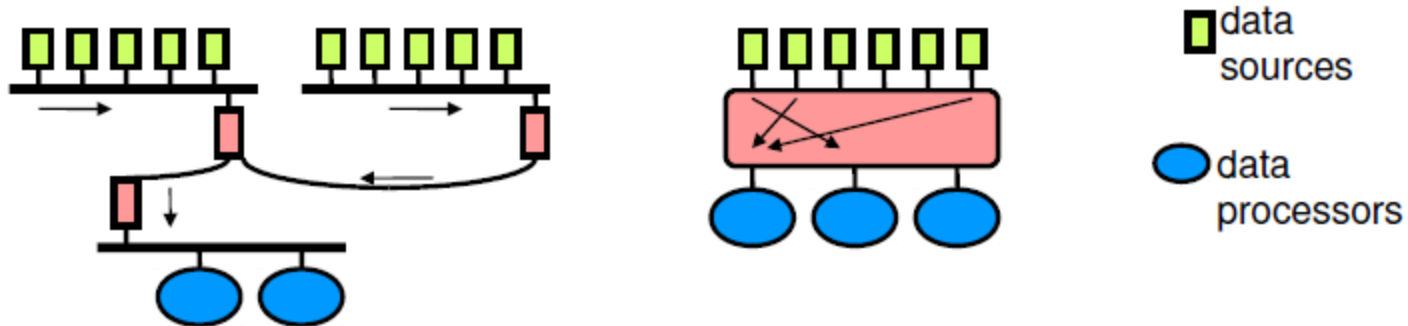
Read-out Topology

- Reading out = building events out of many detector channels
- We define “building blocks”
 - Example: readout crates, event building nodes, ...
- Crate: many modules put in a common chassis which provides
 - Mechanical support
 - Power
 - A standardised way to access the data
 - Provides signal and protocol standard for communication
- All this is provided by standards for (readout) electronics such as **VME** (IEEE 1014)



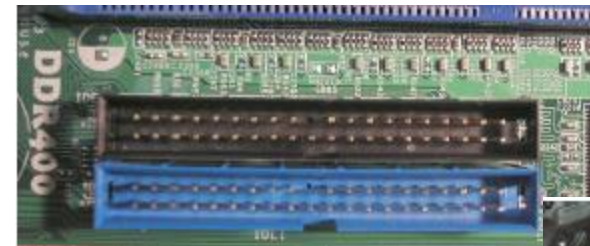
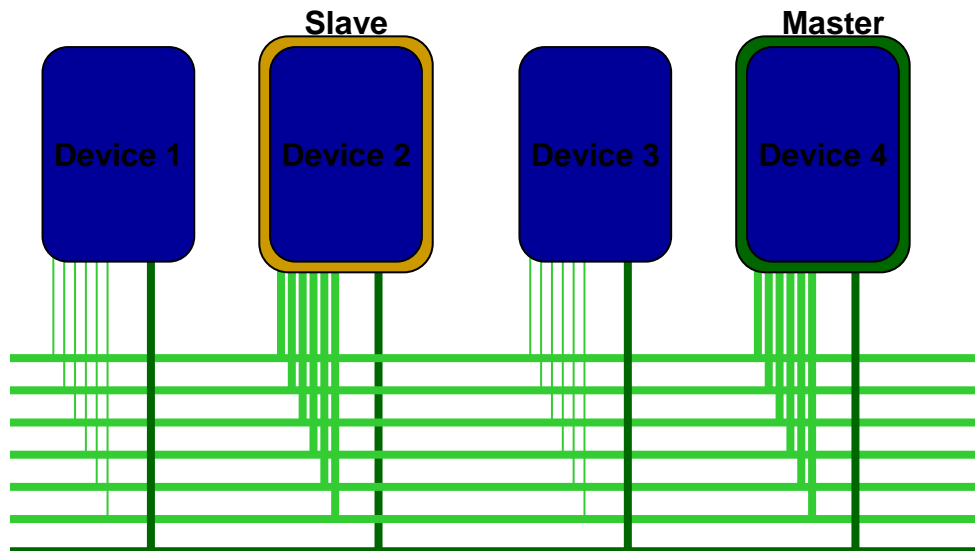
Read-out Topology

- How to organize the interconnections inside the building blocks and between building blocks?
- Two main classes: **bus** or **network**
 - Both of them are very generic concepts



Bus

- ❧ A bus connects two or more devices and allows them to communicate
 - ❧ Bus → group of electrical lines
- ❧ Examples: VME, PCI, SCSI, Parallel ATA, ...
- ❧ The bus is **shared** between all devices on the bus → arbitration is required
- ❧ Devices can be **masters** or **slaves** (some can be both)
- ❧ Devices can be uniquely identified ("**addressed**") on the bus



Data Lines

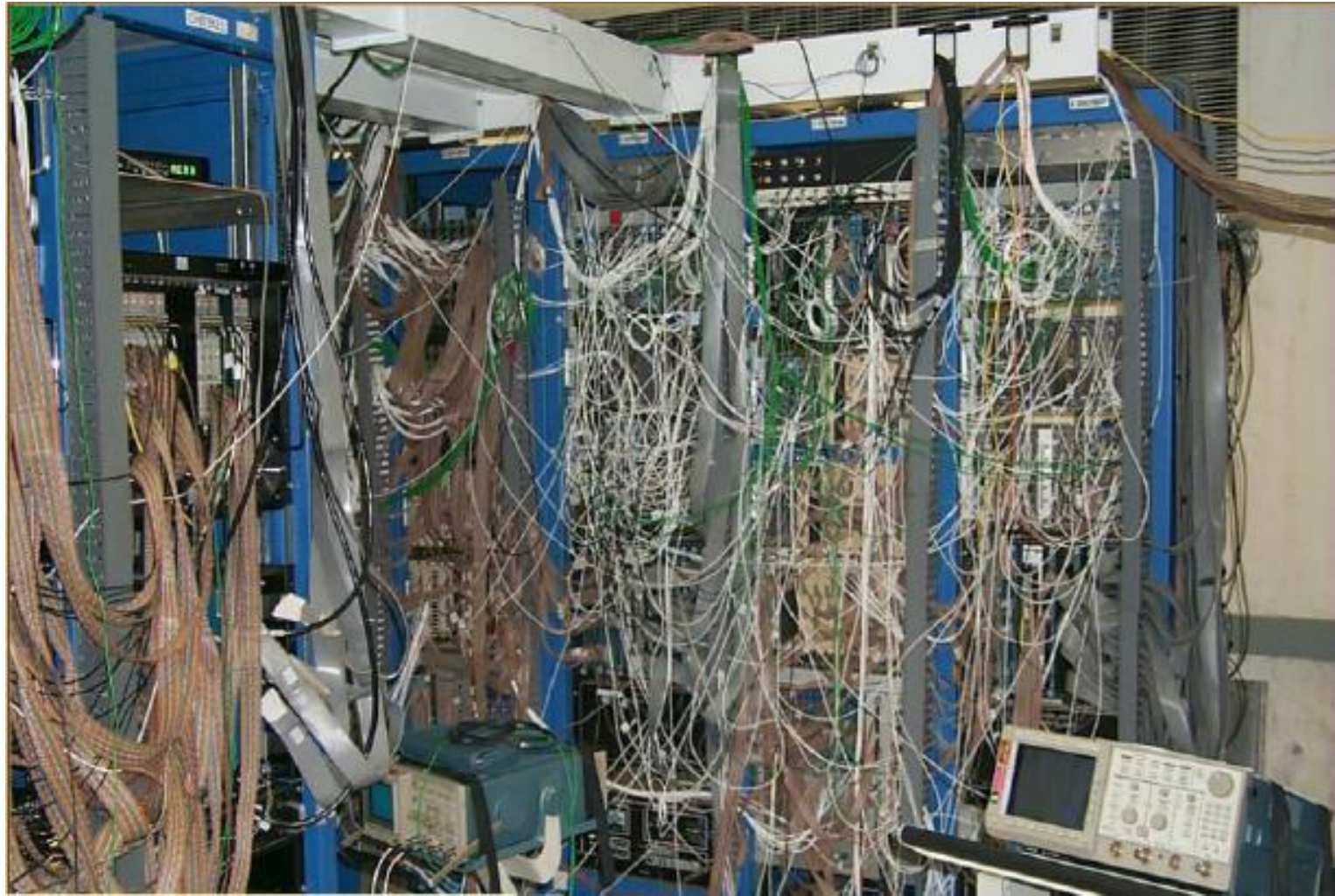
Select Line

Bus

- ☺ Relatively simple to implement
 - 🐾 Constant number of lines
 - 🐾 Each device implements the same interface
 - ➔ Easy to add new devices
- ☹ Scalability issues
 - 🐾 Number of devices and physical bus-length is limited
 - 🐾 Each new active device slows everybody down as bus bandwidth* shared among all the devices
 - 🐾 Maximum bus width is limited (128 bit for PC-system bus)
 - 🐾 Maximum bus frequency (number of elementary operations per second) is inversely proportional to the bus length
- 🐾 Typical buses have a lot of control, data and address lines (e.g. SCSI cable (Small Computer System Interface))
- 🐾 Buses are typically useful for systems $< 1 \text{ GB/s}$

Bandwidth = amount of data transferred / per unit of time (measured in Bytes/h)

Bus: another limitation



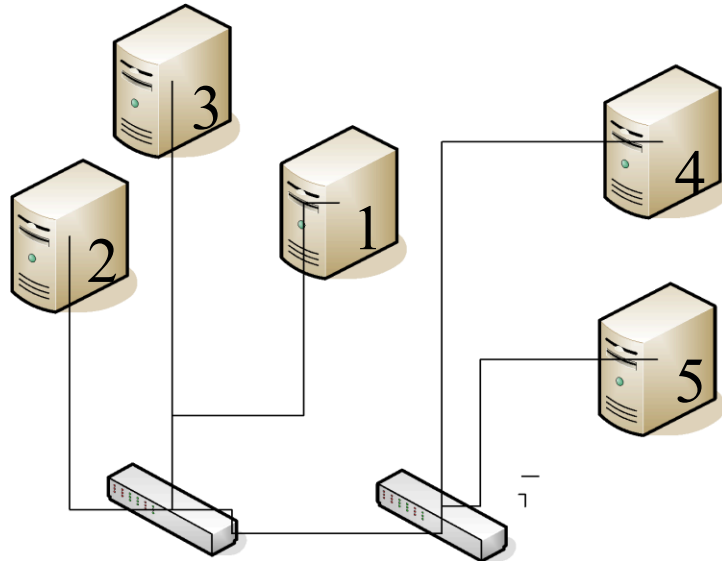
Network based DAQ

- ❖ In large (HEP) experiments we typically have thousands of devices to read, which are sometimes very far from each other
→ *buses can not do that*
- ❖ Network technology solves the scalability issues of buses
 - ❖ Examples: Ethernet, Telephone, Infiniband, ...
 - ❖ Devices are equal ("peers")
 - ❖ They communicate directly with each other by sending messages
 - ❖ no arbitration necessary
 - ❖ bandwidth guaranteed
 - ❖ Data and control use the same path
 - ❖ much fewer lines (e.g. in traditional Ethernet only two)
 - ❖ On an network a device is identified by a **network address**
 - ❖ eg: phone-number, MAC address
 - ❖ At the signaling level buses tend to use parallel copper lines.
Network technologies can be also optical or wire-less



Switched Networks

- Modern networks are *switched with point-to-point links*
- Each node is connected either to another node or to a **switch**
- Switches can be connected to other switches
- A path from one node to another leads through 1 or more switches
- Switches move messages between sources and destinations
 - Find the right path
 - Handle “congestion” (two messages with the same destination at the same time)



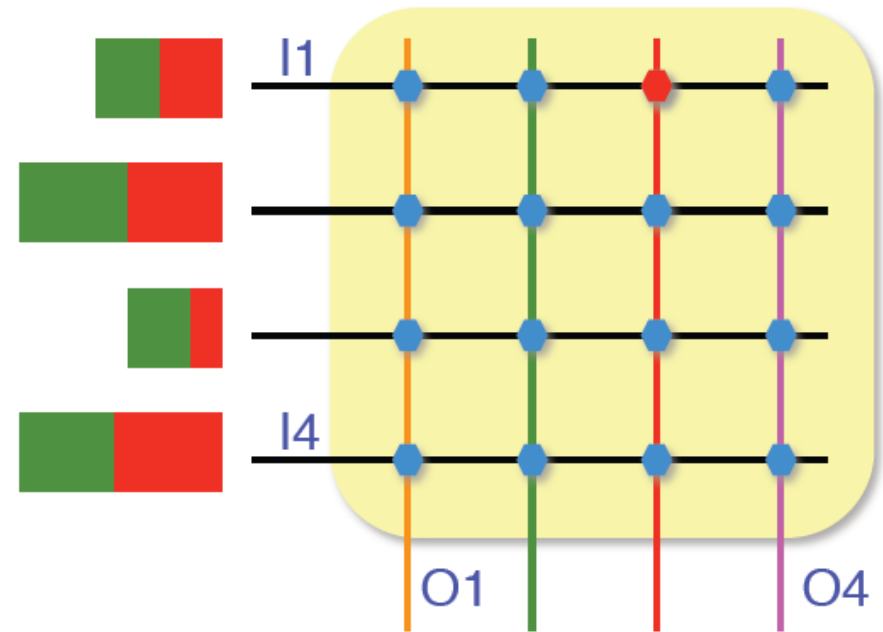
• Example

- While 2 can send data to 1 and 4, 3 can send at full speed to 5
- 2 can distribute the bandwidth between 1 and 4 as needed

Switched Network

Challenge

- Find the right path
- Handle “congestion” (two messages with the same destination at the same time)



DAQ challenges at LHC

Challenge 1

- Physics – Rejection power
- Requirements for TDAQ driven by rejection power required for the search of rare events

Challenge 2

- Accelerator – Bunch crossing frequency
- Highest luminosity needed for the production of rare events in wide mass range

Challenge 3

- Detector – Size and data volume
 - Unprecedented data volumes from huge and complex detectors

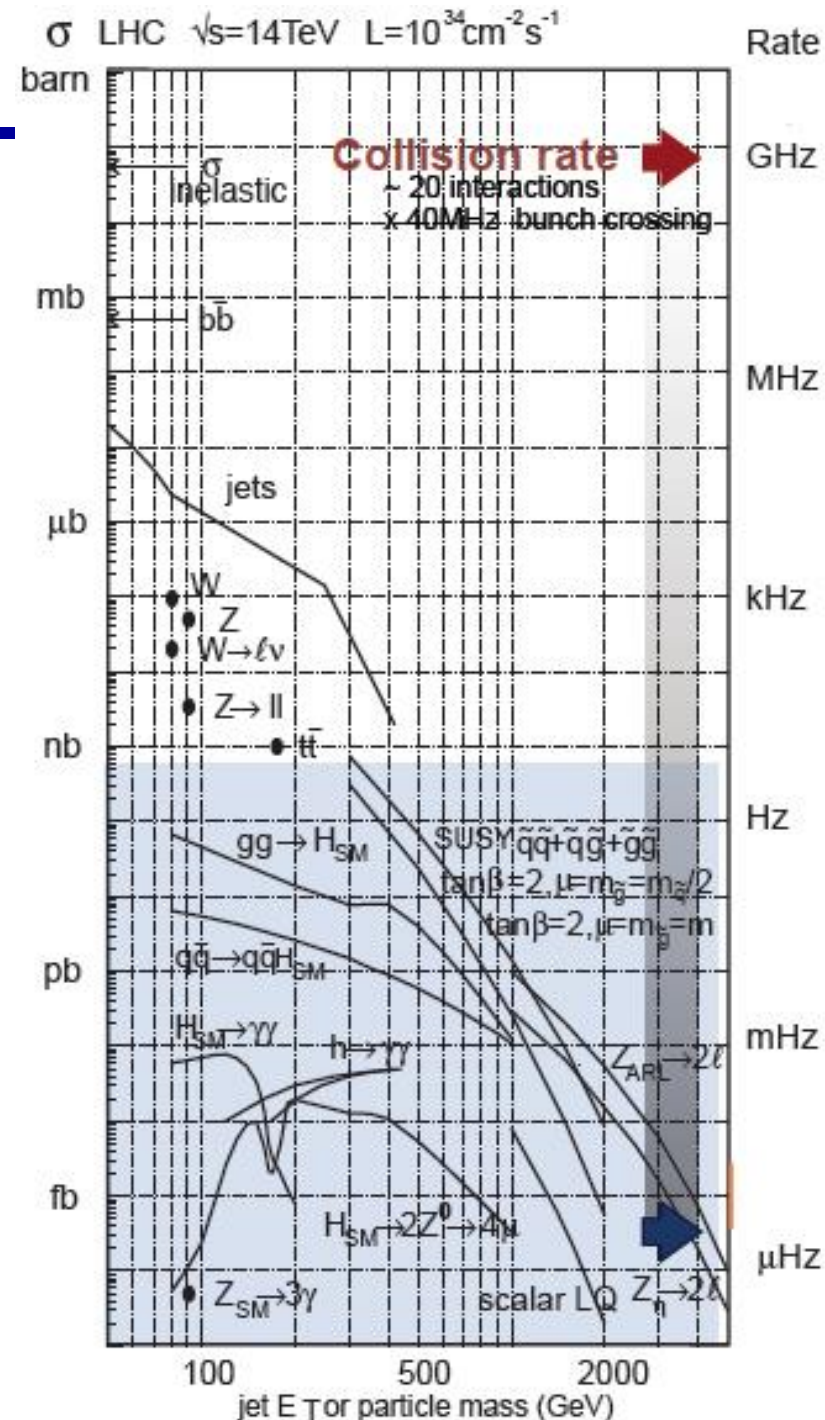


Challenge 1: Physics

- Cross sections for most processes at the LHC span ~ 10 orders of magnitude
- LHC is a factory for almost everything: t , b , W , Z ...
- But: some signatures have small branching ratios (e.g. $H \rightarrow \gamma\gamma$, BR $\sim 10^{-3}$)

| Process | Production Rate $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ |
|-------------------------|---|
| inelastic | $\sim 1 \text{ GHz}$ |
| $b\bar{b}$ | 5 MHz |
| $W \rightarrow \ell\nu$ | 150 Hz |
| $Z \rightarrow \ell\nu$ | 15 Hz |
| $t\bar{t}$ | 10 Hz |
| Z' | 0.5 Hz |
| $H(120) \text{ SM}$ | 0.4 Hz |

- $L=10^{34} \text{ cm}^{-2}\text{s}^{-1}$: Collision rate: $\sim 10^9 \text{ Hz}$.
event selection: $\sim 1/10^{13}$ or 10^{-4} Hz !



Challenge 1: Physics

- ❁ Requirements for TDAQ driven by the search for rare events within the overwhelming amount of “uninteresting” collisions

- ❁ Main physics aim

- ❁ Measure Higgs properties

- ❁ Searches for new particles beyond the Standard Model

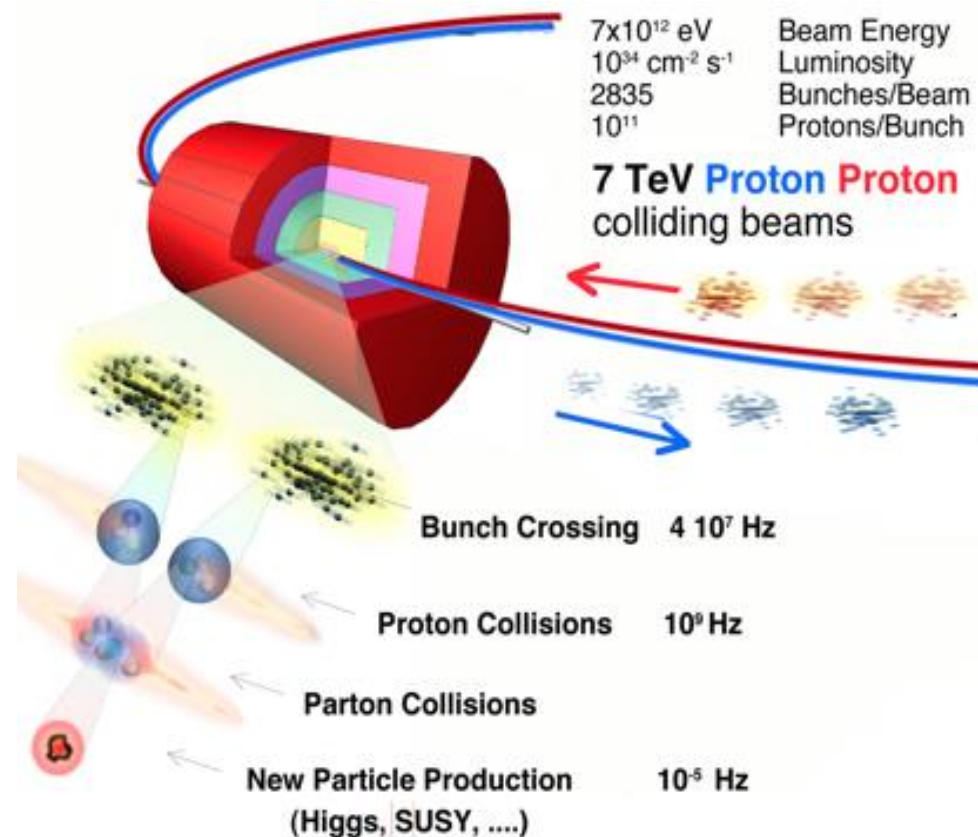
- ❁ Susy, extra-dimensions, new gauge bosons, black holes etc.

- ❁ Plus many interesting Standard Model studies to be done

- ❁ All of this must fit in ~300-1000 Hz of data written out to storage

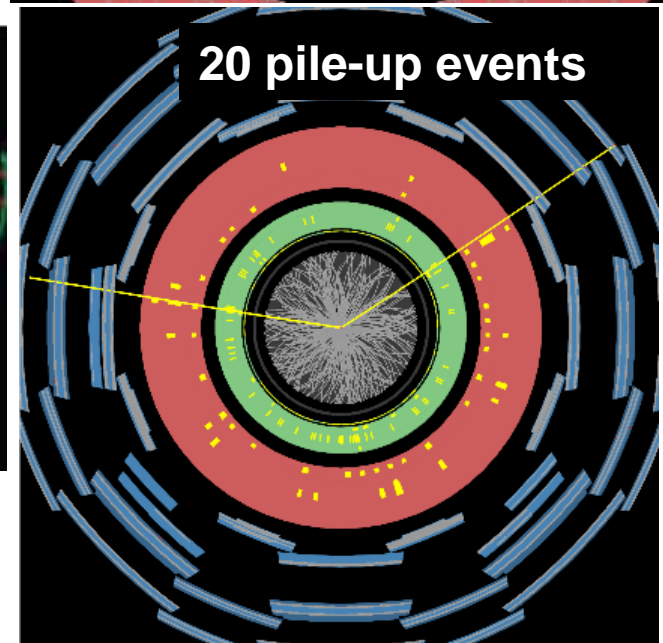
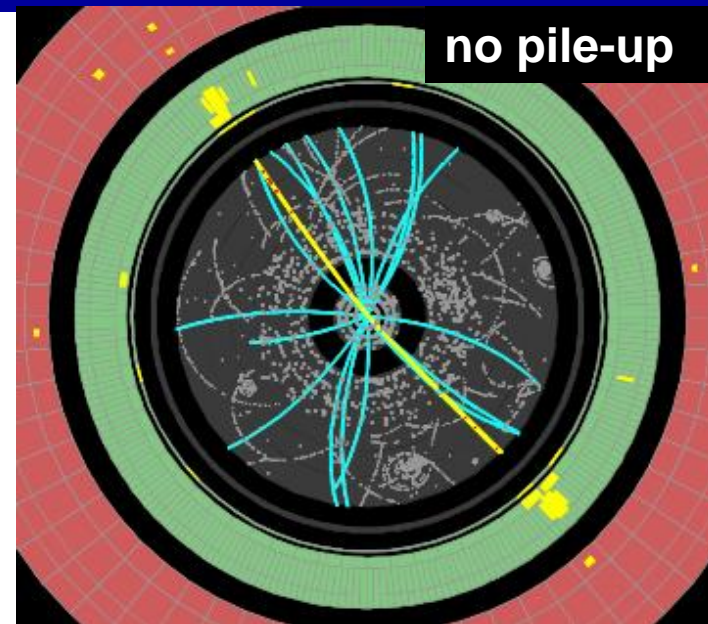
- ❁ Not as trivial, $W \rightarrow l\nu$: 150 Hz

- 👉 “Good” physics can become your enemy!



Challenge 2: Accelerator

- Unlike e^+e^- colliders, proton colliders are more 'messy' due to proton remnants
- In 2012 LHC already produced up to 30 overlapping p-p interactions on top of each collision (pile-up) \rightarrow >1000 particles seen in the detector!



Challenge 3: Detector

- Besides being huge: number of channels are $O(10^6-10^8)$ at LHC, event sizes ~ 1.5 MB for pp collisions, 50 MB for pb-pb collisions in Alice
 - need huge number of connections
- Some detectors need > 25 ns to readout their channels and integrate more than one bunch crossing's worth of information (e.g. ATLAS LArg readout takes ~ 400 ns)
- It's On-Line (cannot go back and recover events)
 - need to monitor selection - need very good control over all conditions



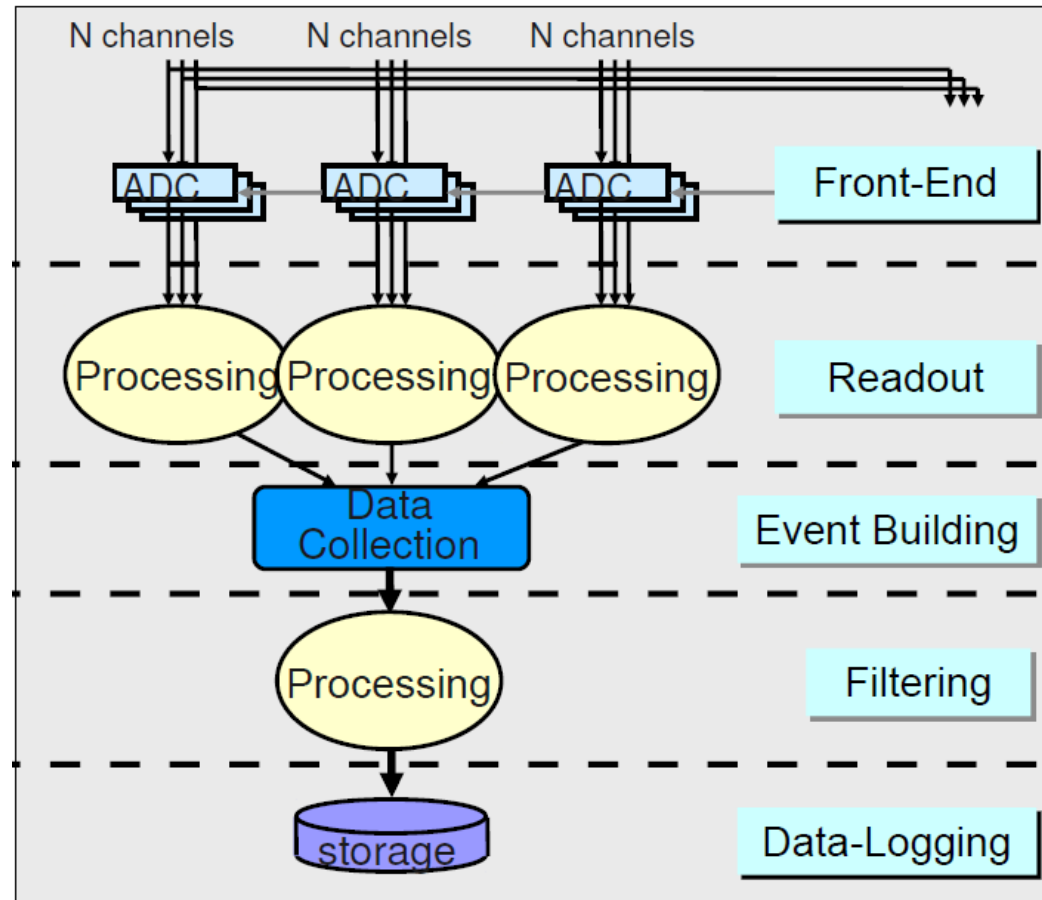
Let's build a Trigger and DAQ for this

- What do we need?

Let's build a Trigger and DAQ for this

What do we need?

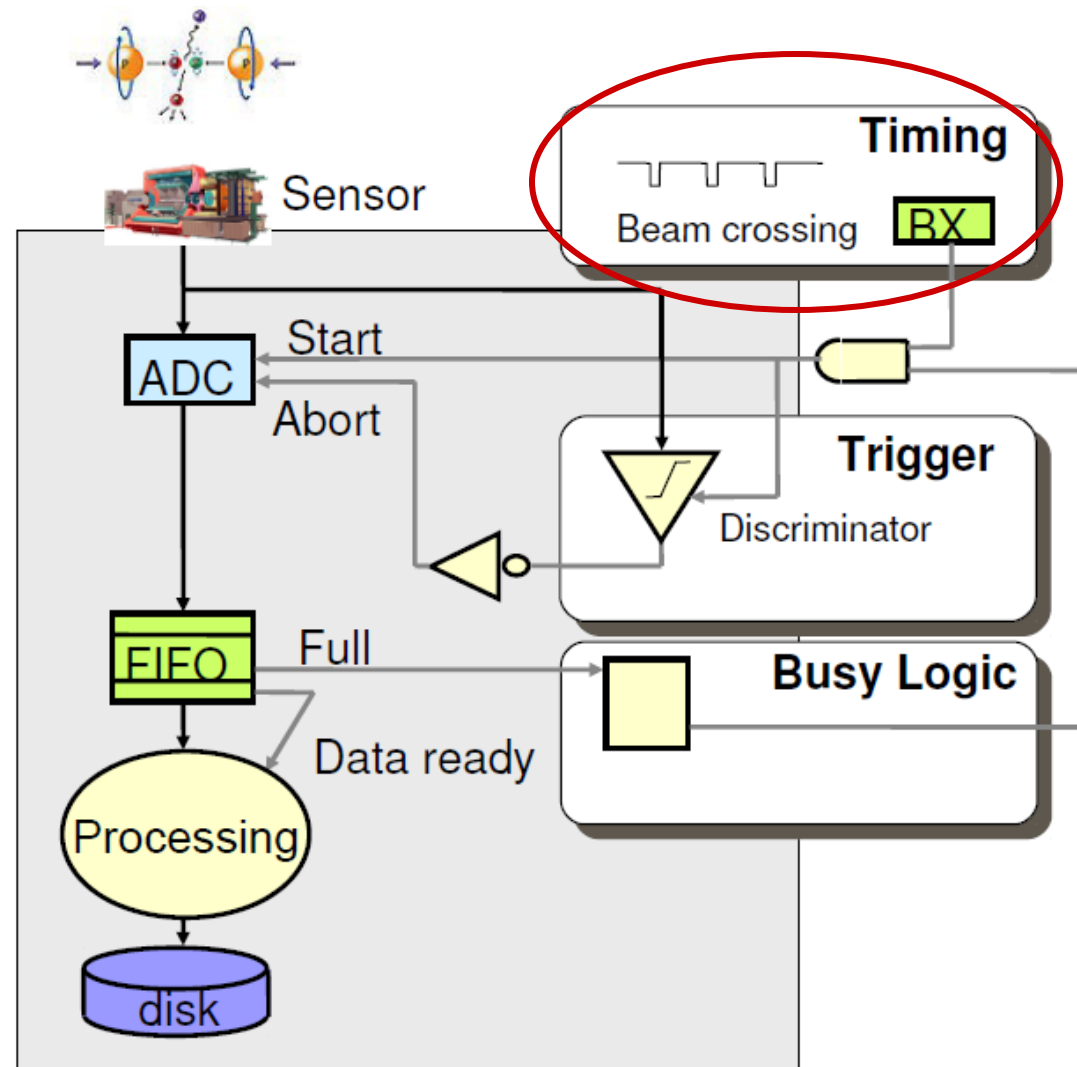
- Electronic readout of the sensors of the detectors (“front-end electronics”)
- A system to collect the selected data (“DAQ”)



Let's build a Trigger and DAQ for this

What do we need?

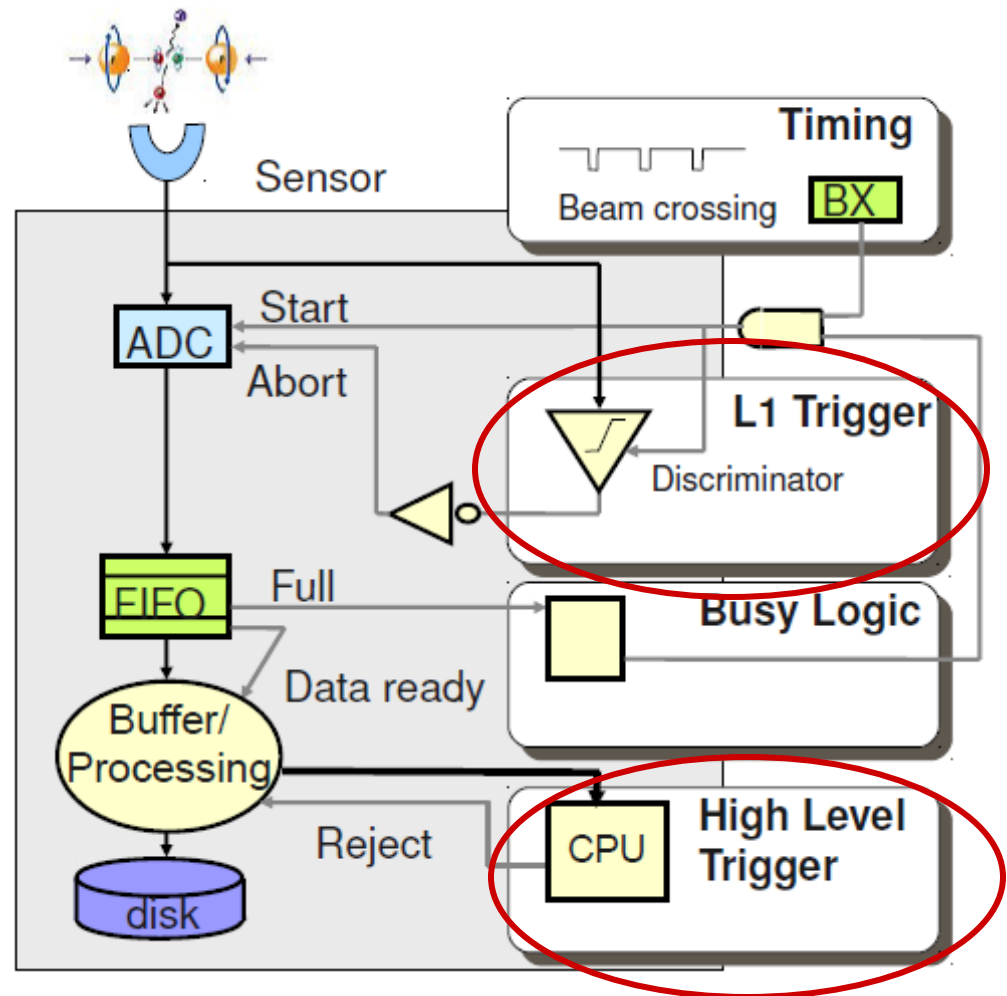
- Electronic readout of the sensors of the detectors (“front-end electronics”)
- A system to collect the selected data (“DAQ”)
- A system to keep all those things in sync (“clock”)



Let's build a Trigger and DAQ for this

What do we need?

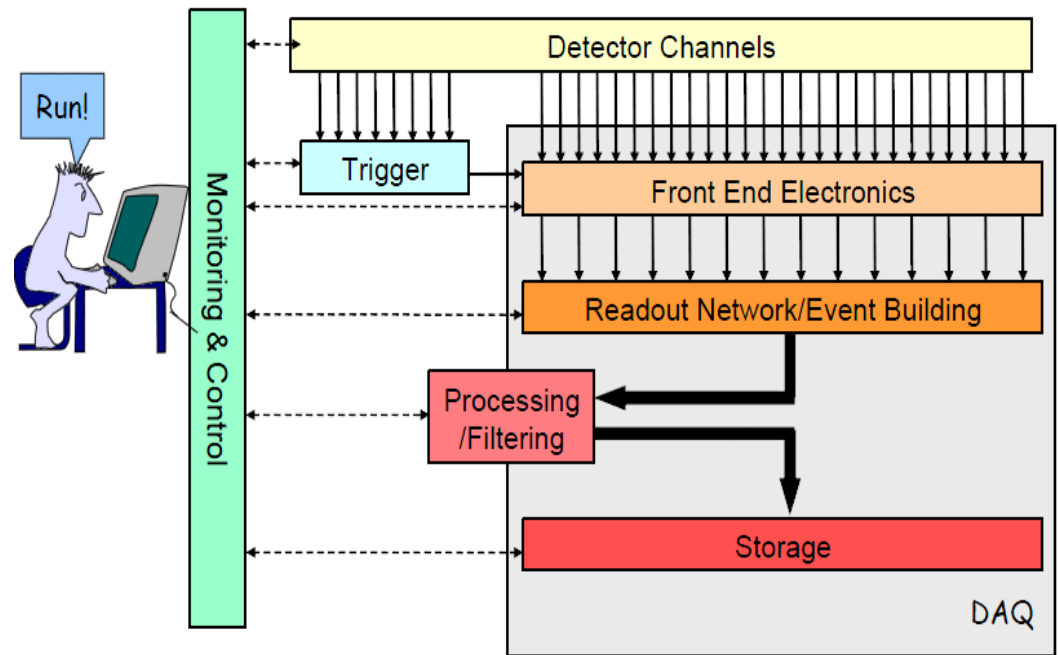
- Electronic readout of the sensors of the detectors (“front-end electronics”)
- A system to collect the selected data (“DAQ”)
- A system to keep all those things in sync (“clock”)
- A trigger – multi-level due to complexity



Let's build a Trigger and DAQ for this

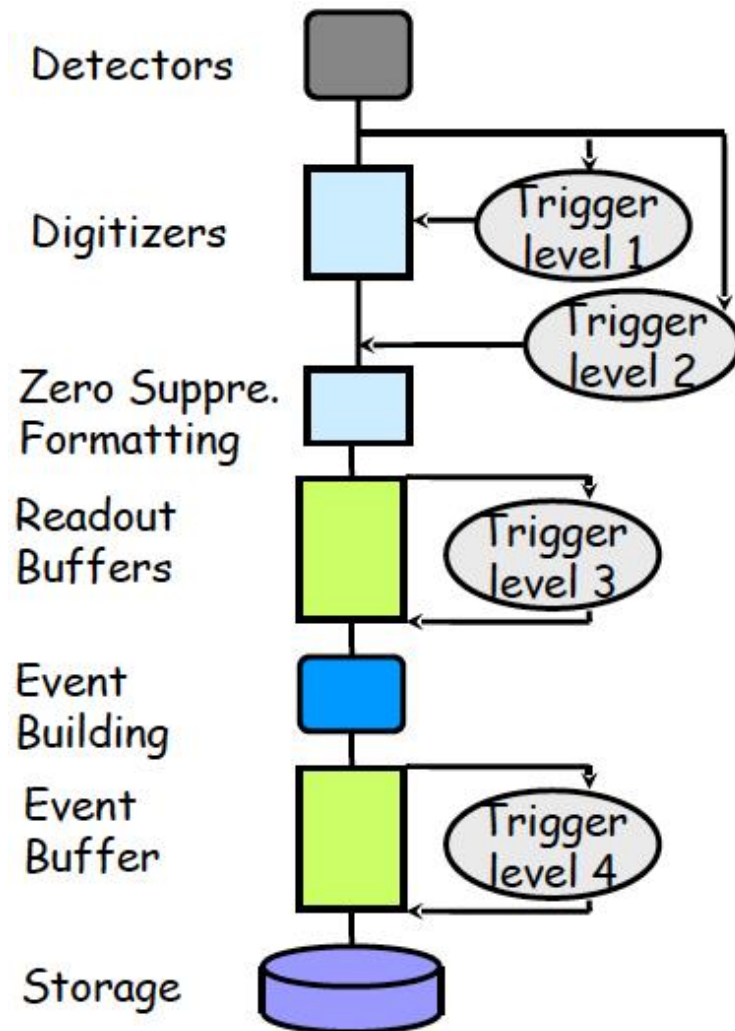
What do we need?

- ❗ Electronic readout of the sensors of the detectors (“front-end electronics”)
- ❗ A system to collect the selected data (“DAQ”)
- ❗ A system to keep all those things in sync (“clock”)
- ❗ A trigger – multi-level due to complexity
- ❗ A Control System to configure, control and monitor the entire DAQ

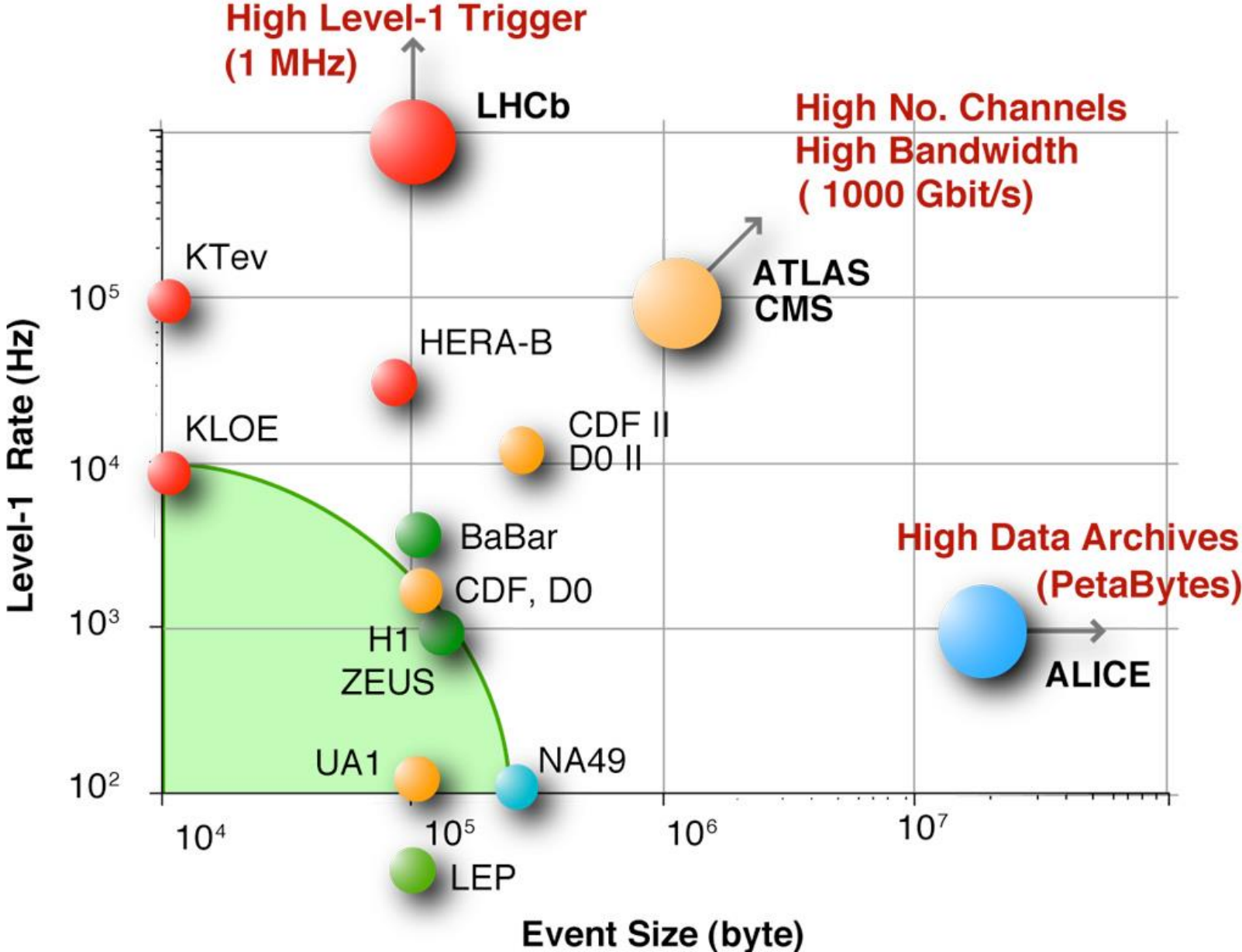


Multi-level trigger system

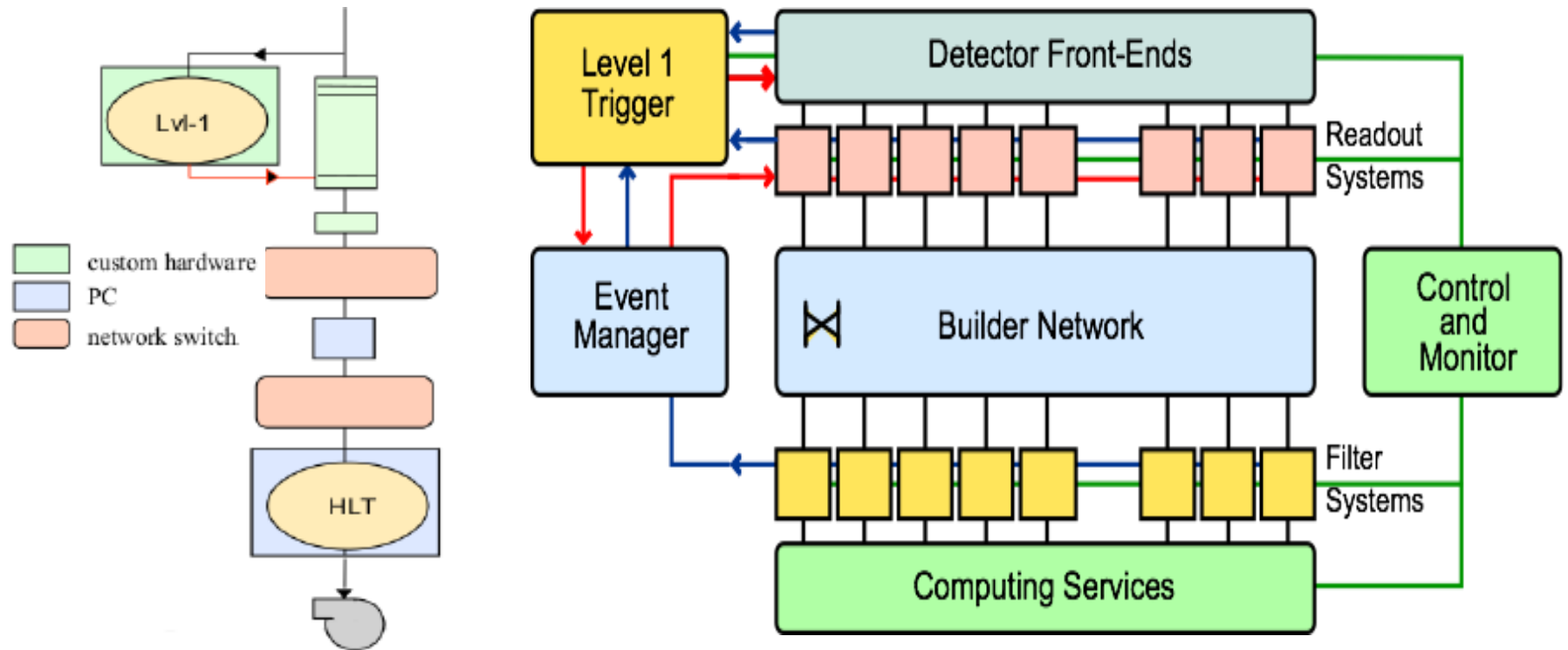
- ❦ Sometime impossible to take a proper decision in a single place
 - ❦ too long decision time
 - ❦ too far
 - ❦ too many inputs
- ❦ Distribute the decision burden in a hierarchical structure
 - ❦ Usually $T_{N+1} \gg T_N$, $f_{N+1} \ll f_N$
- ❦ At the DAQ level, proper buffering must be provided for every trigger level
 - ❦ absorb latency
 - ❦ De-randomize



LHC DAQ phase-space



The CMS Trigger/DAQ System (Run 1)



- Overall Trigger & DAQ architecture: 2 trigger levels

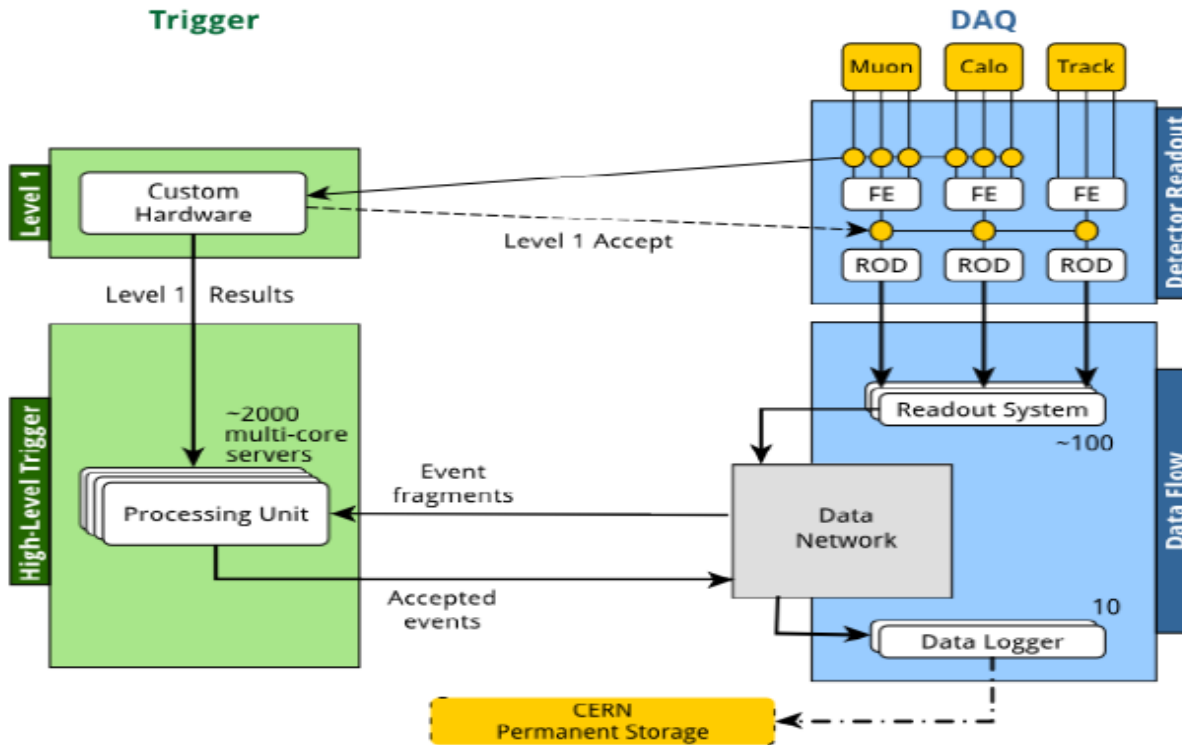
- Level-1:

- 3.2 μ s latency
- 100 kHz output

- DAQ/HLT

- Event building at full L1 rate
- Average event size 1 MB
- Average CPU time 50 ms
- Average output rate in 2012: 350 Hz prompt, 300Hz “parked”

The ATLAS Trigger/DAQ System (Run 1)



- Overall Trigger & DAQ architecture: 3 trigger levels

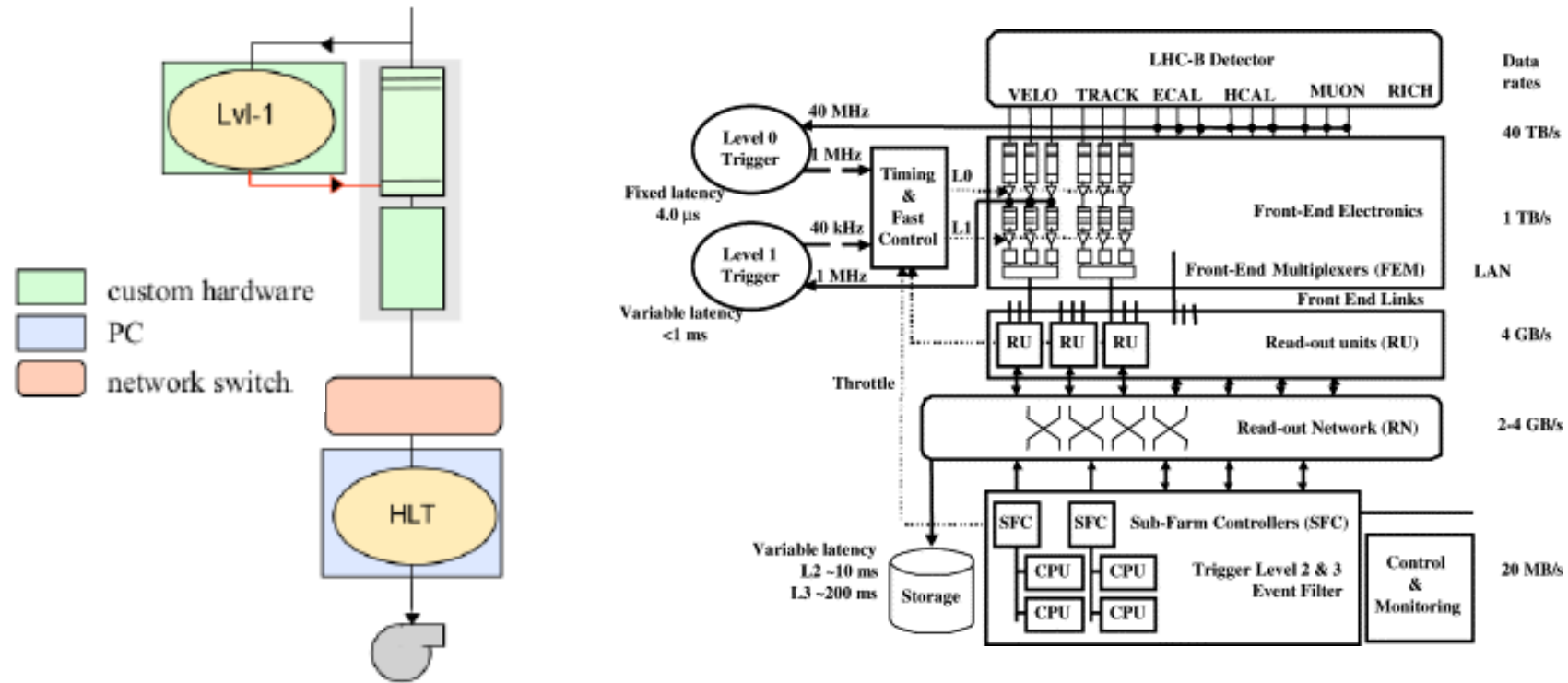
- Level-1:

- 2.5 μ s latency
- 75 kHz output

- DAQ/HLT

- Analyse regions around particles identified at L1
 - Reduce rate to 5.5. kHz
- Average output rate in 2012: 400 Hz prompt, 200 Hz “parked”
- Average event size 1.5 MB

The LHCb Trigger/DAQ System (Run 1)



❖ Overall Trigger & DAQ architecture: 3 trigger levels

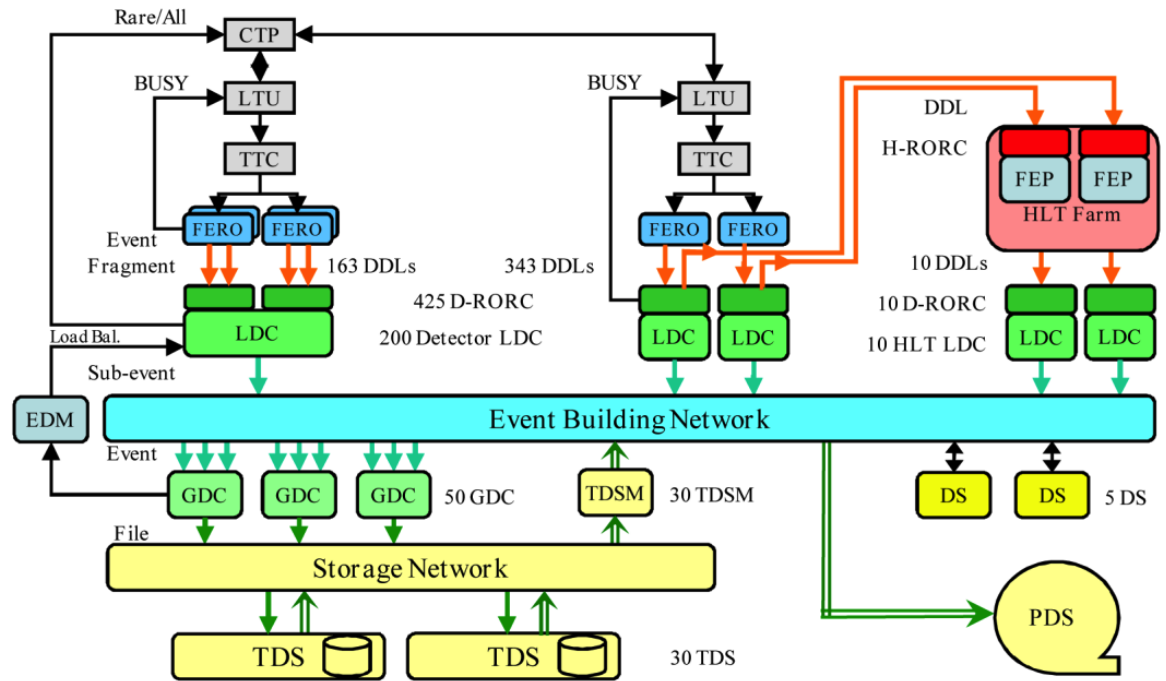
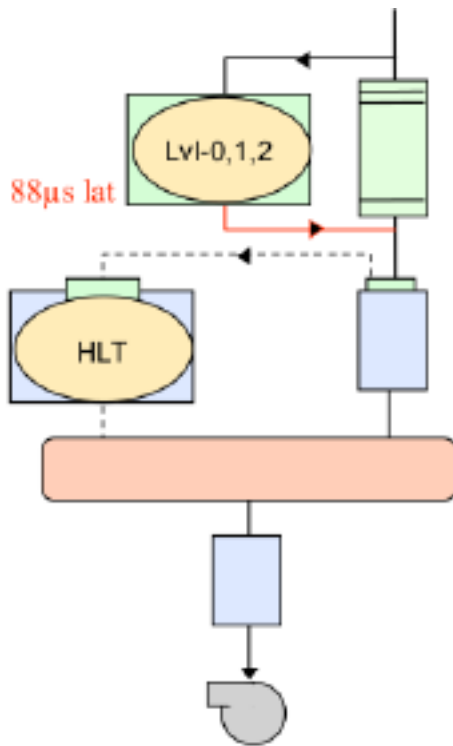
❖ Level-0:

- ❖ 4 μ s latency
- ❖ 1 MHz output

❖ DAQ/HLT

- ❖ L1: look displaced high p_T tracks, output 70 kHz
- ❖ L2: full event reconstruction
- ❖ Average output rate in 2012: 5 kHz
- ❖ Average event size 35 kB

The ALICE Trigger/DAQ System (Run 1)



❖ Alice has different constraints

- ❖ Low rate: max 8 kHz pb+pb
- ❖ Very large events: > 40MB
- ❖ Slow detector (TPC ~ 100 μ s)

❖ Overall Trigger & DAQ architecture: 4 trigger levels

❖ 3 hardware-based trigger, 1 software-based:

- ❖ L0 – L2: 1.2, 6.5, 88 μ s latency
- ❖ L3: further rejection and data compression

Summary

- The principle of a simple data acquisition system
- Introduction to some basic elements: trigger, derandomiser, FIFO, busy logic
- How data is transported
 - Bus versus network
- Challenge to design efficient trigger/DAQ for LHC
 - Very large collision rates (up to 40 MHz)
 - Very large data volumes (tens of MBytes per collision)
 - Very large rejection factors needed ($>10^5$)
- Showed data acquisition used in LHC experiments
 - Now everyone has upgraded their infrastructure for 2015