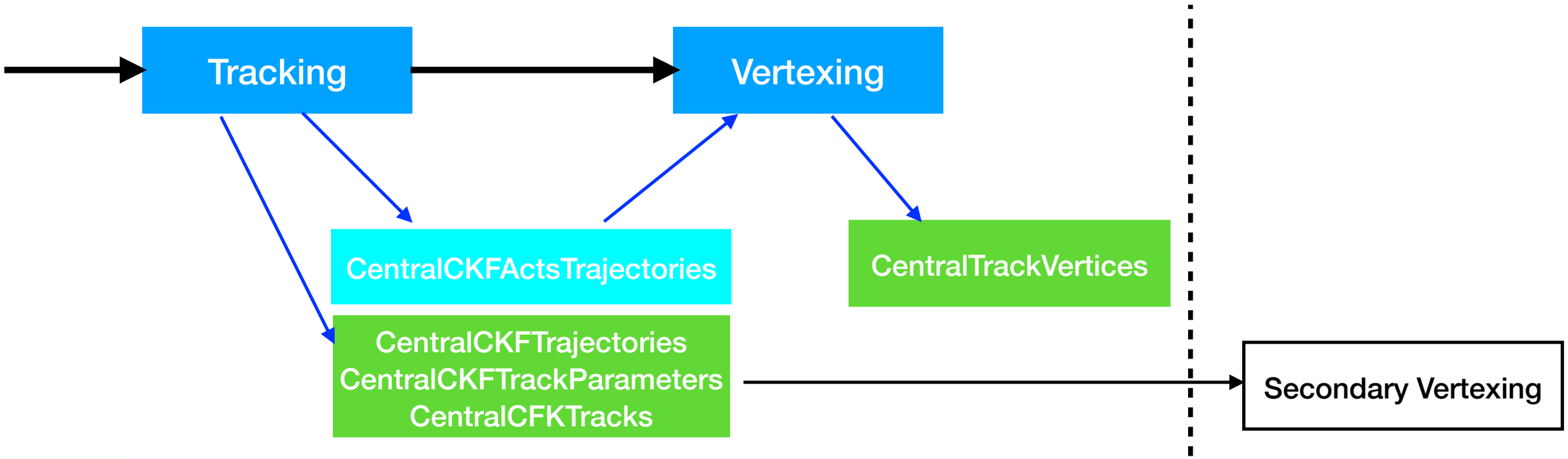
Update on Vertexing Activities

Xin Dong (LBNL)

Tracking/Vertexing Workflow



PODIO

Production

Overall Status:
- Basic workflow in place
- What's in place?
  > All basic components
- What's missing?
  > edm4eic::Vertex associatedParticles not filled

Workflow Inputs/Outputs
- Inputs: CentralCKF(Seeded)ActsTrajectories
- Outputs: CentralTrackVertices (edm4eic::Vertex)

Analysis/User?

Near Term Goals:
- ***primary-vertexing benchmark for TDR***
- ***fill in missing associatedParticles in output***

Long Term Goals:
- algorithm/parameter tuning for different classes of events
- MC/generated vertices and associations
- secondary vertexing

Vertexing Algorithm and edm4eic Vertex

IterativeVertexFinder

- Input: CentralCKFActsTrajectories / CentralCFKSeededActsTrajectories
- default 1D ZScan for vertex seeding (options to use beam line constraints, not in default)
    - logPt weight used with pT_min = 0.4 GeV/c
- output written to CentralTrackVertices (edm4eic::vertex)
    - *associatedParticles not filled at this moment*

https://github.com/eic/EICrecon/blob/main/src/global/tracking/tracking.cc

```
210         app->Add(new JOmniFactoryGeneratorT<IterativeVertexFinder_factory>(
211                 "CentralTrackVertices",
212                 {"CentralCKFActsTrajectories"},
213                 {"CentralTrackVertices"},
214                 {},
215                 app
216                 ));
```

"CentralCKFSeededActTrajectories" works well too, want to update for default in main branch

```
460     ## ============================================================================
461     ## Vertexing
462     ## ============================================================================
463
464     edm4eic::Vertex:
465       Description: "EIC vertex"
466       Author: "J. Osborn"
467       Members:
468         - int32_t             type          // Type flag, to identify what type of vertex it is (e.g. primary, secondary, generated, etc.)
469         - float               chi2          // Chi-squared of the vertex fit
470         - int                 ndf           // NDF of the vertex fit
471         - edm4hep::Vector4f   position      // position [mm] + time t0 [ns] of the vertex. Time is 4th component in vector
472         ## this is named "covMatrix" in EDM4hep, renamed for consistency with the rest of edm4eic
473         - edm4eic::Cov4f      positionError // Covariance matrix of the position+time. Time is 4th component, similarly to 4vector
474       OneToManyRelations:
475         - edm4eic::ReconstructedParticle associatedParticles // particles associated to this vertex.
```

Associated Particles in Vertex

According to Woulter, S&C team is working on a global update to the data model so the PODIO output objects keep the links to the Acts objects. This requires a new version of Acts and will need 3+ months?

In the meantime, we are working on an intermediate solution so users have the access to the associated particles from vertices.

tracking plugins

```
app->Add(new JOmniFactoryGeneratorT<TracksToParticles_factory>(
        "ChargedSeededParticlesWithAssociations",
        {"MCParticles",                                  // edm4hep::MCParticle
        "CentralCKFSeededTracks",                        // edm4eic::Track
        },
        {"ReconstructedSeededChargedWithoutPIDParticles",           //
         "ReconstructedSeededChargedWithoutPIDParticleAssociations"  // edm4eic::MCRecoParticleAssociation
        },
        link_cfg,
        app
        ));

app->Add(new JOmniFactoryGeneratorT<IterativeVertexFinder_factory>(
        "CentralTrackVertices",
        {"CentralCKFSeededActsTrajectories","ReconstructedSeededChargedWithoutPIDParticles"},
        {"CentralTrackVertices"},
        {},
        app
        ));
```

Move to the end in tracking plugins

New input added

Input arguments in IterativeVertexFinder.cc

```
std::unique_ptr<edm4eic::VertexCollection> eicrecon::IterativeVertexFinder::produce(
    std::vector<const ActsExamples::Trajectories*> trajectories,
    std::vector<const edm4eic::ReconstructedParticle*> reconParticles) {
```

New input added

Filling part in IterativeVertexFinder.cc

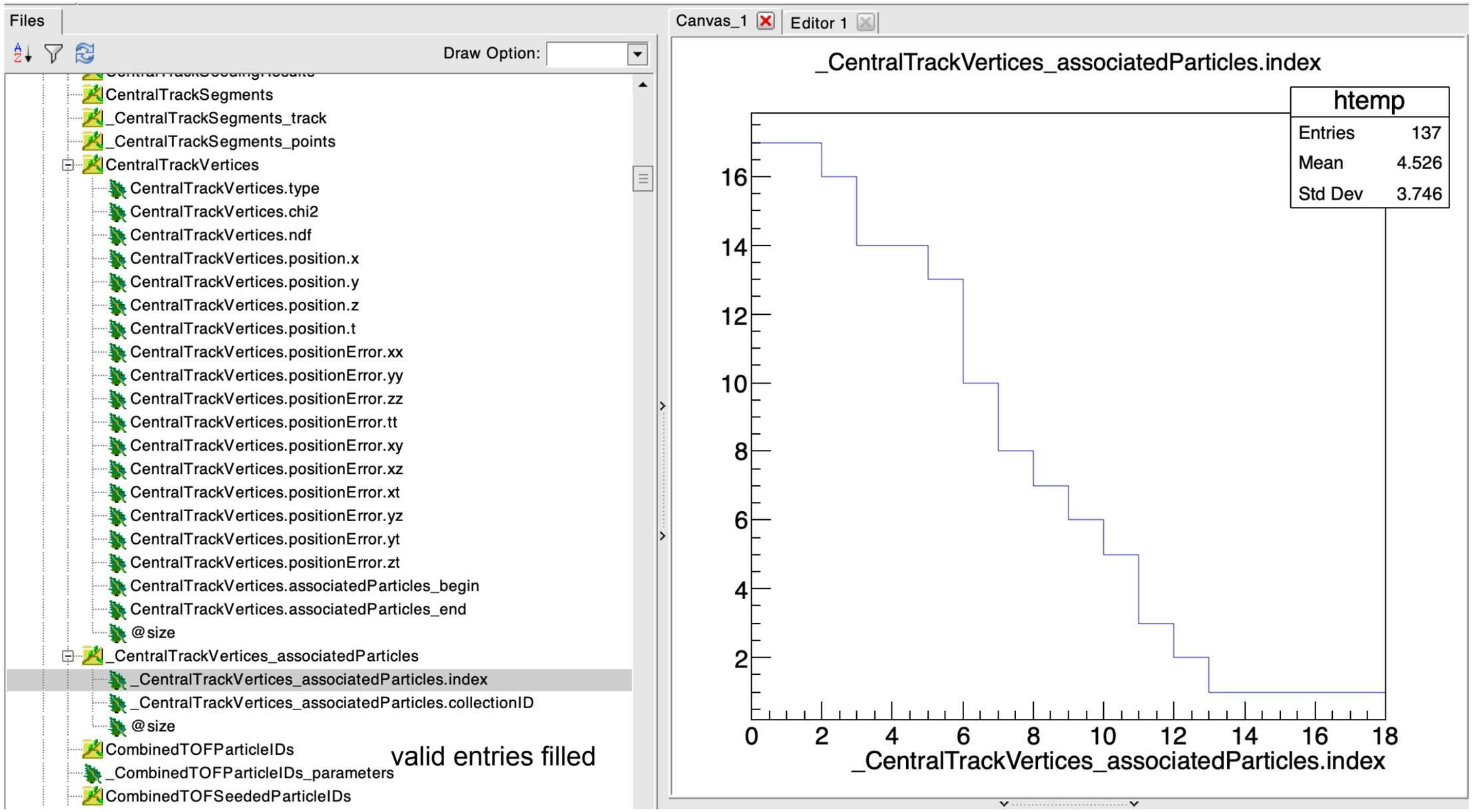
```
    for (const auto& t : vtx.tracks()) {
#if Acts_VERSION_MAJOR >= 33
      const auto& trk = &t.originalParams;
      const auto& par = finderCfg.extractParameters(trk);
#else
      const auto& par = *t.originalParams;
#endif
      m_log->debug(" === track local position from vertex = {}, {}", par.localPosition().x(), par.localPosition().y());
      float loc_a = par.localPosition().x();
      float loc_b = par.localPosition().y();

      for (const auto part : reconParticles) {
        const auto& tracks = part->getTracks();
        for (const auto trk : tracks) {
          const auto& traj = trk.getTrajectory();
          const auto& trkPars = traj.getTrackParameters();
          for (const auto par : trkPars) {
            if(fabs(par.getLoc().a - loc_a) < 1.e-4 && fabs(par.getLoc().b - loc_b) < 1.e-4) {
              m_log->debug(" --- From ReconParticles, track local position = {}, {}", par.getLoc().a, par.getLoc().b);
//              std::cout << " par from ReconParticles " << par.getLoc().a << "\t" << par.getLoc().b << std::endl;
              eicvertex.addToAssociatedParticles(*part);
            } // endif
          } // end for par
        } // end for trk
      } // end for part
    } // end for t
    m_log->info(" +++ This vertex found at (x,y,z) = ({}, {}, {}) mm.", vtx.position().x(), vtx.position().y(), vtx.position().z());
```

Compare track parameters

Next plan: fill in the DCA point information to the vertex in Reconstructed particles.

PODIO output



Compared the associatedParticle array size, consistent with the ACTS vertexing output
Working on more detailed checks on any potential issue

Next Plan to Update ReconstructedParticle

```
## ============================================================================
## Particle info
## ============================================================================

edm4eic::ReconstructedParticle:
  Description: "EIC Reconstructed Particle"
  Author: "W. Armstrong, S. Joosten, F. Gaede"
  Members:
    - int32_t           type              // type of reconstructed particle. Check/set collection parameters ReconstructedParticleTypeNames and ReconstructedParticleTypeValues.
    - float             energy            // [GeV] energy of the reconstructed particle. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f momentum          // [GeV] particle momentum. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f referencePoint    // [mm] reference, i.e. where the particle has been measured
    - float             charge            // charge of the reconstructed particle.
    - float             mass              // [GeV] mass of the reconstructed particle, set independently from four vector. Four momentum state is not kept consistent internally.
    - float             goodnessOfPID     // overall goodness of the PID on a scale of [0;1]
    - edm4eic::Cov4f    covMatrix         // covariance matrix of the reconstructed particle 4vector (10 parameters).
    ##@TODO: deviation from EDM4hep: store explicit PDG ID here. Needs to be discussed how we
    ##       move forward as this could easiliy become unwieldy without this information here.
    ##       The only acceptable alternative would be to store reconstructed identified
    ##       particles in separate collections for the different particle types (which would
    ##       require some algorithmic changes but might work. Doing both might even make
    ##       sense. Needs some discussion, note that PID is more emphasized in NP than
    ##       HEP).
    - int32_t           PDG               // PDG code for this particle
    ## @TODO: Do we need timing info? Or do we rely on the start vertex time?
  OneToOneRelations:
    - edm4eic::Vertex      startVertex    // Start vertex associated to this particle
    - edm4hep::ParticleID  particleIDUsed // particle ID used for the kinematics of this particle
  OneToManyRelations:
    - edm4eic::Cluster     clusters       // Clusters used for this particle
    - edm4eic::Track       tracks         // Tracks used for this particle
    - edm4eic::ReconstructedParticle particles // Reconstructed particles that have been combined to this particle
    - edm4hep::ParticleID  particleIDs    // All associated particle IDs for this particle (not sorted by likelihood)
```

Plan to update these in ReconstructedParticle after vertexing

Vertexing Benchmark

1) Barak is working on a "tracking_performance_dis" benchmark; will integrate the vertexing code into this benchmark:

https://github.com/eic/detector_benchmarks/tree/tracking_performance_dis

2) Shujie also suggested to add it to the existing DIS physics benchmark:

https://github.com/eic/physics_benchmarks/tree/master/benchmarks/dis

3) HF&Jet specific?

Khushi and Rongrong will be helping on integrating the vertexing performance plots into the benchmark repositories