

Cylindrical Segmentation Tracking WG Meeting

Modified version of MPGD Meeting of 07/15

Yann Bedfer

CEA/DPhN Saclay

31 October 2024

CyMBaL: Structure

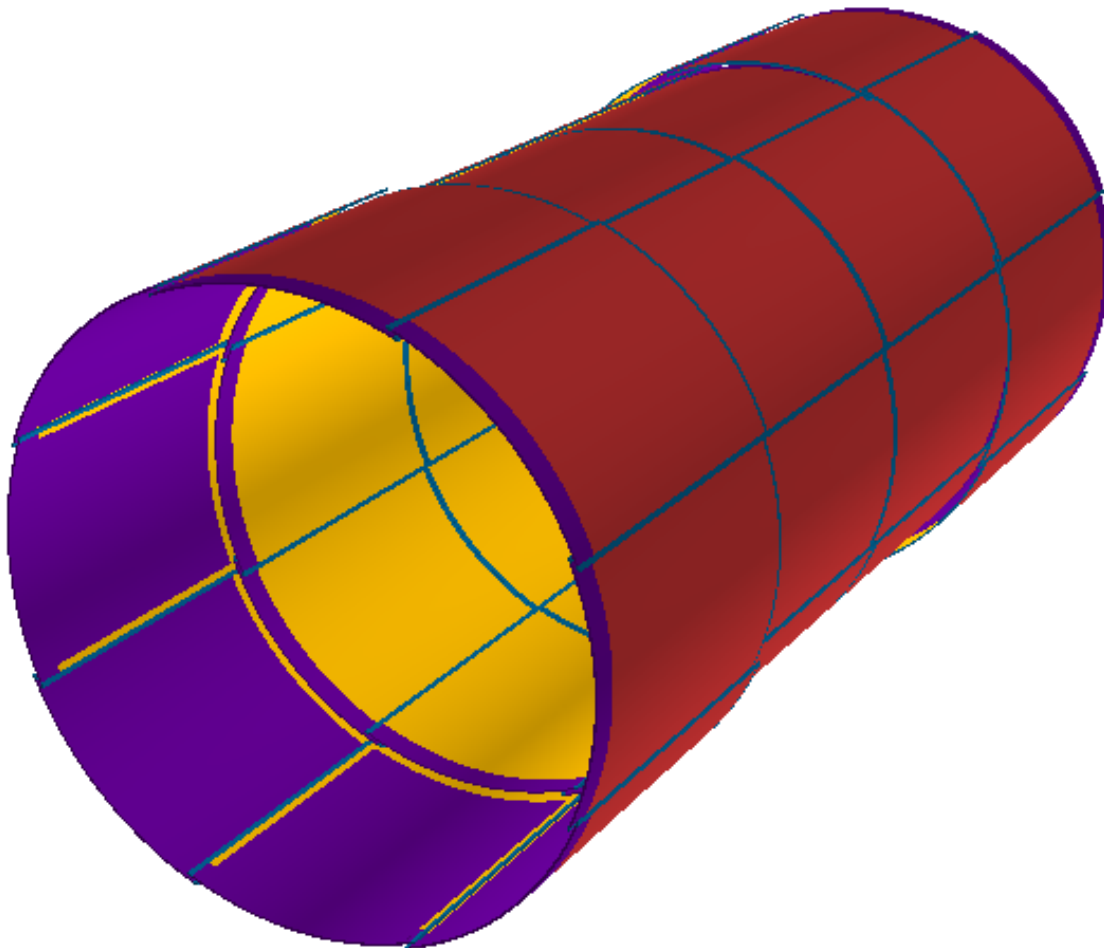
- 4 sectors along $Z = \text{Backward/Forward} \times \text{Inner/Outer}$

Backward/Forward: gap. Inner/Outer: Superposition.

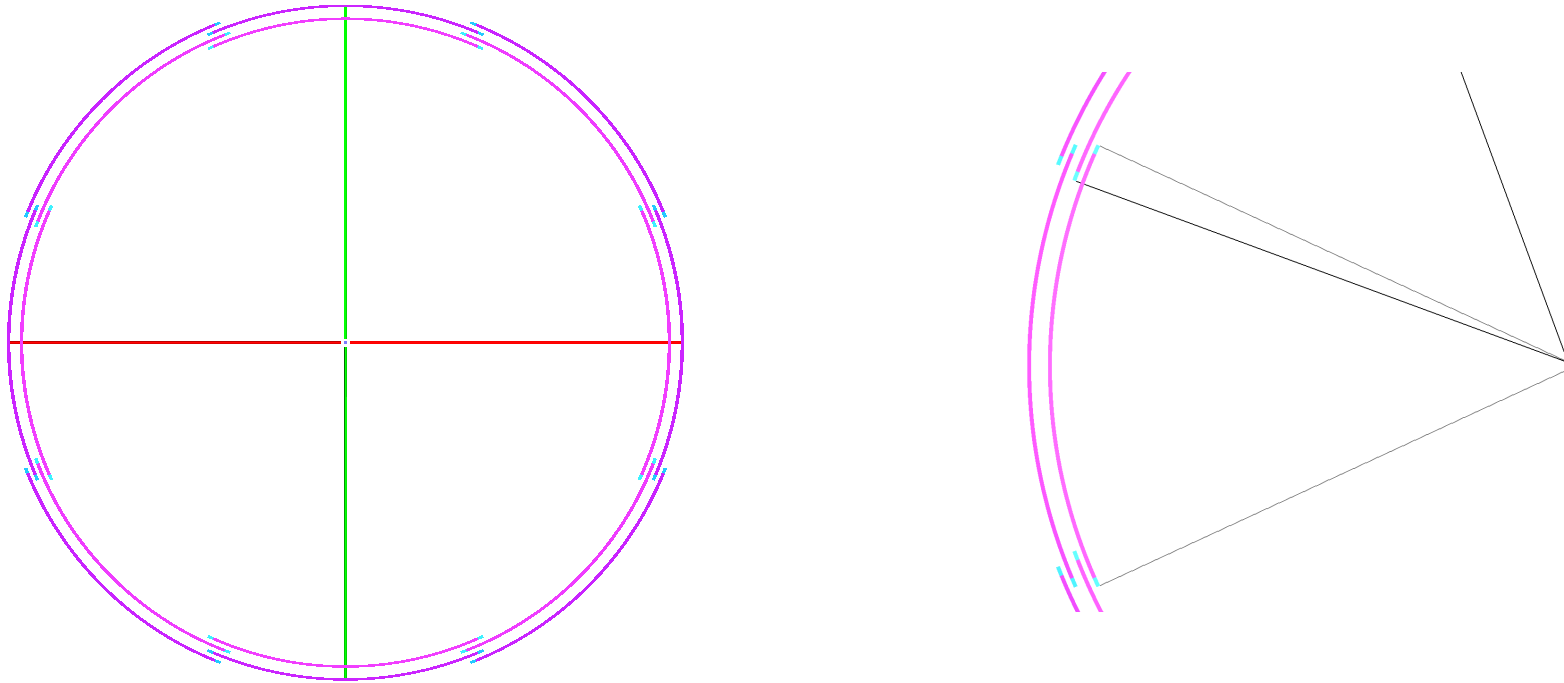
- Along φ , 8 tiles(*staves*) *per* sector: Superpositions.
- **Only one sensitive surface** *per* tile, while **two sets of readout strips**.
- **Only one readout tag in all.**

CyMBaL: Display

- `dd_web_display --export $DETECTOR_PATH/epic_craterlake_tracking_only.xml`
→ `detector_geometry.root`
- `ROOT: TGeoManager::Import("detector_geometry.root");`
`((TGeoVolume*)gGeoManager->GetVolume("InnerMPGDBarrel"))->Draw("ogl");`



CyMBaL: Superpositions: Avoid overlaps



- Along Z : **2 distinct radii**: Smaller for Inner / Larger for Outer.
- Along φ : Same radius. **Offsets**, alternatively > 0 or < 0 .

Encoding/Processing

- XML: `mpgd_barrel.xml` in `epic/compact/tracking`.
 - Geometry `<detector ... />`
 - Readout `<readout ... />`
 - Specifies **Segmentation** and **CellID** encoding.
 - **One to one** `detector` \leftrightarrow `readout`.
 - As of now: one CyMBaL `detector` and hence one `readout`
- **detector** processed by `MPGDCylinderBarrelTracker_geo.cpp` (in `epic/src`)
 - **ROOT/Geant4Geometry**
 - Implements the **offsets** and the **two radii**.
- **Simulation**: `ddsim`
 - Runs Geant4, in above-defined **Geant4Geometry**.
 - Gets (X, Y, Z) of `SimTrackerHit` @ sensitive surface.
 - DD4hep: Parsing `readout`. (X, Y, Z) (\rightarrow *Segmentation class* \rightarrow) **CellID**.
 - Assigns **CellID** to `SimTrackerHit`.
- **Reconstruction**: `eicrecon`
 - DD4hep: `SimTrackerHit`'s **CellID** (\rightarrow *Segmentation class* \rightarrow) (X', Y', Z') of `TrackerHit`,
with present digitization.
 - Runs ACTS

History of Segmentation Specification

- **Readout** in XML: `mpgd_barrel.xml`

```
<detector id="TrackerBarrel_2_ID" ...
      readout="MPGDBarrelHits" ...
</detector>
```

```
<readout> name="MPGDBarrelHits" ...
```

```
<segmentation type="CartesianGridXY"
      grid_size_x="0.150*mm*sqrt(12)" grid_size_y="0.150*mm*sqrt(12)" />
<id>system:8,layer:4,module:12,sensor:2,x:32:-14,y:-18</id>
```

then *to follow curved sensitive surface in detector's volume:*

```
<segmentation type="CartesianeGridXYZ" grid_size_x="0.175*mm*sqrt(12)"
      grid_size_y="0.175*mm*sqrt(12)" grid_size_z="0.175*mm*sqrt(12)" />
<id>system:8,layer:4,module:12,sensor:2,x:32:-11,y:-10,z:-11</id>
```

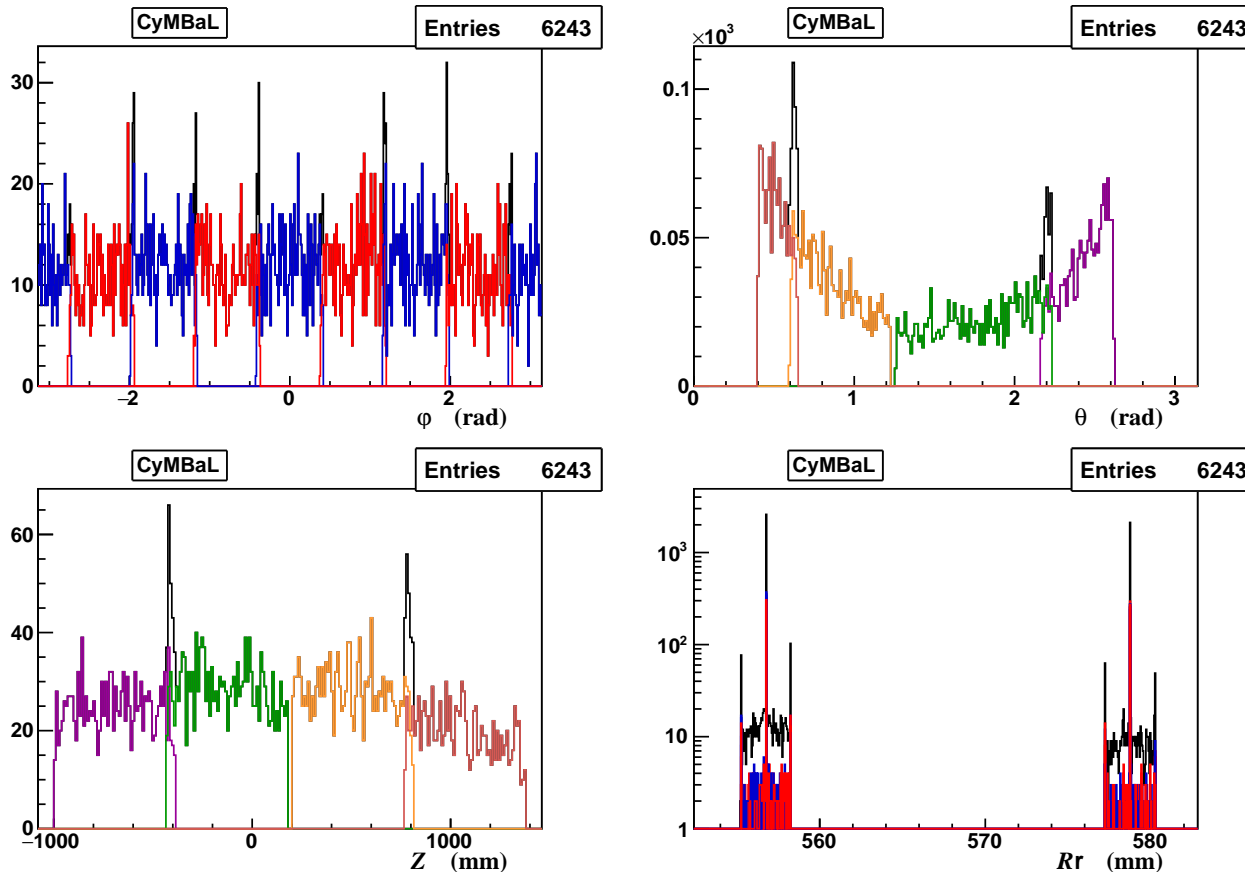
then *to get pixels instead of voxels:*

```
<segmentation type="CylindricalGridPhiZ" "radius"="..."
      grid_size_phi="1*mrad" grid_size_z="0.150*mm*sqrt(12)" />
<id>system:8,layer:4,module:12,sensor:2,phi:32:-16,z:-16</id>
```

```
</readout>
```

Simulation

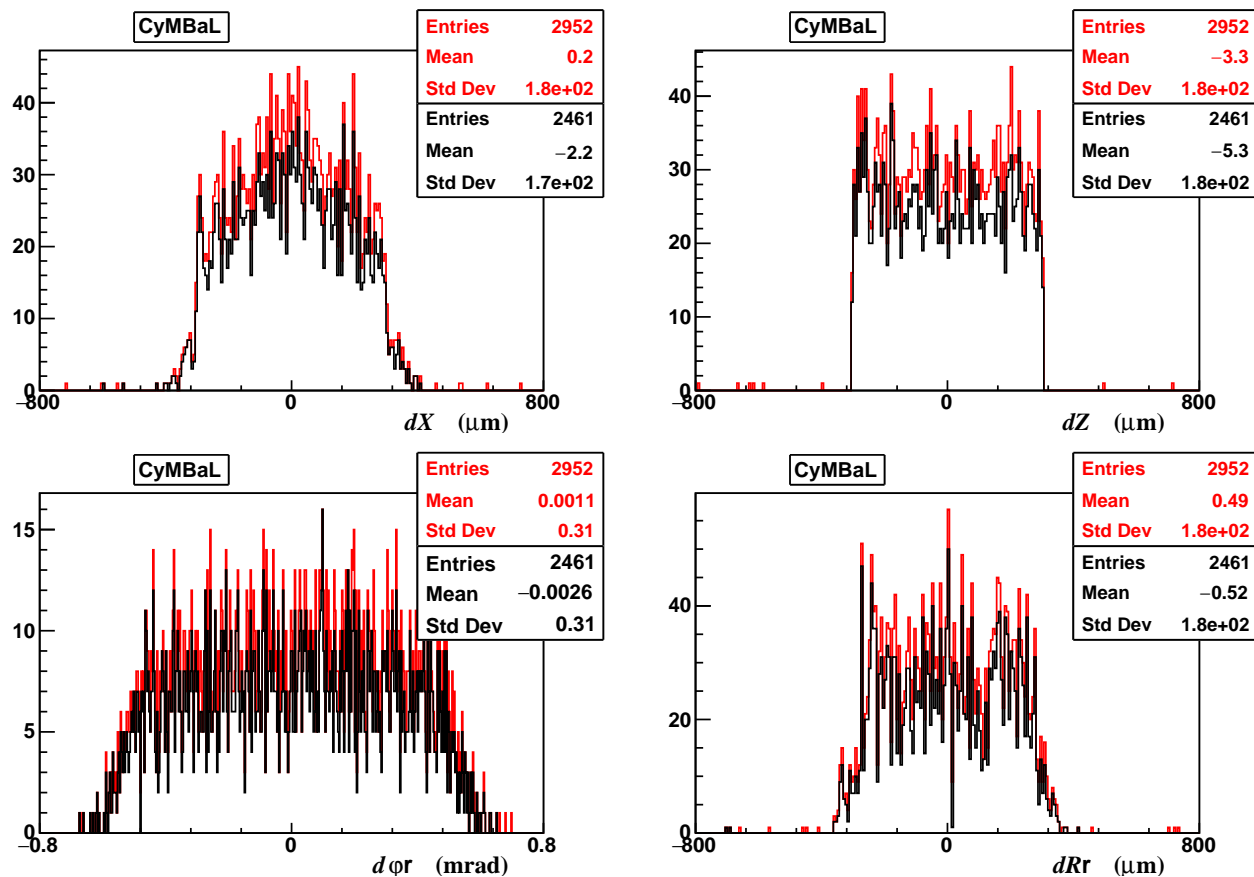
- ddsim: SIM.gun μ^- $15 < \theta < 165$ deg. $P=10$ GeV



- 8 peaks in ϕ corresponding to *superpositions*.
- 2 peaks (*superpositions*) and a dip (*central gap*) in θ and Z .
- Unexpected stray hits in Rr (*reduced radius*).

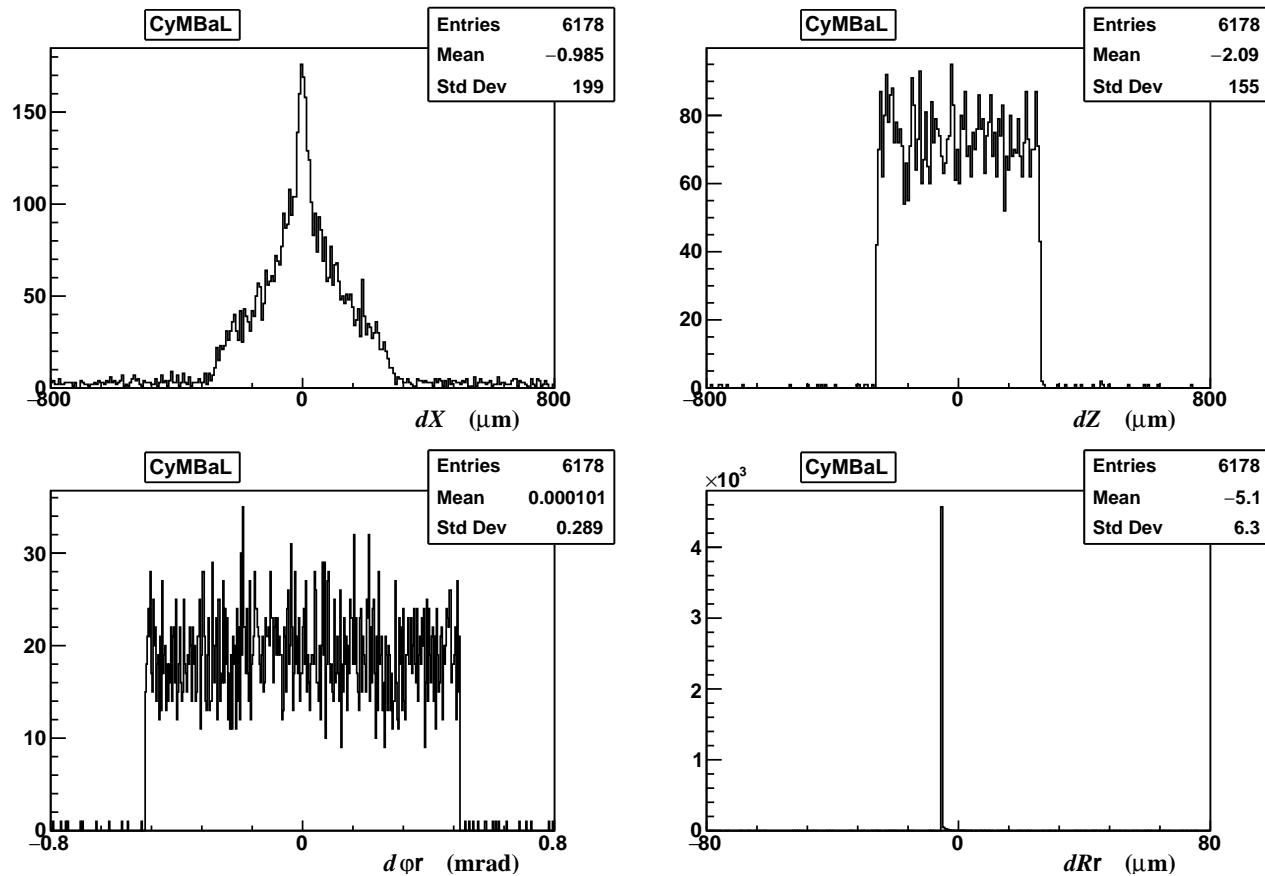
Residuals w/ CartesianGridXYZ of June

- Residuals = $d\mathbf{X}d\mathbf{Z} \dots = \text{Reconstructed} - \text{Simulated} = \text{TrackerHit} - \text{SimTrackerHit}$
- In podio TTree: `MPGDBarrelRecHits - MGDBarrelHits`,
associated *via* `MPGDBarrelHitAssociations_(raw|sim)Hit`
- **Awaited:** $dZ = \pm 175 \mu\text{m}$: fulfilled, $dRr = \text{Dirac}$, not fulfilled



Residuals w/ CylindricalGridPhiZ

- Expected: $dZ = \pm 150 \mu m$: fulfilled, $dRr = \text{Dirac}$, fulfilled, $d\phi r = \pm 1/\sqrt{12}$ mrad, fulfilled



- Unexpected: Stray hits, Inefficiency $\sim 1\%$ (#Entries: 6243 \rightarrow 6178).

Segmentation Class

- **4 member methods** (*basically*):

- Constructor: reads the **segmentation** string
- `cellID(position)`: **Cartesian** position \rightarrow **CellID** (*used by simulation*).
- `position(CellID)`: **CellID** \rightarrow **Cartesian** position in sensitive surface frame (*used by reco*).

Problem in `CylindricalGridPhiZ` (**as opposed to** `CartesianGridXY`):

Need 3 coords, while **CellID** provides 2.

Solution: specify **radius** in **segmentation** string

Then `CyMBaL` requires **2 distinct radii**, while only one **detector**, **readout**, *etc.* . .

Solution: MultiSegmentation:

```
<segmentation type="MultiSegmentation" key="module" >
  <segmentation name="Backward" type="CylindricalGridPhiZ" key_min="0" key_max="7" . . .
  <segmentation name="Inner" type="CylindricalGridPhiZ" key_min="8" key_max="23" . . .
  <segmentation name="Forward" type="CylindricalGridPhiZ" key_min="24" key_max="31" . . .
</segmentation>
```

N.B: **radius** has to coincide w/ that of **sensitive surface**

*Note: There may be a way to not specify the **radius** explicitly.*

- `cellDimensions(cellID)`: provides for Hit uncertainty (*needs **radius** as well*).

- *Note: I put hit @ the radius, not the most precise.*

Strip Segmentation

- Final goal is Strip Segmentation
 - True also for μ RWELL OuterBarrel and EndCap.
 - **Four** segmentations needed:

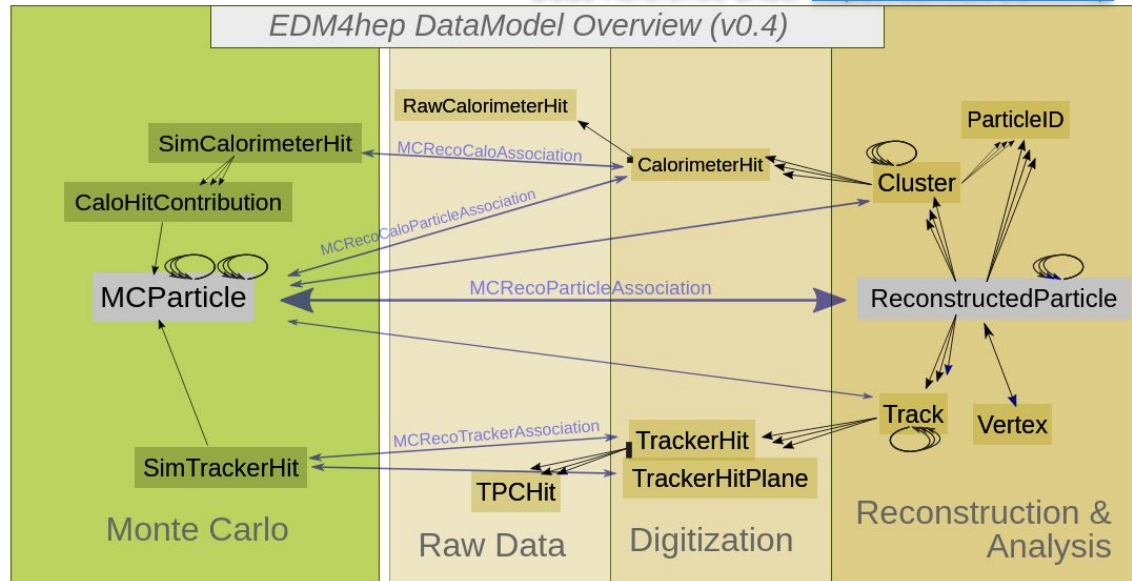
```

<segmentation type="MultiSegmentation" key="sensor" >
  <segmentation name="InnerPhi" type="CylindricalGridPhiZ" key_value="0" radius="RI"
    grid_size_phi="1*mrad" grid_size_z="MMModuleLength"
  <segmentation name="InnerZ" type="CylindricalGridPhiZ" key_value="1" radius="RI"
    grid_size_phi="MMModuleWidth" grid_size_z="0.150*mm*sqrt(12)"
  <segmentation name="OuterPhi" type="CylindricalGridPhiZ" key_value="2" radius="RO" ...
  <segmentation name="OuterZ" type="CylindricalGridPhiZ" key_value="4" radius="RO" ...

```

EventDataModel4hep

Code Reference under <https://cern.ch/edm4hep>



- ~~One to one `SimTrackerHit` ↔ `TrackerHit`~~
 ⇒ “Digitization” needs to be already at the level of `Geant4Tracker::Hit`
- Digitization in EICrecon: new `MPGDTrackerDigi` class based on `SiliconTrackerDigi`.
 Need segmentation again: assign distinct `cellID` to `TrackerHit` along φ/Z (or $X/Y \dots$)
 How to access the segmentation classes? Expert advice welcome.
 - Random draw cluster size according to beam test distribution.
 - Random draw amplitude for given size according to measured correlation.
 - Distribute amplitude along strips to implement measured resolution.
- Clusterization.

Backup

Problem I: CyMBaL SimTrackerHit

- At the level of SimTrackerHit, suspicious stray hits along R .
- Persist when *w.r.t* centre of curvature ($\rightarrow Rr$), where expected is a Dirac peak.
- Stray hits often of bad quality (SimTrackerHit::quality), see **All hits** *vs.* **good quality hits**.
- Stray hits uniformly distributed along φ

