HLT FRAMEWORK

Vladimir Gligorov, University of Glasgow

On behalf of the HLT developers

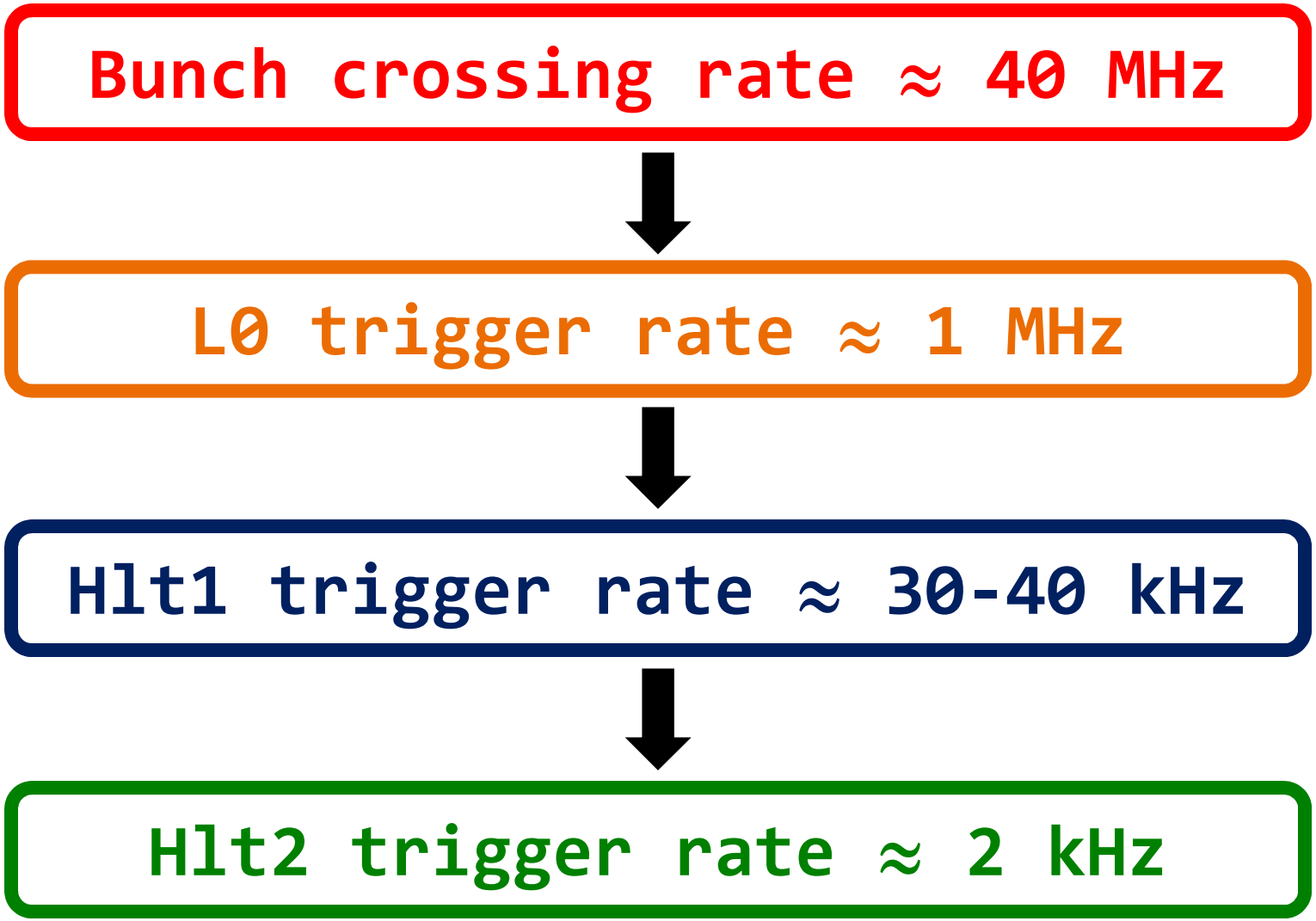
37th LHCb Software Week, CERN

17 June 2009

OVERVIEW

- Reminder of the trigger architecture
- Introduction to Hlt lines
- New Hlt framework
  - Hlt Lines
    - Hlt1 Lines
    - Hlt2 Lines
- Question time

TRIGGER STAGES (14 TEV, $2 \cdot 10^{32}$ LUMI)



HLT1 lines

HLT1 alleys and lines

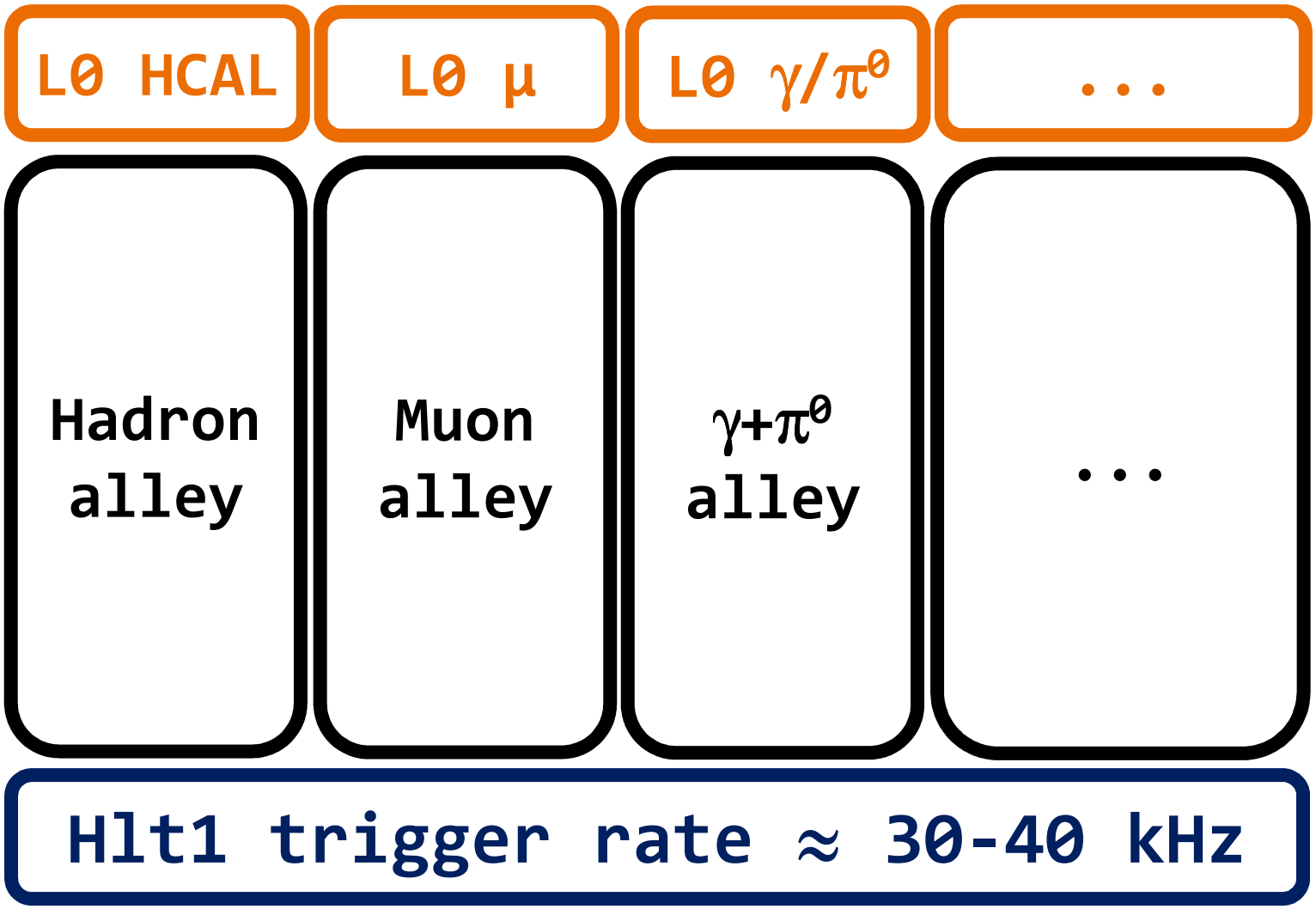
L0 trigger rate ≈ 1 MHz

Hlt1 trigger rate ≈ 30-40 kHz

HLT1 alleys and lines

Hlt1 trigger rate ≈ 30-40 kHz

HLT1 alleys and lines



EXAMPLE: HADRON ALLEY AND LINES

L0 HCAL

Hadron alley

Runs on all events which passed the L0 trigger becase of a high E_T HCAL object

EXAMPLE: HADRON ALLEY AND LINES

L0 HCAL

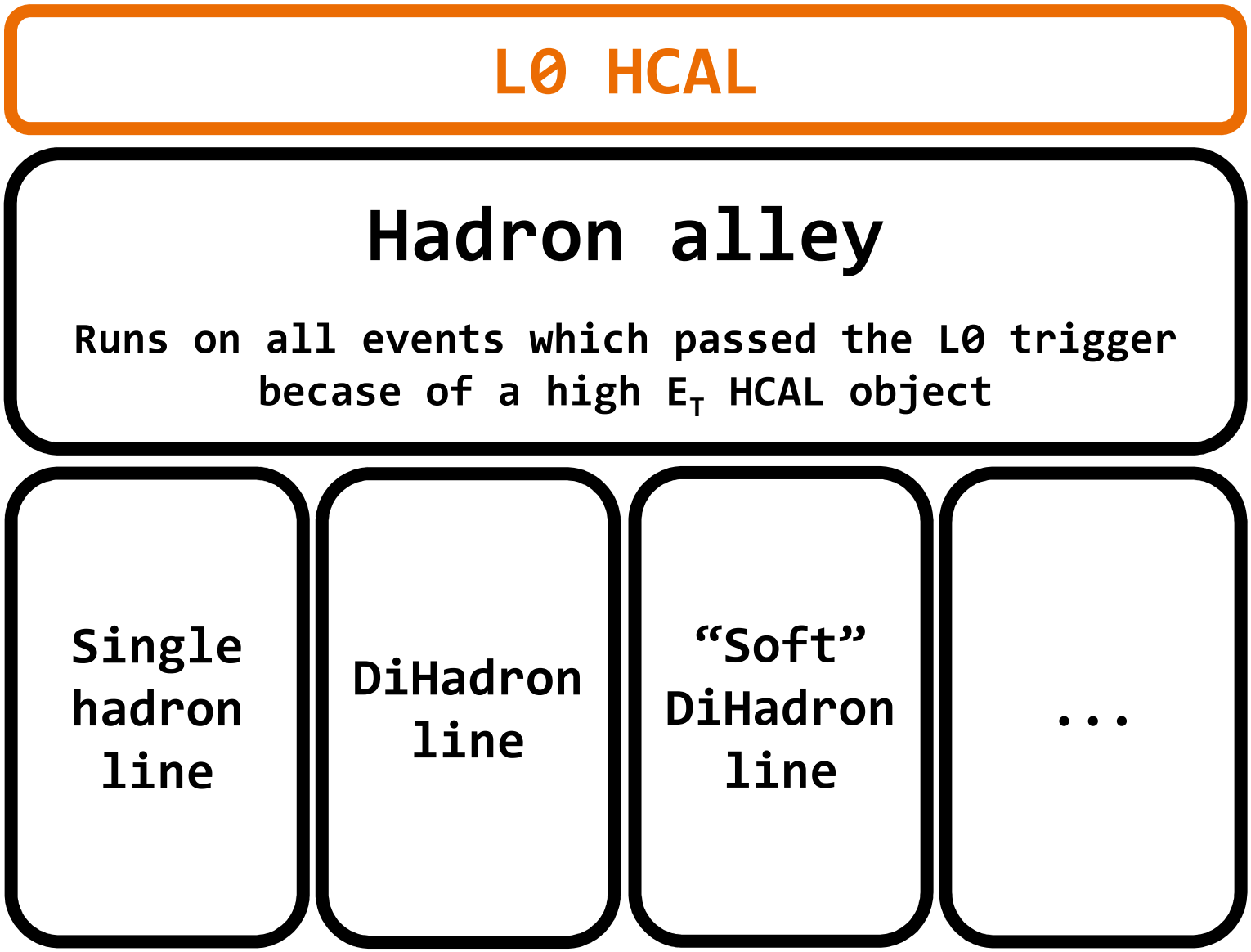
Hadron alley

Runs on all events which passed the L0 trigger becase of a high E_T HCAL object

Single hadron line

Attempts to match a long track to the L0-HCAL object; passes if the match is found and the matching long track has high IP and P_T

EXAMPLE: HADRON ALLEY AND LINES



SLOTS

- Each HltLine has a series of “slots” associated with it
- The “slots” define the values of the cuts used in the HltLine
- Because this is python we can
  - Define the line structure (cut types and order) in one file which is mostly static
  - Define the actual cut values in another file, and just read them into the “slots” when we wish to reconfigure the HLT

HLT1 LINES SUMMARY

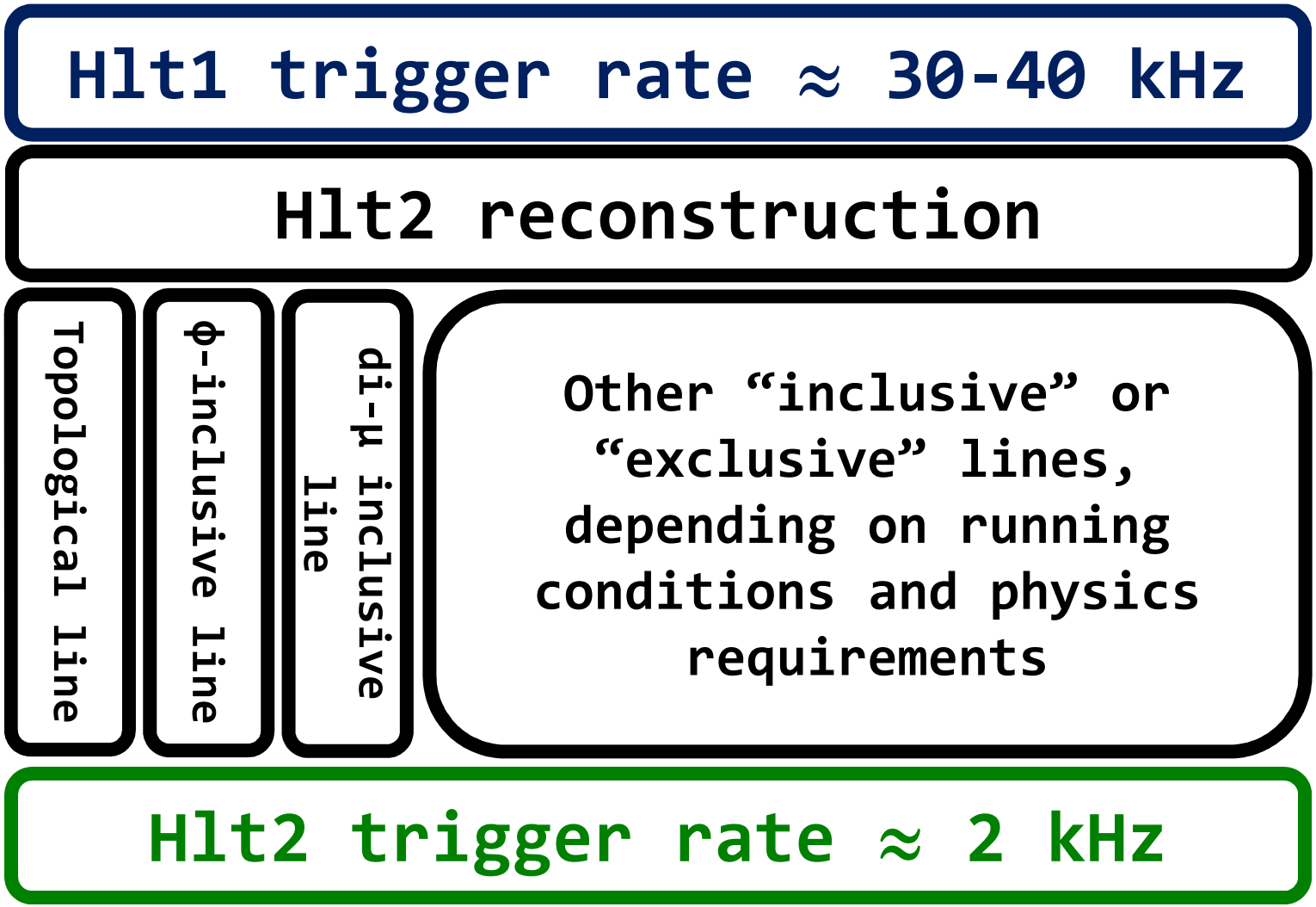
- Each HLT1 alley confirms a specific L0 triger decision
  - An HLT1 alley is a logical structure (no associated code)
- **Each HLT1 alley consists of multiple HLT1 lines** (physical classes which set the trigger decision)
  - **Each HLT1 line decision is independent of all other HLT1 line decisions**
  - **HLT1 lines may share reconstruction steps, executed at most once per event**

HLT2 lines

RECENT CHANGES

- The HLT2 framework has been upgraded to use HLT lines, much like HLT1
- **This does not change the underlying physics of the HLT2 trigger**
  - Confirmation between HLT1 and HLT2 is now trivial to implement but is in no way mandatory (and will not be!)
  - **Shouldn't change the rates and efficiencies (although there are changes because of the new L0xHLT1 bandwidth division)**

HLT2 lines overview



LINES ARE LINES

- HLT2 lines behave like HLT1 lines
  - They have slots for easy configuration
  - Each HLT2 line decision is independent of any other HLT2 line decision
  - An HLT2 line may depend on an L0 or HLT1 line decision (confirmation)
- HLT2 lines share the event reconstruction and intermediate particles (K^*,D,ρ,...)
  - Intermediate particles are made once per event: when first asked for.

CODE STRUCTURE

LESS PHILOSOPHY MORE DOCUMENTATION

- What lives where now?
  - HltConf: overall Hlt configuration
  - HltLines: the machinery which is shared between all Hlt lines
  - Hlt1Lines: the actual Hlt1Lines
  - Hlt2Lines: the actual Hlt2Lines
  - Hlt2SharedParticles: the shared intermediate states used in Hlt2

HLT CONF

- Now contains
  - Configuration.py – the HLT configurable, for use by DaVinci and Moore.
  - HltThresholdSettings.py – the file which defines a concrete bandwidth division for the HLT (currently only for Hlt1)
  - HltReco.py – the HLT reconstruction sequence definition.
  - Hlt1.py – a script which configures the Hlt1 lines (defined elsewhere)

HLT LINES

- Now contains
  - HltLine.{h,cpp} – the sequencer for the HltLines
  - HltLine.py – the code for defining and configuring Hlt1 and Hlt2 Lines
  - Hlt1Units.py – Defines some utilities for dealing with the “units” of HLT1 reconstruction: setup the TES output locations, track types, names, etc.

HLT1/2 lines

- Now contains
  - HltXXXLines.py – the definitions of the actual Hlt1/Hlt2Lines, meaning the cuts used, their order, and the “slots” definitions

SUMMARY

HLT framework summary

- Whole HLT now uses lines
  - Each Hlt line is independent of any other Hlt line (at the same stage)
    - Hlt2 lines could in principle depend on an Hlt1 line, none do so far
  - Hlt lines enforce naming conventions and have “slots” for easy configuration
    - Define the line structure in one place, define some cuts for a concrete running scenario elsewhere
- HLT packages split up in a more logical fashion, should help maintenance

FURTHER DOCUMENTATION

- See the TWiki

  https://twiki.cern.ch/twiki/bin/view/LHCb/LHCbTrigger

- If something in the TWiki is unclear, please e-mail whoever is responsible for the page and ask for clarification...

BACKUP AND TECHNICAL DETAILS

HLT2 SHARED PARTICLES AND MEMBERS

- **Hlt2SharedParticles** are intermediate states made once per event
  - They have their own package called **Hlt2SharedParticles**
- **Hlt2Member** is a wrapper around a CombineParticles/FilterDesktop algorithm
  - Enforces some naming conventions, pythonizes I/O
  - Allows automatic cloning of line algos when cloning a line
  - Technical details in backup slides...

HLT2 LINE SYNTAX

```
myLine = Hlt2Line(    name = "myLine",
                      prescale = 1.0,
                      L0DU = [L0 decision to require],
                      HLT  = [Hlt1 decision to require],
                      algos = [ algo1, algo2, ...],
                      postscale = 1.0
                  )
```

HLT2 line syntax (2)

```
myLine = Hlt2Line(    ...

                      algos = [ algo1, algo2, ...],

                      ...
               )
```

Algo1, algo2, etc. can be any GaudiAlgorithms/Sequencers or Hlt2Members (more on next slide)

Fundamental difference when cloning the line

- Any concrete Algorithms/Sequencers do not get cloned: the same instance is used by both the original and cloned line.
- Any Hlt2Members do get cloned: the Hlt2Member of the cloned line is a new instance.

HLT2 MEMBERS

```
myHlt2Member = Hlt2Member(    Type = CombineParticles,
                             Name = "myHlt2Member",
                             InputLocations = [] )
```

Type = CombineParticles or FilterDesktop (not a string!)

Naming convention : to use the Hlt2Member in other pieces of code, its name is **TypeName**

e.g. CombineParticlesmyHlt2Member

CLONING A LINE

```
myLine = Hlt2Line(    “MyLine”,
                      prescale = 1.0,
                      algos = [...,myHlt2Member,...]
                )
```

Can modify Hlt2Members on the fly when cloning

```
myLine.clone = (      “MyClonedLine”,
                      ...,
                      CombineParticlesmyHlt2Member = {
                            “Property1” : “New value”,
                            “Property2” : “New value”,
                            ...
                      }
                )
```

HLT2 SHARED PARTICLES

- Intermediate particles that are made once per event, used to live in HltSelections
- Now have their own package called
  - Hlt2SharedParticles
- To include them in a line, just do

```
from Hlt2SharedParticles.MyPart import MyPart2MyFinalState

line = Hlt2Line( ...
                 algos = [..., MyPart2MyFinalState, ...]
                 ...)
```

LINE INDEPENDENCE: REFIT EXAMPLE

Robust Hlt2 stage: use standard Hlt2 reconstruction, reduce rate

Kalman-fit forward tracks

Post-fit Hlt2 stage: use significance cuts to reduce rate further

DO WE WANT TWO LINES?

- It might seem tempting to define a robust line and then demand that it had passed in the post-fit line

```
myPostFitLine = Hlt2Line( ...,

                          HLT  = [Hlt2RobustDecision],

                          ...)
```

Don't do it!

- You DO want a line to monitor the robust part, but done in a different way...

YES, BUT IN A DIFFERENT WAY…

- There is no need to link the lines: just link their reconstruction

```
myRobustFitLine = Hlt2Line( ...,

                  algos=[...,RobustReco,...],

                  ...)

myPostFitFitLine = Hlt2Line( ...,

                  algos=[...,
                         RobustReco,
                         TrackFit,
                         PostFitReco,
                         ...],
                  ...)
```

TRIGGER STAGES

Bunch crossing rate ≈ 40 MHz

- **Non-zero interaction rate ≈ 10 MHz**

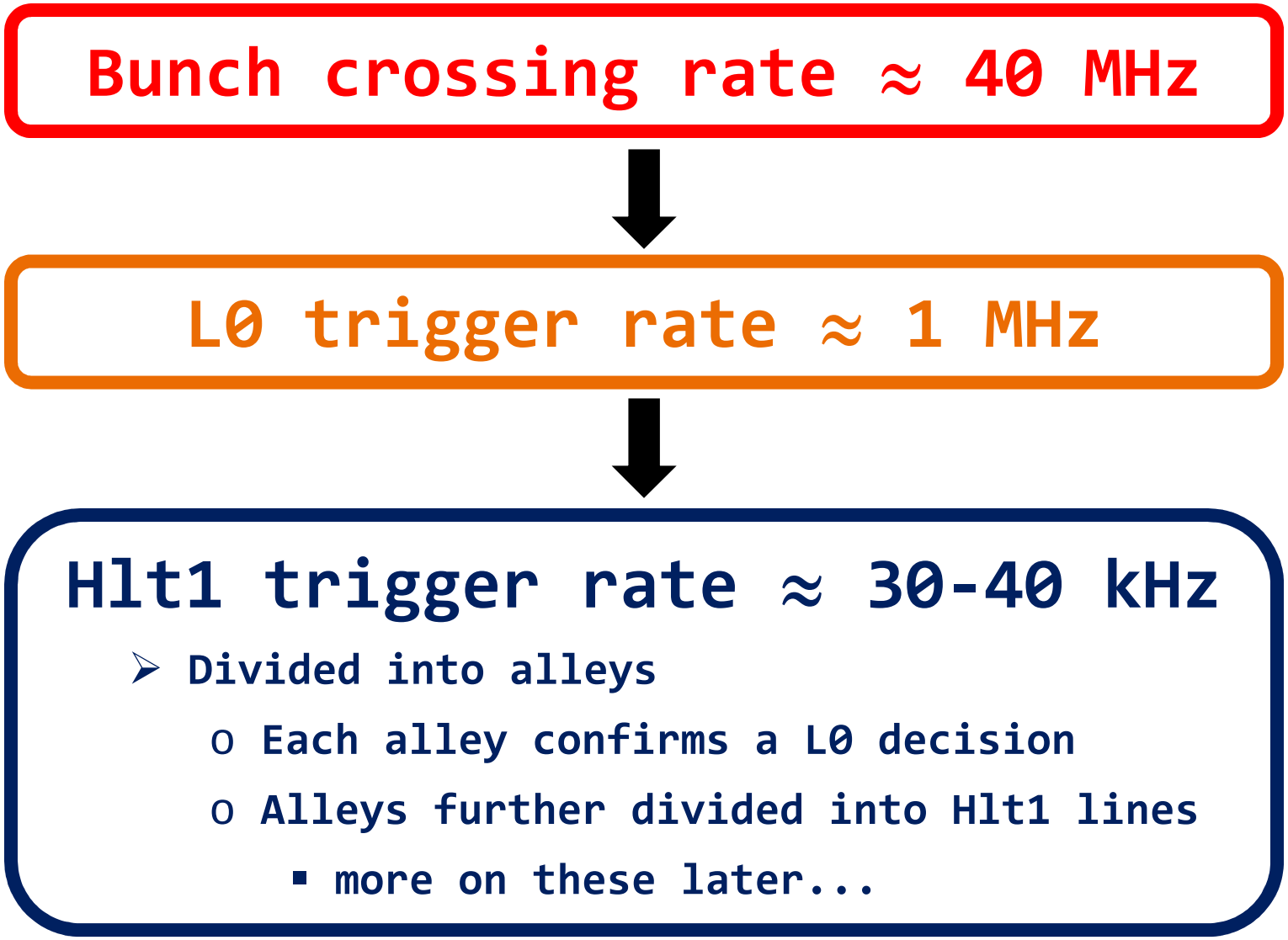
TRIGGER STAGES

Bunch crossing rate ≈ 40 MHz

L0 trigger rate ≈ 1 MHz

- Hardware high-E_T trigger
- Hadron/(di)$\mu^{\pm}/\gamma/e^{\pm}/\pi^0$ decisions

TRIGGER STAGES



TRIGGER STAGES