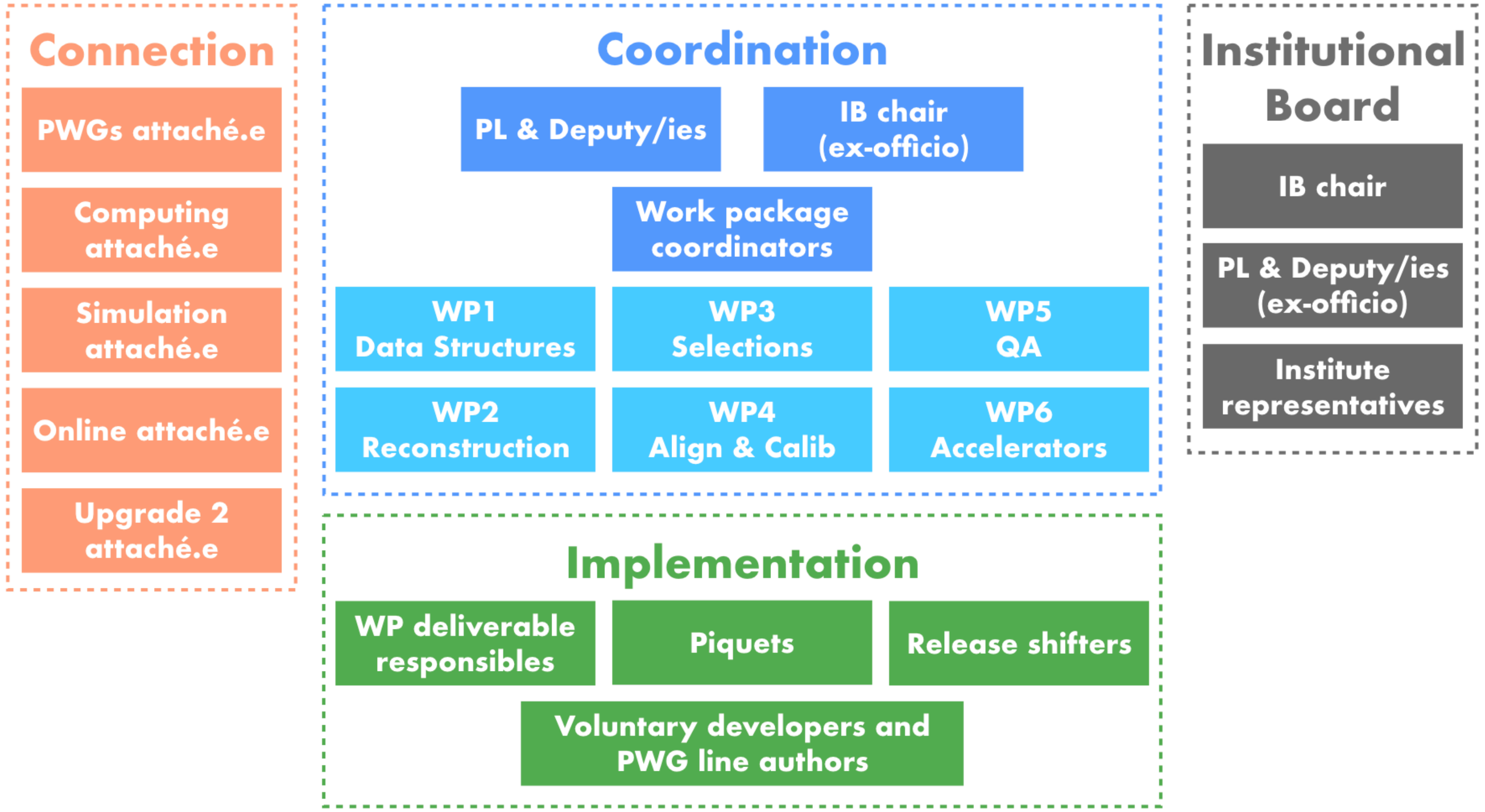
The LHCb real-time analysis project

What is it, why do we need it, and how will it develop?



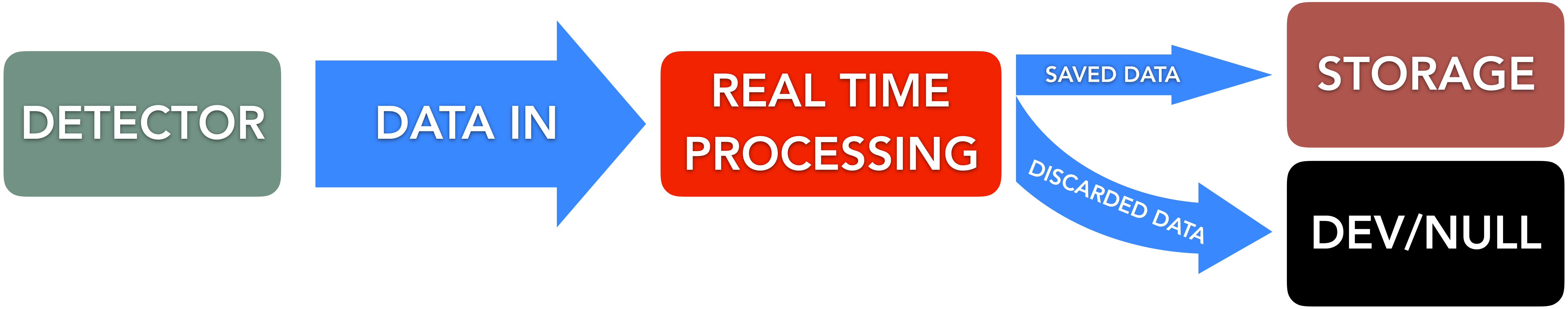
European Research Council
Established by the European Commission

V. V. Gligorov, CNRS/LPNHE

Jahrestreffen der deutschen LHCb-Gruppen

Rostock, 01.10.2019

Q : What is real-time?



A : Any processing of data before it is permanently recorded

Why do we need to process data before recording it?

	LHCb	CMS/ATLAS
Data volume at detector	~30 Eb/ year	~1 Zb/ year

Why do we need to process data before recording it?

	LHCb		CMS/ATLAS
Data volume at detector	~30 Eb/ year	<u>Global internet dataflow 2015</u>: ~640 Eb/ year	~1 Zb/ year

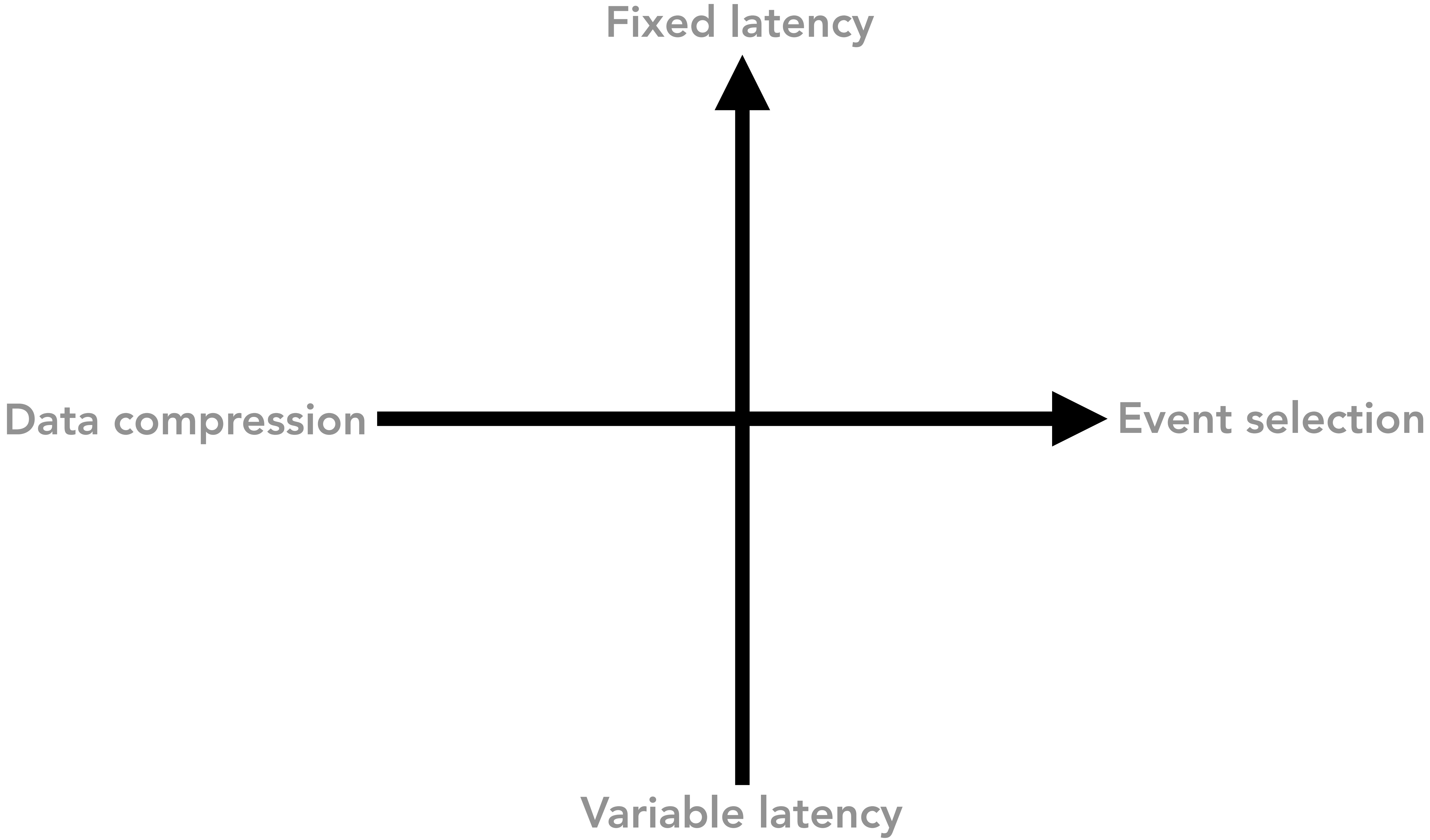
Because HEP detectors produce too much data to store

Data volumes @ LHC after real-time processing

	LHCb	Global internet dataflow 2015	CMS/ATLAS
Data volume at detector	~30 Eb/ year	~640 Eb/ year	~1 Zb/ year
Data volume for analysts	~30 Pb/ year		~40 Pb/ year

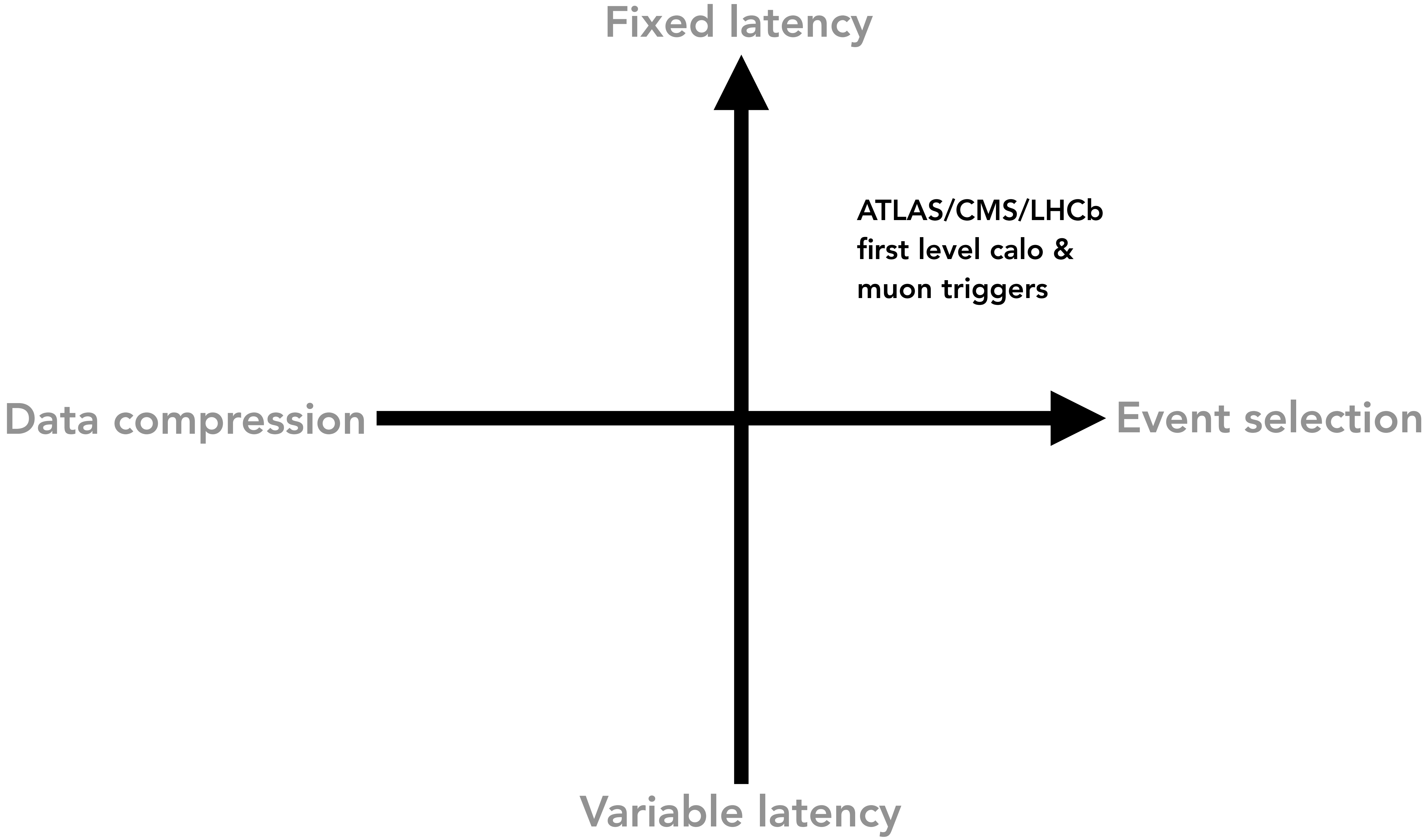
Real-time processing reduces data by 3-5 orders of magnitude

What kinds of real-time data processings exist?



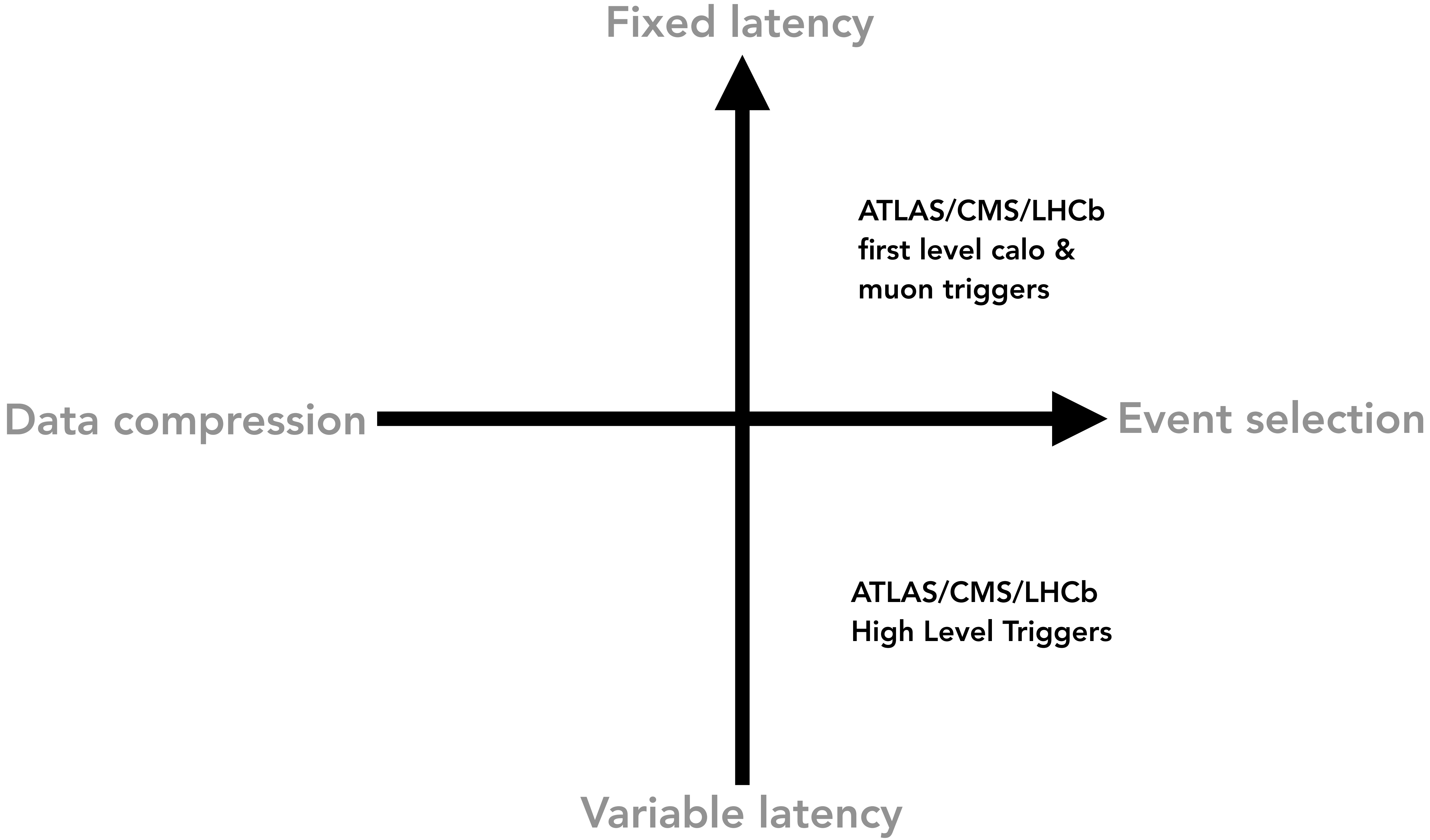
Distinguish fixed & variable latency, selection & compression

What kinds of real-time data processings exist?



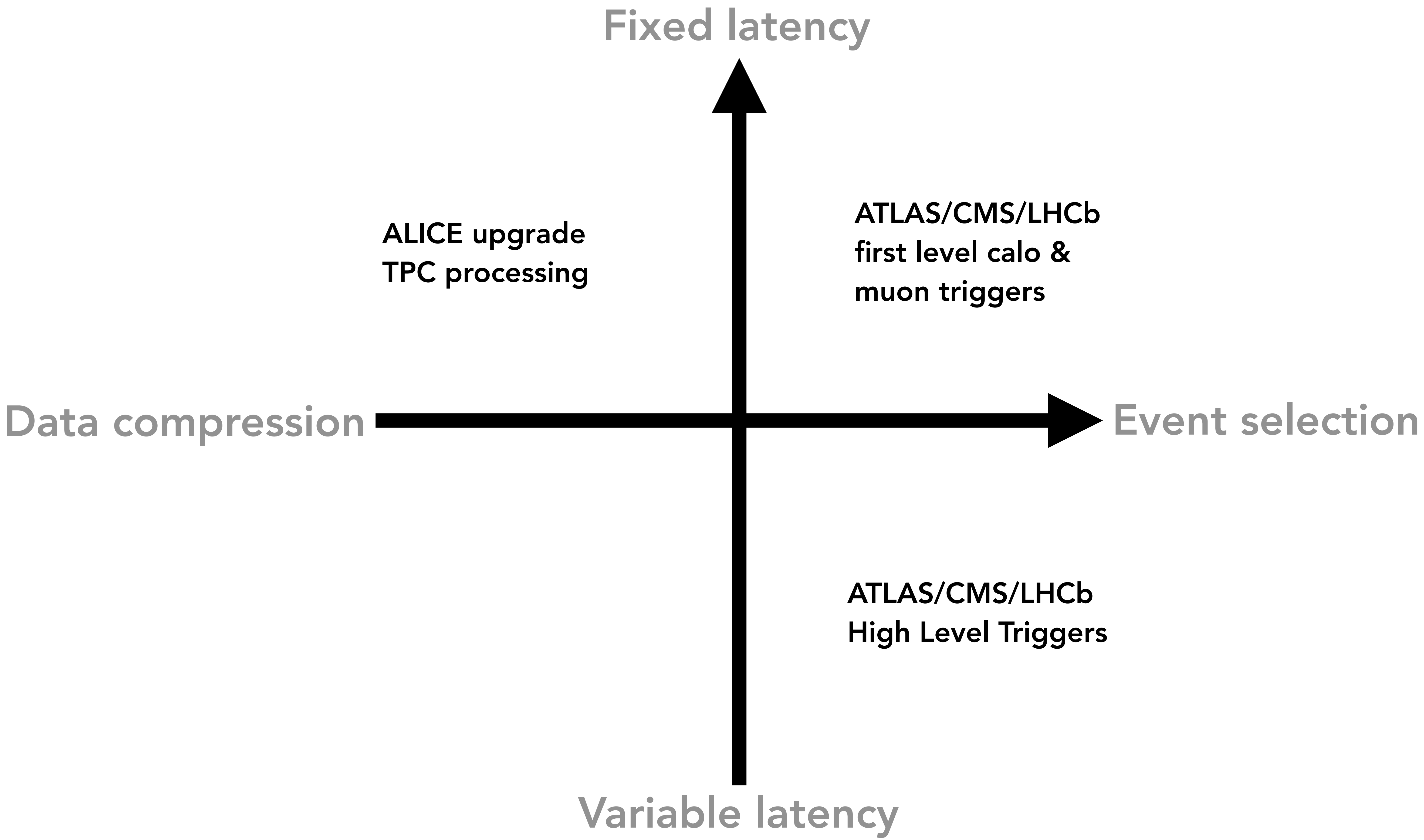
Distinguish fixed & variable latency, selection & compression

What kinds of real-time data processings exist?



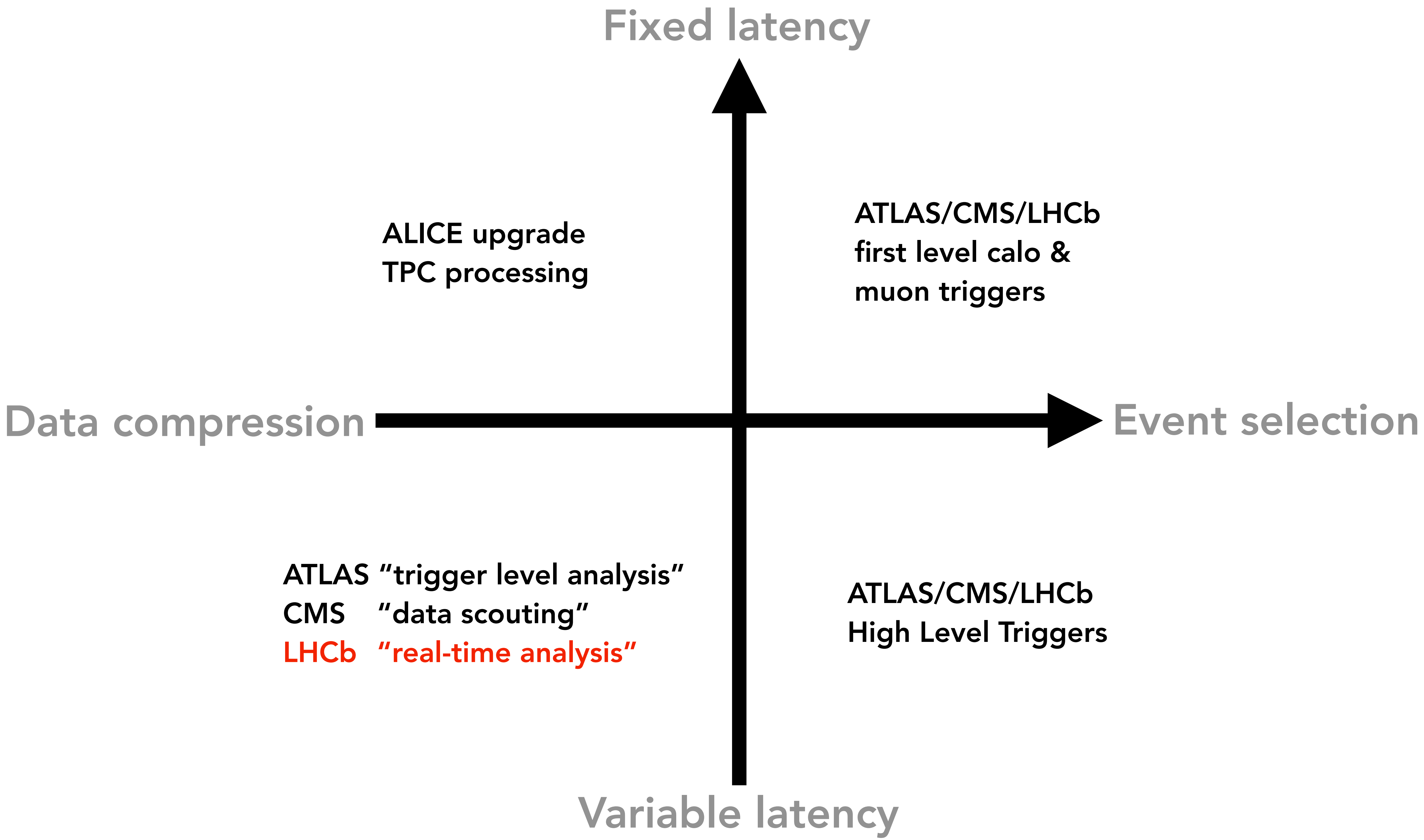
Distinguish fixed & variable latency, selection & compression

What kinds of real-time data processings exist?



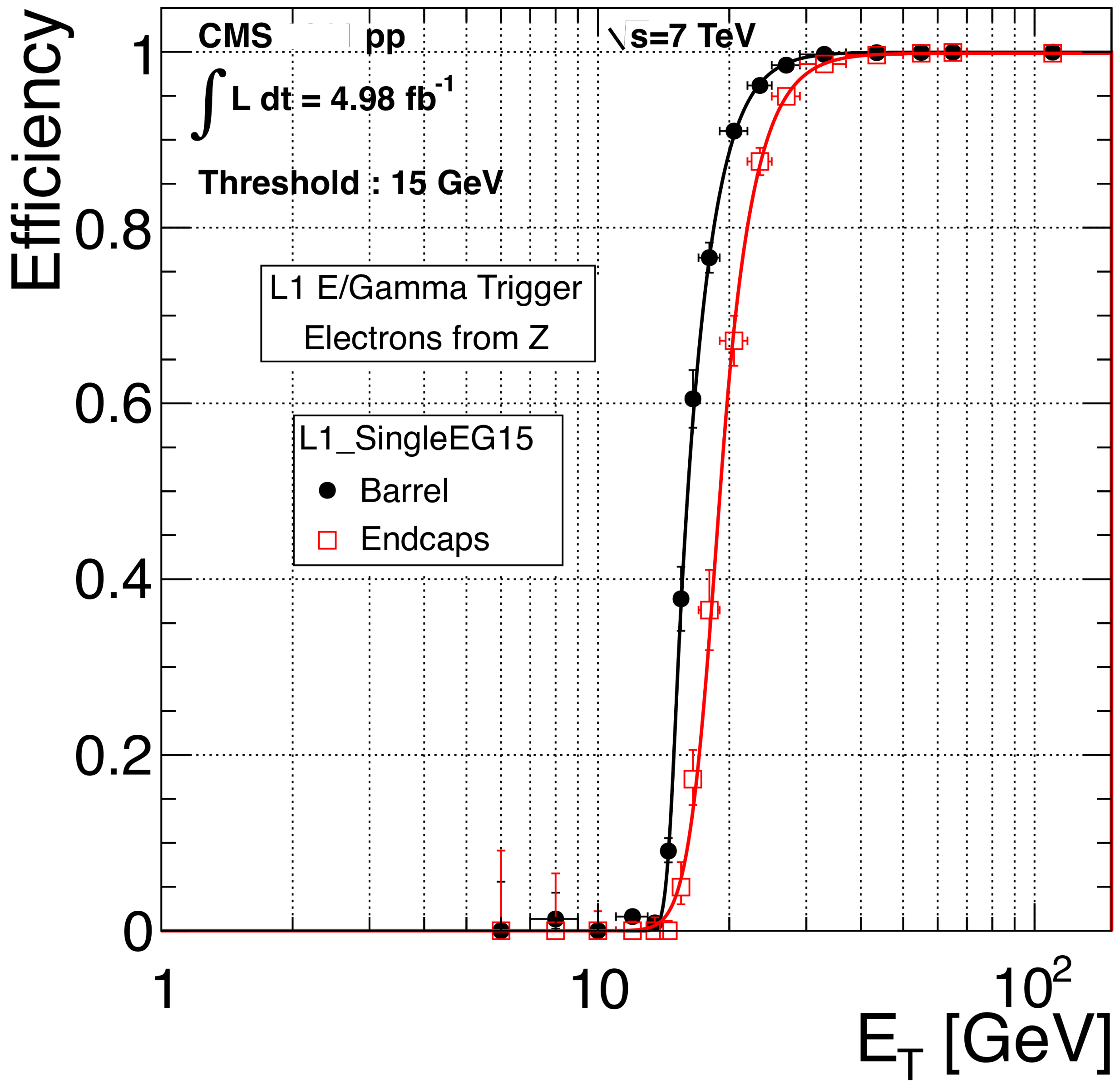
Distinguish fixed & variable latency, selection & compression

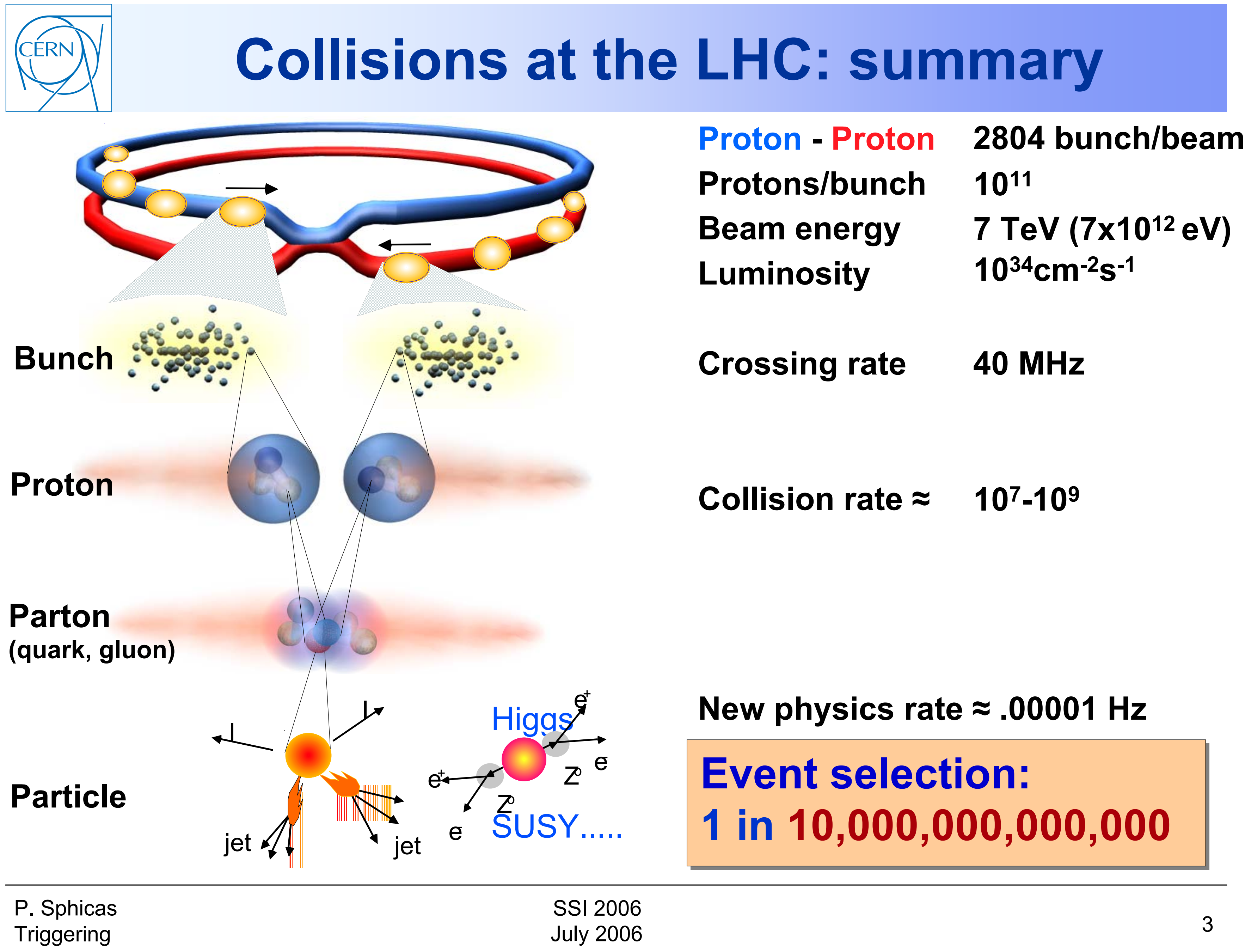
What kinds of real-time data processings exist?



Distinguish fixed & variable latency, selection & compression

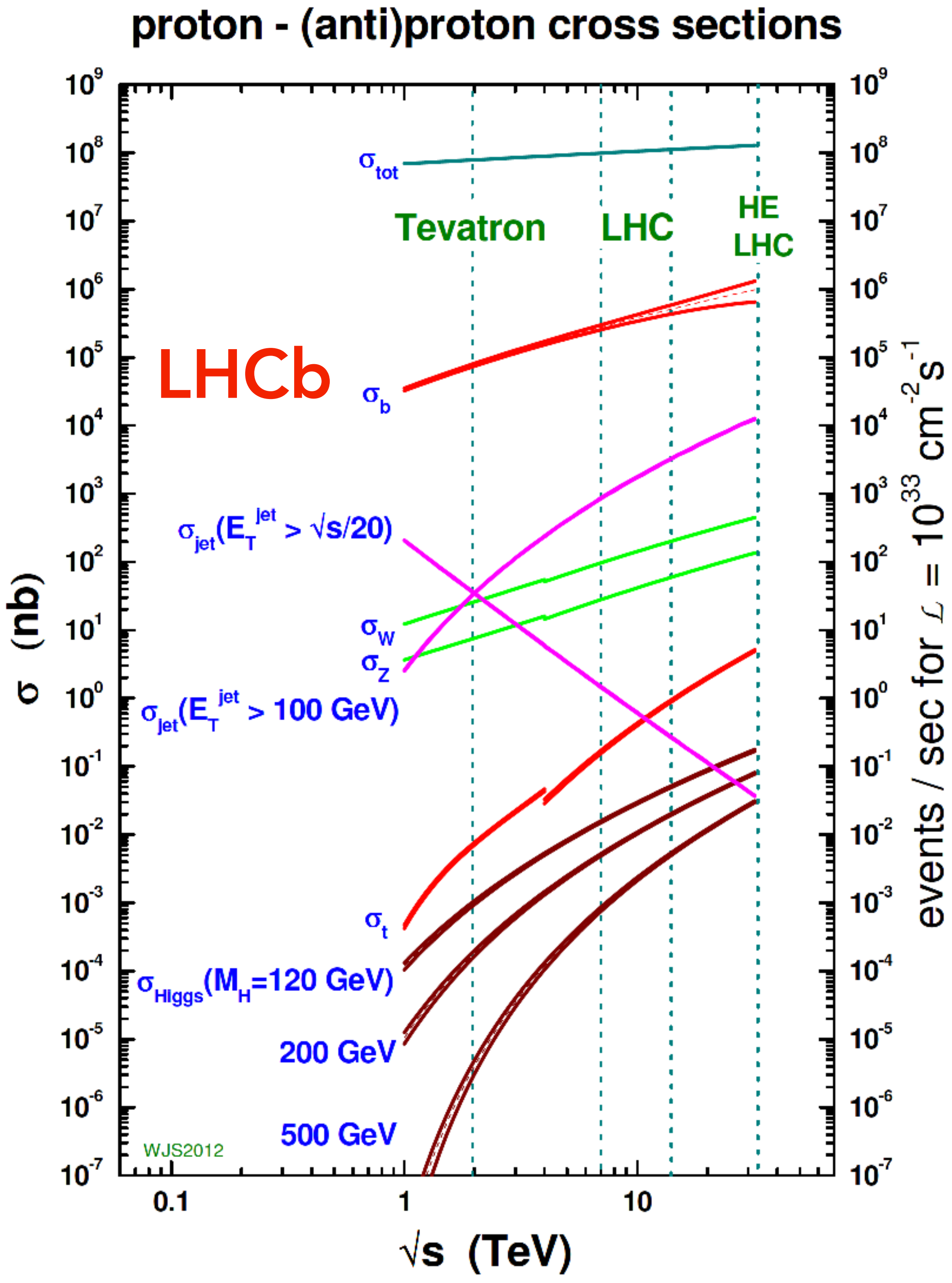
Traditional real-time processing, or "triggering"



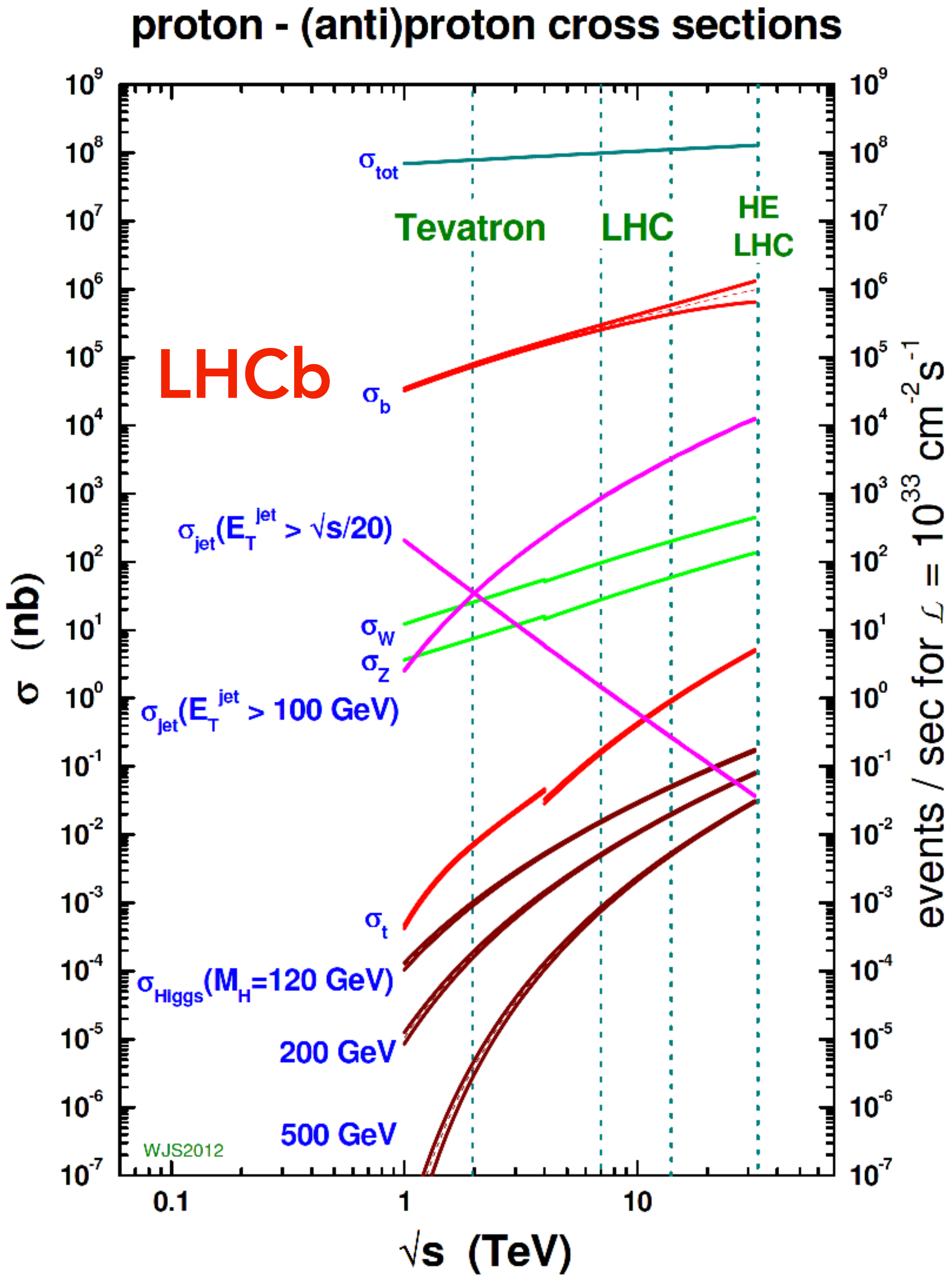


Driven by fixed-latency selection, analysis on efficiency plateau

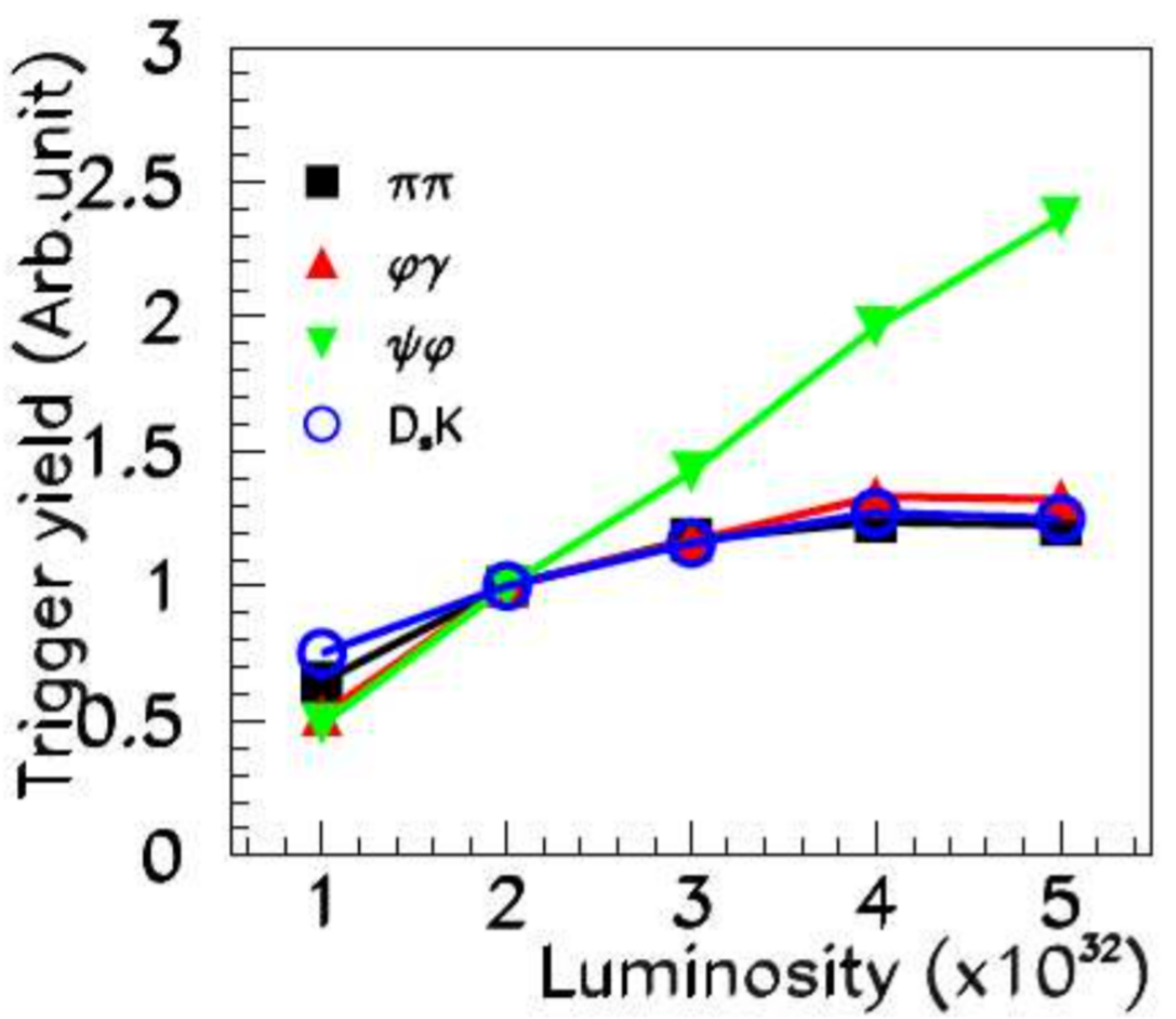
Why does LHCb not run at ATLAS/CMS luminosities today?



Why does LHCb not run at ATLAS/CMS luminosities today?



The plot which basically motivated the LHCb upgrade



Fixed-latency trigger only effective up to around $4 \cdot 10^{32}$

Let's consider the implications of this plot for a moment

Beyond a certain pileup, most bunch crossings will contain charm or beauty hadrons.

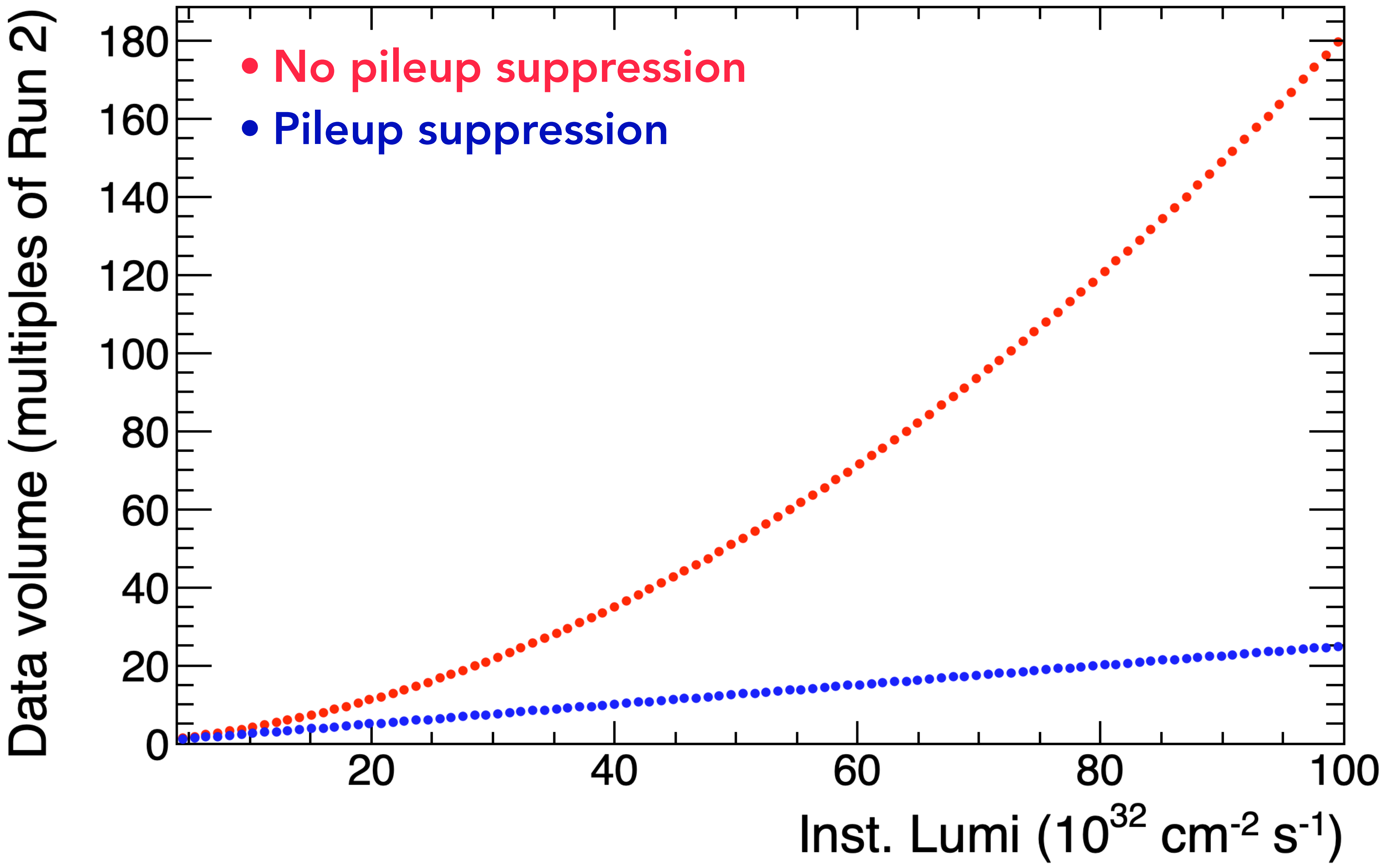
But not all beauty and charm hadrons decay in way which are interesting to us.

We cannot distinguish between interesting and uninteresting bunch crossings without bringing together information from the whole detector to infer *what kind* of beauty or charm hadron decay occurred in a given bunch crossing

This is where our first upgrade design choice comes from: read out the full detector at 30 MHz and make all data available for variable latency (asynchronous) processing. Very hard problem but if you succeed the processing architecture is actually quite simple, compact, and extremely flexible. I'll come back to this later.

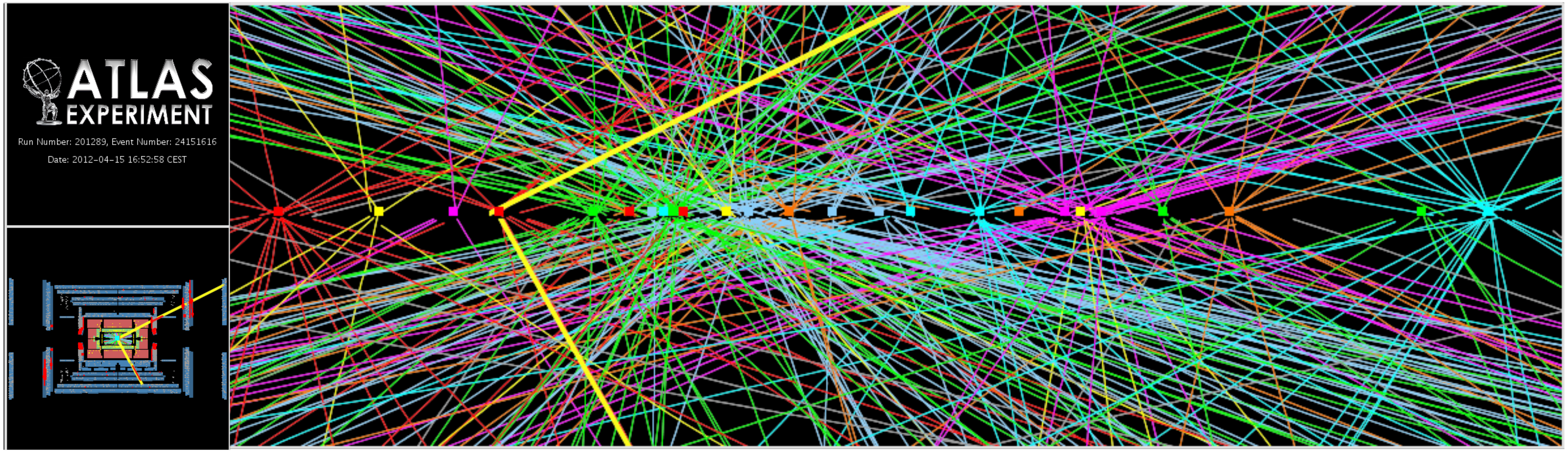
Or in a picture...

And what about data volumes?



Data volume increases quadratically even with 0 background.
Select pp collisions, not bunch crossings, in real time!

From selection to compression : real-time analysis



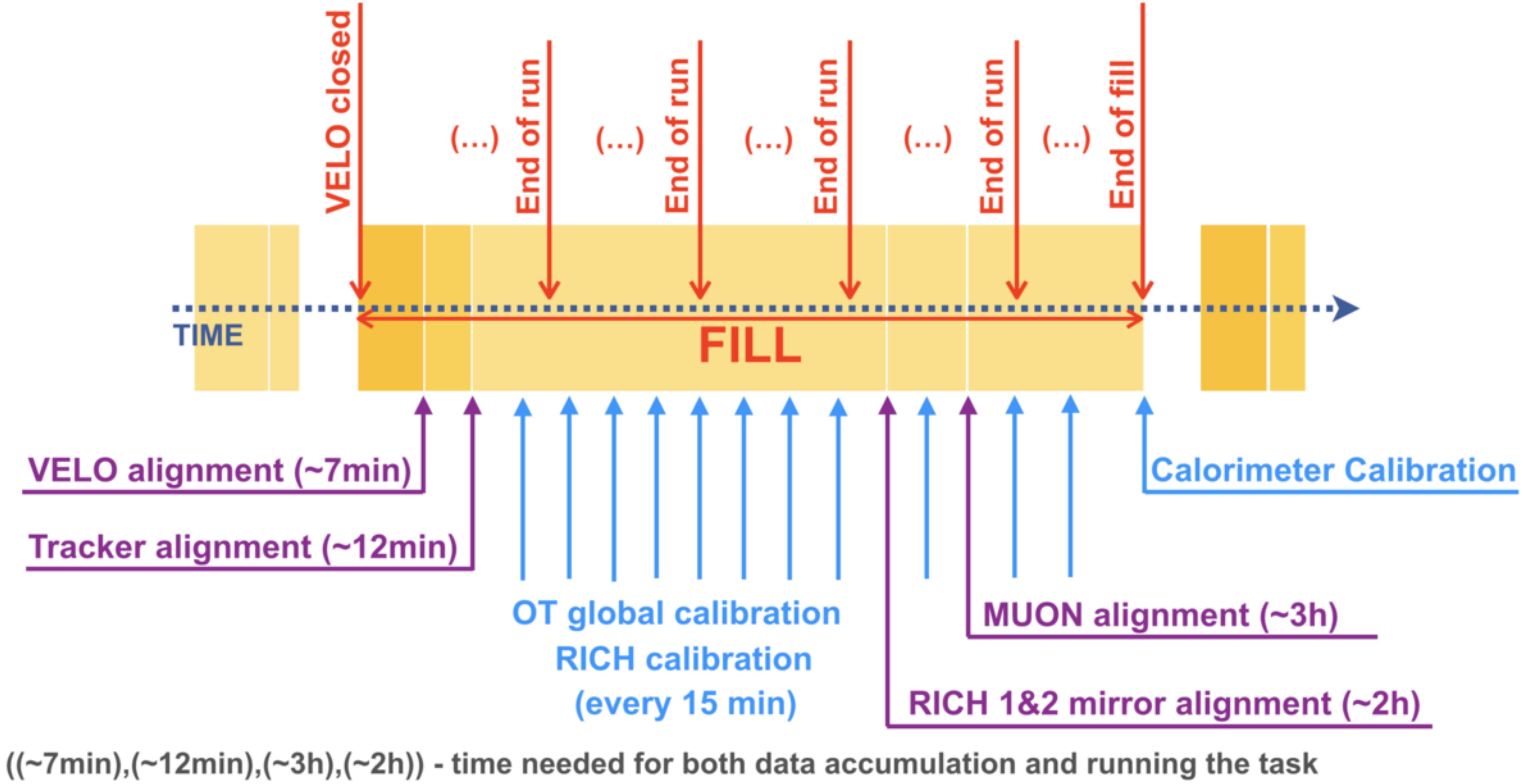
Most physics measurements require only a signal candidate and information about the specific pp collision which produced it → the rest is pileup

The higher the luminosity, the larger the fraction of event data caused by pileup

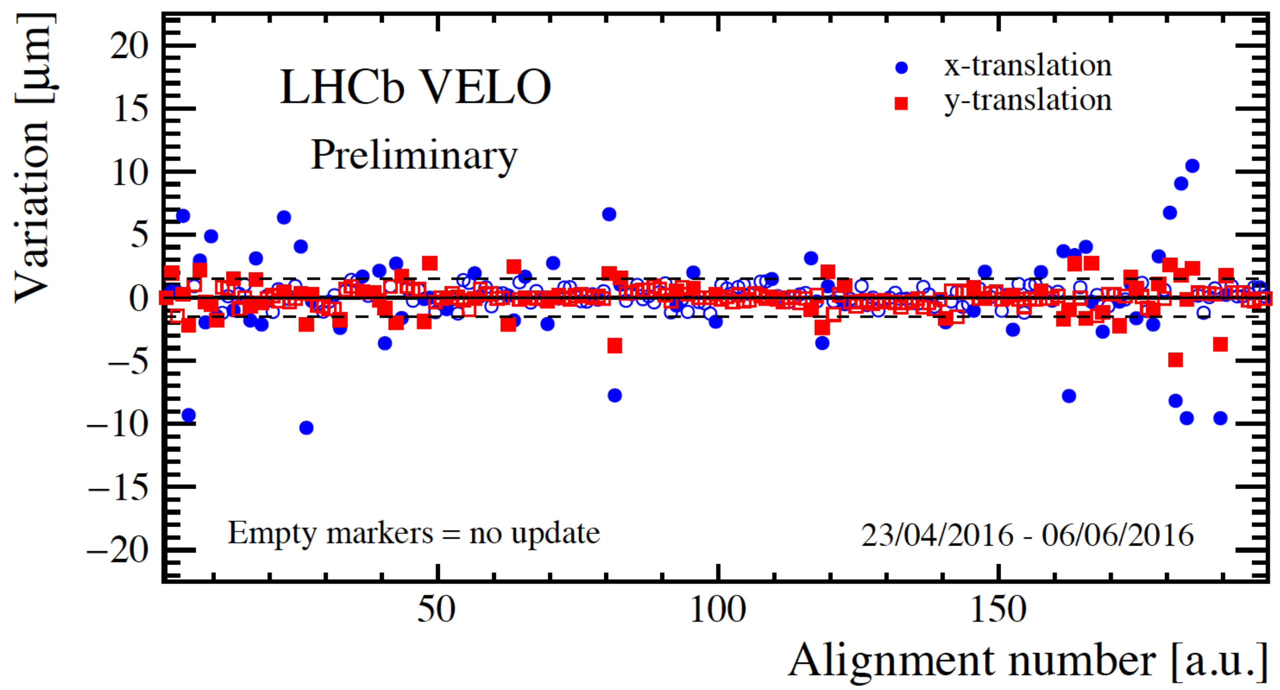
Hence create more room for signal by compressing & removing pileup in real-time!

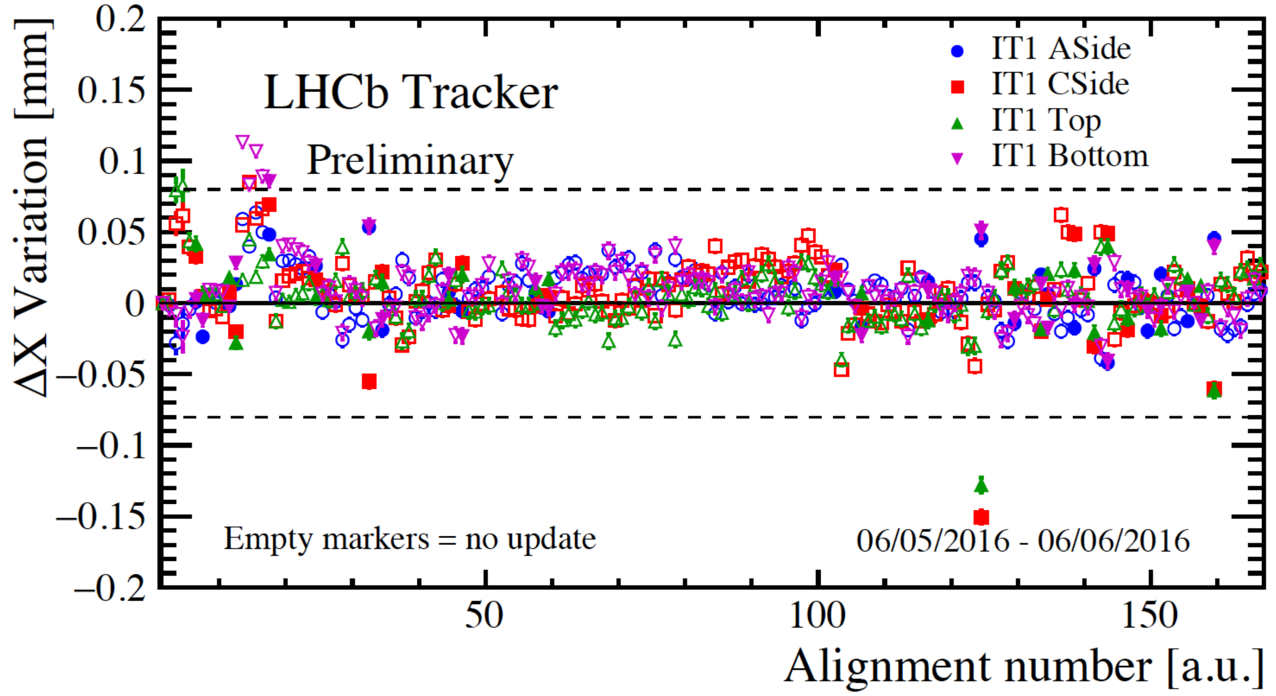
So how do we carry out precise pileup suppression?

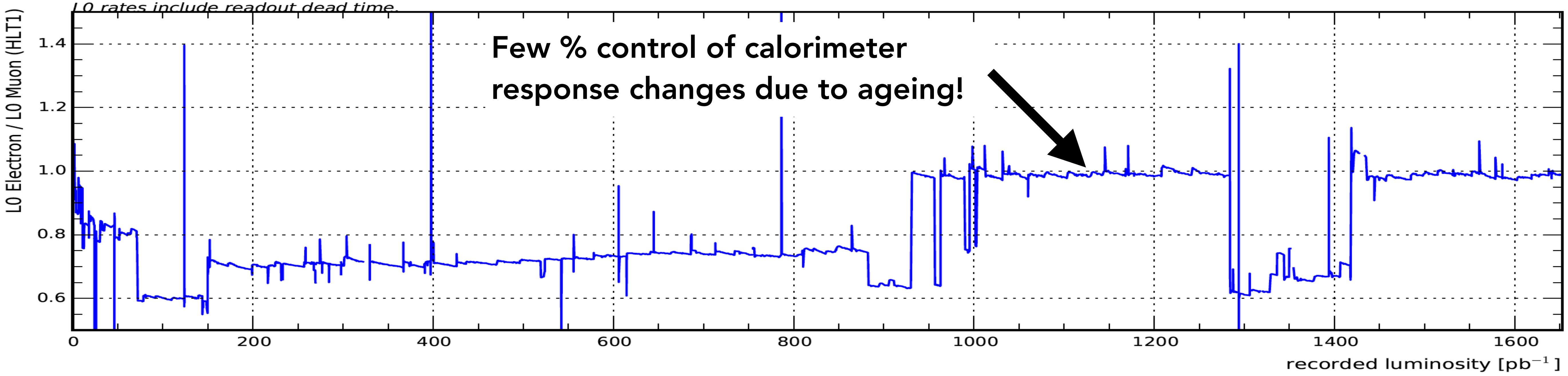
We also need to align and calibrate our detector in real time



So we did!



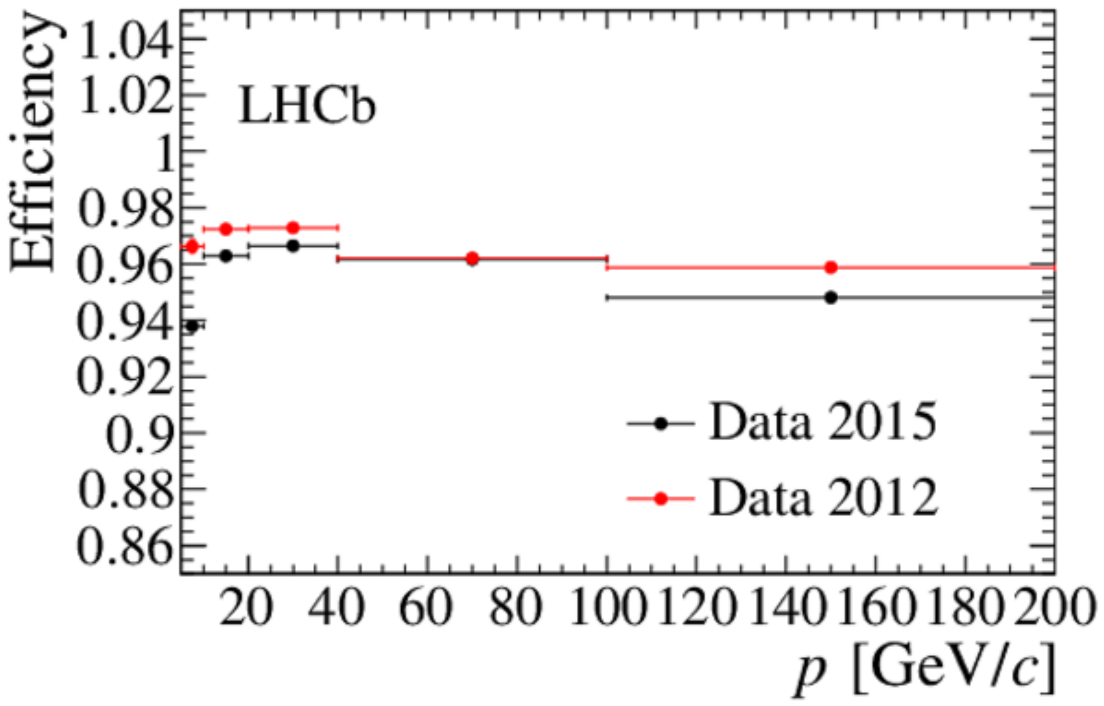


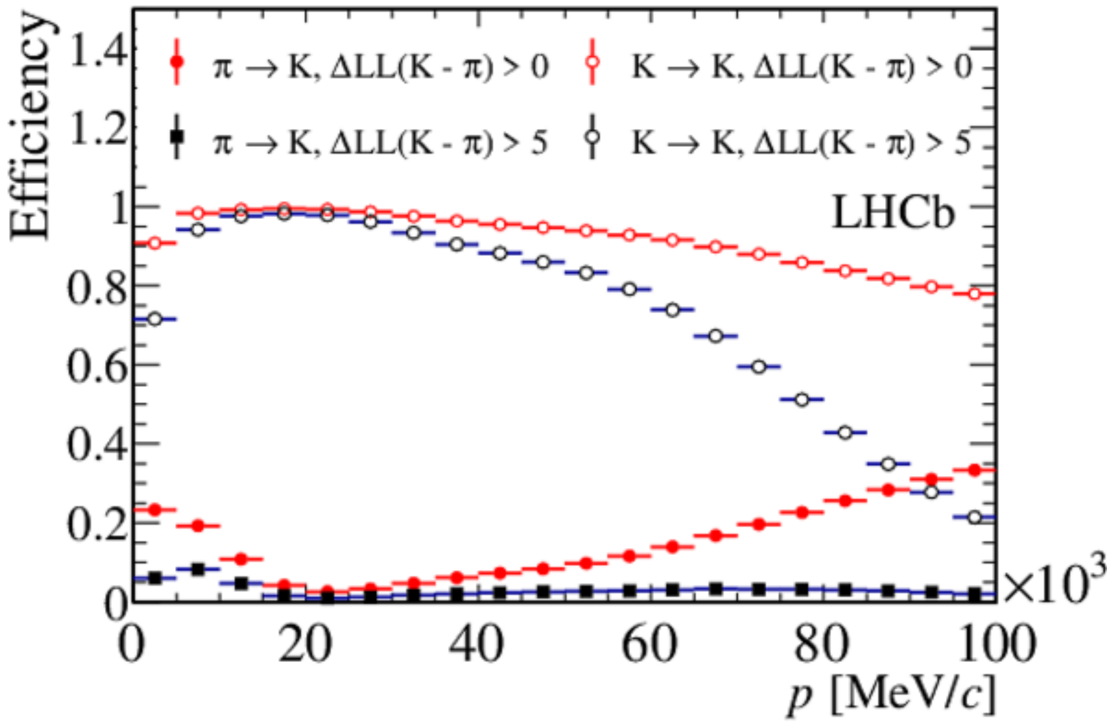


Implemented for the first time in Run 2 with offline like quality from very early in 2015. Not only tracker but also RICH and calorimeter. For me this is the most impressive aspect of LHCb's Run 2 and required a huge team effort across projects and working groups.

We also need to measure our efficiencies in real-time!

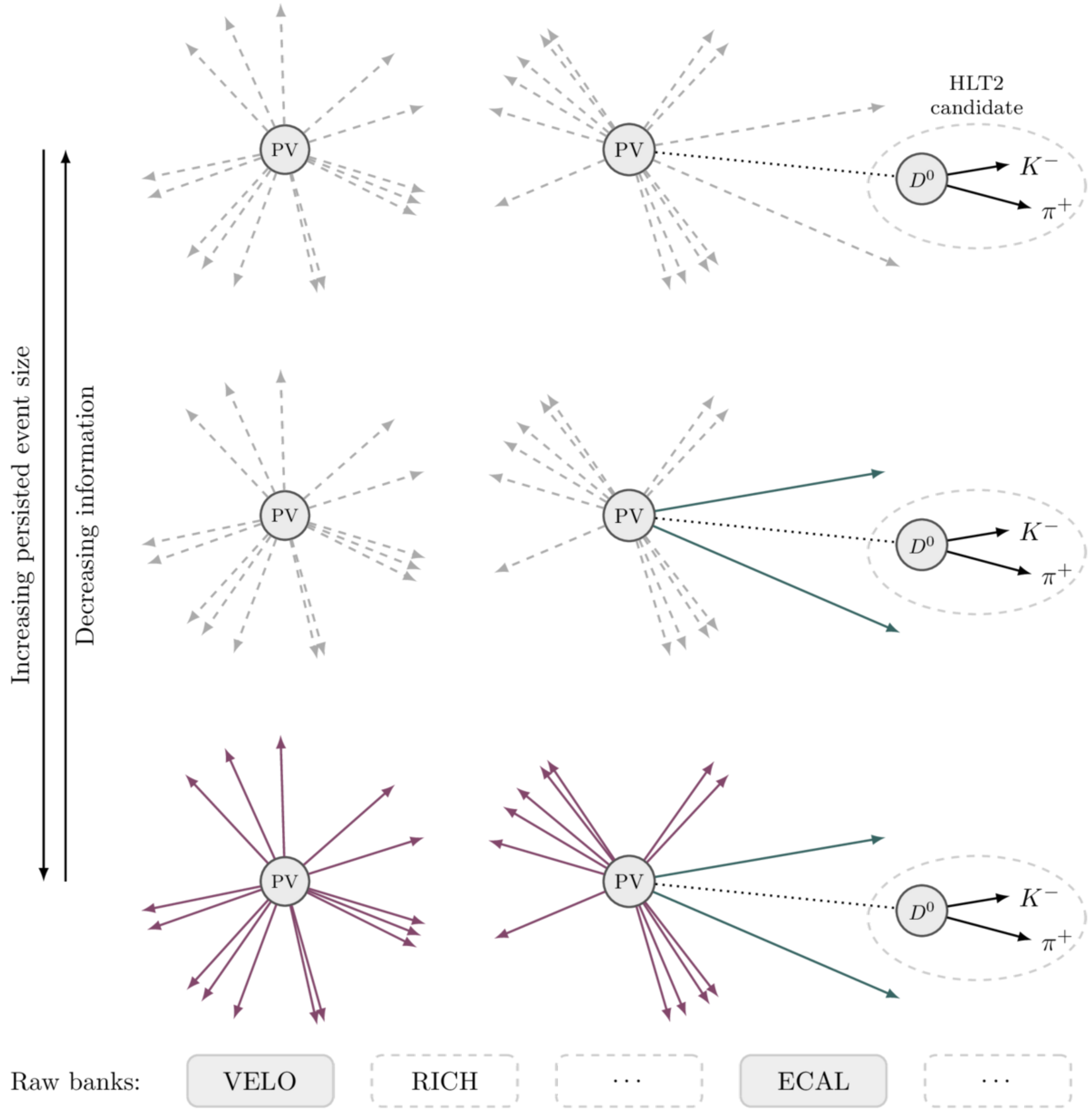
Species	Low momentum	High momentum
$e^\pm$	$B^+ \to J/\psi K^+$ with $J/\psi \to e^+e^-$	
$\mu^\pm$	$B^+ \to J/\psi K^+$ with $J/\psi \to \mu^+\mu^-$	$J/\psi \to \mu^+\mu^-$
$\pi^\pm$	$K^0_S \to \pi^+\pi^-$	$D^{*+} \to D^0\pi^+$ with $D^0 \to K^-\pi^+$
$K^\pm$	$D_s^+ \to \phi\pi^+$ with $\phi \to K^+K^-$	$D^{*+} \to D^0\pi^+$ with $D^0 \to K^-\pi^+$
p , $\overline{p}$	$\Lambda^0 \to p\pi^-$	$\Lambda^0 \to p\pi^-$; $\Lambda_c^+ \to pK^-\pi^+$





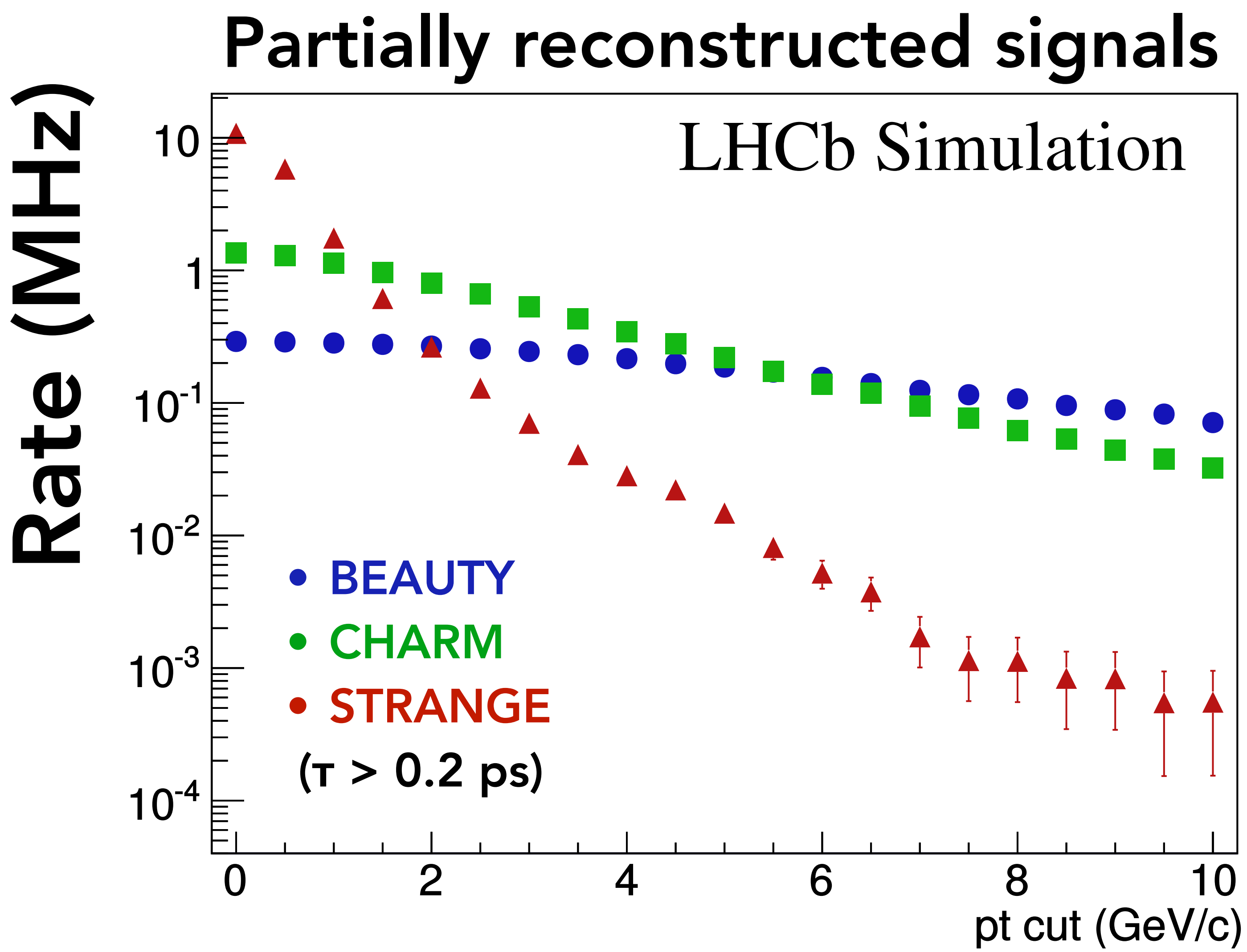
Unlike ATLAS and CMS, LHCb must maintain a data-driven permille level control of its efficiency across the kinematic and geometric acceptance of the detector. Requires collecting an extremely wide range of tag-and-probe samples in real time.

Then select signals and associate them to pp collisions



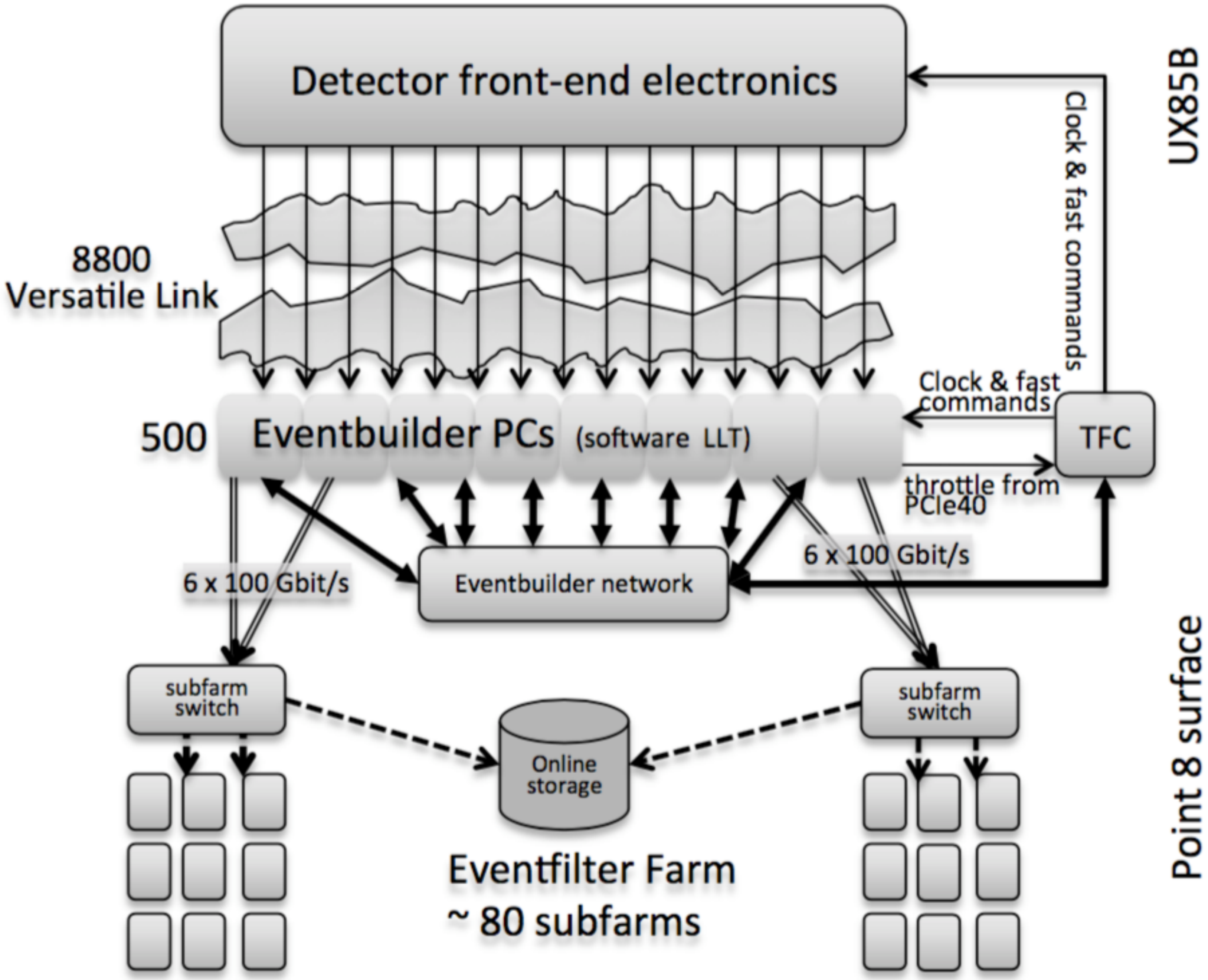
Full flexibility to store "additional" detector information if required by some analyses

So how will this evolve towards the LHCb upgrade?



We will have MHz of signals in our acceptance!
A traditional "inclusive" trigger would struggle to achieve 1/100 rejection efficiently.

The LHCb detector readout for the upgrade



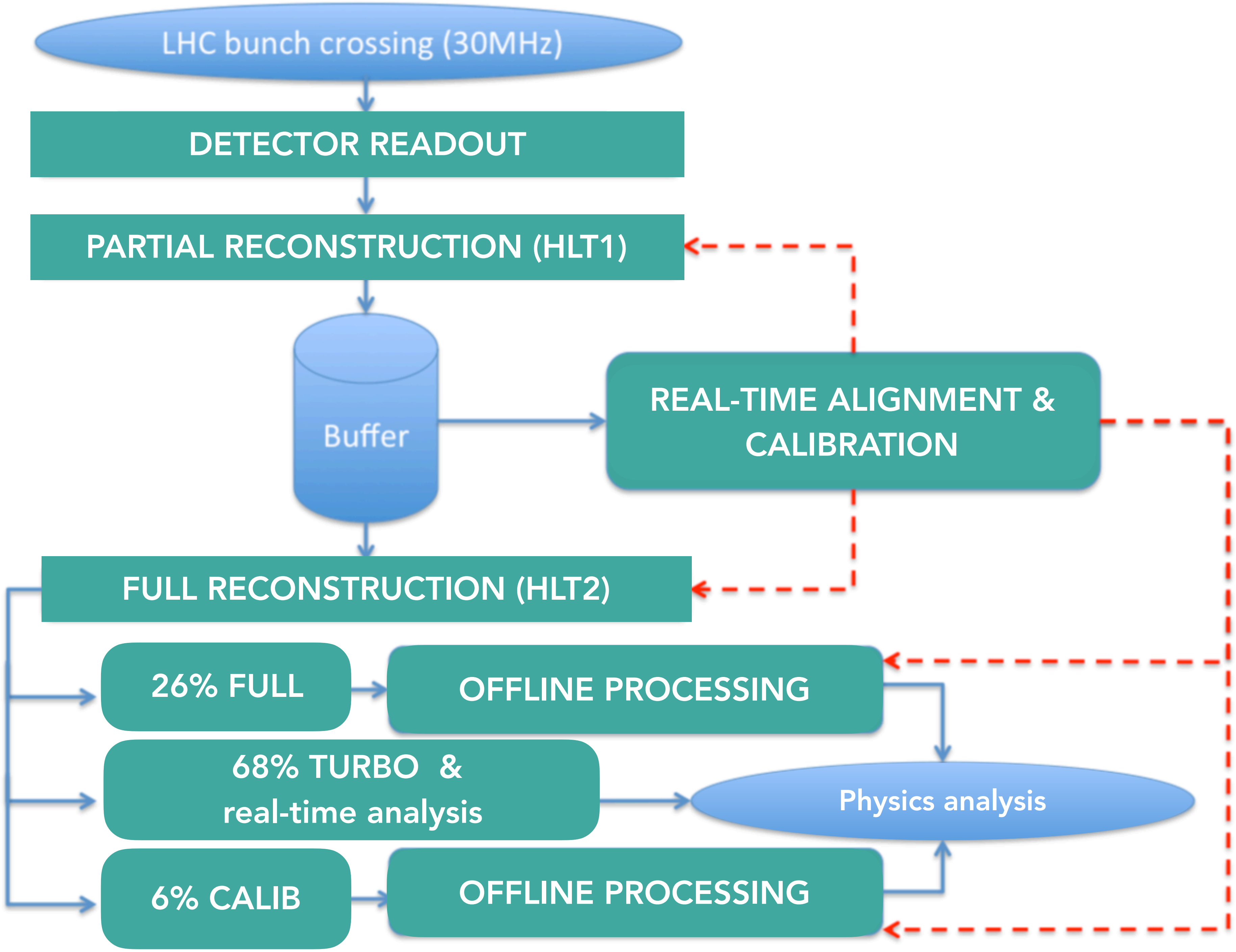
40 Tbit/second made available for processing in a data centre. This is an enormous challenge in itself

LHCb upgrade dataflow

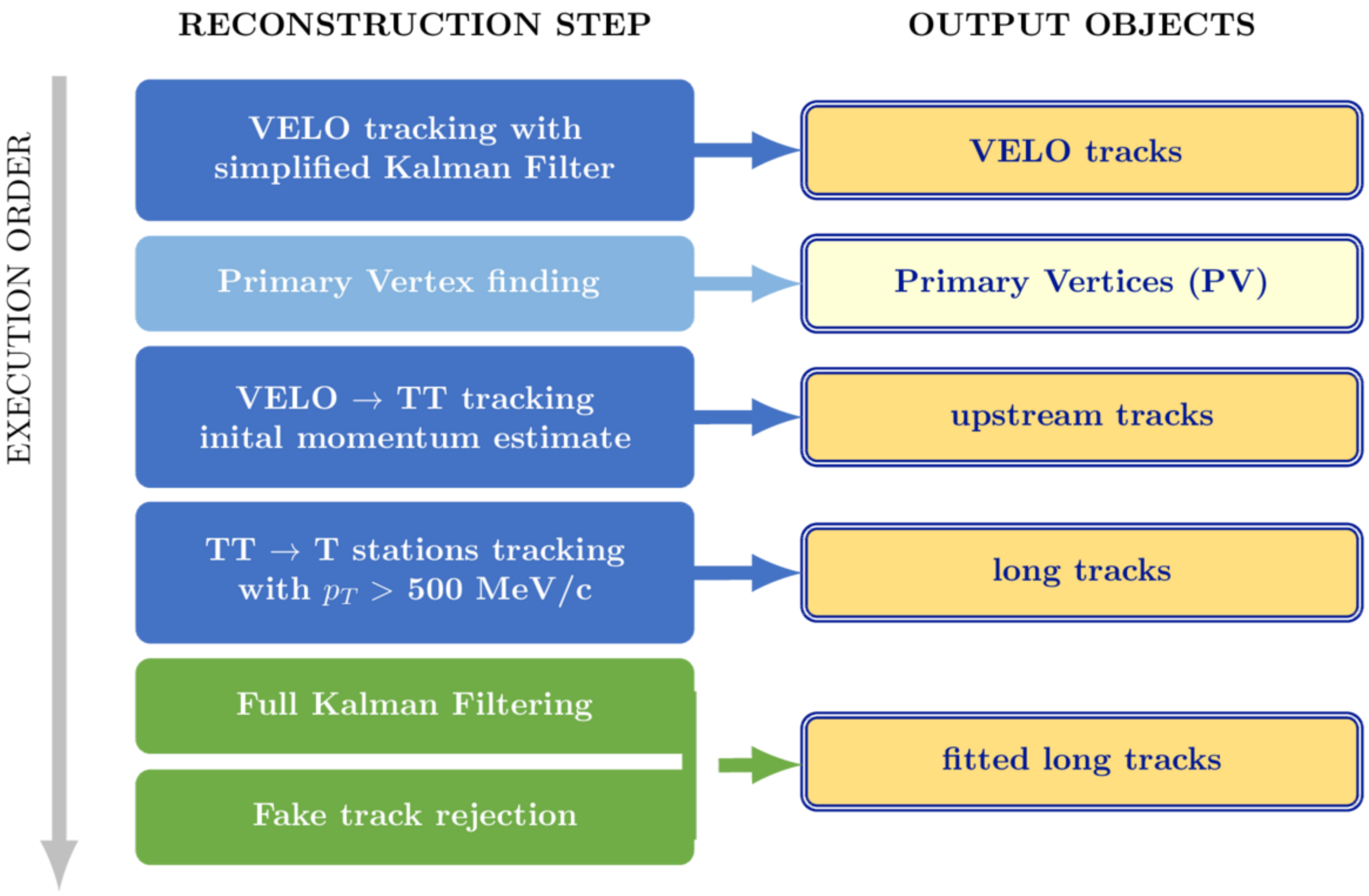
40 Tb/s
30 MHz non-empty pp

1-2 Tb/s
0.5 - 1.5 MHz

80 Gb/s



What is the physics content of HLT1?



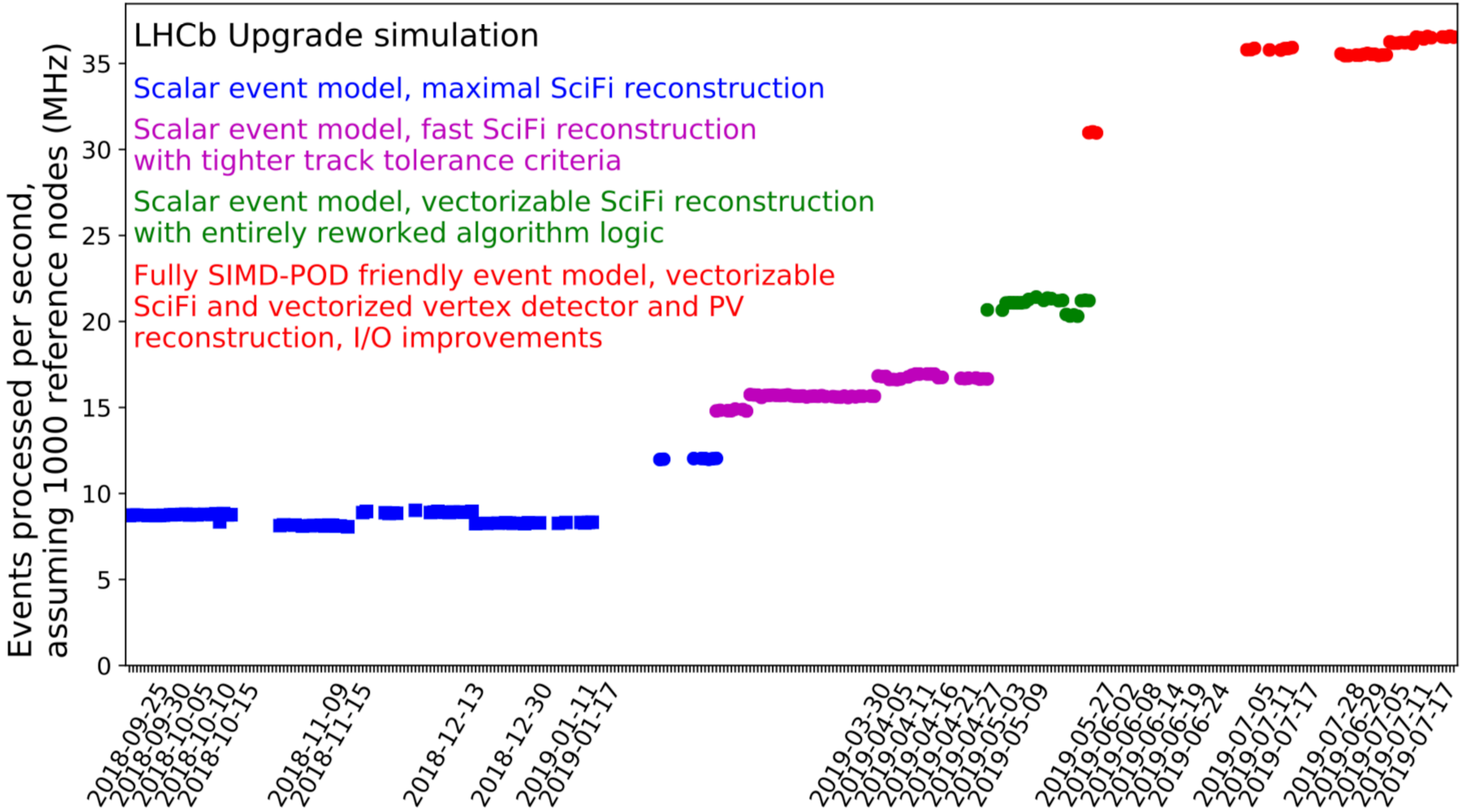
"Traditional" inclusive selections selecting bunch crossings.
Must be based on tracks, so require 30 MHz tracking at $2 \cdot 10^{33}$!

Pause and compare this to ATLAS/CMS HL-LHC processing

CMS detector	LHC Run-2	HL-LHC Phase-2	
Peak ⟨PU⟩	60	140	200
L1 accept rate (maximum)	100 kHz	500 kHz	750 kHz
Event Size	2.0 MB [a]	5.7 MB [b]	7.4 MB
Event Network throughput	1.6 Tb/s	23 Tb/s	44 Tb/s
Event Network buffer (60 seconds)	12 TB	171 TB	333 TB
HLT accept rate	1 kHz	5 kHz	7.5 kHz
HLT computing power [c]	0.5 MHS06	4.5 MHS06	9.2 MHS06
Storage throughput	2.5 GB/s	31 GB/s	61 GB/s
Storage capacity needed (1 day)	0.2 PB	2.7 PB	5.3 PB

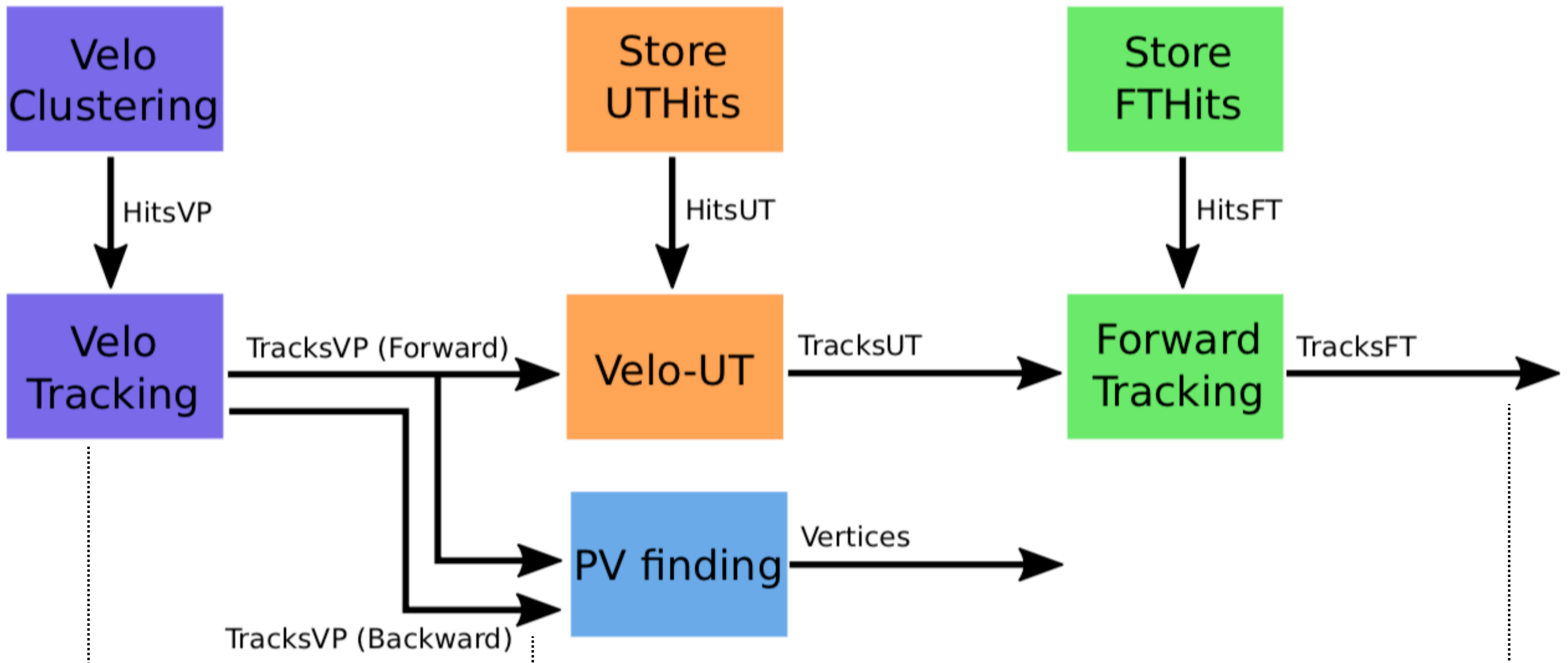
LHCb 2021 real-time tracking has to handle the same data volume as the GPD HL-LHC upgrades! Except earlier and for less money...

And we are there!

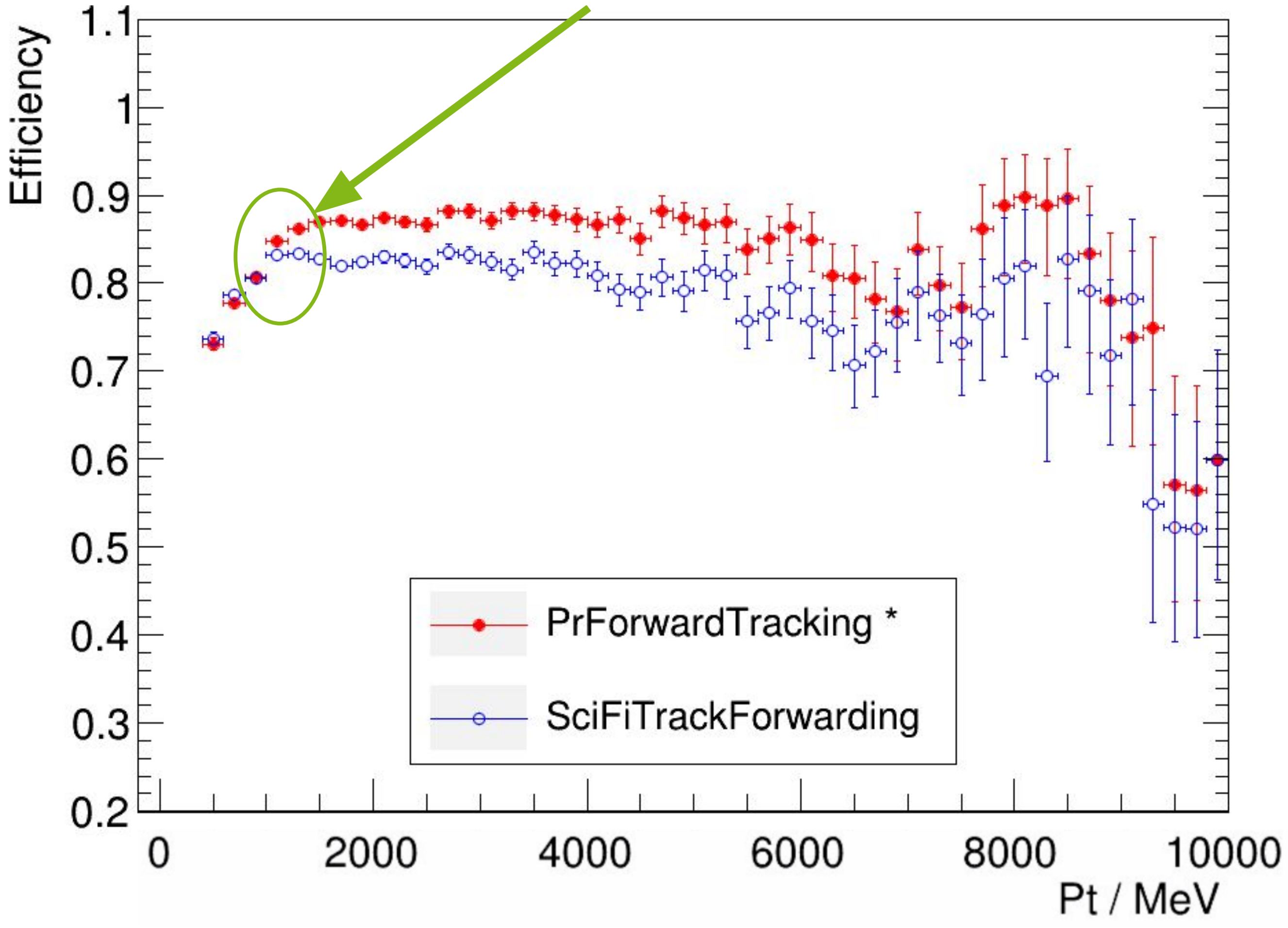


Required HLT1 throughput achieved in 2019!

How was this achieved?



Requires hits on both x-layers in last SciFi station, we are investigating relaxing this



Minimize data being passed within reco sequence so only algorithms which actually need e.g. backwards VELO tracks get them.

VELO clustering and tracking is now merged into a single vectorized algorithm, gained almost x2 in speed while improving physics performance over the “fast” scalar code.

Each tracking algorithm now outputs its own minimal (POD-style) track type as input to the next one. To be seen if these can be consolidated into a general POD-track without major loss of performance.

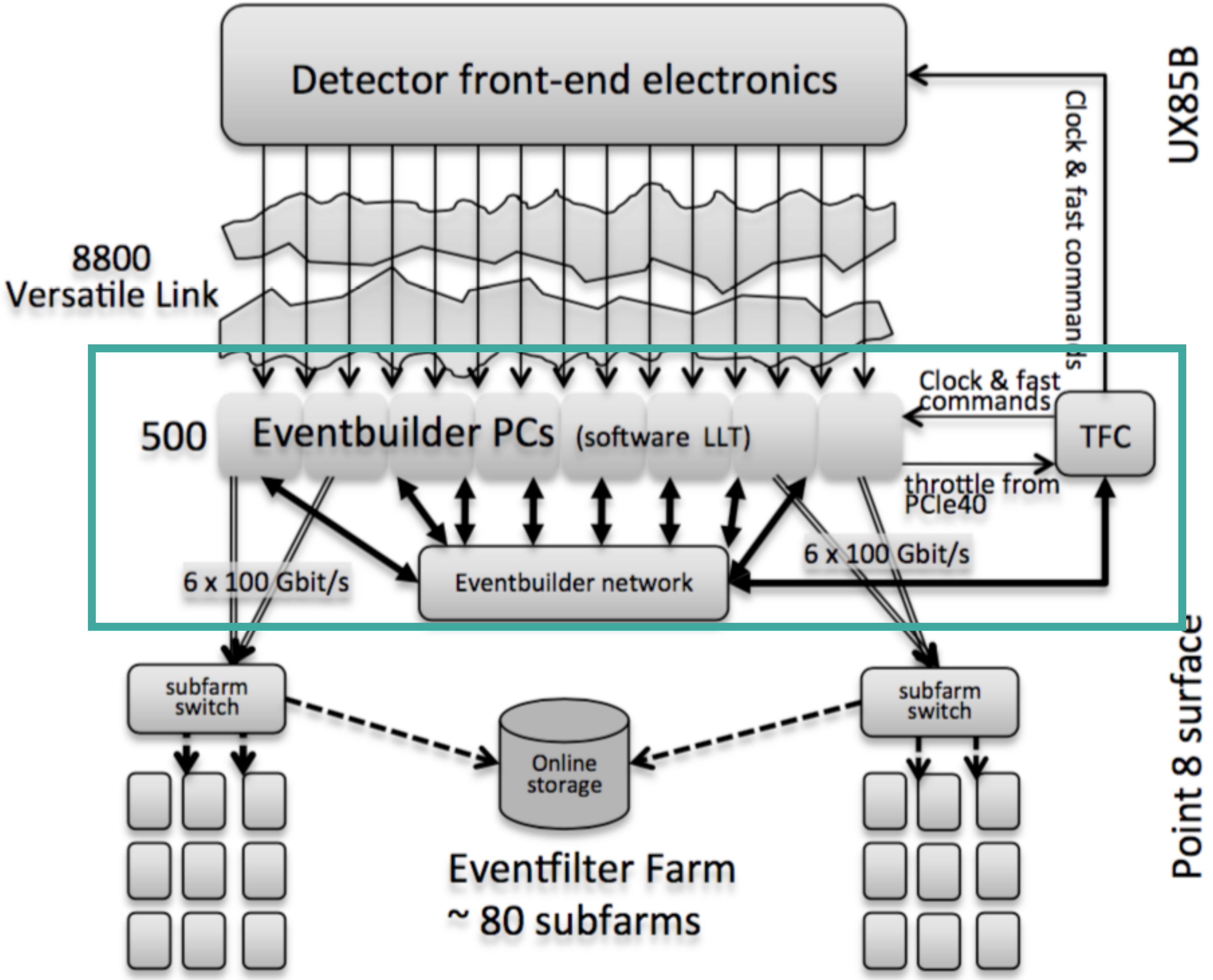
Combine new algorithms with much lighter data structures

And we also developed a GPU HLT1!

LHCb-ANA-20XX-YYY
May 31, 2019

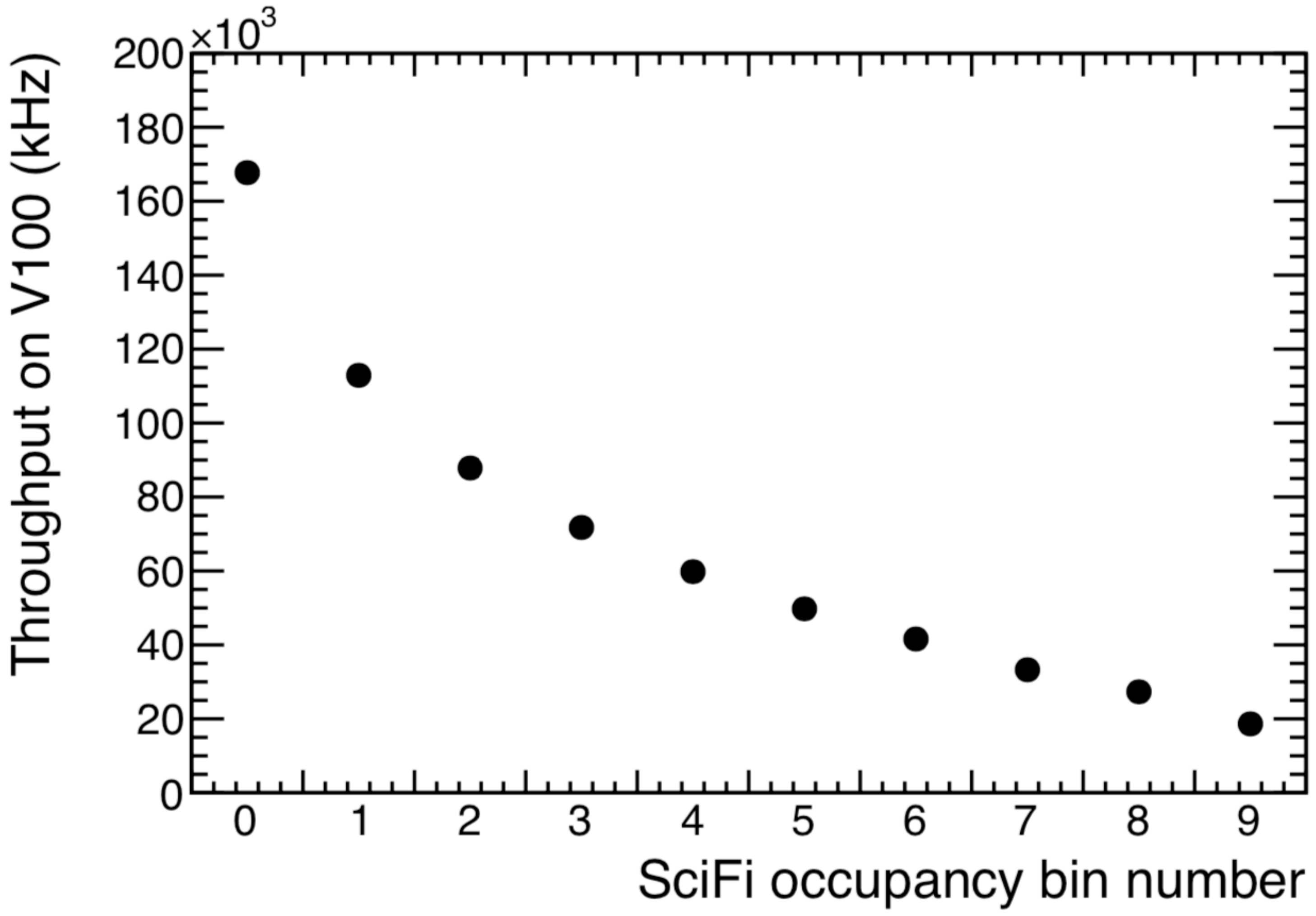
Proposal for an HLT1 implementation on GPUs for the LHCb experiment

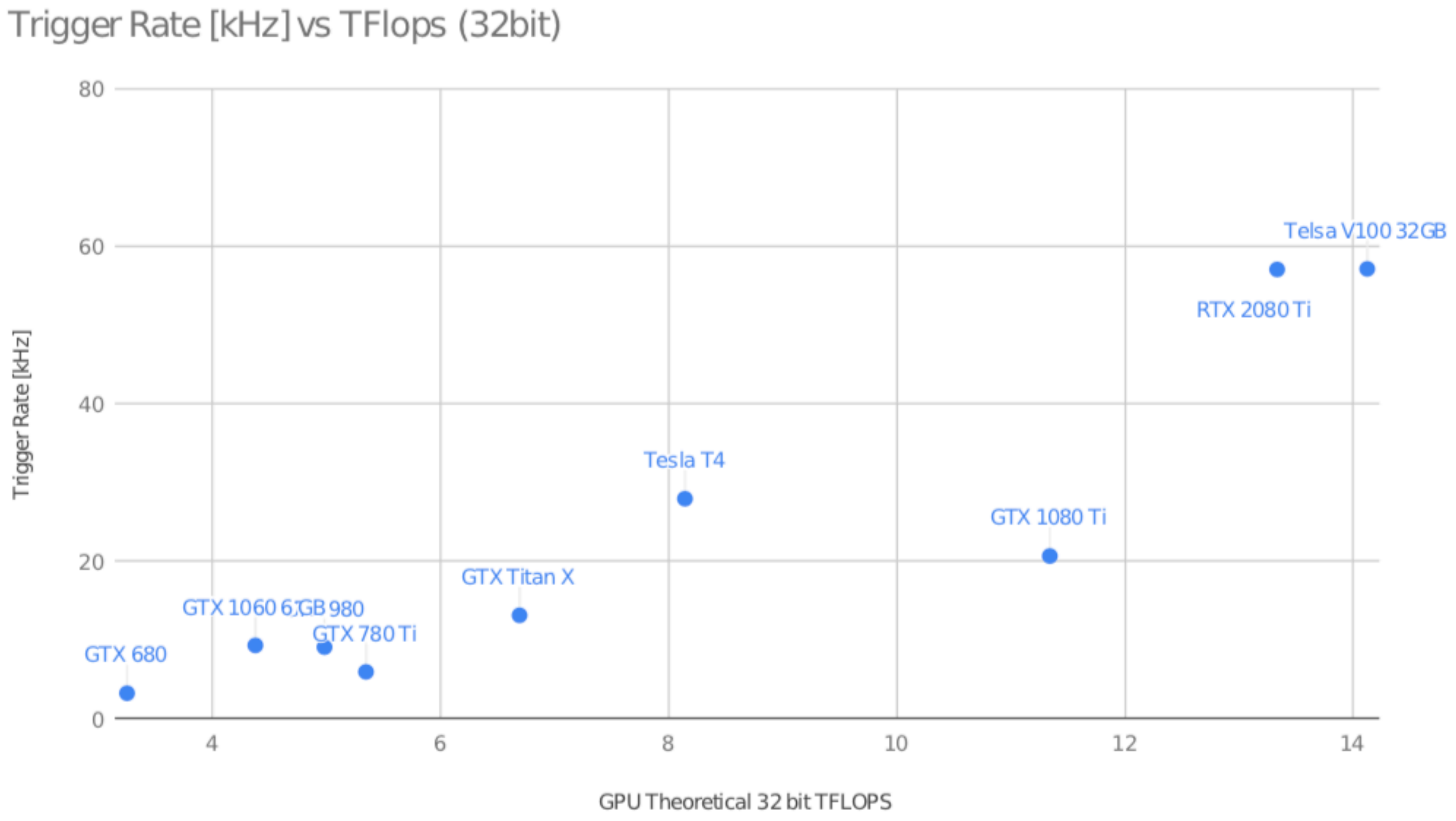
R. Aaij[1], J. Albrecht[2], M. Belous[a,3], T. Boettcher[4], A. Brea Rodríguez[5], D. vom Bruch[6], D. H. Cámpora Pérez[b,7], A. Casais Vidal[5], P. Fernandez Declara[c,7], L. Funke[2], V. V. Gligorov[6], B. Jashal[9], N. Kazeev[a,3], D. Martínez Santos[5], F. Pisani[d,e,7], D. Pliushchenko[f,3], S. Popov[a,3], M. Rangel[10], F. Reiss[6], C. Sánchez Mayordomo[9], R. Schwemmer[7], M. Sokoloff[11], A. Ustyuzhanin[a,3], X. Vilasís-Cardona[8], M. Williams[4]



Exploits flexibility of our Run 3 DAQ by implementing HLT1 directly in the servers receiving the data from the detector. Judged viable by external review, full cost-benefit analysis ongoing to decide if we will use this already in Run 3.

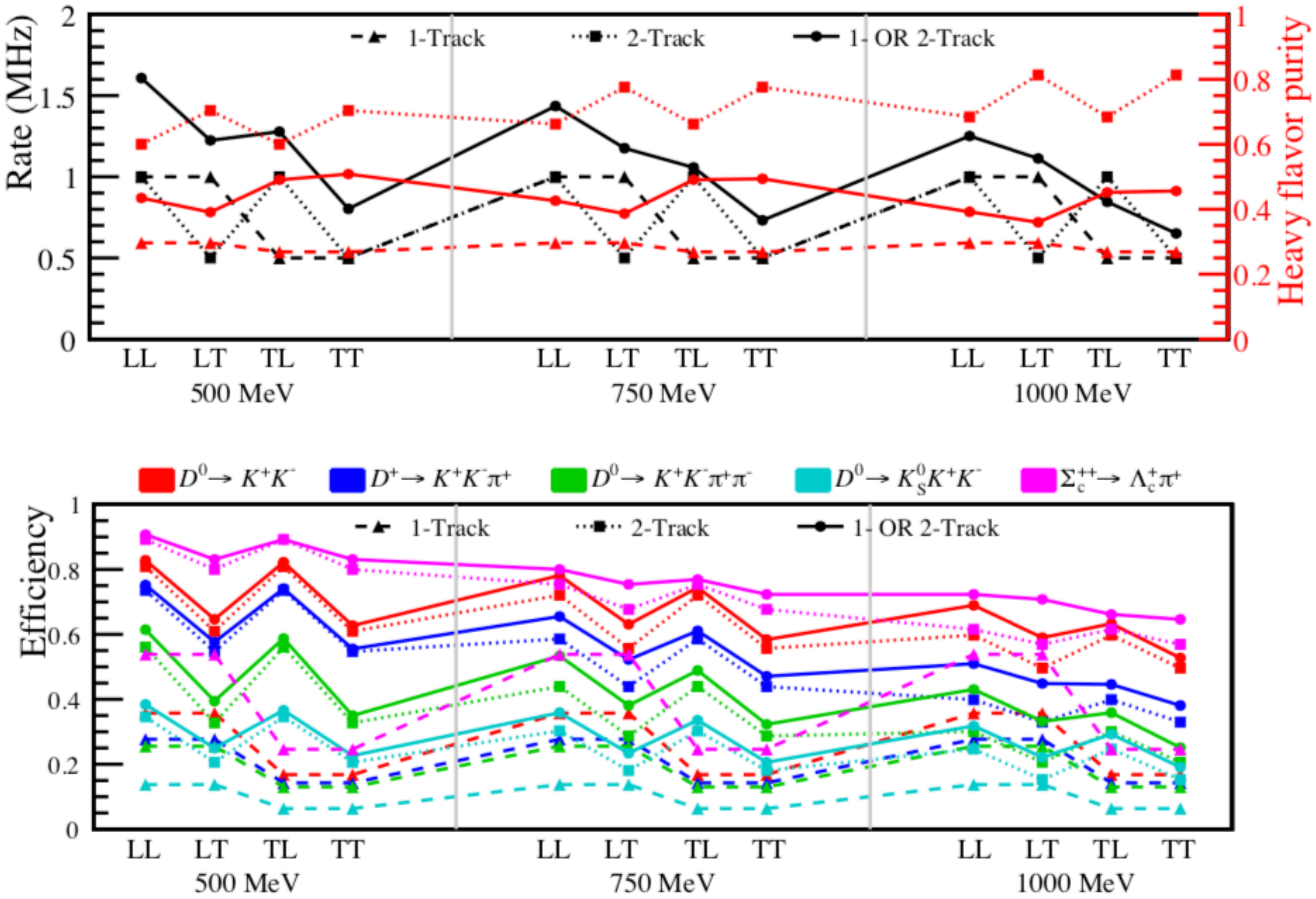
A brief look at the GPU HLT1 performance





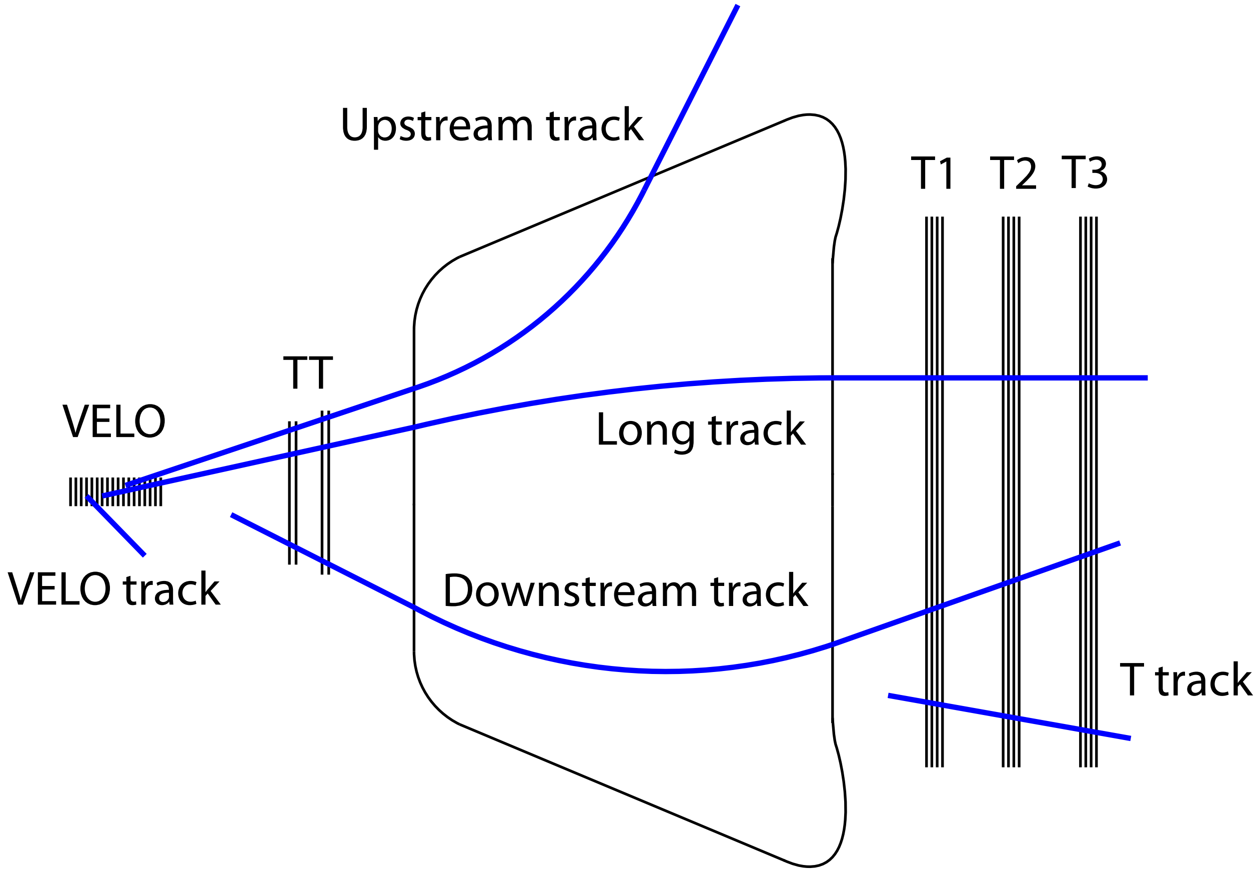
Linear scaling of throughput vs. occupancy, and throughput vs. the theoretical TFLOPS of each card. Optimal use of hardware!

So what kinds of HLT1 efficiencies can we expect?



Optimize for charm as beauty is easier. Efficiencies depend on what HLT2 can consume — ideally 1 MHz, 500 kHz is tolerable

So what is the content of HLT2?



Tracker : chaged particle reconstruction

↓

Particle identification : RICH, Muon, ECAL

↓

Neutral reconstruction : ECAL

Adds tracking for particles produced outside the vertex detector, low p_T particles, CALO reconstruction & full particle identification.

What about the current HLT2 performance?

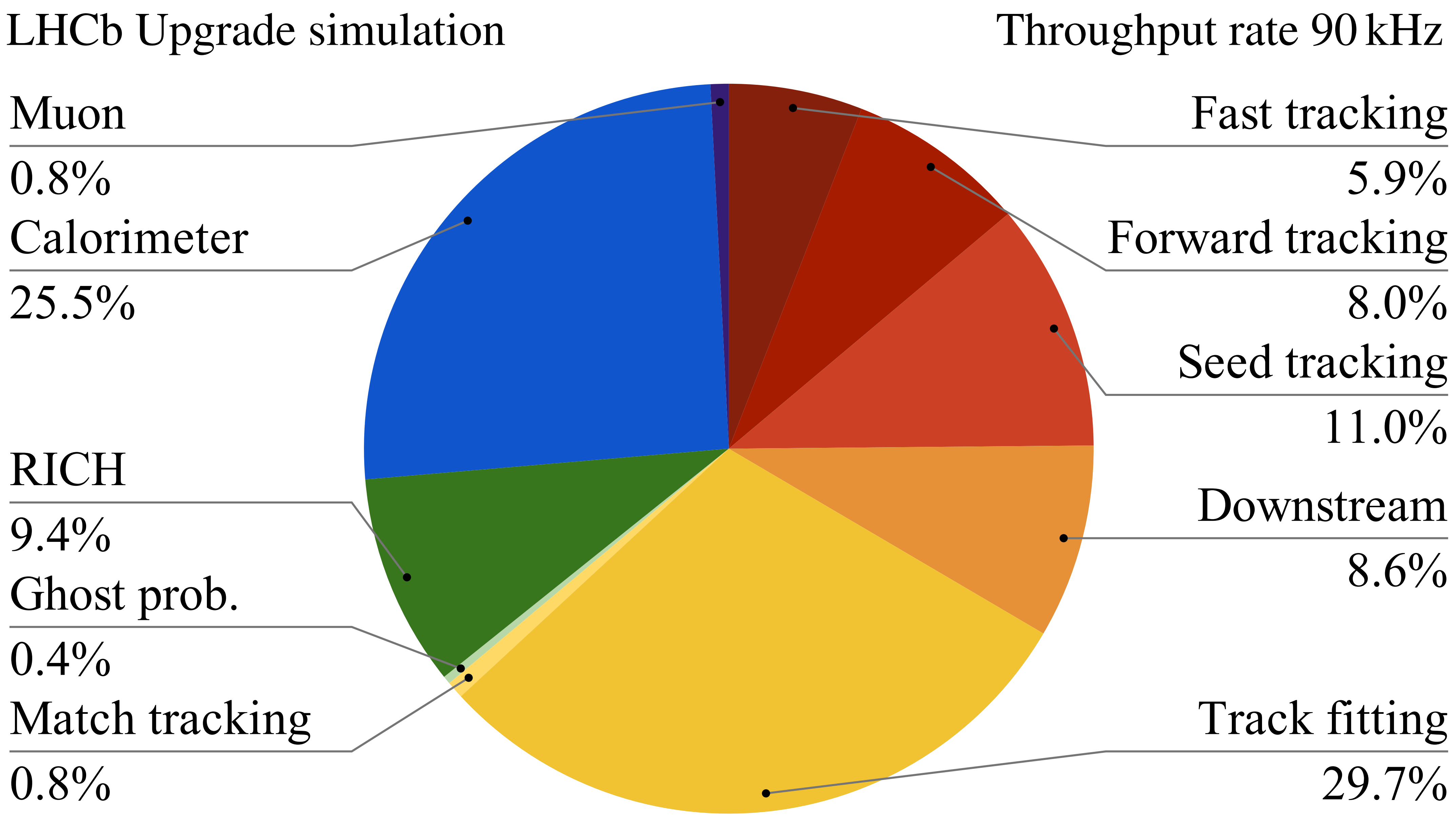
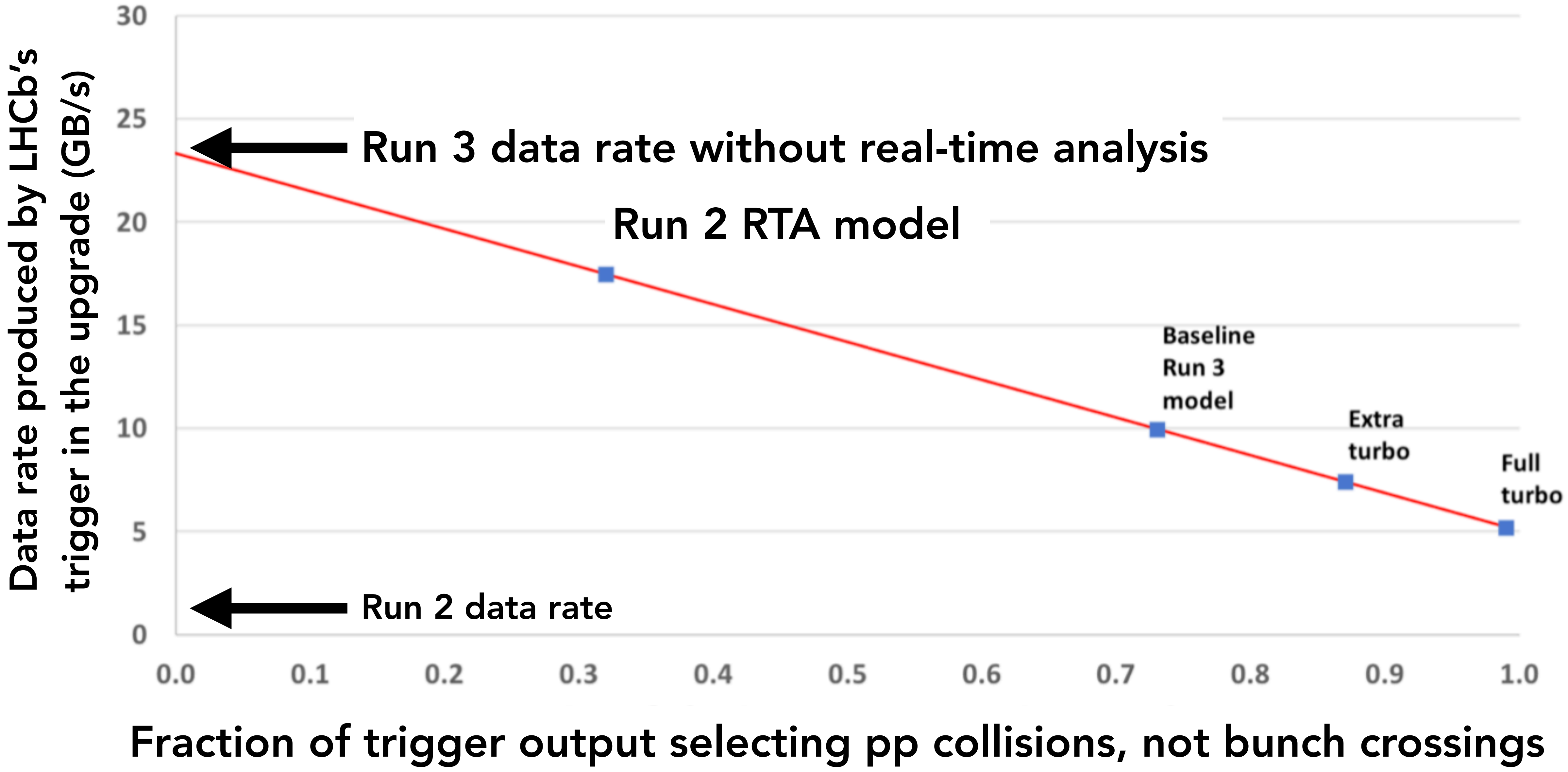


Figure 1: Breakdown of the current Hlt 2 reconstruction throughput rate for the LHCb upgrade.

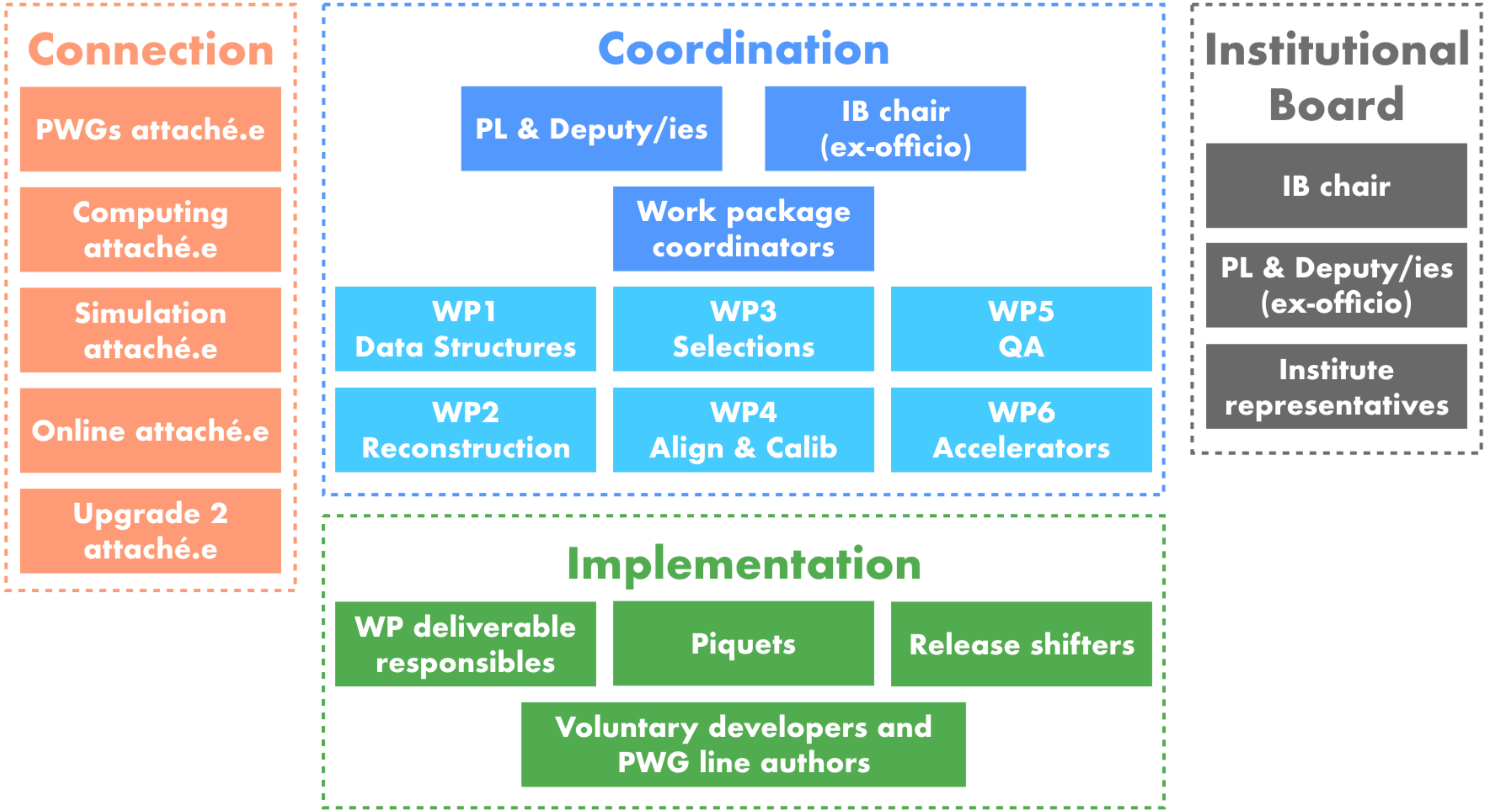
Quite far from the requirement but we have good hope to use what we learned improving HLT1 to improve here as well. Must improve all algorithms!

Coming back to what this does to our data rates



Real-time analysis reduces required resources by more than 2!

So why does all this require a “project”?



Organisation assures quality, reliability, maintainability

Remember: real-time analysis means that for most of our physics, we are now discarding the raw data within a bunch crossing in real-time.

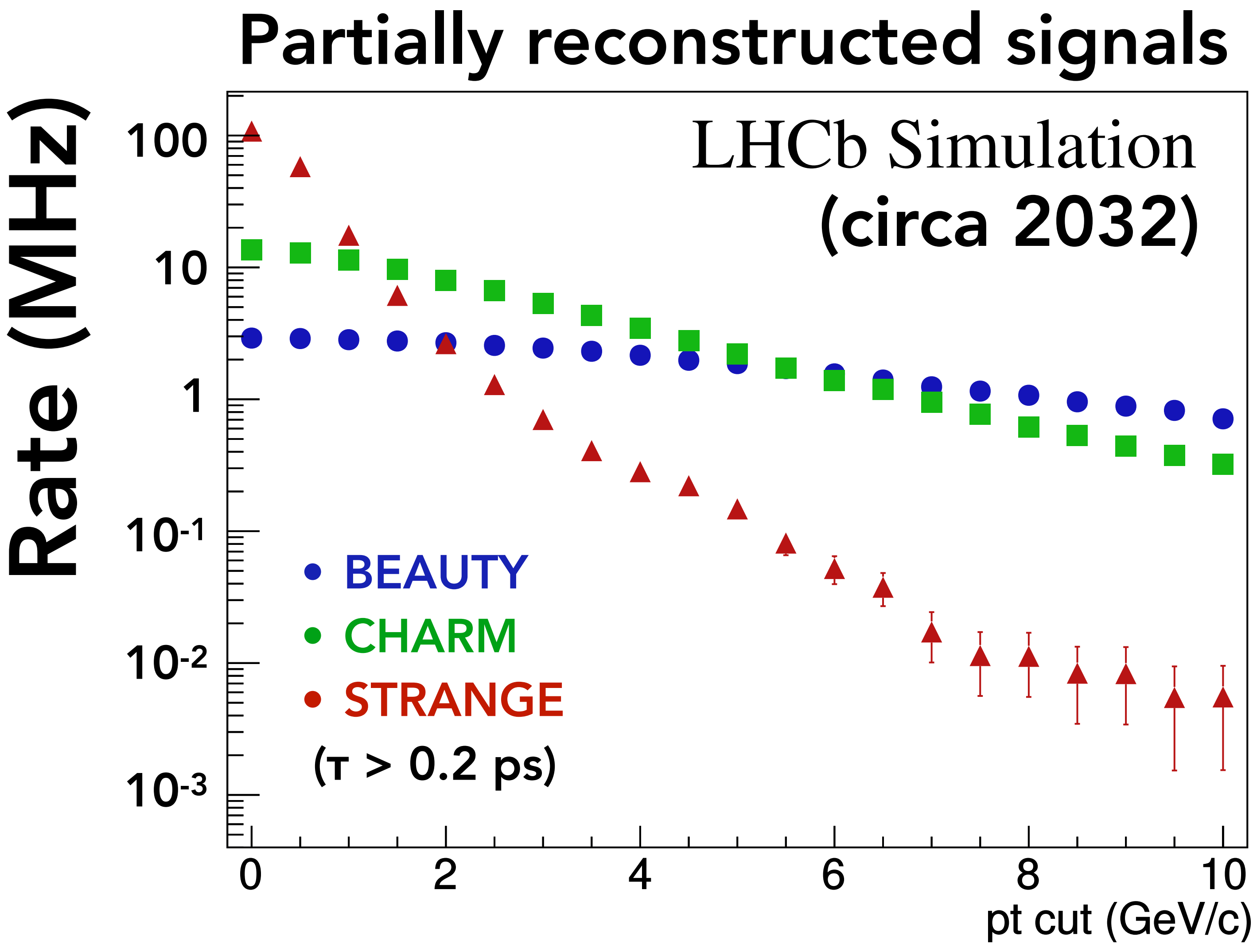
If you make a mistake you cannot redo it. Your software just became more sensitive to errors in real-time processing than any hardware detector!

But we have a strong culture in high-energy physics (especially LHCb) that software is mostly written by students and postdocs with support from a small staff of permanents at CERN. This is simply not sustainable in the upgrade era, much less beyond it.

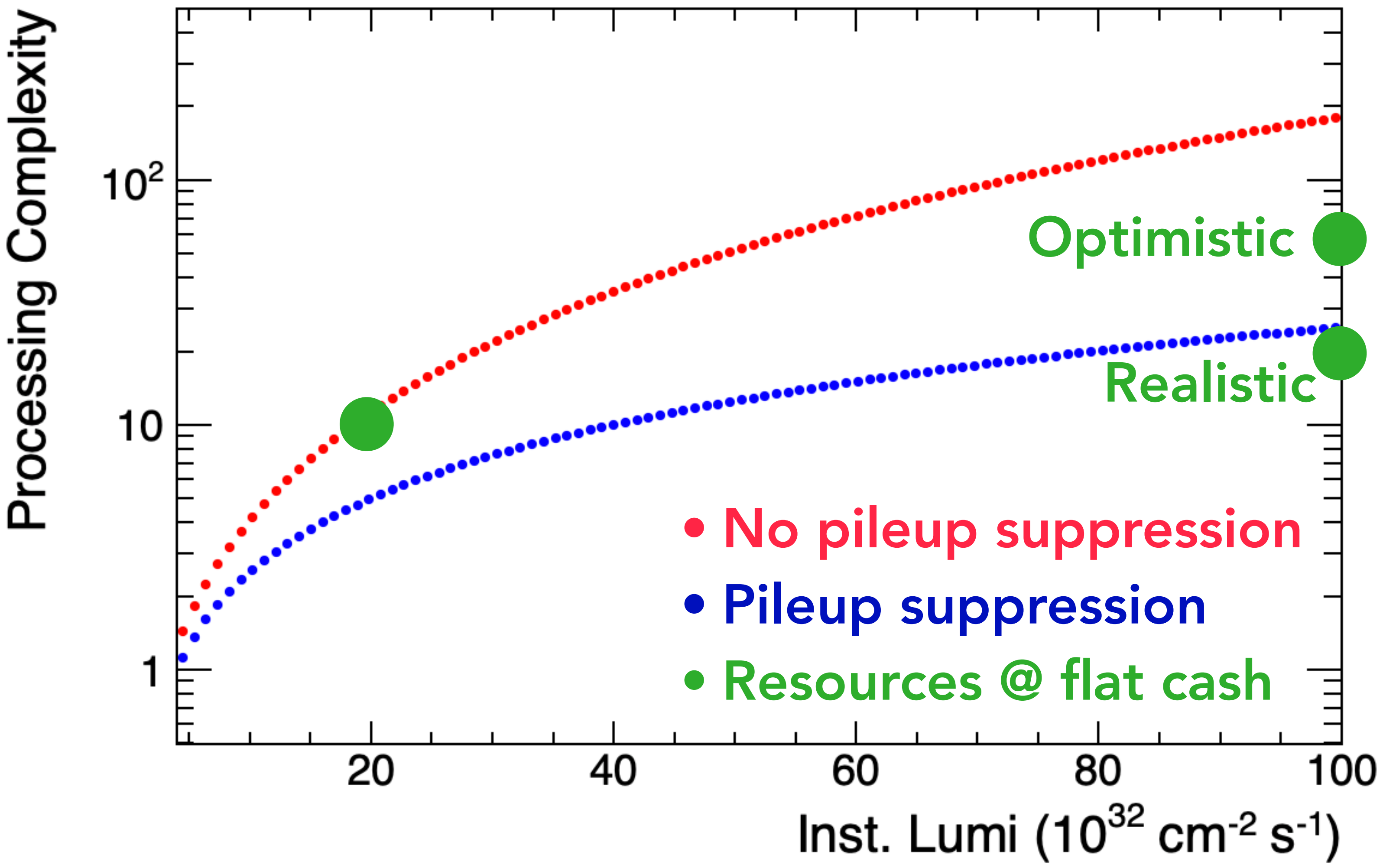
Real-time analysis has been organised as a project to try and see if we can manage software in the same way that we manage hardware, and secure stable positions at member institutes for software engineers and computer scientists to assure the long term maintenance and quality of our "software detector".

The structure is as important as the technical goal!

Looking beyond to a potential second LHCb upgrade



Fine print: this plot assumes that processing complexity goes linearly with detector occupancy, which is in itself an optimistic assumption before we even get to the pileup suppression part!



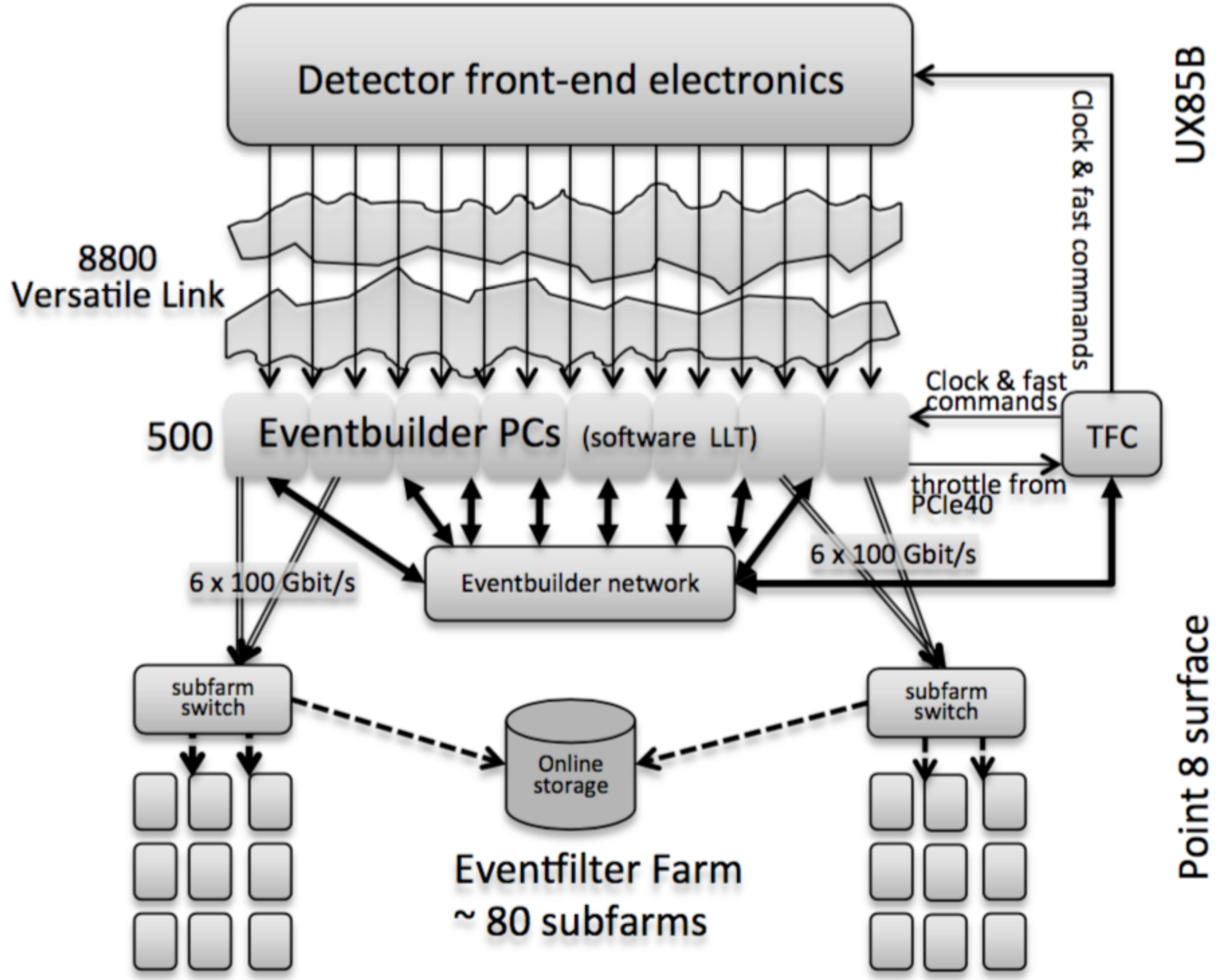
How to suppress pileup with O(60) pp collisions per bunch crossing?

Maintaining the flexibility of our processing will be crucial

GBT link : 4.8 Gb/s Upgrade I
Assume evolution to 10 Gb/s for HL-LHC using aggressive error handling : missing factor 5 compared to data rate growth.

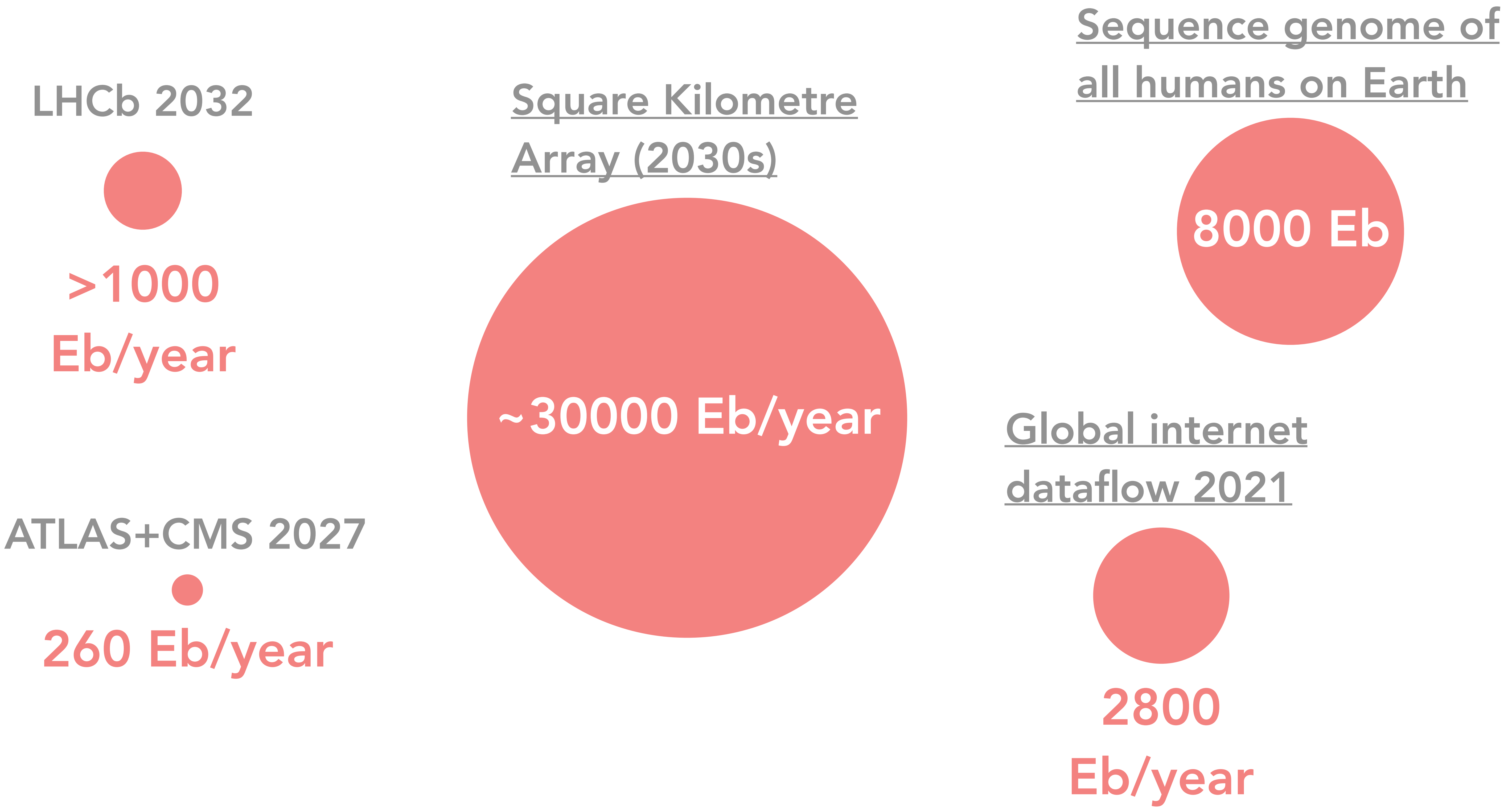
Event-building : current network is 500 servers with 100 Gb/s links. 200 Gb/s readily available, keep an eye on price/performance scaling beyond this?

Farm : carry out R&D in next years on optimal use of hybrid architectures (GPU/CPU/FPGA), remain flexible



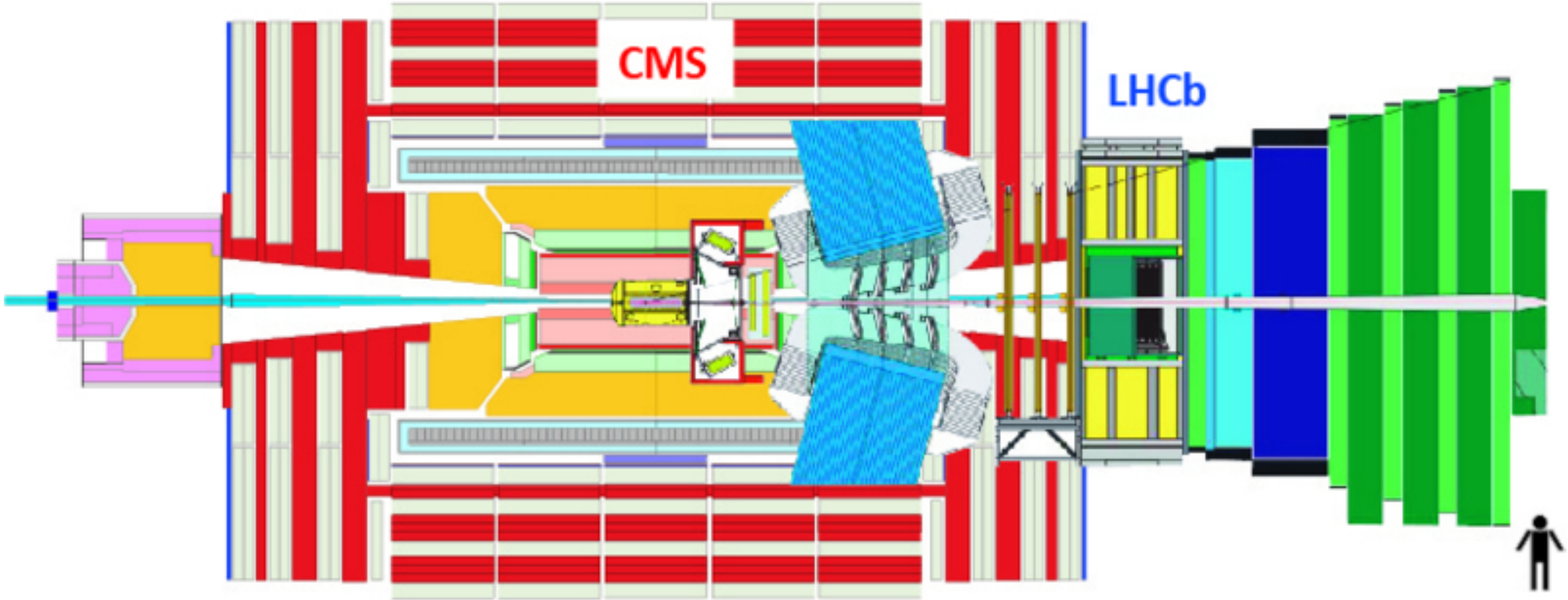
We now have two viable HLT1 models, on x86 and on GPU, already for Run 3! Ability to exploit hybrid architectures crucial to maximize physics/Euro in the long term.

Conclusions and final thoughts



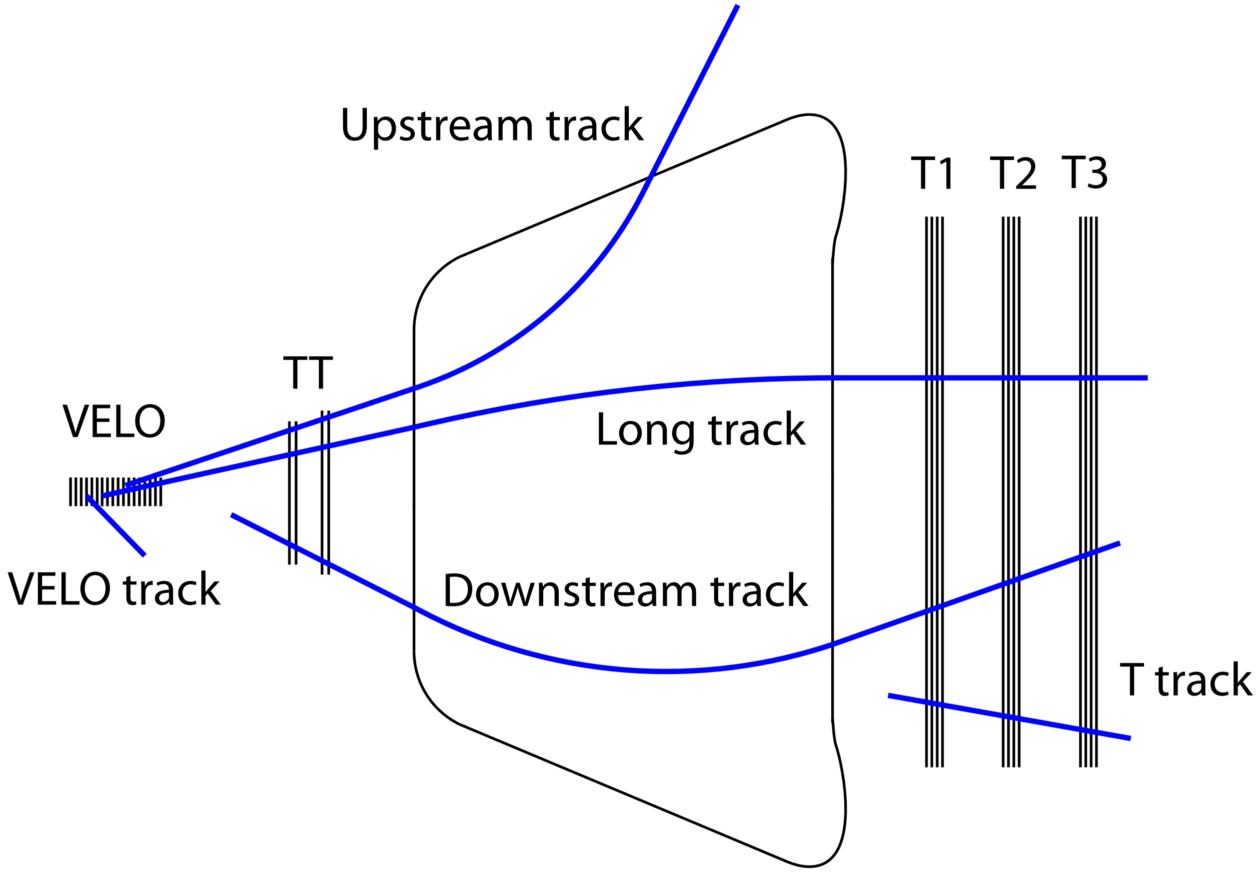
Backup

The LHCb detector at the LHC



Forward spectrometer optimized for precision physics

Reconstruction philosophy and role of subdetectors



Tracker : chaged particle reconstruction

↓

Particle identification : RICH, Muon, ECAL

↓

Neutral reconstruction : ECAL

Optimized for charged particles w/some neutral capability

LHCb analysis methodology and role of calibration samples

Trigger Efficiency

Tag-and-probe calibration method exists & widely used

Tracking efficiency

Tag-and-probe

Existing	Developing
μ	e, π, K, p

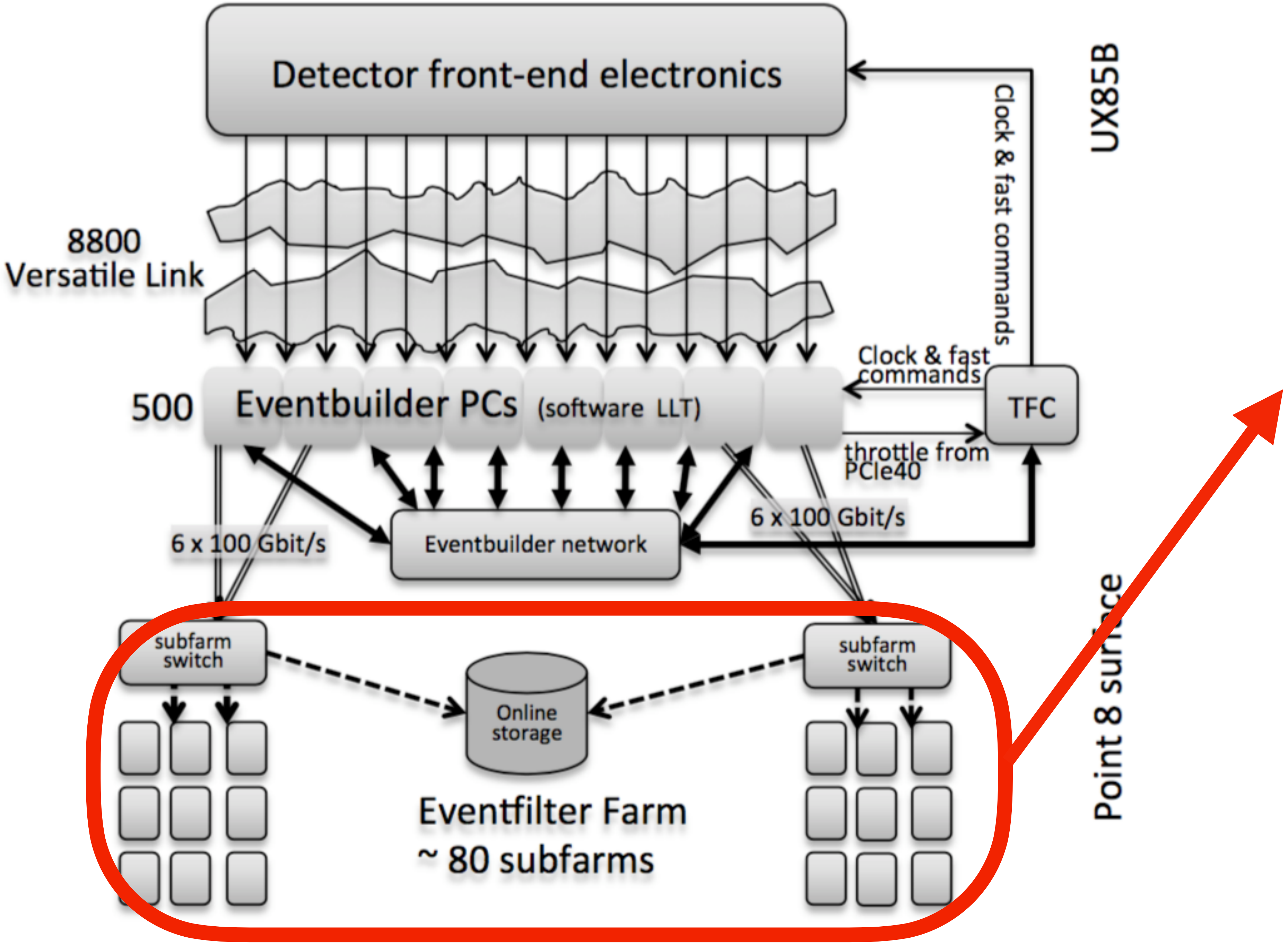
Particle identification

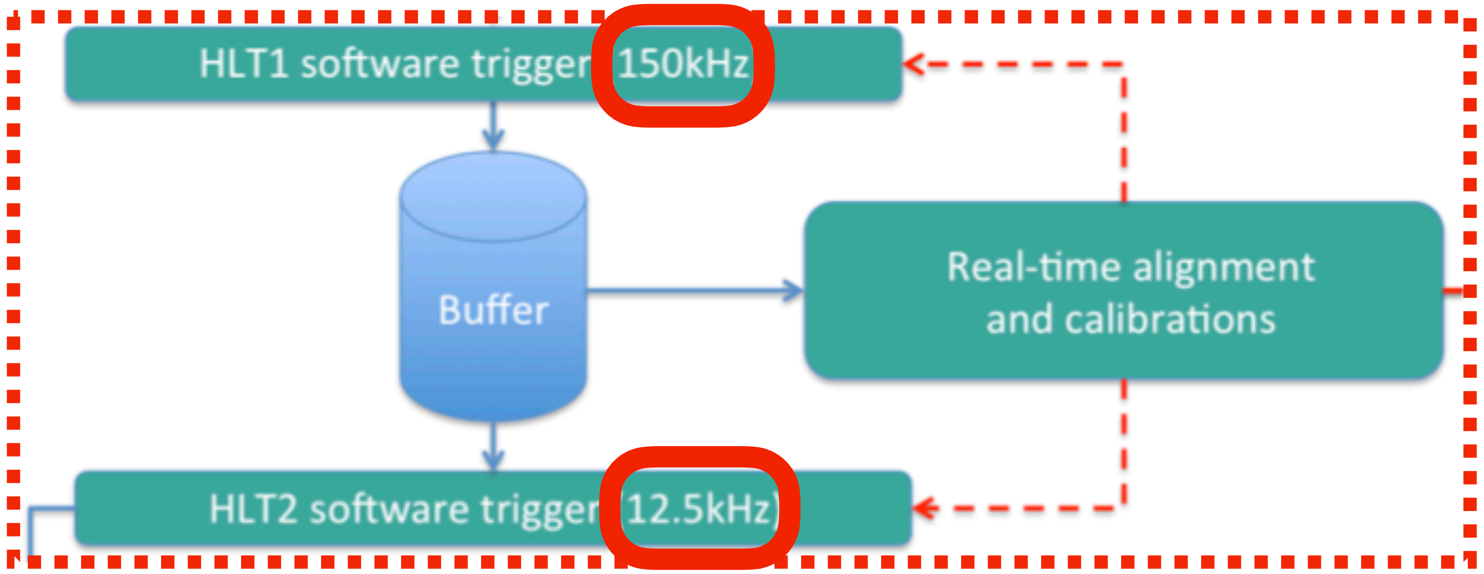
Tag-and-probe

Tag-and-probe calibrations exist for all charged particle species and for π^0/γ, with new sources added over time to improve coverage

Data driven efficiency calibration key to precision physics

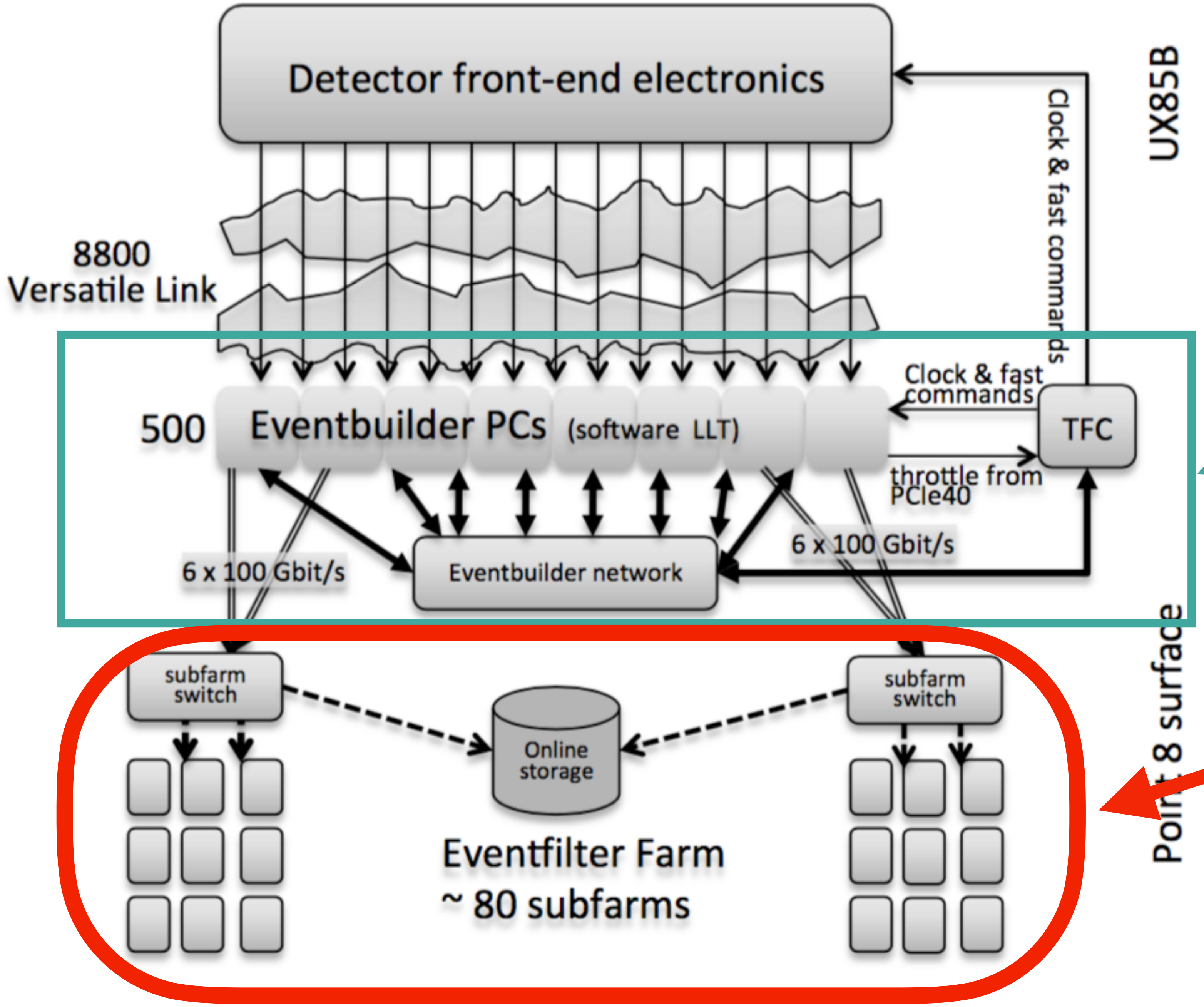
Looking inside the eventfilter farm

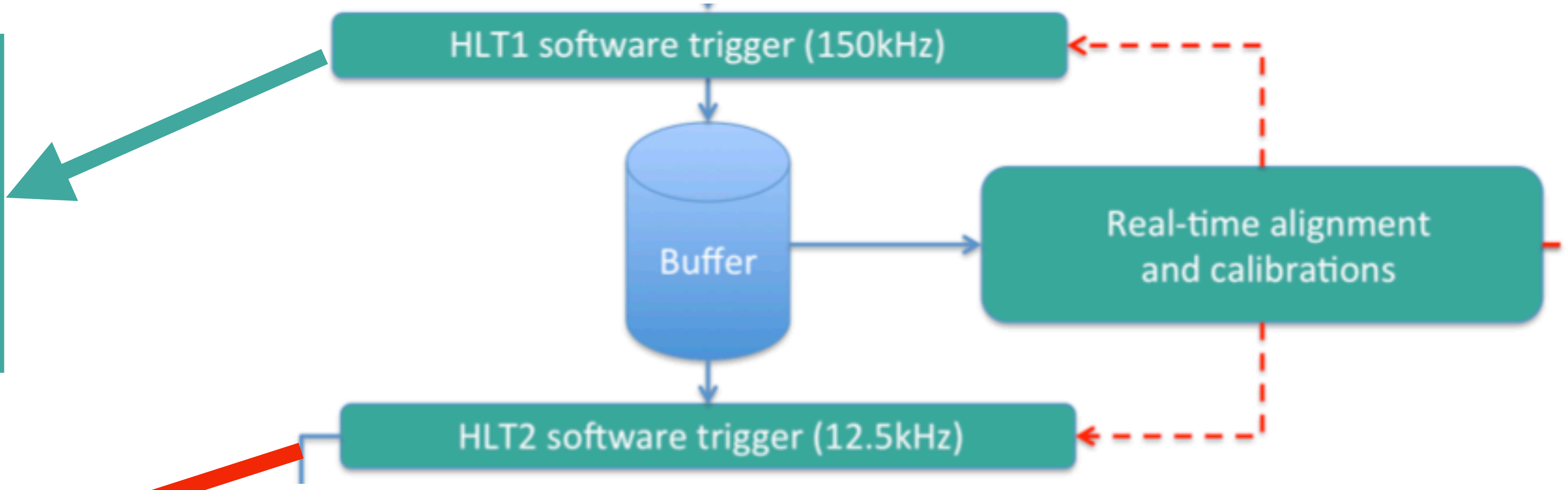




Splitting the HLT — example of a cascade buffer

But we should do a global DAQ optimization





Consider whole system: if coprocessors in event building network reduce event rate by O(10), greatly reduce cost of the network. Also reduces communication cost between x86 and coprocessor, since data goes directly to the coprocessor.

What is a cascade buffer?

Reconstruct high P_T leptons

Reconstruct pp vertices & select displaced leptons

Reconstruct other charged particles & build B candidate

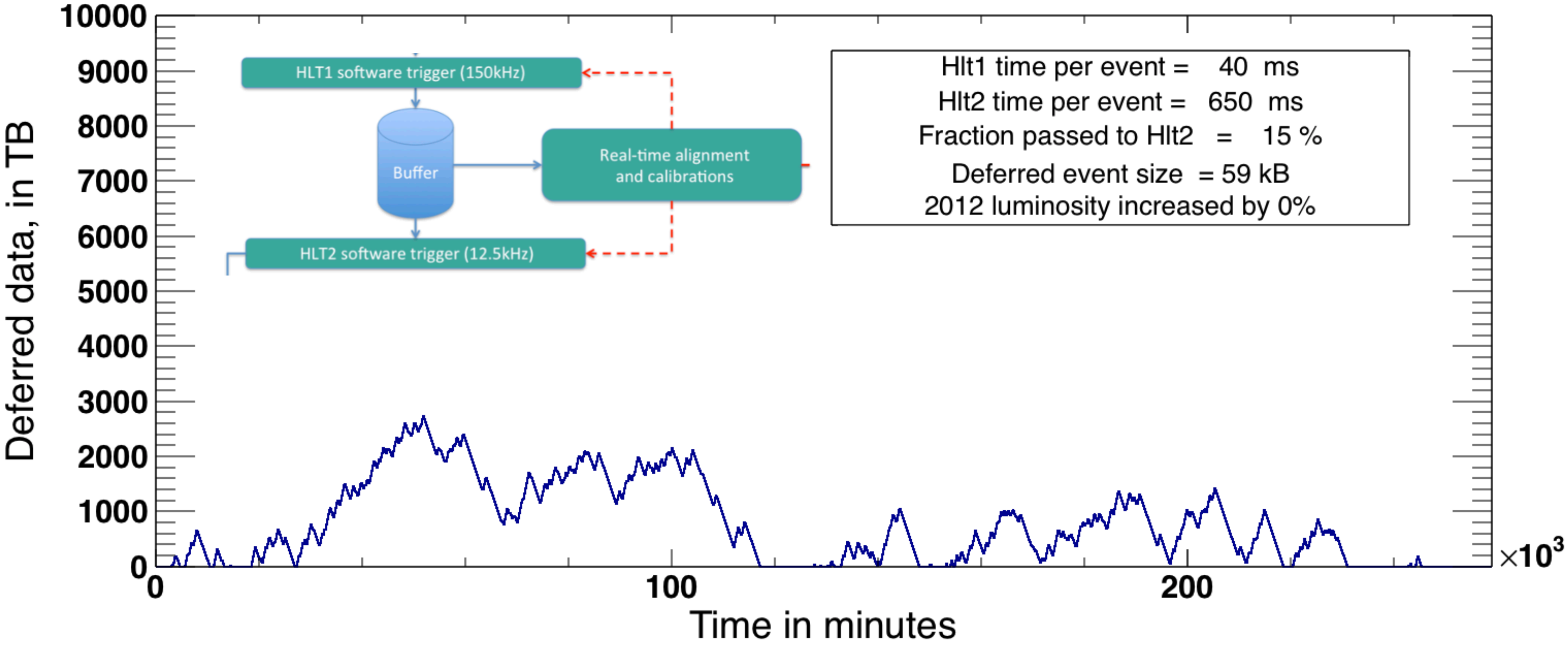
Build particle identification information & purify selection

Bigger data volume

More complex processing

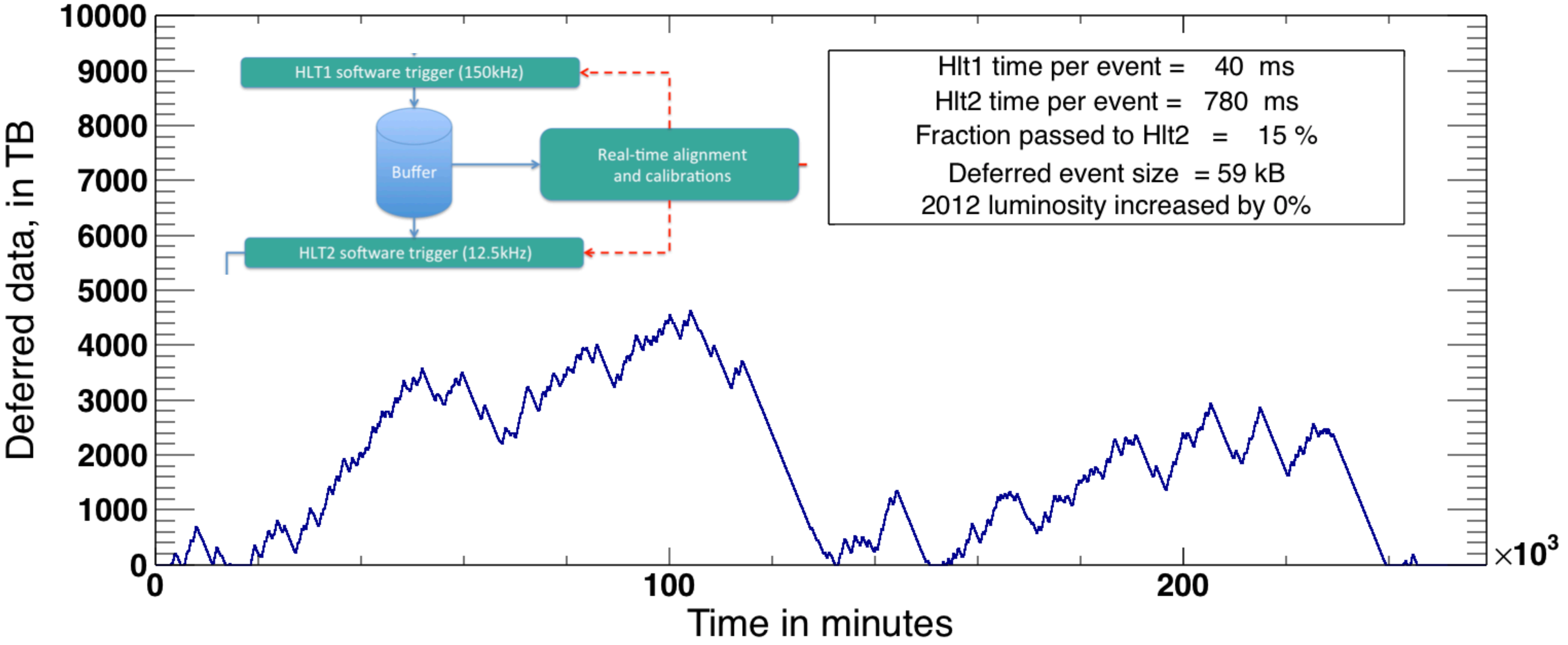
A staged data reduction using increasingly complex algorithms

Optimization of the Run 2 LHCb cascade buffer



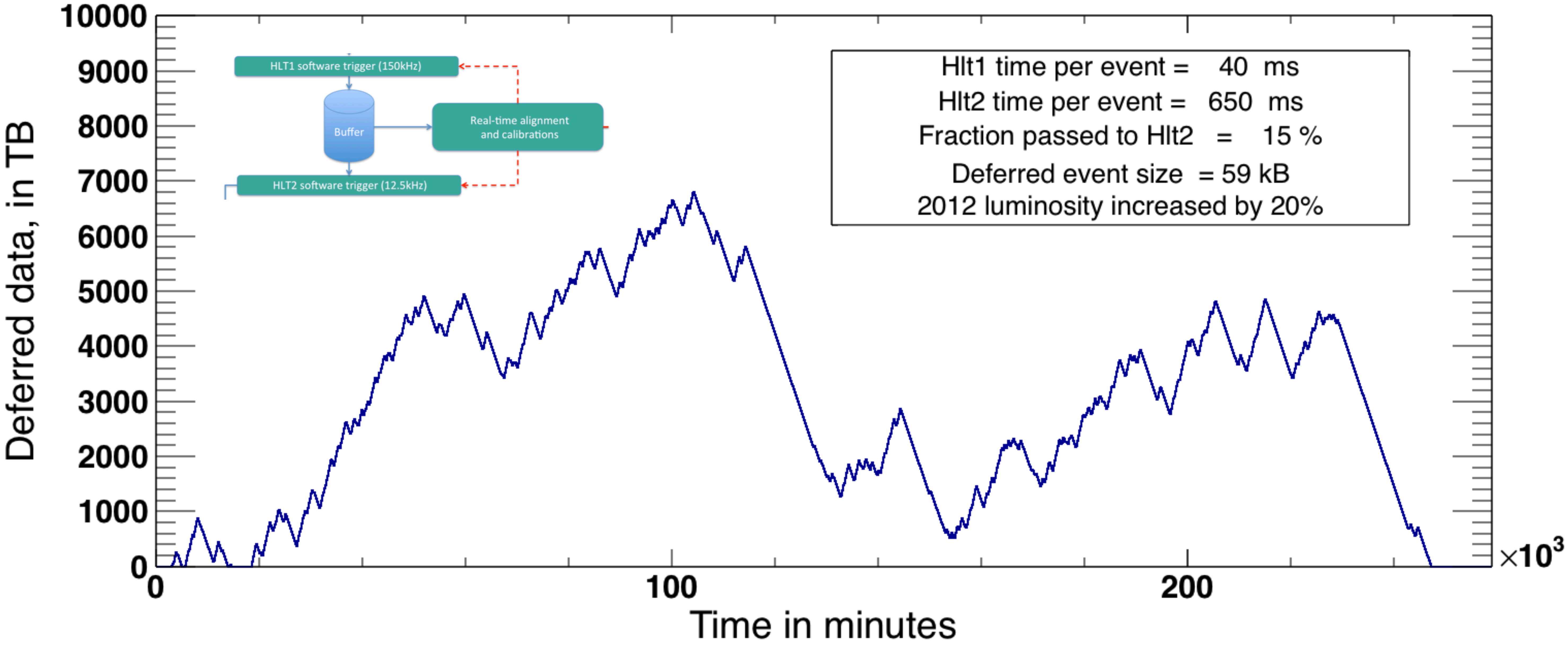
Use Run I LHC fill structure to simulate disk buffer usage

Optimization of the Run 2 LHCb cascade buffer



Use simulation to ensure robustness if timing estimates wrong

Optimization of the Run 2 LHCb cascade buffer



Use simulation to ensure robustness if LHC overperformed