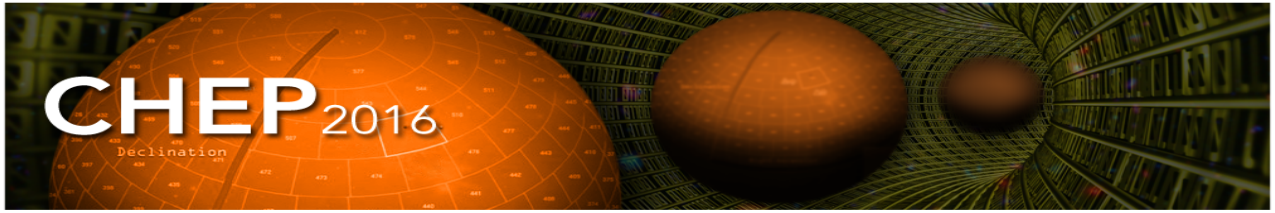


22nd International Conference on Computing in High Energy and Nuclear Physics, Hosted by SLAC and LBNL, Fall 2016

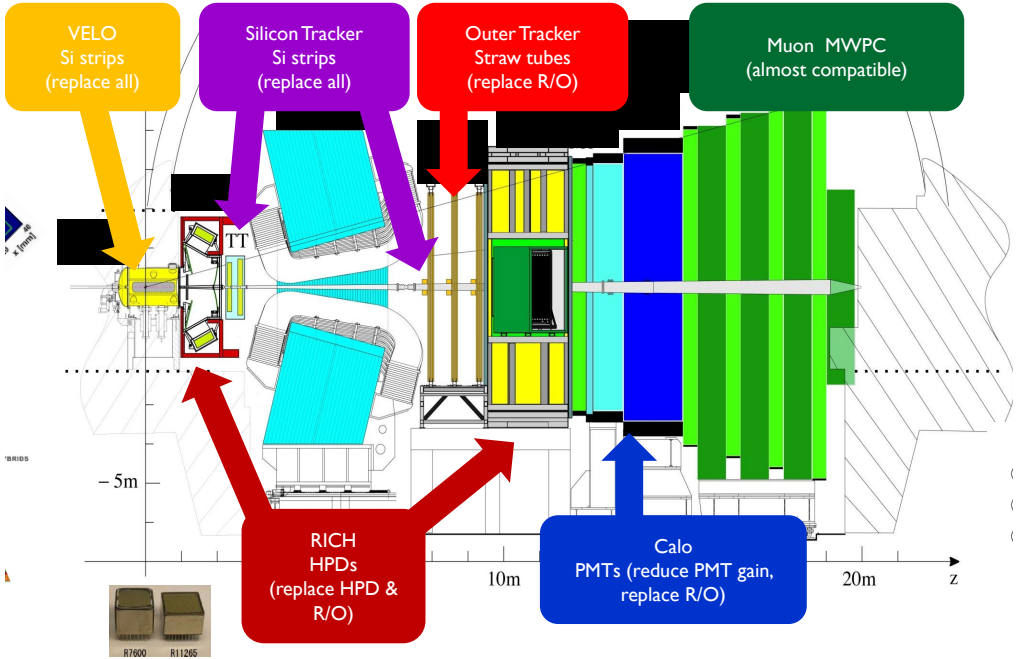
The LHCb software and computing upgrade for Run3: opportunities and challenges

Concezio Bozzi[1,2] and Stefan Roiser[1]
[1]CERN [2]INFN Ferrara

San Francisco, October 11th, 2016

The LHCb Upgrade

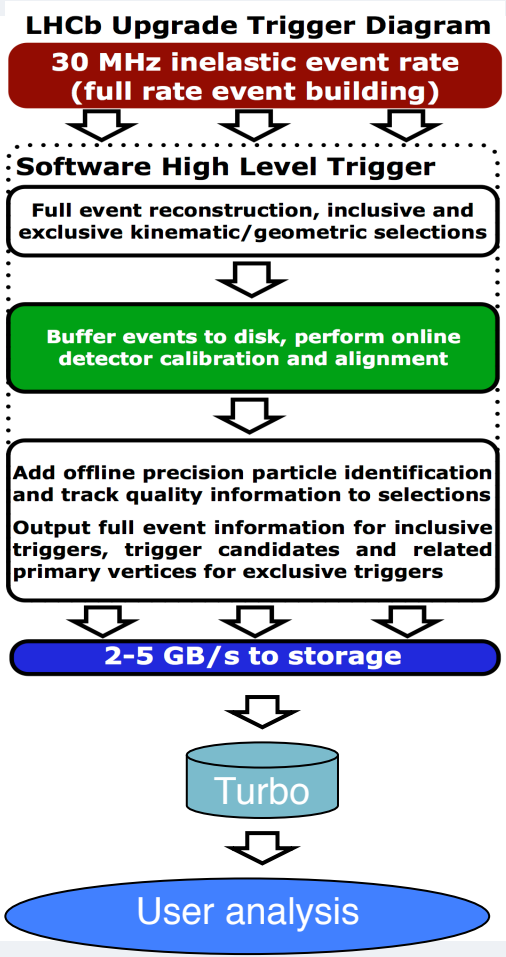
- The LHCb detector will be upgraded for Run3 (2021+)
- Factor 5 increase in instantaneous luminosity
  - 4 10^{32} → 2 10^{33} cm^{-2} s^{-1}
- Output rate of hardware trigger L0 (1MHz) becomes limiting factor
  - Deploy a fully software trigger, able to sustain the 30MHz inelastic event rate



Software and computing in the LHCb Upgrade era

G. Raven, Track 1, Monday

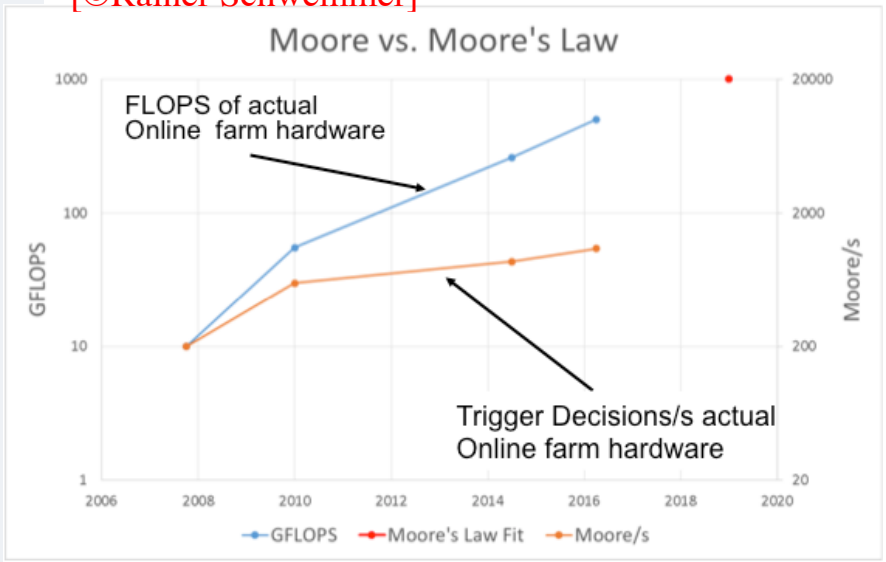
- The concepts deployed for Run2 will be further exploited for the Run3 Upgrade
  - HLT split into two parts
  - Turbo stream
    - final reconstructed physics objects in MDST format
    - RAW information not kept on offline storage
- 30 MHz events triggered in software
  - Strain on CPU efficiency of the trigger software
  - Trigger only signal events, 100% retention offline
    - event selection becomes classification
    - offline storage costs driven by HLT output rate
- HLT output rate O(GB/s), all Turbo
  - Smaller event size, more events, format in a range between MDST and DST
  - Very little offline data processing
- Signal proportional to MC needed
  - Work for simulation explodes



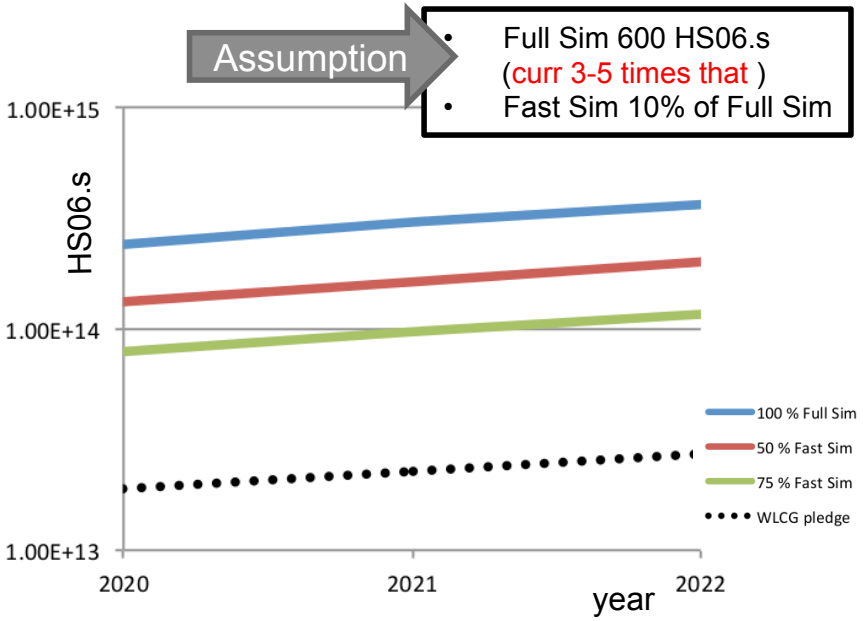
Two "driving forces" of SW and COMP Upgrade

- **Usage of trigger farm not optimal**

[©Rainer Schwemmer]

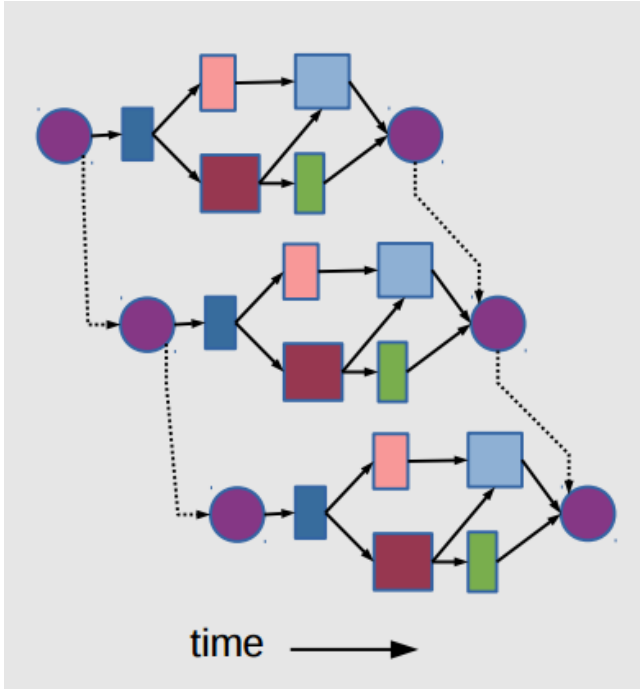


- **MC won't fit (by far) into the "pledgeable" resources**
  - **Aim to simulate 100 % of recorded (currently: 12%)**



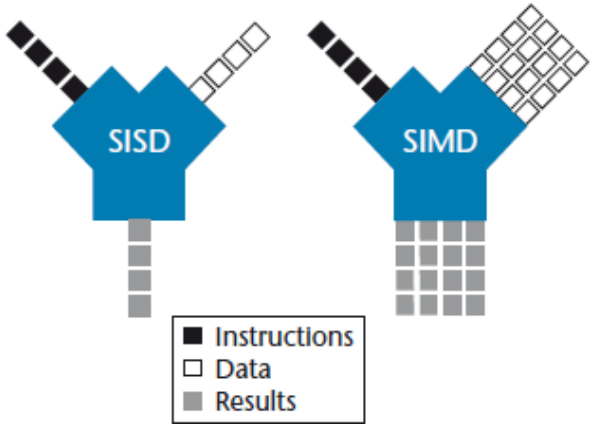
Framework (r)evolution

- Gaudi is and will stay the LHCb software framework
- Current single-threaded, sequential processing of events needs to change
  - Cache misses are increasingly a problem
  - Improve SIMD processing
  - Effective use of multiple cores
- Develop task-concurrent framework by using GaudiHive ideas
  - Scheduler automatically executes algorithms as data dependencies become available
    - Re-think algorithms as re-entrant entities to be executed in parallel
    - Input/output data needs to become immutable



Event model et al

- SIMD exploitation of current event model difficult because of AoS design
- Re-develop event model
  - Read-only, composable, no inheritance
  - Allow different representations AoS / SoA
  - Single precision whenever possible
- Allow the object to be represented in different views



- Tightly coupled on how to use event data from math libraries
- Investigating PODIO as a possible implementation

F. Gaede et al, Track 5, Monday

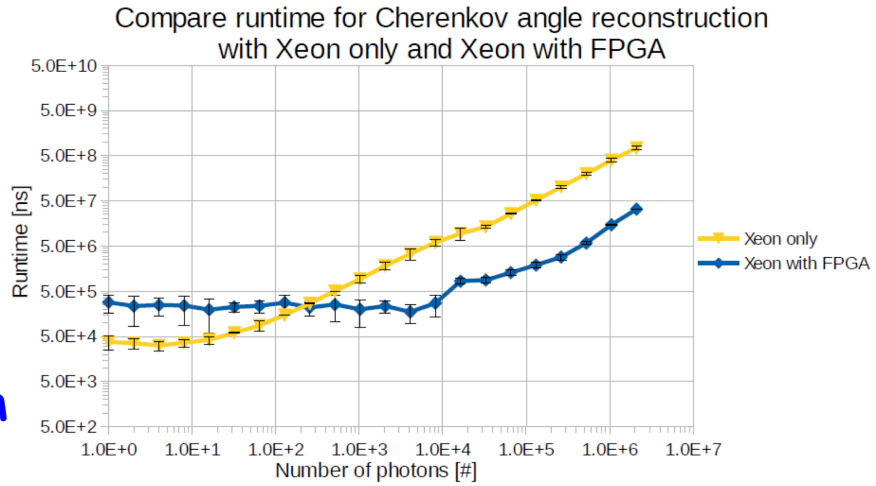
- Conditions database and detector description not optimal
  - Thread safety, XML persistency…
  - Investigating other possibilities (e.g. DD4HEP)

M. Frank et al, Track 2,Tu & Th

New architectures

- More freedom for the Trigger as we "own" the HW
  - Exploit alternative architectures
    - FPGA
    - GPGPU
    - KNL
    - ARM
    - Openpower

D. Campora,
S. Gallorini,
C. Faerber,
Track 1, Thursday



  - If implementation changes, make sure the same algorithm also runs x86_64 on the grid

- Looking into parallelizing specific algorithms
  - Kalman filter
  - Forward tracking
  - RICH reconstruction

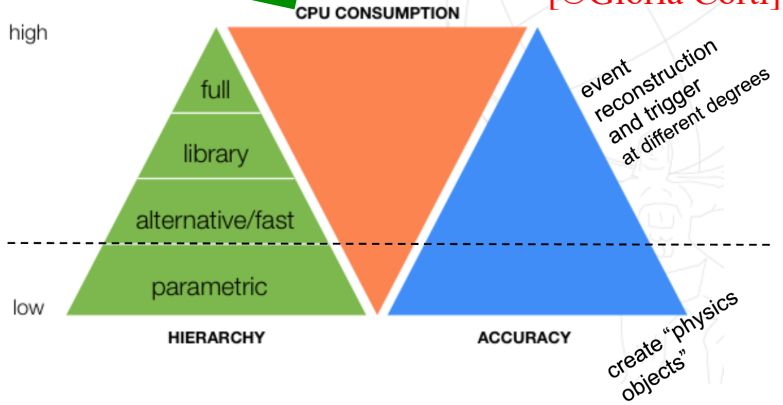
Parts of codebase
- With dominant execution time
- More likely to be parallelizable

Simulation

- Run 3 offline processing will be completely dominated by Simulation
- Need to gain 1-2 orders of magnitude of CPU work in order to keep within "flat funding" scenario
- Two parallel avenues being explored:
  - Fast simulation: from fully parameterized to fast detectors response to reuse of events
  - Ways to speed up simulation – GaudiHive, GaudiMP, Geant4 Multi-threading, use of geometries of different complexity
    - ☆ Note different concurrency models for Gaudi and Geant4

F. Carminati, plenary Tue

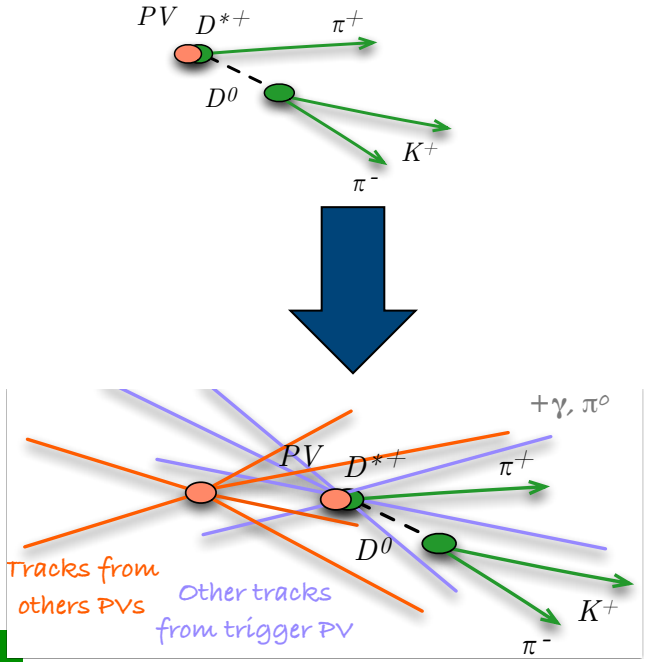
[©Gloria Corti]



- Make LHCb/Gauss simulation framework experiment-agnostic:
  - developing "Gaussino" in collaboration with FCC

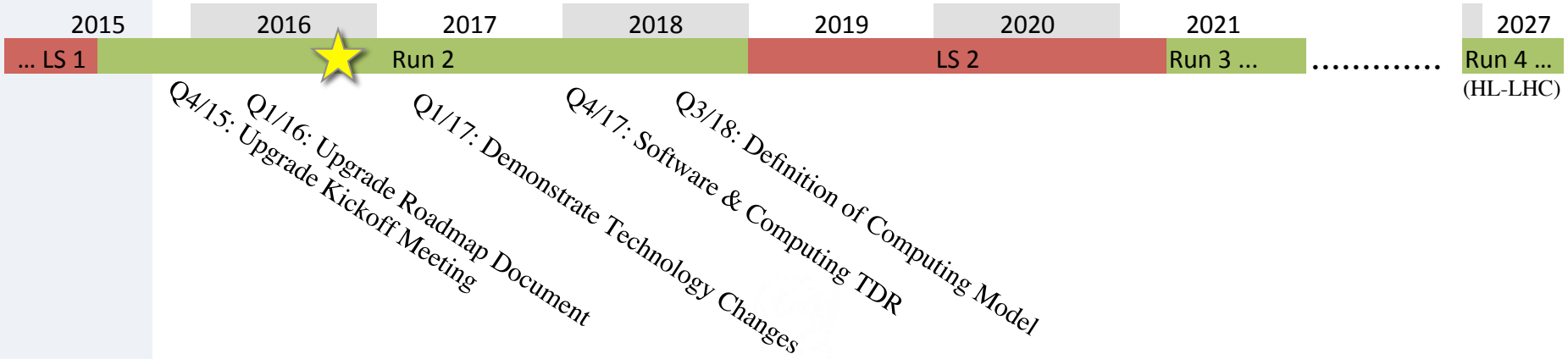
Distributed computing and analysis

- Current dataflow does not scale to Run3
  - RAW event storage too expensive, stripping does not scale
- Run 3 concepts being addressed during Run 2, using current framework:
  - Flexible data format for Turbo stream
    - Save varying level of detail depending on the triggering analysis
  - Centralised Ntuple production
    - Investigate organising "trains" of Ntuple production
- Also working on
  - Collaborative tools for analysis
  - Data preservation



A. Trisovic, A. Ustyuzhanin
Track 8 Thursday

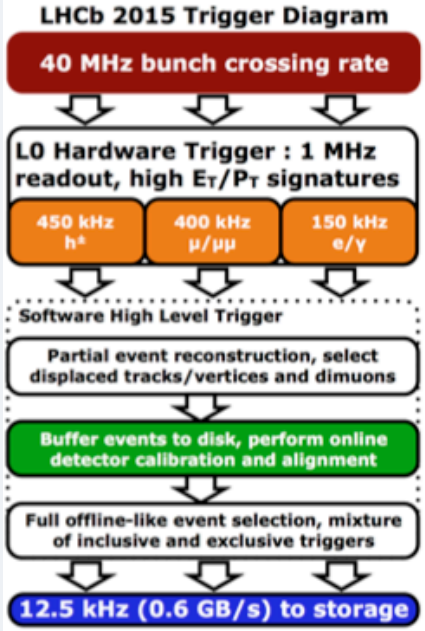
Outlook

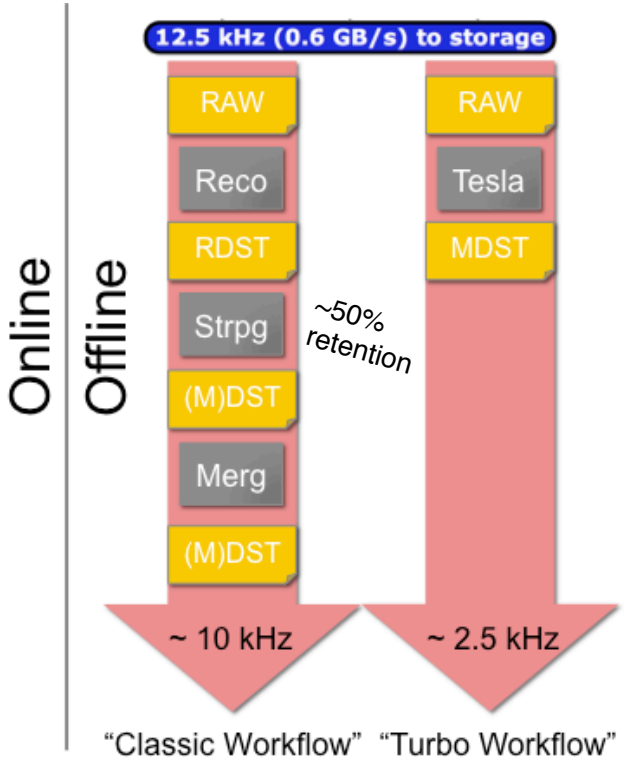


- High level milestones towards software and computing upgrade of Run 3
- Clear division into "revolutionary" and "evolutionary" parts
  - Very tight schedule for architectural work on task-based framework / algorithm vectorization / new event model / conditions / hardware
    - "demonstrators" by Q1/17
  - Run 2 as testbed in several other areas (simulation, analysis model…)

Backup

Workflow in Run2 (2015-2018)





- **Split HLT into two parts**
  - Final detector calibration & alignment done from HLT1 output

    → no second pass processing needed offline

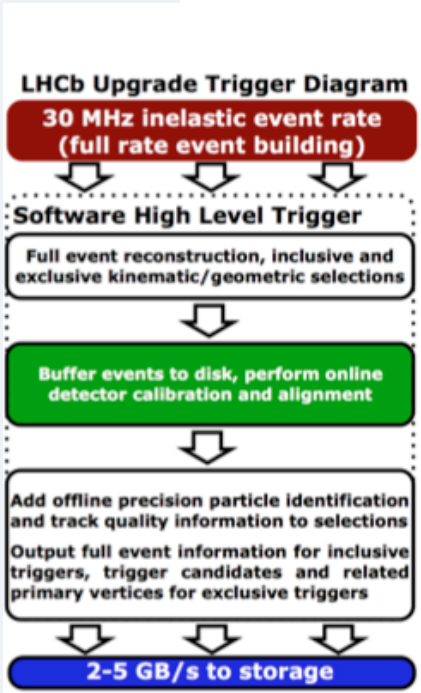
    → higher stripping retention
  - HLT2 reconstruction identical to offline
- **Turbo stream (in addition to Full)**
  - Contains final reconstructed physics objects in MDST format ready for physics analysis without any further processing

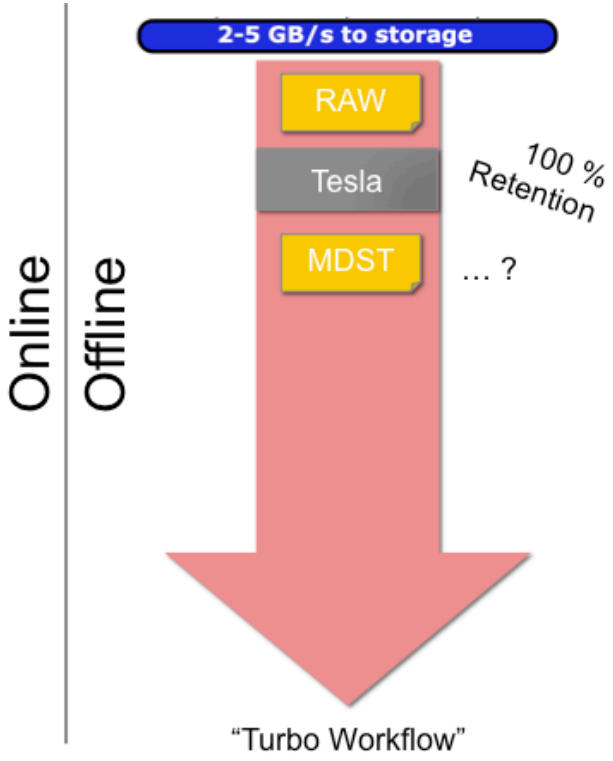
    → 100 % retention, everything selected is signal,
  - RAW information not kept on offline storage

→ One off chance to process data correctly in HLT

  - In 2016 introduced "persistReco"
    - Allow to export reconstruction information in the Turbo stream

Workflow in Run3 (2021+)





- 30 MHz events to be triggered in software
  - Strain on CPU efficiency of the trigger software (Gaudi Online)
- Trigger only signal events
  - event selection becomes classification
  - 100 % retention → offline storage costs driven by HLT output rate
- Stripping becomes streaming of events
- HLT output rate O(GB/s),all Turbo
  - Smaller event size, more events, format in a range between MDST and DST
  - More events per file → sparse reading
    - efficient model for user analysis needed
- Very little offline data processing
- Signal proportional to MC needed
  - Work for simulation explodes