

Data processing in LHCb at 30 MHz

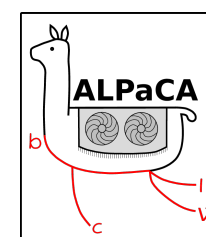
Dorothea vom Bruch

CPPM, Aix-Marseille Université, Marseille, CNRS/IN2P3

Workshop on computing strategy

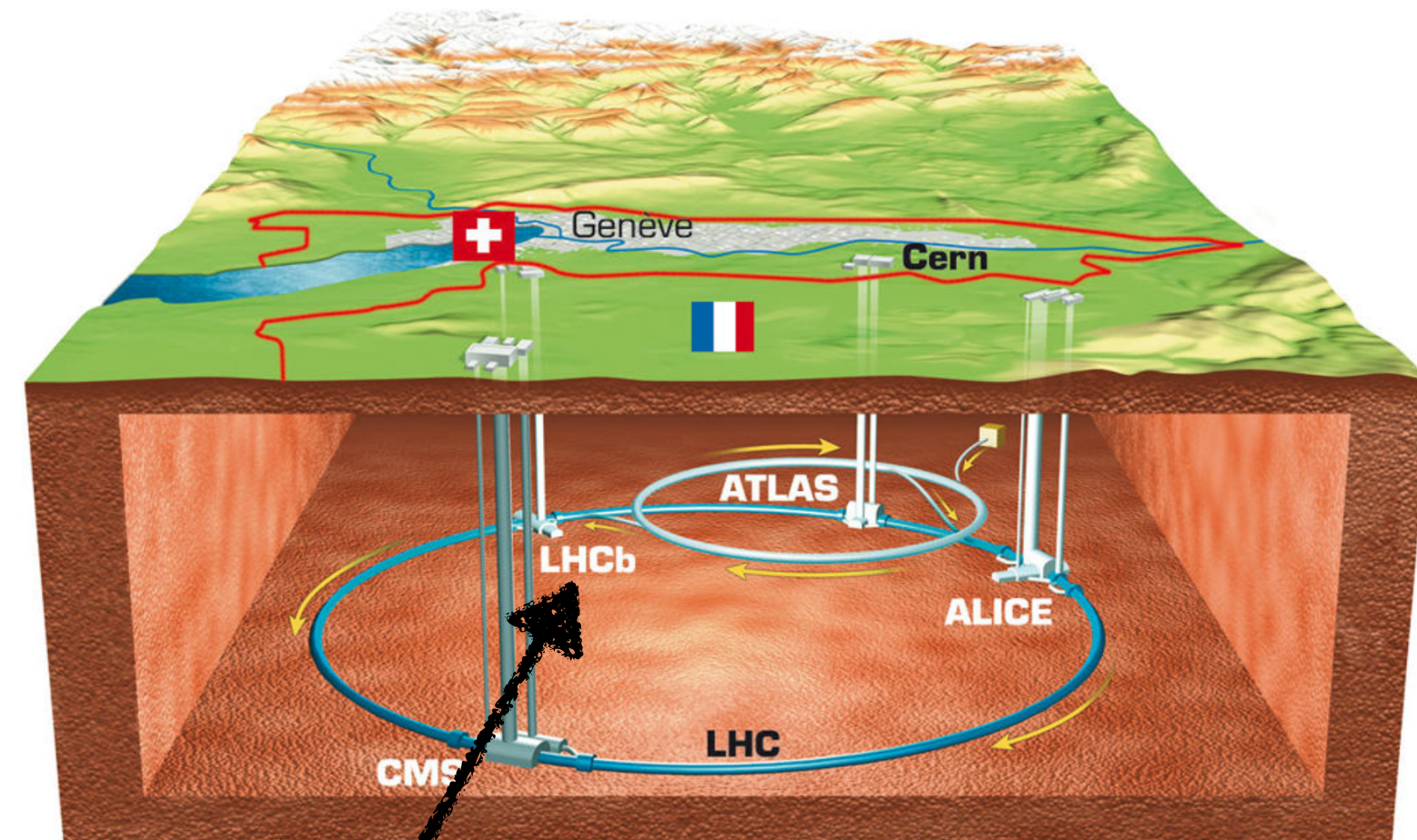
CEA Irfu Centre de Saclay

September 24th 2026

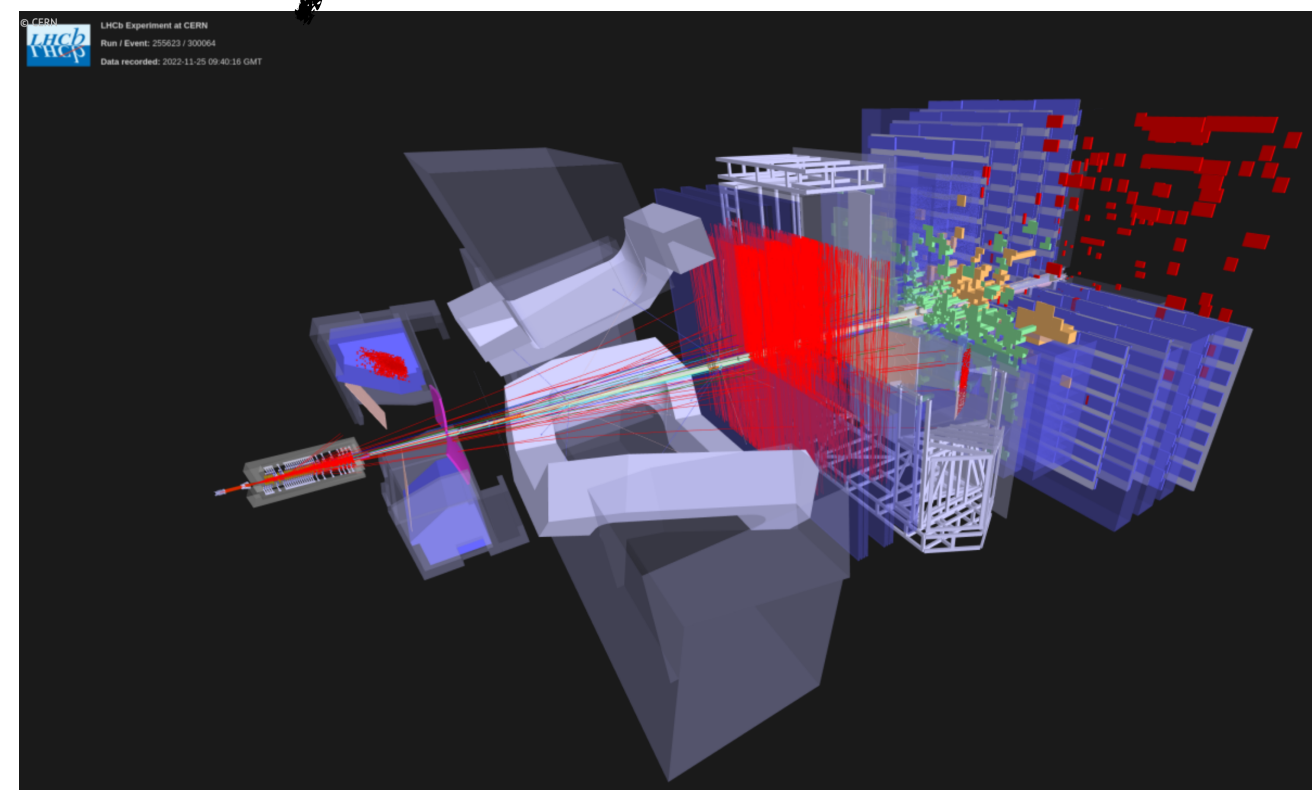


The Large Hadron Collider beauty experiment (LHCb)

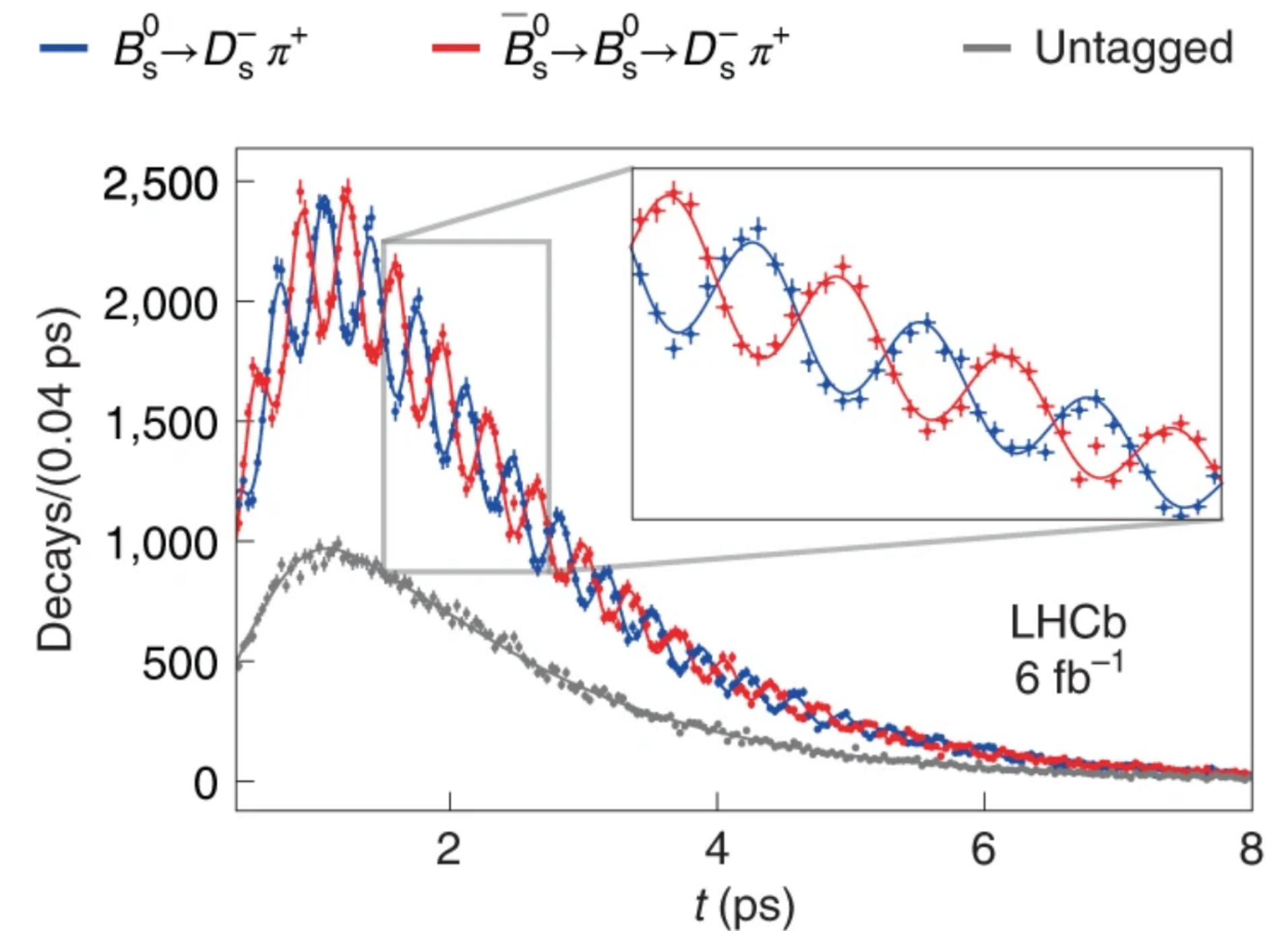
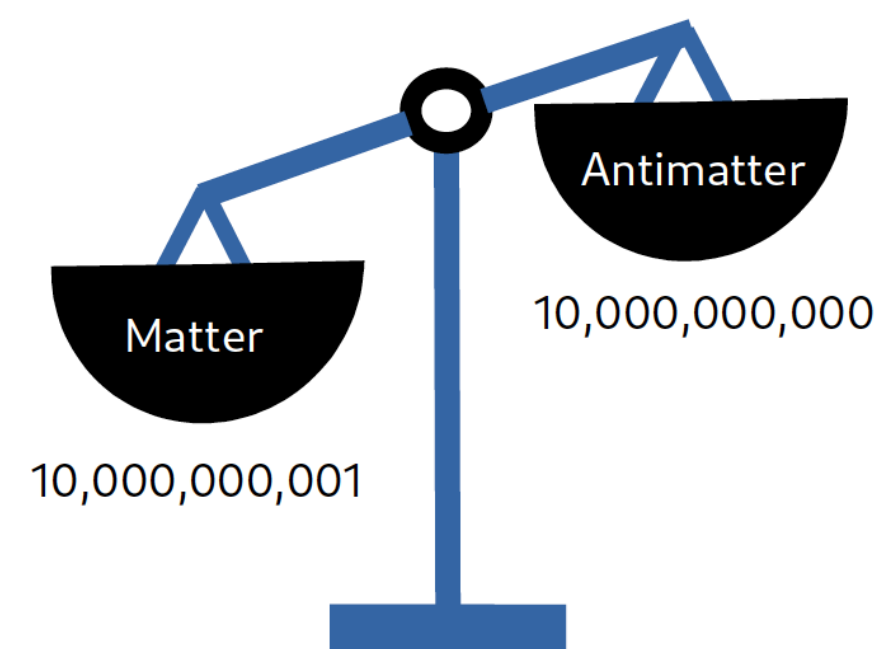
Large Hadron Collider (LHC)



LHCb

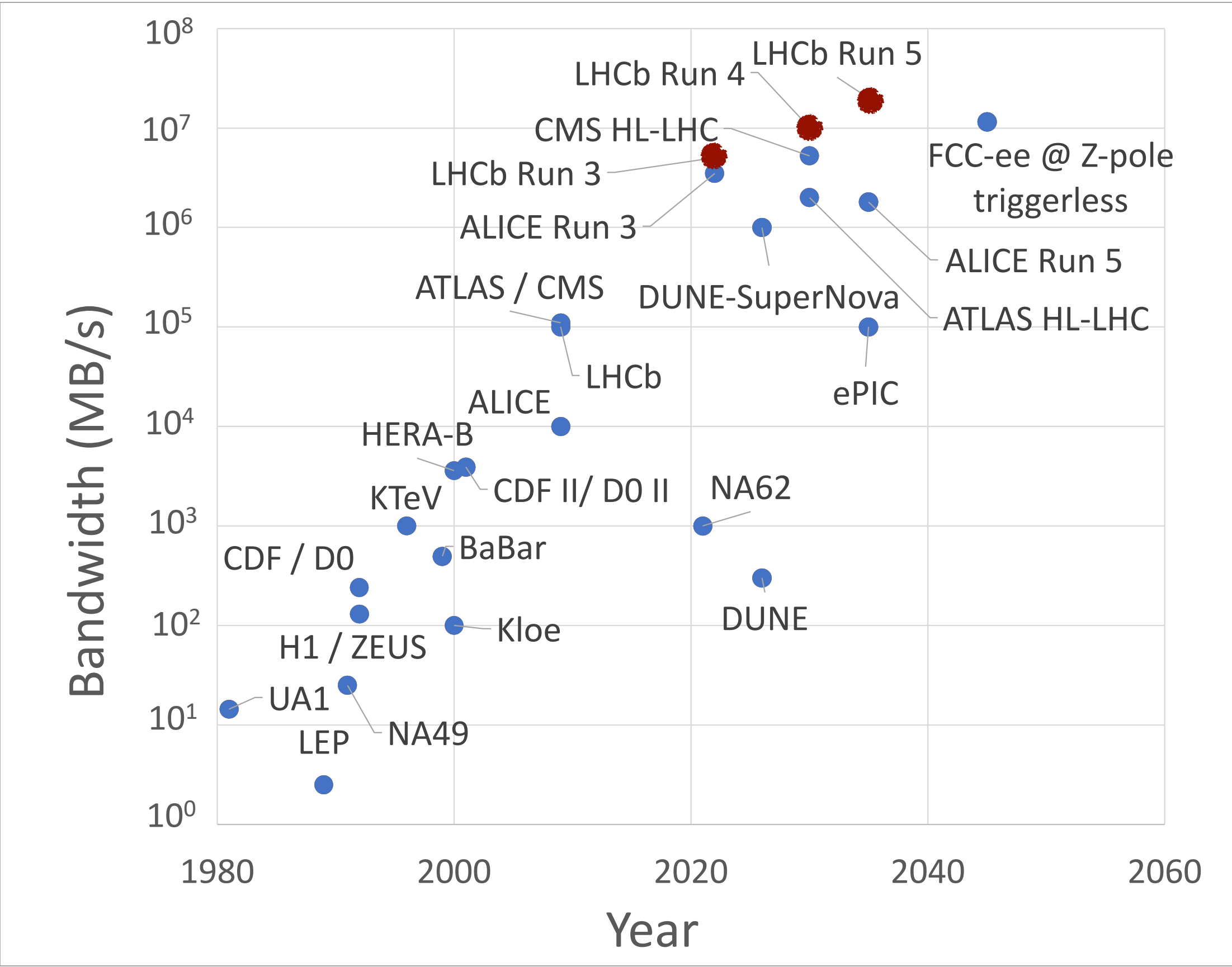
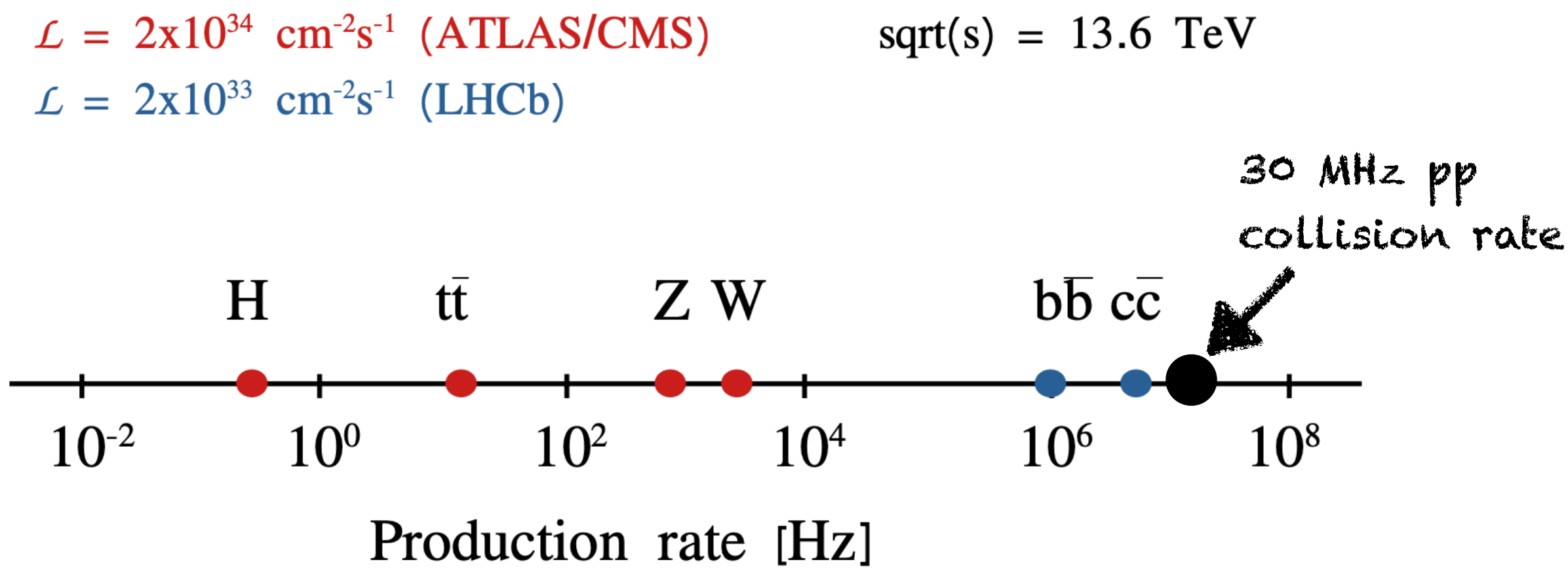


- Originally designed for study of beauty and charm physics
- Today: general-purpose detector in forward region
- Proton-proton and heavy ion / fixed target collisions



R. Aaij et al, Nature Physics 18, 1-5 (2022)

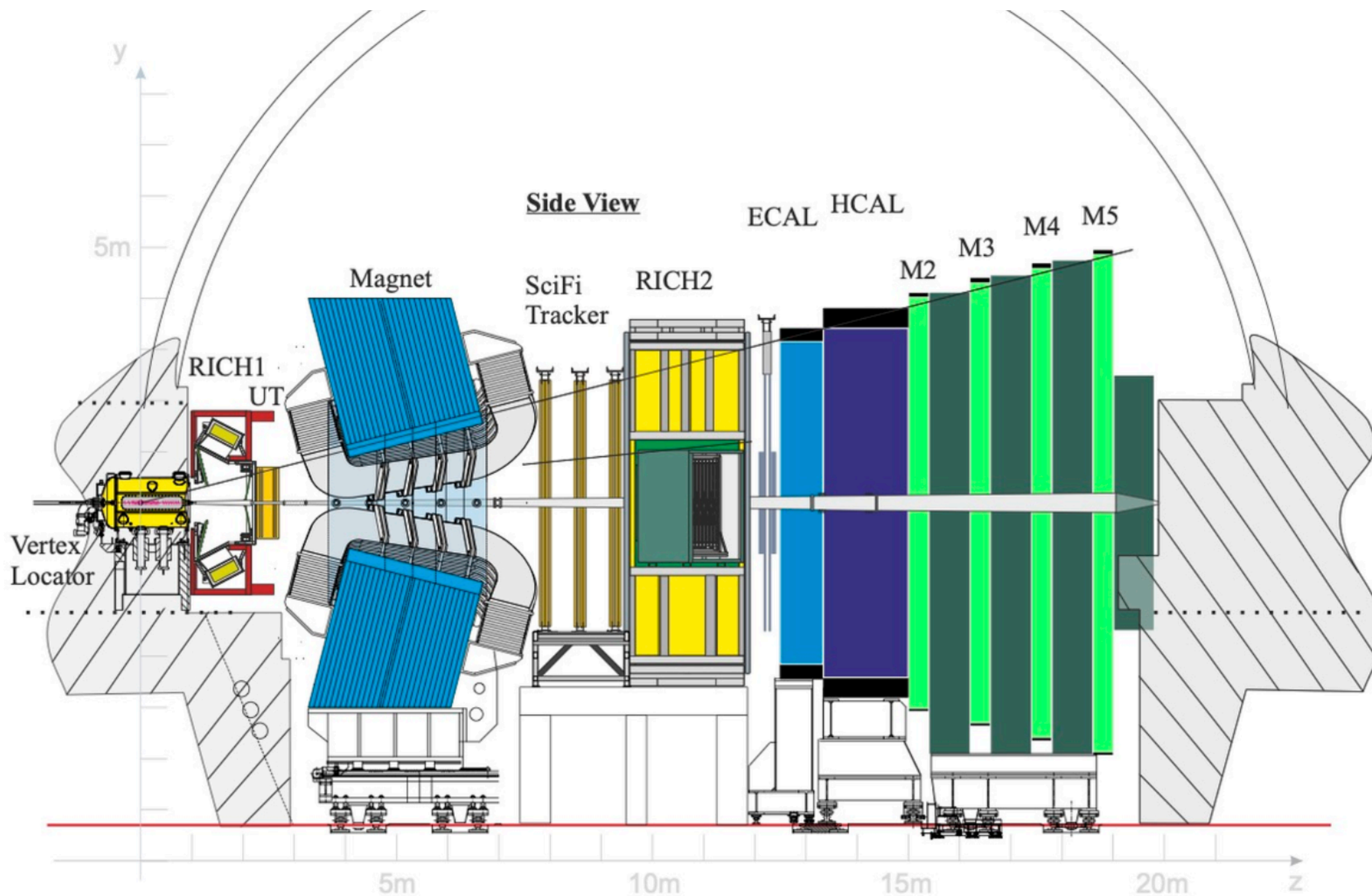
The LHCb trigger challenge



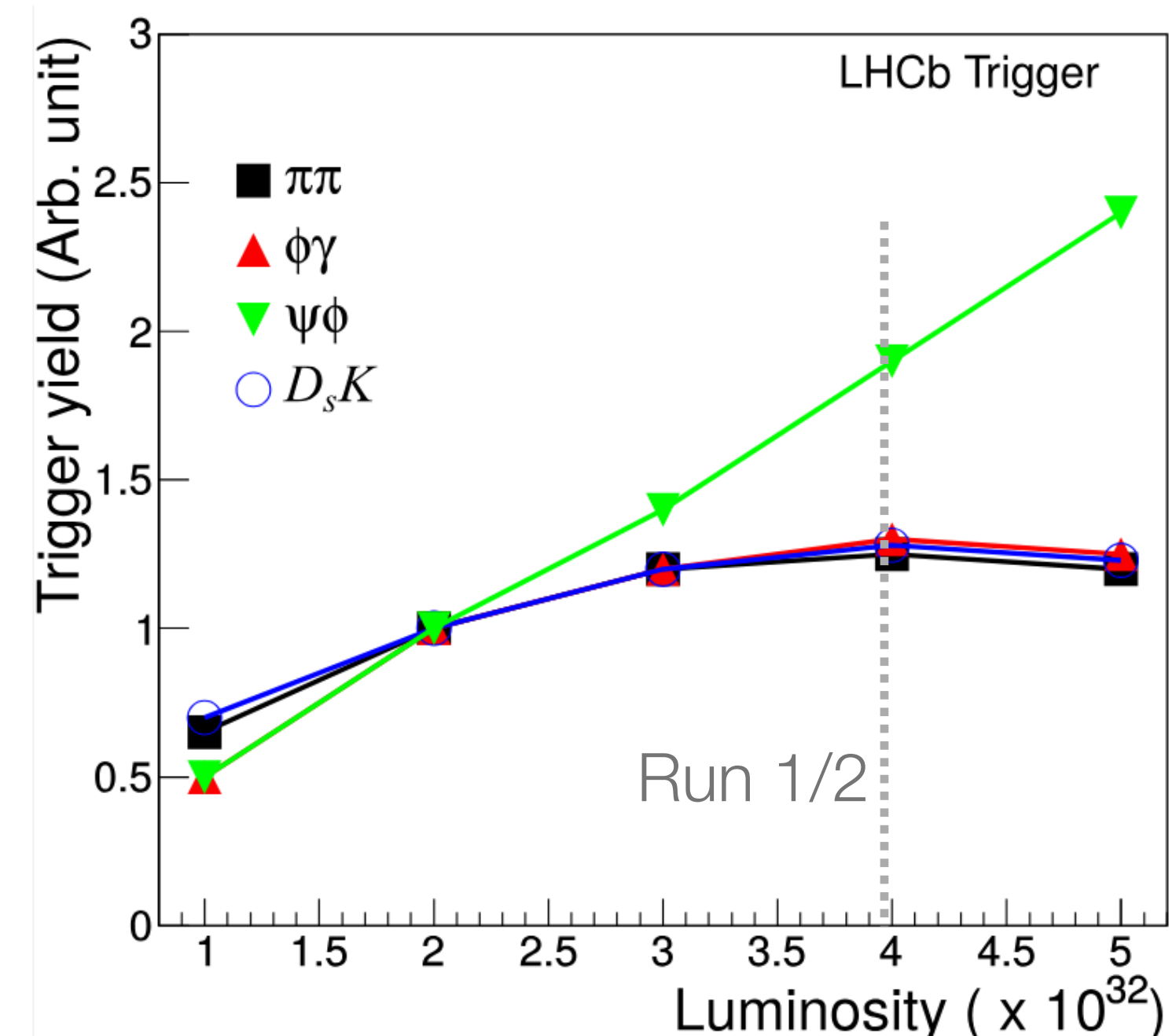
CERN-ESU-2025-001, <https://arxiv.org/abs/2511.03883>

LHCb Run 3 detector (2022 - 2026)

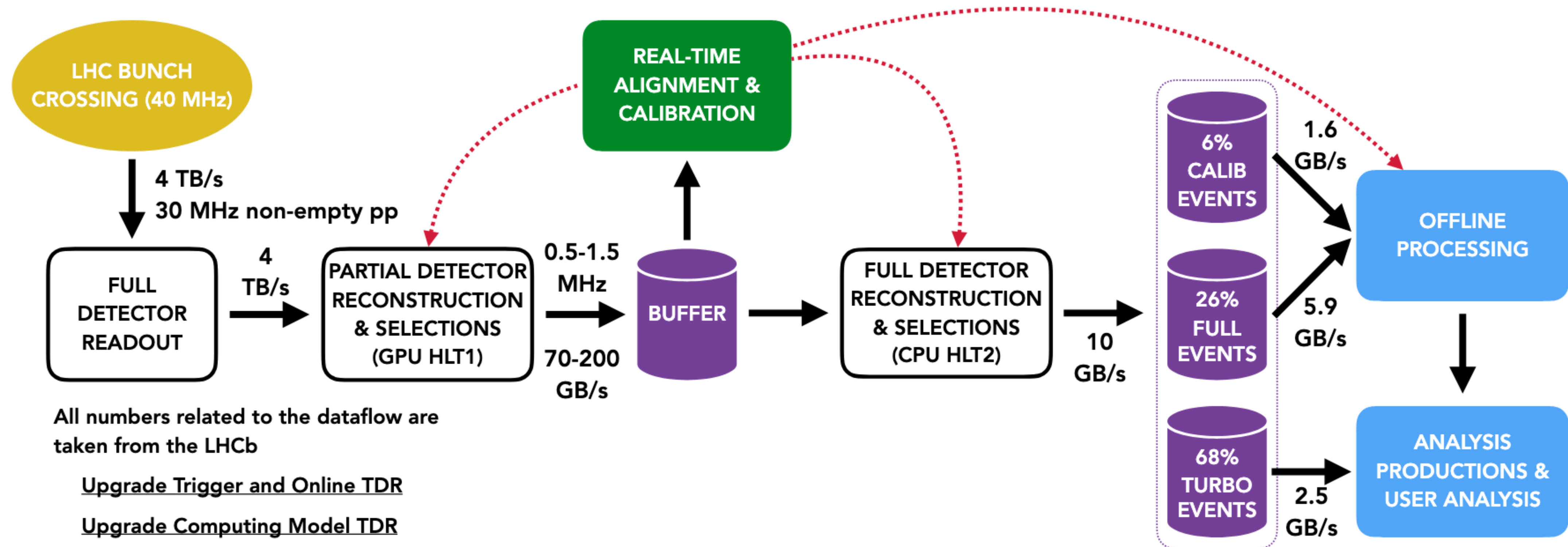
- Five times higher luminosity: $2 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$
- Increased pileup to $\langle \mu \rangle \approx 5$



- Replacement of all tracking detectors
- Removal of hardware trigger
- Fully new readout electronics to read out 40 Tbit/s
- Charged particle reconstruction at 30 MHz in the full detector is necessary

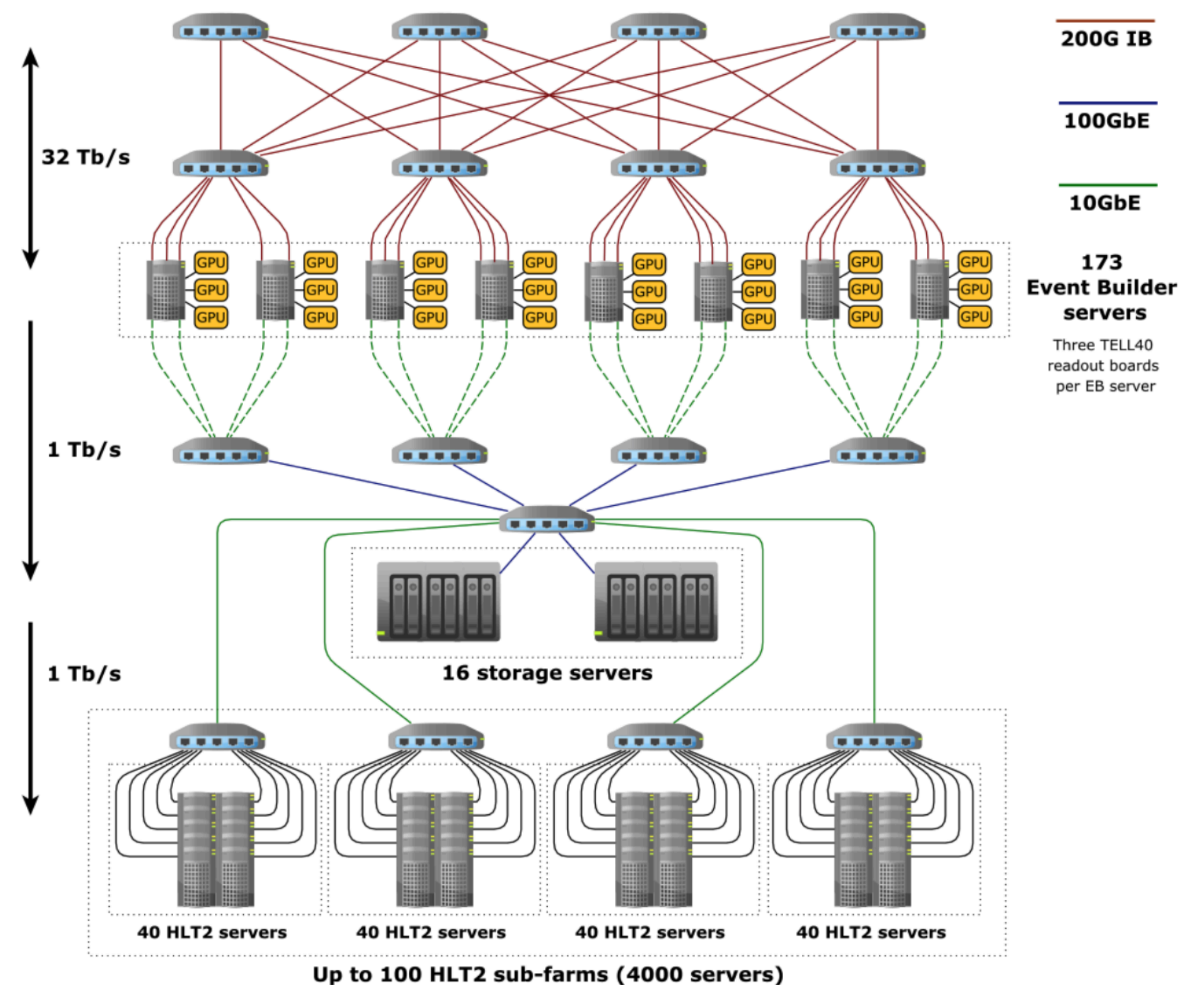


The LHCb trigger in Run 3



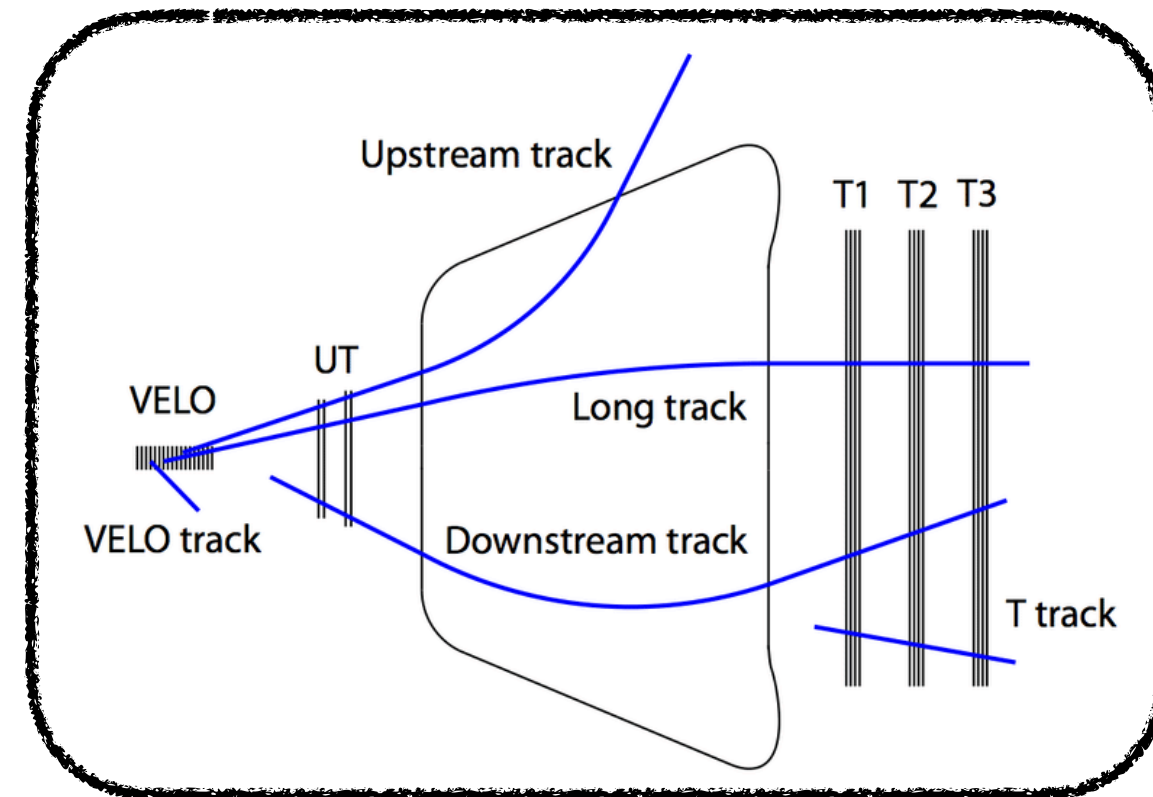
Heterogeneous computing system

- Raw detector data is received by FPGA cards
- 163 event builder servers aggregate data from sub-detectors
- Each server hosts 3 GPU cards on which HLT1 is run
 - Around 500 Nvidia A5000 in total
- Output of HLT1 is stored on storage servers while alignment & calibration is processed
- HLT2 is processed on computing farm with 250k CPU cores

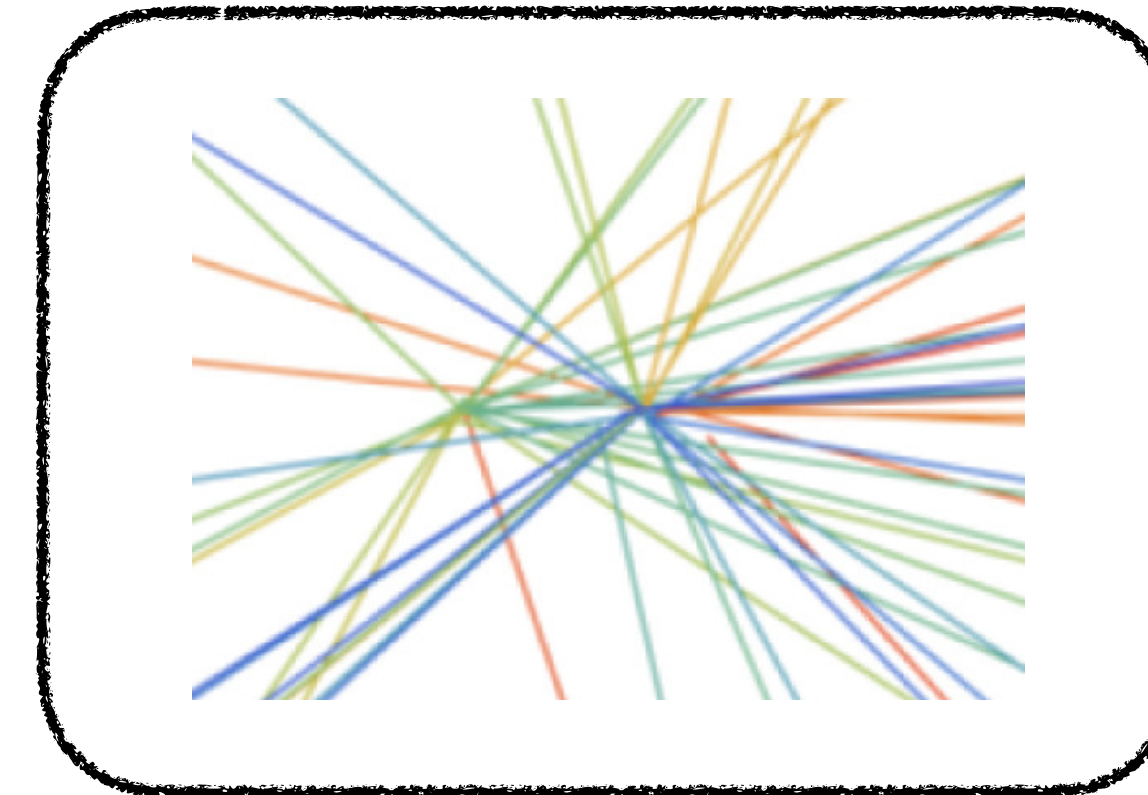


What do we reconstruct in the High Level Trigger (HLT)?

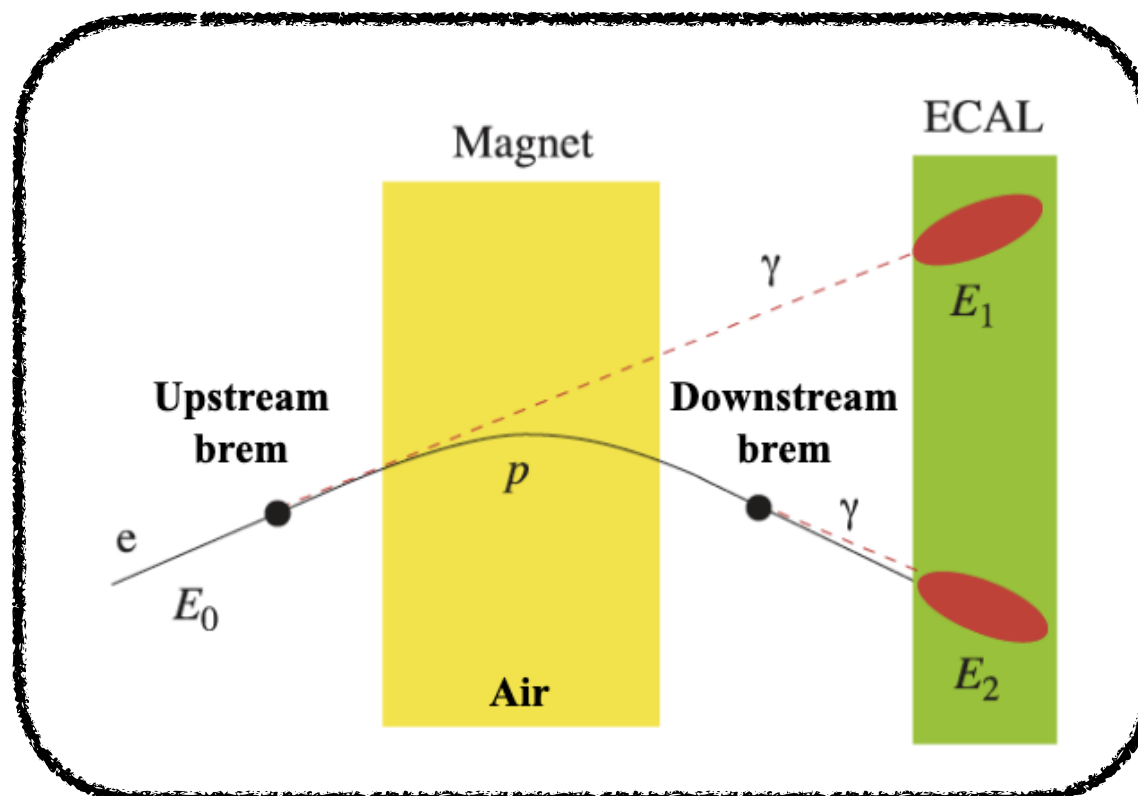
Tracks



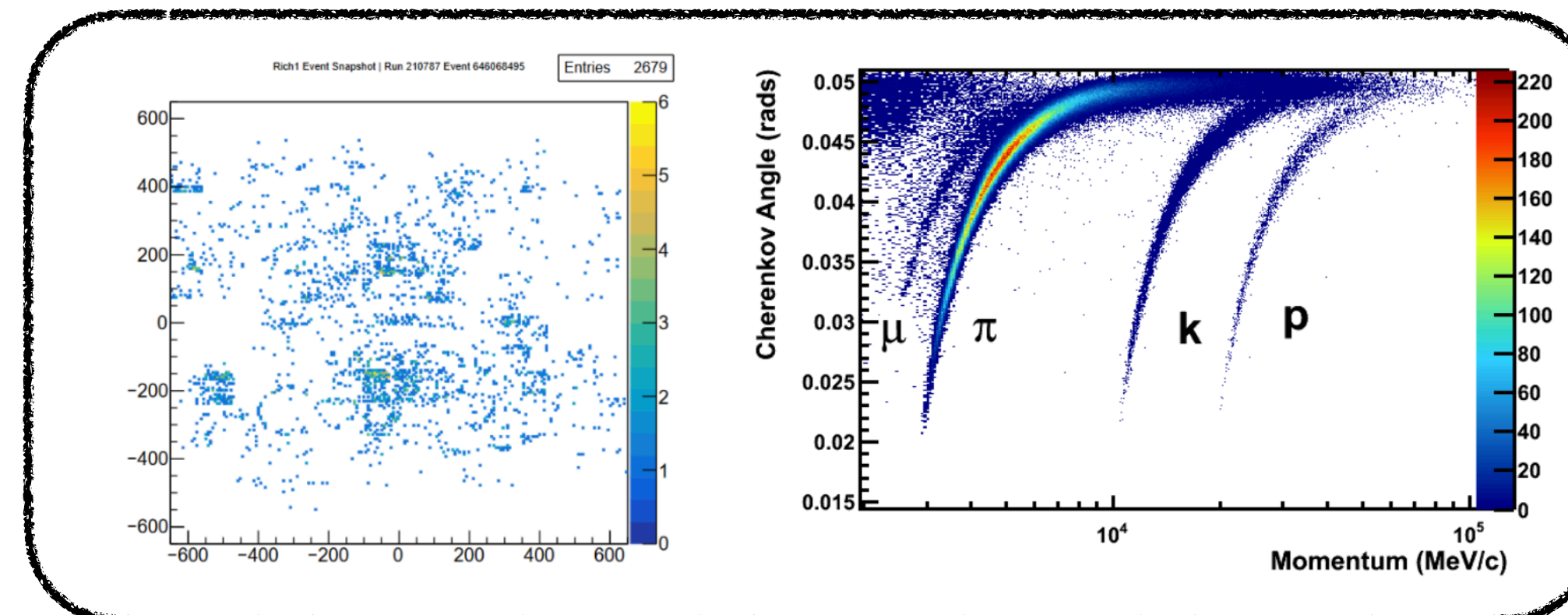
Vertices



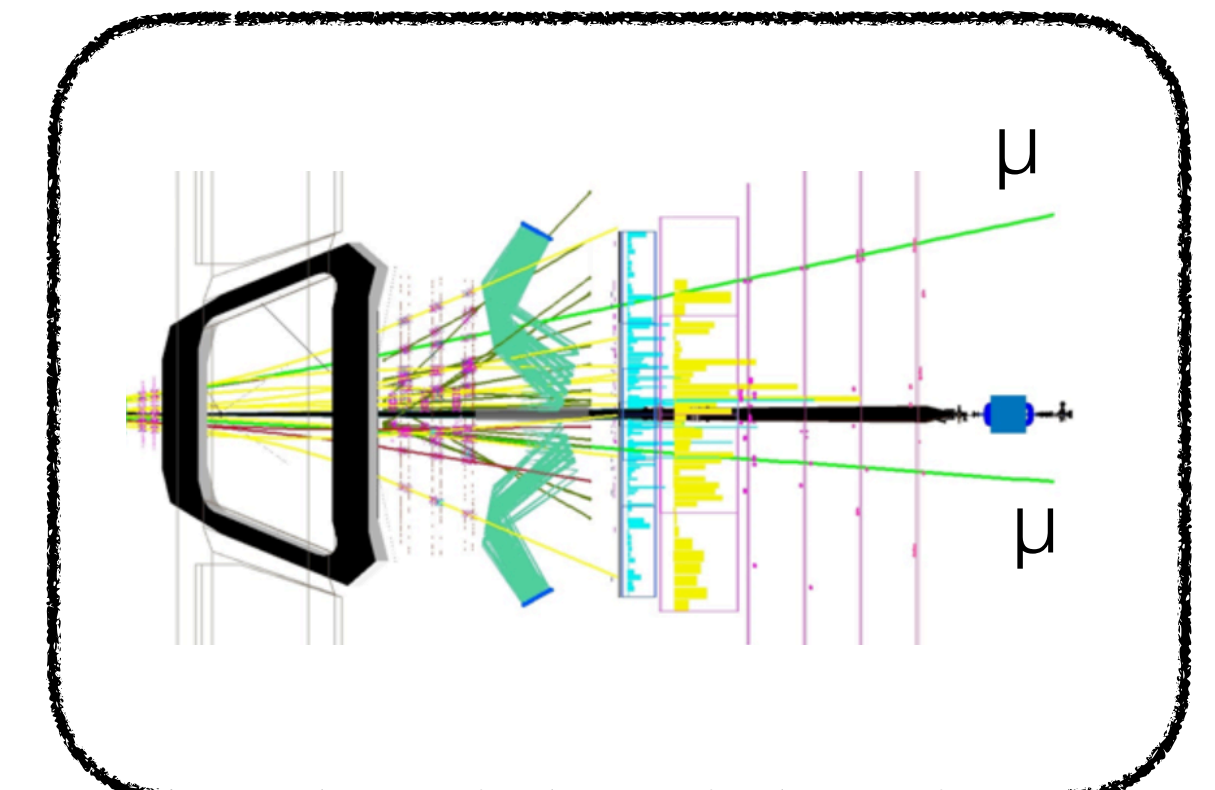
Calorimeter objects



Cherenkov rings



Muon hits



How does the HLT map to GPUs?

Characteristics of LHCb's HLT	Characteristics of GPUs
Reconstruction is intrinsically parallel problem: Process collisions and objects in parallel	Suited for: • Data-intensive parallelizable applications • High throughput applications
Huge compute load	Many TFLOPs available
Full data stream is read out → No stringent latency requirements	Higher latency than CPUs Not as predictable as FPGAs
Small raw event data (~100 kB)	• Connection via PCIe → limited I/O bandwidth • Thousands of events fit into O(10) GB of memory

The Allen software framework

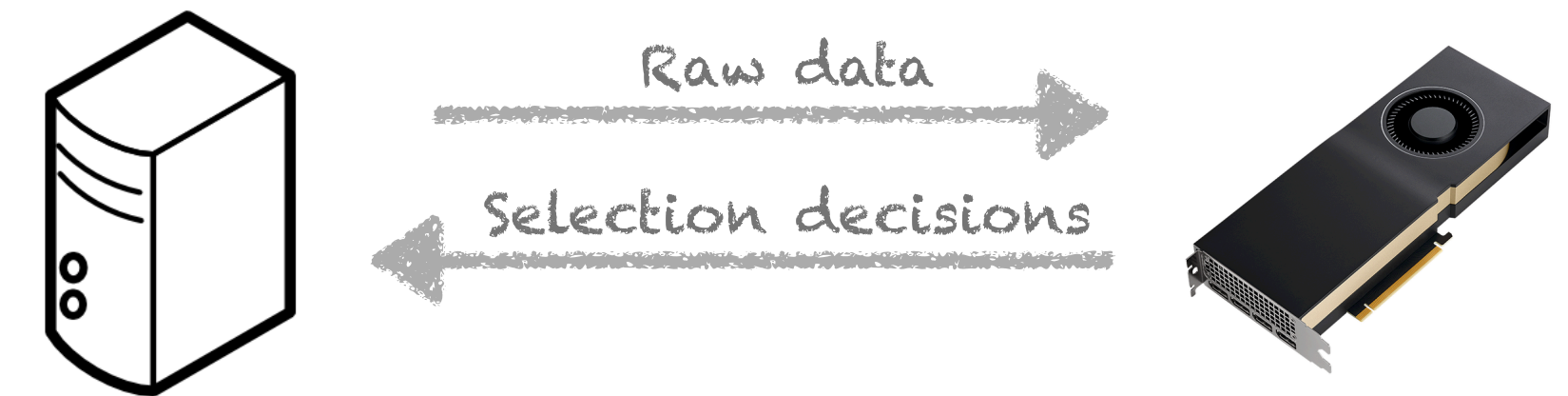
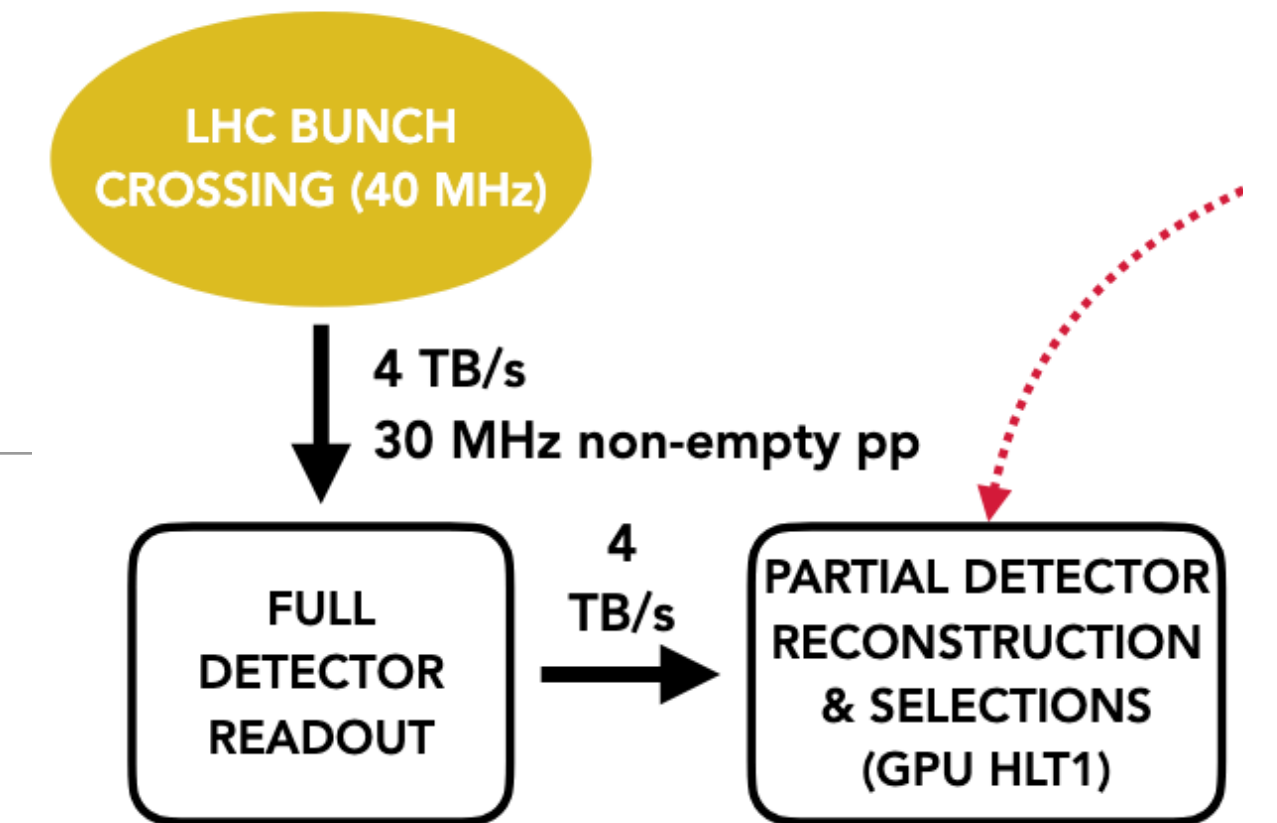
- Heterogeneous high throughput software framework
- Designed for LHCb's HLT1 in Run 3
- Built with CMake
- Single source code, runs on x86 and GPUs (Nvidia and AMD)
 - Portability between architectures provided by macros and few simple coding guide lines
- Memory manager for efficient usage of GPU memory
- Multi-event scheduler for batched execution
- Configuration via python
- Monitoring for development and data-taking
- Geometry loaded from DD4Hep, converted to simple structs for easy use on GPU



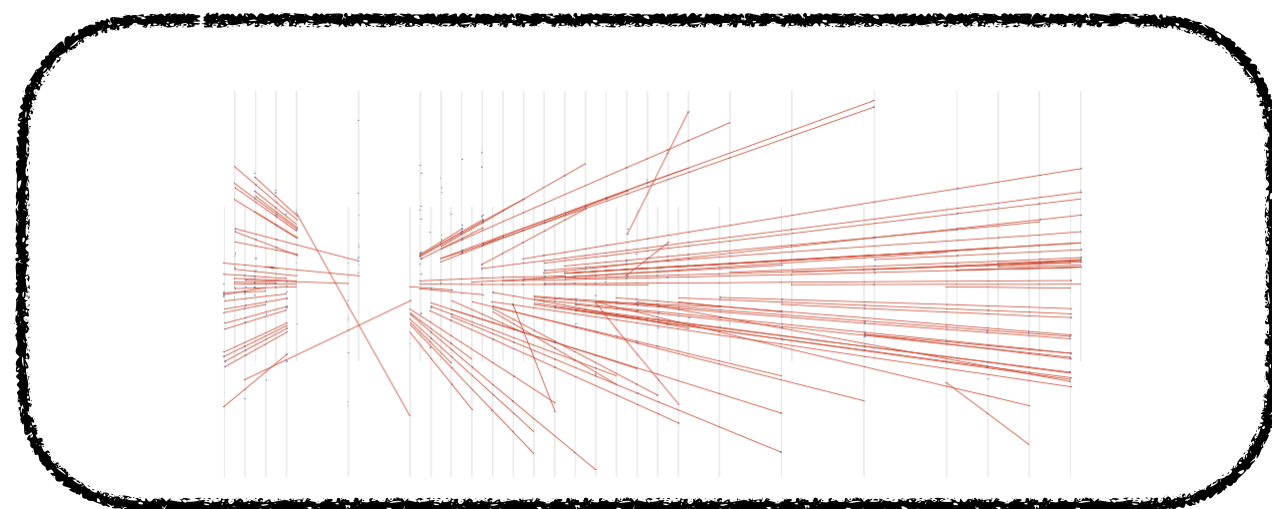
Named after Frances E. Allen

Design of the Allen GPU HLT1

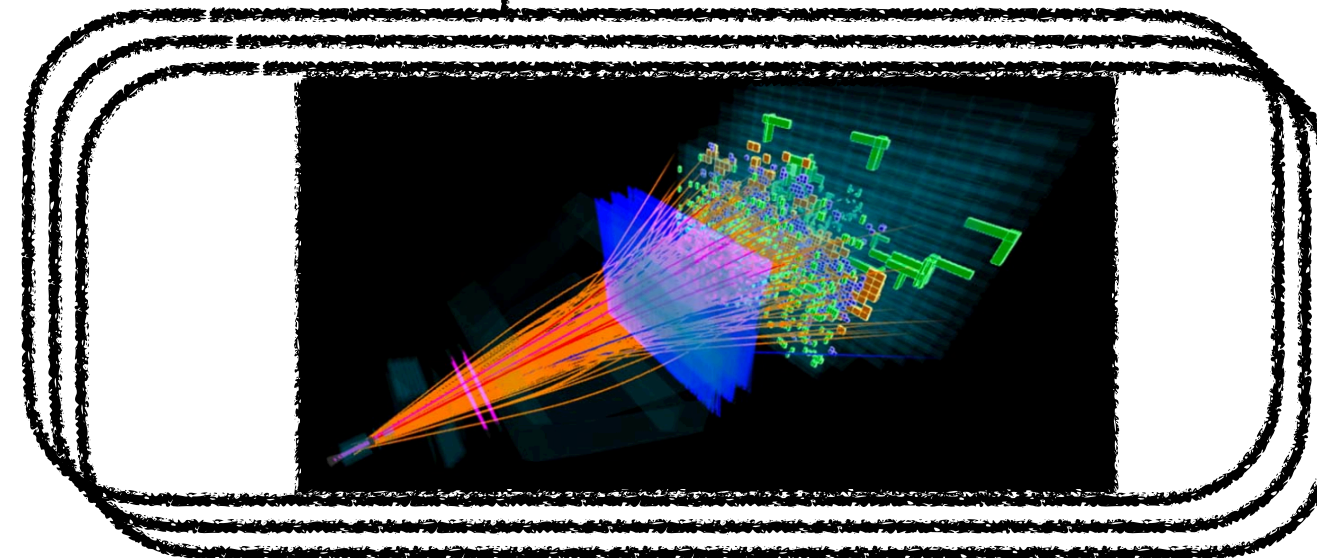
- Do all work on the GPU
 - Minimise copies to/from the GPU
- Parallelise on multiple levels
- Maximise GPU algorithm performance
 - Design software framework to optimise algorithm throughput performance
 - Interleave ML/AI and classical algorithms
 - Single precision only
- Simple event model



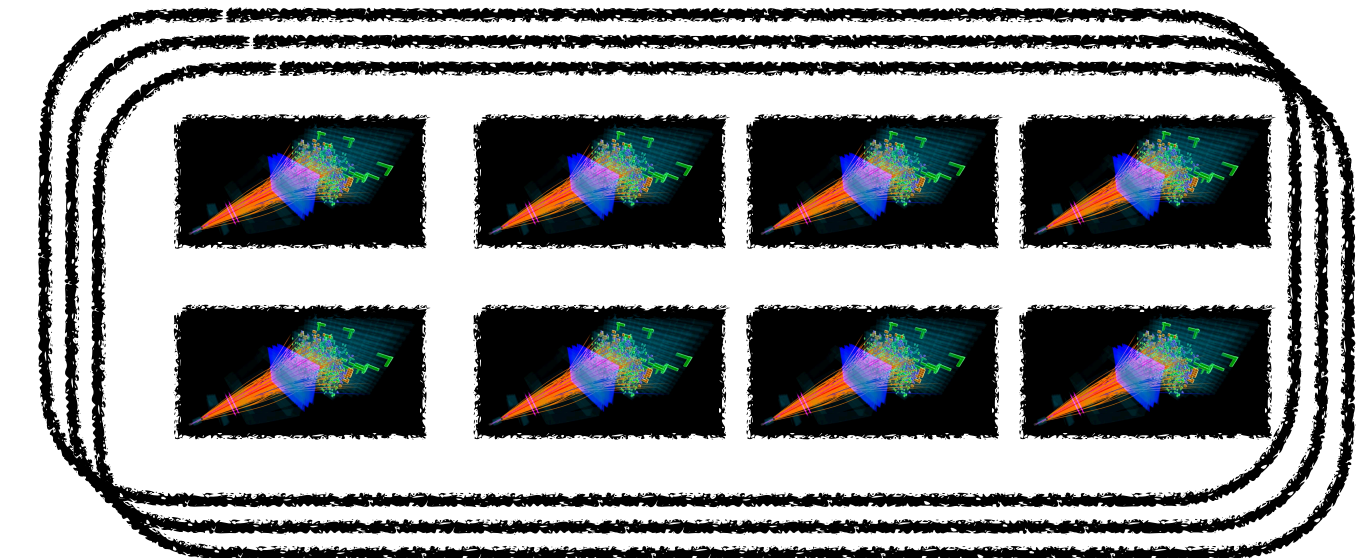
Intra collision: tracks, vertices, ...



Proton-proton collisions

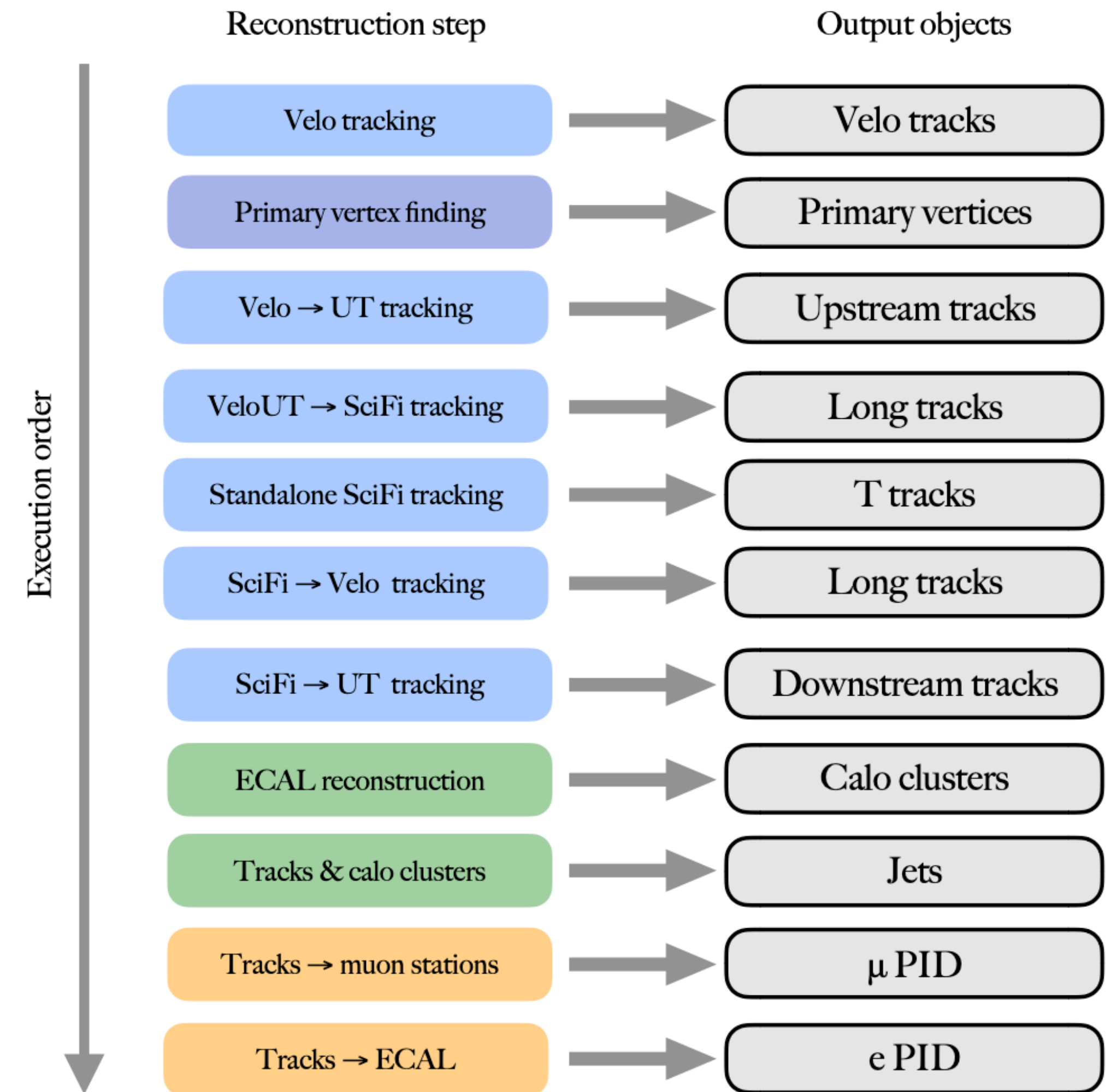
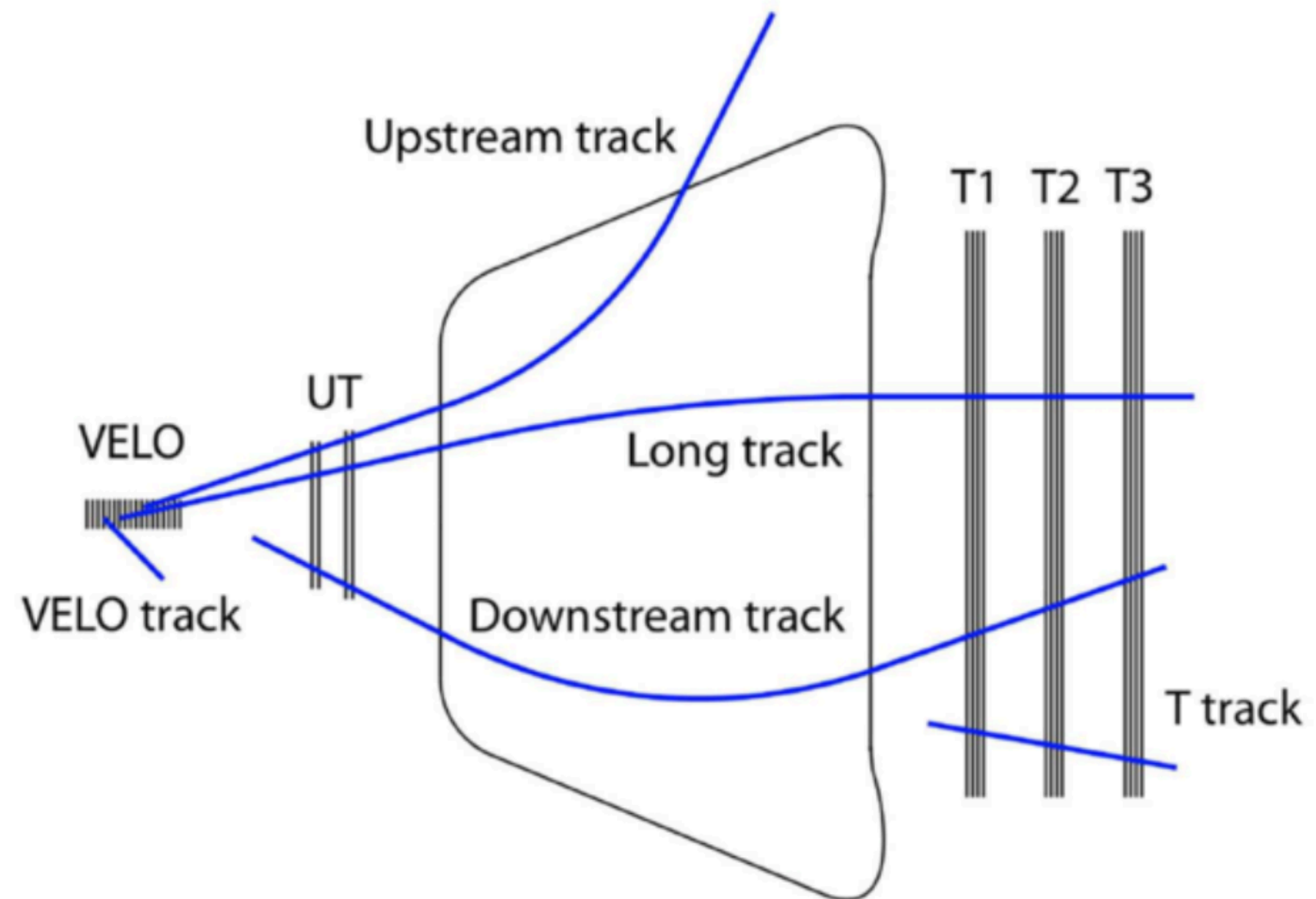


Collision batches



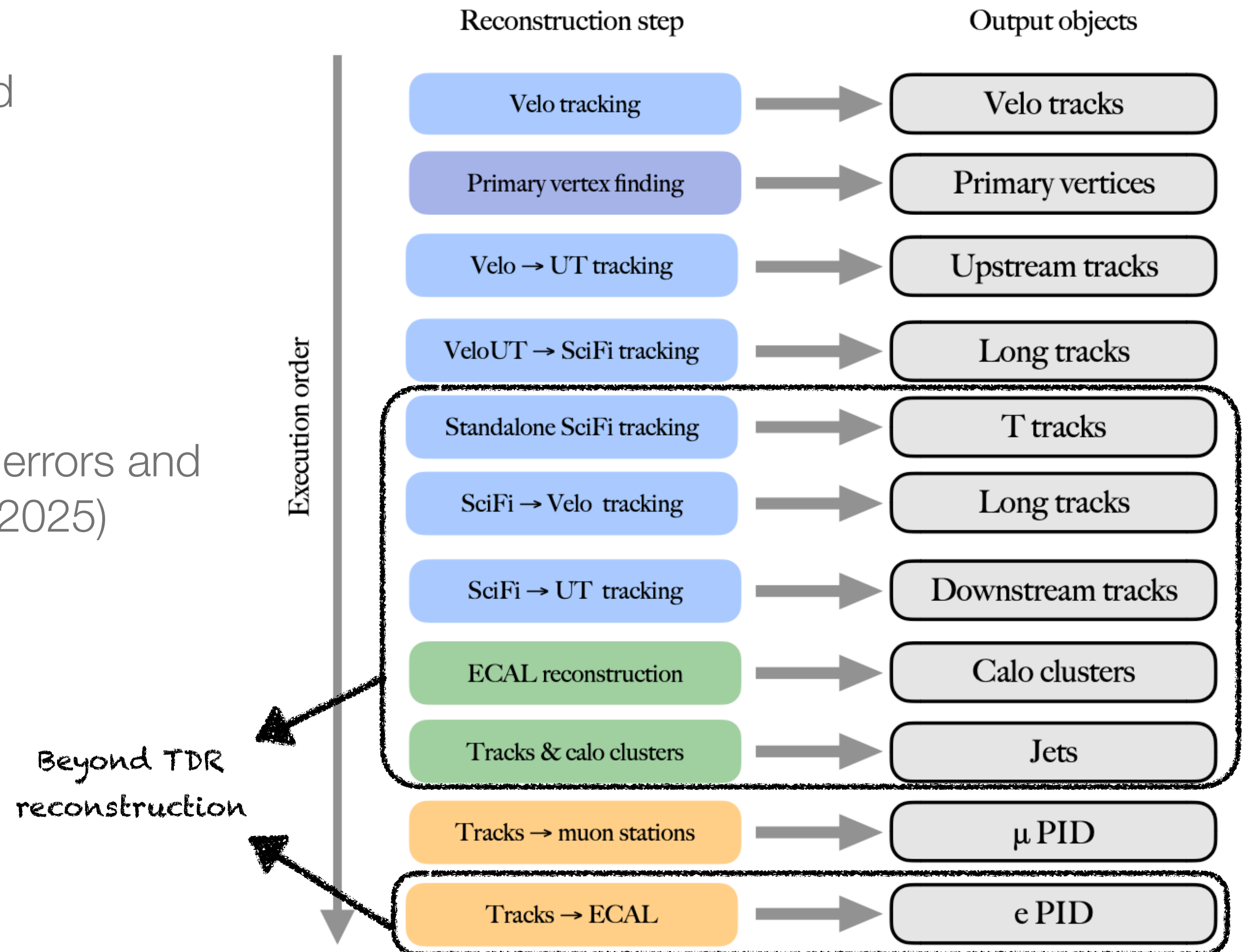
HLT1 reconstruction

- Track reconstruction
- Calorimeter reconstruction
- Lepton identification

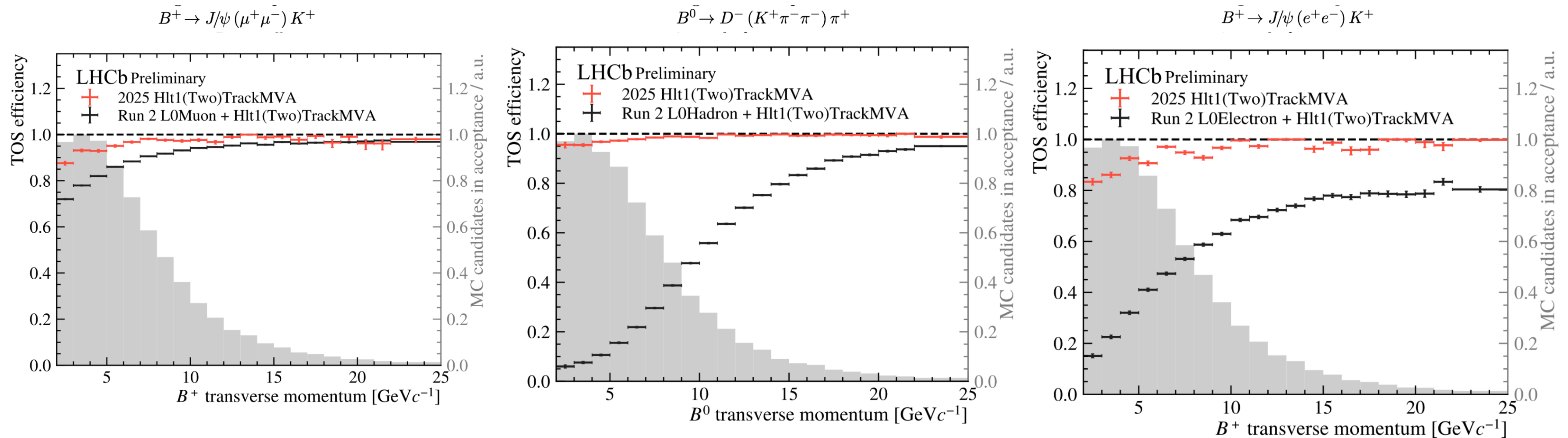


HLT1 reconstruction beyond Technical Design Report (TDR)

- Additional long track reconstruction method
 - Gained 20% efficiency at $p_T = 500$ MeV
 - Robust with respect to detector occupancy
- Downstream track reconstruction
- Ecal & Jet reconstruction
- Kalman filter with parameterised scattering errors and transport through the magnetic field (since 2025)

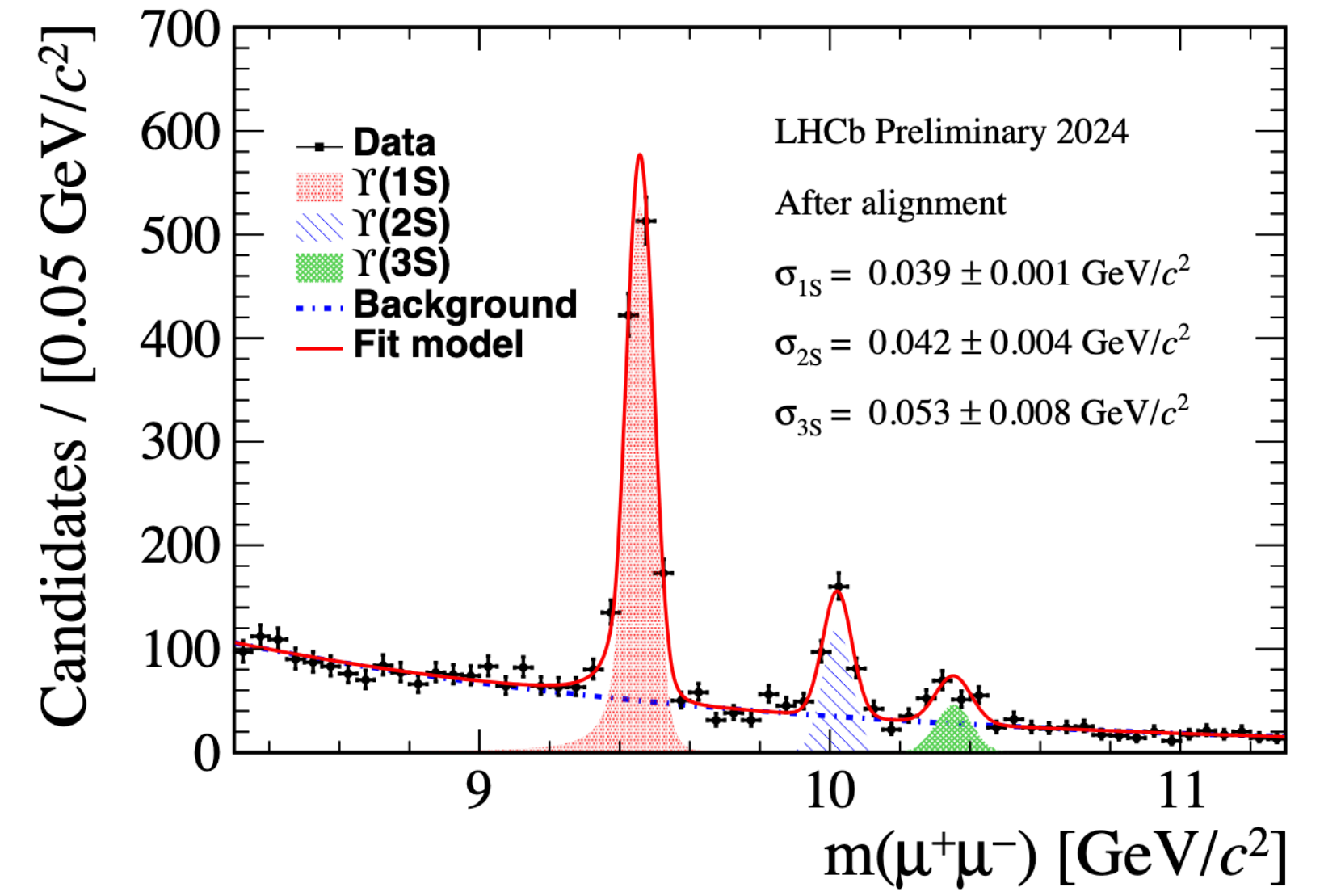
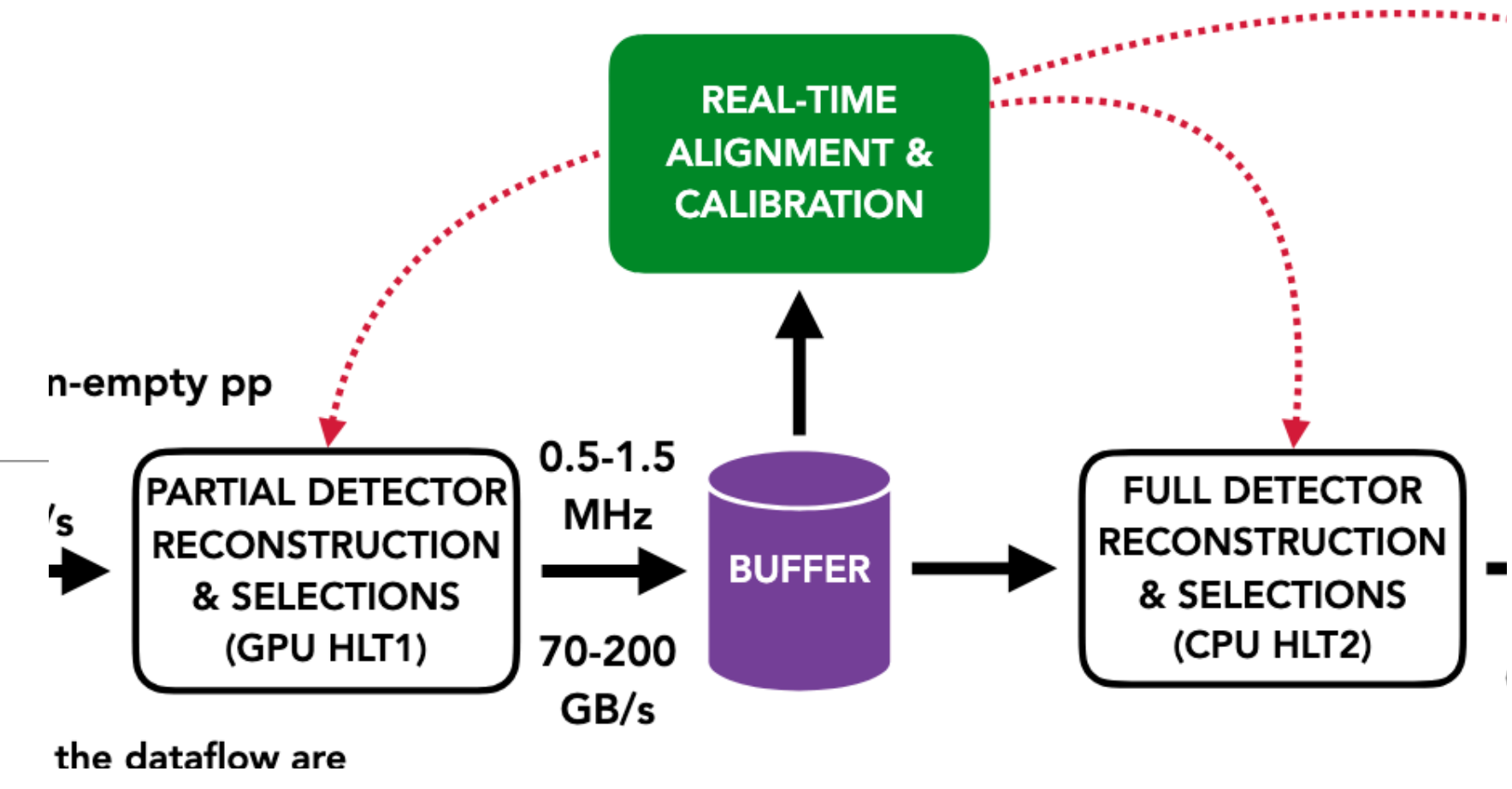
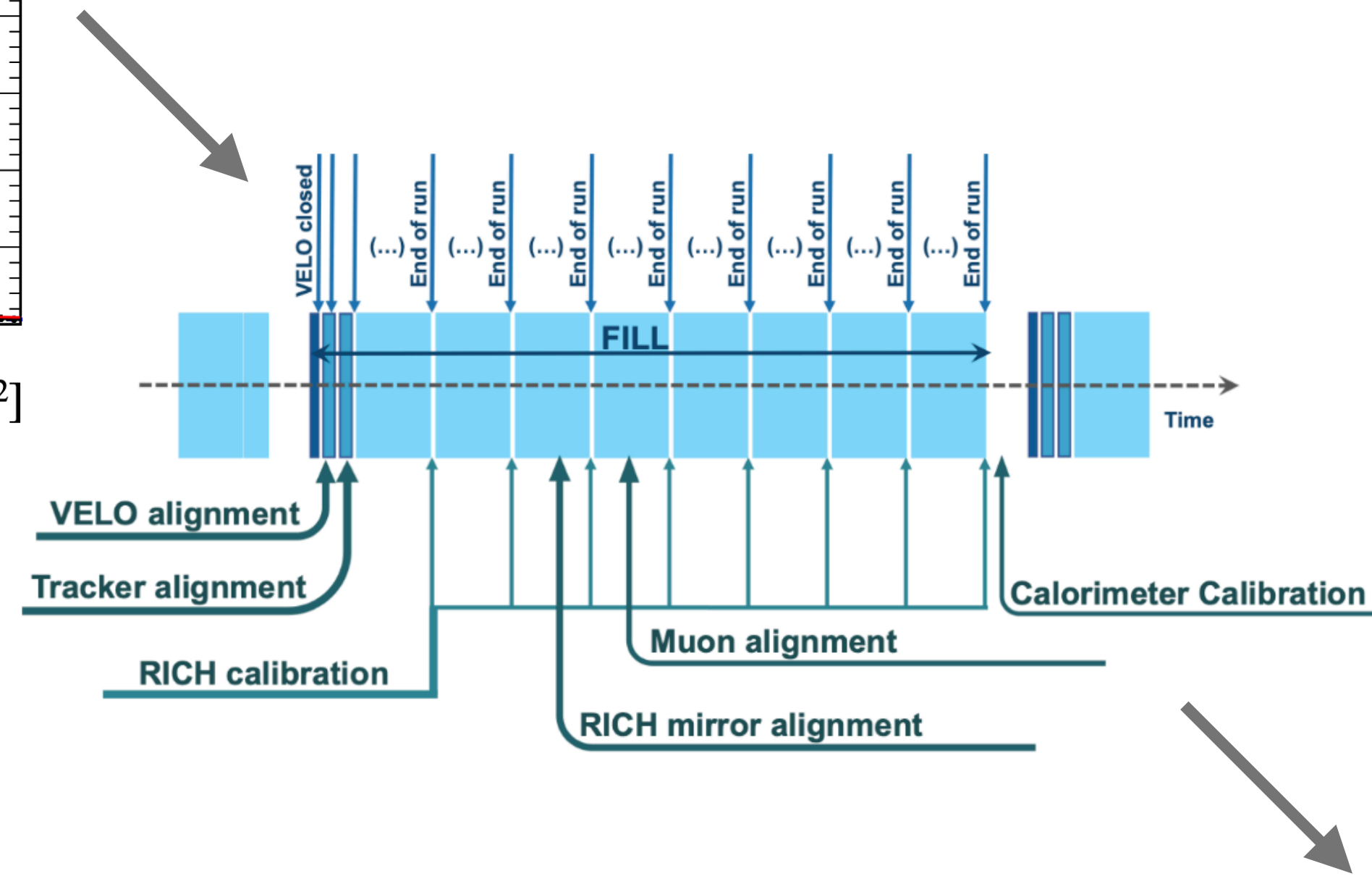
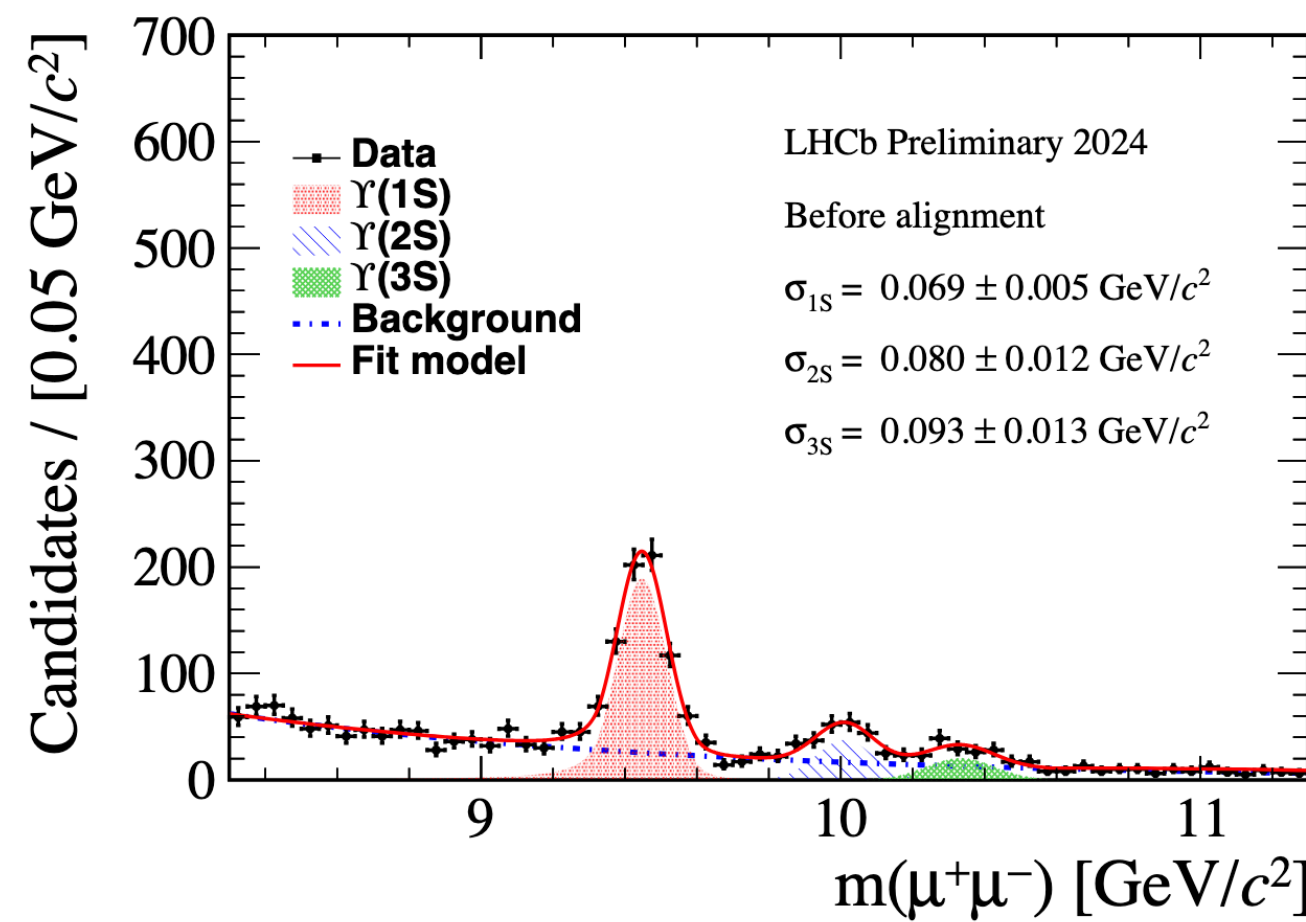


HLT1 trigger performance



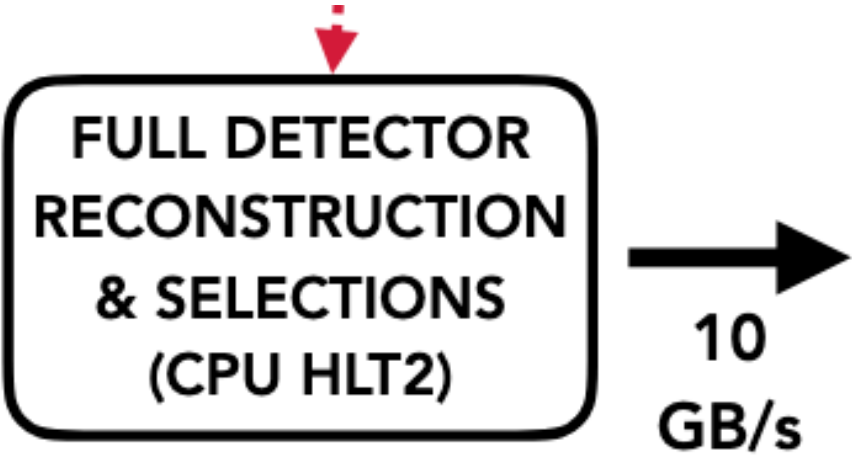
- Heterogeneous software trigger has been a full success
- HLT1 trigger performance greatly surpasses TDR goals set in 2014 and 2020

Real-time alignment and calibration



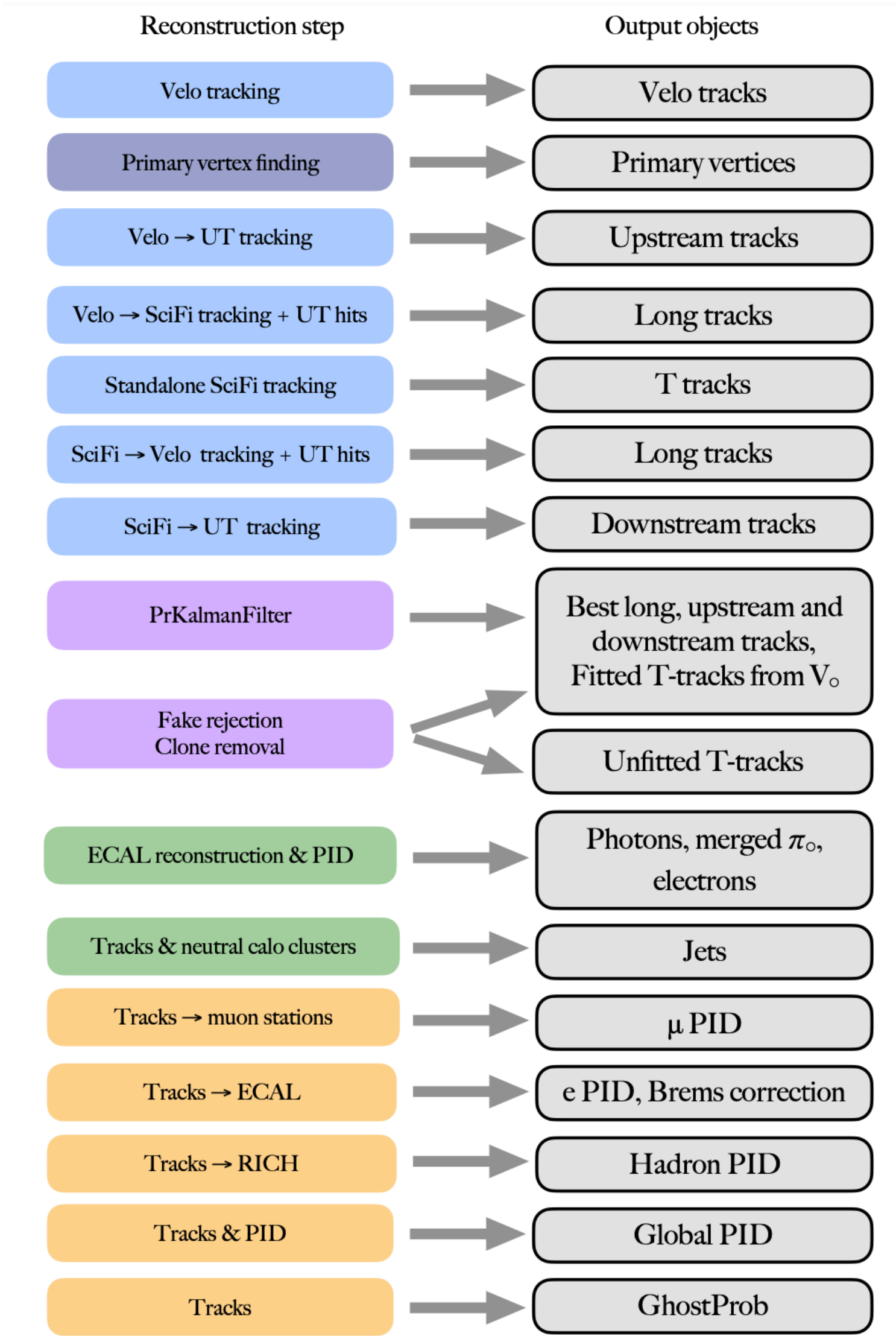
- Crucial for offline-quality HLT2 reconstruction
- Updated constants are also propagated to HLT1

HLT2 reconstruction



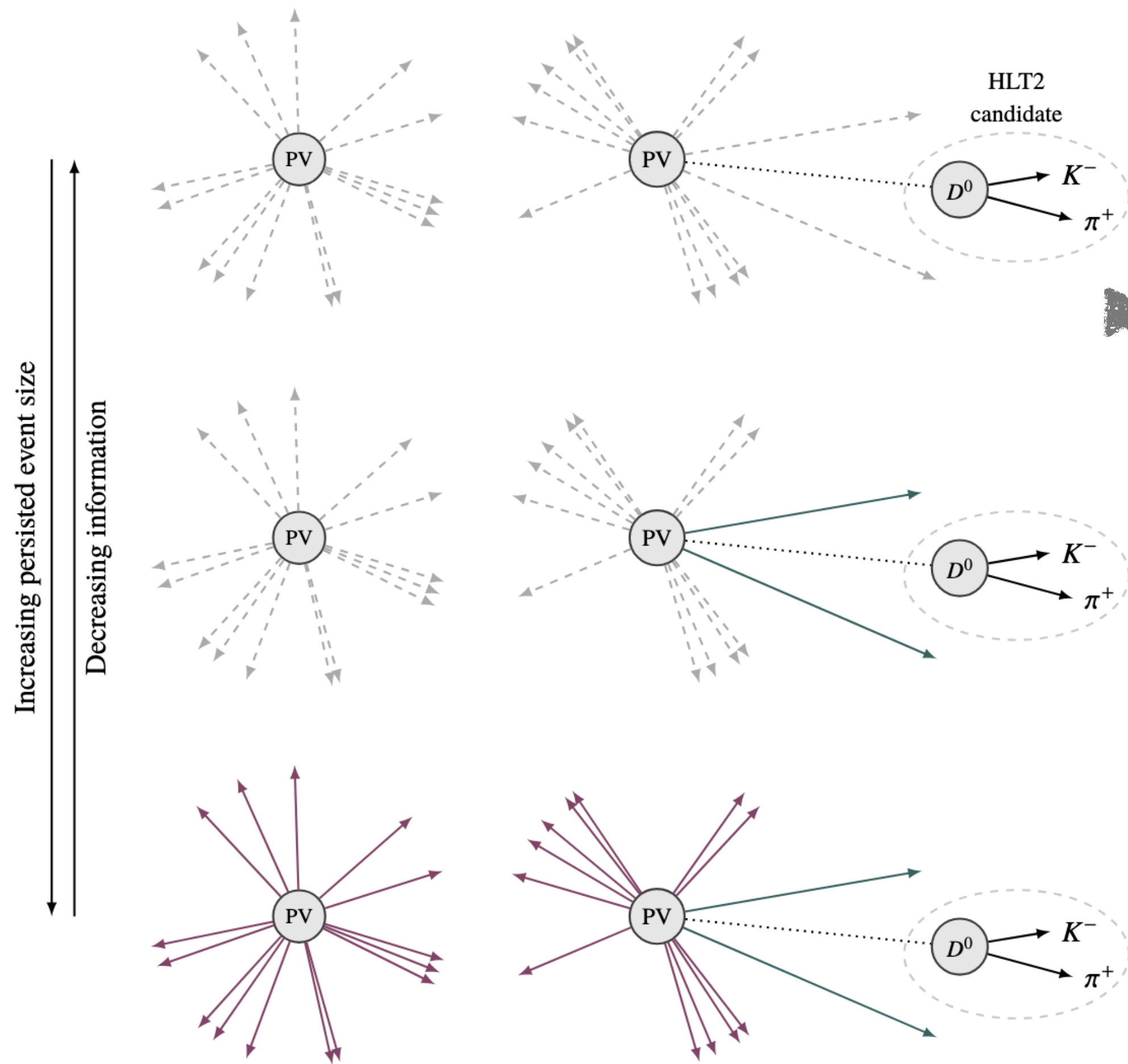
Offline quality reconstruction

- Track reconstruction algorithms tuned for efficiency
- Kalman filter with parameterised scattering errors
- Hadron and global PID



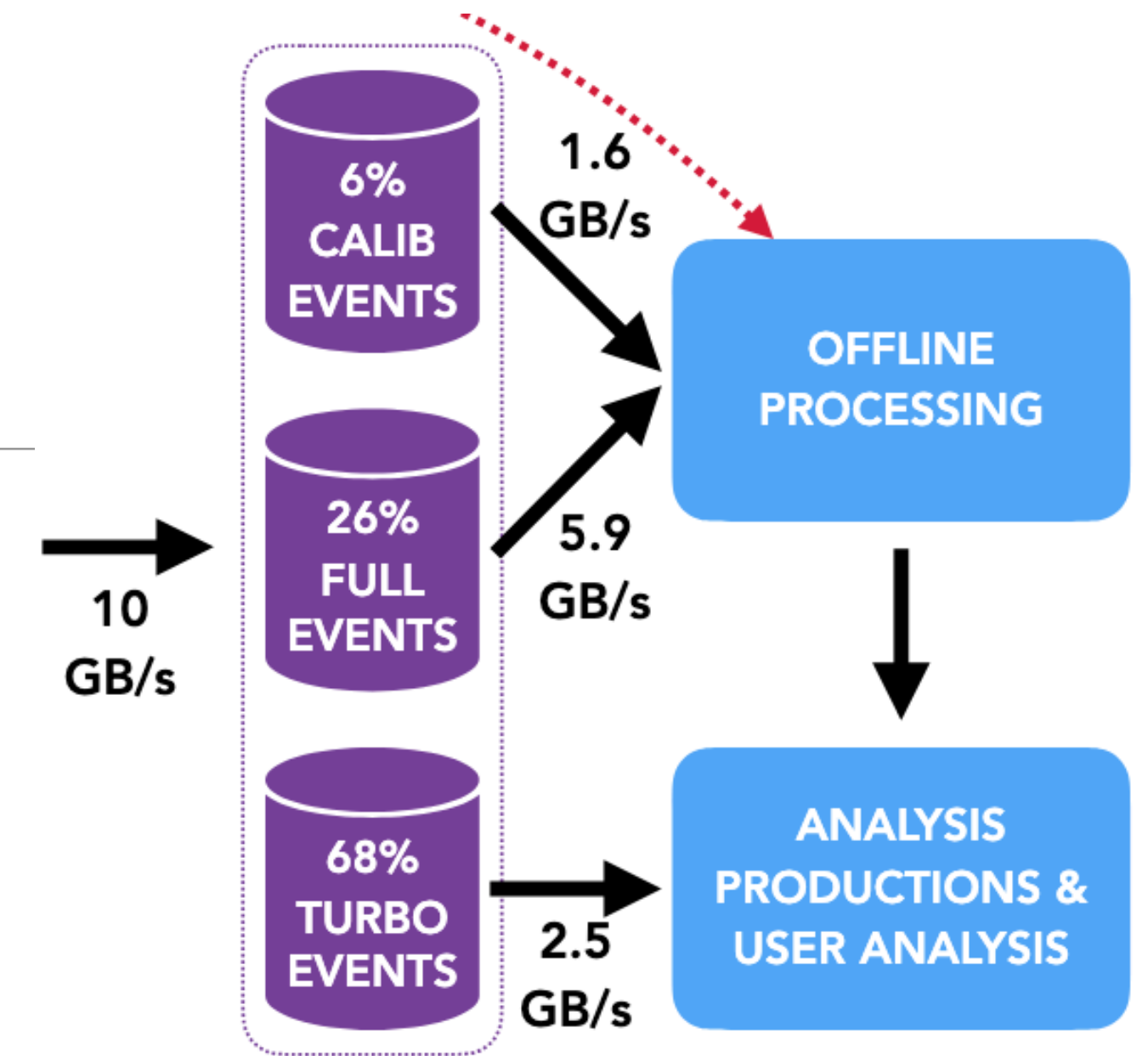
HLT2 persisted output

JINST 14 (2019) P04006



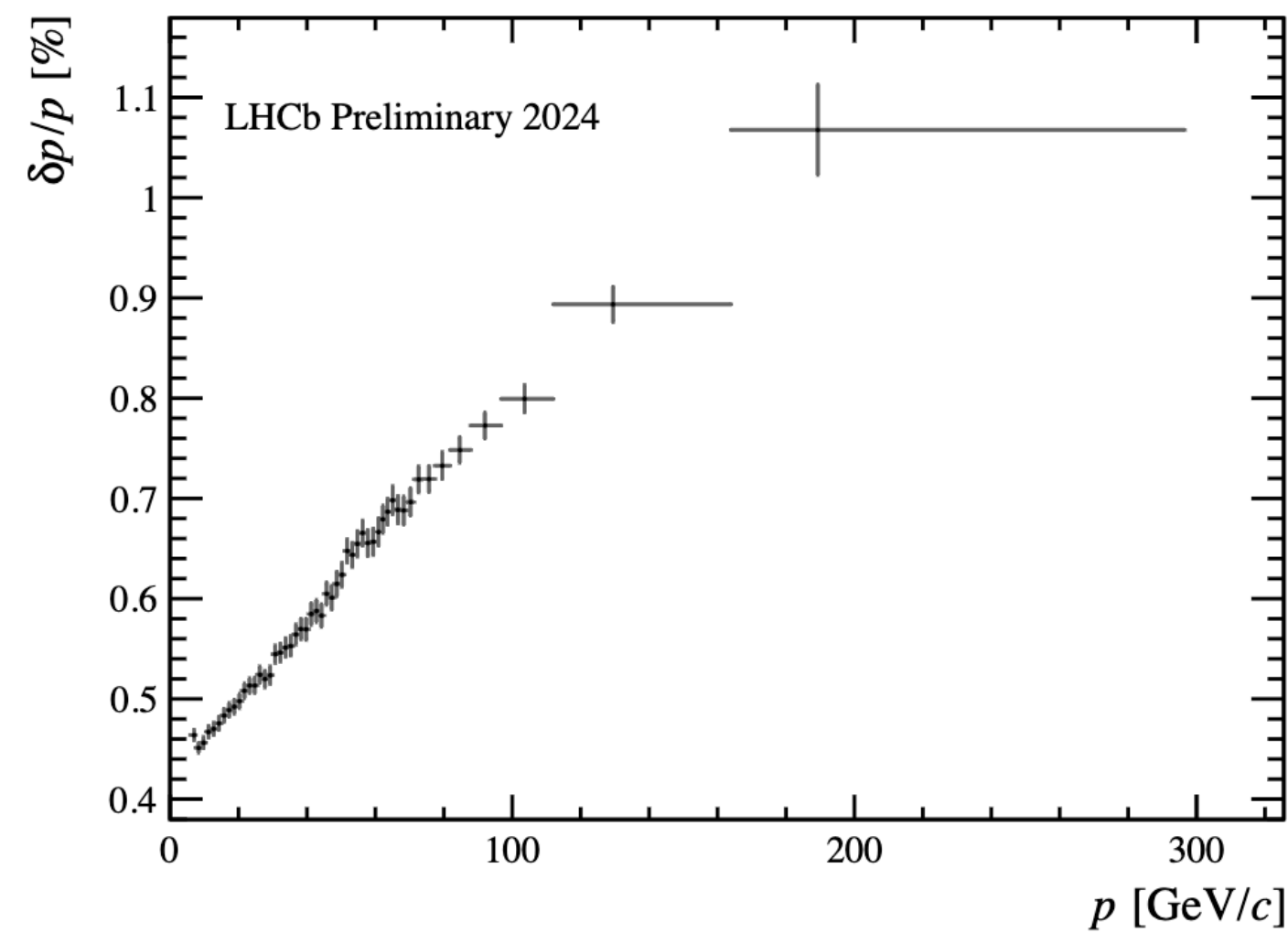
Three levels of persistency

- Turbo: Save only information related to signal candidate reducing event size by a factor 10
- Selective persistency: Save additional objects
- Full persistency



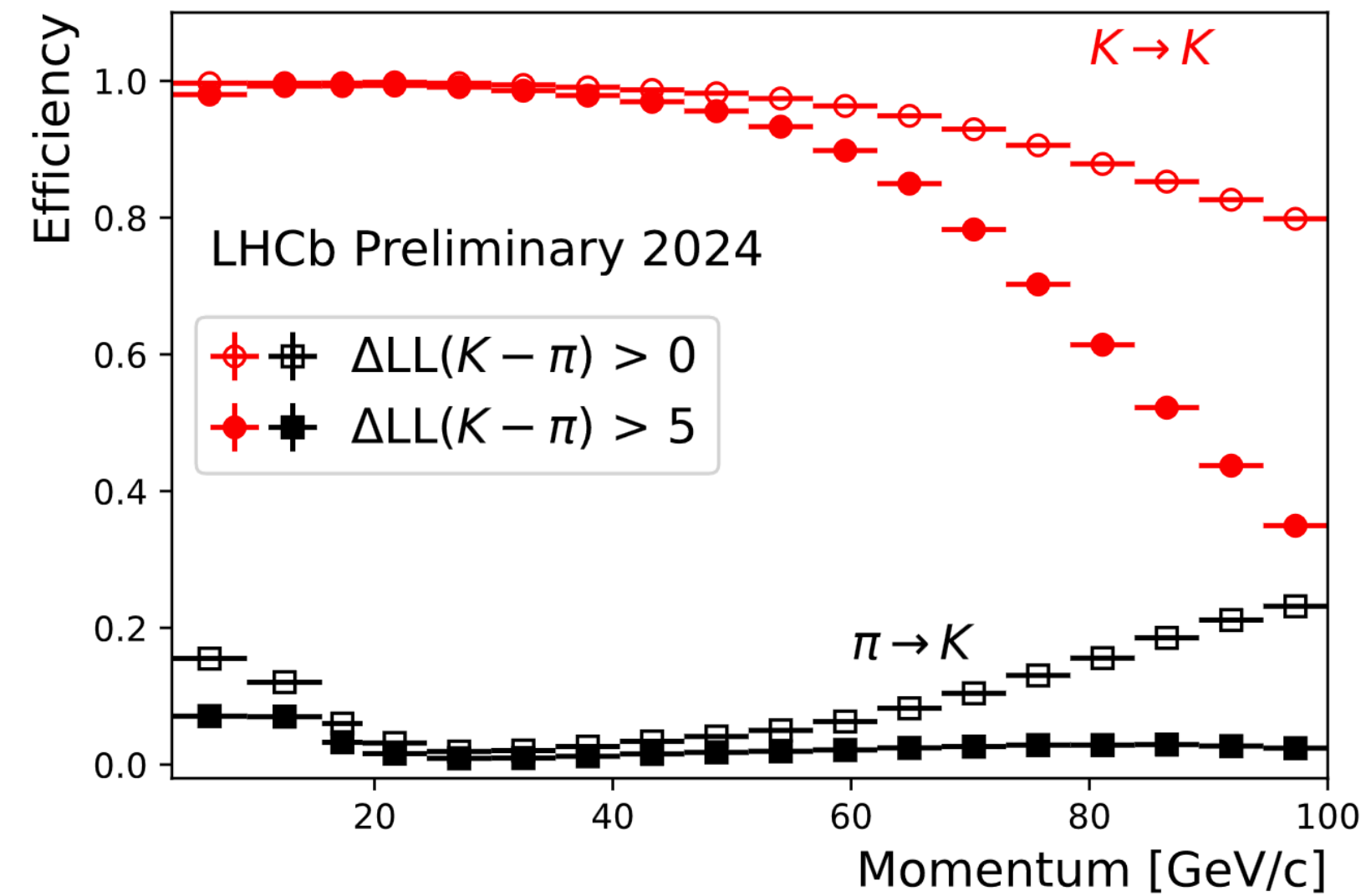
LHCb trigger performance during Run 3

Momentum resolution similar as in Run 2



LHCb-FIGURE-2024-040

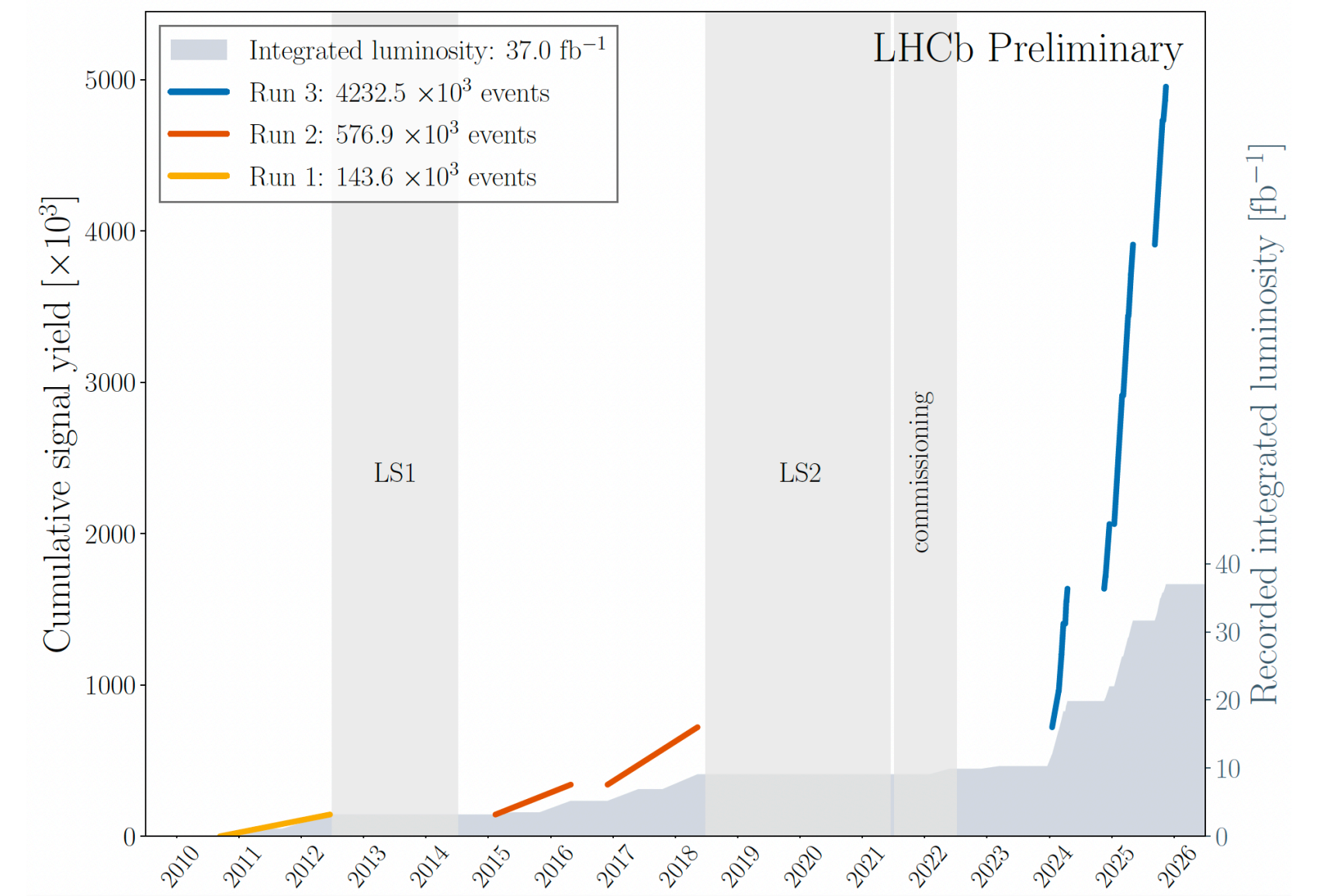
Hadron PID outperforms Run 2



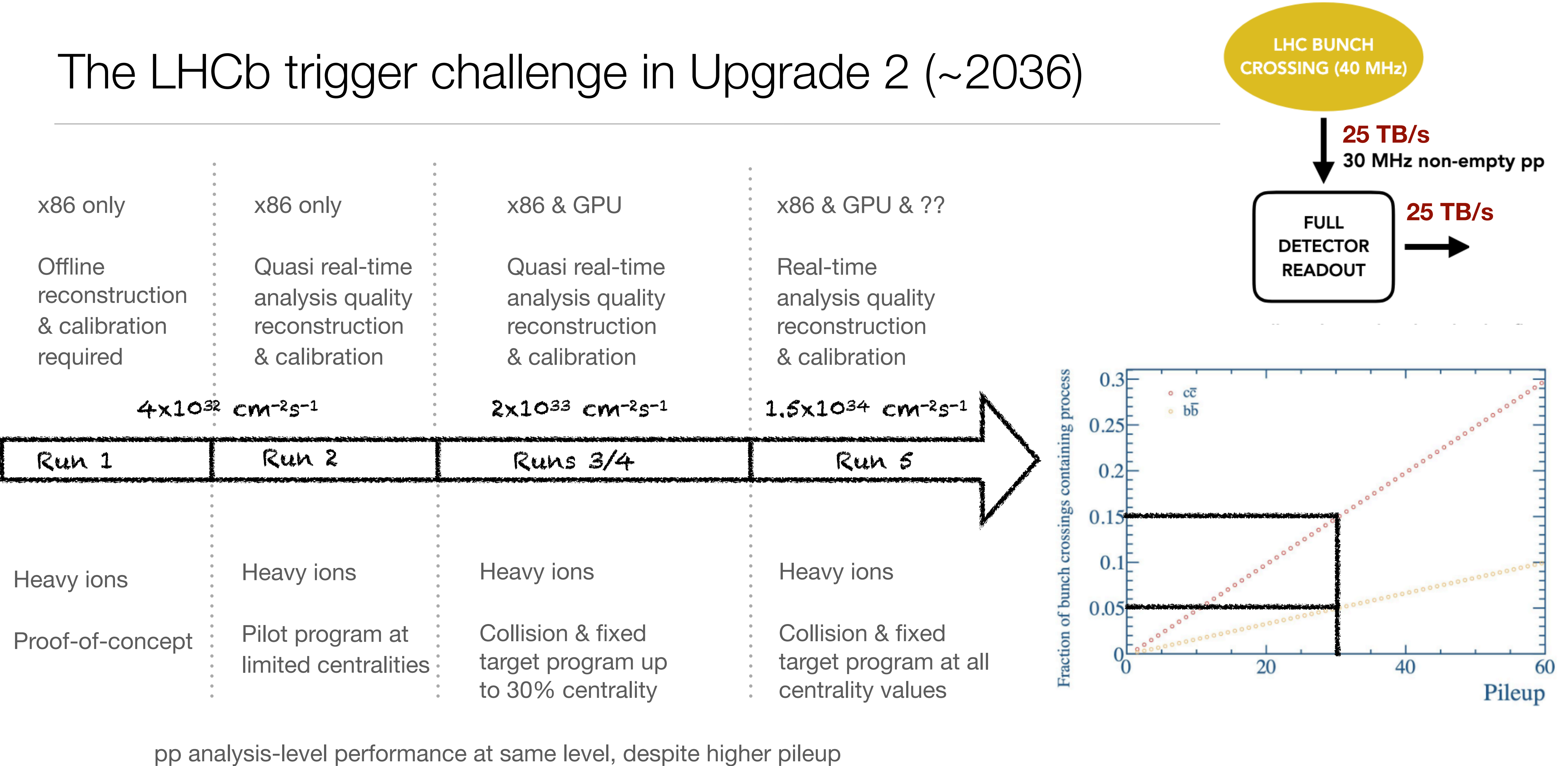
LHCb-FIGURE-2024-031

Yield per luminosity outperforms Runs 1&2 for several decay channels

$$B^\pm \rightarrow K^\pm J/\psi, J/\psi \rightarrow e^+e^-$$

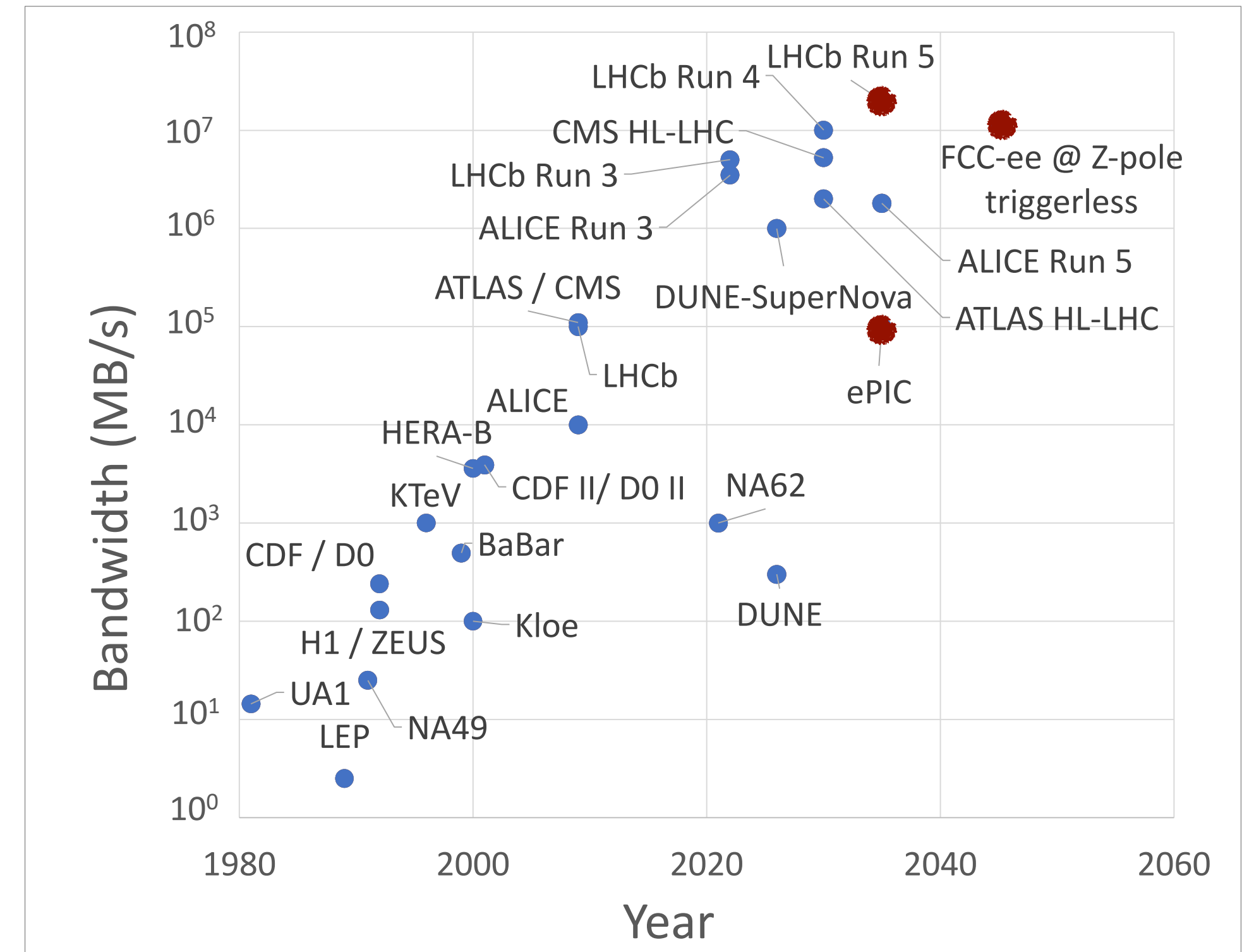


The LHCb trigger challenge in Upgrade 2 (~2036)



Processing 25 TB/s

- Only currently viable option is to process the full reconstruction on GPUs
- Best suited architecture to process thousands of selections still to be determined
- Split (or no split) between HLT1 and HLT2 still to be determined
- Adapt currently used Allen and Gaudi software frameworks to cope with the challenges
- ML/AI techniques increasingly important for reconstruction, classification, selection
- HEP-specific challenge: Fast inference at 30 MHz on GPUs, FPGAs and emerging technologies



CERN-ESU-2025-001, <https://arxiv.org/abs/2511.03883>

LHCb Upgrade 2 trigger can act as technology driver and catalyst for future experiments

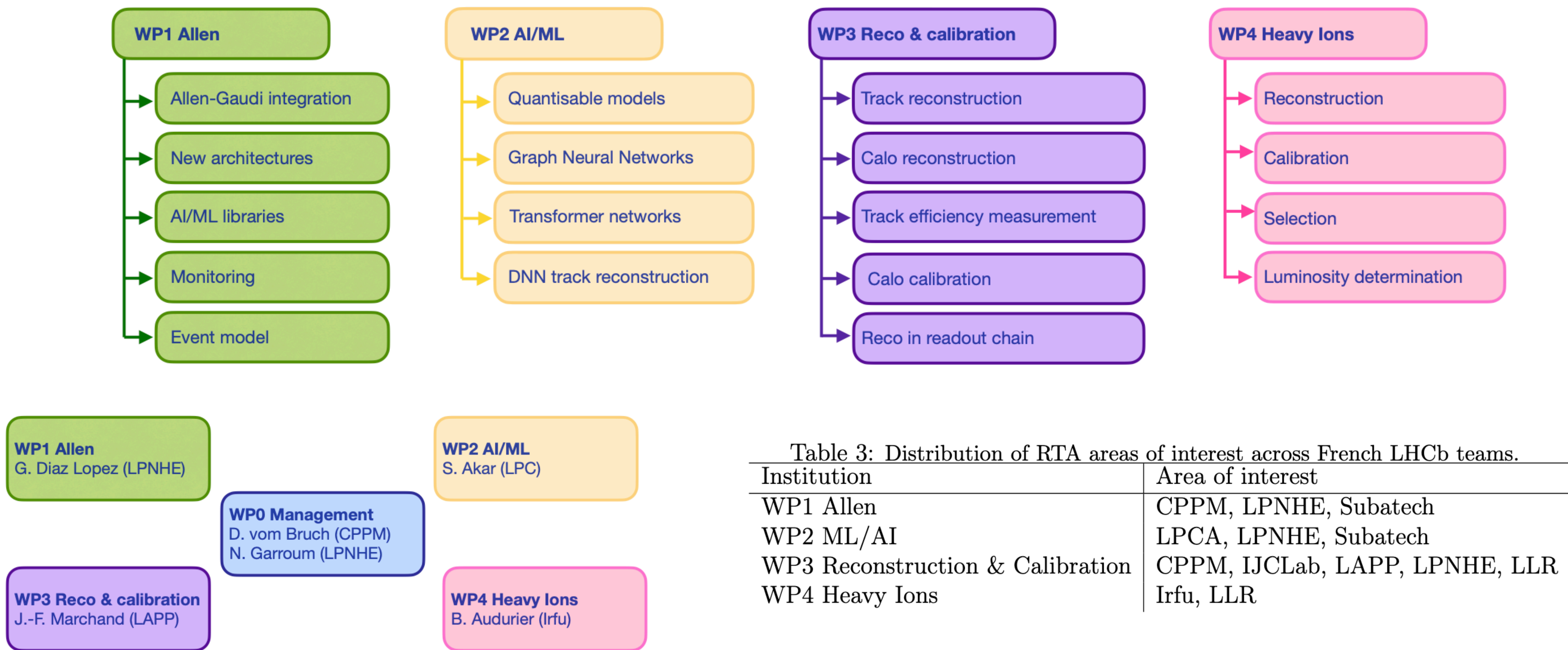
Path from 5 TB/s to 25 TB/s

- Challenge 1: Process full reconstruction on GPU architectures
 - Design robust and maintainable framework capable of processing large selection of complex offline quality algorithms
 - Explore AI/ML methods for reconstruction and classification tasks
 - Design high-throughput algorithms to be processed on GPU architectures
 - Explore emerging architectures to choose most suited technology
- Challenge 2: Reconstruct most central heavy ion collision events
 - Need to reduce fake track rates in high multiplicity environment
 - Explore AI/ML methods and stand alone track seed methods

Path from 5 TB/s to 25 TB/s

- Challenge 1: Process full reconstruction on GPU architectures
 - Design robust and maintainable framework capable of processing large selection of complex offline quality algorithms
 - Explore AI/ML methods for reconstruction and classification tasks
 - Design high-throughput algorithms to be processed on GPU architectures
 - Explore emerging architectures to choose most suited technology
 - Challenge 2: Reconstruct most central heavy ion collision events
 - Need to reduce fake track rates in high multiplicity environment
 - Explore AI/ML methods and stand alone track seed methods
- Possible only with continued support of long-term team of software engineers and physicists for Allen project
 - Only two countries / institutions currently provide software engineers for RTA: CERN and France
 - CERN has no long-term FTE on Allen
 - Chance for France to play a leading role in designing and maintaining the U2 trigger framework

Proposed RTA-France project

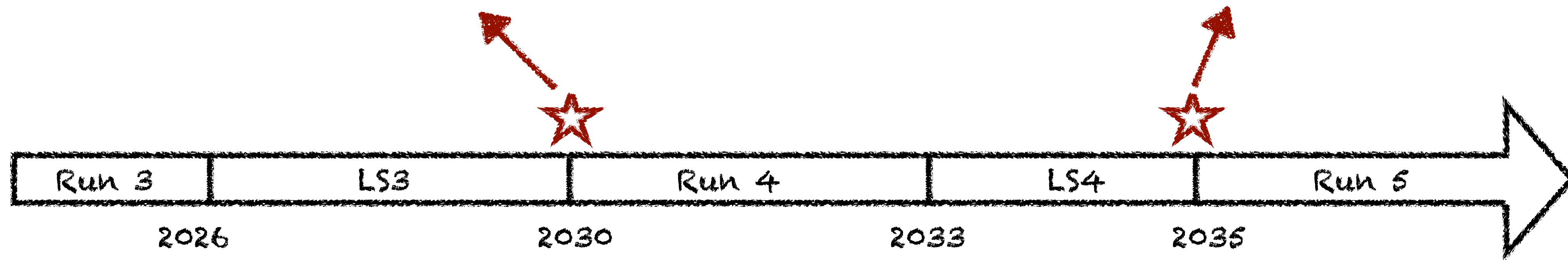


Deliverables

Generic developments

AllenCore depending on Gaudi:
Infrastructure for high-throughput heterogeneous
computing in experiment-independent framework

AllenCore extended for emerging architectures,
AI/ML support, energy-efficiency evaluation



LHCb-specific

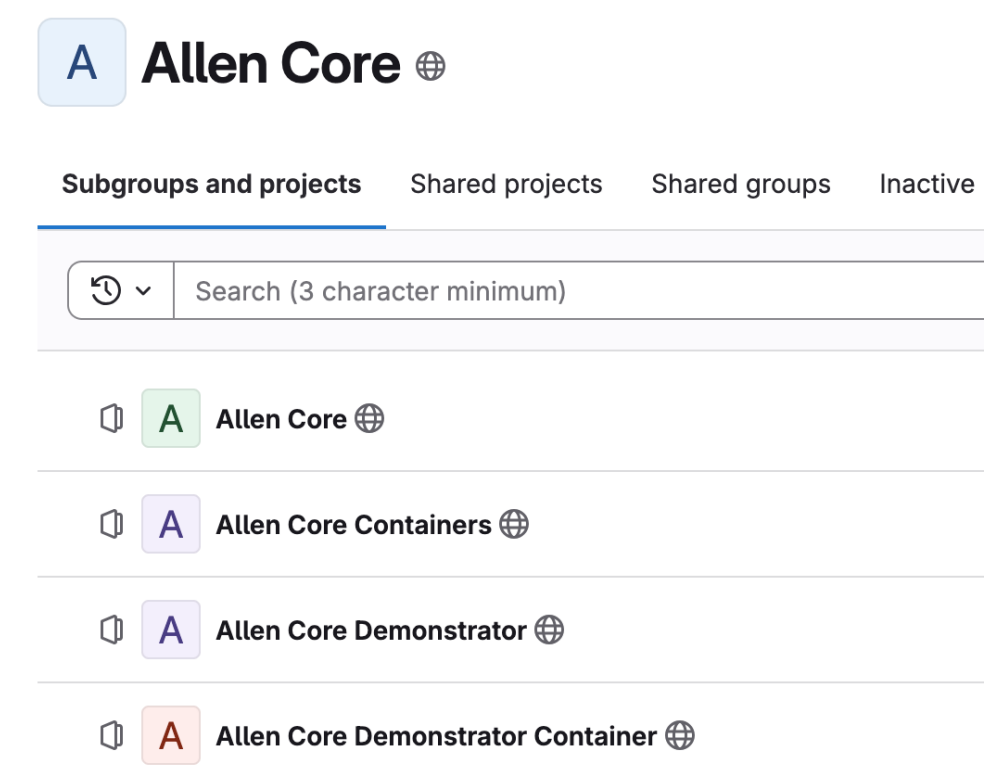
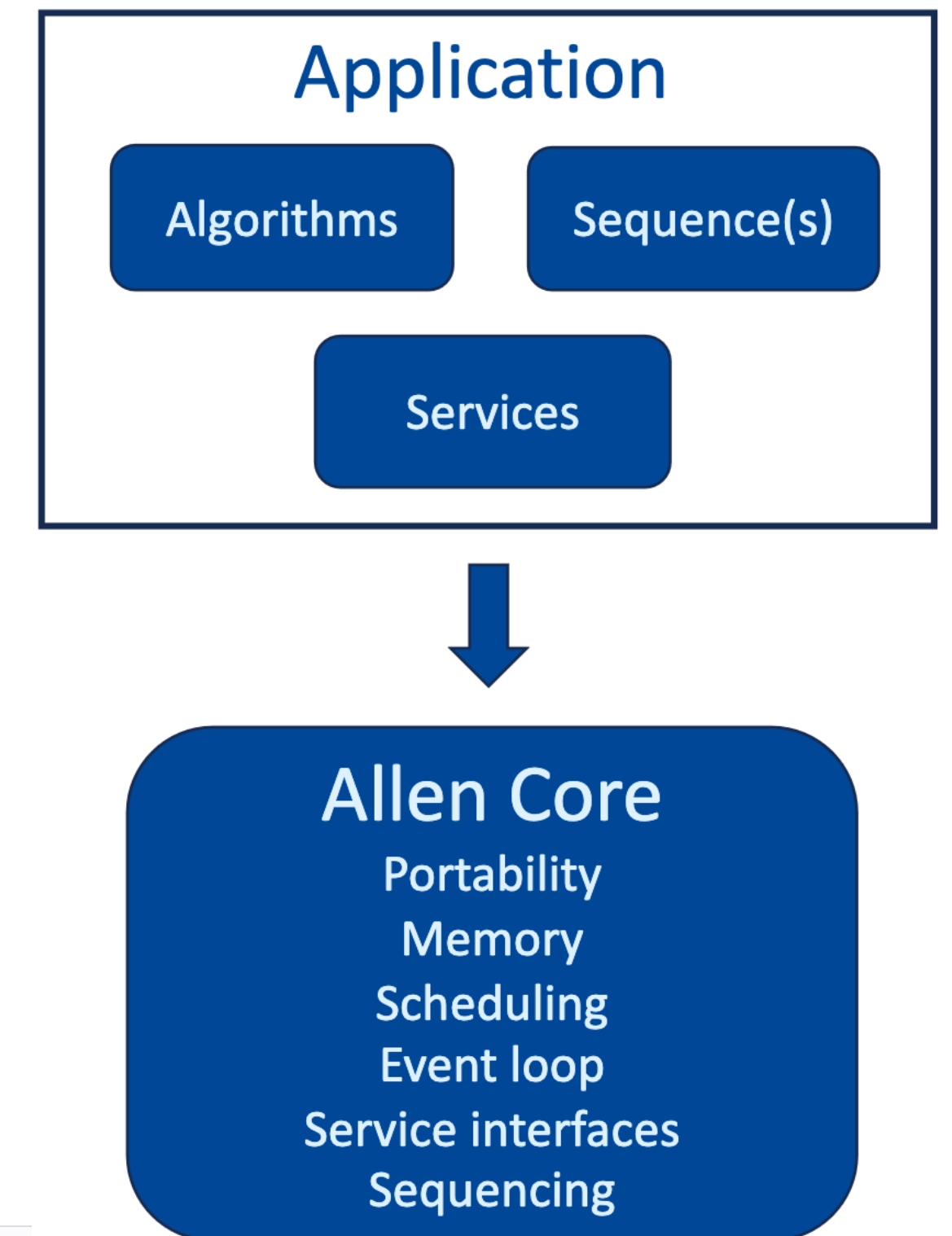
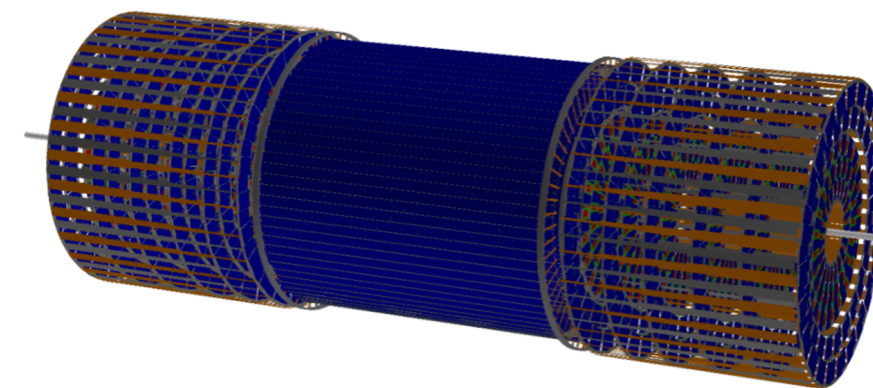
LHCb reconstruction
For pp and heavy ions
ready for Run 4

Prototypes of ML/AI-based
reconstruction ready

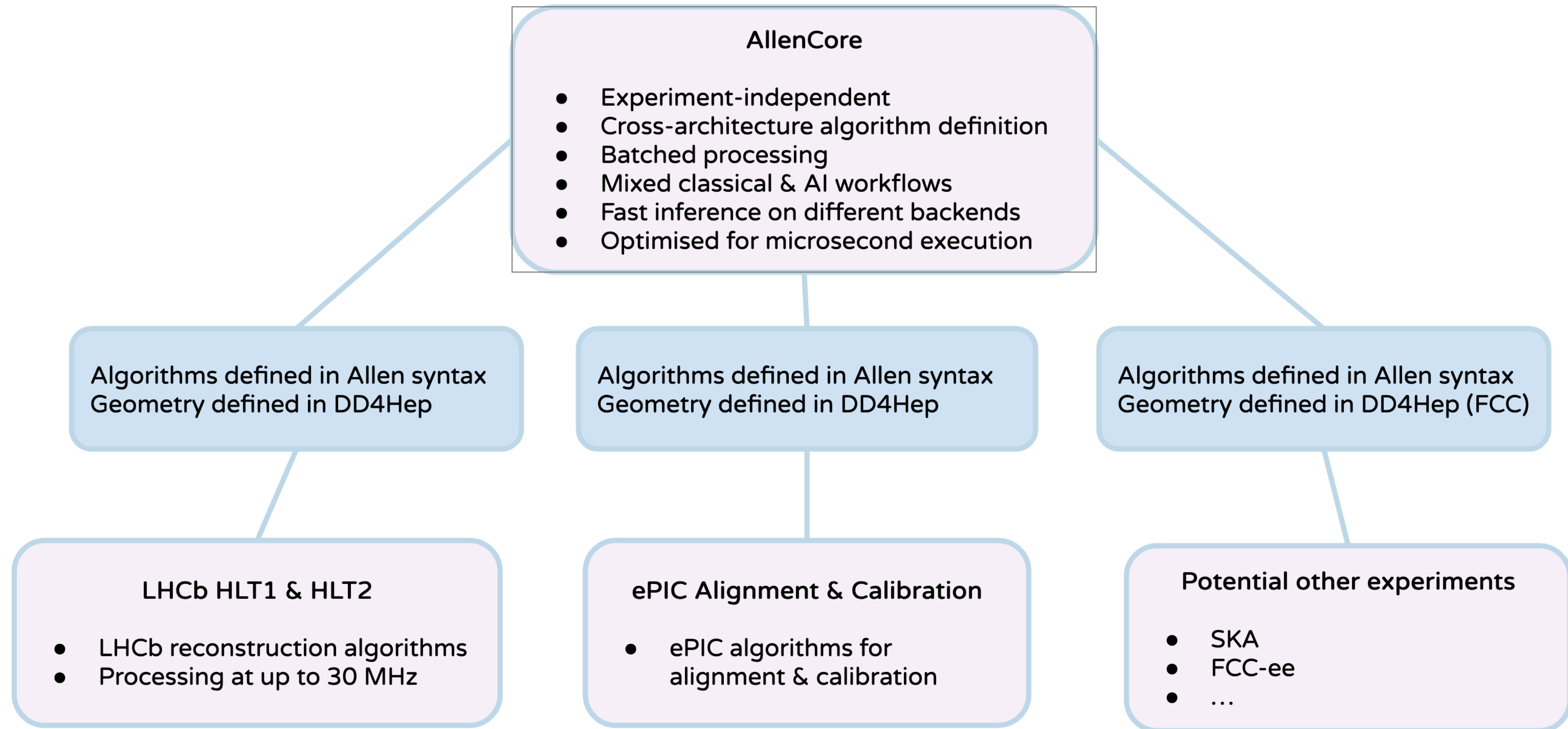
LHCb reconstruction
For pp and heavy ions
ready for Run 5

AllenCore: A heterogeneous high-throughput software framework

- Core parts extracted into experiment-independent AllenCore framework
 - Optimised for high-throughput execution on heterogeneous architectures
- Demonstrator with Open Data Detector
 - Read geometry in DD4Hep format
 - Use reconstruction algorithms from A Common Tracking System (ACTS)
 - GPU implementations available in tracc
 - First sequence of reconstruction algorithms runs
 - Proved independence from LHCb
- Integration of AllenCore within LHCb's software stack ongoing
- Hosted on gitlab: <https://gitlab.cern.ch/allen-core>



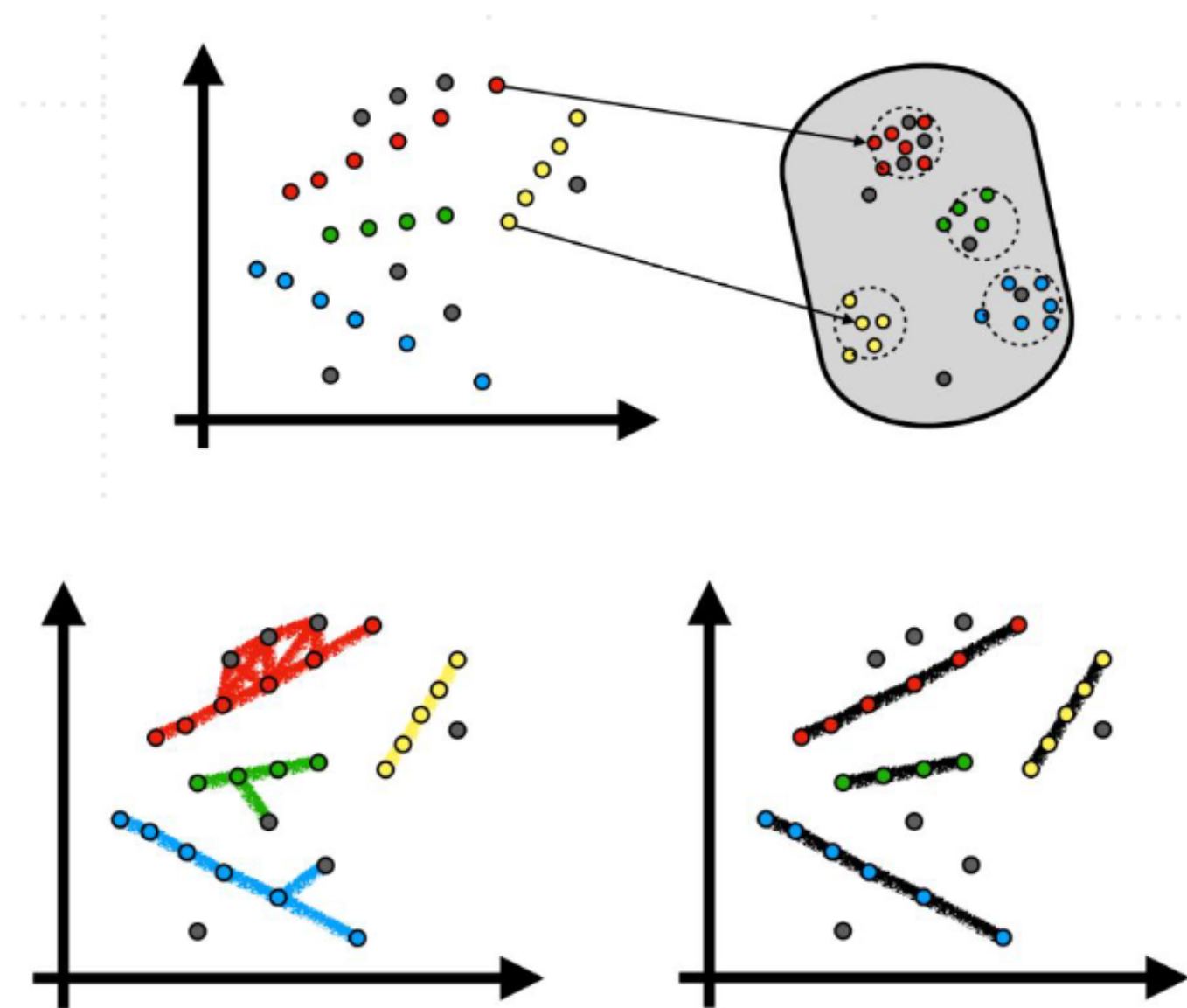
Vision of AllenCore interfaces with HEP experiments



R&D for reconstruction Graph Neural Networks (GNNs)

Track reconstruction in the vertex locator

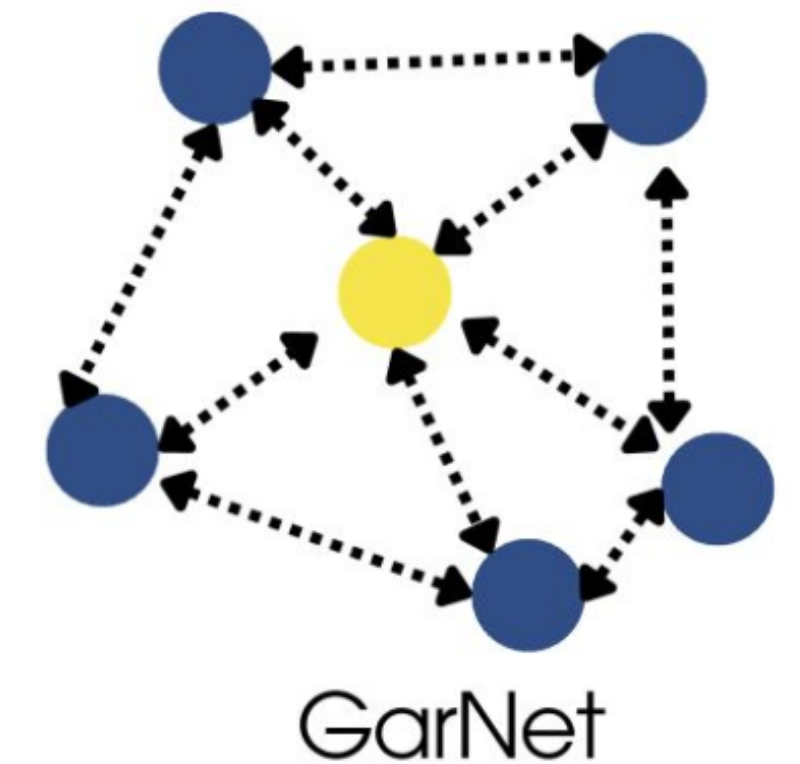
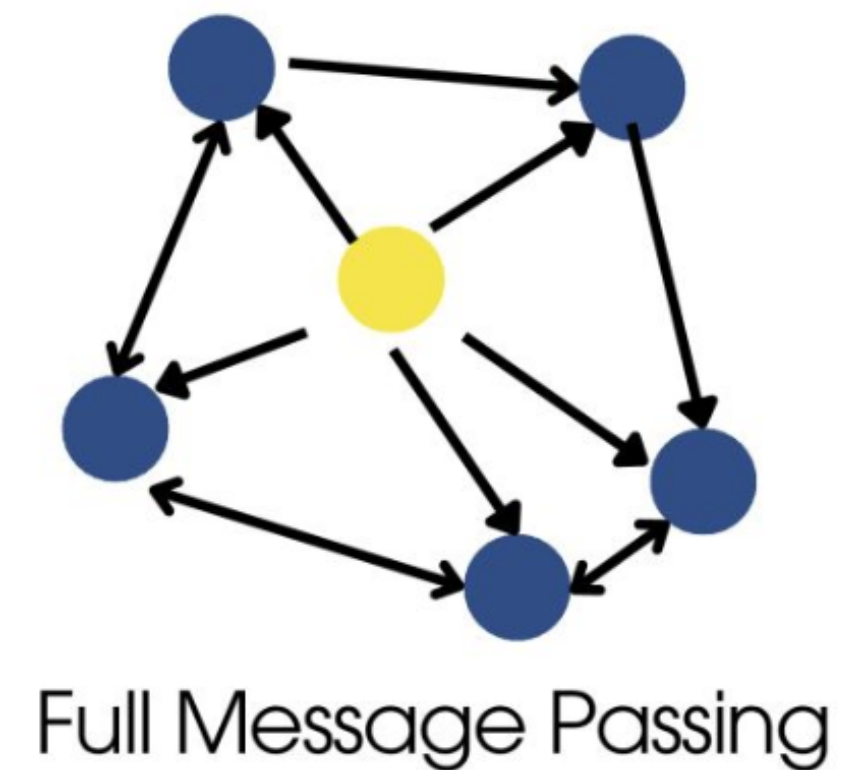
ETX4VELO



Calorimeter reconstruction

Inspiration from CMS

- Distance-weighted graph networks
- FPGA implementation

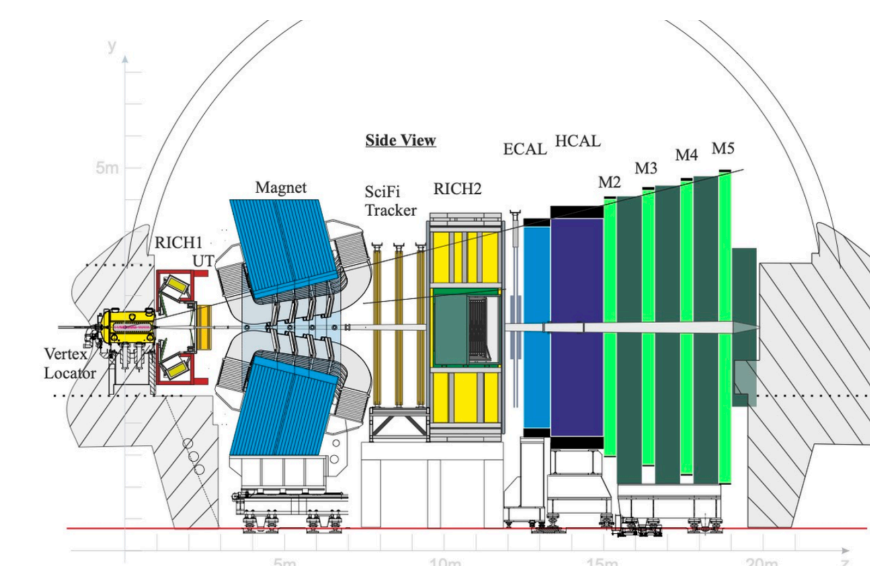


Common challenge: High-throughput inference on GPUs

<https://arxiv.org/abs/2406.12869>

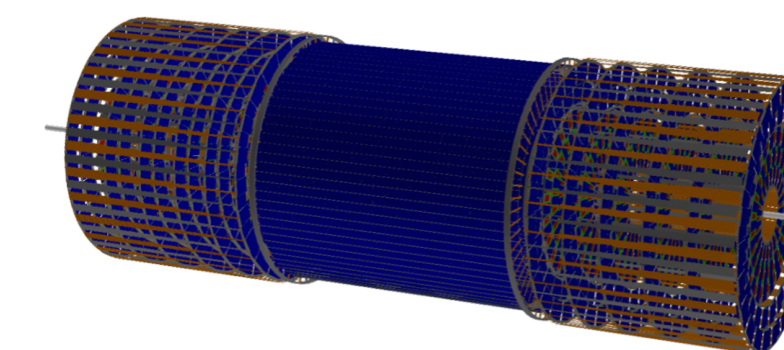
Summary

- LHCb has revolutionised its trigger system in Run 3 to use a fully software system based on GPUs and CPUs
- Allen framework has enabled HLT1 to go far beyond the original TDR design
 - Allen provides generic tools for high-performant processing on heterogeneous architectures
- Next challenge: LHCb Upgrade 2 with 25 TB/s to process
 - Can act as technology driver for future experiments
- Experiment-independent high-throughput heterogeneous software framework AllenCore developed
 - Demonstrator with Open Data Detector and ACTS algorithms in place
- R&D for high-throughput reconstruction with AI/ML methods on GPUs



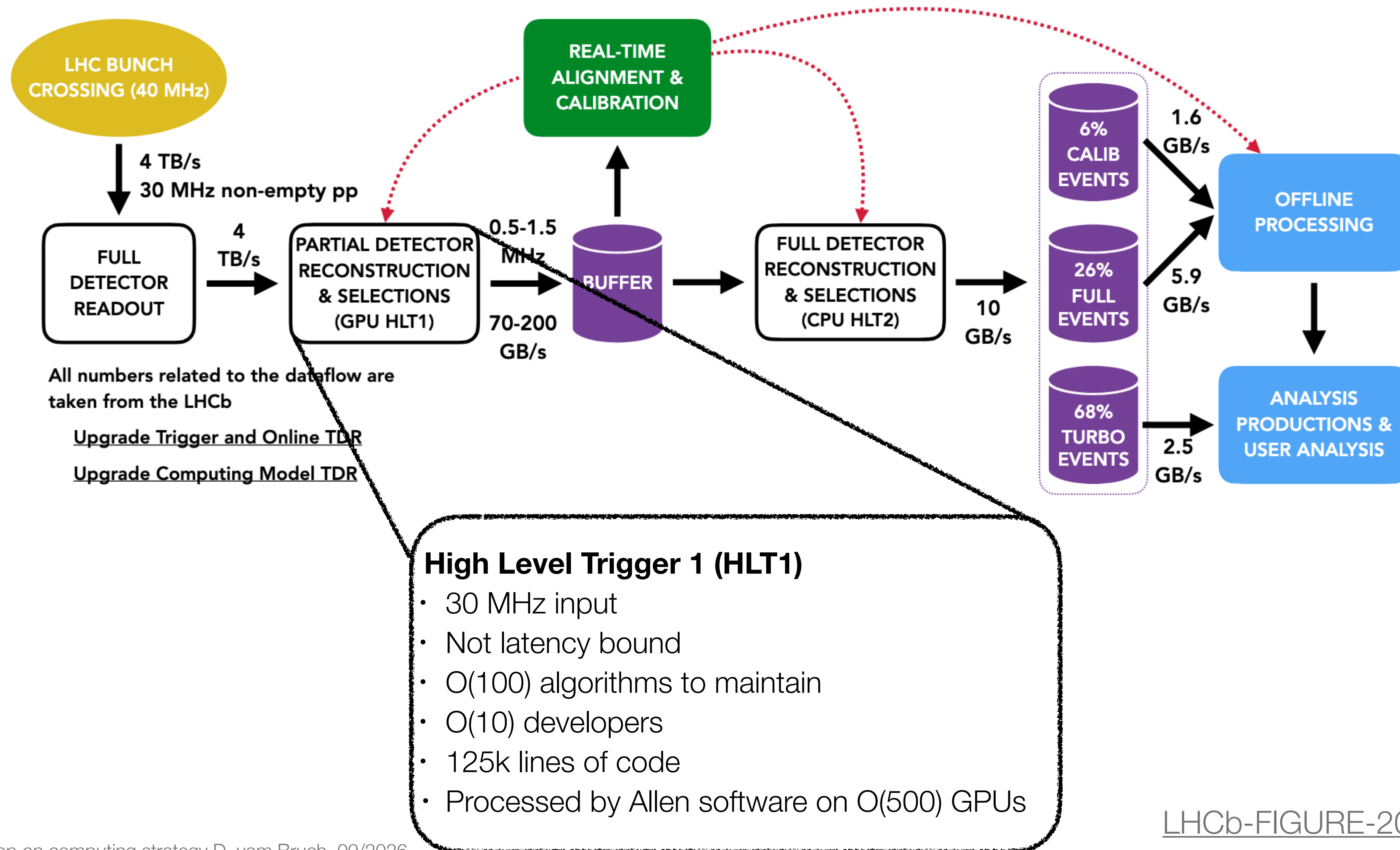
Allen Core

Portability
Memory
Scheduling
Event loop
Service interfaces
Sequencing

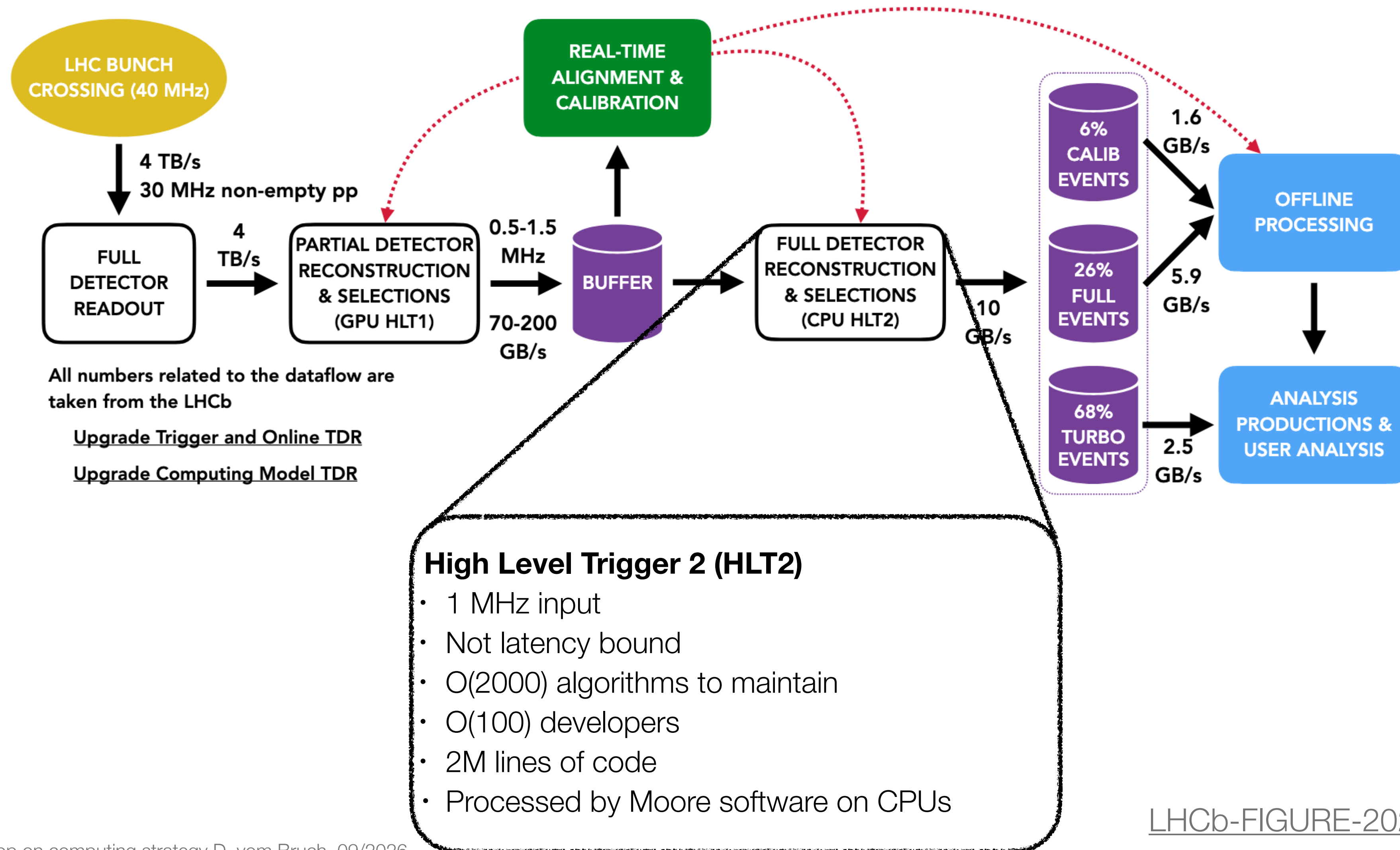


Backup

The LHCb trigger in LHC Runs 3 & 4

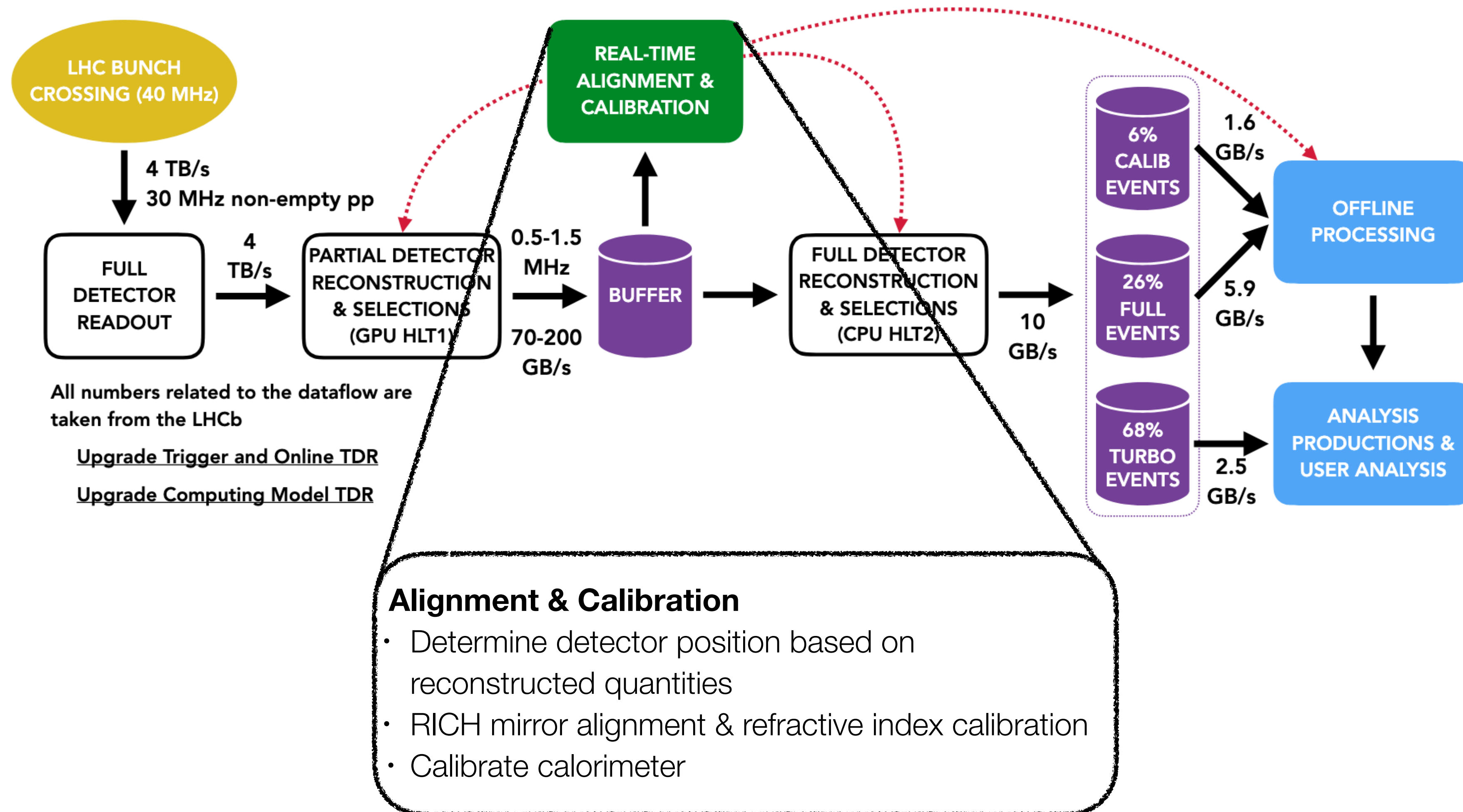


The LHCb trigger in LHC Runs 3 & 4



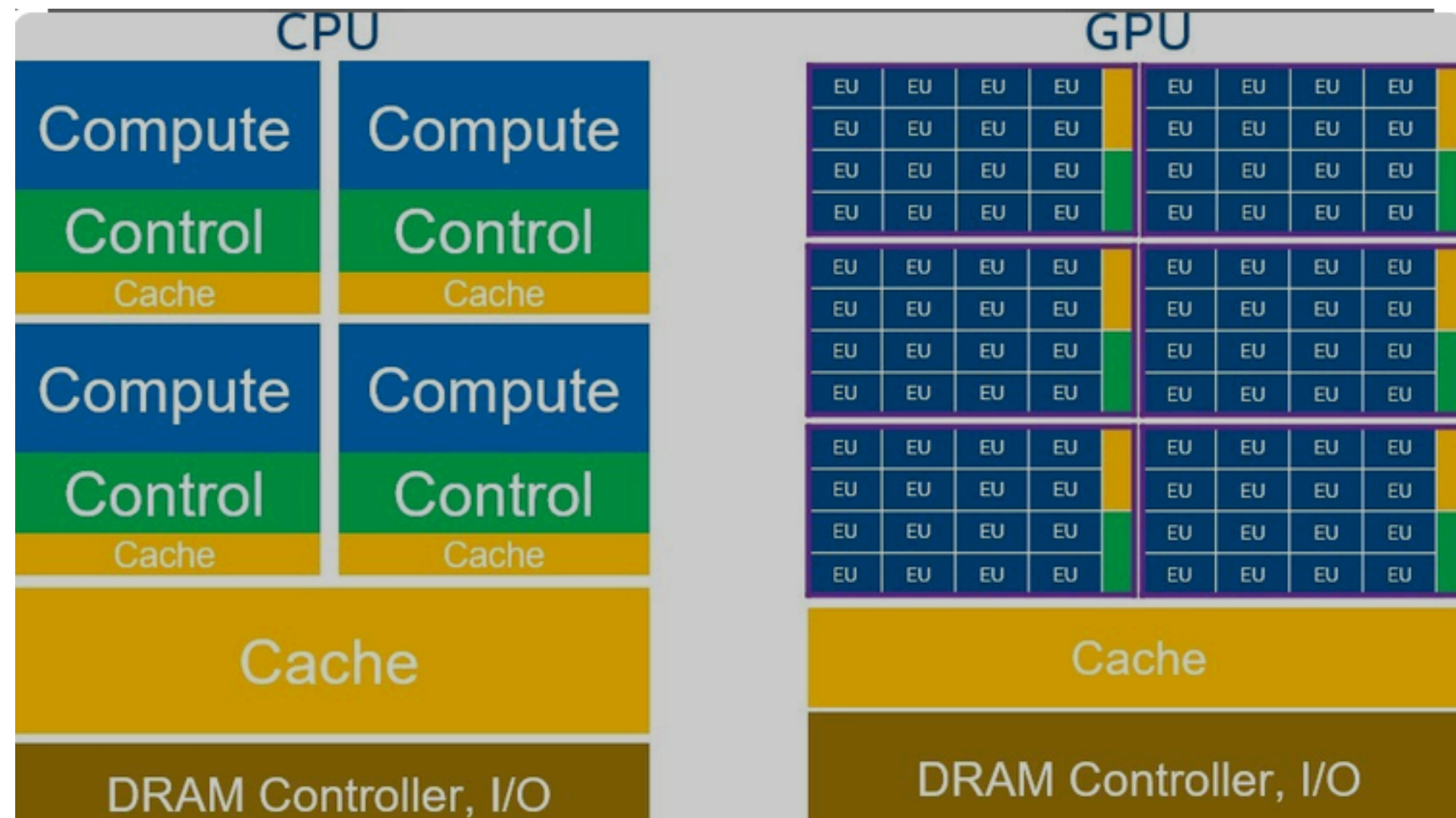
LHCb-FIGURE-2020-016

The LHCb trigger in LHC Runs 3 & 4



LHCb-FIGURE-2020-016

Allen: Memory management and algorithm scheduling



CPU

- Few strong cores
- Suited for serial workloads
- Quick access to large system memory

GPU

- Many weaker cores
- Suited for parallel workloads
- Limited memory on card

GPU challenges

- Fit workload into limited memory resources
 - Allen uses count first - write later model
 - Only allocate as much memory as needed
 - Allen provides memory manager and algorithm scheduler
 - Only keep objects in memory for as long as needed
- Sufficient parallelisable work to utilise compute cores
 - In Allen, every algorithm processes many collision events in parallel
 - « Multi-event scheduler » in Allen generates static sequence of algorithms to be processed using event masks

Allen: Versatile and scalable user interface

