SHOE
Software Status

Matteo Franchini
University of Bologna

Outline

* Software structure scheme

* Plan and Status for beam-test' systems (for each)

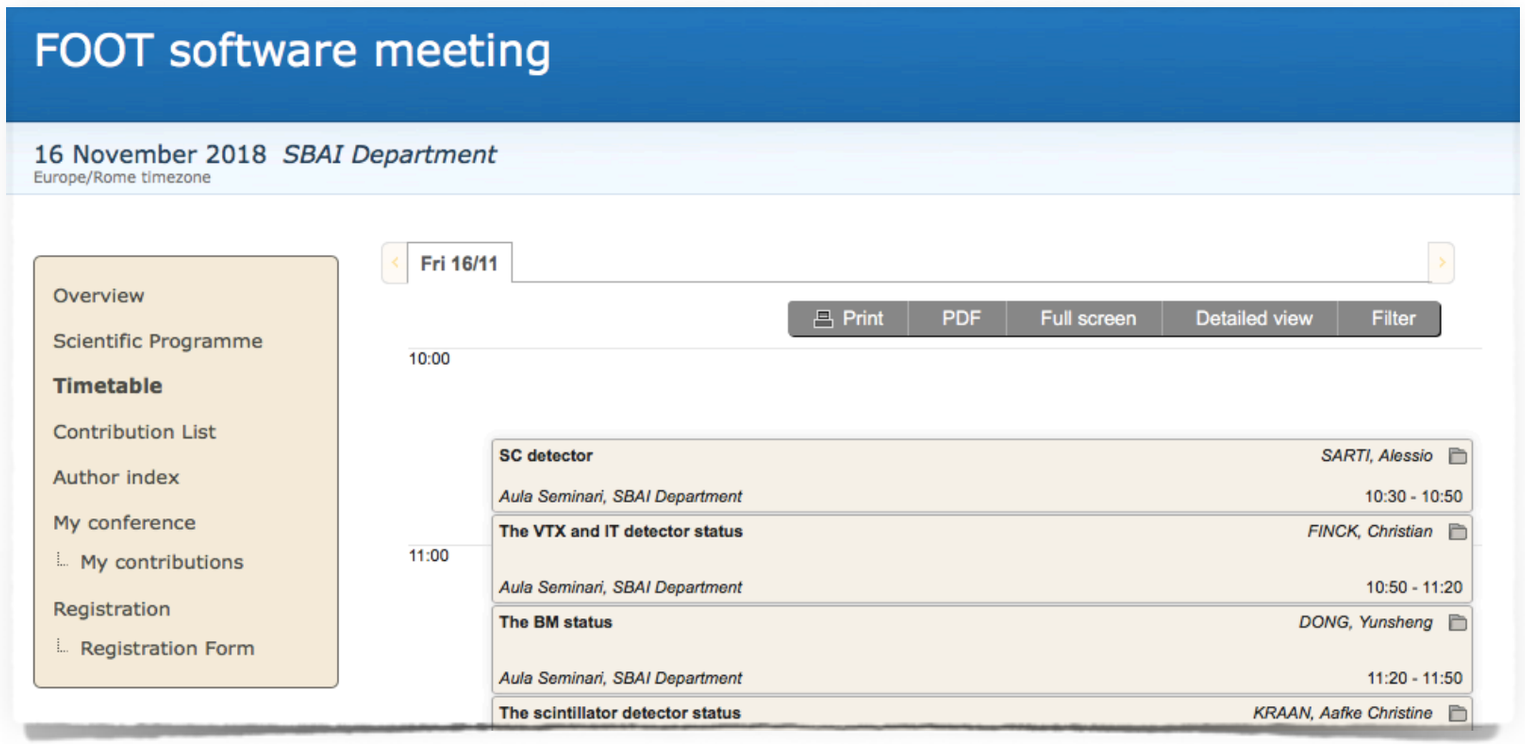
* Beam test roadmap

* Post beam test roadmap and general status

Nothing written on stone, all this is meant to stimulate discussion (especially on roadmap)

Introduction

- Software meeting in Rome
  - https://agenda.infn.it/conferenceTimeTable.py?confId=16920#20181116
  - you can find the minute on the indico page
- Summary of each software part status and needs;
- Organisation of the work



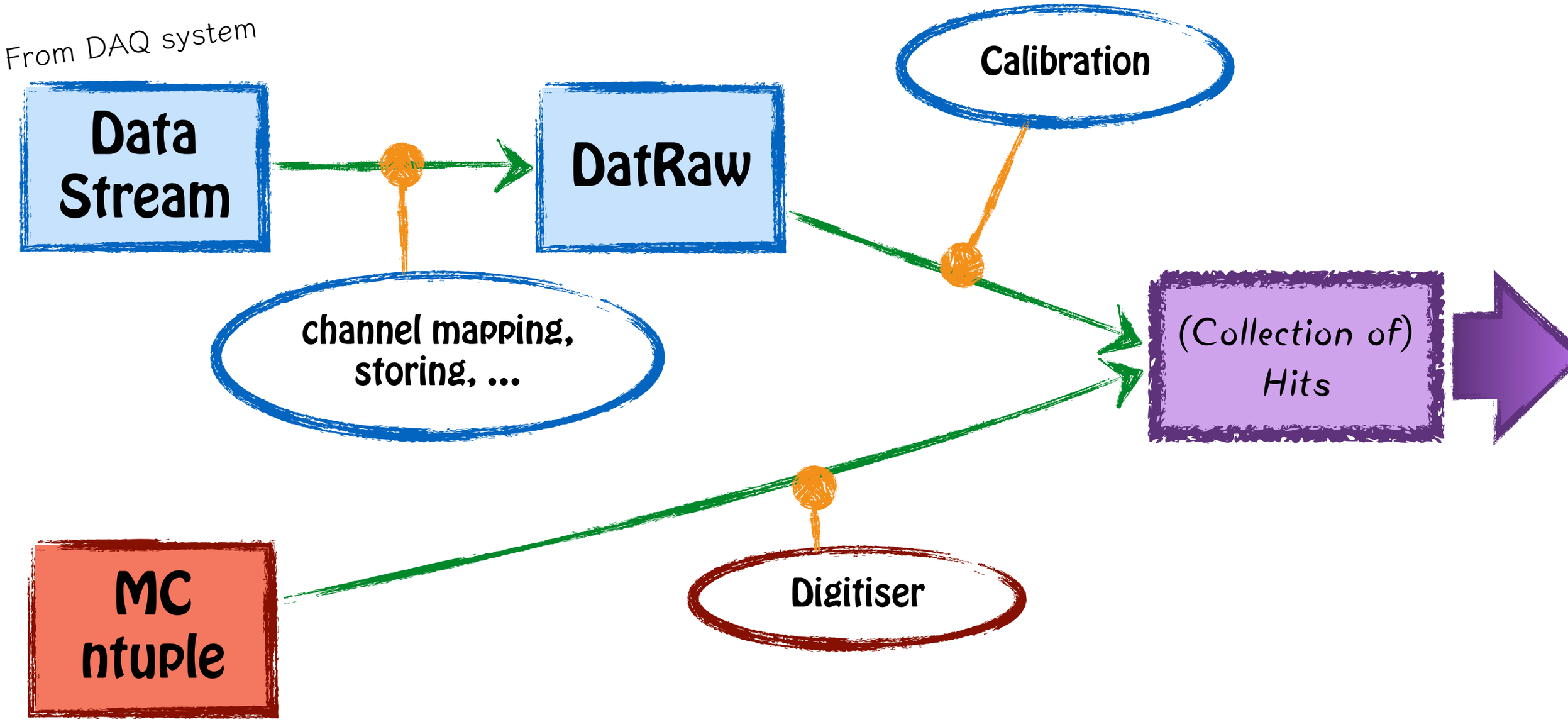
Introduction

* SHOE reconstructs the event (performs global alignment, track reconstruction and matching) and pass the results to the analysis code (to be included)

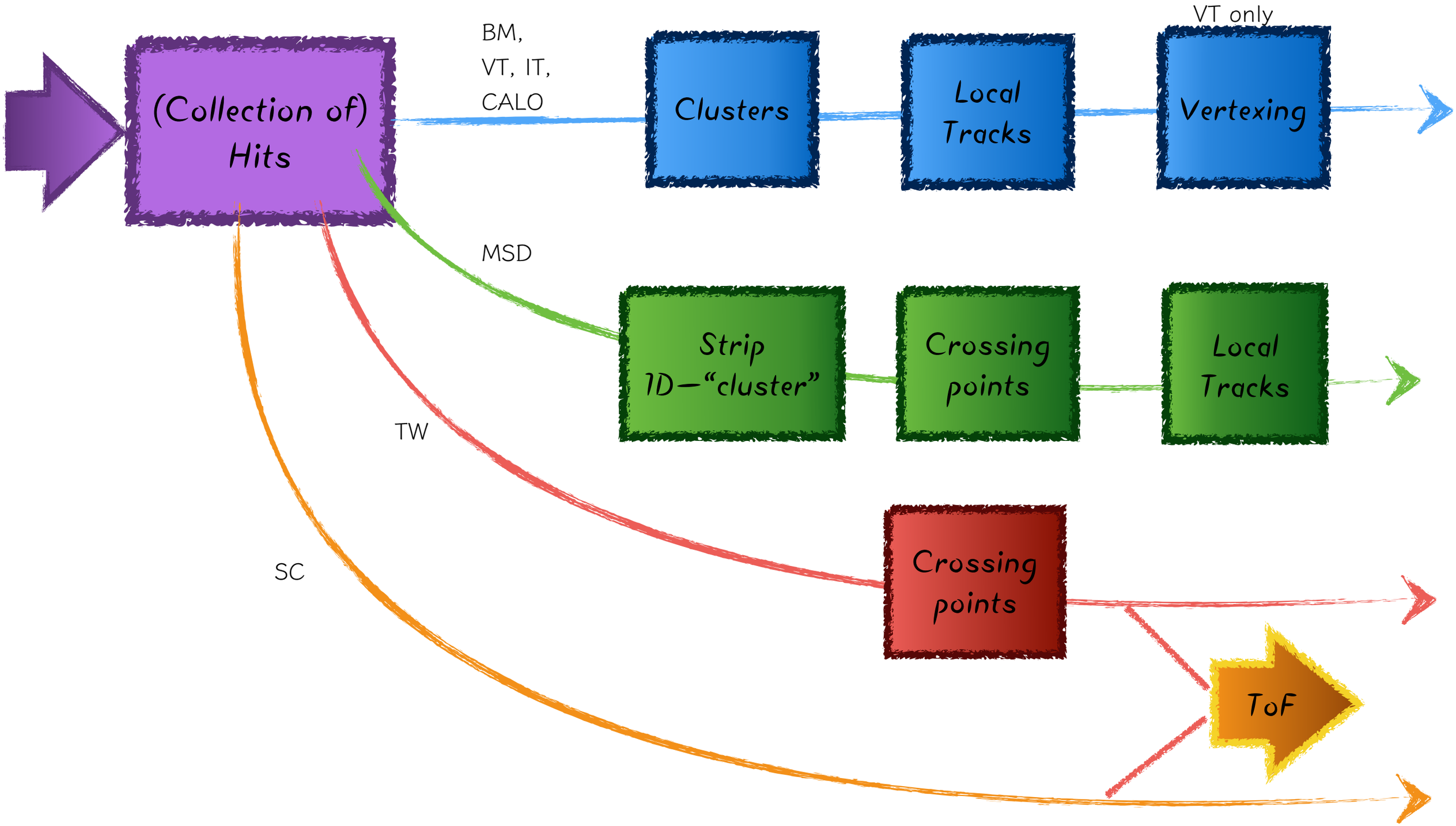
* Next future main goal: having a working reconstruction software for April beam test:
  - focus on beam test configuration (only participating systems);
  - not all "global" functionalities required;

* (not so) Longer term goal: Include full missing functionalities for final FOOT configuration (CDR)

SHOE logic structure



SHOE Logic structure



Beam test software Staus

...for each detector...

SC, BM, VT, TW

Beam test goals

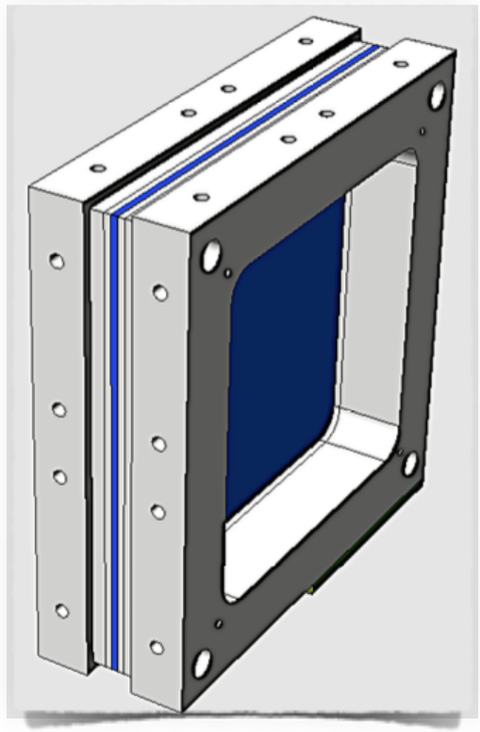
- Study each detector performances:
  - VTX tracking and saturation;
  - Setup a partial but "realistic" FOOT-DAQ system interfaced with SHOE;
  - First serious combined data-taking;
  - SC fragmentation monitor by BM;
  - Each detector ... stuff...

- Global stuff and Physics:
  - Test inter-detector "hit" matching (with Kalman);
  - Time of Flight measurement and resolution;
  - "Differential" cross section as a function of the atomic number (Z) from TW dE/dx.

Start Counter - SC

Contact: Alessio

* Alias "The Margarita" detector will provide:
  - Time reference for ToF calculations
  - ~ Number of ions traversing the detector
  - ToF evaluated in SHOE "talking"with the scintillator



* <u>Status:</u> full working version based on FIRST' skeleton

* <u>ToDo:</u>
  - DAQ interface and channel mapping;
  - Check the geometry and material simulation in FLUKA and ROOT

* All Pretty straightforward implementations!

Beam Monitor - BM

Contact: Yun

* All in Beam Monitor branch: bm_calibration
  - from the master branch of the last software meeting held in Bologna ~ 2/2018, so quite easy to merge with the new versions.
  - merge foreseen in December/ begin-of-January.

* The Beam Monitor does "not" deal with downstream reco, so it can be treated as a separated detector.

* Produce tracks objets stored in ntuple, ready for global works.

Status

* MC chain and track reconstruction working

* Data acquisition chain ready and tested in a test beam @Trento in September

Beam Monitor - BM

Contact: Yun

ToDo

* Current tracking with Genfit. Slow.

* In BM small B field. Try simpler and faster methods:
  - Least square fitting (restored from FIRST and debugged)
  - Legendre Transform

* Improving digitisation (data adding fake hits, cutting real hits etc.)

* Code to be ready for beam test @Trento in December

The code status is at a very good stage

It should not need relevant changes or development for the March beam test

Vertex - VT

Contact: Chris

* Complex software object, many passages;

* Clusterisation: groups together all adjacent pixels.

* Local Tracking: ~ χ^2 minimisation already tested in FIRST
  - finds straight tracks across the 4 VT layers
  - ok because layers are close one each other

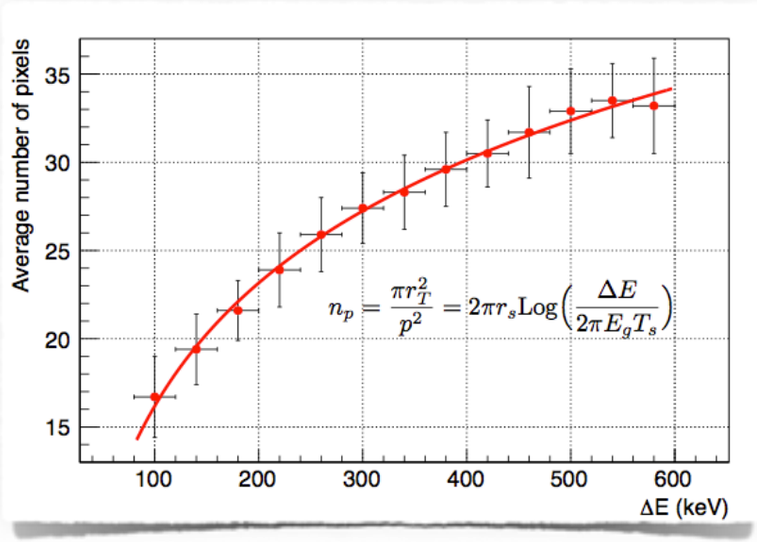
* Vertexing: finds the intersection of the tracks, gives the interaction point.

Vertex - VT

Contact: Chris

Status

- Software structure in place! Chris adapted code from FIRST;
  - works in a separate branch with many changes
- Digitisation
  - Add "cluster" of pixels to the single MC-pixel
  - Add noise signal
  - Add pileup signals



- Produce Clusters, Tracks and make the Vertexing
- Data stream reading and calibration – structure ready

Vertex - VT

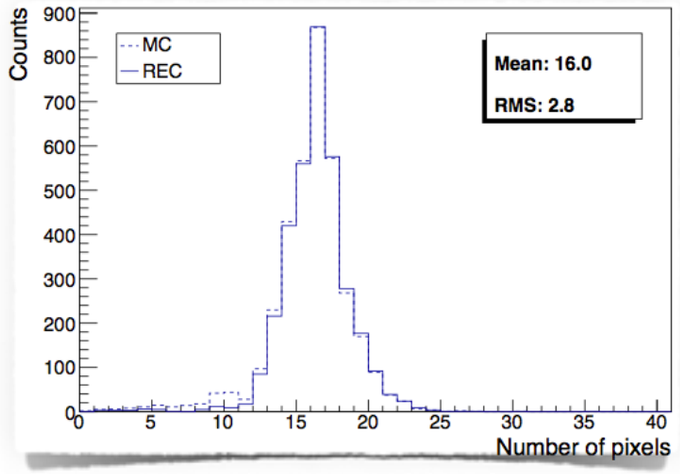
from Chris

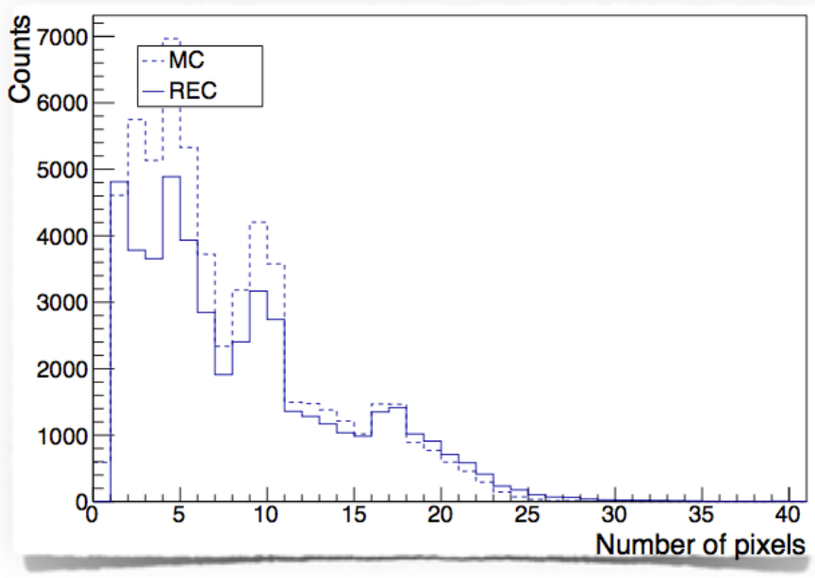
ToDo - Test!

* Very well known software system imported from FIRST.
* Good shape + complex system = test already started!

Some Examples

* C⟹VT @ 400 MeV (no target)
  - comparison between expected fired pixels (from digitisation) and reconstructed ones (clustering) -> good agreement





* If O⟹C2H4 target @ 200 MeV
  - Less agreement between expected and reconstructed number of pixels in the cluster.
  - To understand why (maybe delta ray? to be investigated).
  - Possible to improve agreement?

Scintillator - TW

Contact: Aafke, Niccolo'

- "Simple" software object
  - Geometry: 2 crossed layers of scintillator bars.
  - Given info:
    - fired bar (x or y position),
    - deposited energy,
    - timing.

- Status: read MC hits, identify the fired bar and create the crossing points (with x,y coordinates)

- Tested and inserted in the global tracking

Scintillator - TW

Contact: Aafke, Niccolo'

ToDo

- Data acquisition classes
  - read from DAQ stream;
  - calibrate the charge;

- Digitisation
  - Start with parametric calibration

  - If needed simulate the charge distribution... (see calorimeter)
  - No cross-talk among the bars!

Bigger amount of work among the beam test systems.
Discussion already started.

Versioning

* Many branches, one for each system development. Must be merged in one.
* All quite similar with almost independent changes -> easy!
  - but one with big changes: digitiser_v3

* Current master: last tutorial version (+updates)
  - Works with past simulation versions
  - Updated to V15 simulation, asymmetric magnets (yesterday talk by Sandro);
  - Include MC reading, passing from true to reco objects (not precise digitisation yet)
  - Reco tested on global reconstruction for VT, IT, MSD, TW.
  - CALO still uses true objects.
* Performances in Alberto's talk.

Versioning: Digitizer_v3 branch

- Chris did a lot of work on that including:
  - VT & IT acquisition chain;
  - Adapting all the past FIRST functionalities for VT & IT:
    - digitisation, cluster, vertexing
  - event display functionalities;
  - cmake compiler!!!

for beam test...

Merge needed!

- a lots of work done!!! All in one go -> a pain to merge
- some conflicts with previous strategy, new tests
- step by step merge with beam test priorities

Beam test SHOE roadmap

2nd week December
- Start TW work
- Try BM merge (foreseen to be smooth)

3rd week January
- Working and full SC merge

Mid February
- VTX merged (updated geo, digit, cluster, track)

End of February
- All parts together and working
- Event display?
- minor updates and refinements to be done

November | December | January | February | March

2nd week January
- Working BM merged
- TW new part skeleton

4th week January
- Global tracking + matching (VTX - TW)

3rd week February
- Working and full TW merge
- DAQ reader for all involved systems

MARCH
- Test,
- test and…
- …test

Post-beam test

Post beam test SHOE roadmap

Take a breath after Beam Test

June 2019
- MSD software structure ready
- CALO geometry
- CALO basic software structure ready
- Alignment

Autumn 2019
- Full FOOT setup complete!
- Ready for tests and optimisation

2019

May 2019
- Refined Global tracking (optimise hit preselection, add DAF)
- Working IT part
- Event display

End of summer 2019
- IT complete software
- MSD complete software
- CALO complete software
- Global Tracking “complete”
- Analysis code in SHOE (maybe before…)

Of course, geometry and various algorithms updates should be a continuous effort…

MicroStrip Detector - MSD

Contact: Riccardo

* MC reading and chain in place and tested in global reco
* Strip crossing implemented

ToDo

* Data acquisition chain
* Digitisation (charge drift)
* "1-D clustering"

Inner Tracker - IT

Contact: Chris

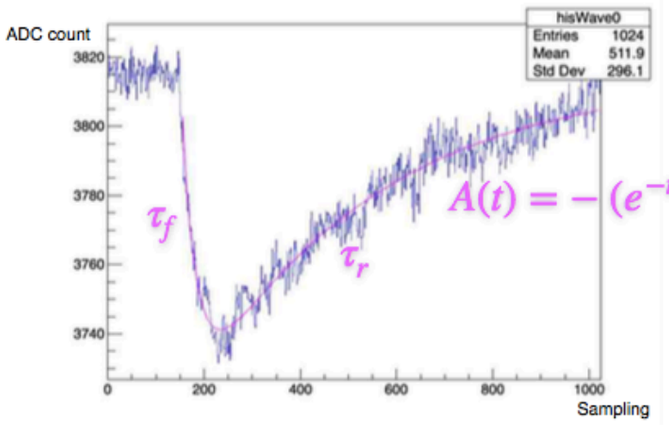
- Basically same strategy as VT, if one is ready also the other will be
- Differences:
  - More complex geometry
  - No local tracking needed

Calorimeter

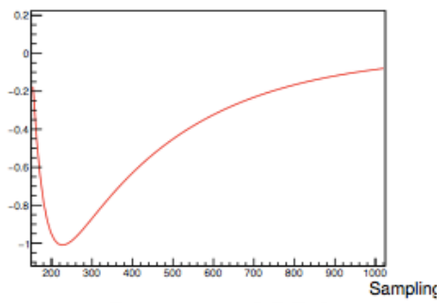
Contact: Lorenzo

- Implementing crystal geometry. Now parallelepiped, moving to trunked pyramids.
- MC hits reading and processing, building up...

1. Use function to simulate the signals in a range between (0,-1)



$-1 < A(t) < 0$



$0 < t < 1024$

from Lorenzo

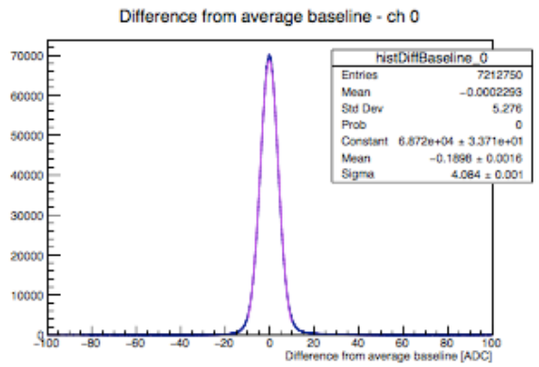
2. Float value from MC: E_{MC} — ε is conversion factor to photo-electrons → $\epsilon \cdot E_{MC}$

3. Simulated pulse shape: $A_{sim}(t) = \epsilon \cdot E_{MC} \cdot A(t)$

4. Normalisation to dynamic range of the digitiser: $S(t) = \frac{A_{sim}(t)}{A_{max}} \cdot 4096$

$A_{max} = 1\ V$

5. Add noise:



6. Convert float into integer values

Conclusion

* Main goal: SHOE ready for beam test in simplified configuration:
  - ST: almost ready, only geometry checks.
  - BM: ready!
  - VT: structure in place, heavy code merging to be done…
  - TW: Geo and MC chain in place. To do data chain and digitisation.
  - DAQ: we roughly know what to expect, waiting for the definitive structure.

* Much work to be done. Reasonable (?) roadmap defined, we have to focus on following it.

* Beam test's not the end, continue other parts development:
  - Mainly involved systems: MSD, CALO, Global Tracking and alignment
  - A working and complete version by end 2019!

* Remarks:
  - focus on a restricted number of items at a time and THEN MERGE in a common version
  - start to think at testing

Backup

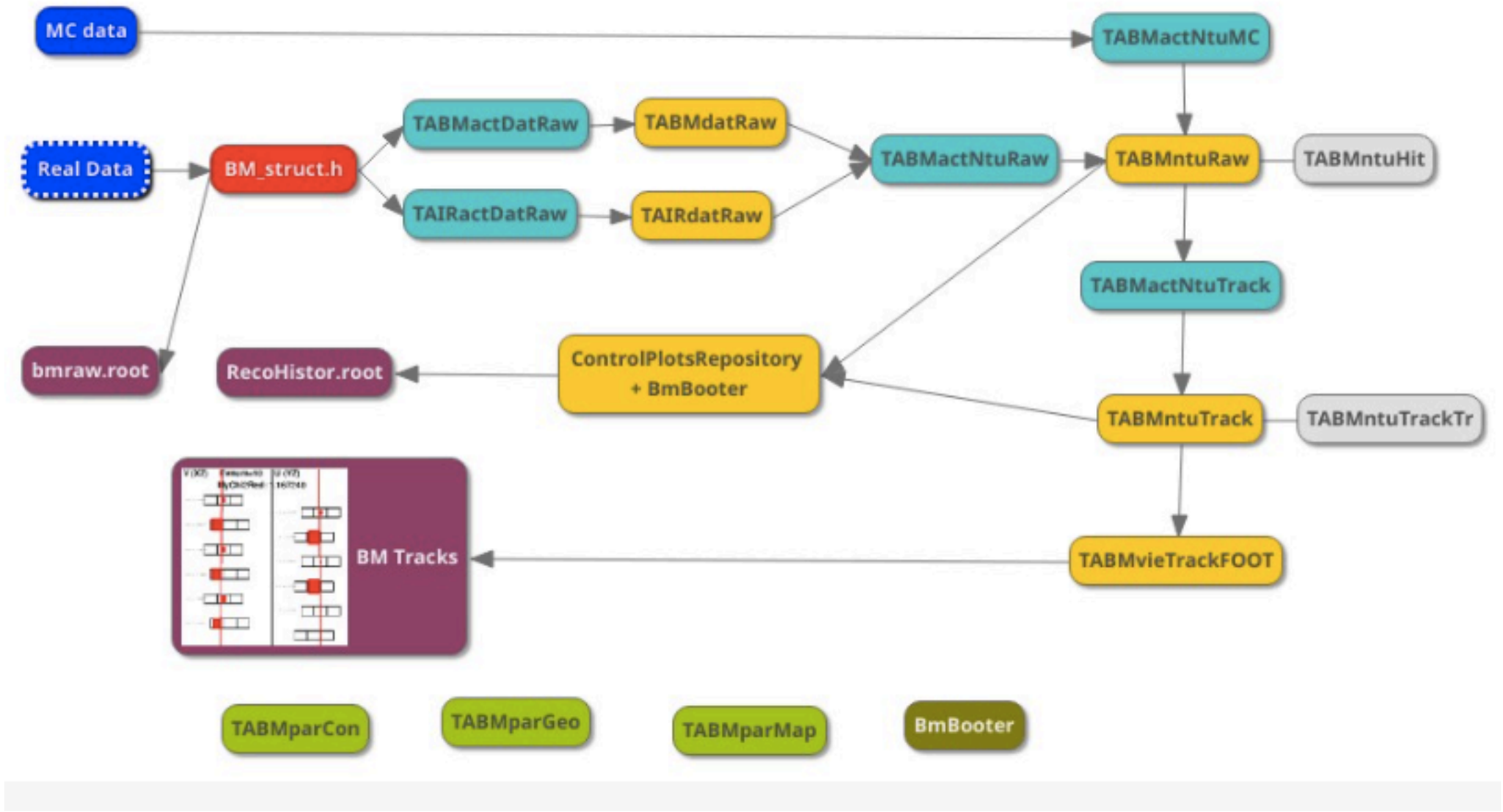
Start Counter

- Pretty straightforward

```
class TAIRrawHit : public TObject {
  public:
    TAIRrawHit();
    TAIRrawHit(int typ, int cha, double charge, double time);
    virtual        ~TAIRrawHit();

    void           SetData(Int_t type, Int_t id, Double_t time, Double_t charge);
    Double_t       Time() const;
    Double_t       Charge() const;
    Int_t          ChID() const;
    Int_t          Type() const;

    void           SetTime(double time);
    void           SetCharge(double charge);
    void           SetChID(int id);  //SC channel ID
    void           SetType(int typ); //meaningless for now.

    ClassDef(TAIRrawHit,1)

  private:
    Double_t ir_time;
    Double_t ir_chg;
    Int_t ir_typ;
    Int_t ir_chid;
};
```

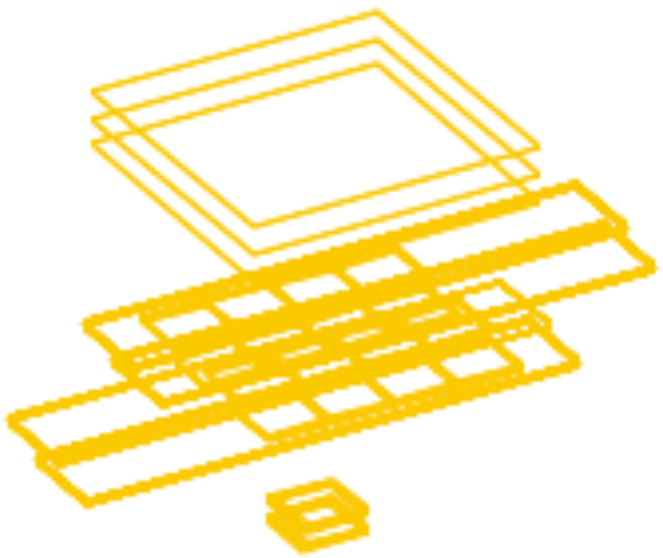
```
TAIRrawHit*       Hit(Int_t i_ind);
const TAIRrawHit* Hit(Int_t i_ind) const;

void           SetTrigTime(double time);
Double_t       TrigTime() const;

Int_t          NTdc() const;
Int_t          NAdc() const;
Int_t          NDrop() const;


public:
  Int_t          nirhit;          //
  TClonesArray*  hir;             // hits

private:

  double         trg_time;        //SC trigger time
  Int_t          fiNAdc;          //
  Int_t          fiNTdc;          //
  Int_t          fiNDrop;         //
```

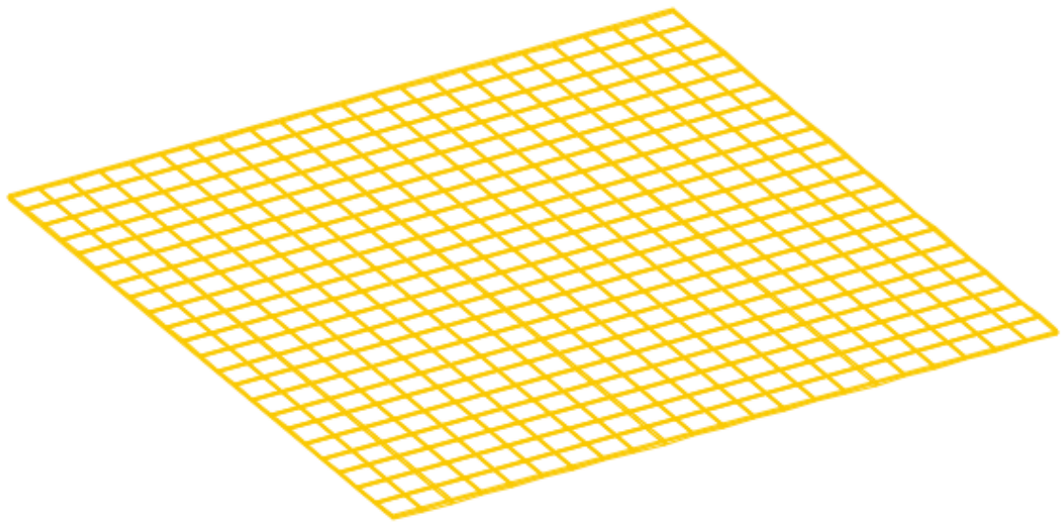
BM

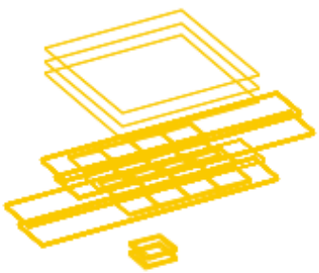

MC data
TABMactNtuMC
Real Data
BM_struct.h
TABMactDatRaw
TABMdatRaw
TABMactNtuRaw
TABMntuRaw
TABMntuHit
TAIRactDatRaw
TAIRdatRaw
TABMactNtuTrack
bmraw.root
RecoHistor.root
ControlPlotsRepository
+ BmBooter
TABMntuTrack
TABMntuTrackTr
BM Tracks
TABMvieTrackFOOT
TABMparCon
TABMparGeo
TABMparMap
BmBooter

Scintillator in tracking

Serena's contribution



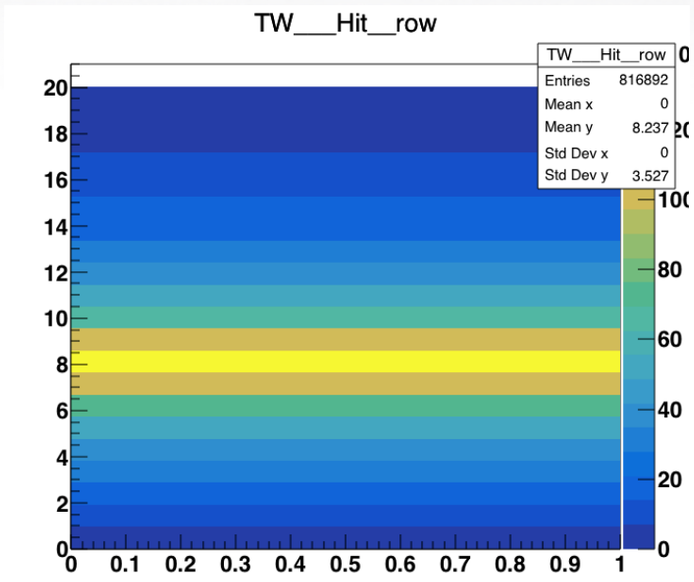


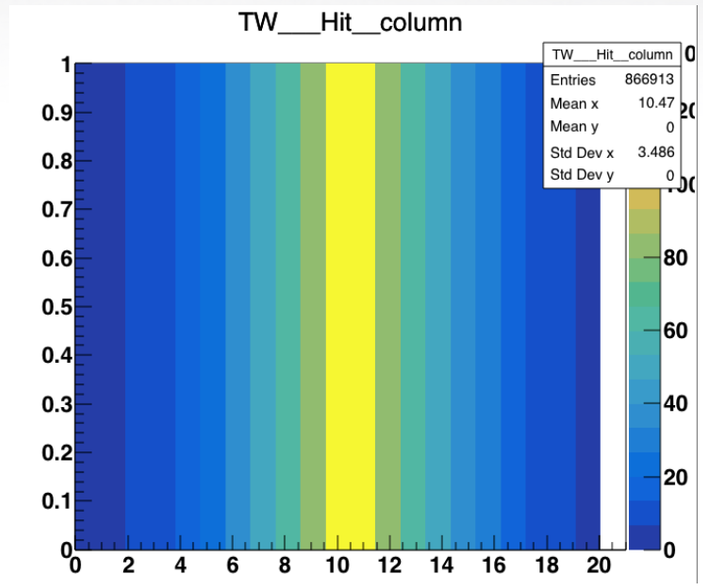


Scintillator in tracking

- **Included** the Scintillator in the Global Reconstruction;

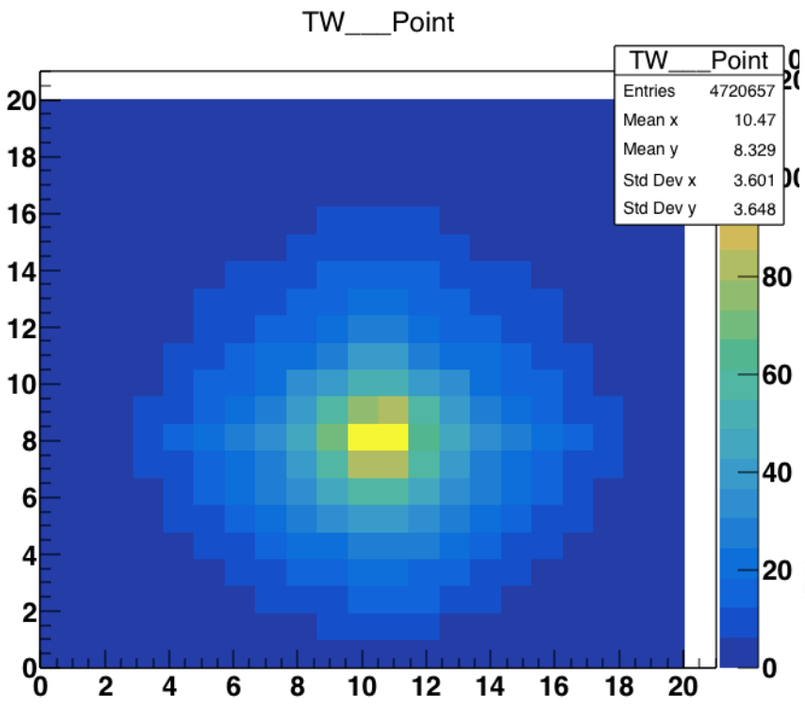
- **Spatial position (x, y)** given by the hit bar, spatial resolution given by
  - **$\sqrt{12}$*lenght = 0,57cm**.



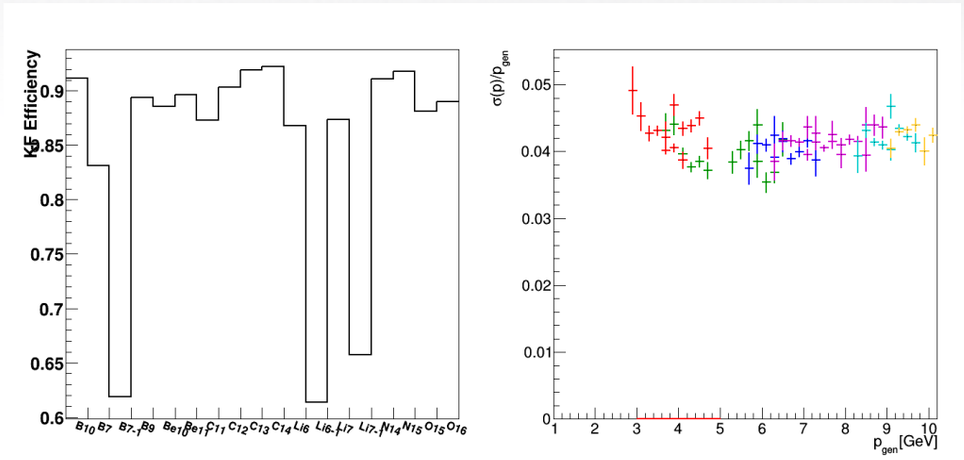
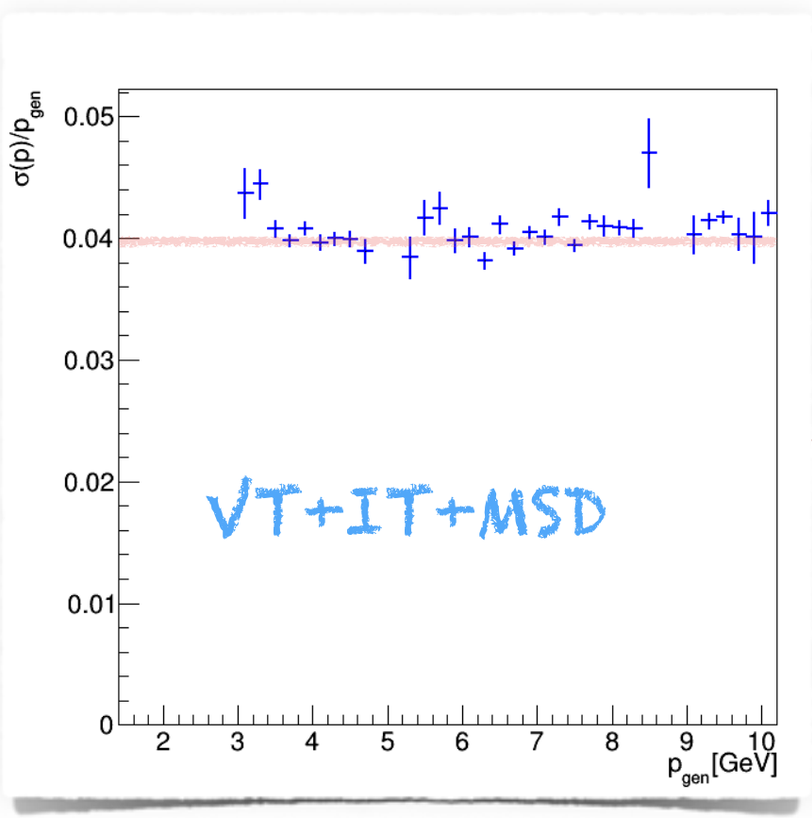
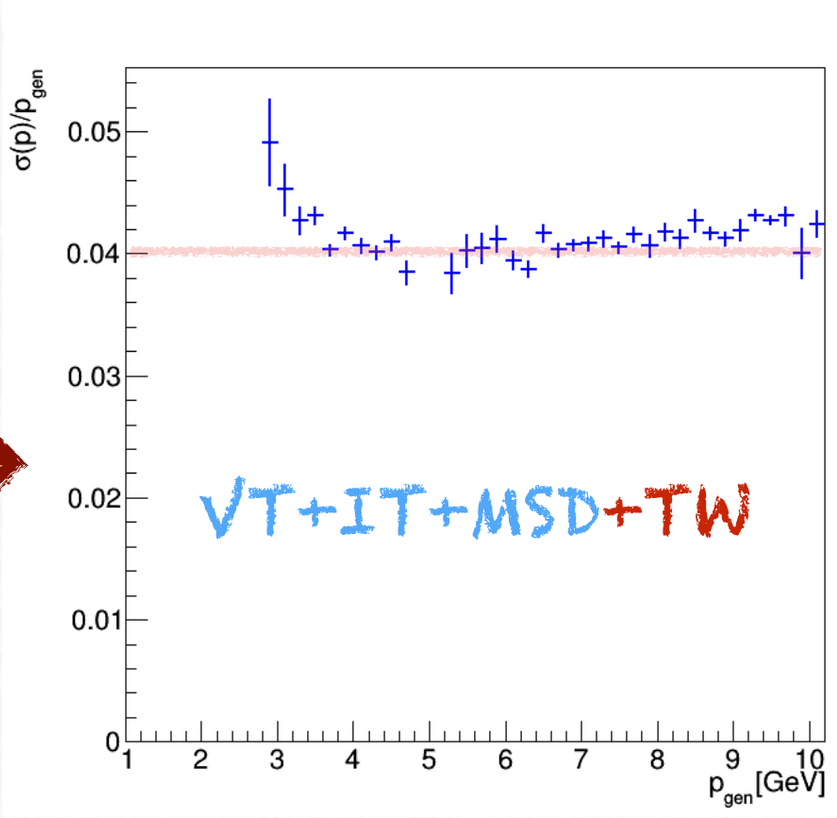


- Founding all the **intersections** between bars with at least one hit;

- Using only **non-ghost** intersections for tracking



No ghosts here!

Scintillator in tracking

- Longer lever-arm, worse resolution;
- **Very preliminary result:** cannot improve momentum resolution much…
- … **but** probably useful for **track founding** in the tracking process