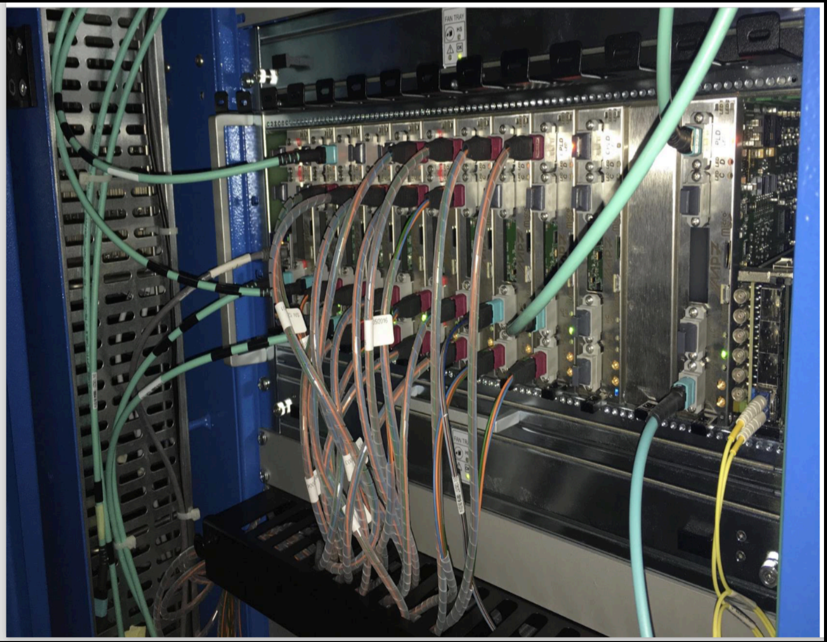
Time-Multiplexed Track-Trigger for CMS Detector Upgrade Overview

Ian Tomalin (RAL)

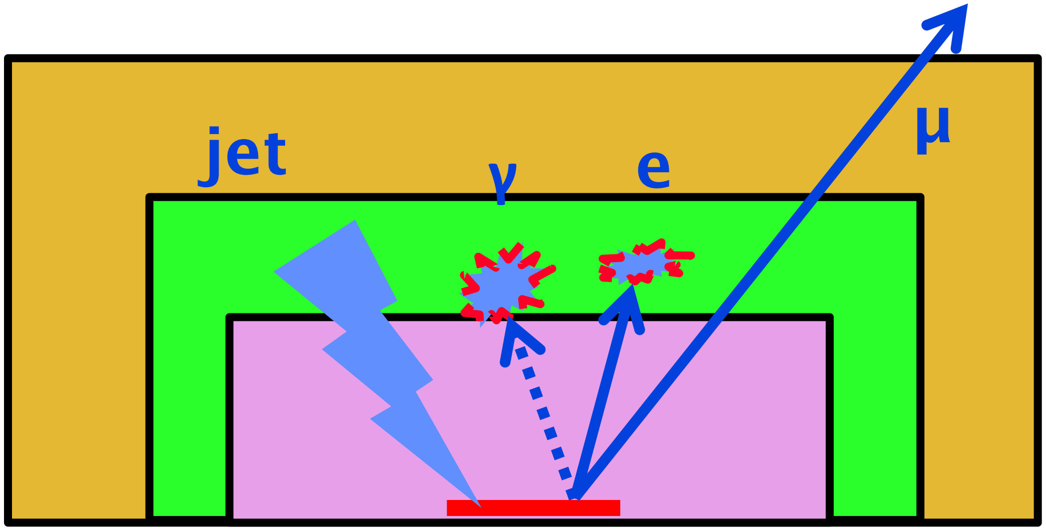


Science & Technology Facilities Council Rutherford Appleton Laboratory
INFIERI
CERN
Imperial College London
University of Bristol
Karlsruhe Institute of Technology
ÖAW Austrian Academy of Sciences
Brunel University London
MAXELER Technologies

Introduction

- From 2025+, HL-LHC will produce an event containing ~140 pp collisions every 25ns. (~5 times current rate).
- CMS L1 trigger must select interesting events within ~12 μs (max. data buffering time), whilst rejecting ~99% of boring events.
- To achieve this, L1 trigger will use not only data from CMS calorimeter & muon chambers (traditional), but also charged particle tracks reconstructed in the CMS tracking detector (new).
- **Our goal is to reconstruct these tracks within ~4 μs.**
  - **Tricky: LHC will produce ~5000 charged particles every 25 ns!**

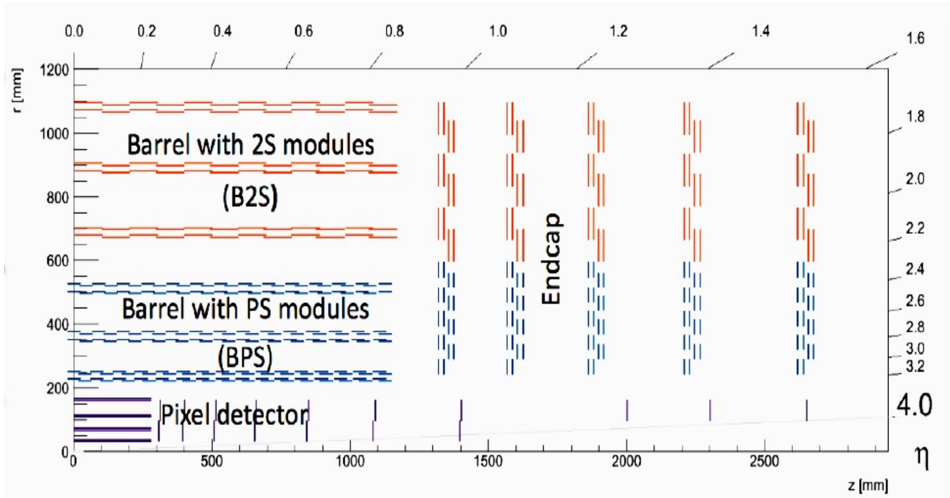
- Why do tracks improve L1 trigger performance?
  - Muons: tracks improve Pt resolution
  - Electrons: tracks distinguish them from photons
  - Jets + EtMiss: tracks check they come from main pp collision vertex, not from boring pileup vertices.

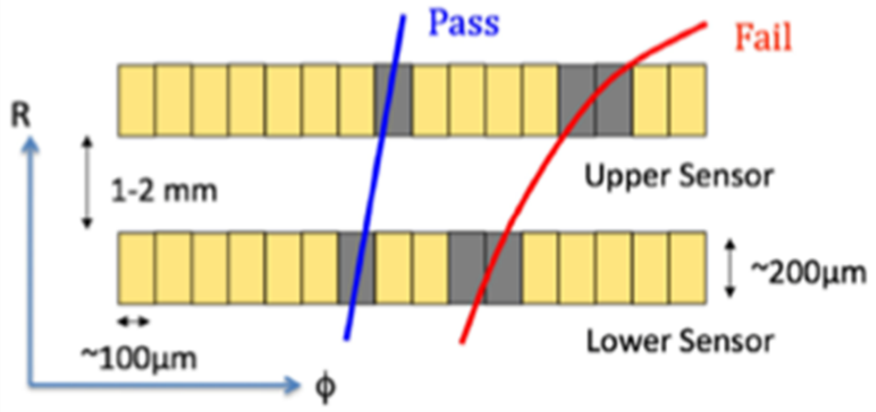


The upgraded CMS Tracking Detector

The CMS silicon tracking detector will be replaced for 2025+ :

- Each tracker module consists of 2 closely spaced silicon sensors.
- A charged particle produces a pair of hits (known as a `stub') in these two sensors.
- Assuming the particle originates from the LHC beamline, the relative position of the two hits determines the track Pt.

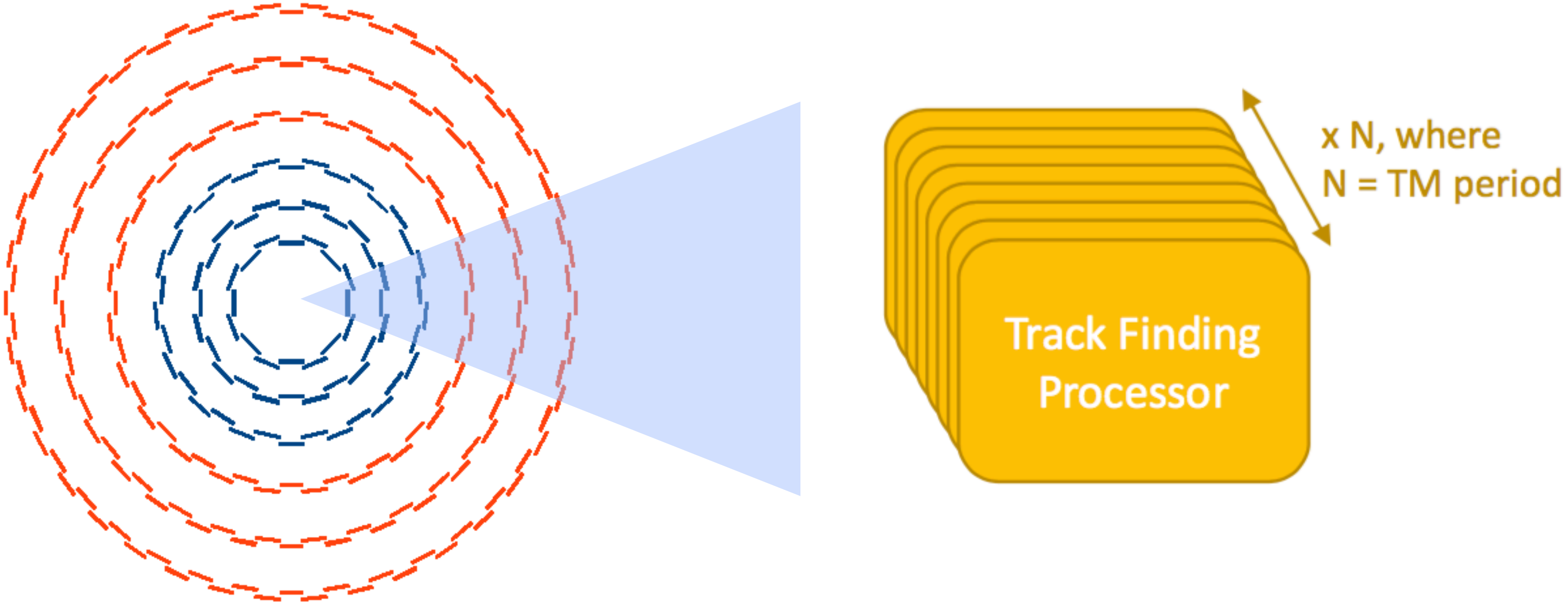




- On-detector electronics transmits only stubs consistent with Pt > 2 GeV to off-detector electronics, reducing by factor ~30 the number of stubs that L1 track-finding electronics must handle.

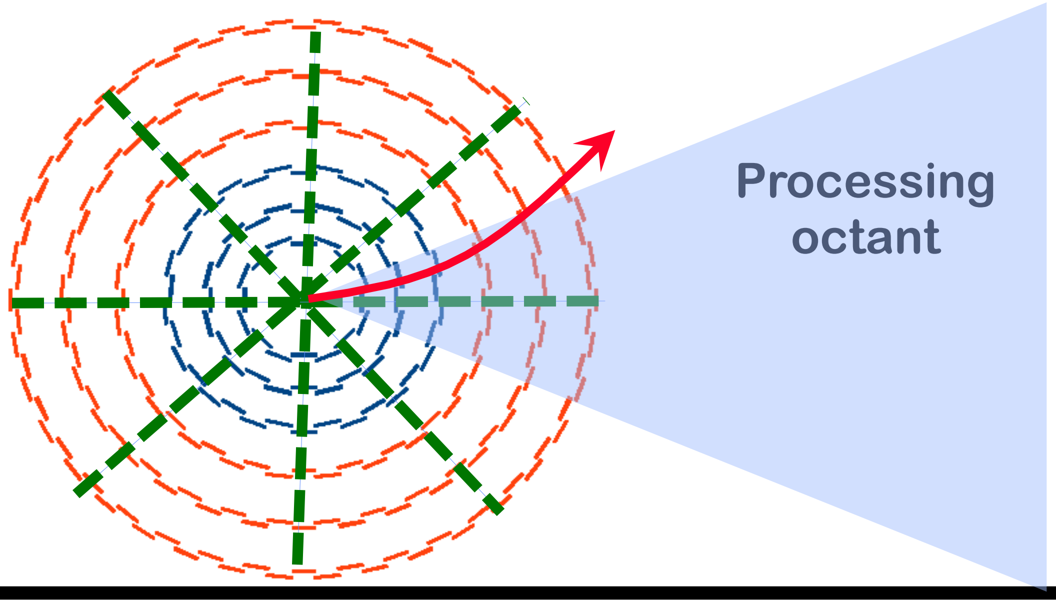
Time-Multiplexed Track-Finder: overall approach

- Our "track-finding processor" is off-detector electronics containing FPGA(s).
- Each track-finding processor reconstructs all the tracks within one φ octant of the tracker,(known as a "processing octant"), for just one LHC event in 36.
- We thus need 36 track-finding processors to reconstruct all the tracks within a φ processing octant in all events. The 1st processor reconstruct tracks from the 1st event, the 2nd processor those from the 2nd event etc.
  - This is time-multiplexing



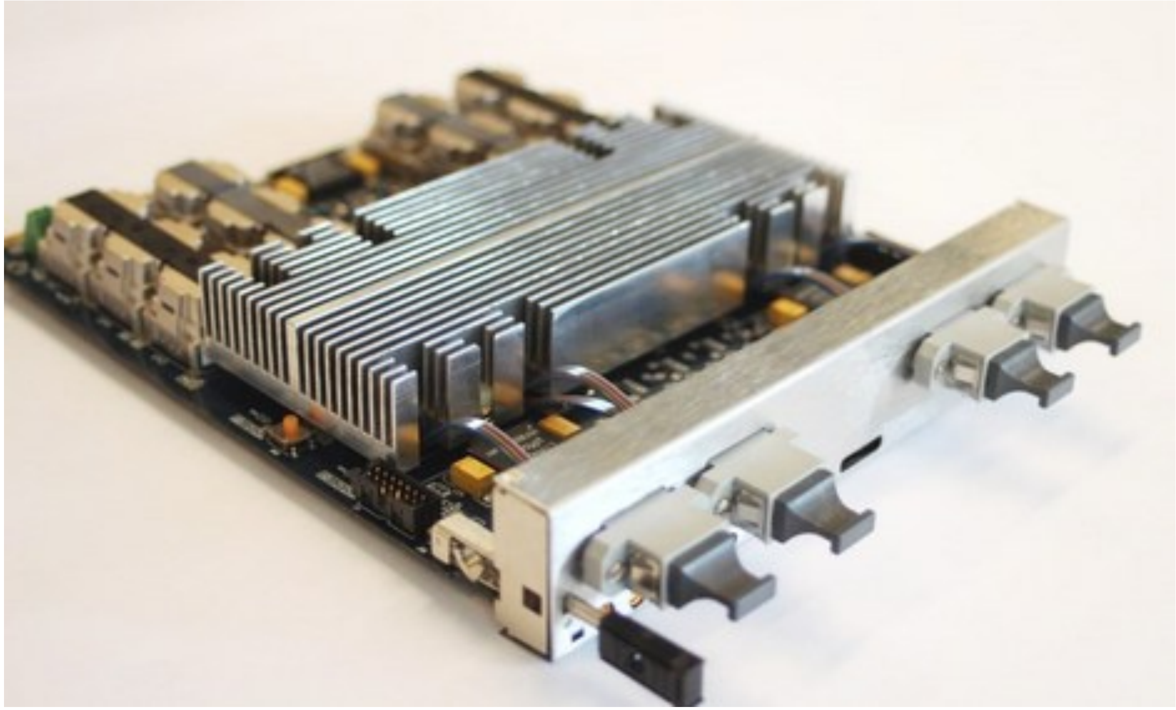
Accommodating constraints from CMS tracker cabling scheme

- Seen in r-φ plane (perpendicular to LHC beam), the tracking detector is divided into φ "detector" octants, each of which are read out by different groups of off-detector FPGA boards named "DTC".
- The DTC calculate the coordinates of each signal (= "stub") in our silicon tracker from the module number & pixel/strip number it is in.
- Our processing octants read data from the DTCs, and are rotated by half an octant w.r.t. the detector octants.
  - Motivation: No processing octant needs data from more than 2 detector octants, despite track curvature in B-field.



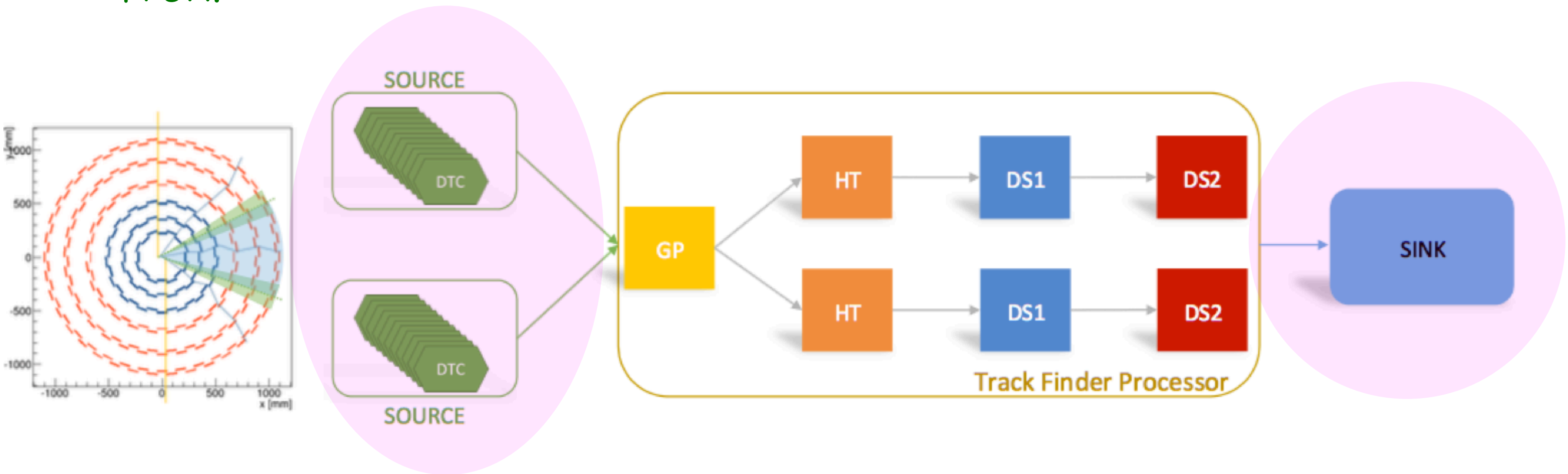
TMTT Demonstrator

- The hardware demonstrator we have constructed to prove this concept is viable, corresponds to one track-finding processor.
  - i.e. It reconstructs tracks in an entire φ octant of the tracker for one LHC event in 36.
- We have implemented it on a number of "MP7 boards", which is μTCA card equipped with a Virtex7 FPGA with ~1 Tb/s I/O capacity.



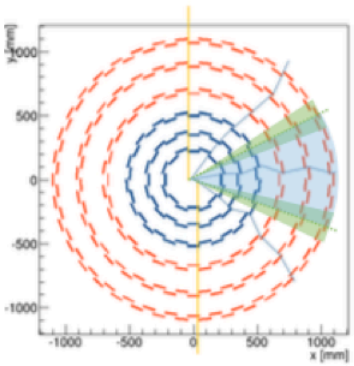
TMTT Demonstrator

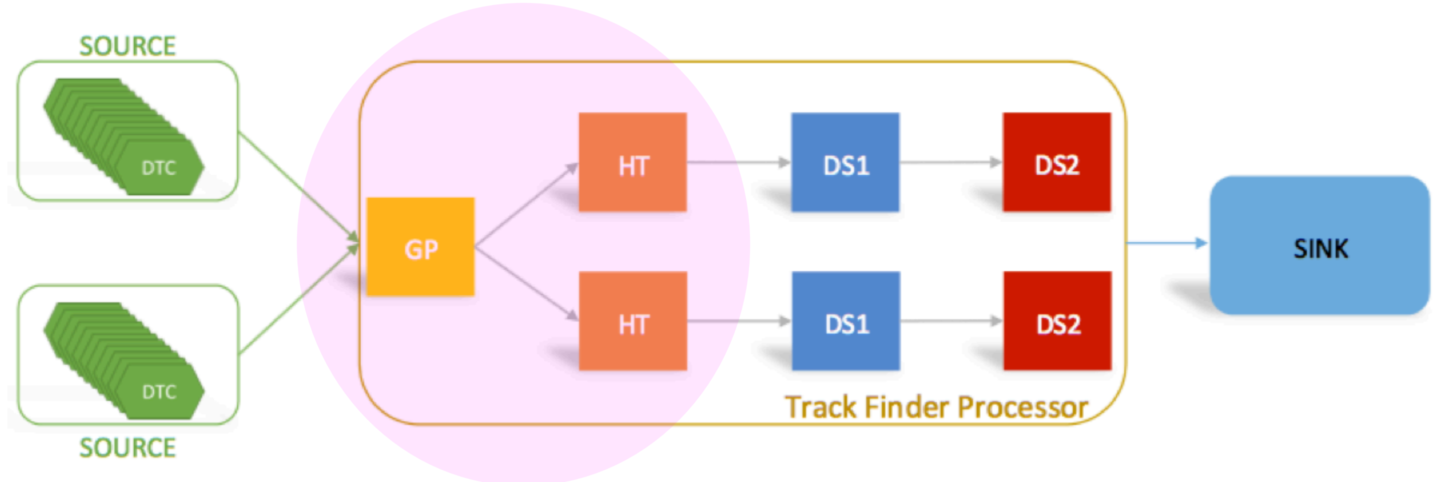
- Our demonstrator uses 5-7 MP7 to implement the track-finding processor.
- For the final 2025 system, it should be possible to replace these with a single FPGA.



- The MP7 in our demonstrator have the following roles:
  - "Source" = transmits stub coordinates from simulated LHC events to the track-finding processor, in DTC format.
  - "Sink" = receives tracks found by track-finding processor.

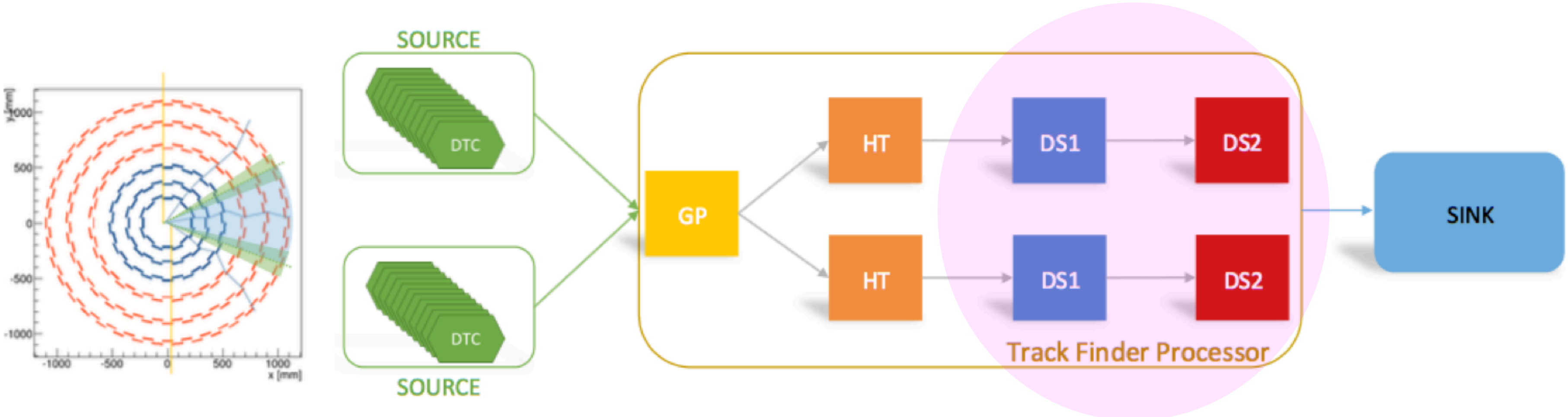
TMTT Demonstrator





- Additional MP7 in our demonstrator have the following roles. (Luigi's talk for details):
  - "Geographic Processor (GP)" = subdivides each processing octant into 18 sectors in polar angle times 2 sectors in azimuthal angle. And assigns each stub to one or more of these sectors.
  - "Hough transform (HT)" = does track-finding in the r- φ plane (perpendicular to LHC beam axis), with track-finding being run independently in each sector, using only the stubs assigned to that sector.

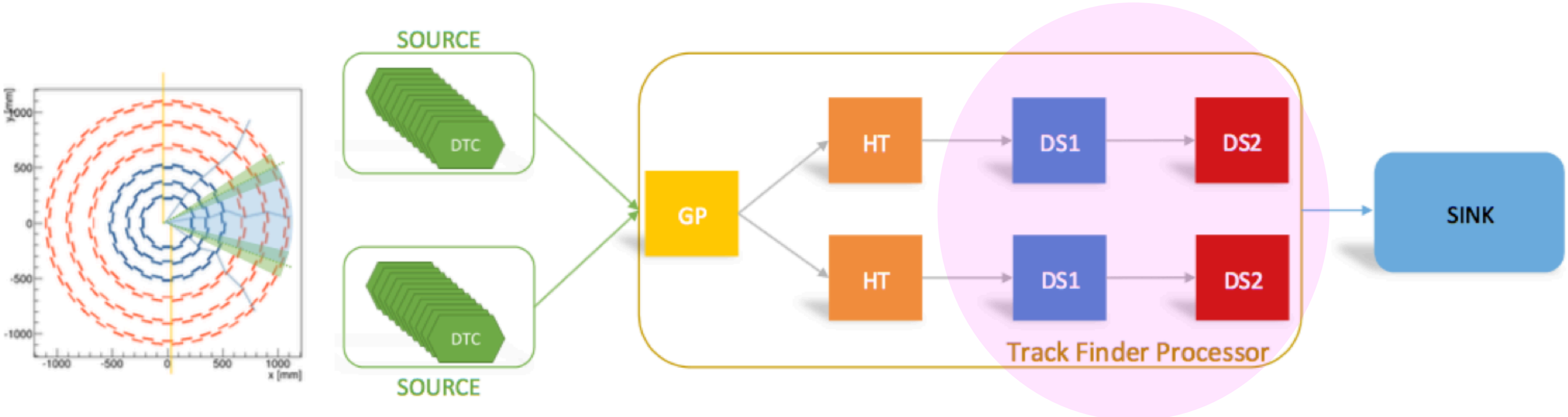
TMTT Demonstrator



- Additional MP7 further downstream have roles still being finalized: (Davide's talk for details):

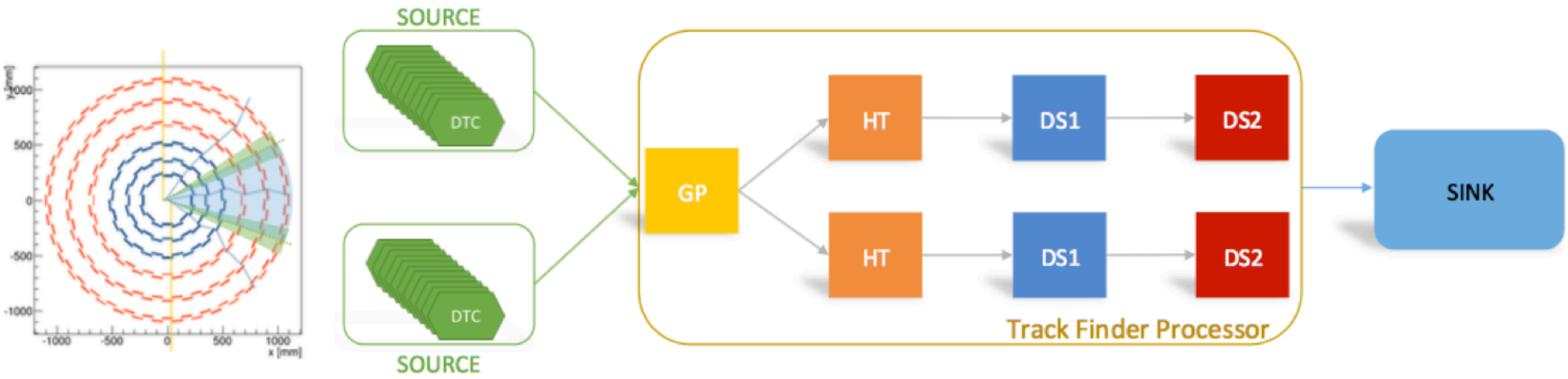
    - "R-Z Seed Filter (SF)" = checks that the tracks found by the Hough transform are consistent with straight lines in the r-z plane, so cleaning up the tracks.

    - "Track Fitter" = Does a fit to the stubs assigned to each track, so as to determine the track trajectory (=helix parameters). This also further cleans up the tracks by rejecting incorrect stubs.



- *Additional MP7 are used to kill duplicate tracks . (i.e. Cases where we accidentally reconstruct a single particle as two or more tracks).*
  - These "Duplicate Removal (DR)" algorithms are explained in detail in Luis' talk.

TMTT Variants

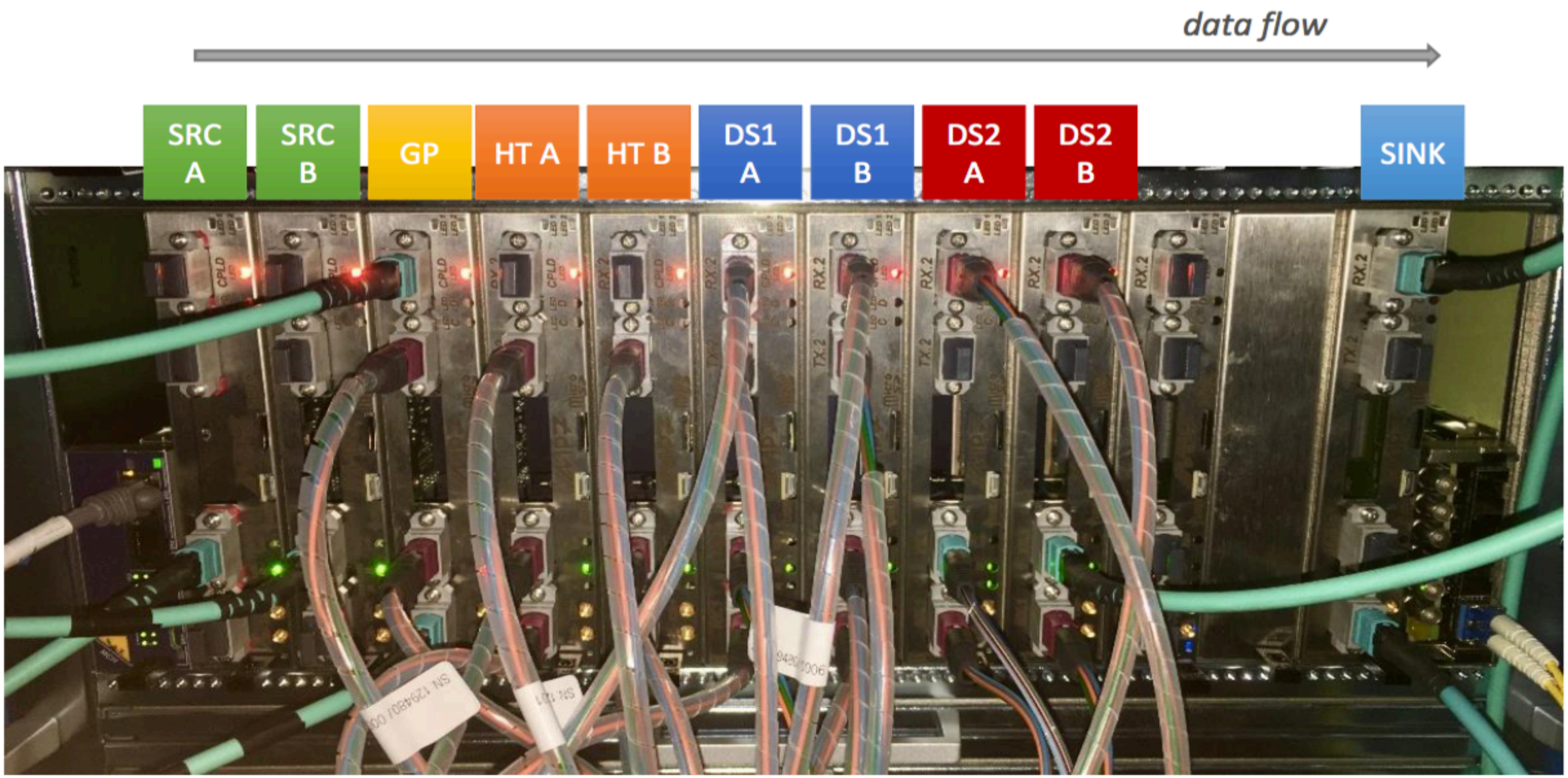


- We are exploring a few variants of this basic design:

  1) Don't use any "R-Z Seed Filter", so saving money. This option relies on the track fitters being capable of processing all the tracks output by the Hough transform, & successfully cleaning up the incorrect stubs on these tracks.
  2) We have two alternative track fitting algorithms: "Kalman Filter" (sophisticated) & "Linear Regression" (simple)
  3) Run the "Duplicate Track Removal" before or after the "Track Fit". We also have two alternative duplicate track removal algorithms.

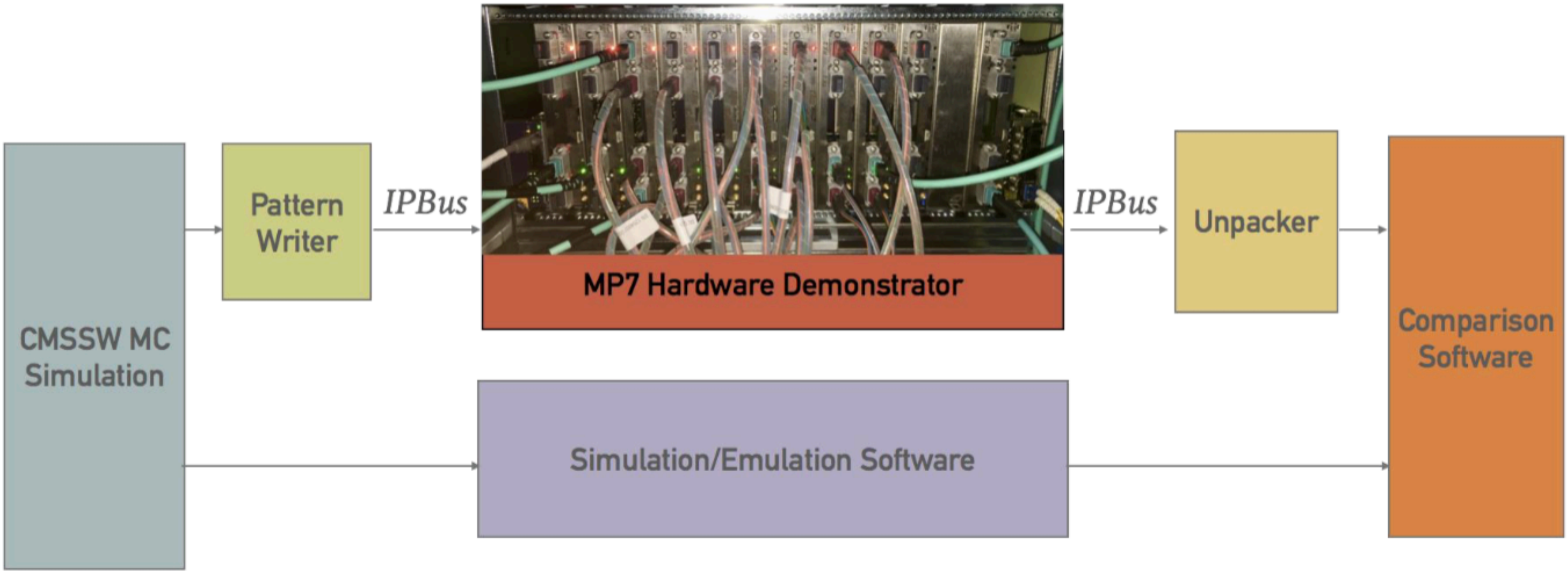
TMTT Hardware Status

- All MP7 boards needed for our demonstrator are in a μTCA crate at CERN, connected by timed-in optical links.
- All the firmware modules are running in an MP7 (i.e. GP, HT, SF & DR), except for the track fitters (firmware still being finished) & one of the two duplicate track algorithms.



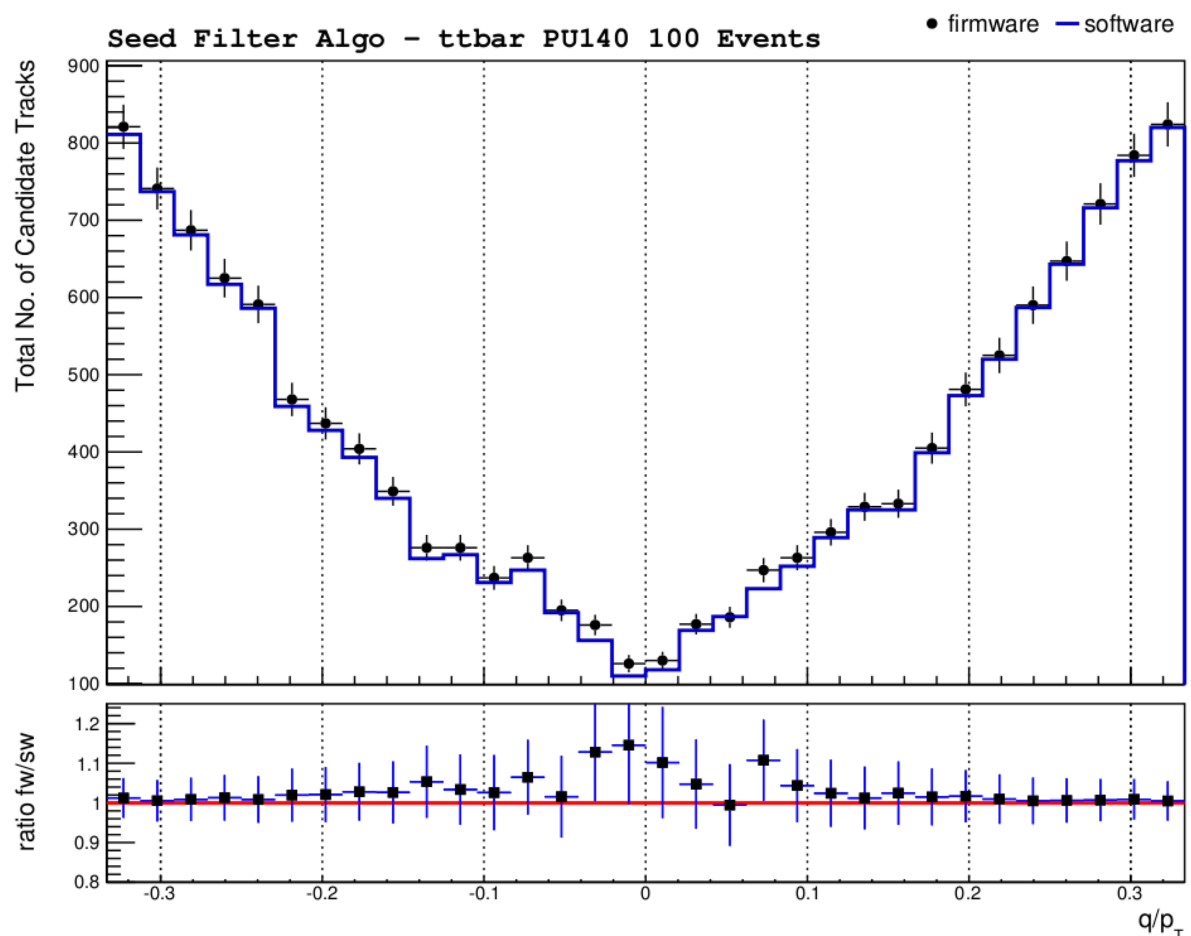
TMTT Validation

- We take stubs from simulated physics events ("ttbar + 200 pileup") & inject them both into our hardware demonstrator & into C++ software that emulates what the hardware should do.
- Additional C++ software compares the tracks found in hardware & software to check that they are identical.
- Running on hardware is particularly useful for identifying truncation effects due to finite band-width etc.



TMTT Validation

- Example: comparison of tracks found by R-Z Seed Filter in hardware (points) & software (histogram).
- The distribution of track q/P_T is shown.



Conclusions

- We already have most key components of our L1 tracking chain running on our hardware demonstrator, processing data from events with 140-200 PU, & giving good agreement with predicted tracks from software analysis.

  Namely:
  - Geographic Processor
  - r-φ Hough transform.
  - r-z Seed Filter.
  - One duplicate track removal algorithm.

- Track fitting firmware almost finished.

Aim to integrate remaining components into our hardware demonstrator in coming weeks.

Now Luigi, Davide & Luis will tell us more about the status of these individual components!