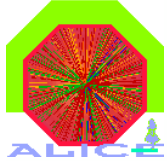


Dimuon HLT Design and AliRoot Intergration

By Artur Szostak

UCT-CERN Research Group

UCT CERN

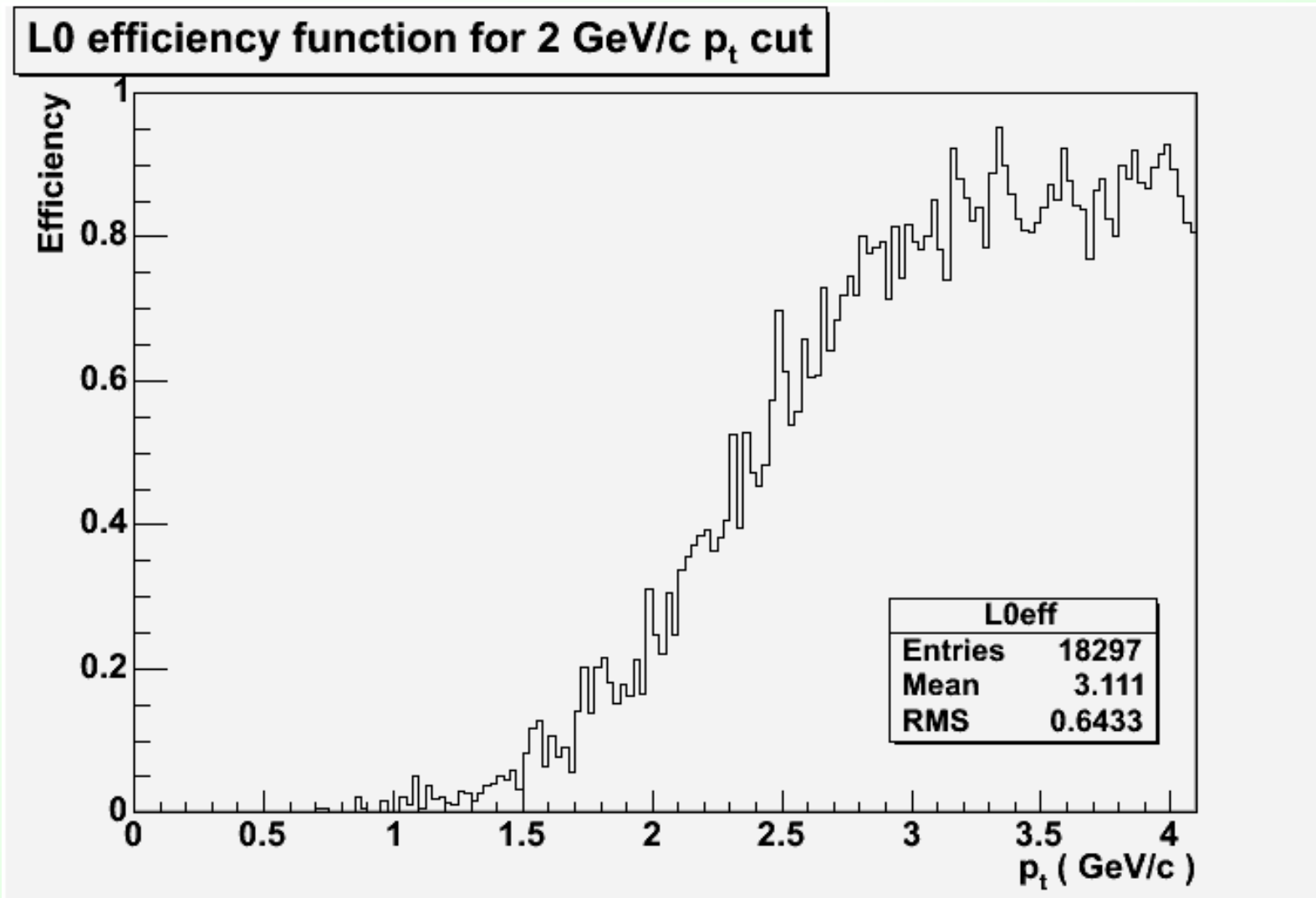


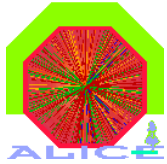
Function of Dimuon HLT (dHLT)

- Improve Pt cut resolution.
- Identify dimuon candidates.
- Reduce data rate to storage by removing background.
- Try improve on L0 trigger efficiency.

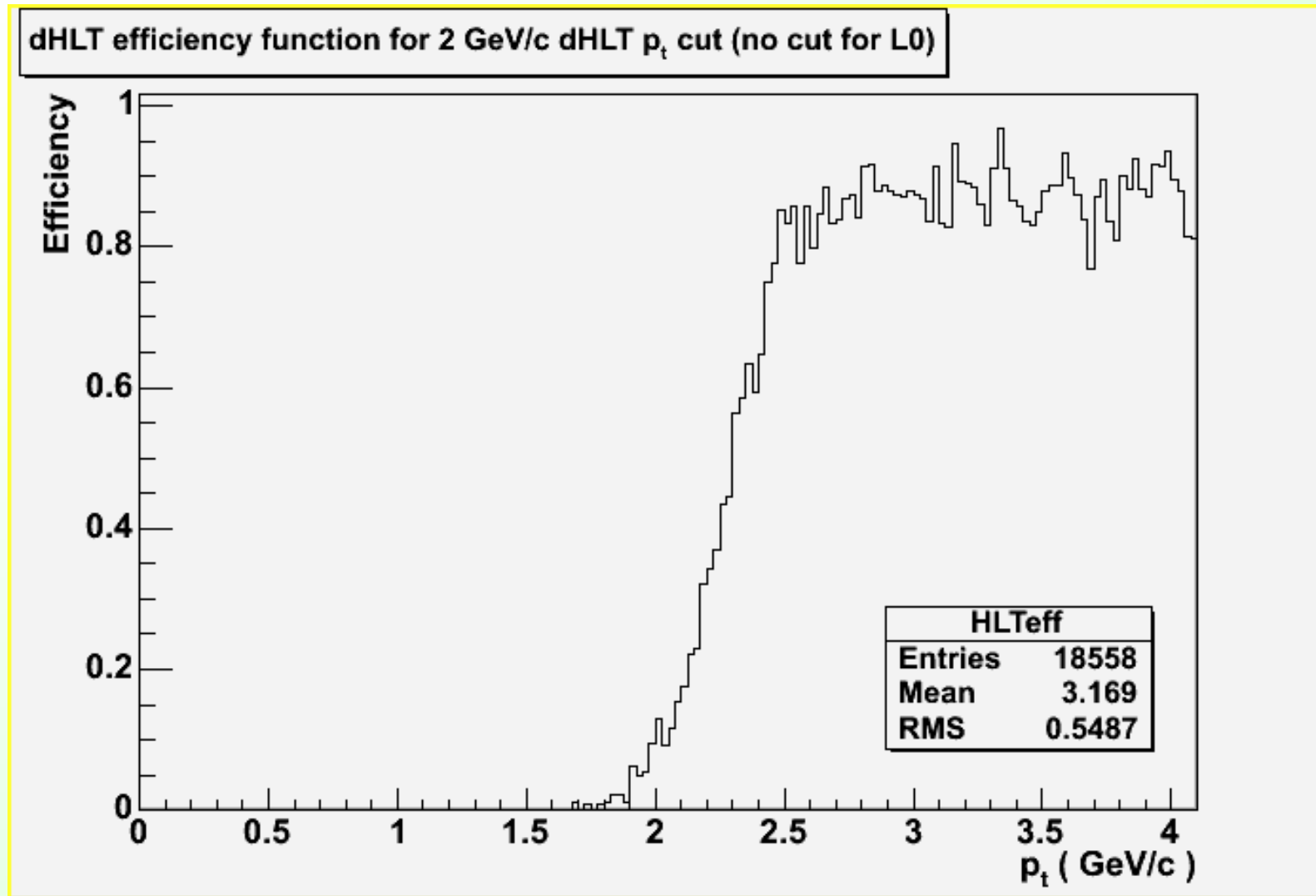


Improvement in dHLT Pt Cut.





Improvement in dHLT Pt Cut.



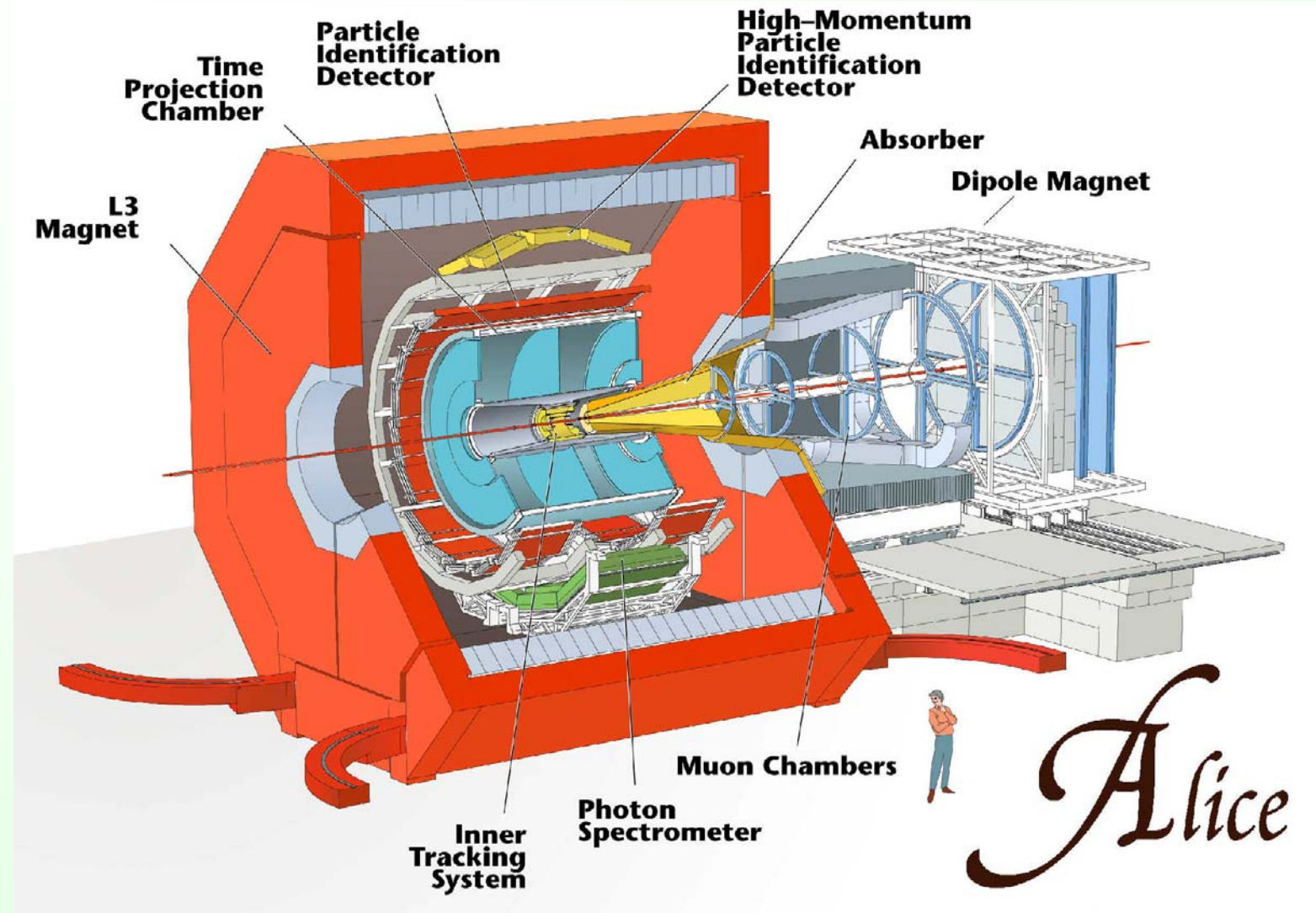


Some Basic Facts:

- Spectrometer composed of
 - 10 RPC tracking chambers, with bending (Y axis) and non bending (X axis) cathode planes.
 - 4 RPC trigger chambers, behind 2m muon filter.
 - Large cone absorber in front of whole spectrometer.
 - Dipole magnet centred on tracking chambers 5 and 6.
- Total event size ~ 500 kBytes.
- About 150 tracks in spectrometer per central event.
- L0 trigger rate = 1. kHz
- L0 latency $\sim 1.6 \mu\text{s}$

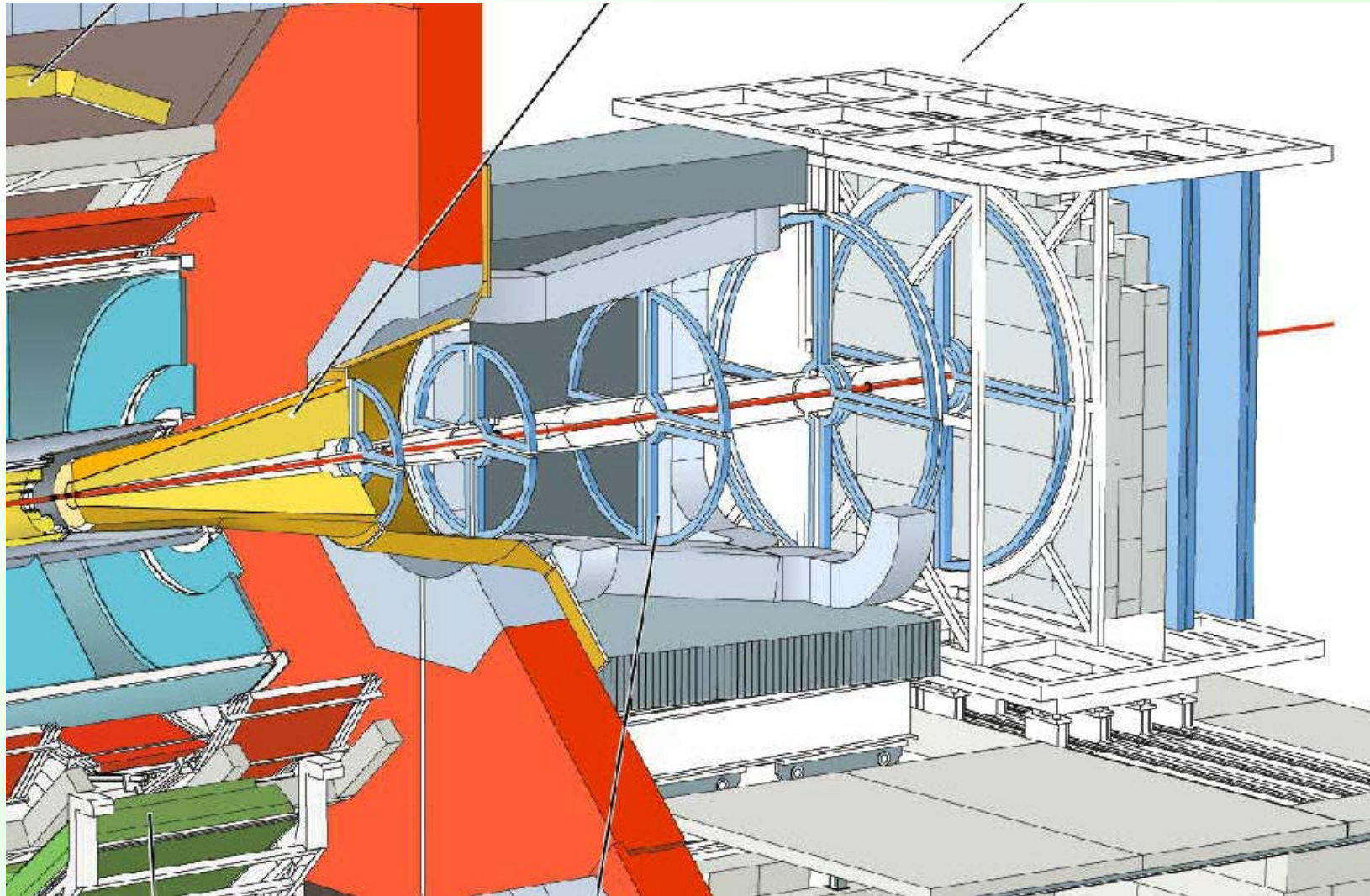


Spectrometer's Layout



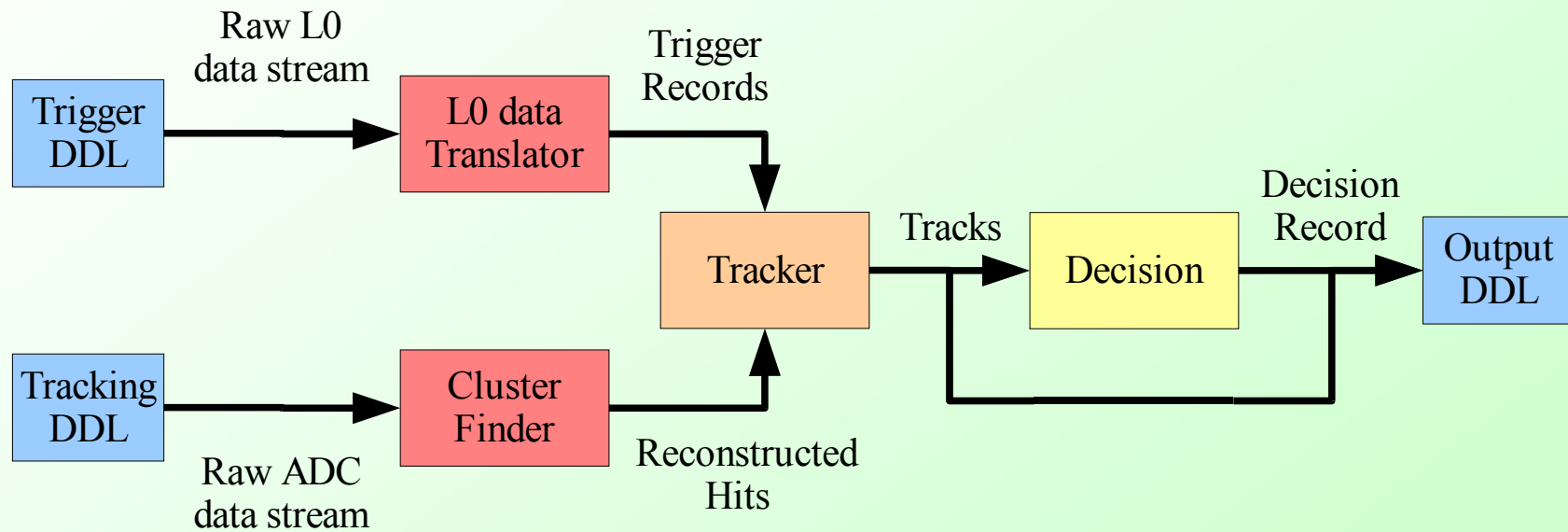


Spectrometer's Layout





dHLT Components

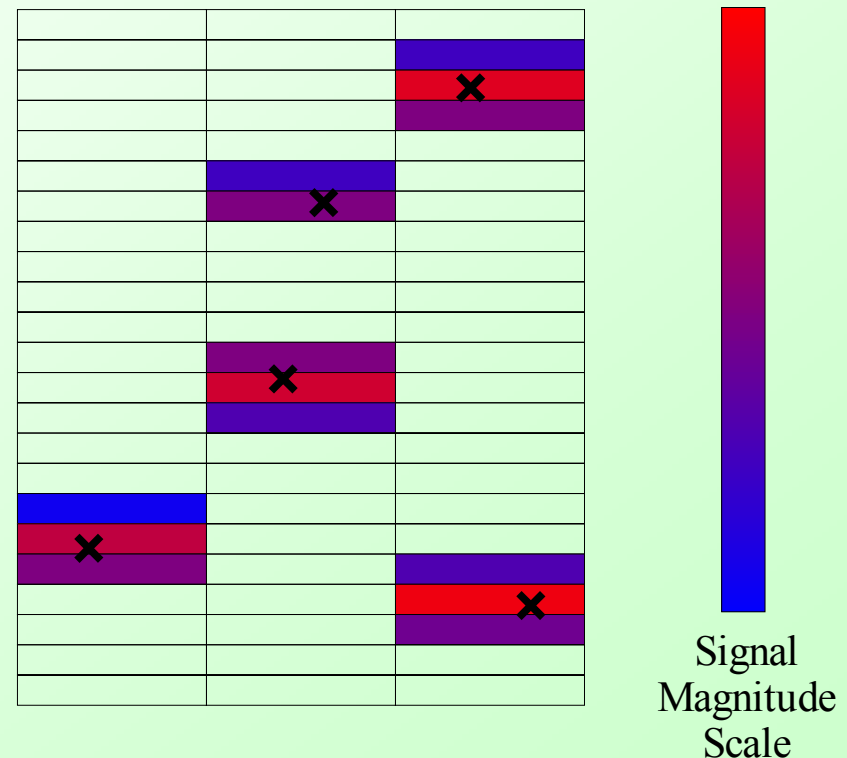


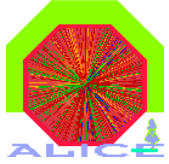


Cluster Finder Algorithm

- Find maxima along pad column.
- Find centre of gravity (weighted by charge) of pad with maximum and its nearest neighbour above and below.
- This becomes the reconstructed hit point.

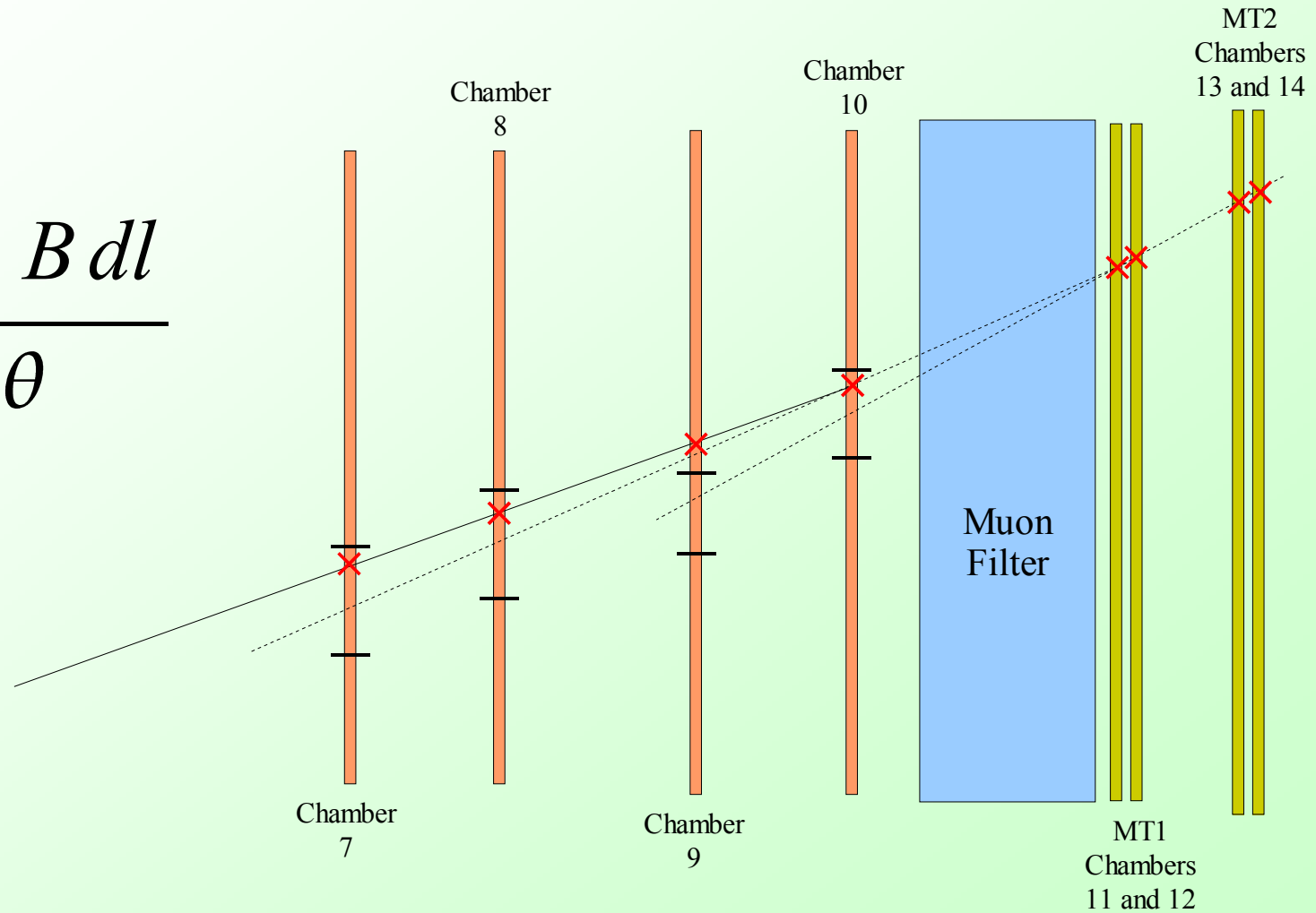
Example of chamber pads with hits:

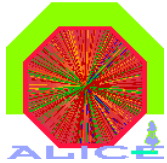




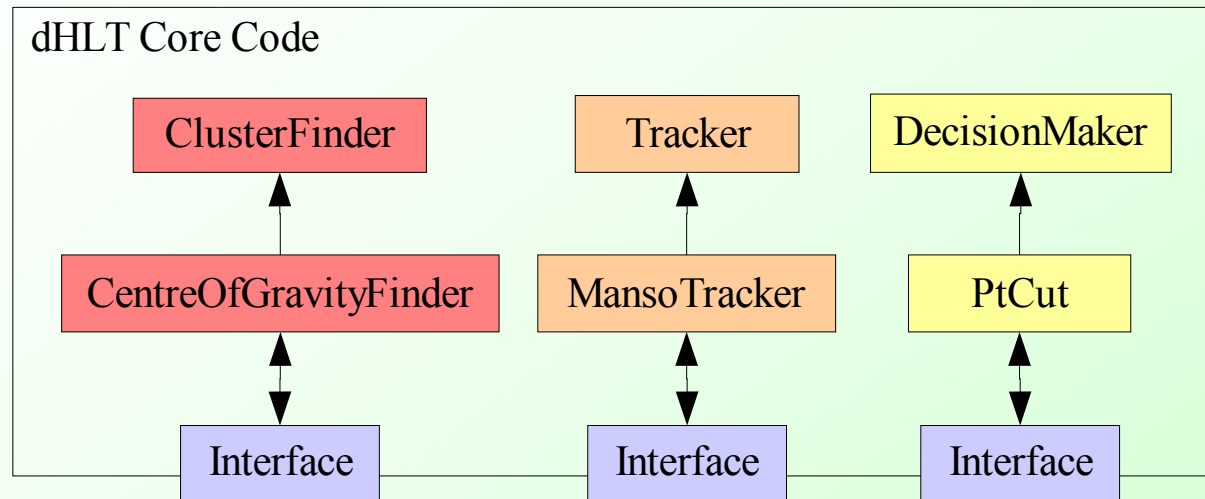
Tracker Algorithm

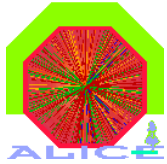
$$p = \frac{q \int B dl}{\theta}$$



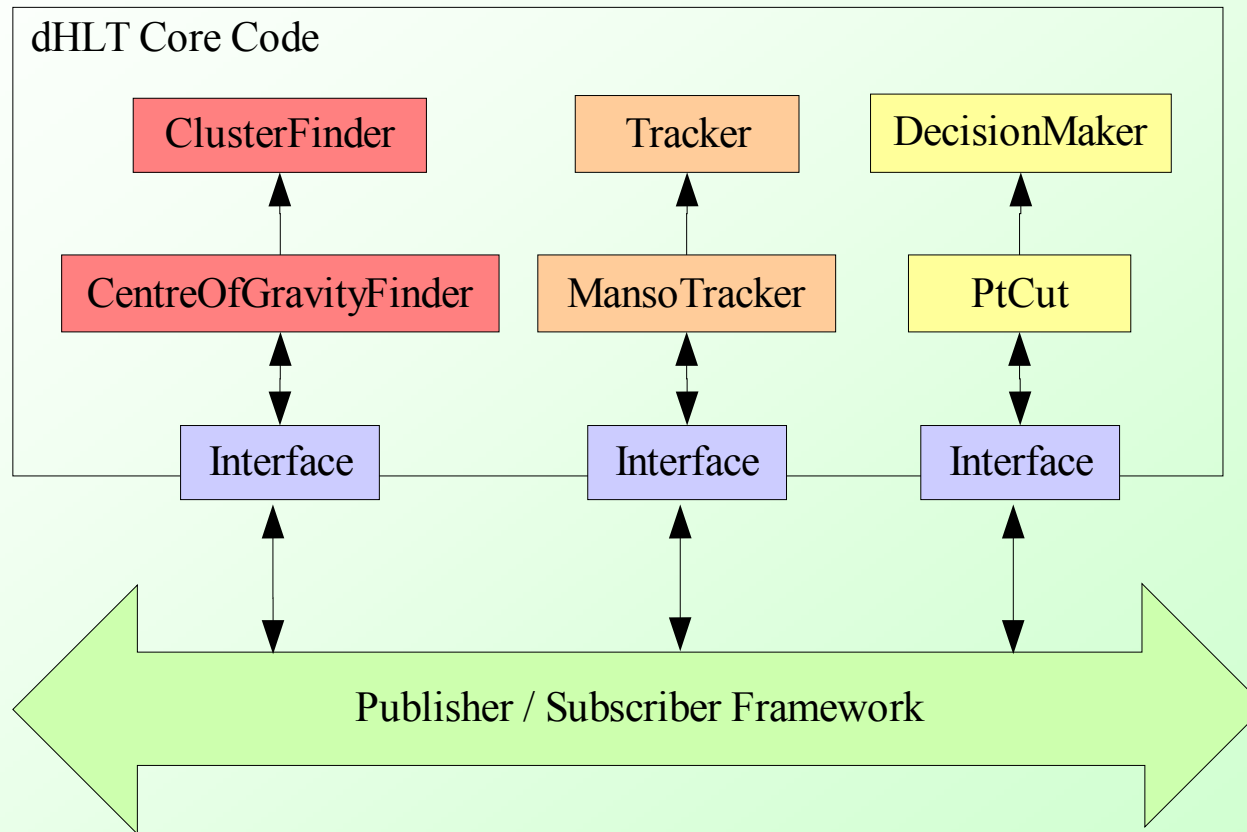


Code Structure



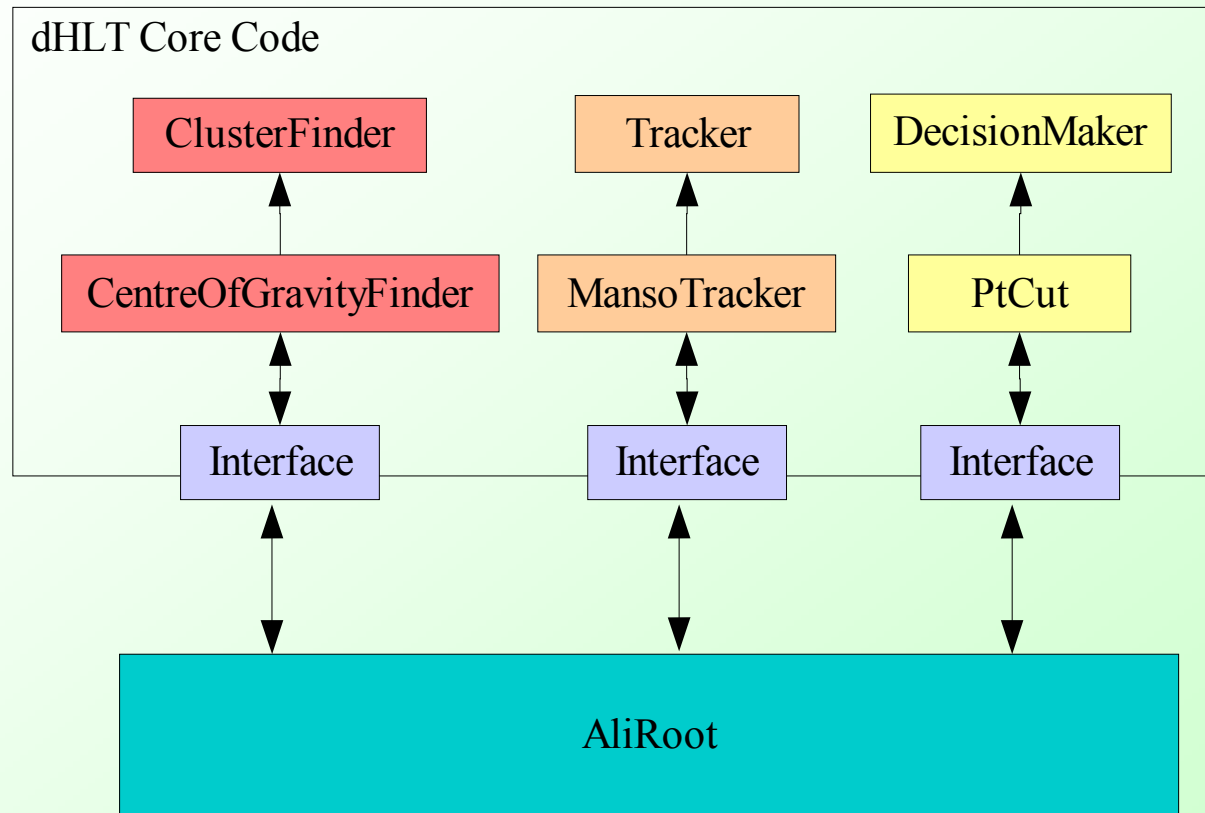


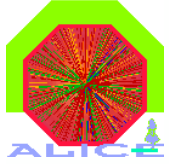
Code Structure (online case)



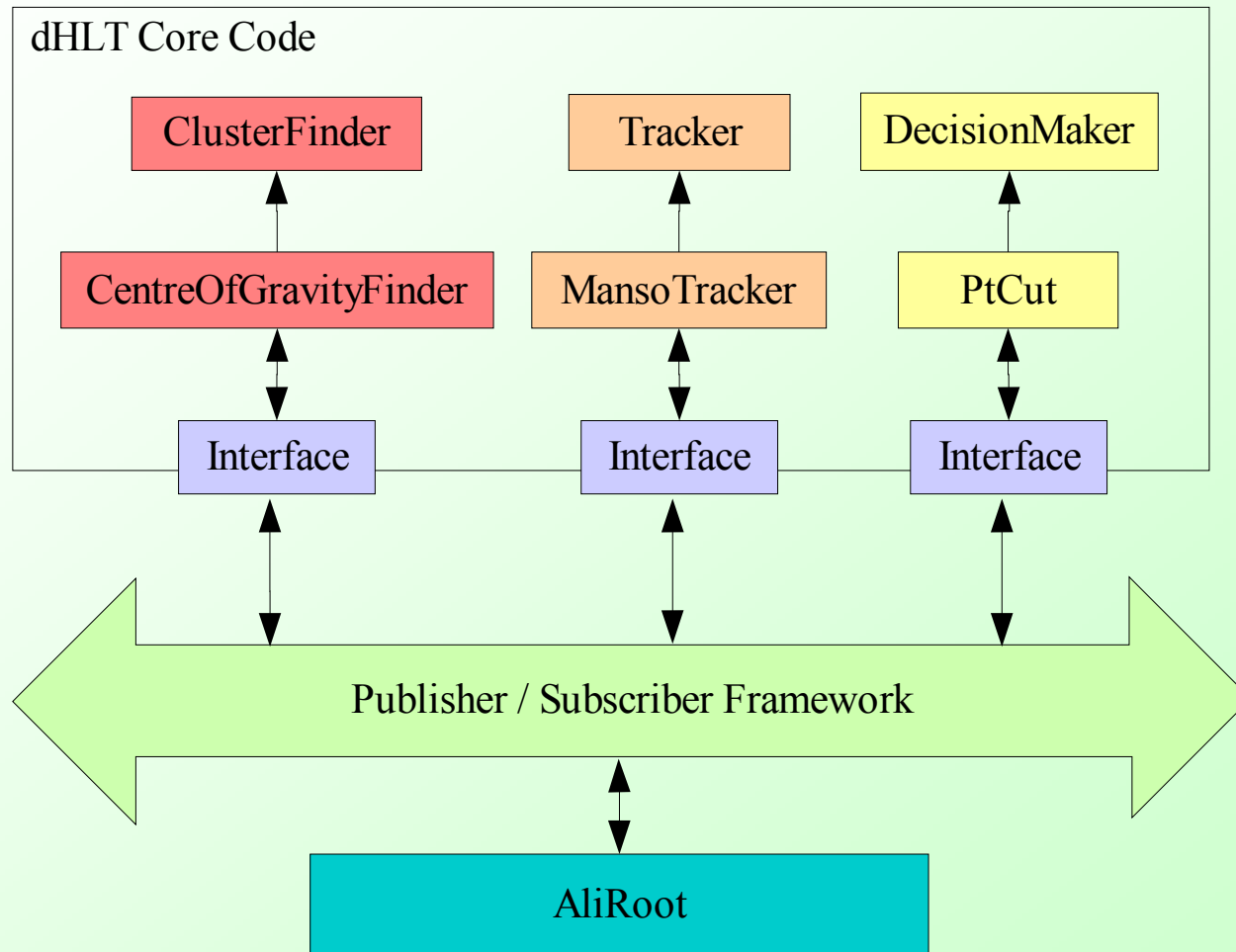


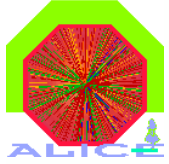
Code Structure (offline case)





Is this useful to anyone else?





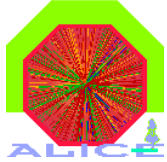
dHLT/AliRoot Integration Status

- MansoTracker integrated into MicrodHLT implementation, all compiled as a Root shared library (.so).
- A few auxiliary classes to fetch AliRoot data and store dHLT output also added.
- Added functionality for MicrodHLT to use interpreted Tracker code for fast prototyping.
- Can run whole dHLT over a galice.root file with about 10 lines of macro code.



Some Comments and Questions

- User interface to HLT in general should be similar to, or integrated with, AliSimulation. But what exactly should it look like?
 - It has to be flexible enough to handle all sorts of HLT configurations, now and in the future.
- Analysis and Debug components for the online HLT system should also be integrated with AliRoot, but how?



Conclusion

- We require clear and very well defined interfaces between AliRoot and HLT.
- There are large parts of the dHLT code which should overlap with the rest of HLT (TPC HLT for example). We need to identify these common modules, write them once, but make them useful to everyone at the same time.
- We have a lot to discuss this week.