BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

ePIC Track Reconstruction Status

Shujie Li
ePIC tracking WG meeting

Feb 23, 2023

ePIC

Summary

works

- Generate test particles
- GEANT simulation
  - Detailed geometry
  - Digitization at pixel level*
- Hit info to ACTS
- Initial guess for CKF
  - truth params smeared
  - seeding to init params
- CKF track finding/fitting algorithm
- Track params from fit
- Event display
  - root script available on github (Shyam)
  - HSF/Phoenix online server (Sakib)

To finish

- Hit clustering (Shujie)
  - Smearing at hit rather than pixel level to resolve multi hits
  - *clusterization algorithm*
- ACTS Seed finding/filter (Rey, Barak)
- Track info from ACTS
  - Raw hits → primary particle association (Barak)
  - Hits used w/track association
  - χ^2, # of measurements to rootfile (Shyam)
- Optimize track quality cuts (Beatrice)
  - χ^2 , # of measurements
- Validation plots
- Background embedding (Kolja)

Track Reconstruction in EICrecon

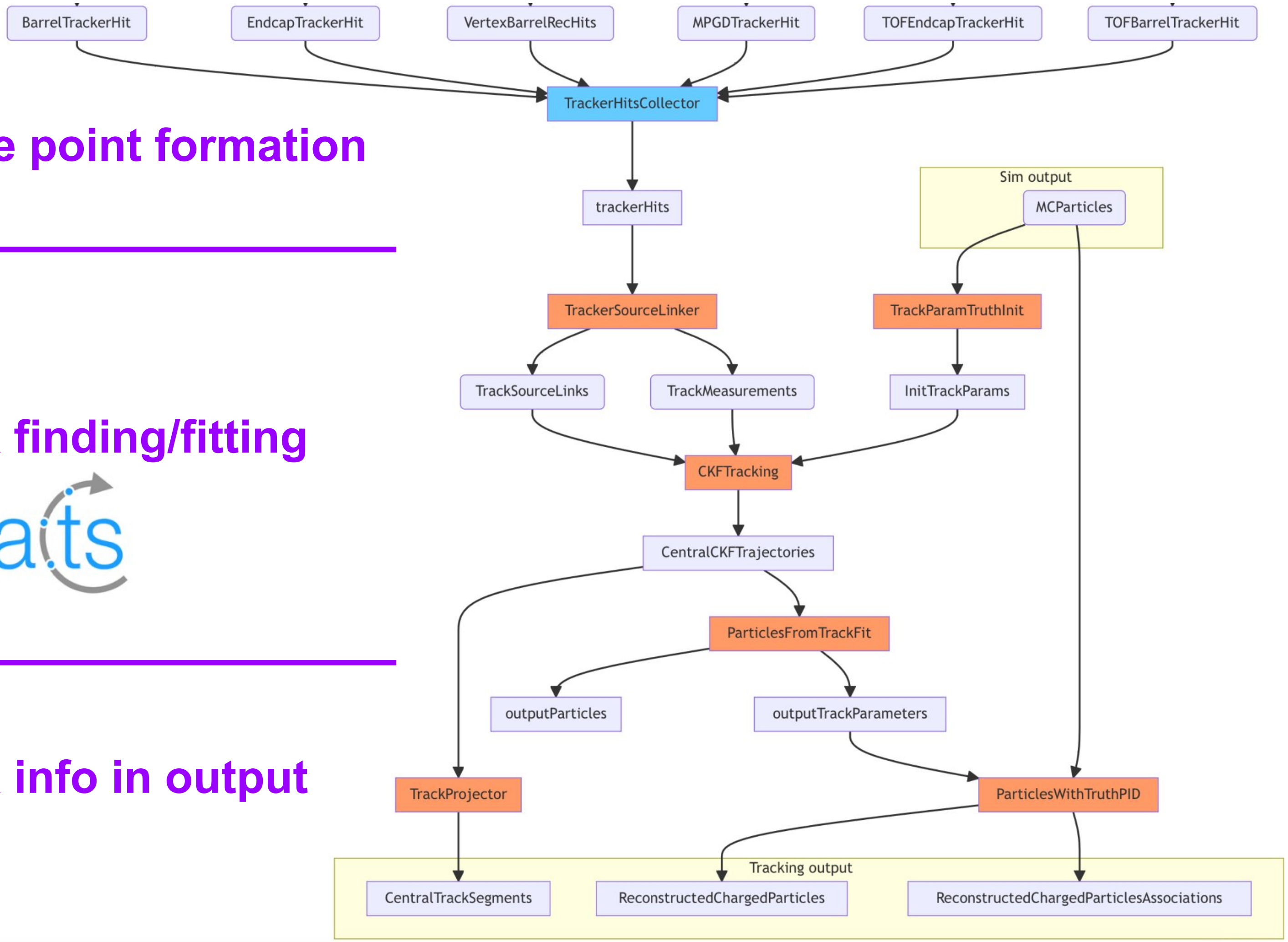
contact: Dmitry Romanov

Full diagram at https://eic.github.io/EICrecon/#/design/tracking?id=full-diagram

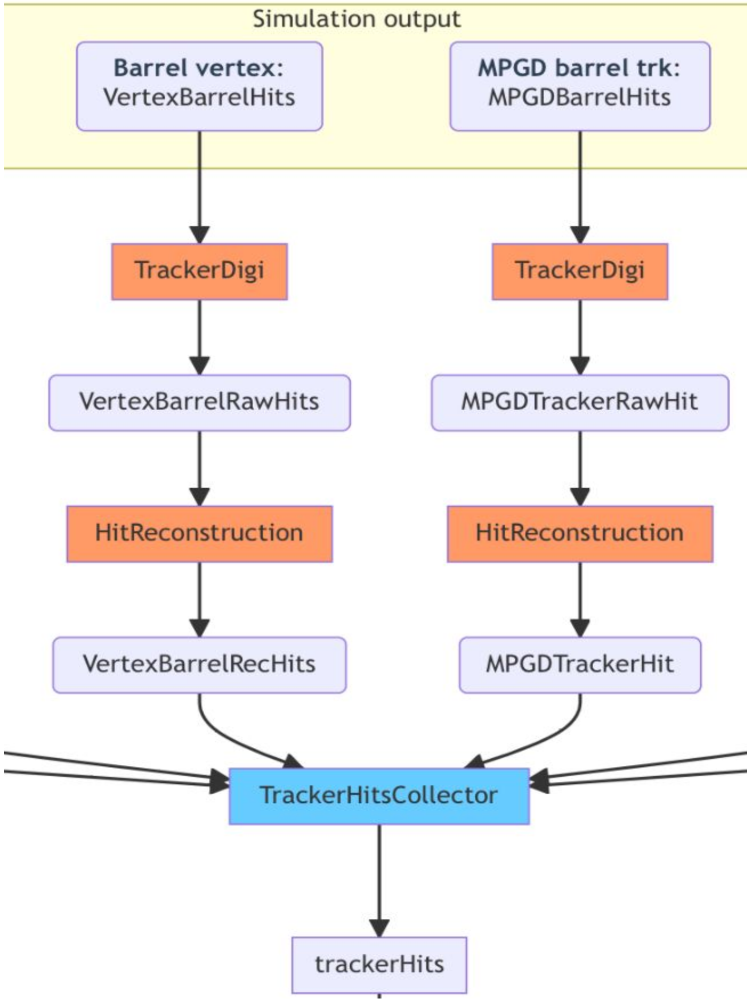
Space point formation

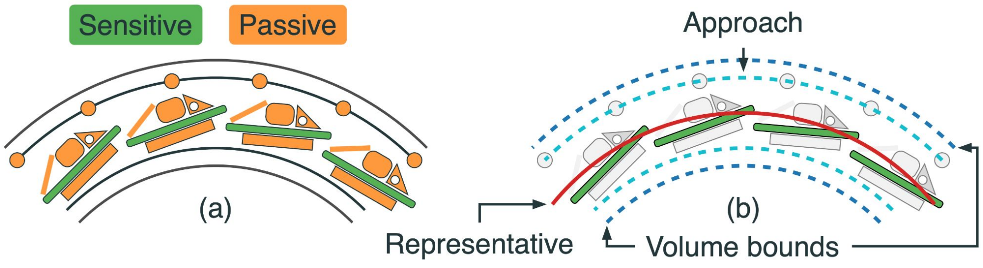
Track finding/fitting with acts

Track info in output

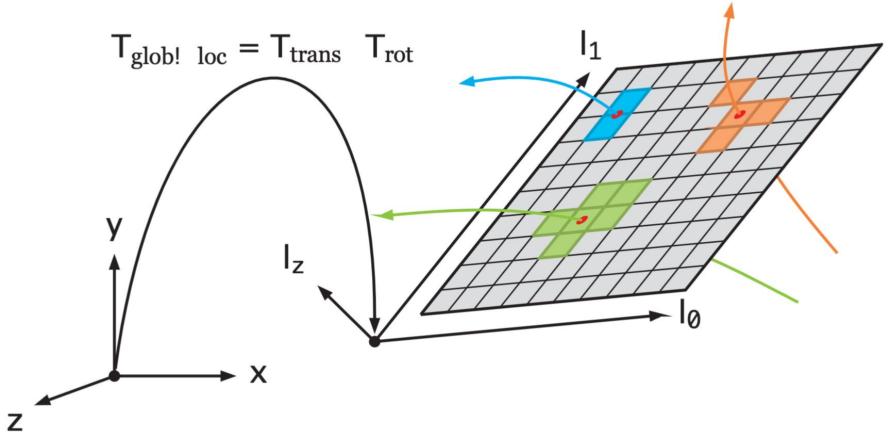


Space Point Formation

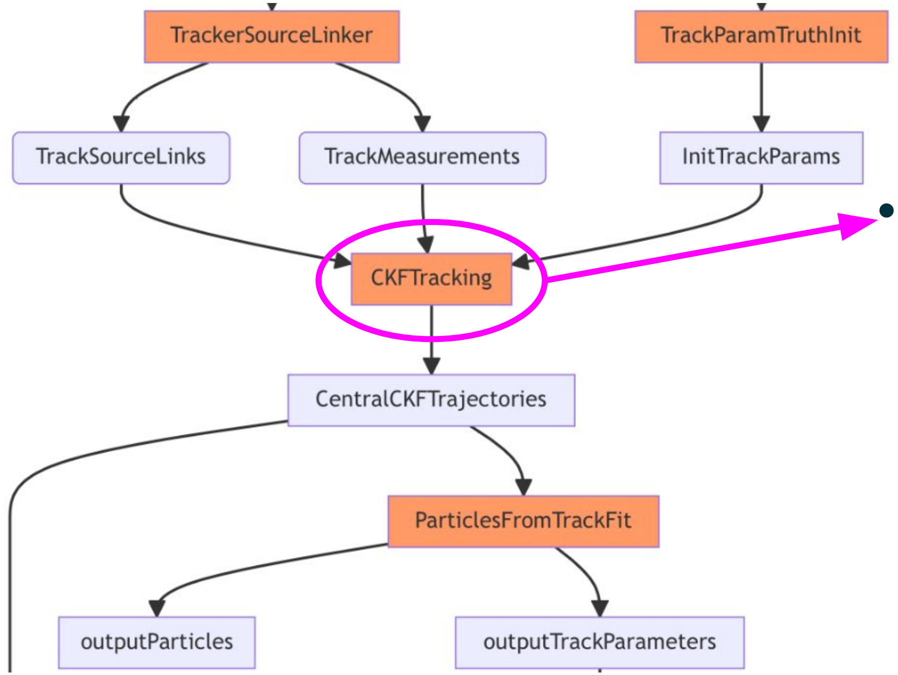




- Global / local coord. transformation
- Digitization:
    - Raw hits -> Surface and cell ID
    - Energy deposit threshold:
        - Now: 0, to use : 110 electrons
    - Clustering algorithm available at https://github.com/acts-project/acts/pull/1190 (Louis-Guillaume Gagnon, March 9th)


$T_{\text{glob!}\ \text{loc}} = T_{\text{trans}}\ \ T_{\text{rot}}$
l_1
y
l_z
l_0
x
z

Track Finding/Fitting with ACTS



TrackerSourceLinker

TrackParamTruthInit

TrackSourceLinks

TrackMeasurements

InitTrackParams

CKFTracking

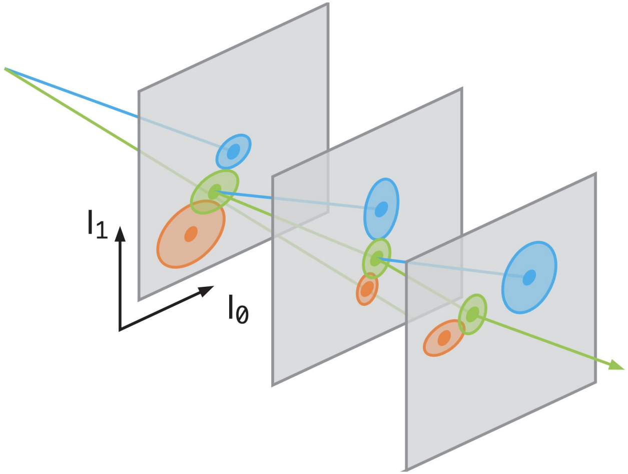
CentralCKFTrajectories

ParticlesFromTrackFit

outputParticles

outputTrackParameters

- **Combinatorial Kalman Filter (CKF)**
  - combine track finding and fitting
  - allows track branching
    - → user-defined measurement selector (number, chi2)
  - high efficiency
  - **Need a reasonable “initial guess”**



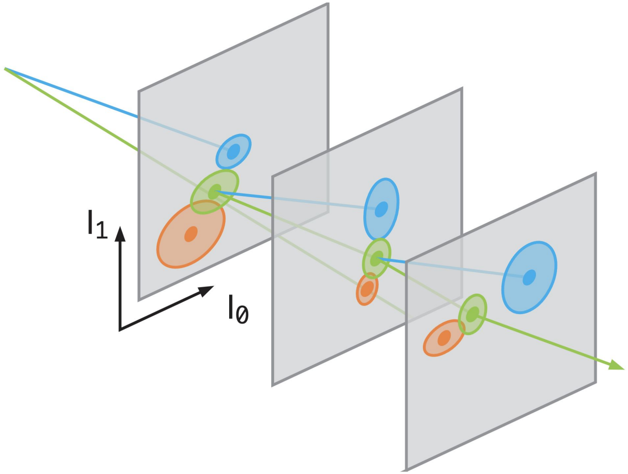
EICrecon: JANA2 based recon framework

EICrecon factory (interface)

ACTS: CKF Algorithm

Hits selection

acts/Core/include/Acts/TrackFinding/MeasurementSelector.hpp



CKF:

if no hits on surface→ **nHoles++**

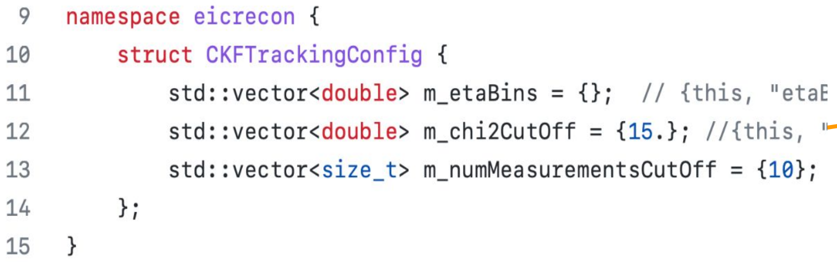
for (track state : track state candidates):

    Track state → hits on surface

    Calculate chi2 of all hits and rank, find chi2min

    if chi2min >chi2CutOff → save chi2min as **outlier**

    <chi2CutOff → save up to num**Measurements**CutOff candidates



```
namespace eicrecon {
    struct CKFTrackingConfig {
        std::vector<double> m_etaBins = {};  // {this, "etaB
        std::vector<double> m_chi2CutOff = {15.}; //{this, "
        std::vector<size_t> m_numMeasurementsCutOff = {10};
    };
}
```

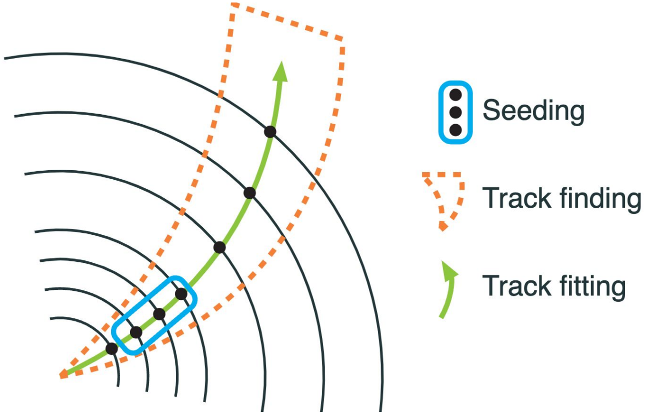
optimize cuts (Beatrice)

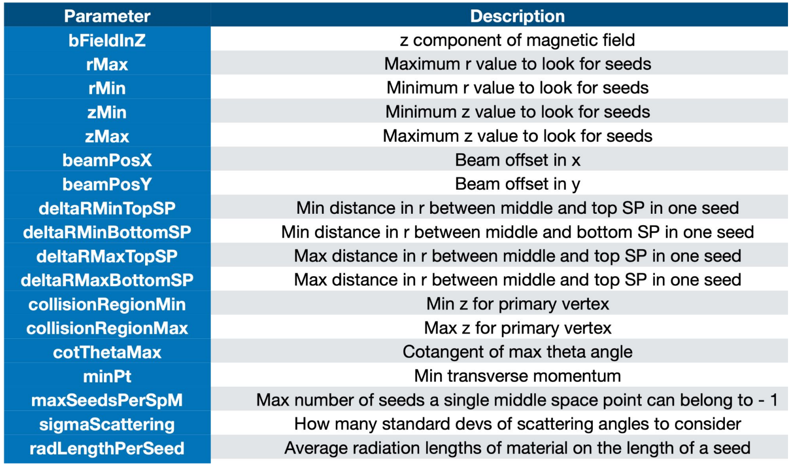
# of sensitive surfaces = nHoles + nMeasurements + nOutliers

Initial Guess for CKF: 2. realistic seeding

Seeder: a set of three space points to estimate initial track parameters

- **Binned seeder**: loop over φ-z binning to try all combinations. Slow at large η
  - tested and bugs fixed. See YueShi Lai's work
- **Orthogonal seed finder**: can efficiently search for space points within a given range.
  - Initial implementation in EICrecon - Joe Osborn
  - Seeder configuration:
    - default parameters from binned seeder
    - parameter optimization - See Rey Cruz-Torres's work
  - Seeder confirmation/filter





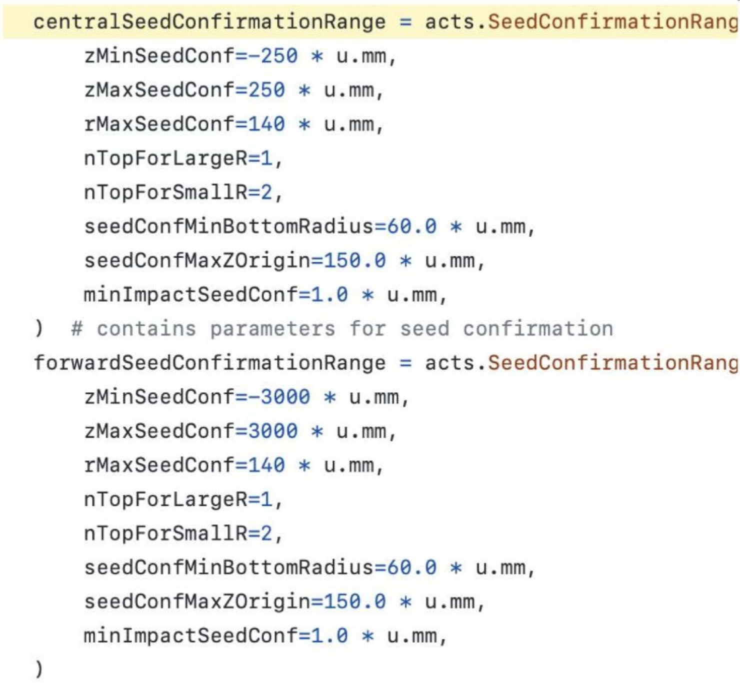
Parameter	Description
bFieldInZ	z component of magnetic field
rMax	Maximum r value to look for seeds
rMin	Minimum r value to look for seeds
zMin	Minimum z value to look for seeds
zMax	Maximum z value to look for seeds
beamPosX	Beam offset in x
beamPosY	Beam offset in y
deltaRMinTopSP	Min distance in r between middle and top SP in one seed
deltaRMinBottomSP	Min distance in r between middle and bottom SP in one seed
deltaRMaxTopSP	Max distance in r between middle and top SP in one seed
deltaRMaxBottomSP	Max distance in r between middle and top SP in one seed
collisionRegionMin	Min z for primary vertex
collisionRegionMax	Max z for primary vertex
cotThetaMax	Cotangent of max theta angle
minPt	Min transverse momentum
maxSeedsPerSpM	Max number of seeds a single middle space point can belong to - 1
sigmaScattering	How many standard devs of scattering angles to consider
radLengthPerSeed	Average radiation lengths of material on the length of a seed

Initial Guess for CKF: 2. realistic seeding

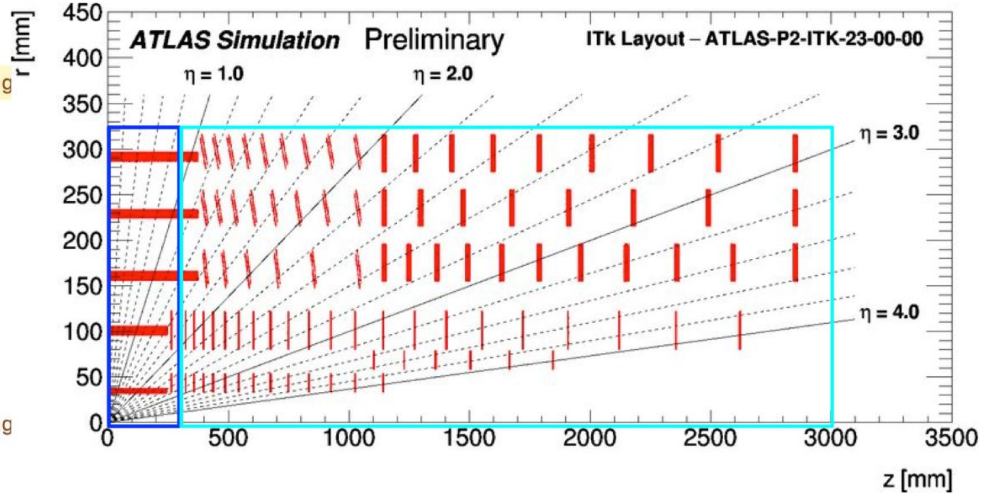
Seed Confirmation/Filter

Individual filter settings for each geometry region.

- Experience from ATLAS-ITK , see Luis Falda Coelho's work
- implementation in EICrecon, **TBD** - Rey, Barak Schmookler



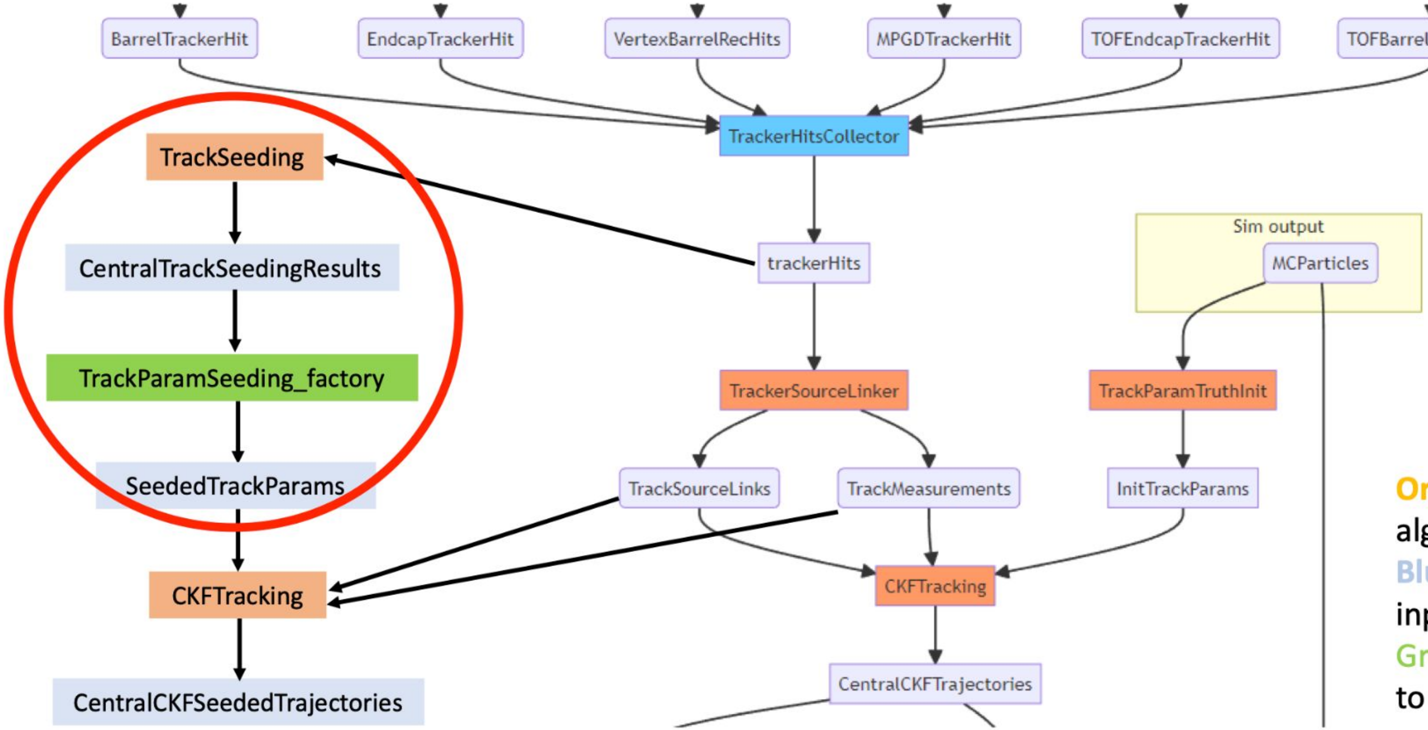
```
centralSeedConfirmationRange = acts.SeedConfirmationRang
    zMinSeedConf=-250 * u.mm,
    zMaxSeedConf=250 * u.mm,
    rMaxSeedConf=140 * u.mm,
    nTopForLargeR=1,
    nTopForSmallR=2,
    seedConfMinBottomRadius=60.0 * u.mm,
    seedConfMaxZOrigin=150.0 * u.mm,
    minImpactSeedConf=1.0 * u.mm,
)  # contains parameters for seed confirmation
forwardSeedConfirmationRange = acts.SeedConfirmationRang
    zMinSeedConf=-3000 * u.mm,
    zMaxSeedConf=3000 * u.mm,
    rMaxSeedConf=140 * u.mm,
    nTopForLargeR=1,
    nTopForSmallR=2,
    seedConfMinBottomRadius=60.0 * u.mm,
    seedConfMaxZOrigin=150.0 * u.mm,
    minImpactSeedConf=1.0 * u.mm,
)
```



Initial Guess for CKF: 2. realistic seeding

Supply realistic init parameters to CKF

- CKF with realistic seeding in addition to truth seeding. See Barak’s work
  - retain data structure for current downstream analysis
- Switch between truth / realistic seeding. TBD. See Dmitry’s work



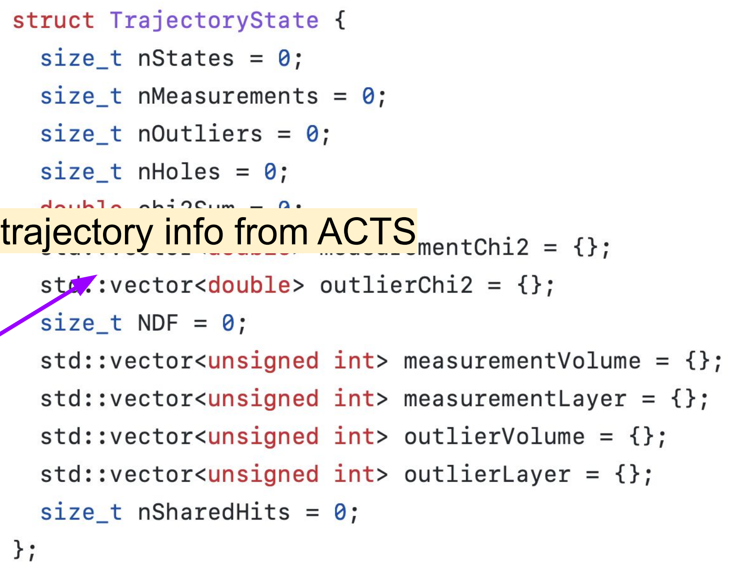
Orange boxes are algorithms. Blue/Grey boxes are input/output data. Green box is a factory to reformat data type.

Track Info in Output

- Track parameters from fit - Done
- Track projection - Done

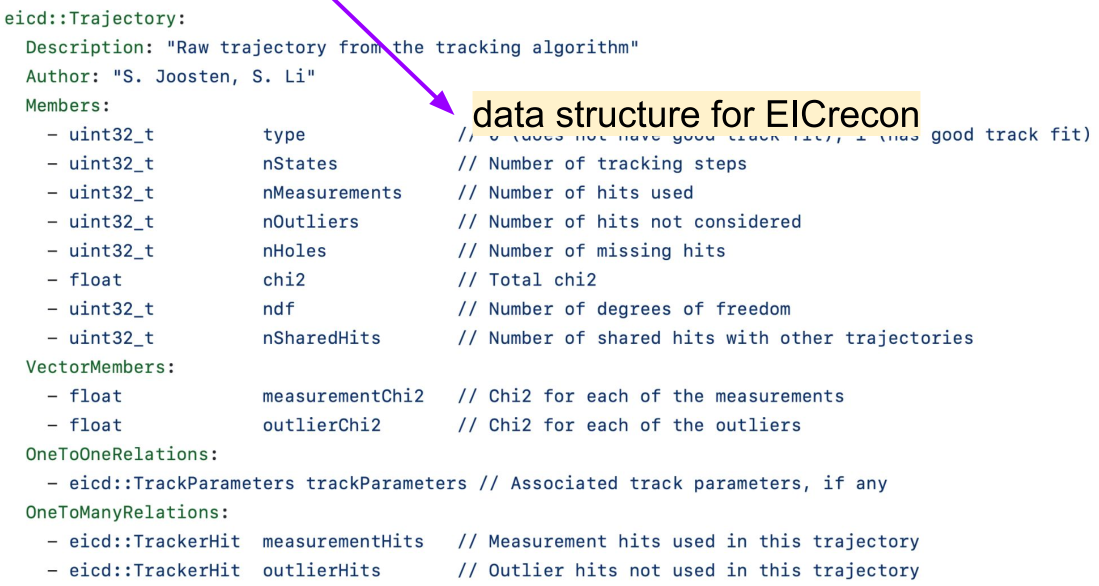
- Trajectory info (chi2, number of hits …)
  - save to histograms with EICrecon plugins
  - save to output rootfile:
    - **TBD**: write an EICrecon factory to write trajectory info into data structure

- **TBD:** Hits associated with tracks



trajectory info from ACTS

```
struct TrajectoryState {
  size_t nStates = 0;
  size_t nMeasurements = 0;
  size_t nOutliers = 0;
  size_t nHoles = 0;
  double chi2Sum = 0;
  std::vector<double> measurementChi2 = {};
  std::vector<double> outlierChi2 = {};
  size_t NDF = 0;
  std::vector<unsigned int> measurementVolume = {};
  std::vector<unsigned int> measurementLayer = {};
  std::vector<unsigned int> outlierVolume = {};
  std::vector<unsigned int> outlierLayer = {};
  size_t nSharedHits = 0;
};
```



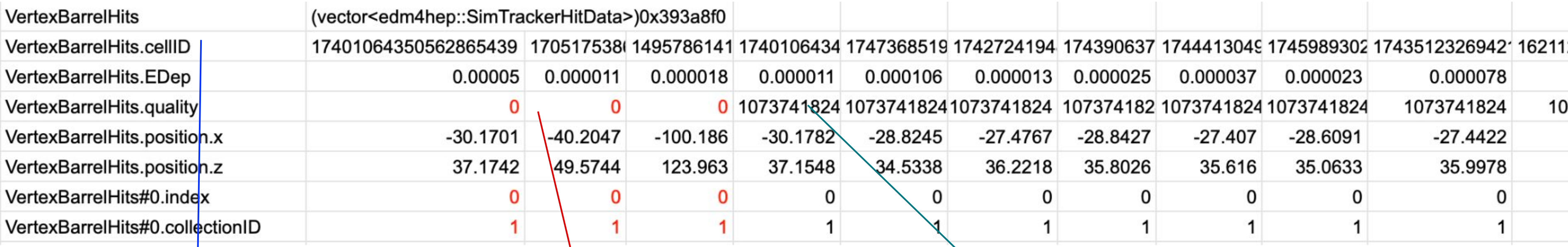
data structure for EICrecon

```
eicd::Trajectory:
  Description: "Raw trajectory from the tracking algorithm"
  Author: "S. Joosten, S. Li"
  Members:
    - uint32_t         type              // 0 (does not have good track fit), 1 (has good track fit)
    - uint32_t         nStates           // Number of tracking steps
    - uint32_t         nMeasurements     // Number of hits used
    - uint32_t         nOutliers         // Number of hits not considered
    - uint32_t         nHoles            // Number of missing hits
    - float            chi2              // Total chi2
    - uint32_t         ndf               // Number of degrees of freedom
    - uint32_t         nSharedHits       // Number of shared hits with other trajectories
  VectorMembers:
    - float            measurementChi2   // Chi2 for each of the measurements
    - float            outlierChi2       // Chi2 for each of the outliers
  OneToOneRelations:
    - eicd::TrackParameters trackParameters // Associated track parameters, if any
  OneToManyRelations:
    - eicd::TrackerHit  measurementHits   // Measurement hits used in this trajectory
    - eicd::TrackerHit  outlierHits       // Outlier hits not used in this trajectory
```

MCParticle-Hits Association

https://github.com/AIDASoft/DD4hep/issues/126
https://github.com/AIDASoft/DD4hep/pull/145

See Barak’s work

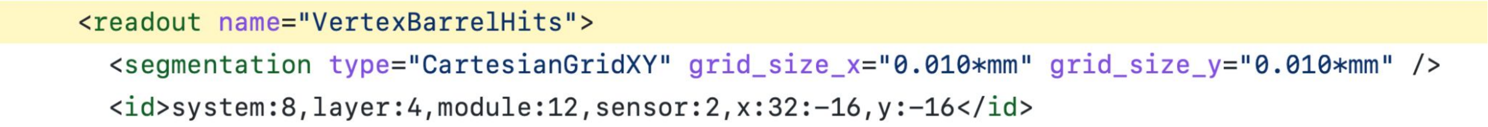


VertexBarrelHits	(vector<edm4hep::SimTrackerHitData>)0x393a8f0										
VertexBarrelHits.cellID	17401064350562865439	170517538	1495786141	1740106434	1747368519	1742724194	174390637	1744413049	1745989302	1743512326942	16211
VertexBarrelHits.EDep	0.00005	0.000011	0.000018	0.000011	0.000106	0.000013	0.000025	0.000037	0.000023	0.000078	
VertexBarrelHits.quality	0	0	0	1073741824	1073741824	1073741824	107374182	1073741824	1073741824	1073741824	10
VertexBarrelHits.position.x	-30.1701	-40.2047	-100.186	-30.1782	-28.8245	-27.4767	-28.8427	-27.407	-28.6091	-27.4422	
VertexBarrelHits.position.z	37.1742	49.5744	123.963	37.1548	34.5338	36.2218	35.8026	35.616	35.0633	35.9978	
VertexBarrelHits#0.index	0	0	0	0	0	0	0	0	0	0	
VertexBarrelHits#0.collectionID	1	1	1	1	1	1	1	1	1	1	

segmentation defined in each detector xml file

3 primary hits on 3 vertex layers with quality = 0

quality>0 means hits from secondaries **BUT** those secondaries didn’t pass dd4hep cut **10 MeV**, so the are **NOT** saved in MCparticle.
⇒Those hits are linked back to primary (parent) particles.

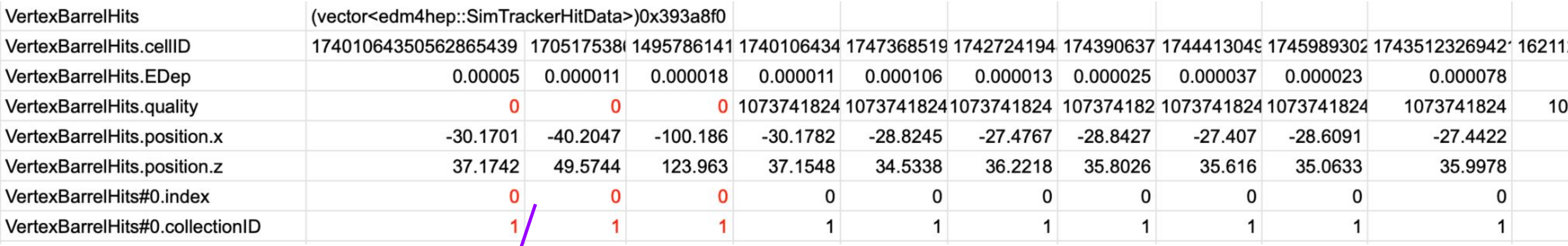


```
    <readout name="VertexBarrelHits">
      <segmentation type="CartesianGridXY" grid_size_x="0.010*mm" grid_size_y="0.010*mm" />
      <id>system:8,layer:4,module:12,sensor:2,x:32:-16,y:-16</id>
```

MCParticle-Hits Association

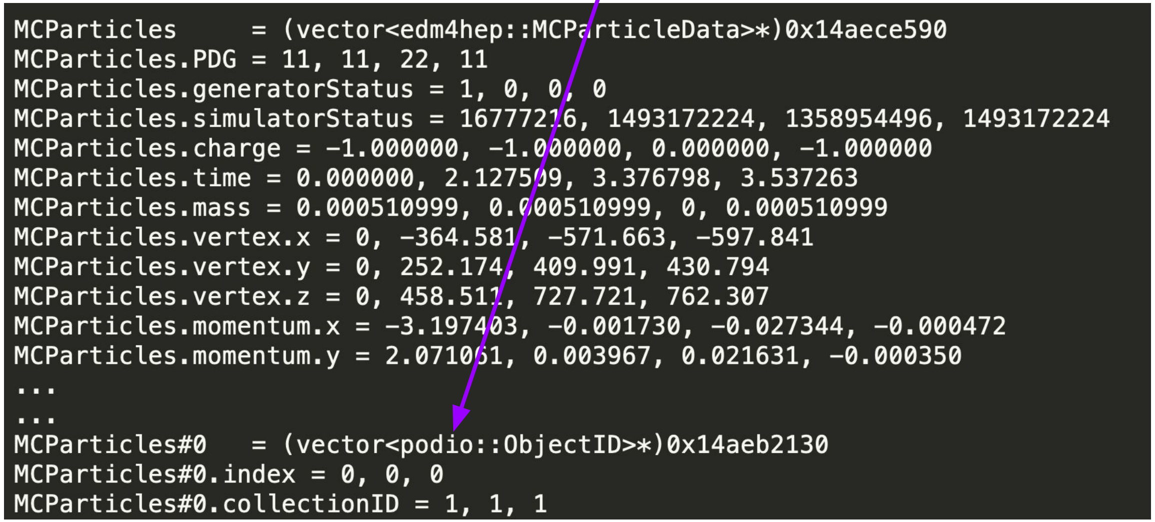
https://github.com/AIDASoft/DD4hep/issues/126
https://github.com/AIDASoft/DD4hep/pull/145

See Barak's work

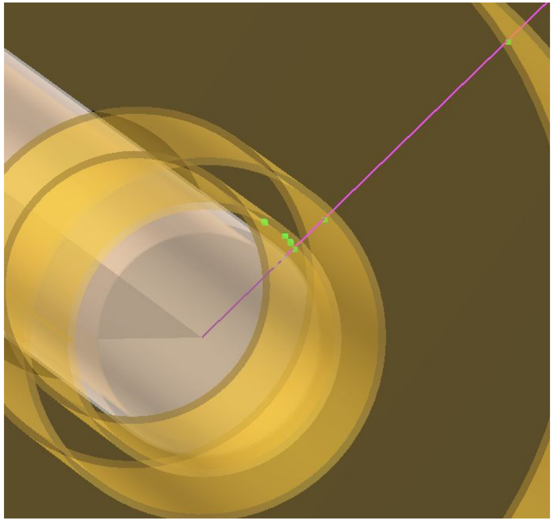


VertexBarrelHits	(vector<edm4hep::SimTrackerHitData>)0x393a8f0										
VertexBarrelHits.cellID	17401064350562865439	170517538	1495786141	1740106434	1747368519	1742724194	174390637	174441304	174598930	174351232694	16211
VertexBarrelHits.EDep	0.00005	0.000011	0.000018	0.000011	0.000106	0.000013	0.000025	0.000037	0.000023	0.000078	
VertexBarrelHits.quality	0	0	0	1073741824	1073741824	1073741824	107374182	1073741824	1073741824	1073741824	10
VertexBarrelHits.position.x	-30.1701	-40.2047	-100.186	-30.1782	-28.8245	-27.4767	-28.8427	-27.407	-28.6091	-27.4422	
VertexBarrelHits.position.z	37.1742	49.5744	123.963	37.1548	34.5338	36.2218	35.8026	35.616	35.0633	35.9978	
VertexBarrelHits#0.index	0	0	0	0	0	0	0	0	0	0	
VertexBarrelHits#0.collectionID	1	1	1	1	1	1	1	1	1	1	

use index to access associated particle/initial track



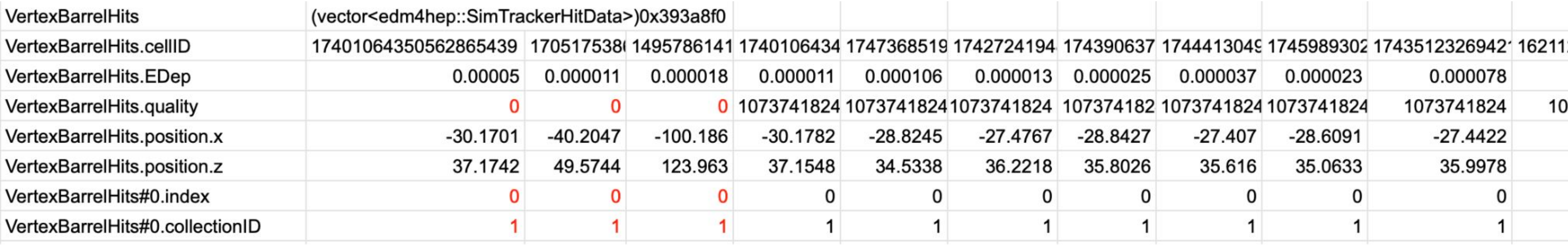
```
MCParticles     = (vector<edm4hep::MCParticleData>*)0x14aece590
MCParticles.PDG = 11, 11, 22, 11
MCParticles.generatorStatus = 1, 0, 0, 0
MCParticles.simulatorStatus = 16777216, 1493172224, 1358954496, 1493172224
MCParticles.charge = -1.000000, -1.000000, 0.000000, -1.000000
MCParticles.time = 0.000000, 2.127509, 3.376798, 3.537263
MCParticles.mass = 0.000510999, 0.000510999, 0, 0.000510999
MCParticles.vertex.x = 0, -364.581, -571.663, -597.841
MCParticles.vertex.y = 0, 252.174, 409.991, 430.794
MCParticles.vertex.z = 0, 458.511, 727.721, 762.307
MCParticles.momentum.x = -3.197403, -0.001730, -0.027344, -0.000472
MCParticles.momentum.y = 2.071061, 0.003967, 0.021631, -0.000350
...
...
MCParticles#0   = (vector<podio::ObjectID>*)0x14aeb2130
MCParticles#0.index = 0, 0, 0
MCParticles#0.collectionID = 1, 1, 1
```



MCParticle-Hits Association

https://github.com/AIDASoft/DD4hep/issues/126
https://github.com/AIDASoft/DD4hep/pull/145

See Barak's work



VertexBarrelHits	(vector<edm4hep::SimTrackerHitData>)0x393a8f0										
VertexBarrelHits.cellID	17401064350562865439	170517538	1495786141	1740106434	1747368519	1742724194	174390637	1744413049	1745989302	1743512326942	16211
VertexBarrelHits.EDep	0.00005	0.000011	0.000018	0.000011	0.000106	0.000013	0.000025	0.000037	0.000023	0.000078	
VertexBarrelHits.quality	0	0	0	1073741824	1073741824	1073741824	107374182	1073741824	1073741824	1073741824	10
VertexBarrelHits.position.x	-30.1701	-40.2047	-100.186	-30.1782	-28.8245	-27.4767	-28.8427	-27.407	-28.6091	-27.4422	
VertexBarrelHits.position.z	37.1742	49.5744	123.963	37.1548	34.5338	36.2218	35.8026	35.616	35.0633	35.9978	
VertexBarrelHits#0.index	0	0	0	0	0	0	0	0	0	0	
VertexBarrelHits#0.collectionID	1	1	1	1	1	1	1	1	1	1	

Solutions to multi-hits:

1. reject secondaries by quality and index values
2. clusterization algorithm in ACTS https://github.com/acts-project/acts/blob/main/Core/include/Acts/Clusterization/Clusterization.hpp