Proposed TrackerHit Datamodel changes

```
317  317
318  318      edm4eic::TrackerHit:
319  319        Description: "Tracker hit (reconstructed from Raw)"
320       -    Author: "W. Armstrong, S. Joosten"
     320  +    Author: "W. Armstrong, S. Joosten, S. Gardner"
321  321        Members:
322  322          - uint64_t          cellID           // The detector specific (geometrical) cell id.
323       -      - edm4hep::Vector3f position         // Hit (cell) position and time [mm, ns]
324       -      - edm4eic::CovDiag3f positionError   // Covariance Matrix
325       -      - float             time             // Hit time
326       -      - float             timeError        // Error on the time
     323  +      - edm4hep::Vector3f position         // Hit (cell) position [mm]
     324  +      - edm4hep::Vector2f loc              // 2D location on surface [mm]
     325  +      - edm4eic::Cov2f    locError         // Covariance on loc
     326  +      - float             time             // Hit time [ns]
     327  +      - float             timeError        // Error on the time [ns]
327  328          - float             edep             // Energy deposit in this hit [GeV]
328  329          - float             edepError        // Error on the energy deposit [GeV]
     330  +    OneToManyRelations:
     331  +      - edm4eic::RawTrackerHit rawHits      // Raw signals clustered into hit point
329  332
```

Simon Gardner – Simon.Gardner@Glasgow.ac.uk

Review of early track reconstruction

Current Dataflow

Simulation output – *edm4hep::SimTrackerHit*

Digitization – *edm4eic::RawTrackerHit*
Rounds signal value in cell to * 1e6 -> [keV]
Rounds time stamp to * 1e3 -> [ps]
Sums all rounded signals in cell and earliest timestamp

Position Reconstruction – *edm4eic::TrackerHit*
Global 3D position of cell hit (internally uses 2D)
Local 2D covariance of cell hit
Converts back to [ns] and [GeV]

Source Linking
Converts back to 2D and looks up surface
(This information should already be readily available)

Seeding
Converts *edm4eic::TrackerHit* to *eicrecon::SpacePoint*

Proposed Dataflow

Simulation output – *edm4hep::SimTrackerHit*

Signal Sharing – *edm4hep::SimTrackerHit*
Spread out hits DDDigi?
Developers would like customers.

Digitization – *edm4eic::RawTrackerSignal*
Multitude of digitization tasks/steps (e.g. noise). DDDigi again?
Limit signal cell summing by integration time.
(High occupancy long event windows)

Clustering – *edm4eic::TrackerHit (Measurement?)*
Groups together signals to form single hit/measurement
Local 2D position and covariance, time, energy…
Convert raw signals into common units (Calibration)

Source Linking
Should already be linked coming out from digitization to mirror experimental data.

Seeding
Add intermediate step and datatype to do:
2D position, covariance + context -> 3D position, covariance

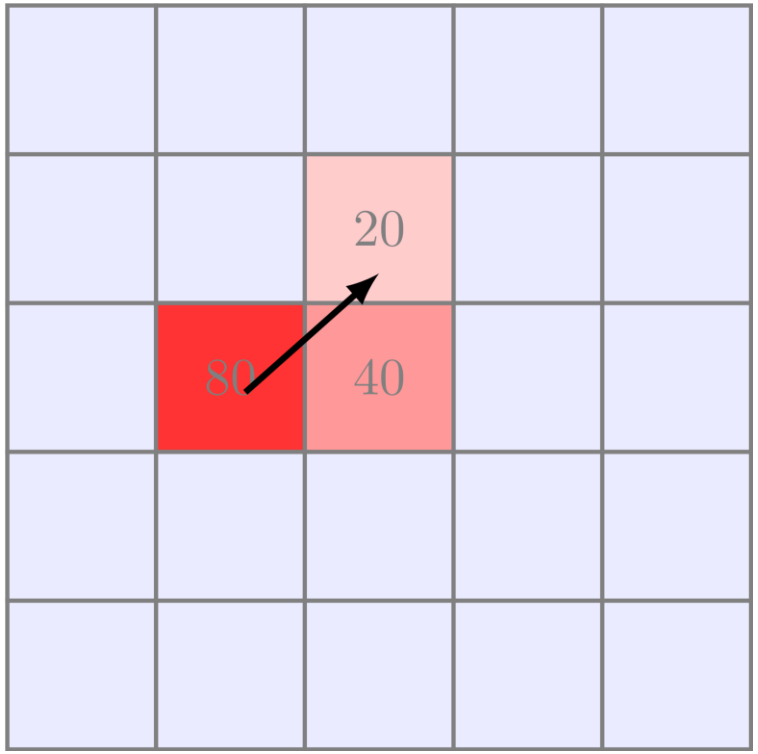
Disclaimer – Currently I have no knowledge of the interface with ACTS or its workings

Datamodel transparency

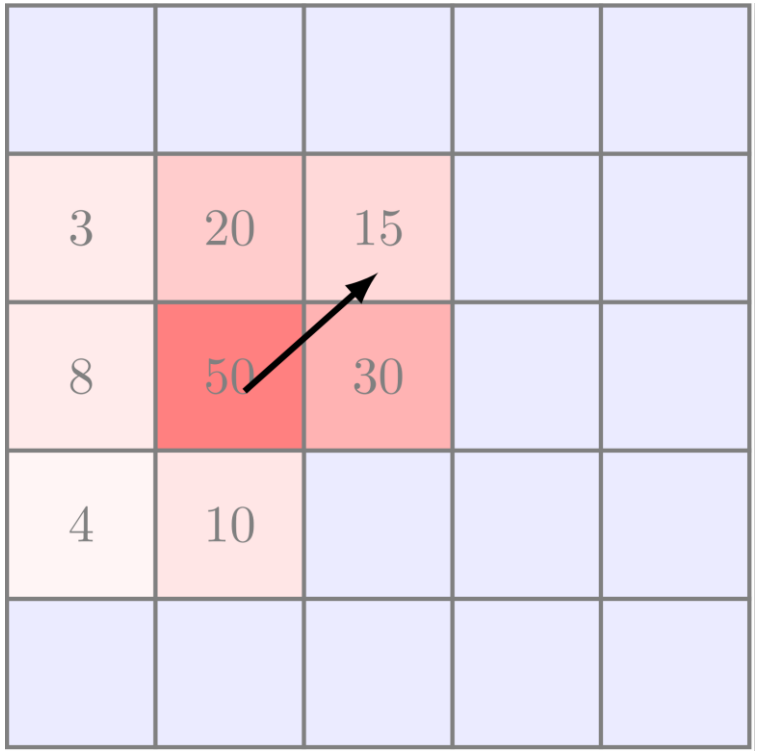
Proposal to introduce graphics to EDM4eic, illustrating the design use case for each structure.

Would make model more transparent along with a clear flow diagram.

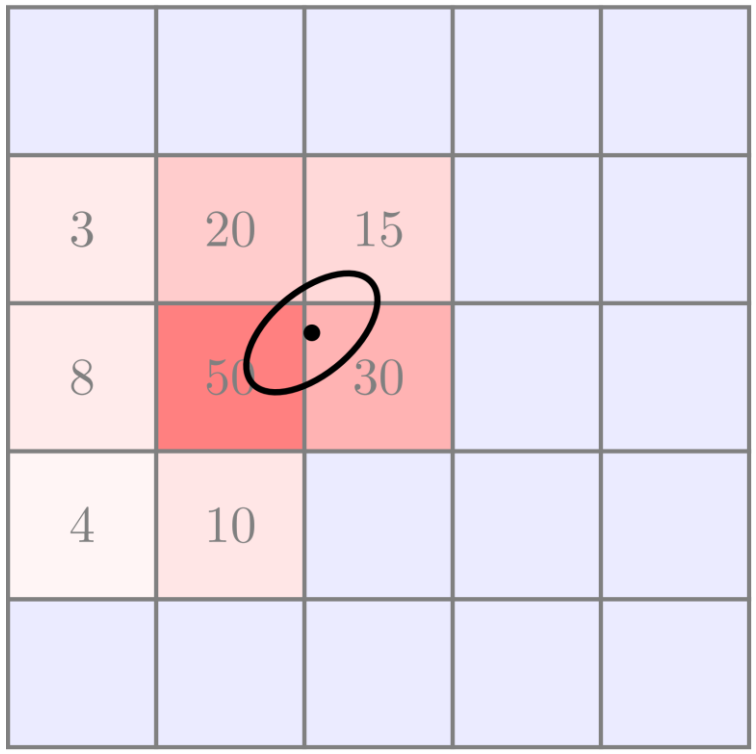
(Loosely similar to Acts “tracking in a nutshell” diagrams)



SimTrackerHit (Raw)



SimTrackerHit (Shared)



TrackerHit