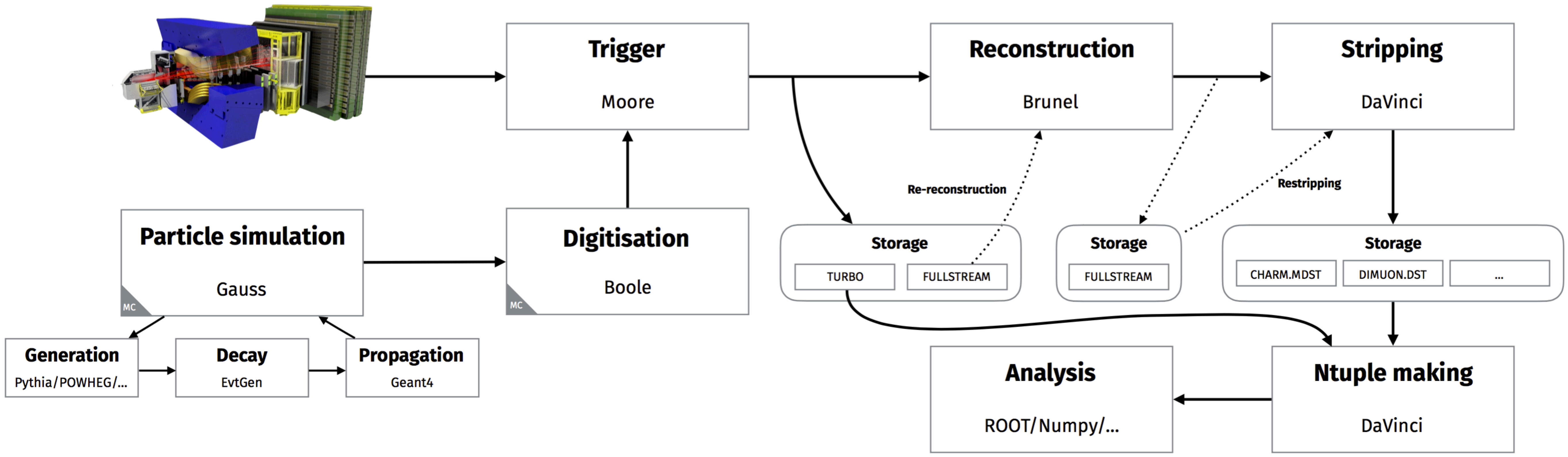
The LHCb Run 3 data flow

LHCb Starterkit - 22-26 November 2021

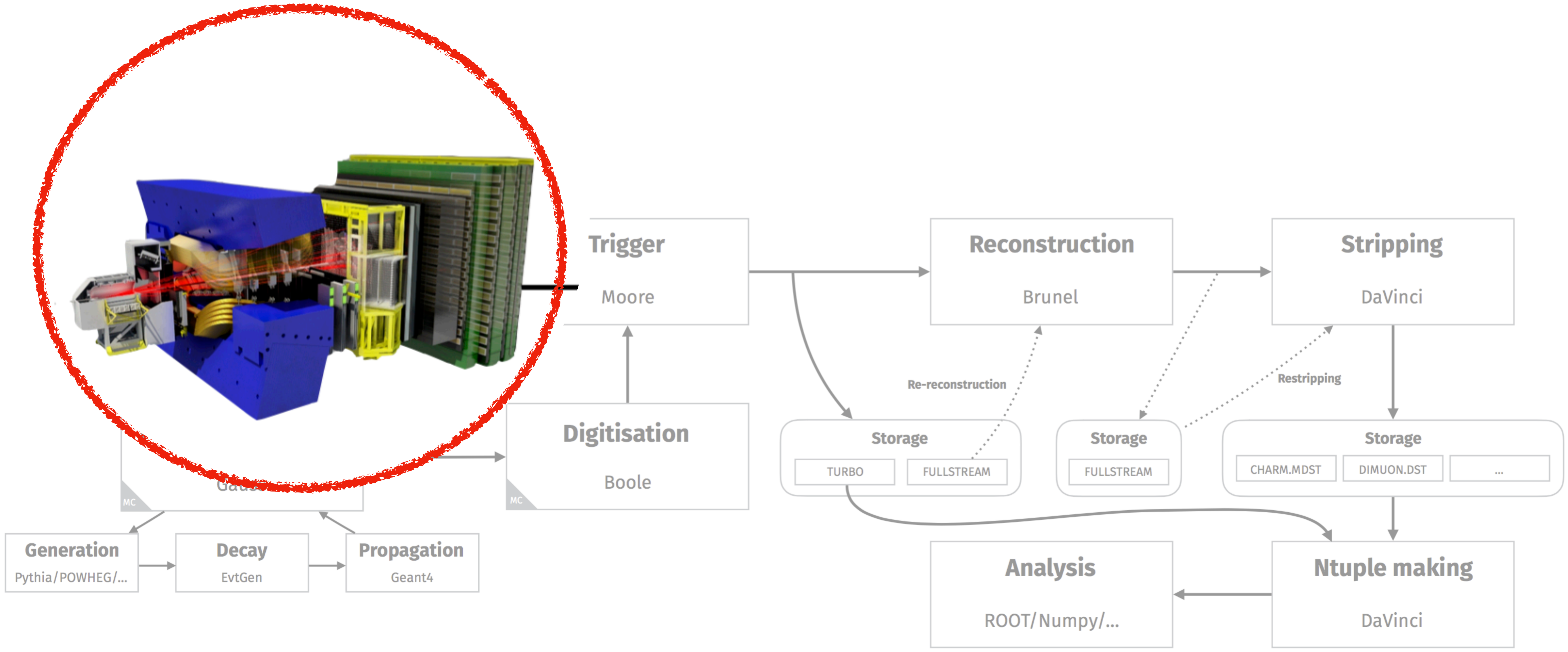
Christina Agapopoulou

Reminder of the Run 2 data flow

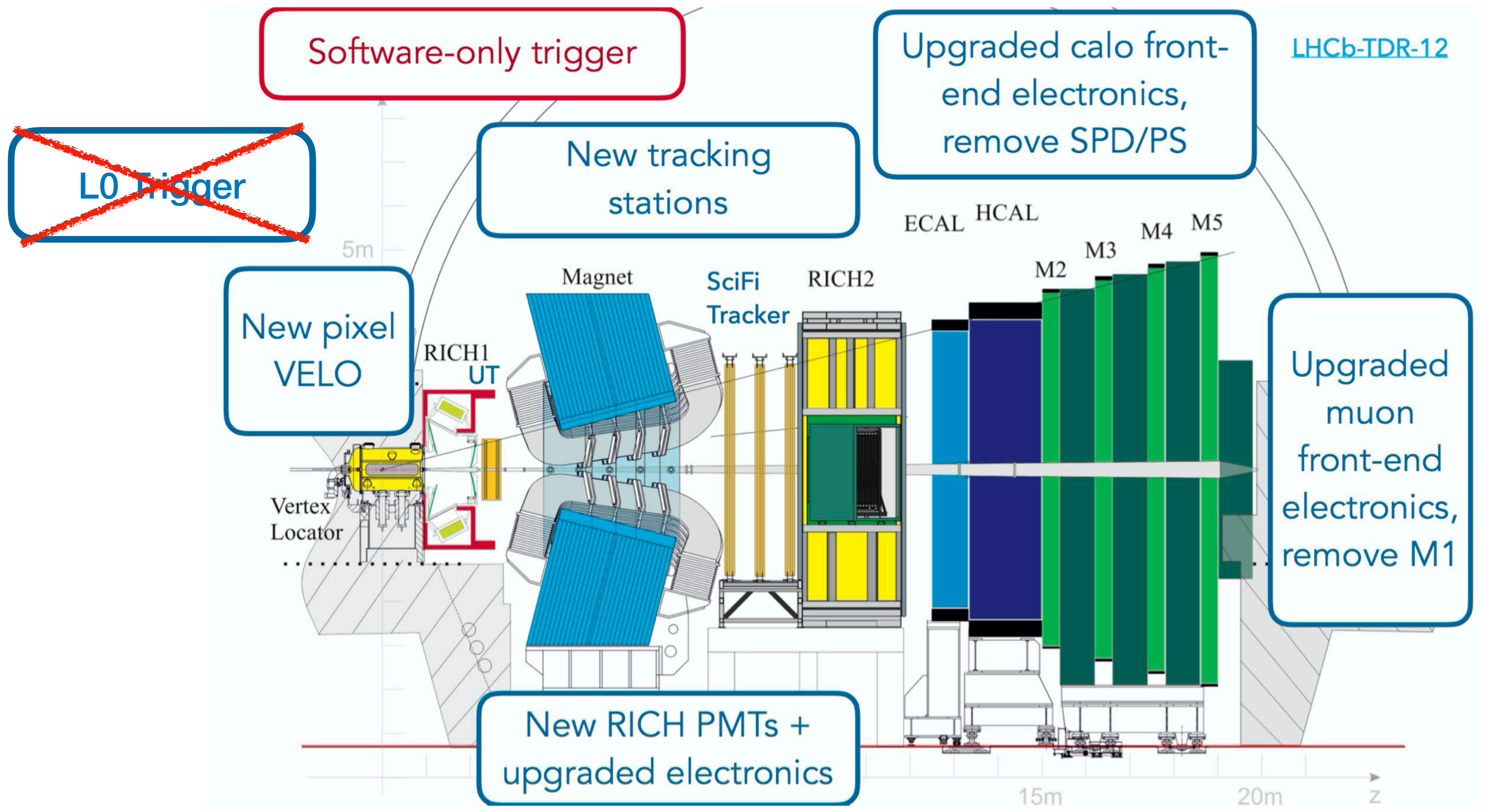
Checkout Valeriia's Wednesday lesson for details on the RunI+II data flow!



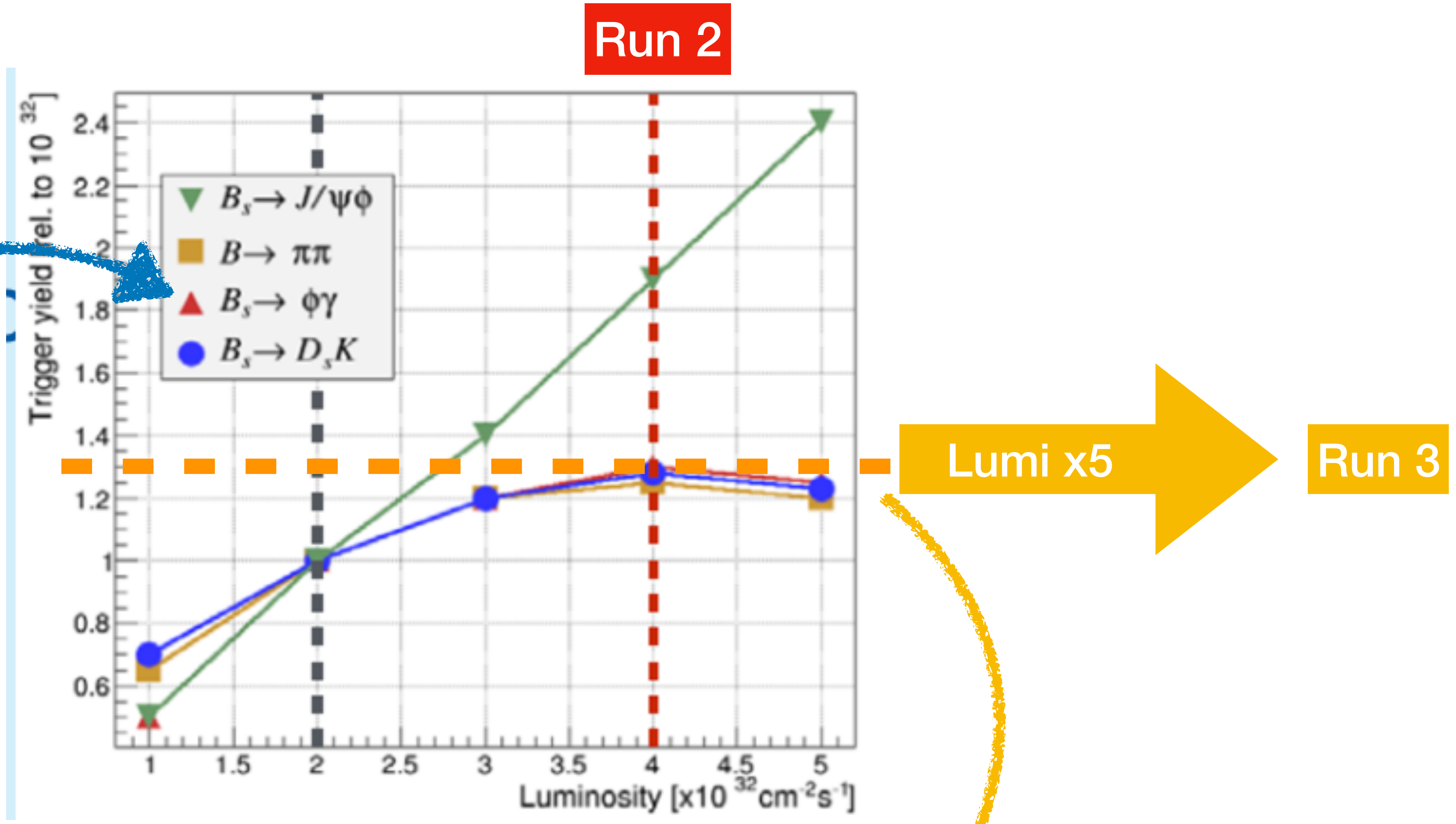
So let's see what's new in Run 3!



The LHCb Run 3 upgrade



No L0 trigger, why?

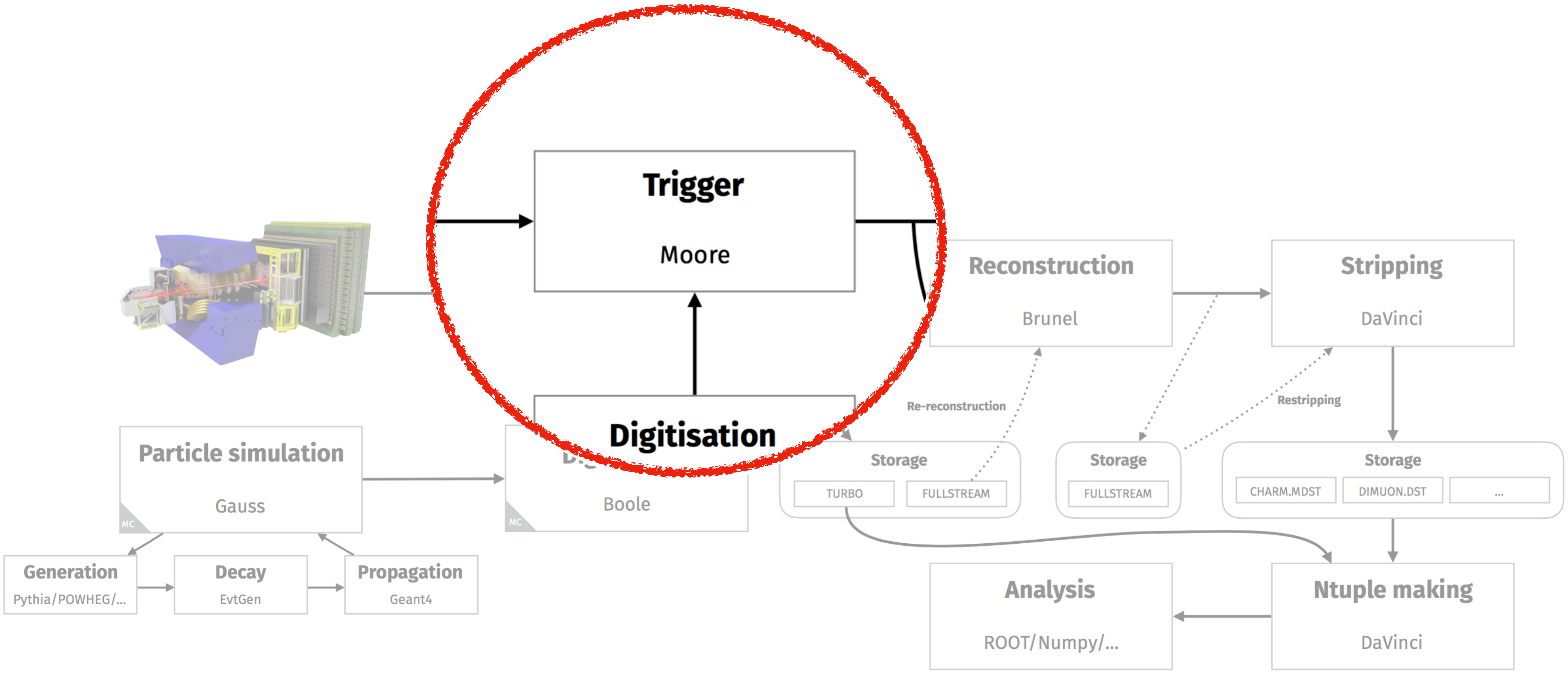


We care a lot about those in LHCb!

But our trigger yield is flat-lining :(

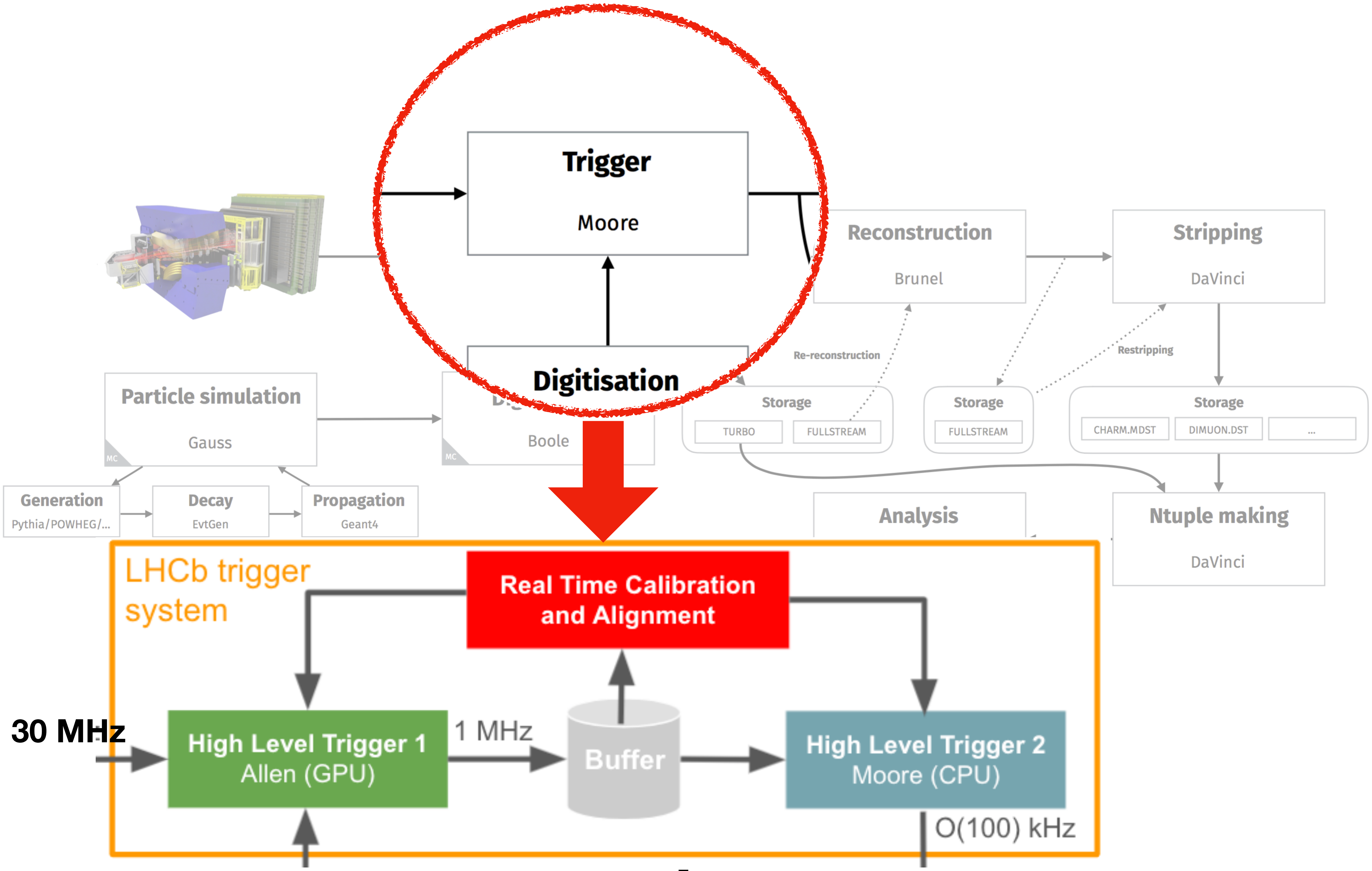
Why? Because L0 has a 1MHz output rate limitation

So let's see what's new in Run 3!



It's ALL about ONLINE reconstruction, calibration and selection!

So let’s see what’s new in Run 3!



Our brand-new HLT1 GPU trigger

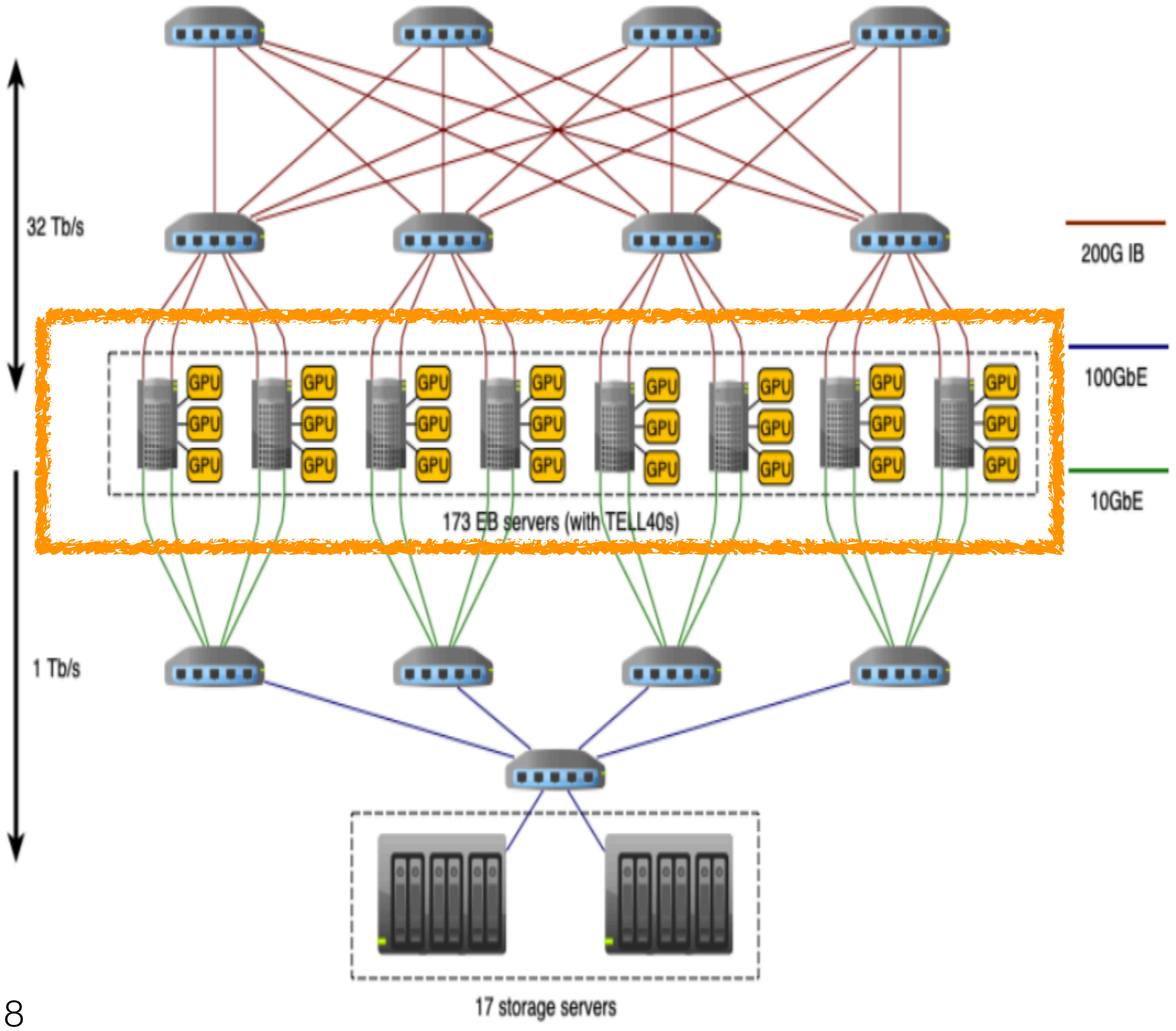
HLT1 philosophy still the same

Partial (but fast) reconstruction and coarse selection to enable online alignment & calibration before HLT2

BUT has to run at 30 times the input rate now!

- HLT1 tasks inherently parallelizable
- GPUs map well into LHCb DAQ architecture
- Cheaper network between HLT1 & HLT2
- Can run on O(200) GPUs
- **Was chosen as the baseline for the upgrade!**

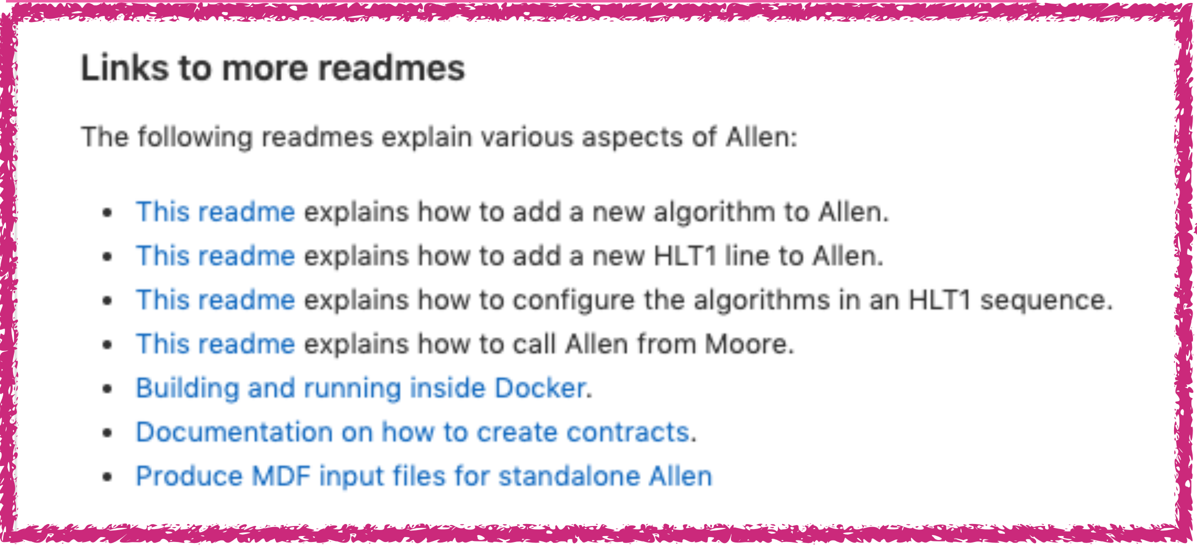
When you find the GPU you have been hunting for months at ~~MSRP~~ LHCb



Where is the code?

- Allen software project: gitlab repo
- Runs on CPU & GPU
- GPU code written in CUDA
- Supports three modes:
  - Standalone
  - Compiling within the LHCb framework (event loop steered from Allen)
  - Compiling within the LHCb framework (event loop steered from Moore)

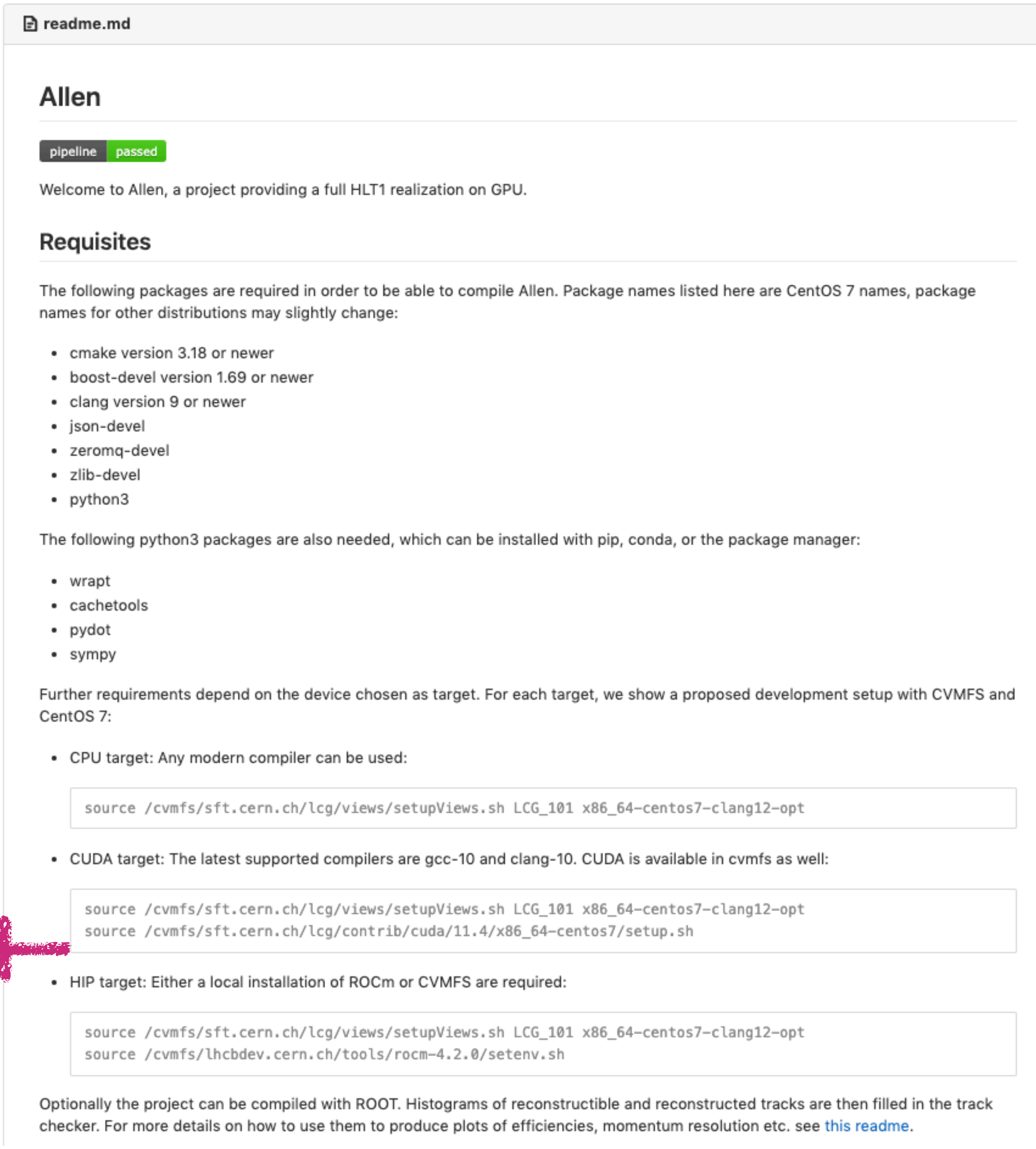
Documentation in Allen Readme's



Links to more readmes

The following readmes explain various aspects of Allen:

- This readme explains how to add a new algorithm to Allen.
- This readme explains how to add a new HLT1 line to Allen.
- This readme explains how to configure the algorithms in an HLT1 sequence.
- This readme explains how to call Allen from Moore.
- Building and running inside Docker.
- Documentation on how to create contracts.
- Produce MDF input files for standalone Allen



readme.md

Allen

pipeline passed

Welcome to Allen, a project providing a full HLT1 realization on GPU.

Requisites

The following packages are required in order to be able to compile Allen. Package names listed here are CentOS 7 names, package names for other distributions may slightly change:

- cmake version 3.18 or newer
- boost-devel version 1.69 or newer
- clang version 9 or newer
- json-devel
- zeromq-devel
- zlib-devel
- python3

The following python3 packages are also needed, which can be installed with pip, conda, or the package manager:

- wrapt
- cachetools
- pydot
- sympy

Further requirements depend on the device chosen as target. For each target, we show a proposed development setup with CVMFS and CentOS 7:

- CPU target: Any modern compiler can be used:

```
source /cvmfs/sft.cern.ch/lcg/views/setupViews.sh LCG_101 x86_64-centos7-clang12-opt
```

- CUDA target: The latest supported compilers are gcc-10 and clang-10. CUDA is available in cvmfs as well:

```
source /cvmfs/sft.cern.ch/lcg/views/setupViews.sh LCG_101 x86_64-centos7-clang12-opt
source /cvmfs/sft.cern.ch/lcg/contrib/cuda/11.4/x86_64-centos7/setup.sh
```

- HIP target: Either a local installation of ROCm or CVMFS are required:

```
source /cvmfs/sft.cern.ch/lcg/views/setupViews.sh LCG_101 x86_64-centos7-clang12-opt
source /cvmfs/lhcbdev.cern.ch/tools/rocm-4.2.0/setenv.sh
```

Optionally the project can be compiled with ROOT. Histograms of reconstructible and reconstructed tracks are then filled in the track checker. For more details on how to use them to produce plots of efficiencies, momentum resolution etc. see this readme.

HLT2: Moore

gitlab repo

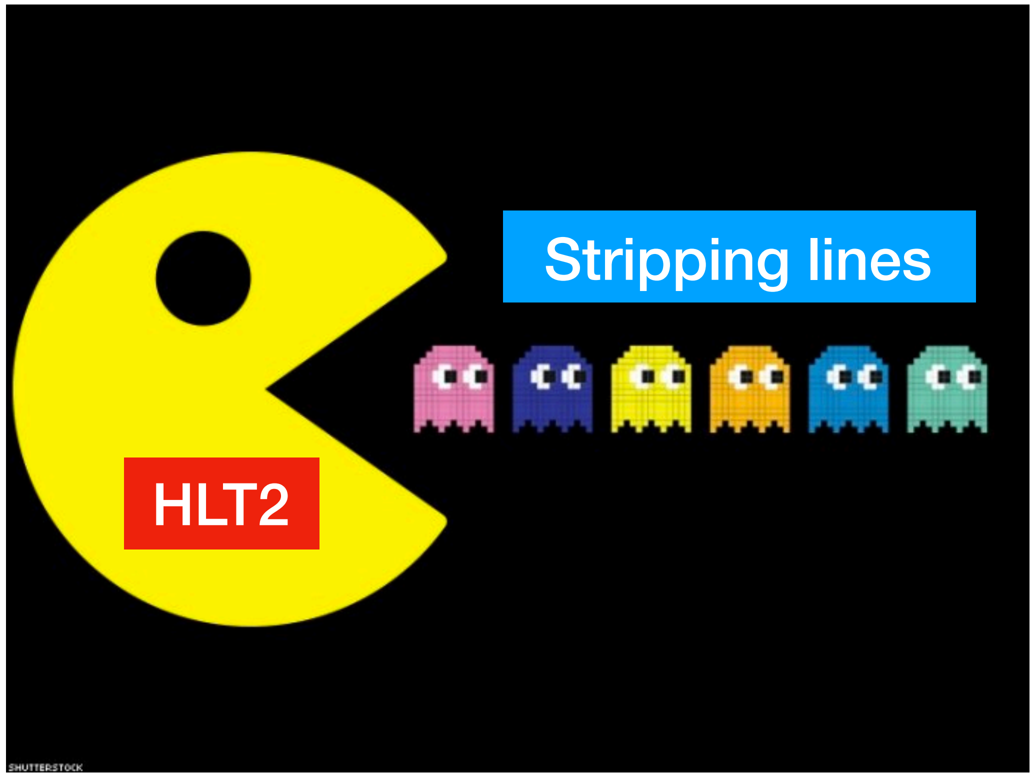
RUN MO²RE

HLT2 philosophy kind of the same (?)

Refined reconstruction and complex selection algorithms

BUT most physics selections will be moved from Stripping to HLT2

AND event processing rate (i.e. throughput) has to NOT explode



New, more vectorizable event model

We could just buy more CPUs ($$)

BUT

Can our code do better?

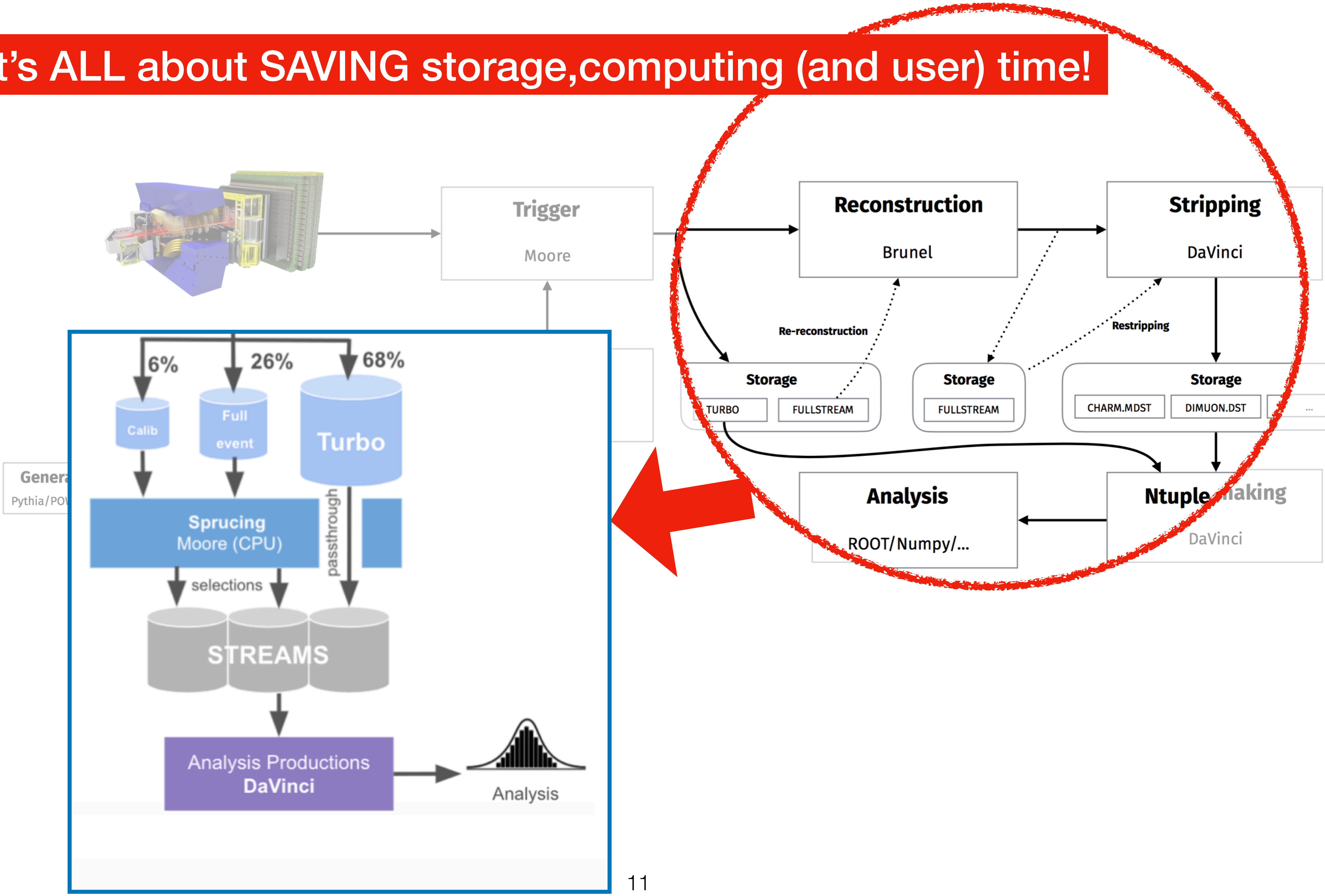
Configurable selections: LoKi→ThOr functors

Attention! Deadlines for Run 3 HLT2 lines are approaching.
Check if your analysis is already covered! And if not, write some lines ;)

More details on ThOr functors and on how to develop HLT2 lines in Lukas's lecture afterwards!

What else?

It's ALL about SAVING storage,computing (and user) time!



Going TURBO

70% in Run 2

32% in Run 3

Offline reconstruction

30% in Run 2

68% in Run 3

TURBO

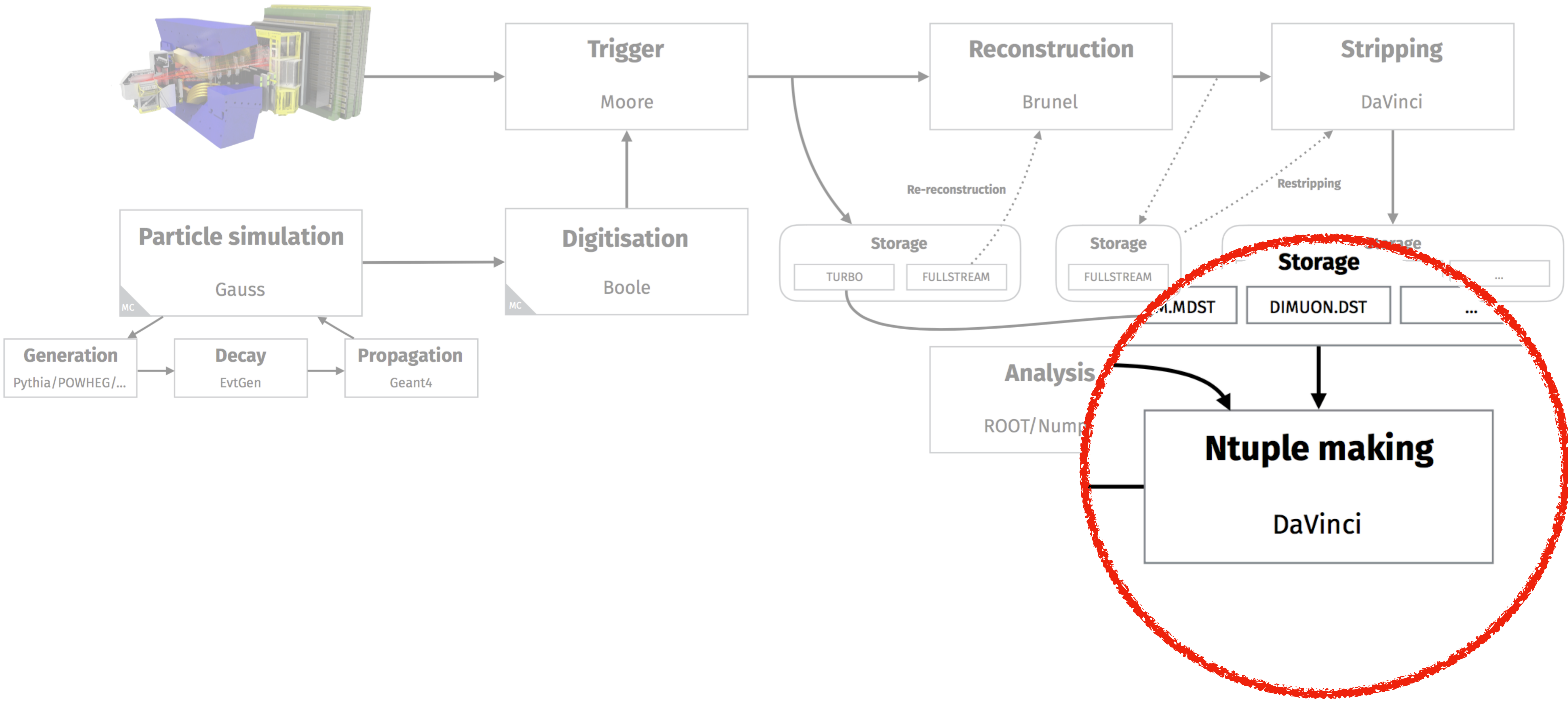
TURBO reminder: save only useful information from event

Bypass full offline reconstruction & stripping

Save storage and computing power

In Run 3, offline-like quality already at HLT2 → TURBO is the natural choice

There's more?



Trigger
Moore
Reconstruction
Brunel
Stripping
DaVinci
Re-reconstruction
Restripping
Particle simulation
Gauss
MC
Digitisation
Boole
MC
Storage
TURBO
FULLSTREAM
Storage
FULLSTREAM
Storage
M.MDST
DIMUON.DST
...
Generation
Pythia/POWHEG/...
Decay
EvtGen
Propagation
Geant4
Analysis
ROOT/Nump
Ntuple making
DaVinci

Analysis productions

- Move from user (Ganga) to central productions (DIRAC)

- Analysis productions!
  - Automatic testing of option files
  - No more babysitting grid jobs
  - Automatic preservation of job details and config files
  - Automatic error interpretation
  - Web interface

More details on Analysis productions were given by Aidan on Wednesday
Take a look here !

Changes to DaVinci

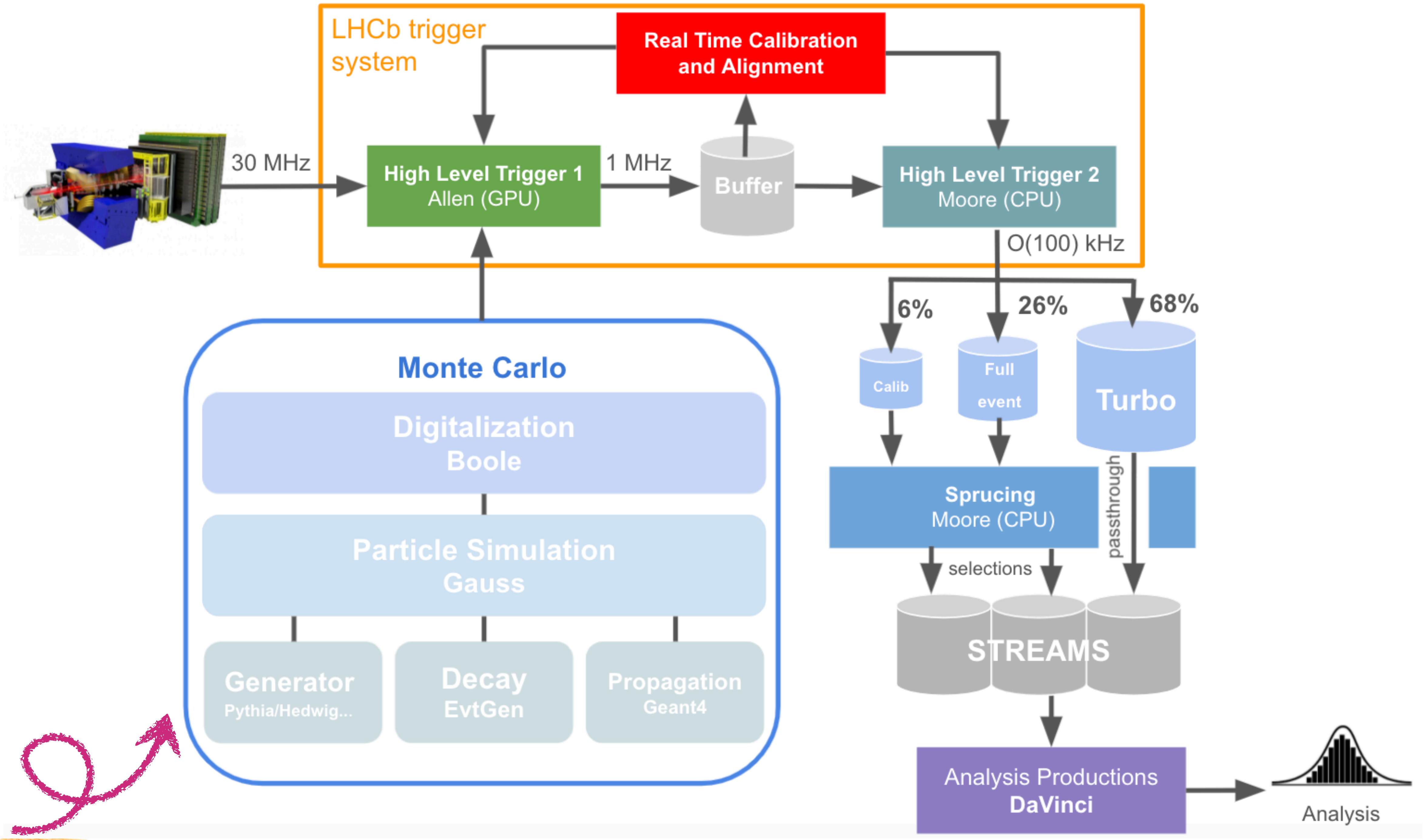
$$\begin{tikzcd} \boxed{\xcancel{\textbf{TupleTools}}} \arrow[r] & \boxed{\textbf{FunTuple}} \end{tikzcd}$$

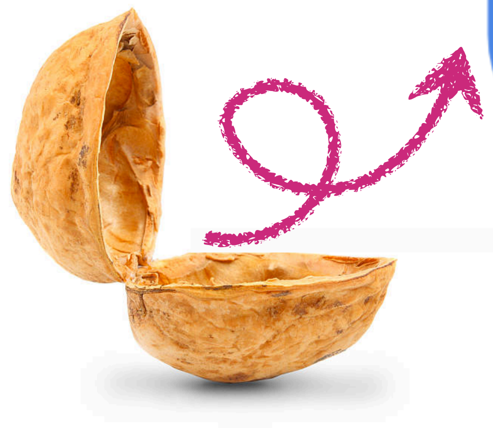
TupleTools: very easy to use BUT can lead to too many unused brunches (remember, it's ALL about saving storage)

FunTuple: gives the power to the analyst!

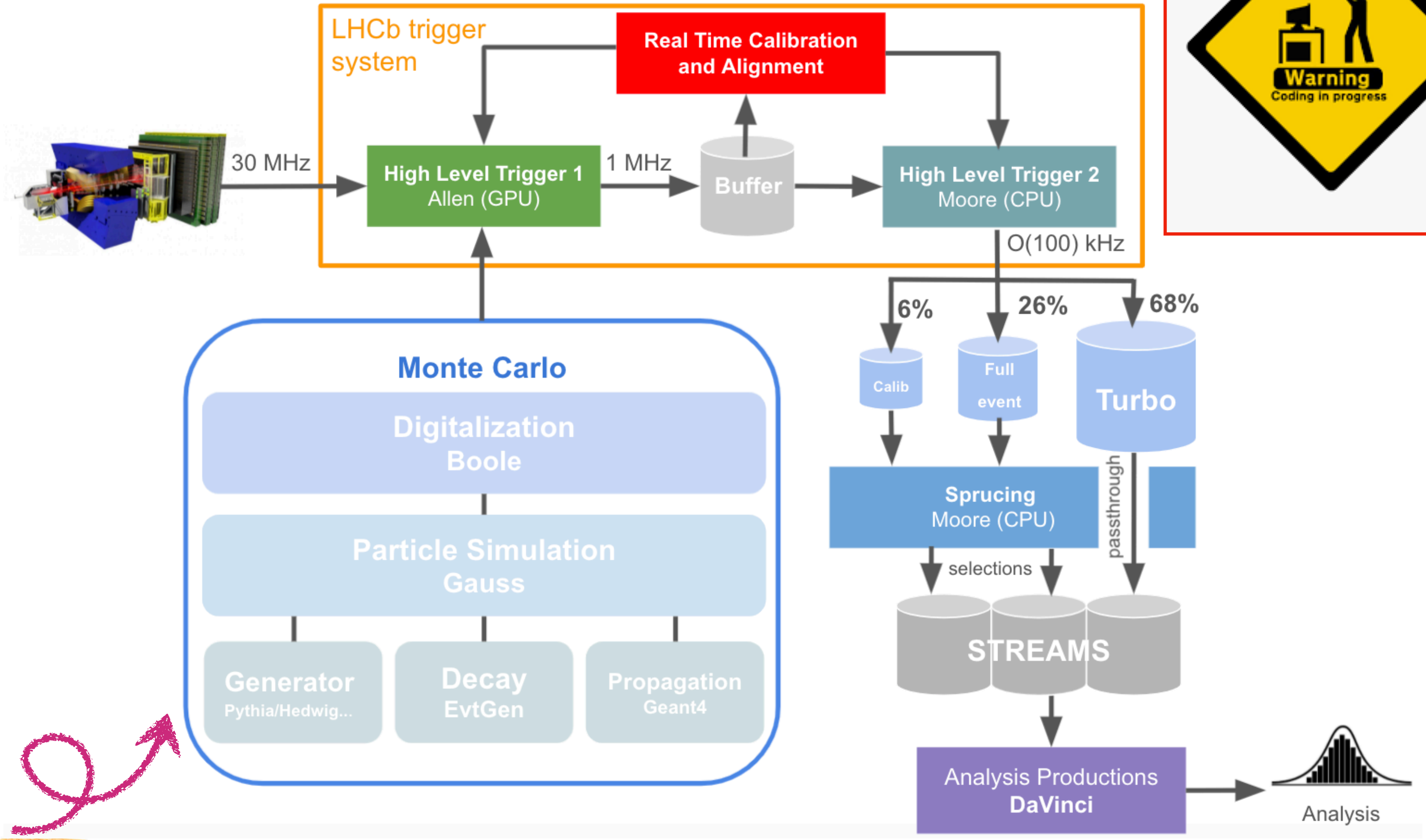
Documentation can be found here

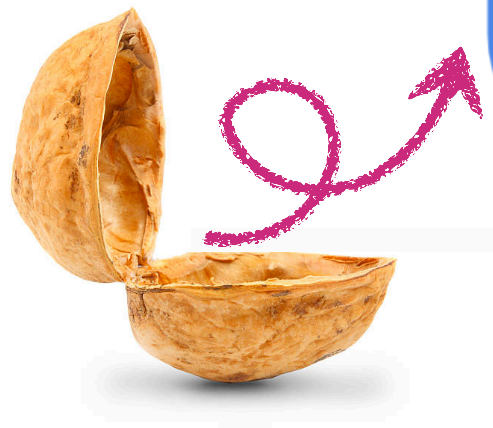
The LHCb Run 3 dataflow in a nutshell





The LHCb Run 3 dataflow in a nutshell

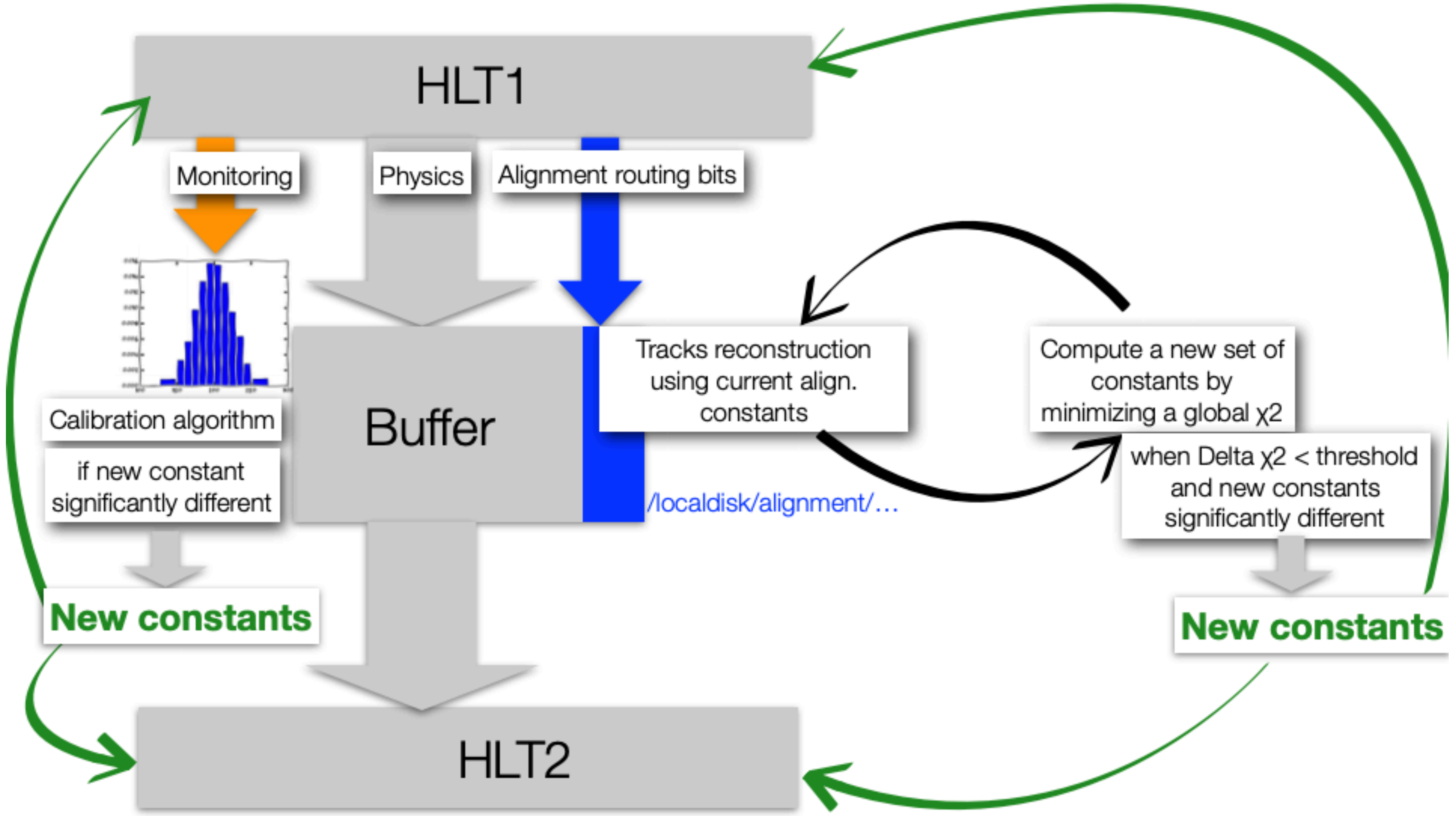




Thank you!

Backup

Between HLT1 and HLT2


HLT1
Monitoring
Physics
Alignment routing bits
Calibration algorithm
if new constant significantly different
New constants
Buffer
Tracks reconstruction using current align. constants
Compute a new set of constants by minimizing a global χ2
when Delta χ2 < threshold and new constants significantly different
/localdisk/alignment/…
New constants
HLT2