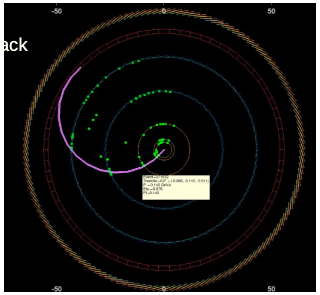
From Geant4 to Reconstruction



From Geant4:

Simulated Hits
Location
Energy deposition
Time

CalorimeterHitDigi
Summed energy → ADC
Time and cellId from most energetic hit

SiliconTrackerDigi
Mostly 1-1
Uses energy deposition
Rare multiple hits in a cell are summed

PhotoMultiplierHitDigi
…

Re-translation ADC → physical value

***RecHits**
Vertex, Tracker, Barrel, Endcap, …

***RecHits**
HCal, Ecal, Barrel, Endcap, …

PID Hypothesis

Track Reco → **Jet Finding** → **Clustering**

Particle Flow

…

HitDigi == HitRaw, the terms are used interchangeably

We use **pedestal and σ XOR nσ threshold**

Code lives in
`eicrecon/src/algorithms/digi/`

Code lives in
`eicrecon/src/detectors/*/`

From here on, there is (should be) no differentiation between data and simulation

Pertinent Code in SiliconTrackerDigi.cc

```
for (const auto& sim_hit : sim_hits) {
    // time smearing
    double time_smearing = m_gauss();
    double result_time = sim_hit.getTime() + time_smearing;
    auto hit_time_stamp = (std::int32_t) (result_time * 1e3);

    if (cell_hit_map.count(sim_hit.getCellID()) == 0) {
        // This cell doesn't have hits
        cell_hit_map[sim_hit.getCellID()] = {
            sim_hit.getCellID(),
            (std::int32_t) std::llround(sim_hit.getEDep() * 1e6),
            hit_time_stamp  // ns->ps
        };
    } else {
        // There is previous values in the cell
        auto& hit = cell_hit_map[sim_hit.getCellID()];

        // keep earliest time for hit
        auto time_stamp = hit.getTimeStamp();
        hit.setTimeStamp(std::min(hit_time_stamp, hit.getTimeStamp()));

        // sum deposited energy
        auto charge = hit.getCharge();
        hit.setCharge(charge + (std::int32_t) std::llround(sim_hit.getEDep() * 1e6));
    }
}
```

Observations

0. CalorimeterHitDigi.cc differes in details, not fundamentals
1. Both integrate over the entire slice time and assign just one time value
   → problematic for length ~2 μs, untenable for length ~5 ms
2. Time resolutions exist, currently only used for
   - smearing in silicon
   - ADC resolution in calorimeters (similar in spirit)
3. TrackerHitReconstruction.cc:

```
rec_hits->create(
    raw_hit.getCellID(), // Raw DD4hep cell ID
    edm4hep::Vector3f{static_cast<float>(pos.x() / mm), static_cast<float>(pos.y() / mm),
    edm4eic::CovDiag3f{get_variance(dim[0] / mm), get_variance(dim[1] / mm), // variance
    std::size(dim) > 2 ? get_variance(dim[2] / mm) : 0.},
        static_cast<float>((double)(raw_hit.getTimeStamp()) / 1000.0), // ns
    m_cfg.timeResolution,                                  // in ns
    static_cast<float>(raw_hit.getCharge() / 1.0e6),   // Collected energy (GeV)
    0.0F);                                               // Error on the energy
```

→ Raw Hits do have timing resolution information (not as covariance)
(Calorimeter Raw Hits do have a timeResolution field, currently unfilled as far as I can tell)

Questions/Tasks

0. What actually *is* the integration time? Every time I think I know, somebody tells me I don't.
1. We need to start a new time bucket after the integration time (inside a slice).
    1. What triggers such a new bucket?
    2. What time should it have (the exact time at the beginning? In the middle? The beginning == end of previous bucket?
2. Apart from the far backward region, do we need to take beam crossing times, and offsets during to z-position, into account?
3. How does the time resolution included in the raw tracker hits currently enter into ACTS? How should it?
    1. Any reason to use a 4-covariance in raw hits already?

Future, not now: A full daq length slice almost must be pre-processed into candidate events, especially if seeding cannot use time information.