A very brief sketch of some design choices in real-time analysis

V. V. Gligorov, CNRS/LPNHE

For a longer writeup see https://arxiv.org/abs/1806.10912

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?

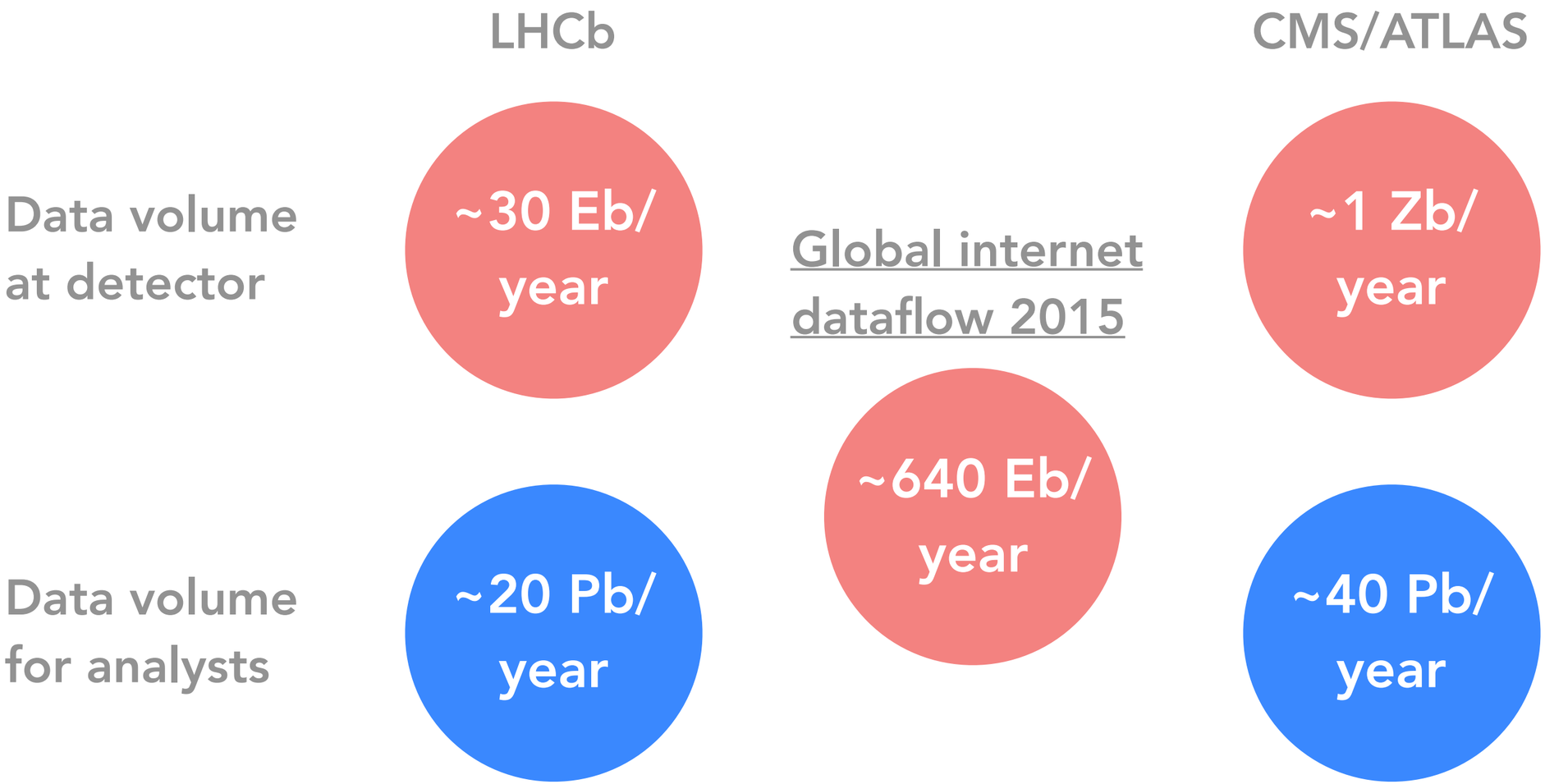
	LHCb	CMS/ATLAS
Data volume at detector	~30 Eb/ year	~1 Zb/ year

Global internet dataflow 2015

~640 Eb/ year

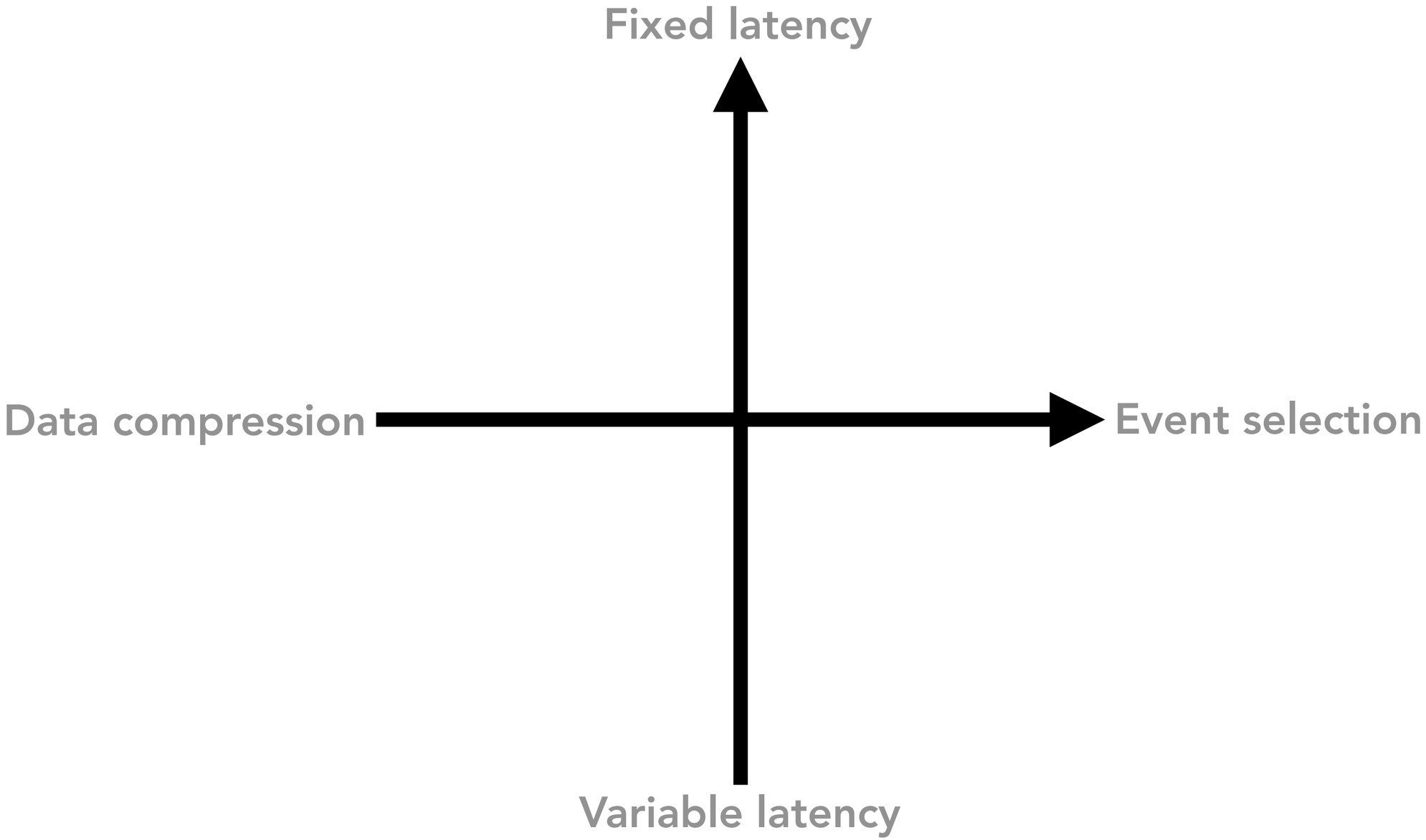
Because HEP detectors produce too much data to store

Data volumes @ LHC after real-time processing



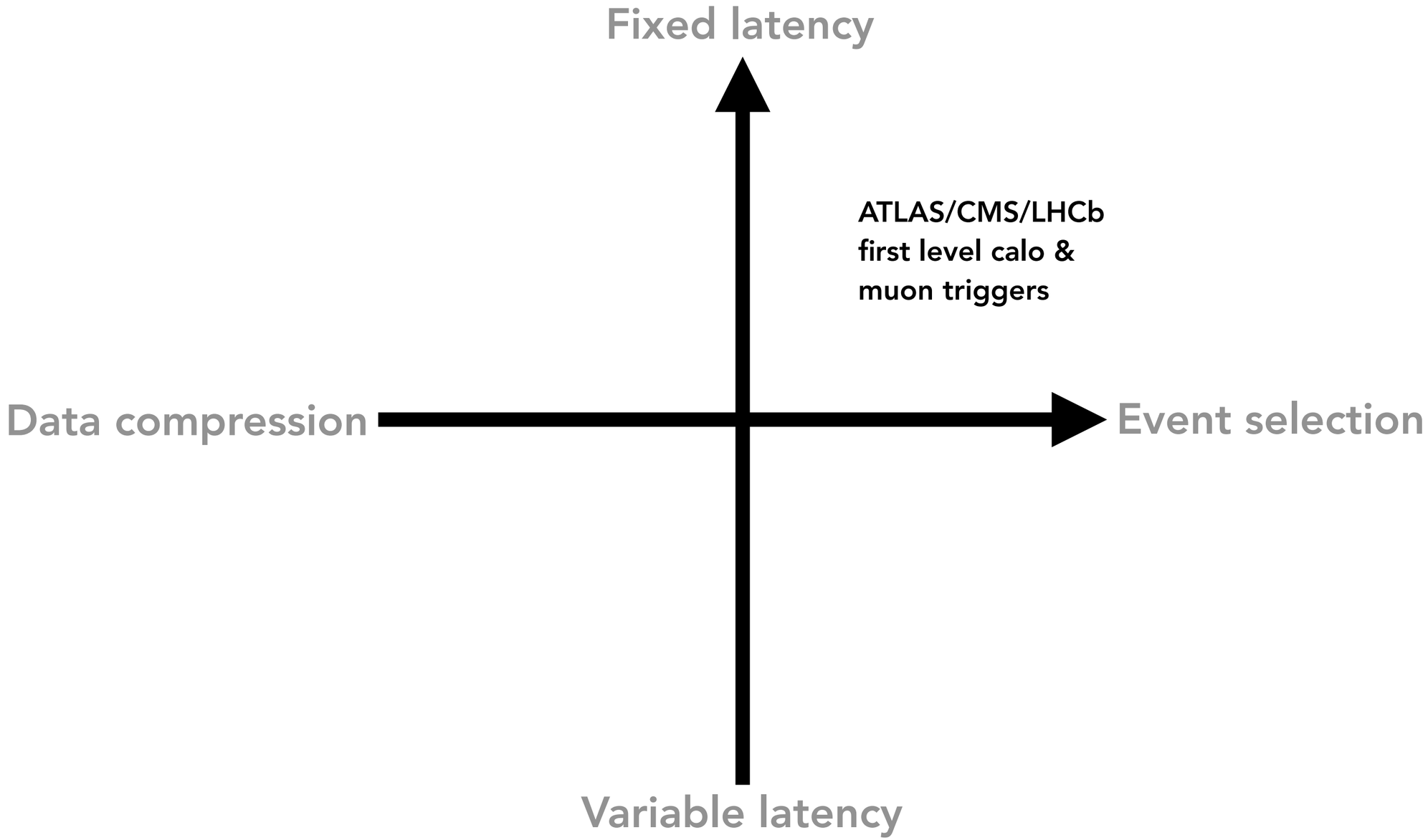
Real-time processing reduces data by 3-5 orders of magnitude

What kinds of real-time data processings exist?



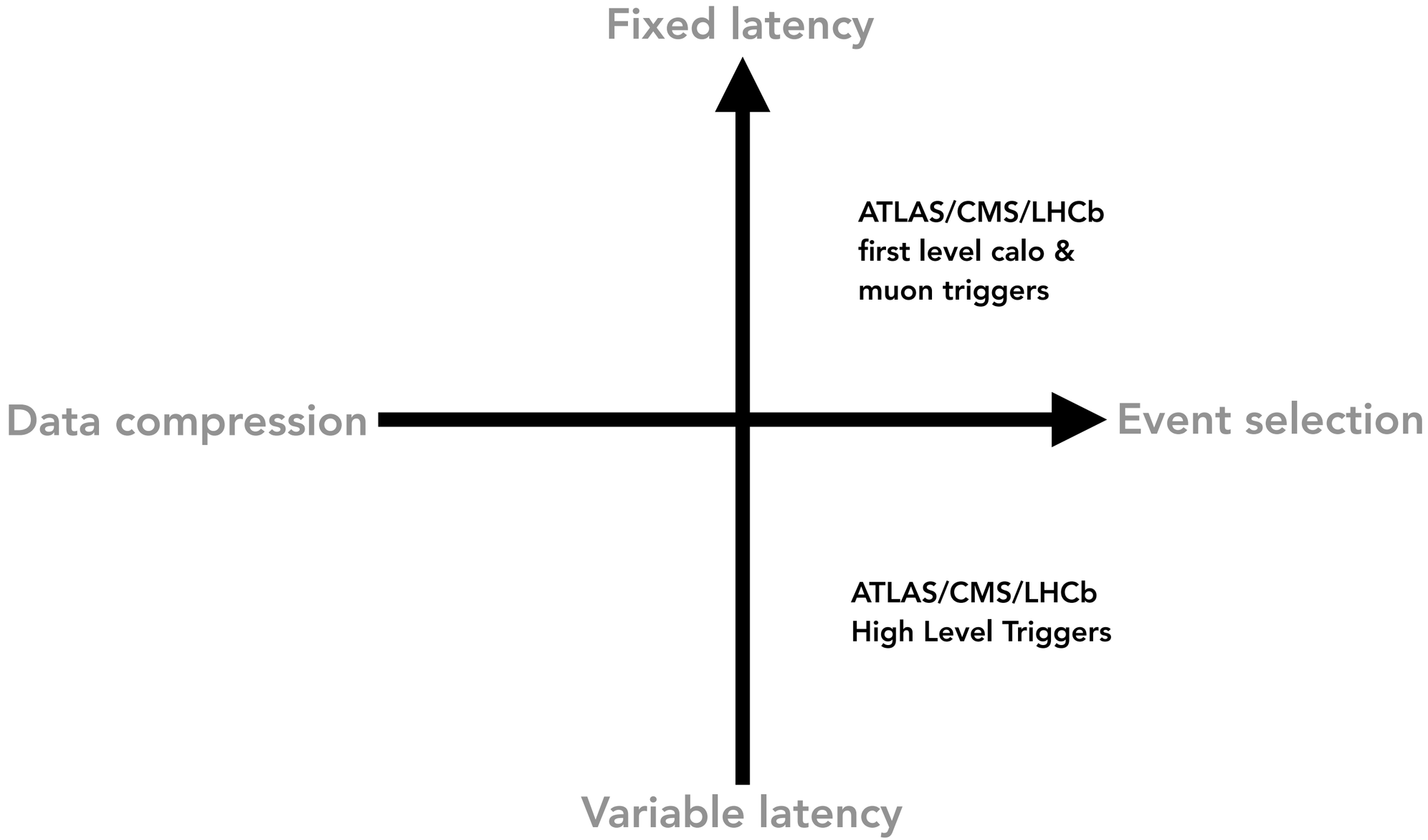
Distinguish fixed & variable latency, selection & compression

What kinds of real-time data processings exist?



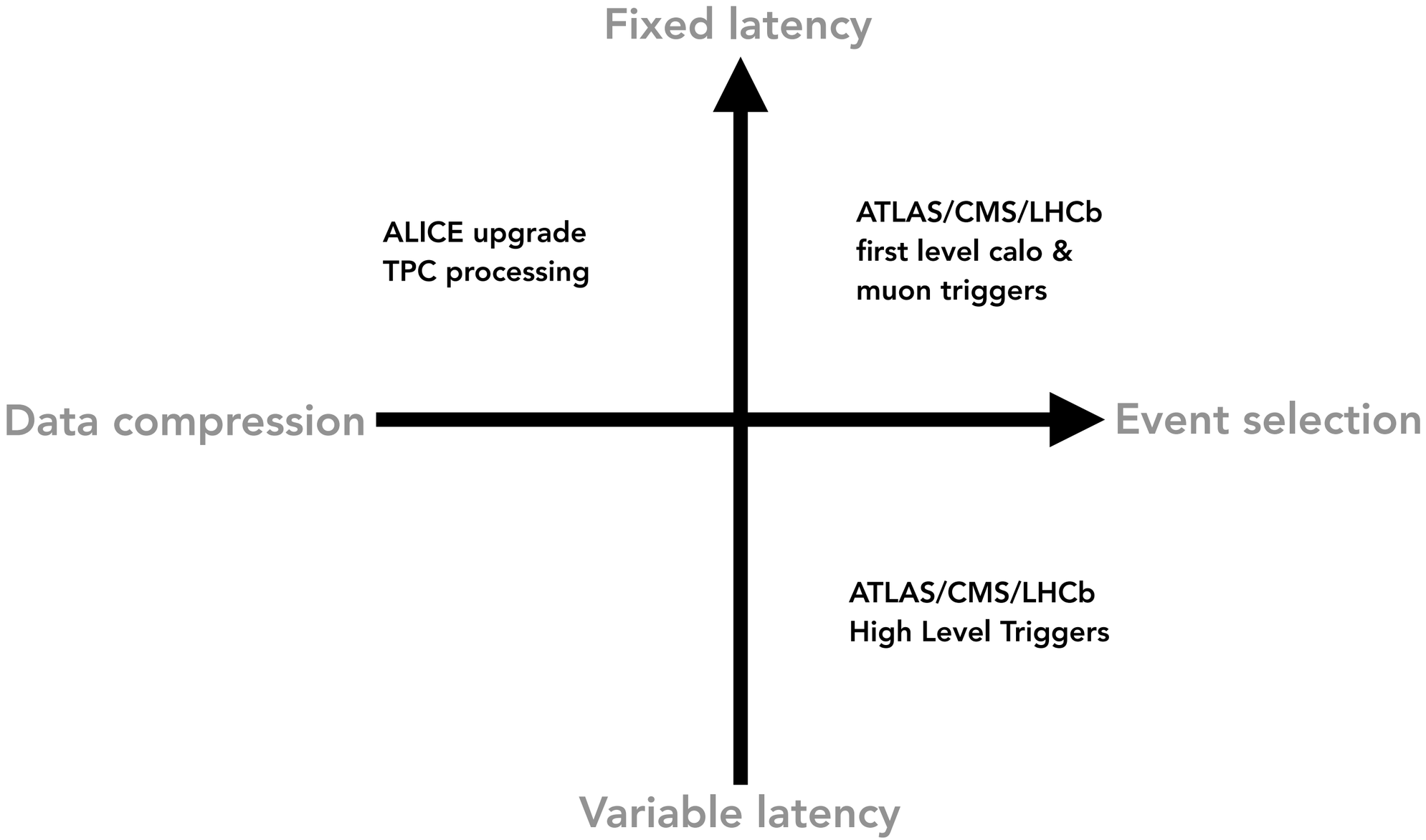
Distinguish fixed & variable latency, selection & compression

What kinds of real-time data processings exist?



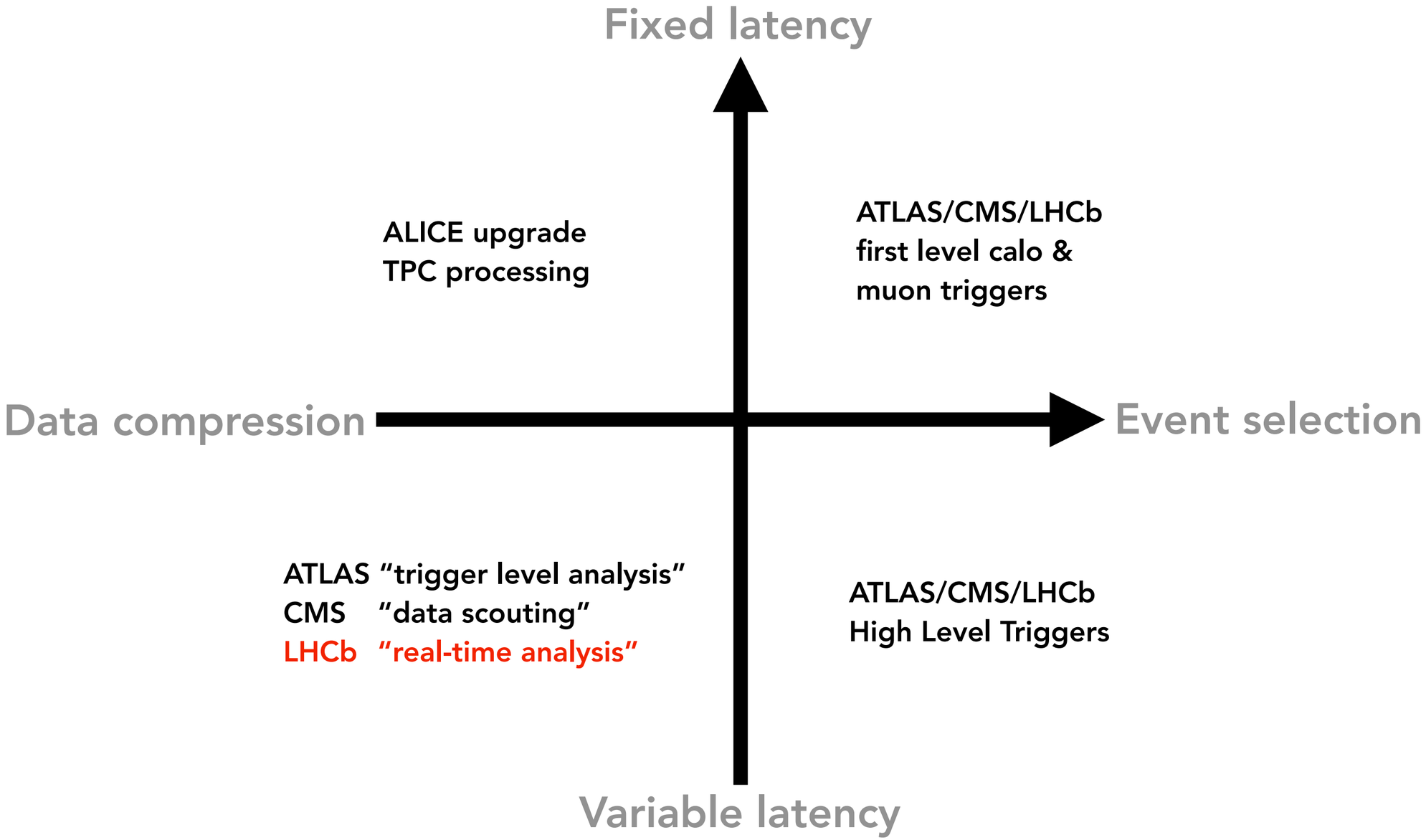
Distinguish fixed & variable latency, selection & compression

What kinds of real-time data processings exist?



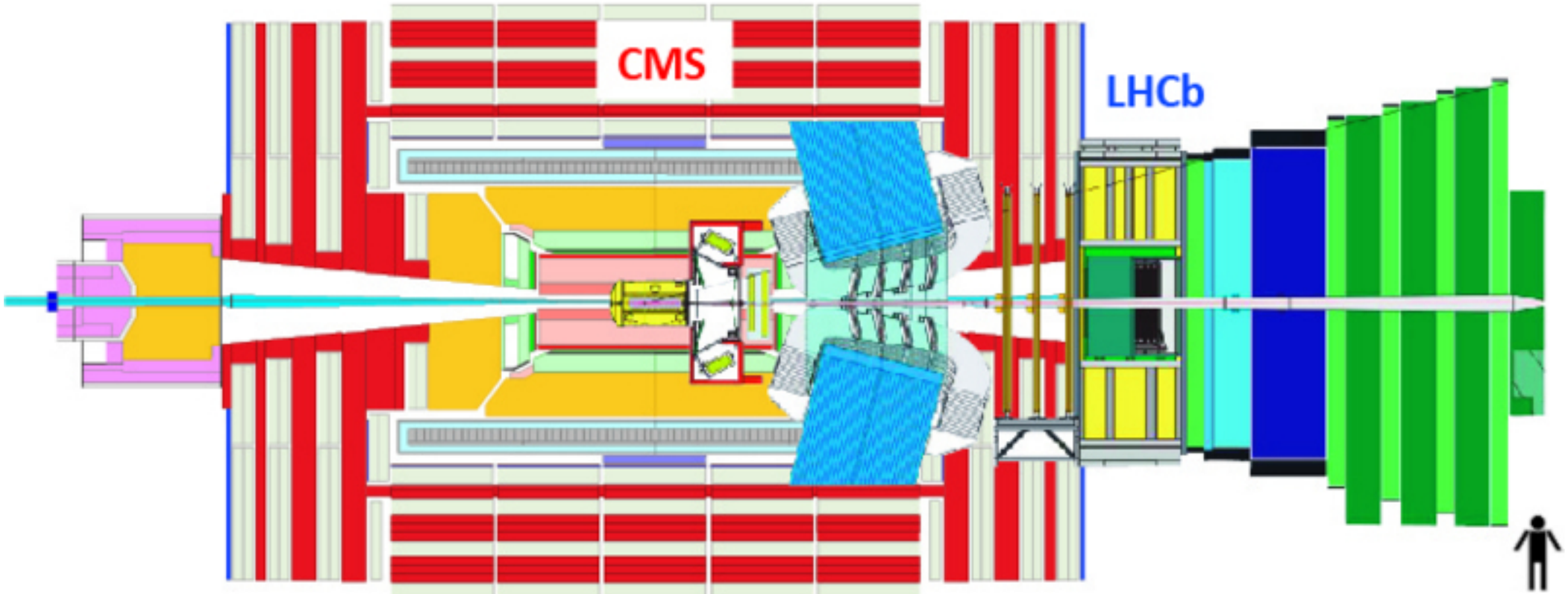
Distinguish fixed & variable latency, selection & compression

What kinds of real-time data processings exist?



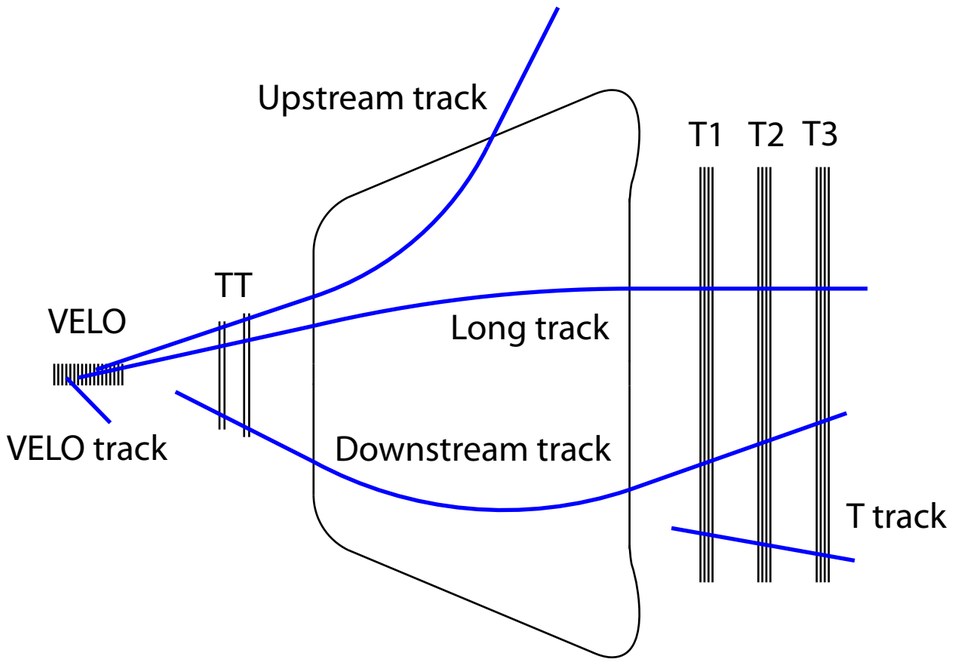
Distinguish fixed & variable latency, selection & compression

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

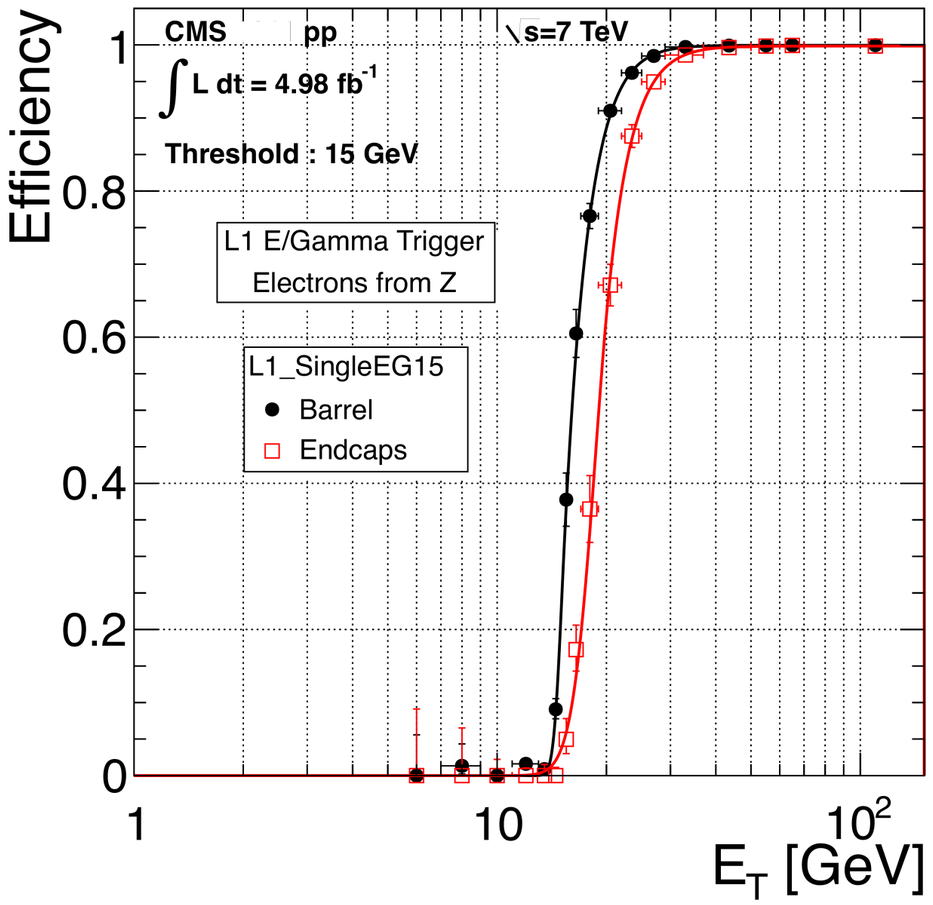
Particle identification

Tag-and-probe

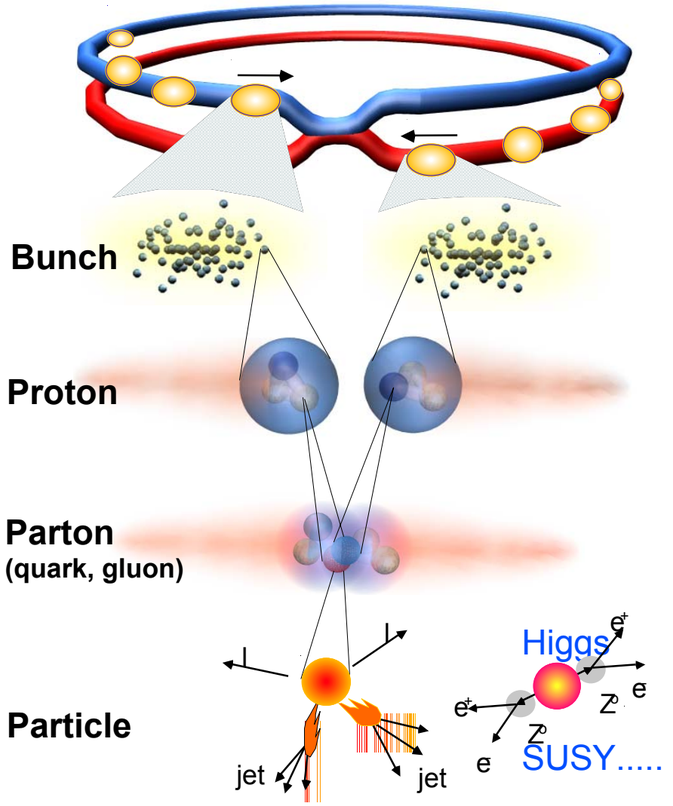
Tag-and-probe calibrations exist for all charged particle species and for π^0/γ, with new sources added over time to improve coverage

Data driven efficiency calibration key to precision physics

Traditional real-time processing, or "triggering"



Collisions at the LHC: summary



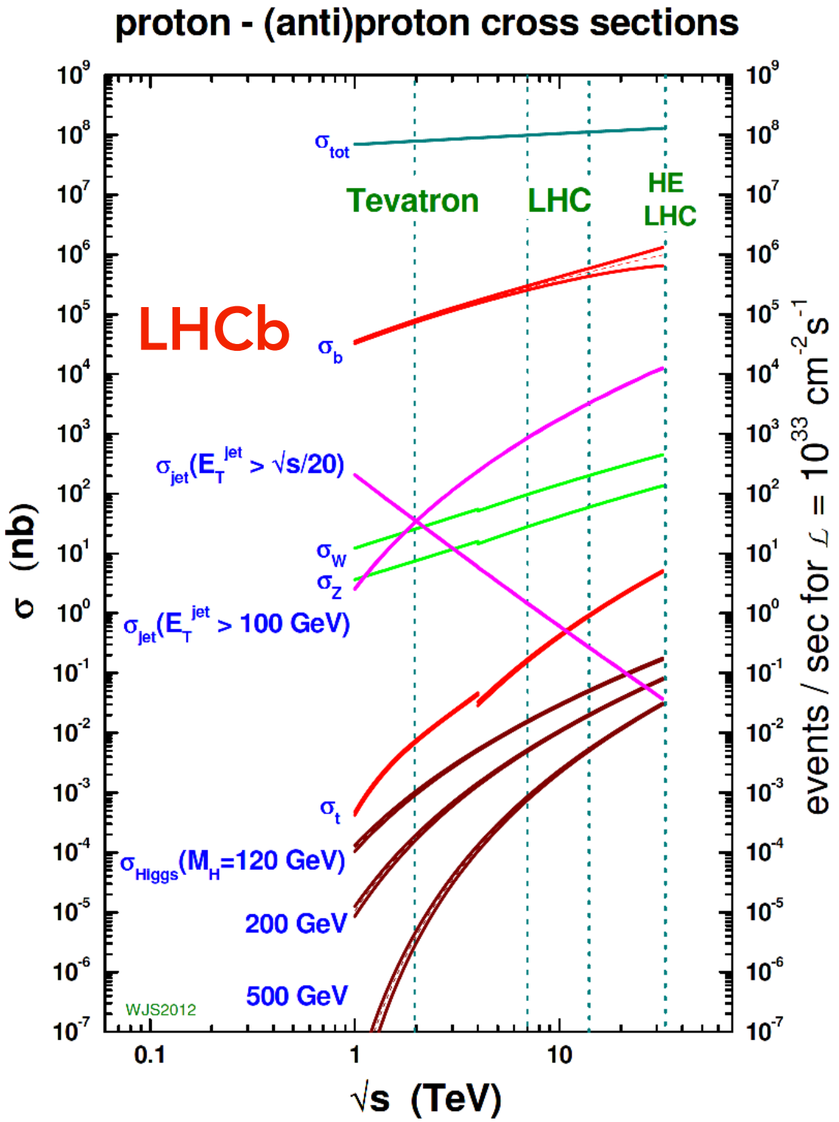
Proton - Proton	2804 bunch/beam
Protons/bunch	10^{11}
Beam energy	7 TeV (7×10^{12} eV)
Luminosity	$10^{34}cm^{-2}s^{-1}$
Crossing rate	40 MHz
Collision rate ≈	10^{7}-10^{9}

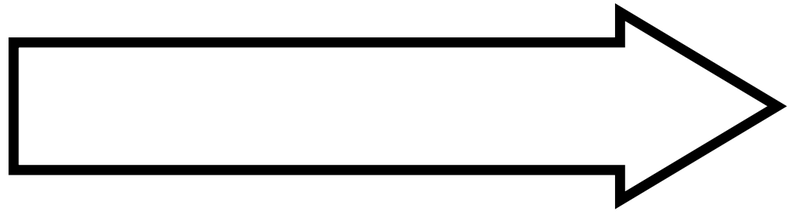
New physics rate ≈ .00001 Hz

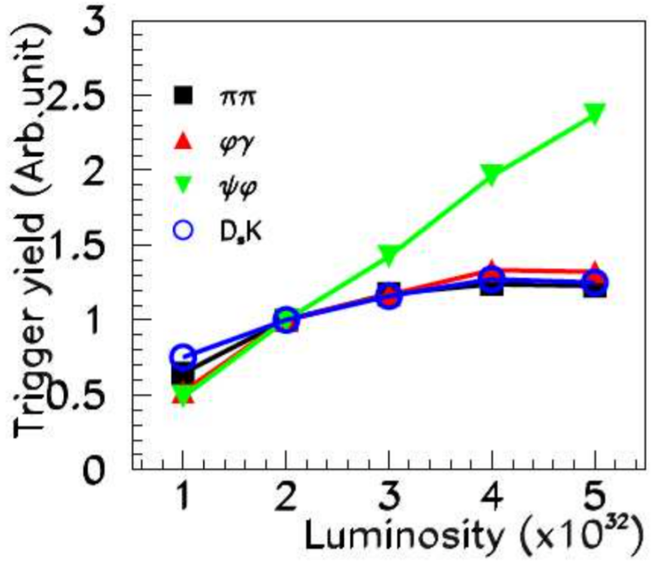
Event selection:
1 in 10,000,000,000,000

Driven by fixed-latency selection, analysis on efficiency plateau

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







LHC increases its luminosity by generating multiple pp interactions in a single bunch crossing

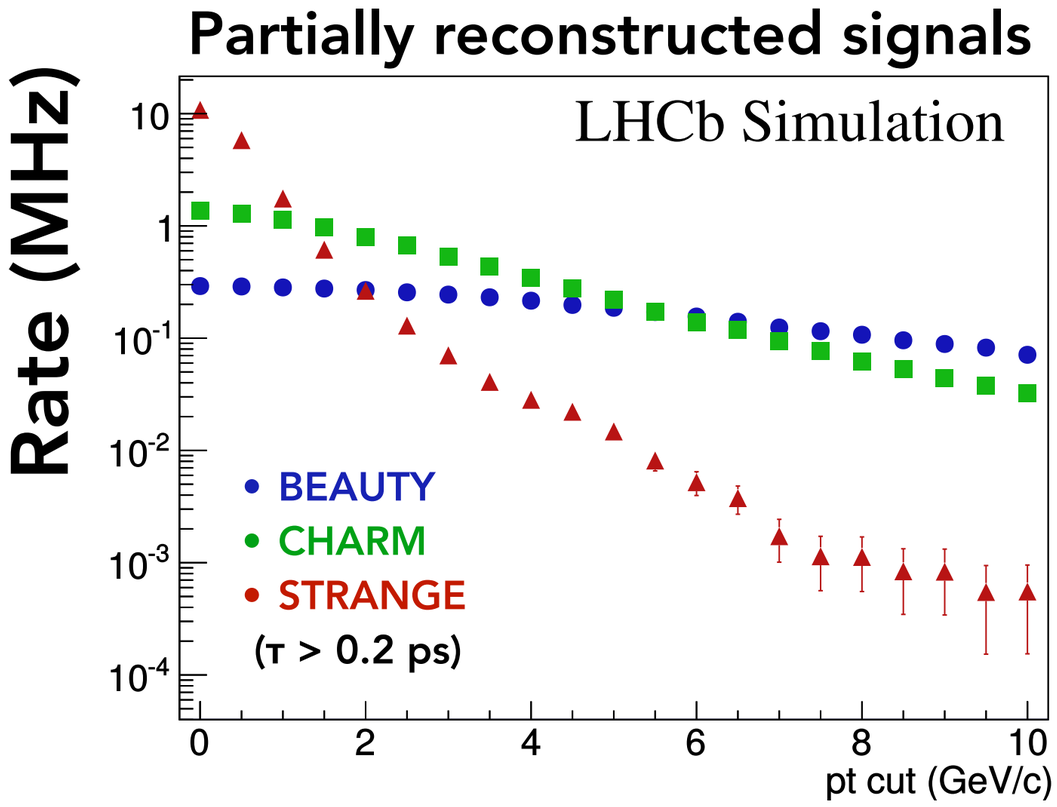
Fixed latency triggers select bunch crossings

Beyond some luminosity, all bunch crossings contain signal. Select interactions, not bunch crossings => real-time analysis.

No possibility to work on efficiency plateau!

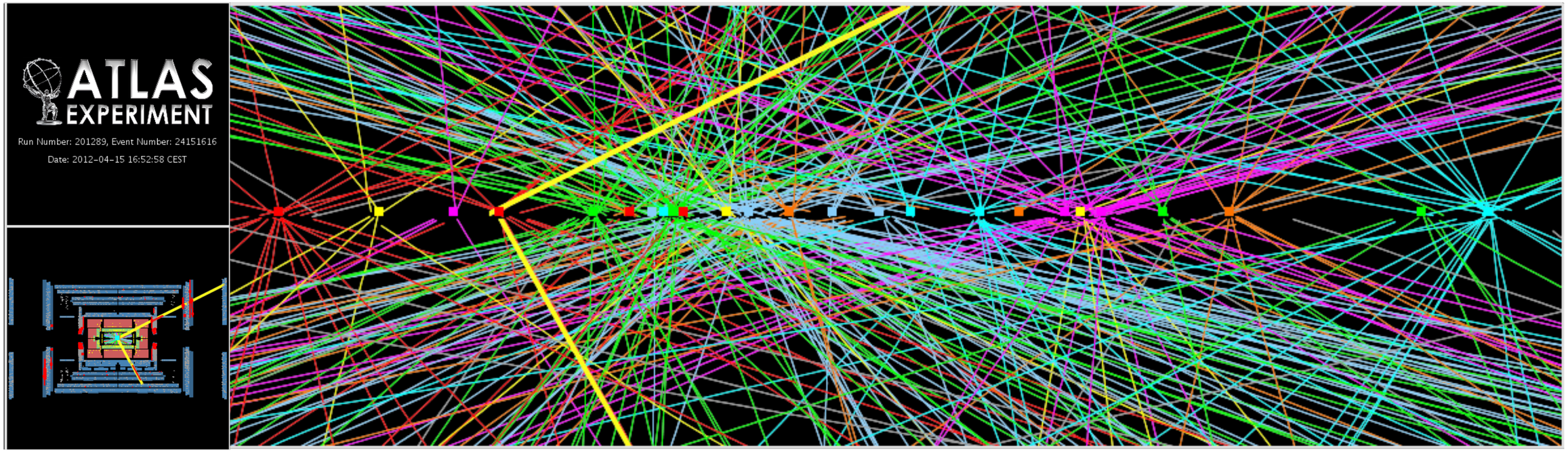
Fixed-latency trigger only effective up to around $4 \cdot 10^{32}$

Signal and data rates at LHCb in the upgrade



Plus data volume increases quadratically because of pileup

From selection to compression : real-time analysis



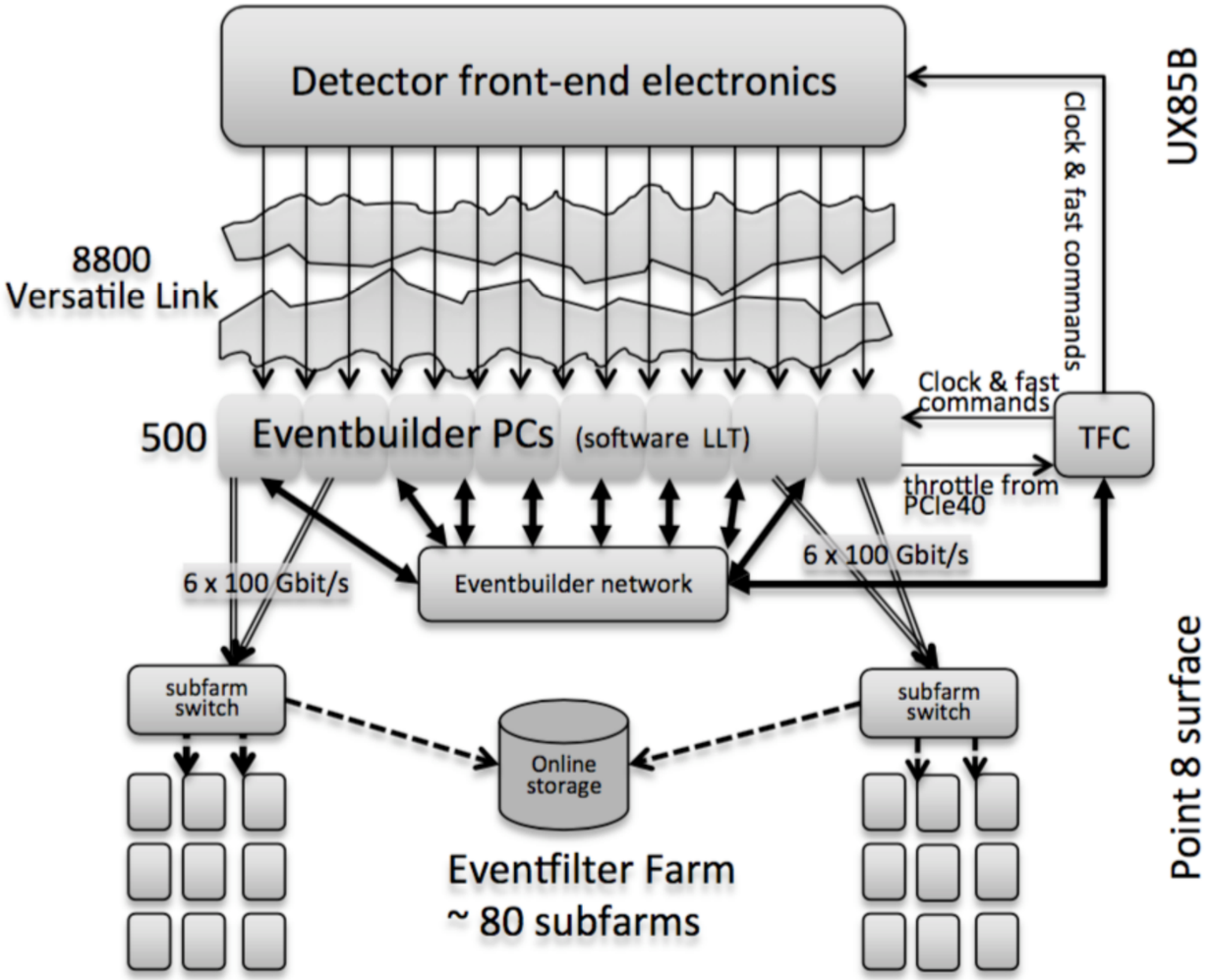
Most physics measurements require only a signal candidate and information about the specific pp collision which produced it → the rest is pileup

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

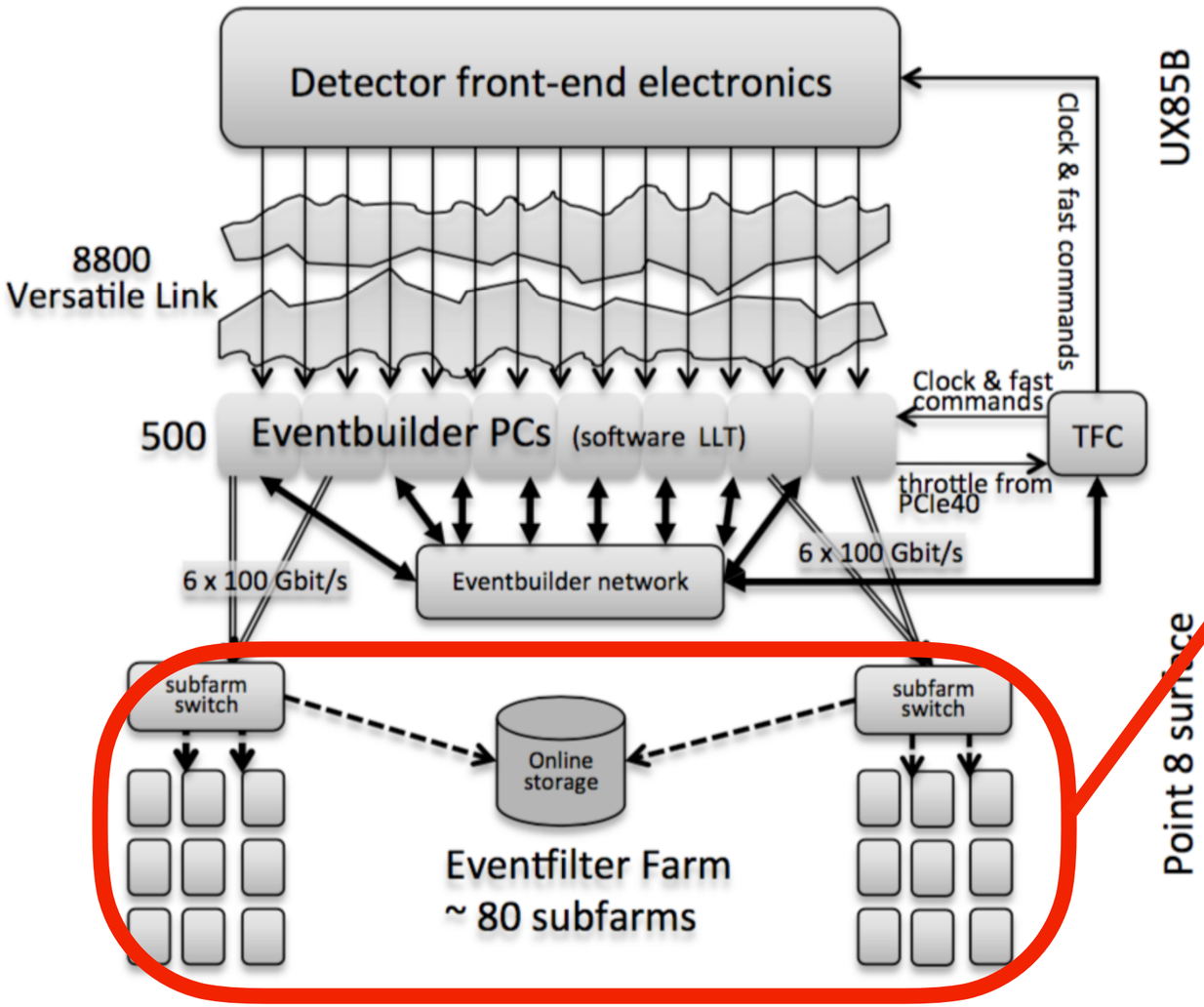
Hence create more room for signal by compressing & removing pileup in real-time!

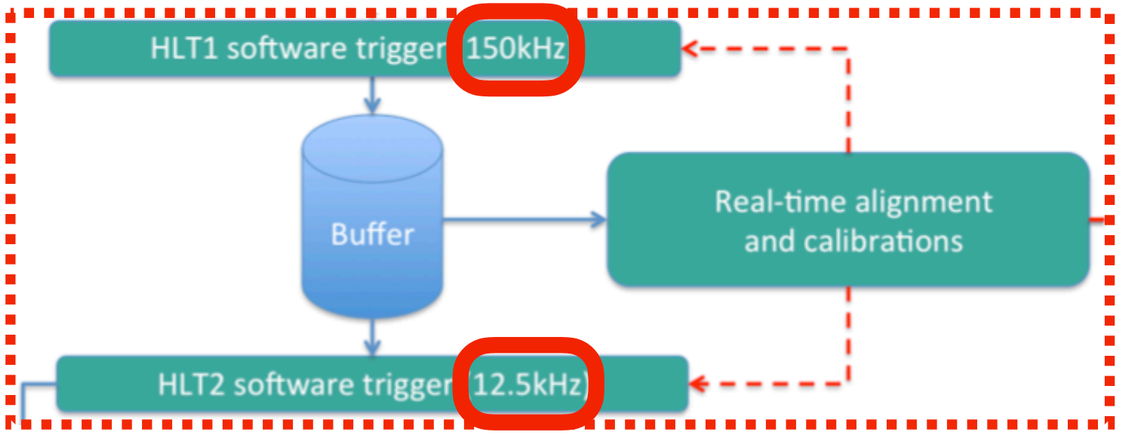
Requires the ability to carry out precise pileup suppression

The LHCb detector readout for the upgrade



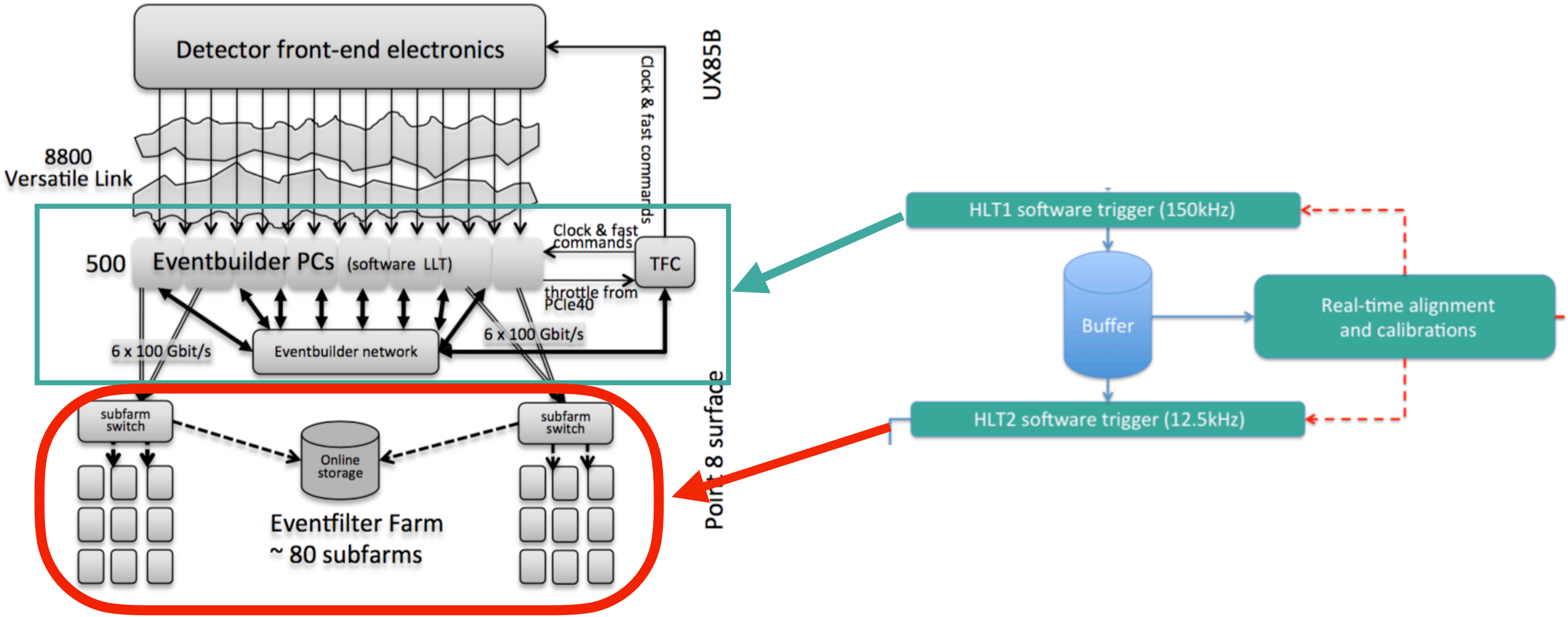
Looking inside the eventfilter farm





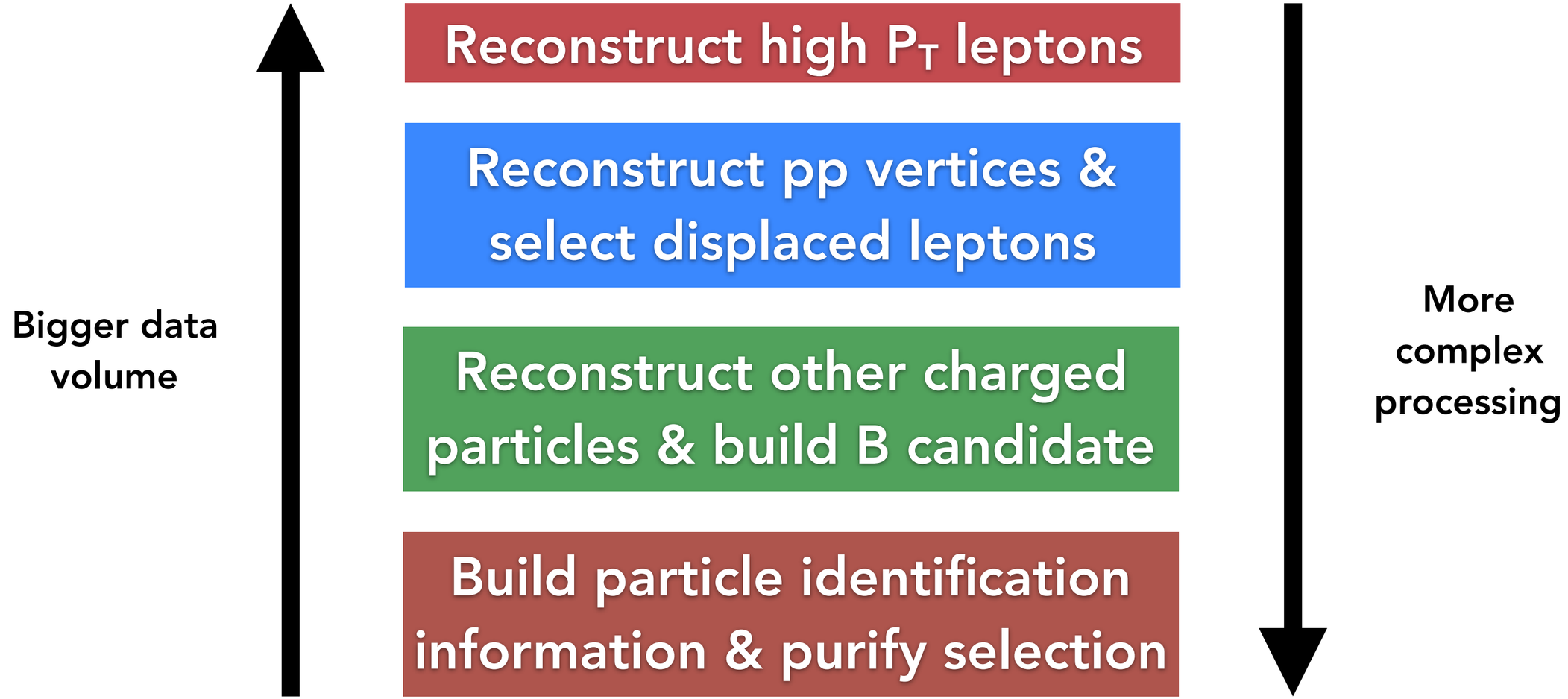
Splitting the HLT — example of a cascade buffer

But we should do a global DAQ optimization



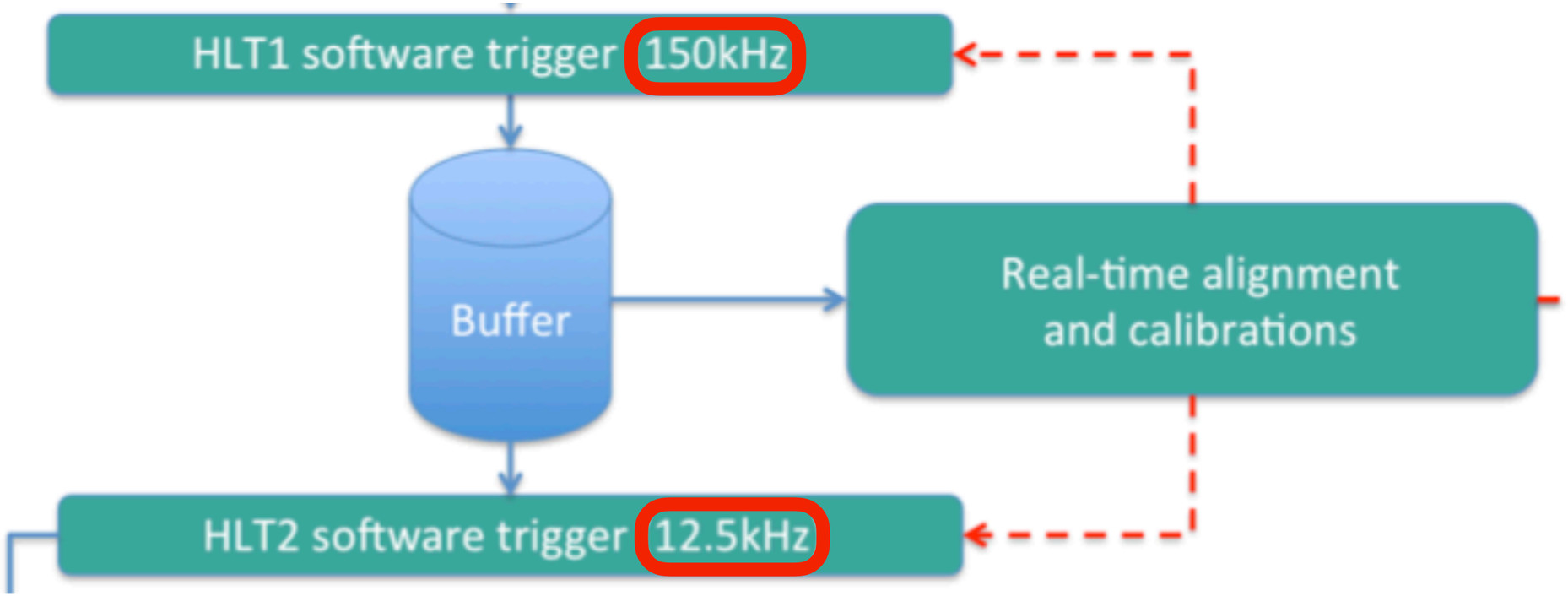
Consider whole system: if coprocessors in event building network reduce event rate by O(10), greatly reduce cost of the network. Also reduces communication cost between x86 and coprocessor, since data goes directly to the coprocessor => motivation for Allen.

What is a cascade buffer?



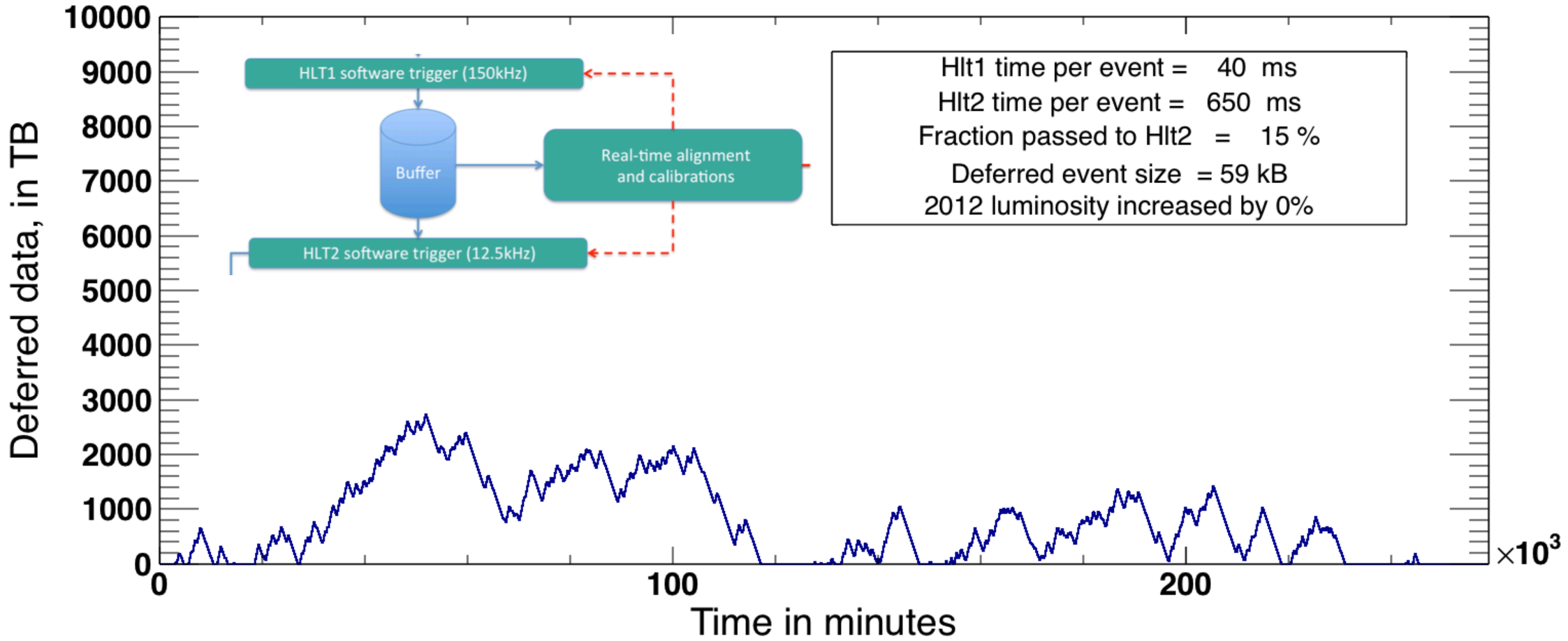
A staged data reduction using increasingly complex algorithms

Take a step back and see how we optimized this for Run 2



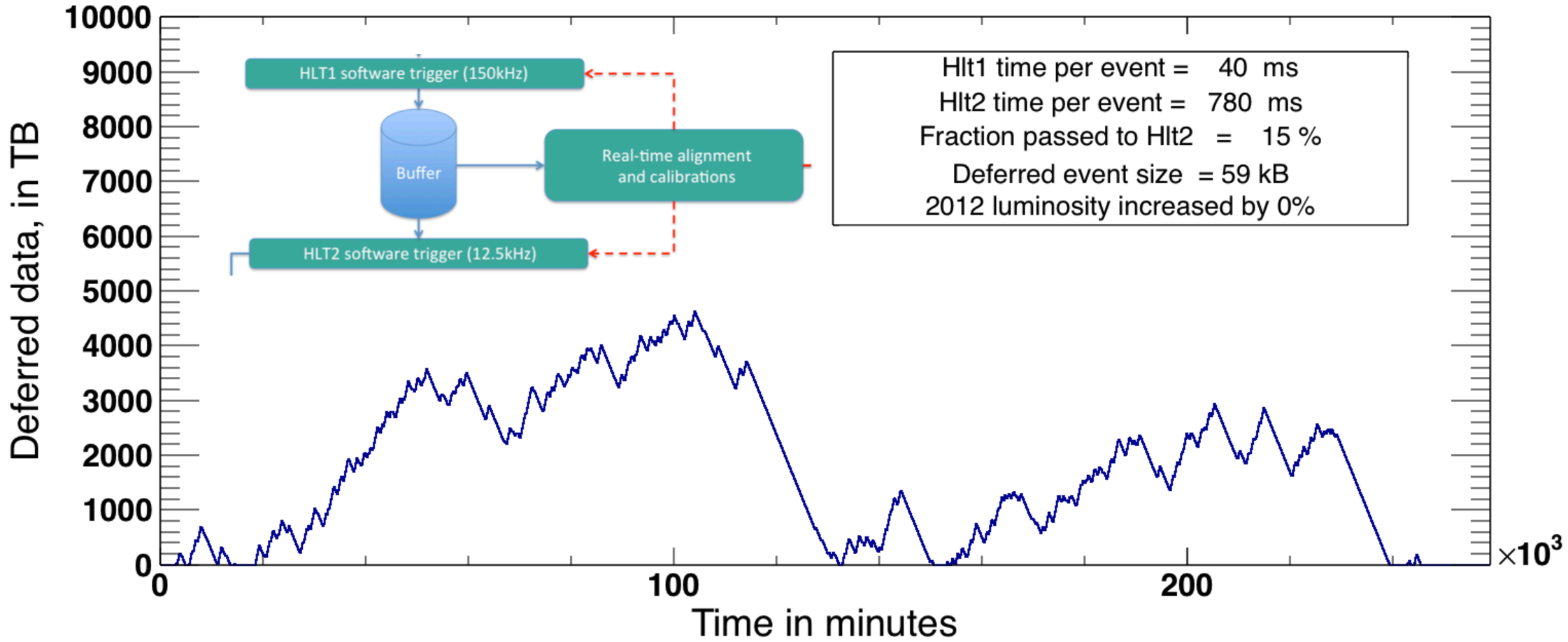
Balance retention of HLT1 against processing time of HLT2

Optimization of the Run 2 LHCb cascade buffer



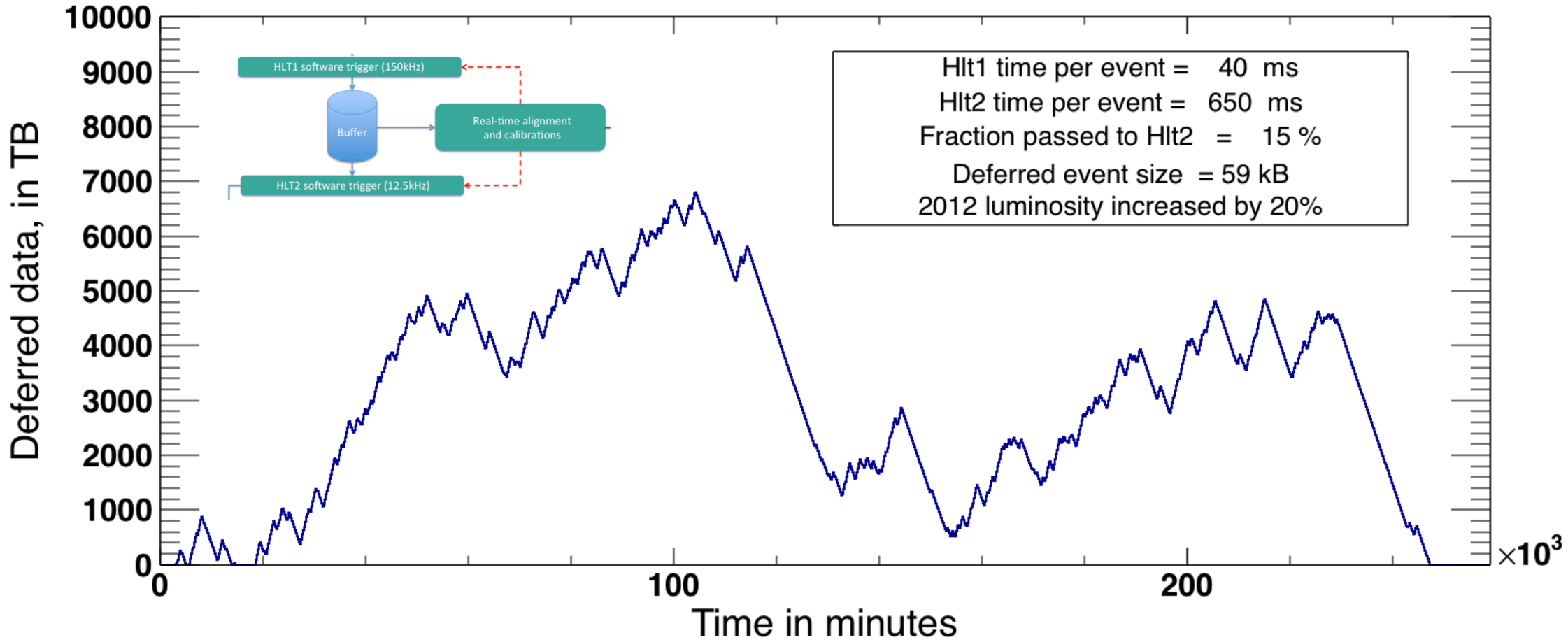
Use Run I LHC fill structure to simulate disk buffer usage

Optimization of the Run 2 LHCb cascade buffer



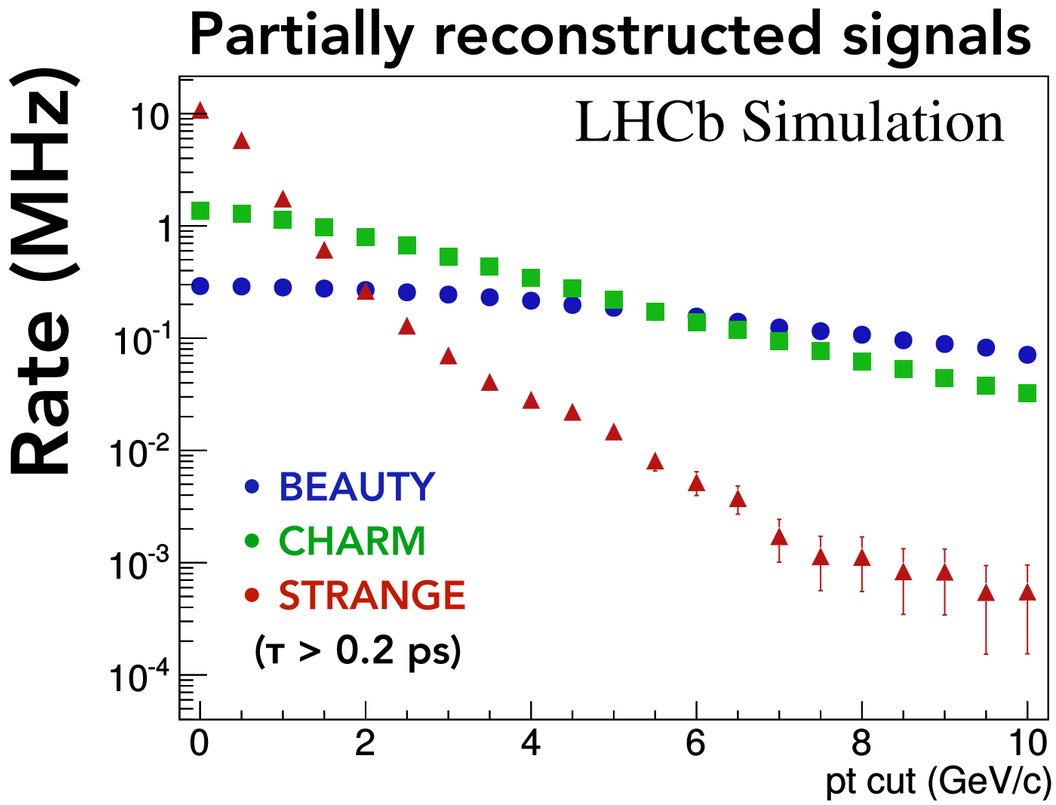
Use simulation to ensure robustness if timing estimates wrong

Optimization of the Run 2 LHCb cascade buffer



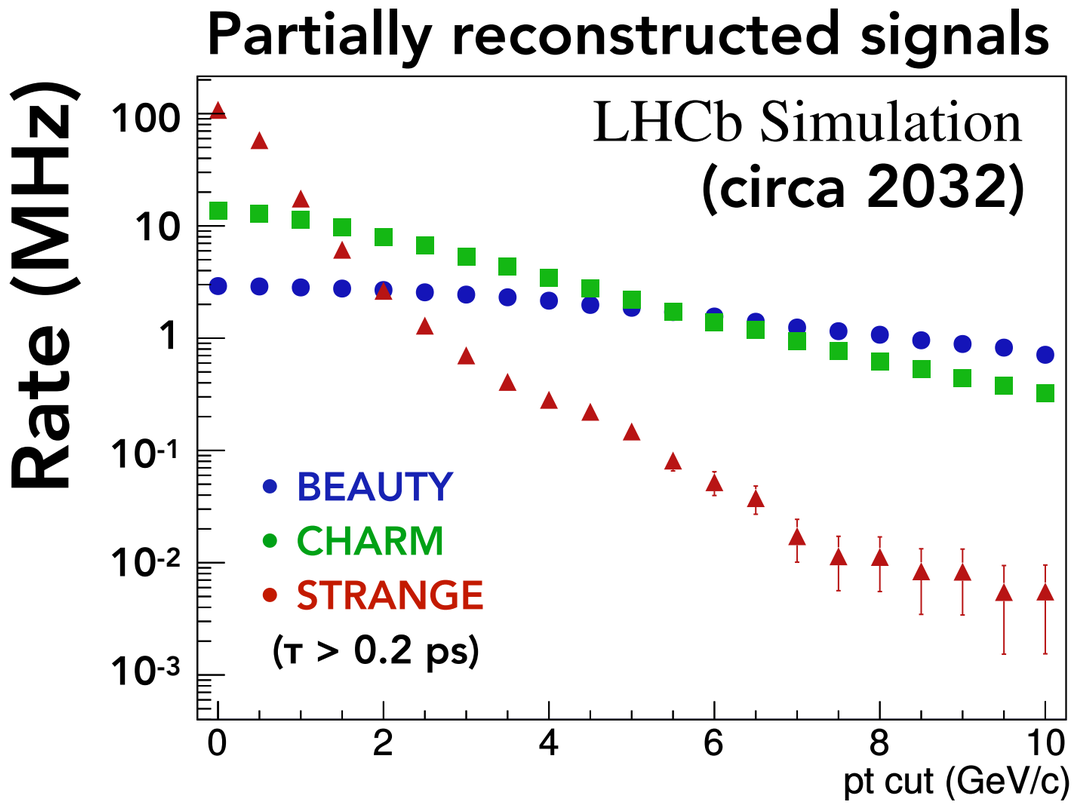
Use simulation to ensure robustness if LHC overperformed

Evolution of real-time analysis towards the LHCb upgrade...



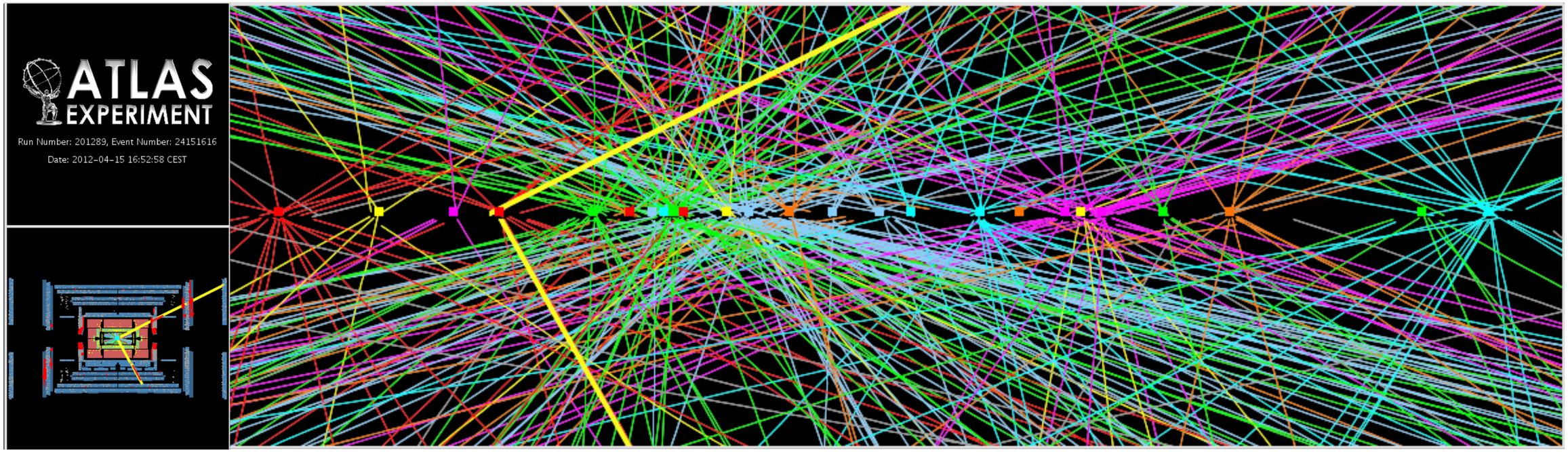
Still just about room for a first level selective trigger

...and a potential second upgrade



But at $2 \cdot 10^{34}$, even that will no longer be possible

Why is real-time analysis here to stay?



Almost all bunch crossings will contain interesting signal, most proton-proton collisions will not

➡ Our triggers should select collisions, not bunch crossings

Requires ~offline-quality real-time reconstruction, detector alignment&calibration

Requires access to “rest of event” information (tagging, isolation…) in real-time

Fundamentally because it is driven by physics, not technology.