
DAQ and Trigger Beyond HL-LHC

Emilio Meschi
CERN/EP

“ECFA Task Force 7”
Roadmap Symposium

Disclaimer

Will give a very partial (and probably biased) point of view

Not an expert in many of the subjects touched

Apologies to those who know more already

Mildly inspired by questionnaire and answers
(mistakes are all mine)

Various material stolen from different sources

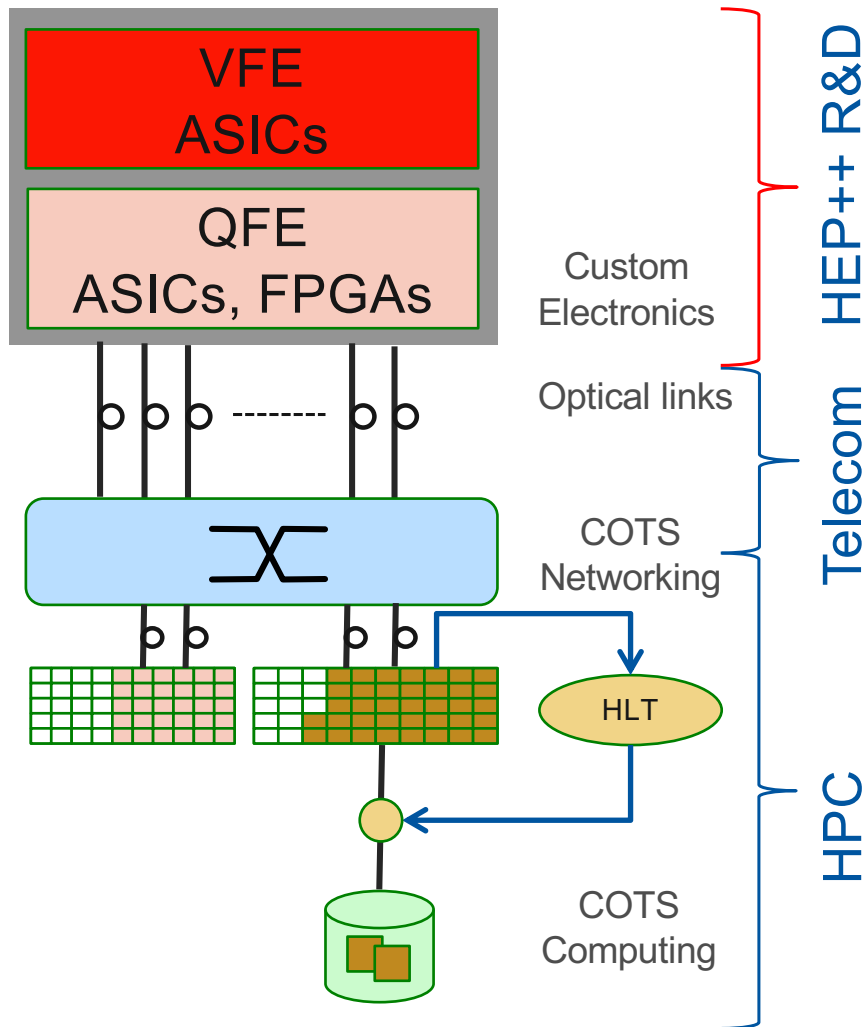
Should be

~~DAQ and Trigger~~ Beyond HL-LHC ?

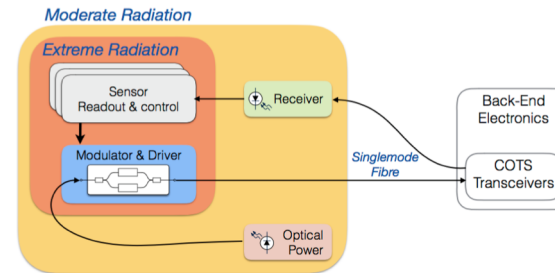
Emilio Meschi
CERN/EP

“ECFA Task Force 7”
Roadmap Symposium

DAQ: nice and easy



VFE does the analog part, ADC, low-level calibration, zero suppression, lossless compression, *optical links*



low-power, rad-hard (*rad-tolerant*)

QFE does medium scale aggregation, local reconstruction, "lossy" compression, transition to standard protocol on optical links

asynchronous – precision clock - timestamping

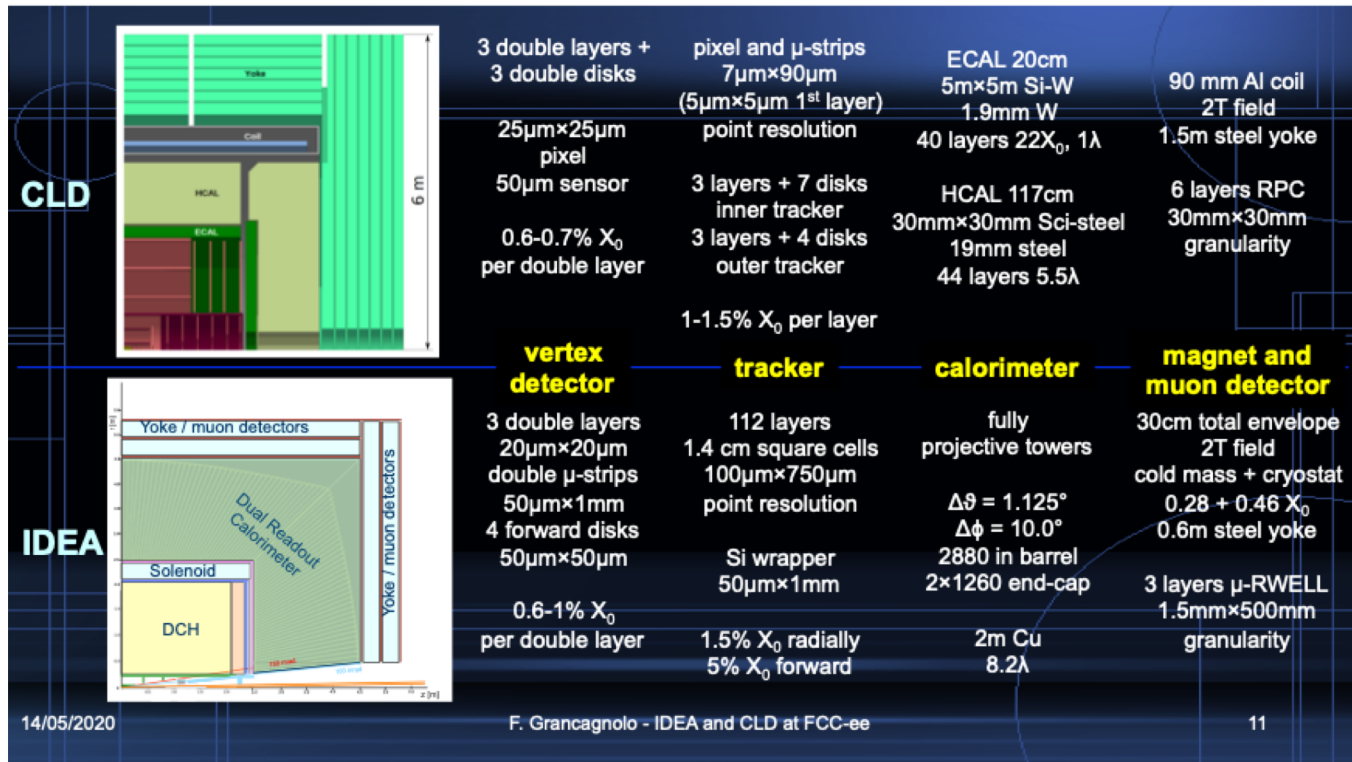
COTS switched networks provide further aggregation, up to and including event building

COTS servers with co-processors do the final selection

One-size-fits-all type of problem

Some “real”-life problems

Beyond HL-LHC



3.7 ns interbunch at Z pole
 Rates are high but events small

Tracker with many channels but occupancy low (~kB??)

Trigger not challenging, but precision measurements benefit from multiple strategies

Trigger-less data rates

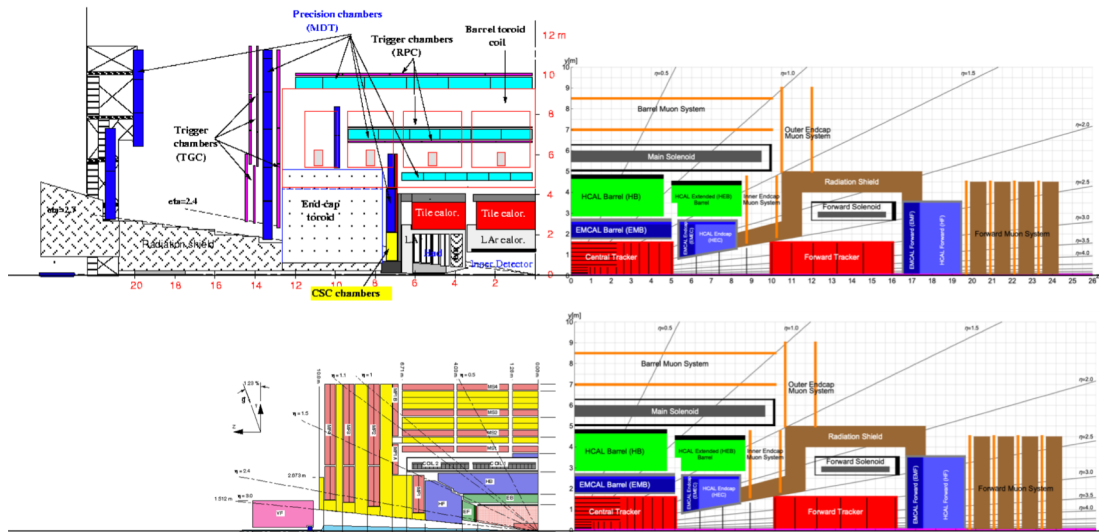
similar to HL-LHC

Stepping stone for the hh detectors

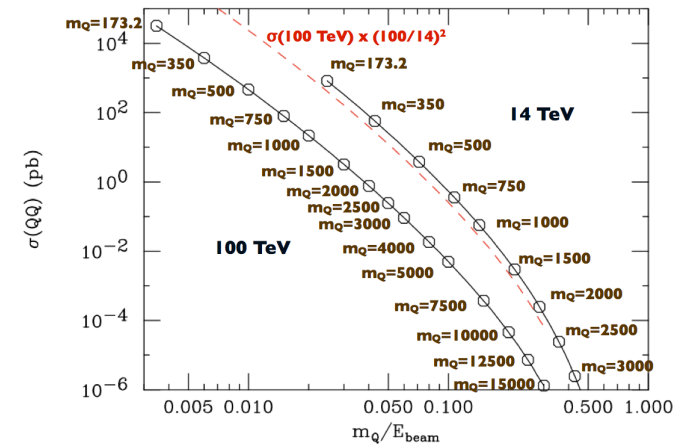
Beyond HL-LHC

- Higher energy
 - larger B field, solenoid bore radius -> tracker
- 400 m² silicon, 10¹⁰ channels
- Higher to extreme fluences, less accessibility
- Luminosity: always as high as possible
 - Shorter interbunch to reduce PU ?

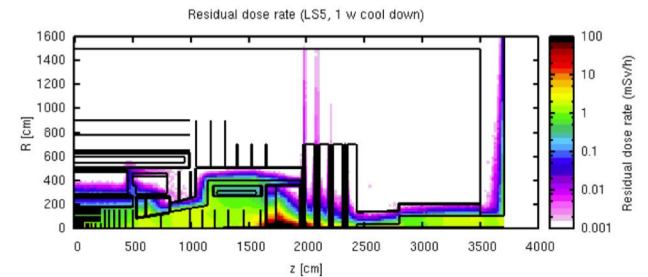
Comparison to ATLAS & CMS



https://indico.cern.ch/event/727555/contributions/3461232/attachments/1869213/3075082/fcc_hh_detector_brussels_june_2019_riegler.pdf



31 GHz of pp collisions
 Pile-up 1000
 4 THz of tracks



Un-triggered readout at 40MHz **2000 - 3000TByte/s** over optical links to the underground service cavern and/or HLT

Not Just Colliders

- Neutrino@accelerator (DuNE...)
- Dark matter at BD or LL
- Next-generation specialized experiments and FT
- Neutrino (IceCube-2..)
- Astroparticle (CTA...)
- Radioastronomy (SKA...)
- GW (ET, CE...)

Multi-messenger astronomy: network
all the above in real time

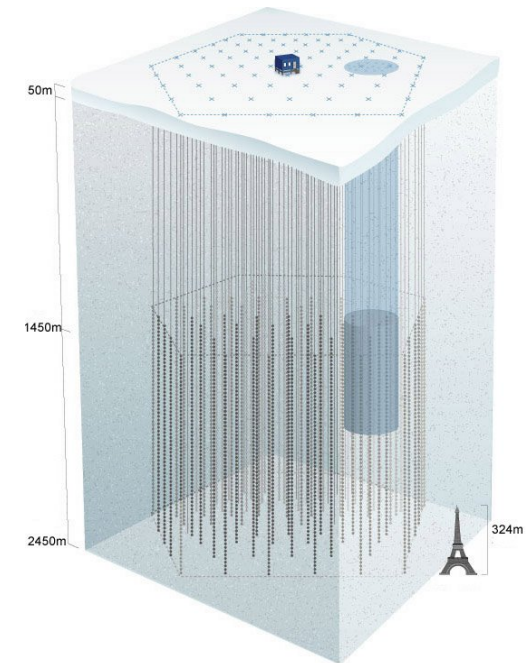
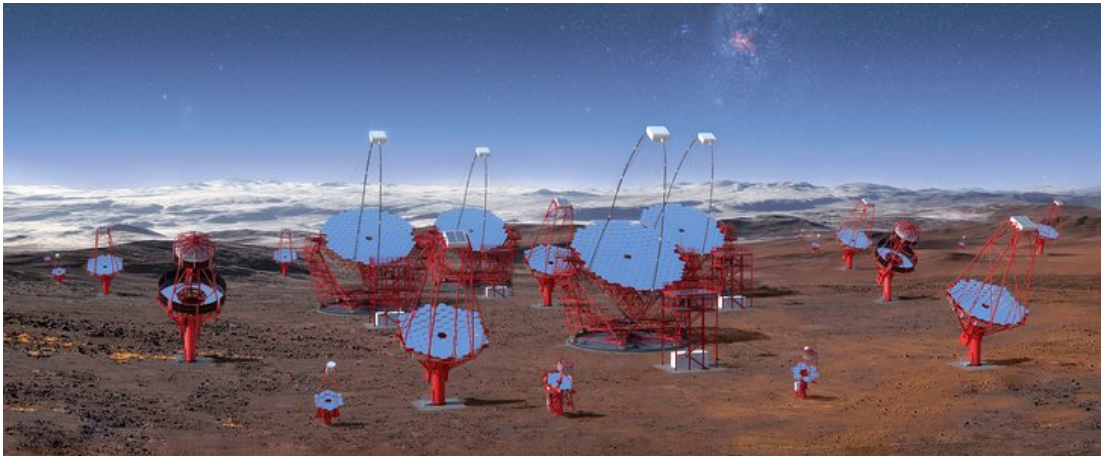
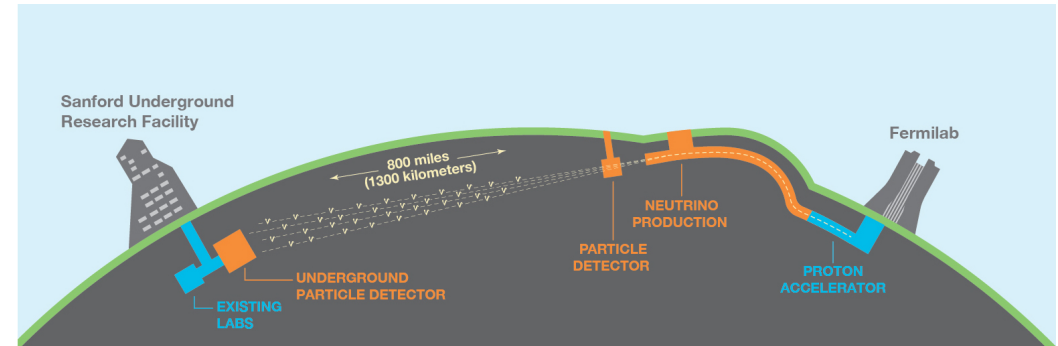
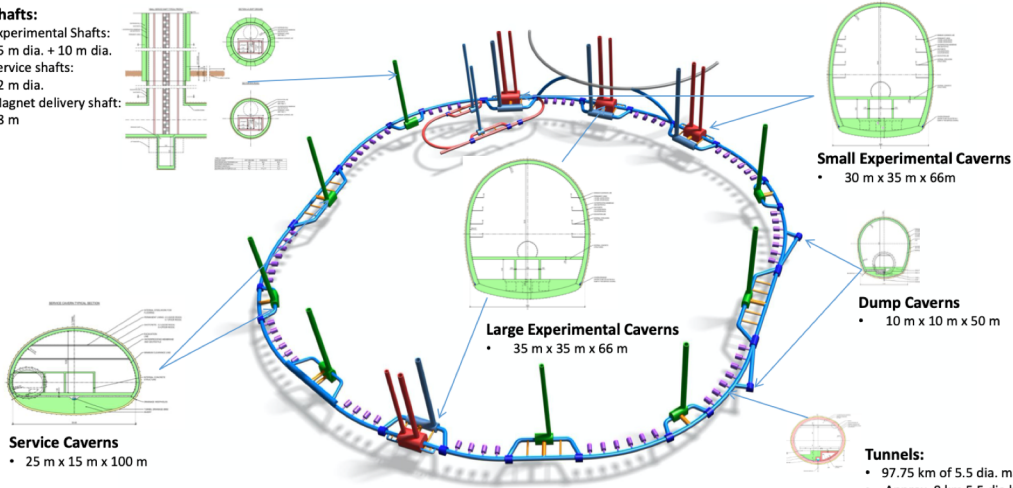
Data reduction
Synchronization
Data rates
“Image” processing
AI applications
(CNN, GN, AE?)
Reliability

Precision clock
distribution
Synchronization
High-bandwidth long-
distance links
Reliability

Common needs for reliable systems

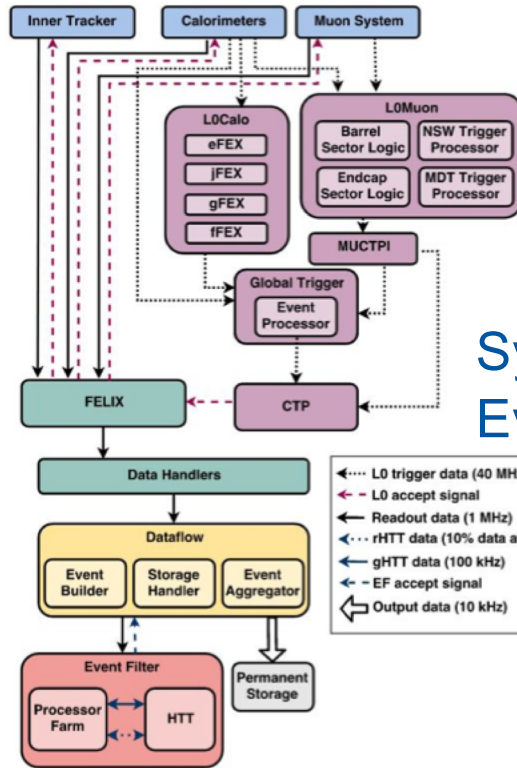
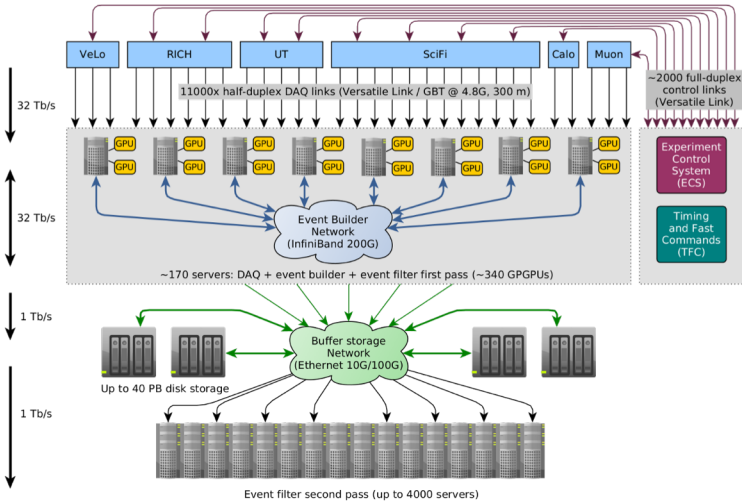
Shafts:

- Experimental Shafts: 15 m dia. + 10 m dia.
- Service shafts: 12 m dia.
- Magnet delivery shaft: 18 m



Reality...

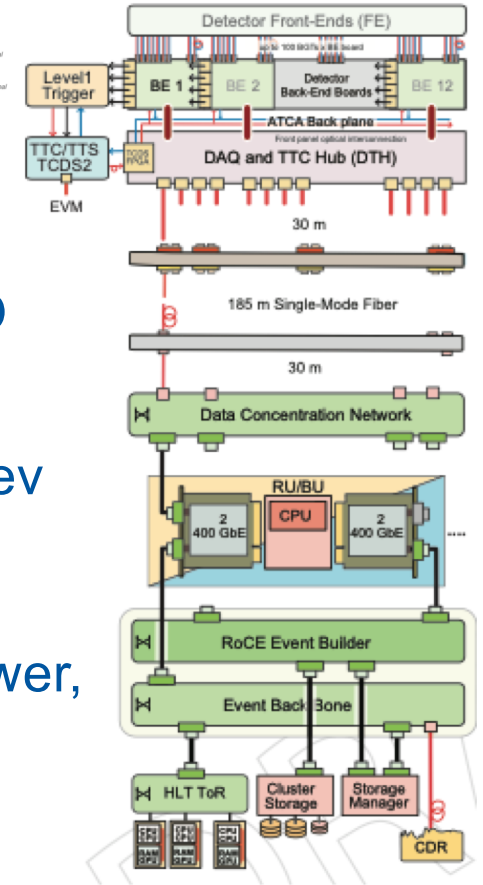
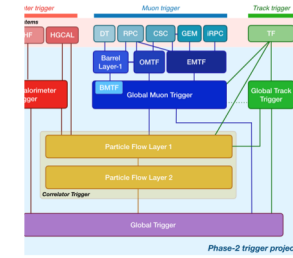
...hits



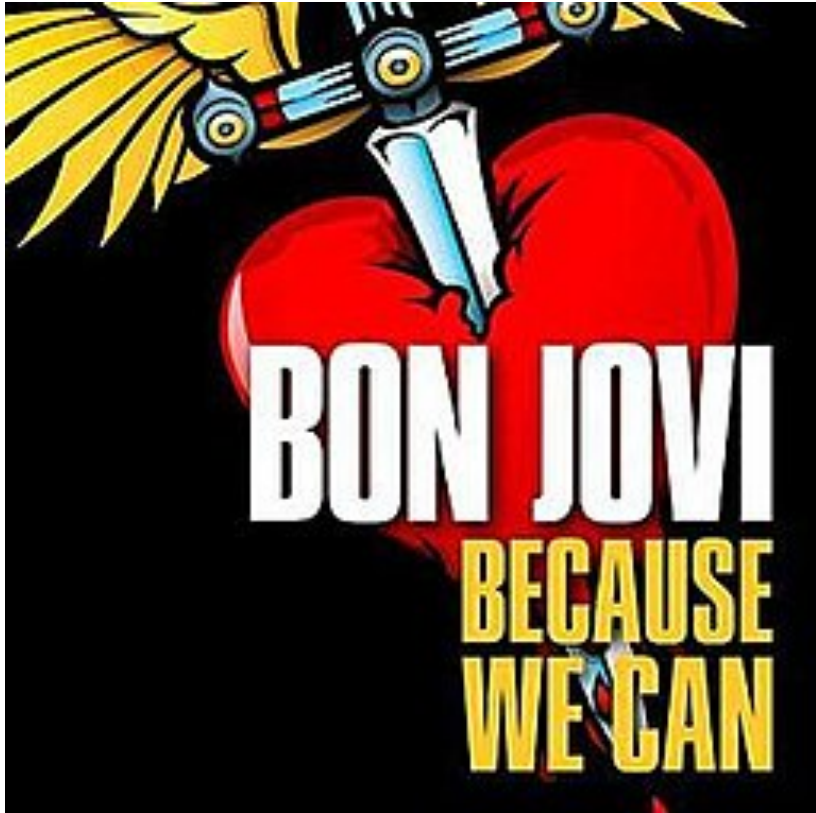
- ⋯ L0 trigger data (40 MHz)
- ← L0 accept signal
- ← Readout data (1 MHz)
- ⋯ rHTT data (10% data at 1 MHz)
- ← gHTT data (100 kHz)
- ← EF accept signal
- ↔ Output data (10 kHz)

Synchronous R/O Event builder

Common dev optical links from FE (low-power, rad-hard)



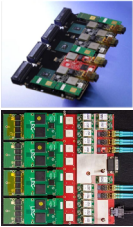
Commonalities in R/O approach A.A.L.
Trigger still there (no hw in L.)



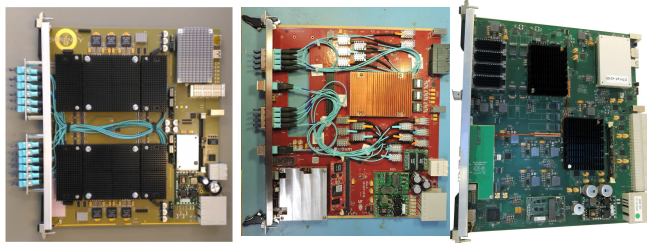
One example:

When and how we go to COTS networking

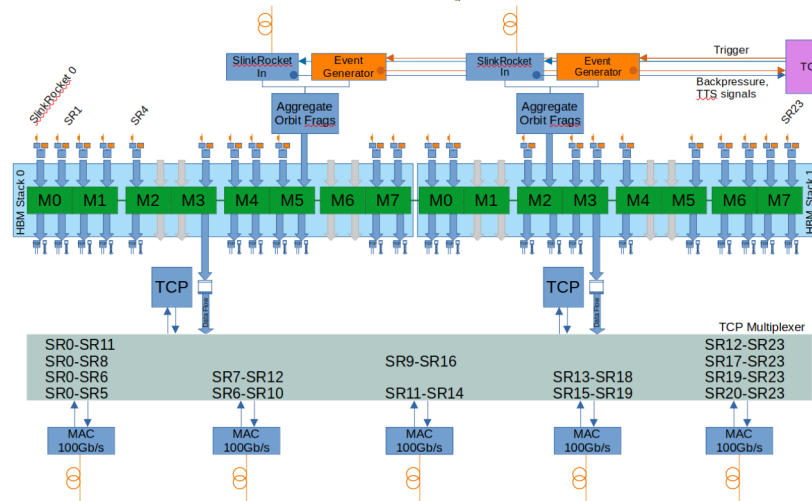
When reality hits



Commercial FPGAs “on”
detector (mild radiation)
“commercial” server as crate



Multiple (DAQ and trigger)
BE boards
ATCA crate
Reliable protocol to EVB



Orbit aggregation

HBM

Simplified TCP/IP

CWDM4 100G links
to surface

- Single BE platform ideal
- Fw “customization” is challenging
- No trigger BE board flavors helps
- Obsolescence of server platform a **risk factor** (see PCI-X to PCIe)
- More **computers in service cavern** (special racks, cooling)

- HBM enables reliable protocol
- Essentially **equivalent to PCIe (with long range)**
- Orbit aggregation ~ mild timestamping (enables optimal working point for link aggregation, evb, multi-BX triggers)
- Future: move whole data aggregation/TCP engine to BE ?

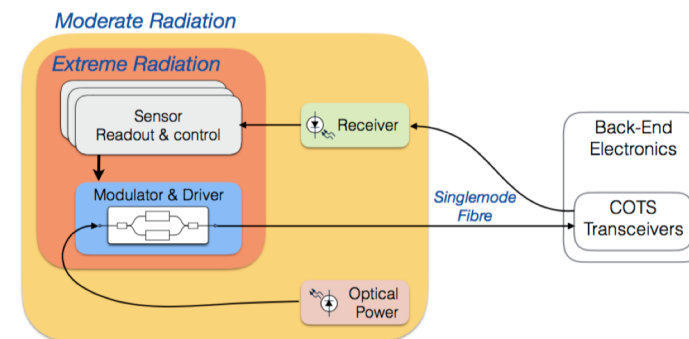
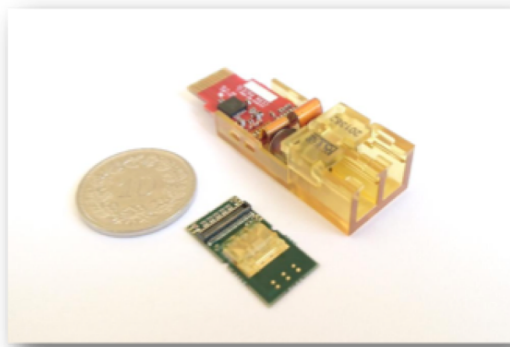
Optimistic outlook / lessons learned

- Move “as early as possible” to COTS networking
 - Requires buffer at sender (PC memory or something else)
 - Bare Ethernet not lossless unless sufficient buffer at switch (must keep usage below ~50%) - residual losses as deadtime (accounting) – need reliable delivery
- Could move “EVB” functionality to BE
 - Reliable protocol using HBM for congestion, aggregation
 - Integrated in back-end
 - Complete fw implementation of e.g. TCP
 - Delegate final EVB to HLT nodes (mild timestamping)
- **Economic and sociological reasons (may) make this hard**

The other side

First stop: Readout at FE

- Optical Data Transmission technology is key
 - High Bandwidth **low mass, low power**
 - Immune to **electromagnetic interference** (+ isolation between power and readout)
 - Sufficiently **radiation tolerant**



- Optical layer is only part of the story
 - **Acquisition, aggregation, and serialisation before** transmission over the link
- **ASICs** need to be specified and designed to meet system requirements
 - Increasingly complex over generations
 - Common developments are key
 - Reuse (of design, concept)

synchronous or asynchronous

- Using COTS at FE == profit of high bandwidth “for free”
 - Fixed speed and narrow locking range
- Asynchronous readout (with time stamping...)
 - less complex back-end electronics
 - In principle can use “standard” protocol (e.g. Ethernet) and connect “quasi-FE” to “commercial” equipment
 - Challenging clock distribution, disciplined clock, phase stability at FE
 - Additional bandwidth
- Synchronous
 - RF stability (ramps) and distribution vs. locking range
 - COTS not compatible with specific RF frequencies
 - need deterministic behavior (and if not, can losses be afforded/quantified)
 - Deterministic protocols hard
 - Reliable protocols require lots of buffer at the sender (and are not deterministic by construction)

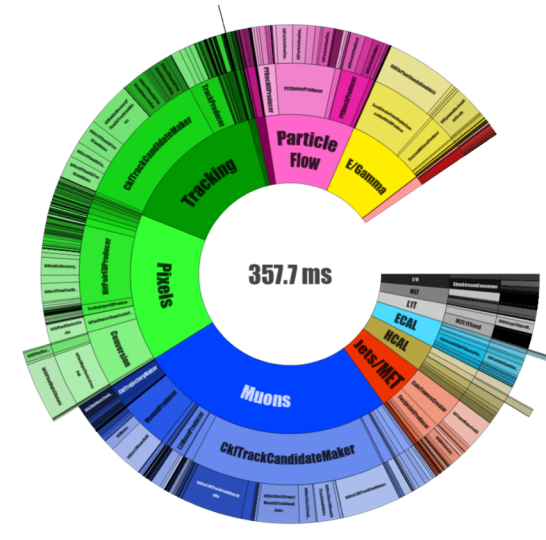
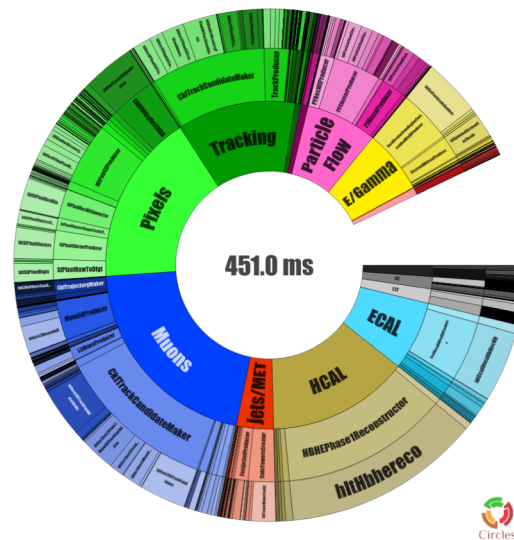
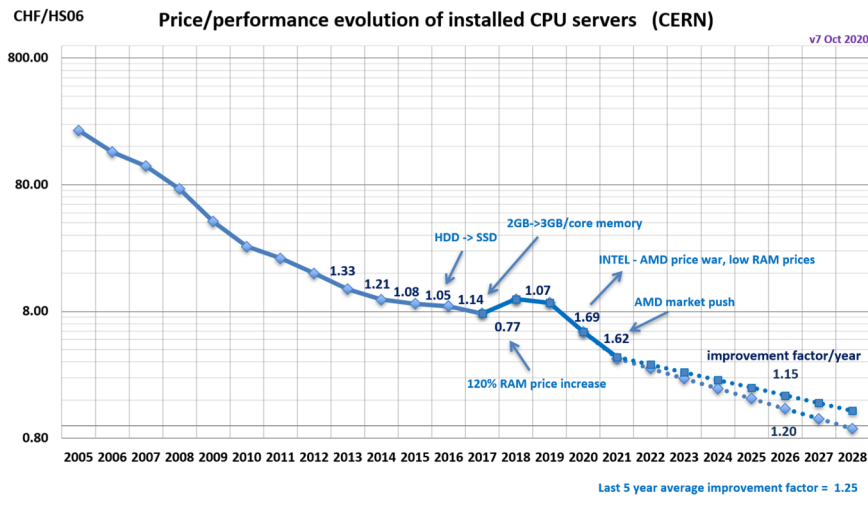
“trigger-less” is good ?

Trigger ::= read out everything (or “just without” L1 trigger) ?

- + Reduce relevance of custom processors and special data paths
- More **high speed links at or near the front end** (with the usual problems...)
 - material budget (for power, cooling, fibres)
 - rad hardness (of optical transmitters, fibres)
 - Will get worse at “future” colliders
 - Optical on FE needs a lot of technology for which we are the only customers
- Do aggregation and optical links **as close as possible** to FE
 - “Lossy” data reduction, it looks a bit like trigger ? – but, at FE, level of **aggregation is limited by**
 - **geometrical distribution of sources**
 - **space available**
 - **Environment to place connectivity / intelligence (the usual...rad,temp,field...)**
- Transition to COTS
 - Link speeds are constrained by COTS standards
 - Speaks in favor of abandoning synchronous readout

Adelante...con juicio

- Turns out that “doing everything in software” is not so easy even with a HW L1 trigger
 - size of the switched network, amount of buffer to allow for “non-deterministic” behaviour of software algorithms
- amount of computing needed vs. CPU evolution
 - With asynchronous time-stamped data, just assembling the right pieces would not be a trivial task
 - Lots of low-hanging fruit ends up being a bit too high
 - GPUs...help but transition is slow (but at least it is happening)

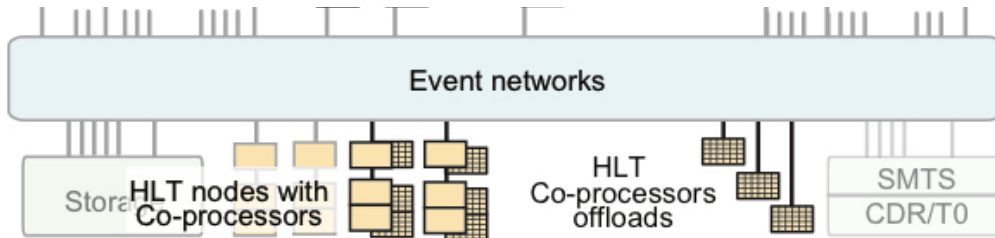


- ...people end-up proposing FPGA custom co-processors...
- ...or pre-processors (anyone ?) [while there is general consensus now that ASICs are hard a lot of people still think that custom boards with FPGAs are easy]...beats the purpose ?
 - Paying for CPU is not popular...people will want to develop their own boards
 - That's how projects get funded at institutions: **not to develop software, or even firmware** (see later in “sociology”)

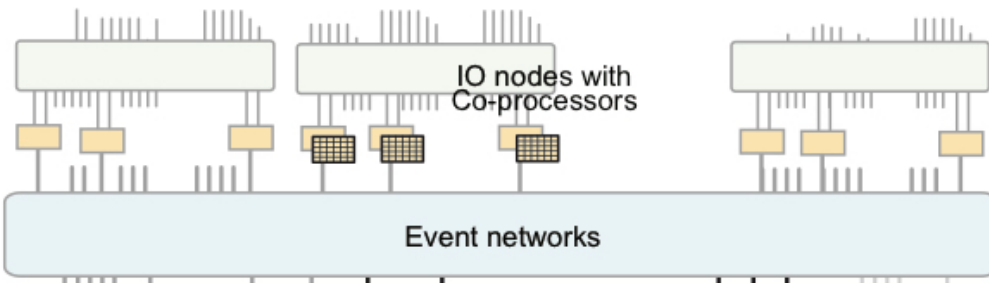
So...we also need to think about the use of co-processors...

”Grand-unification” of accelerator programming models (OneAPI...)

Abstraction libraries (e.g. alpaka...) – because not everybody has a GPU/FPGA/TPU



- GPGPU for memory-local algorithms
 - Pattern recognition
 - Space partitioning containers
 - Image processing
- FPGA-accelerated ML for transforms, feature extraction and classification

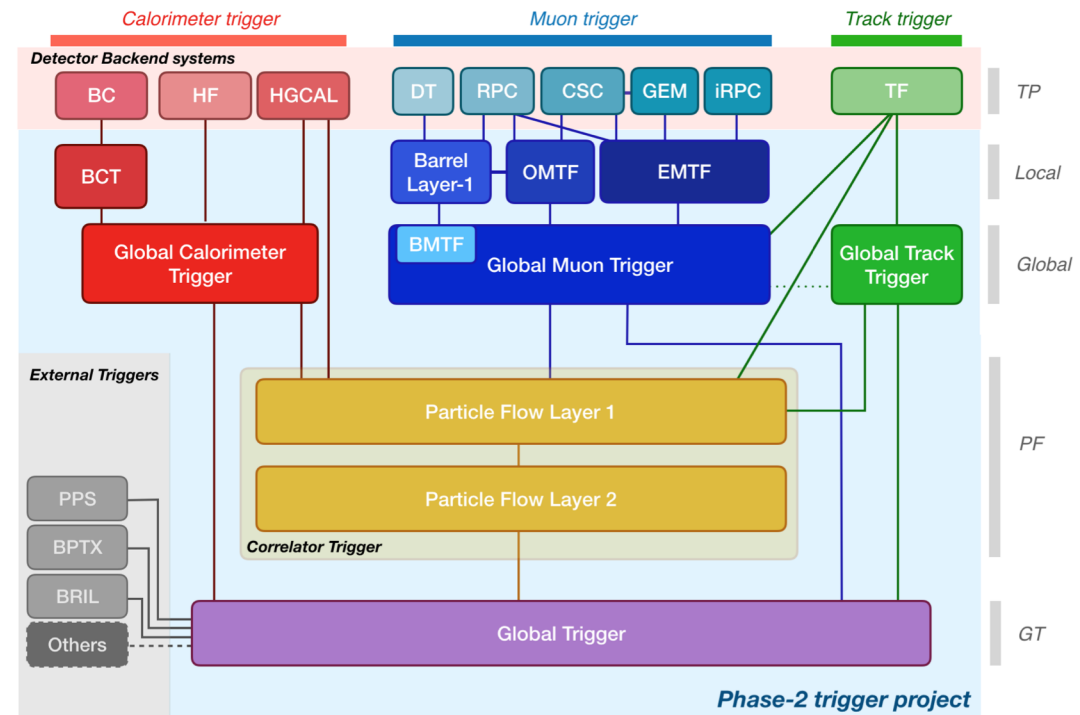
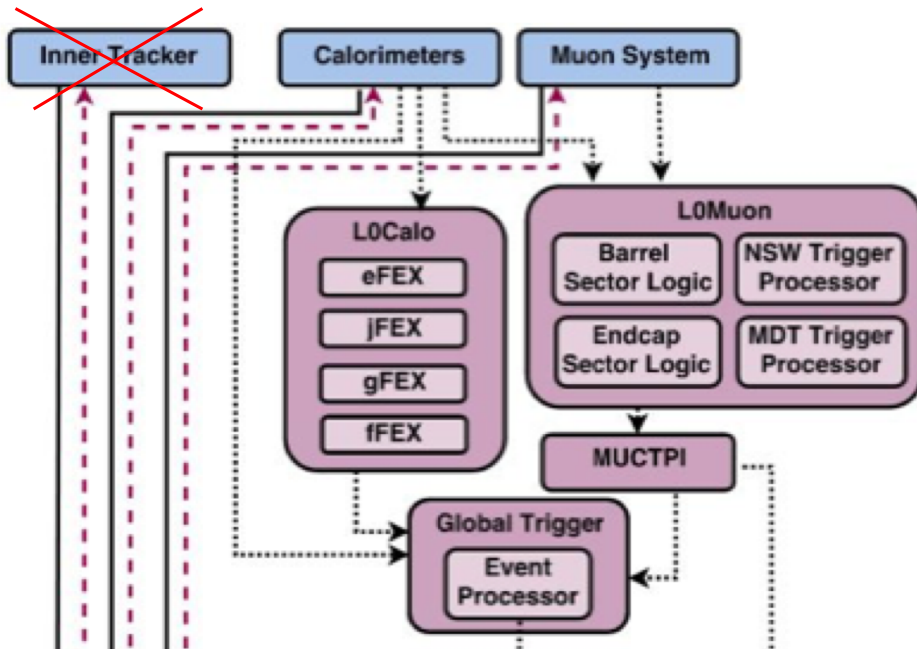


- May have to decide, case-by-case among
 - **Traditional approach** to instrument each HLT processor with offload engines
 - Limits choice of hardware
 - Forces early choice and backward compatibility
 - Creates “live-locks”
 - **Preemptive reconstruction** in dedicated co-processor “farms”
 - Choose the best hardware for the task
 - Add the right type of resources when needed
 - Easier upgrade
 - Limited number
- + need a lot of learning

This, if we solve the problem of the read-out – you can't get rid of complex front-end and back-end if you need a hw trigger

So...let's have a look at this hw trigger thing some want to get rid of

A look at the bad guys



Some draconian “**lossy compression**” here

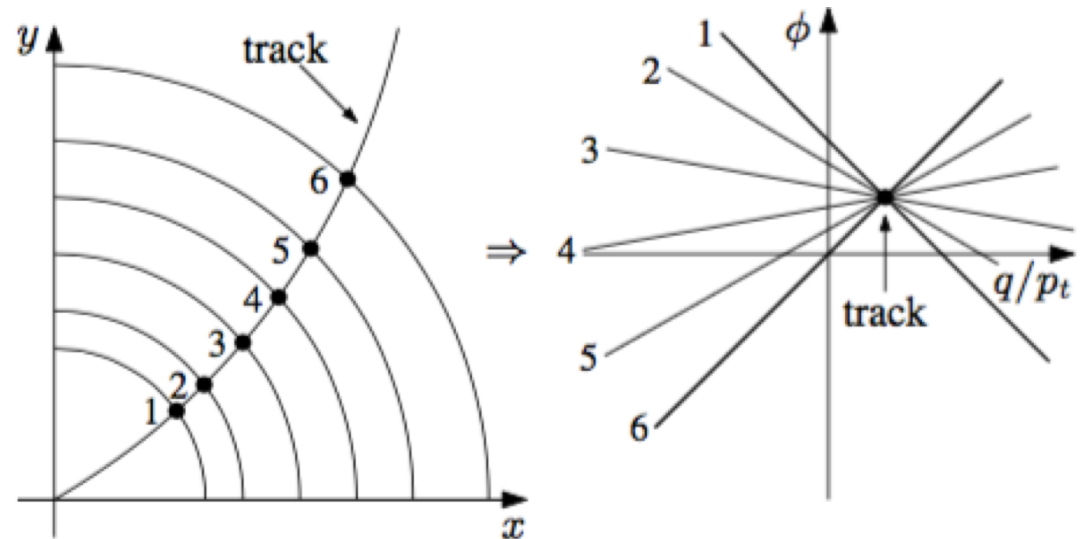
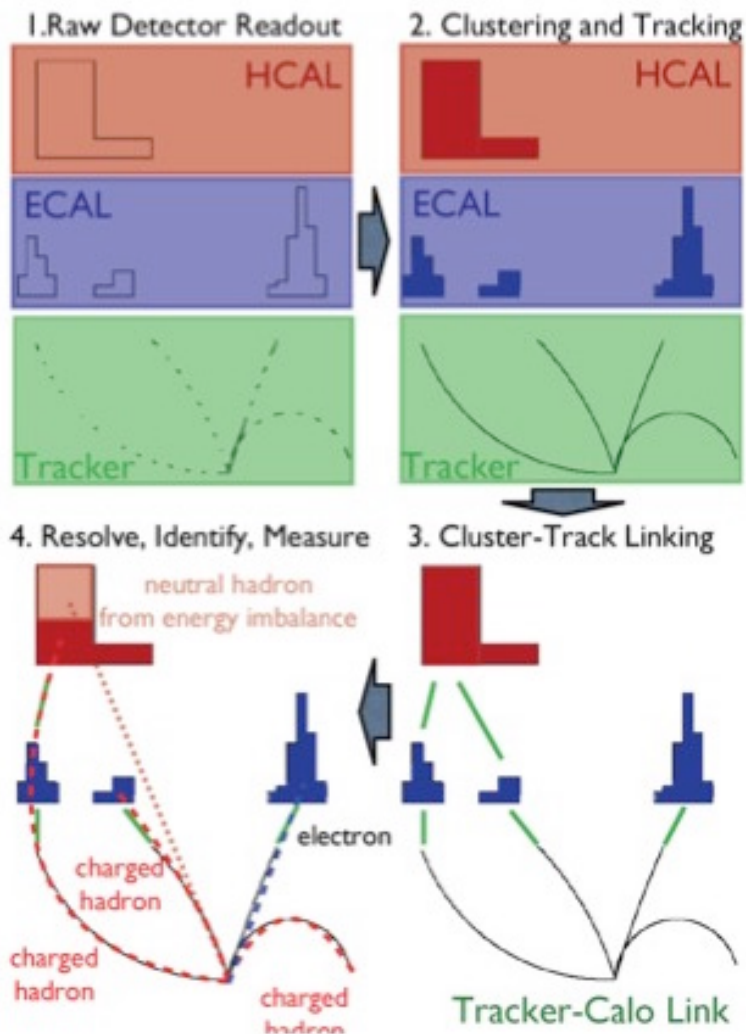
- Get rid of entire detectors, make “trigger cells”, “trigger primitives”
- Final outcome is...well, one bit

Some “trigger-less” readout

- Trigger inputs ARE **read out at BX rate** (in some cases “primitives” are made at back-end, from streaming data)
- Trigger processors do **aggregation**
 - In some cases information from **multiple sub-detectors** are combined

There is a lot of information in the intermediate layers of the hw trigger...perhaps something to be learned there

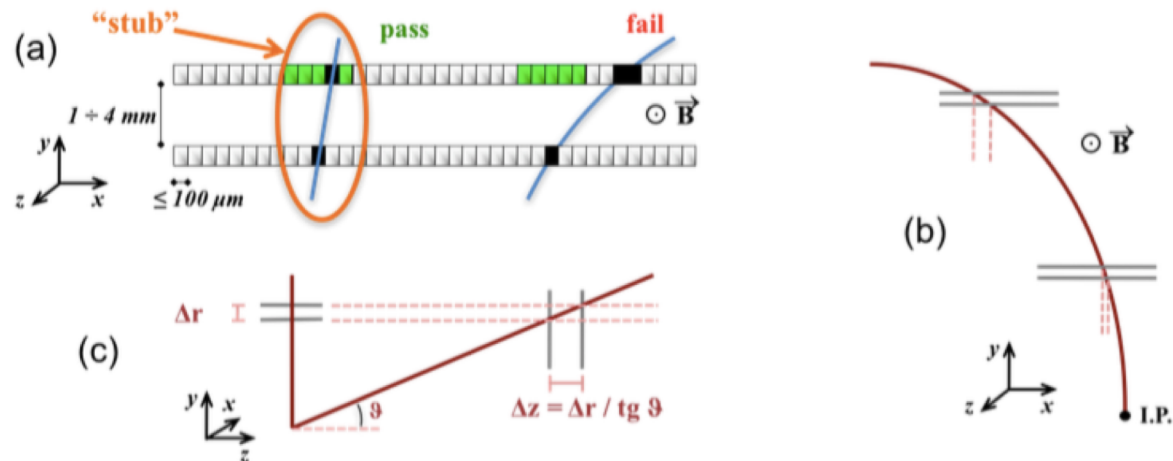
Yet one wants to do fancy things



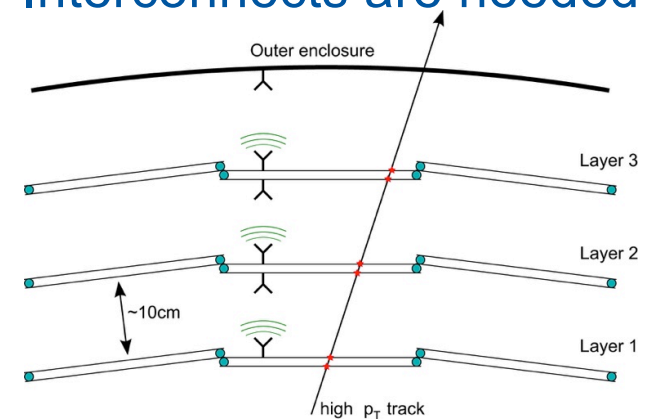
Several aspects to point out here:

- Enough information to “do the job”
- Aggregate it in the correct dimension
- Do all this within a short time
- Modular systems with custom interconnects
- Lots of complex algorithms to code in firmware

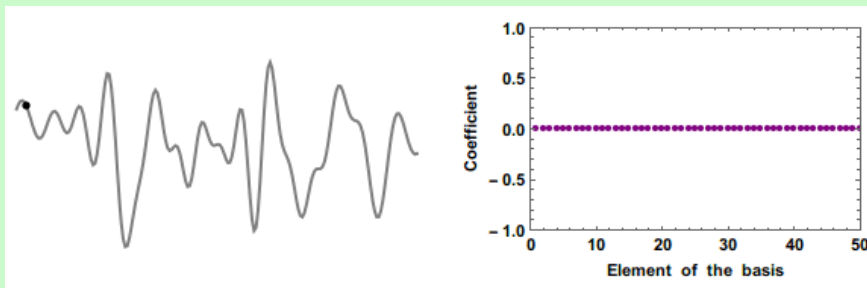
Some intelligence in the detector...



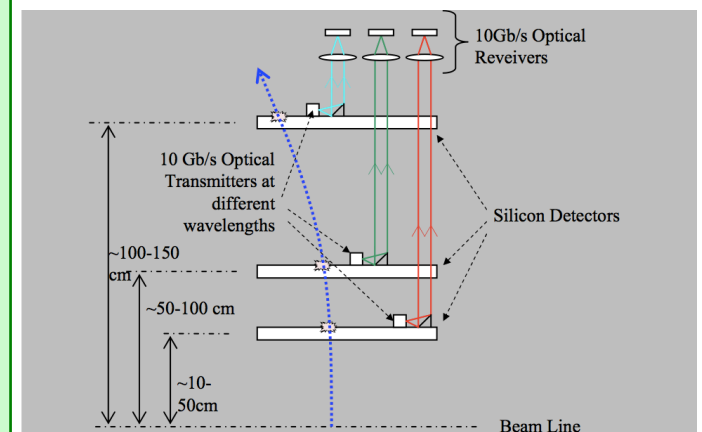
If one wants to do more Interconnects are needed



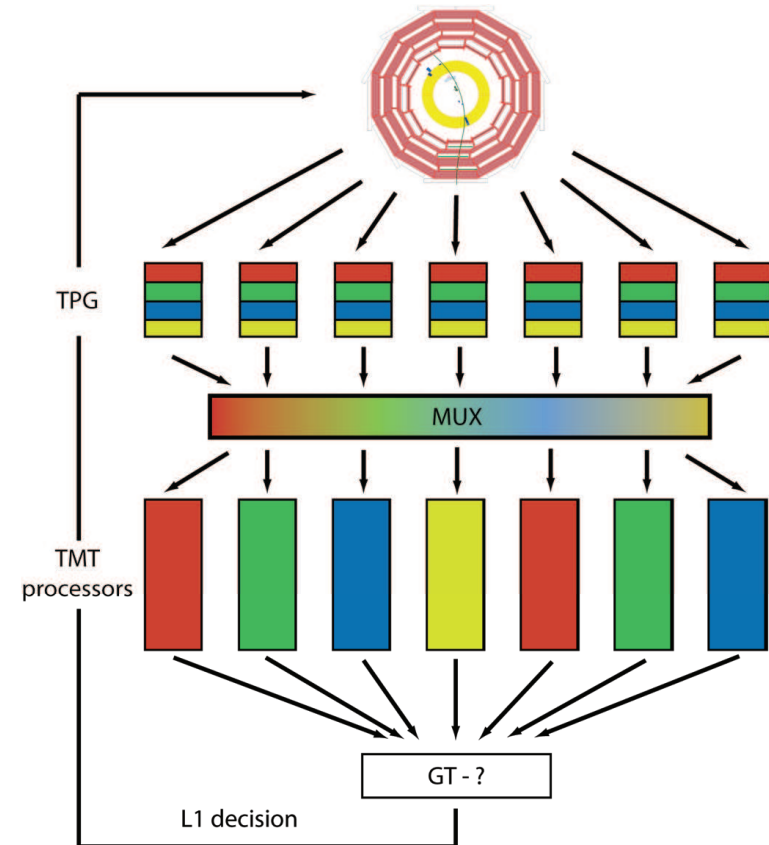
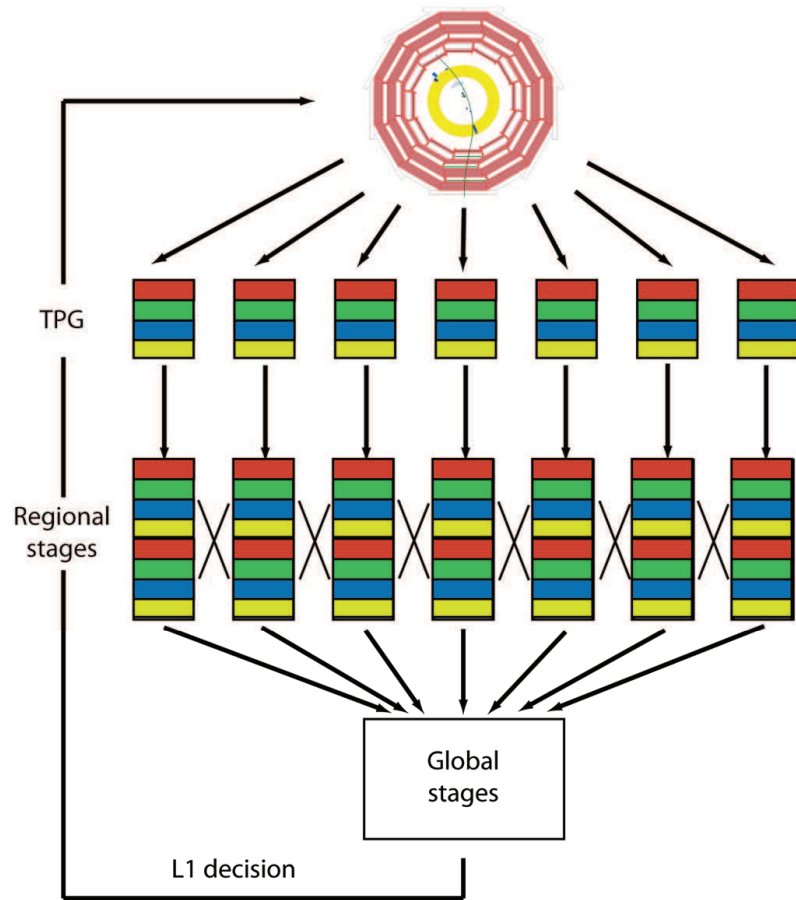
Inefficient at low P_t , large impact parameter, large η



Lossy compression requires sparsity in some space
VFE connectivity insufficient (or in the wrong dimension)



Complex algorithms in “short” time



-
- Zero-th order problem: how to use complex, highly granular detectors in the trigger
 - First order problem: how to make high-rate read-out possible (with or without trigger)
 - You can't get rid of complex front-end (and back-end) if you can't get rid of a hardware trigger
 - Second order problem: build a tracker that spits out tracks, a calorimeter that spits out clusters ?

Some cursory conclusion

Hardware L1 is now a system of (generic) processors doing “local” event building, and processing events in parallel – **L1 and HLT are approaching**

- As an aside, this can and must be exploited to reduce the computation needs of the HLT
 - L1 objects much more useful
- Can capture L1 data for use in “physics at BX rate” (L1 “scouting”)

Both L1 and HLT only require “**trivial**” **parallelism**: work on one BX (event) at a time.

- Driven by the almost-synchronous nature of the task – could be challenged by **asynchronous readout** and/or **multi-bx phenomena** (VLL)
- Information **only flows in one direction** – sub-optimal use of bi-directional links (but easy traffic) [on-demand event building...]

Some cursory conclusion

L1 can live with (very) lossy compression because it just needs enough information to classify BXs and be “**mostly correct**” – that L1 is only mostly correct is **an accepted fact of life** (see below)

Whatever readout scheme one chooses **needs to be “almost always correct”** – at the lowest level possible (i.e. not miss a track or know exactly what you missed):

- Ability to **extract all relevant information** from FE
 - Readout scheme
 - Lossy compression and traffic equalization (CNN, AE (?), CS (?))
- Ability to **aggregate and process data at sufficient scale** “on-detector”
 - Interconnect at “quasi” front-end
 - Powerful FPGA-based processors that can work in QFE environment
 - Still a lot of downstream connectivity, if possible on COTS
- Accept a redefinition of what “raw-data” means (notice that such a redefinition is mostly NOT yet accepted at the next level, i.e. HLT)

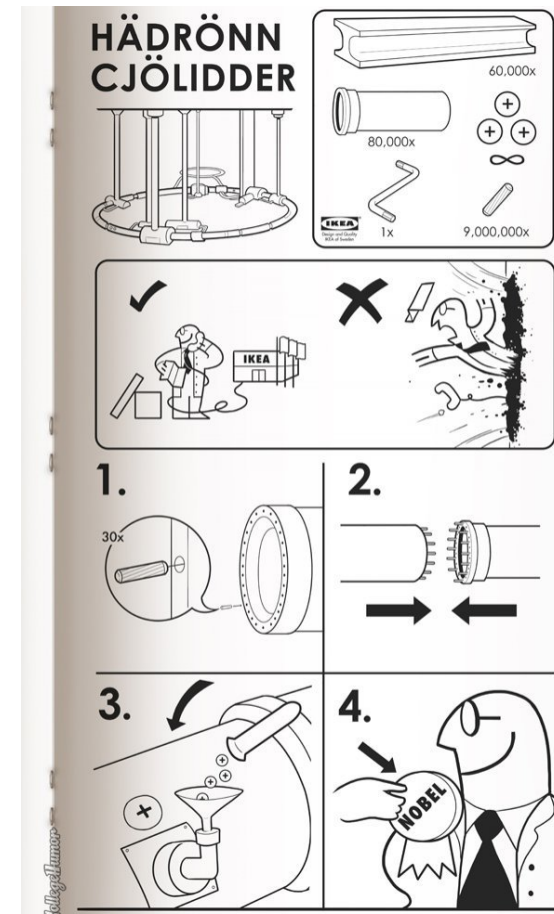
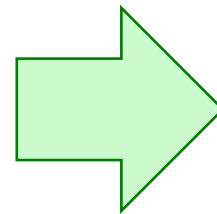
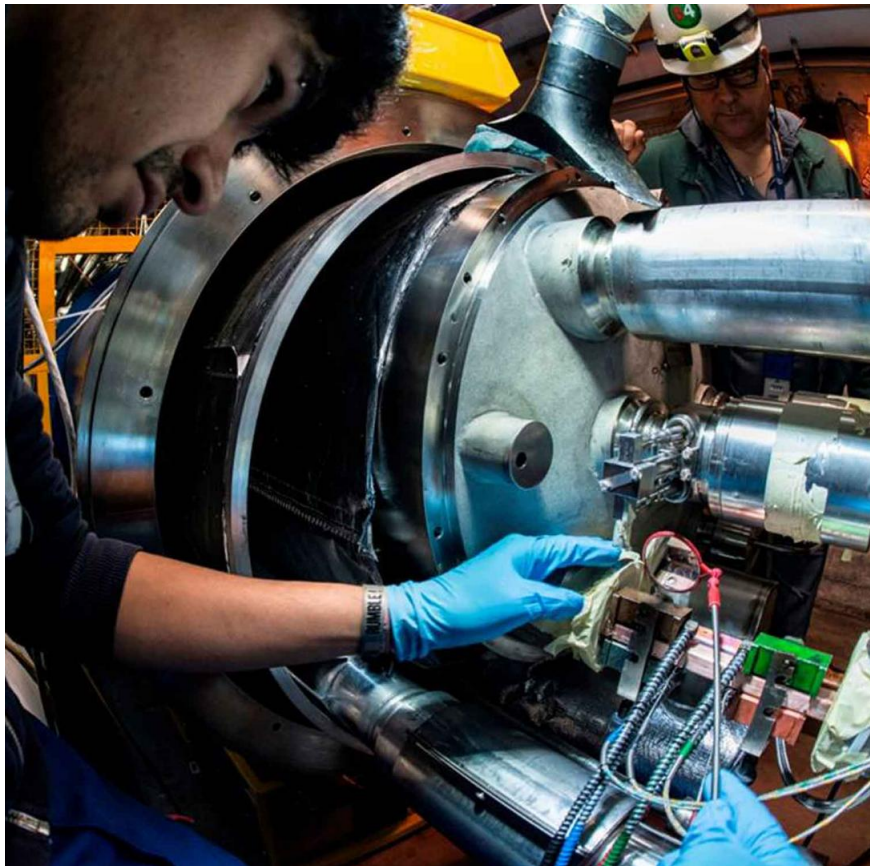
**Need “holistic” system engineering
at detector design level that takes into
account readout (and trigger if
necessary)**

What (would be) needed

- **Read-out optics** (of course) – and what goes with it, trigger and timing distribution, or precision clocks for trigger-less systems
- Data aggregation at **on-detector boards with FPGA/SoCs** characterized for moderate radiation
 - Material, powering, cooling
- Correctly **connected** for inter-operation (e.g. multiple tracker or calorimeter layers, overlaps)
- Using specific resources for **buffering** (least impact on logics usability)
- **Bidirectional** optical links used to configure/steer/control operation
 - SoCs could offer substantial advantages
 - Deal with communication, configuration, calibration
 - FPGA does the algorithms, possible use of ML techniques (e.g. clustering, classification)
- **Need extra-fast calibration loops**

Pain points: Industry

(developed by others)
(for commercial applications)



HEP and Industry

- In many respect HEP just niche customers
 - Board technology (PCB layout, materials, high-speed design) becoming a science of its own
 - FPGAs (high-speed I/O vs. generic logics) – the big customers want AI-oriented architectures
 - SoCs – heading for autonomous vehicles, IOT, automation in general: how to best make use of them
 - Hyperscalers ~~Telecoms~~ define (when) the next link speed is needed
 - High performance ~~throughput~~ computing drives the cluster interconnects (and latency is their prime concern)
- Buy solutions, or learn from industry ?
 - Clearly, industry does not cater for all our needs
 - The processes used by industry are person-power-intensive
 - Buying a solution is often out of reach because funding is for development
- Focused R&D
 - Rad-hard Si-photonics ?
 - Rad-hard FPGAs (with some limited overlap with space science)
 - Hardened logics
 - Board design, cooling, whatnots...
 - Distributed processing (run algorithm where data are) from IOT, autonomous vehicles...

Pain points: Software

- At the lowest-level, i.e. hardware access, at least as many “frameworks” as there are boards
 - Long-term maintenance hard (because sw is often done by students)
 - Grand-unification attempts needed here
 - Effort to combine this with fw in common CI/CD
- At the next level, integration
 - Person-power intensive and often delegated to “last in line”
- Data Transport
 - A lot of commonality but lack of interest in code reuse
 - Optimization for different networks
 - Sharing test setups for the above
- Control
 - Fast and slow commonalities not exploited
 - SCADA products adopted at different levels
 - New needs with less and less accessible experiments

”Sociology”

- Independent developments: hardware, firmware and software
 - ...definitely more solutions developed than problems to solve
 - In big collaborations, institutes/labs funded for “new” projects (that means hw)
 - Central teams with expertise vs. multiple institutions, consortia
 - Flip side: More creativity, Engineering staff at institutions stay up-to-date, expertise improved/retained
- Good Practices
 - For hardware: sometimes the job is done when the board is out and tested.
 - Integration, commissioning, maintenance often fall on the shoulders of “central” groups
 - Engineers tend to blame physicists
 - It is true that the physicist philosophy is “get it done”
 - It is also true that the TCO of cheap solutions is seldom taken into account
 - On the other hand, the field pathologically lacks person-power
 - Graduate students and post-docs used as cheap replacement for technical staff
- Having to choose, a more ambitious detector is chosen over a more practical off-detector system (readout, trigger, data aggregation and pre-processing are still second thoughts in the mind of many decision-makers)

”Sociology”

- Do other fields do better than HEP ?
 - At first glance it may seem so: (e.g. SKA process) – but the devil is in the detail and one should stay tuned to see how the story ends.
 - The structure of collaborations tends to hamper **centralized decision-making** (because the funds are elsewhere). Other schemes, such as consortia may work better (but the decision-making must be good)
- Can we do better ?
 - Situation may (paradoxically) get better as developing e.g. a board becomes too complex for small teams
 - Modular solutions w/ **independent customization** may be “sold” as new development
- Why people are not complaining about the same things for software ?

Summary

In the future we will only do trigger-less architectures

Yes but how do I deal with cooling, material, and how to do data reduction ? Ah and we've got more radiation and higher magnetic field

Mmh...maybe we need a trigger after all

Okay, but now I want to make my detector 100 times more granular so the trigger will have to use that information

I can make it but will cost a lot to get and process all those data and accept rate will be rather large anyway..

But CPU are expensive and my HLT sucks...can you help me with GPUs ?

We need intelligent front-end!

By the way engineers at Uvattelapesc need a project so they get funding... maybe we make a board?

I cannot afford a new ASIC, plus my detector has a very complex geometry... can you do that with an FPGA that can work in xx fluence ?

We could do lossy compression !

I am not prepared to do away with my raw data...

I will think about the rad-hard FPGA R&D business

I have no money for that R&D because I spent it all on sensors...