Task 12.2 - Turnkey Software

AIDAinnova, annual meeting

Thomas Madlener

Mar 18, 2024

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 101004761.

AIDA innova

CERN

DESY.

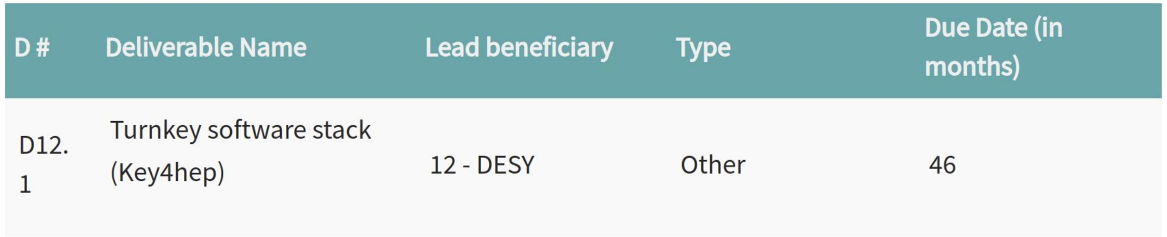
Task 12.2 - Turnkey Software

Overview

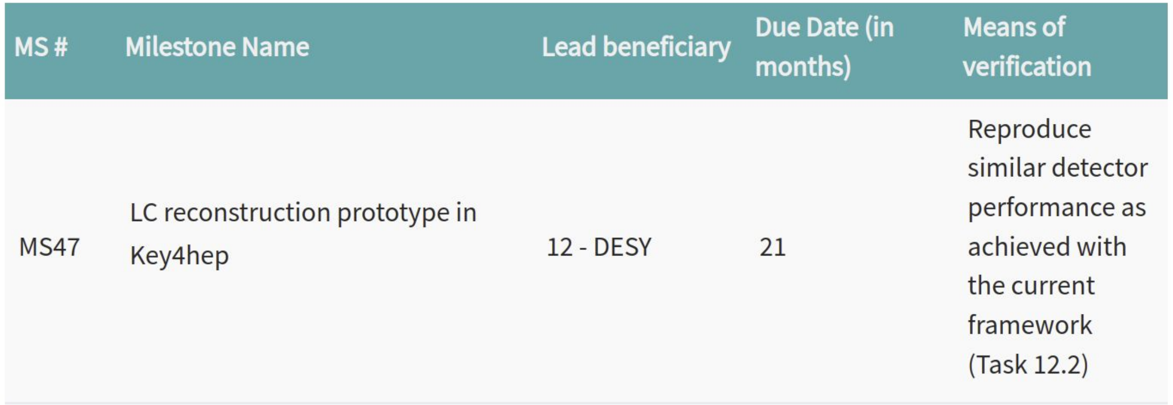
- Integrated Turnkey Software Stack, for physics and performance studies
- Simplified data model toolkit for modern hardware platforms
- Digitisation extensions for geometry toolkit
- R&D study on frameworks to manage heterogeneous resources

DESY (lead), CERN,
INFN-PI, (INFN-PD, INFN-BA, INFN-BO - unfunded)
IHEP, SDU - associated

Symbols on coming slides:
✅ Done
👷 Ongoing or planned



D #	Deliverable Name	Lead beneficiary	Type	Due Date (in months)
D12.1	Turnkey software stack (Key4hep)	12 - DESY	Other	46

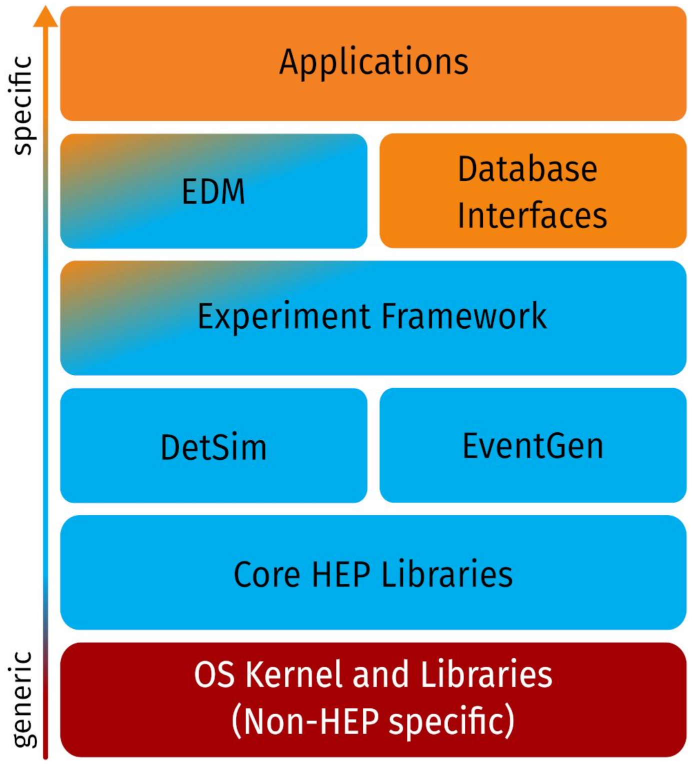


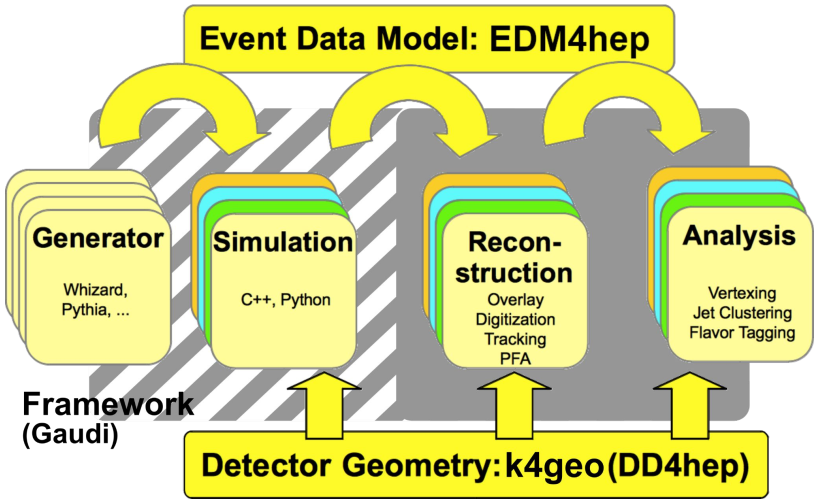
	MS #	Milestone Name	Lead beneficiary	Due Date (in months)	Means of verification
✅	MS47	LC reconstruction prototype in Key4hep	12 - DESY	21	Reproduce similar detector performance as achieved with the current framework (Task 12.2)

Key4hep

Turnkey software stack for all future collider projects

- Develop a **common turnkey software stack** for future collider studies
- Take existing tools where possible, provide necessary interfaces and contribute to the development
  - A lot of existing software from the shared **iLCSoft** developed by ILC and CLIC for many years
- Develop new tools or libraries where necessary
- All major players involved: CEPC, CLIC, FCC, ILC, EIC, …
- Provide a complete data processing framework
  - Shared components reduce overhead for all users
- Make things as easy to use as possible for everybody (librarians, developers, users)
- Supported by **HSF**, **CERN** and **AIDAinnova**





PODIO

github.com/AIDASoft/podio

The event data model toolkit

- Generate code from simple YAML definition of EDM4hep
- Based on using and storing POD (plain old data) structures
- Thread-safe and potential to target multiple I/O backends

✅ **Schema evolution** of generated EDMs (basic use cases, extend as necessary)

✅ RNTuple based backend ✅ Improved python bindings

✅ Stable collection IDs (necessary for multithreading)

✅ Interface types (inheritance free interfaces)

✅ First version of Julia code generation (main feature: machinery, templates still WIP)

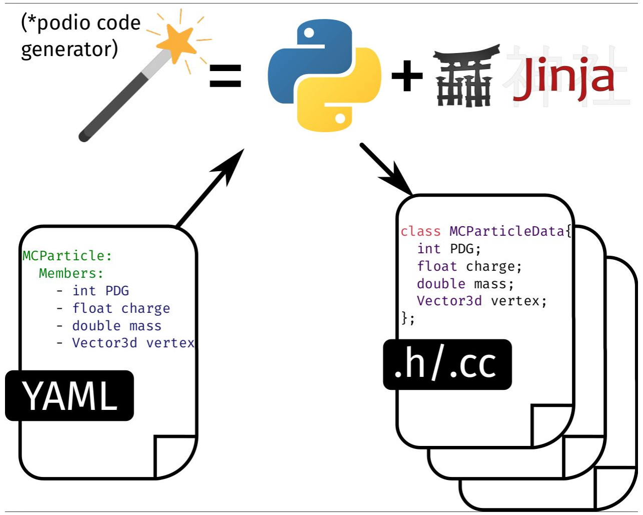
✅ Fixed subtle memory issues (present for quite some time, hard to trigger)

✅ Removal of deprecated EventStore → Only Frame based I/O

👷 **Version 1.0** (effectively done, cleaning up minor issues & documentation now)

👷 Some prototyping and testing on heterogeneous resources

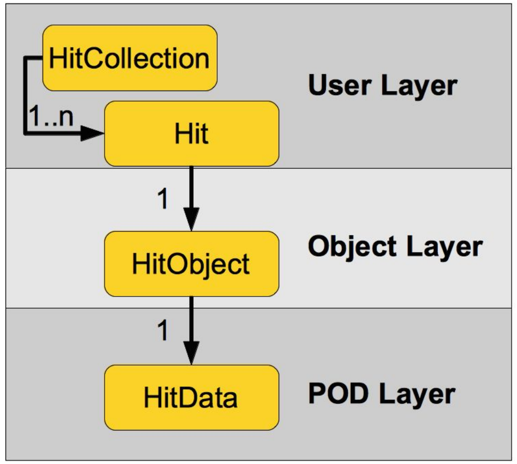
👷 Additional features (already a few on the wish list: e.g. streaming of Frames)



(*podio code generator)

```
MCParticle:
  Members:
    - int PDG
    - float charge
    - double mass
    - Vector3d vertex
```

```
class MCParticleData{
  int PDG;
  float charge;
  double mass;
  Vector3d vertex;
};
```

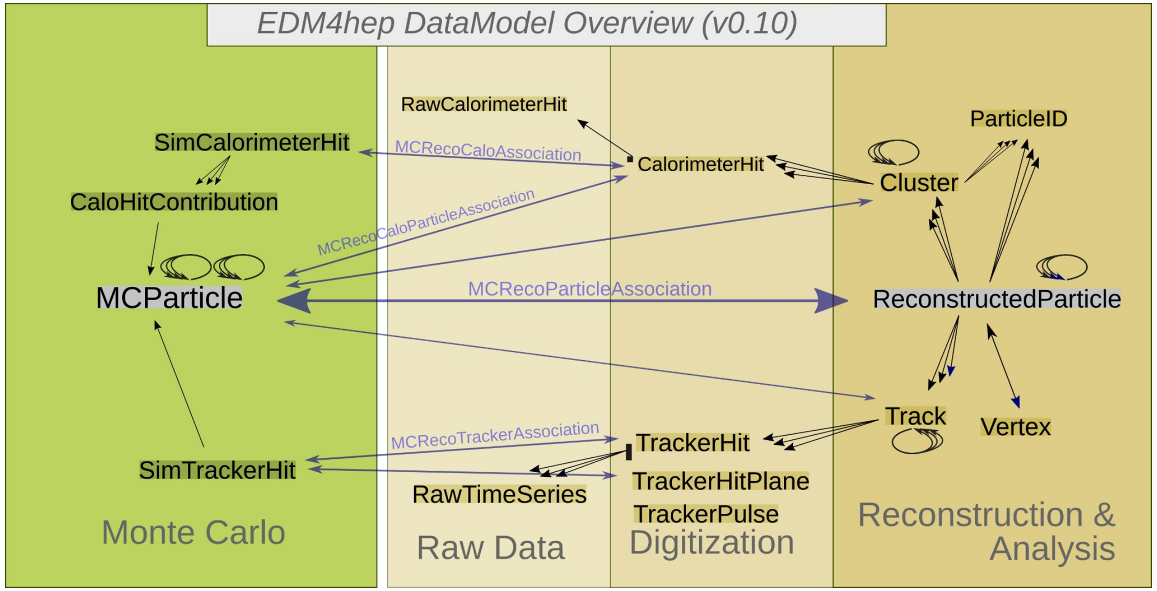


github.com/key4hep/EDM4hep
edm4hep.web.cern.ch

EDM4hep

The common event data model

- EDM4hep defines the common *language* for all Key4hep components to communicate
- Heavily inspired by LCIO that has been successfully shared by ILC and CLIC
- Generated by the PODIO EDM toolkit
  - EDM4hep and EICD main customers of PODIO



✅ (Standalone) conversion from LCIO

✅ Improved python bindings

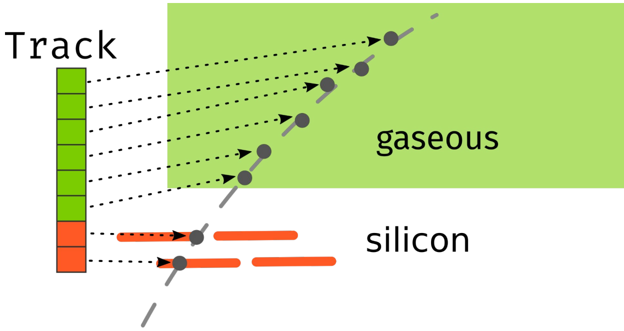
✅ Introduction of TrackerHit interface (covering TrackerHit3D and TrackerHitPlane)

✅ Cleanup of type inconsistencies / naming / documentation

👷 **Version 1.0** (backwards compatible from then on, almost there)

👷 Conceptual changes w.r.t. LCIO

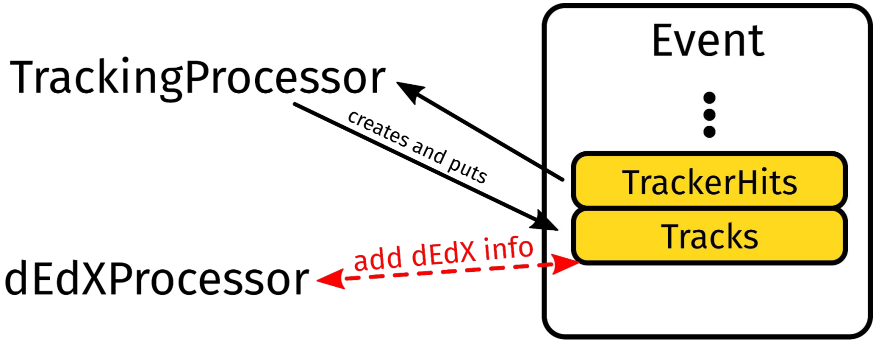
👷 Utility functionality as necessary

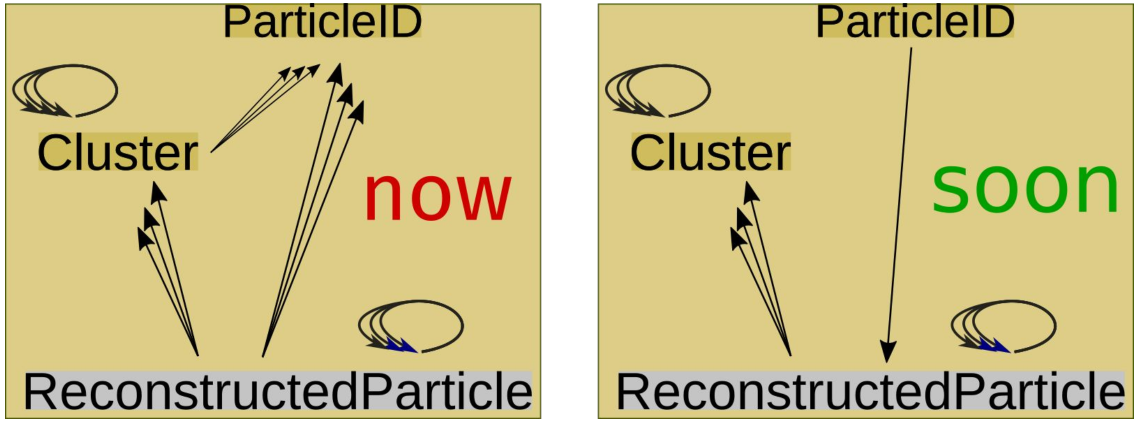


EDM4hep v1.0

The last missing pieces

- Breaking things without intention of doing schema evolution
  - Too many things that do not have an easy evolution mechanism
- **Consistent mutability concept**
  - Some inconsistencies inherited from LCIO
  - Very big conceptual change
  - Requires substantial work in conversion
  - Requires new utilities for users
- Many things already addressed
  - TrackerHit interface
  - Multiple EventWeights
  - Consistent relation naming
- Keeping track in [EDM4hep v1.0 github project]()

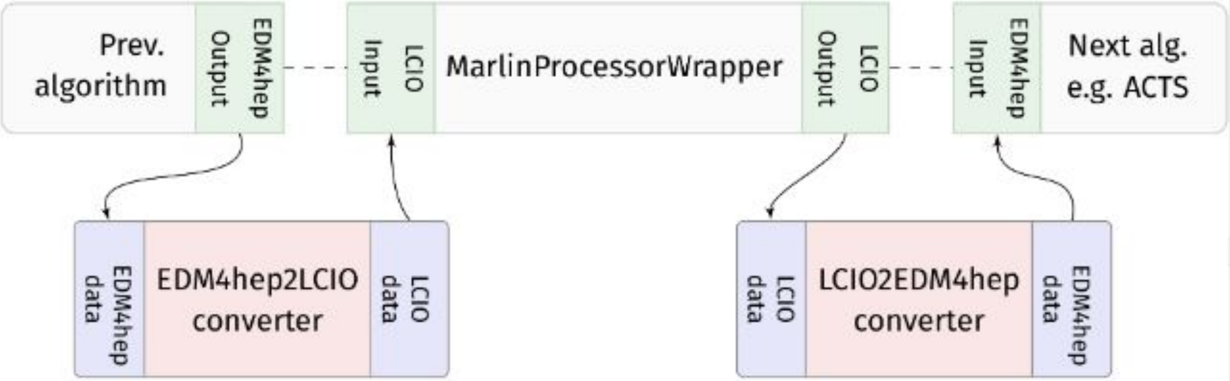




k4MarlinWrapper & EDM converters

Bridging old and new

- Run unchanged Marlin Processors via Wrapper
- Configurable on-demand conversion between LCIO and EDM4hep (and vice versa)
- Steering file conversion script



✅ Unified all EDM4hep ↔ LCIO converters in one library

✅ Added basic unit-test like converter tests

✅ Generalized relation resolving mechanism

✅ Keep track of all converted objects in an event (effectively incrementally build up parallel EDM4hep & LCIO events)

✅ Improved ergonomics (e.g. always convert EventHeader)

👷 Finalize conversion after EDM4hep v1.0

Key4hep Framework & Services

Core services & algorithms

github.com/key4hep
key4hep.web.cern.ch

- Gaudi based core framework
  - **k4FWCore** provides I/O for PODIO based EDMs
- Core services and tools for CI and development

✅ Switch FW core to use podio Frames (+removal of legacy EventStore)

✅ Thread safe way for handling metadata

✅ **Support for Gaudi Functional algorithms**

✅ Improved k4run (better error & help messages)

✅ Builds for multiple OSs

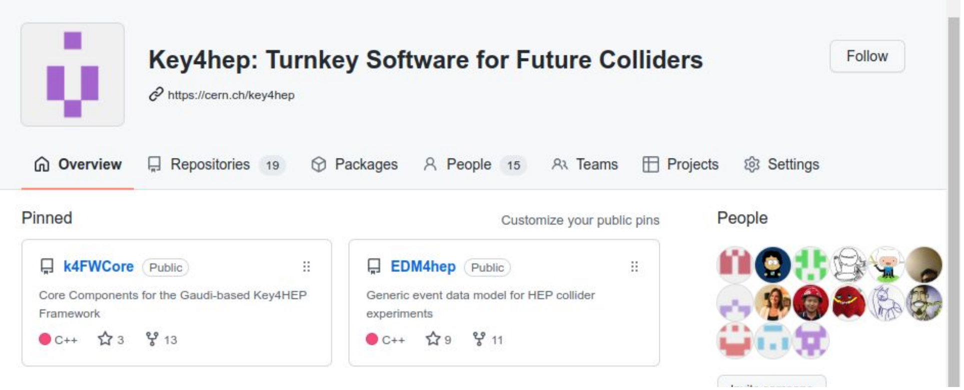
✅ New github actions to facilitate CI for Key4hep packages

✅ Convenience improvements (e.g. read all collections by default)

👷 Proper multithreading support for Functional algorithms

👷 Integration of ACTS (requires TrackerHit interface)

👷 Consolidation of simulation approaches (postponed for now)



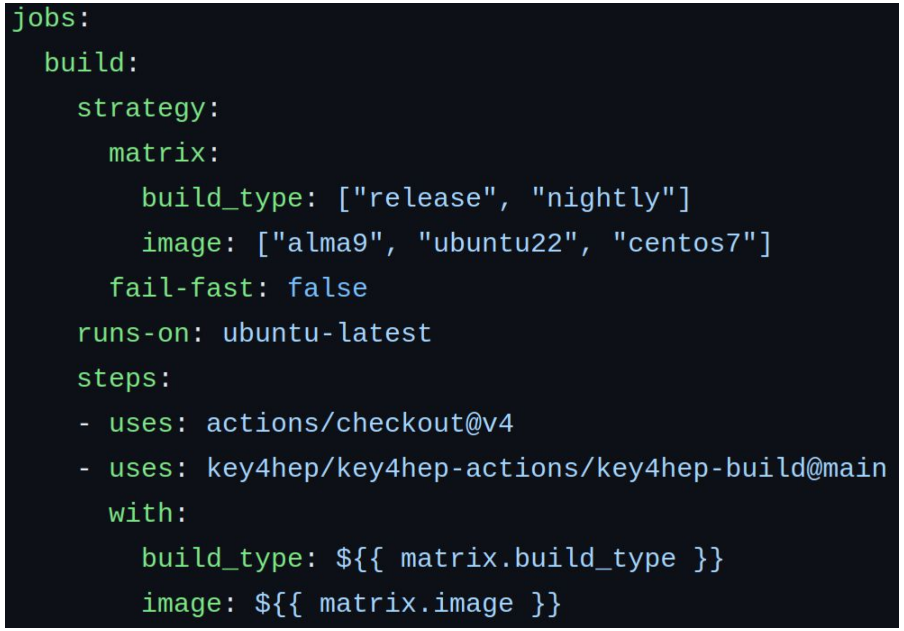
Key4hep: Turnkey Software for Future Colliders

https://cern.ch/key4hep

Pinned

k4FWCore Public
Core Components for the Gaudi-based Key4HEP Framework
C++ ☆ 3 13

EDM4hep Public
Generic event data model for HEP collider experiments
C++ ☆ 9 11



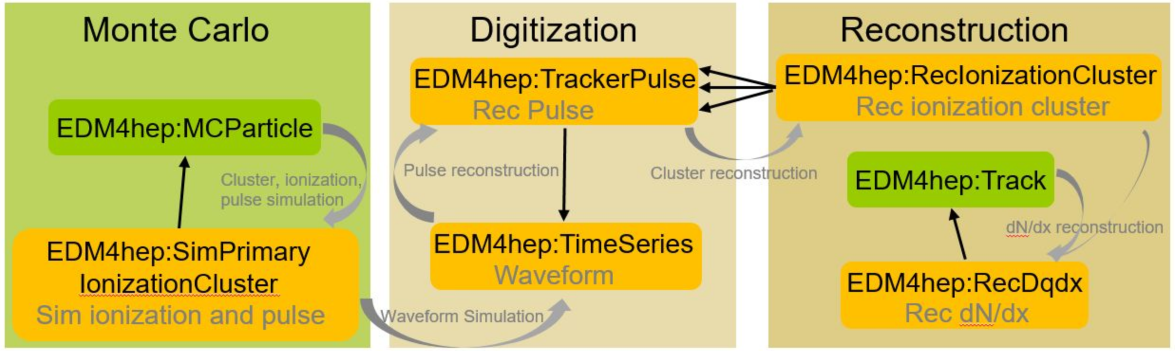
```
jobs:
  build:
    strategy:
      matrix:
        build_type: ["release", "nightly"]
        image: ["alma9", "ubuntu22", "centos7"]
      fail-fast: false
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v4
    - uses: key4hep/key4hep-actions/key4hep-build@main
      with:
        build_type: ${{ matrix.build_type }}
        image: ${{ matrix.image }}
```

Work at INFN

R&D study on framework to manage heterogeneous resources

- No applicant on second call for dedicated position
  - Despite trying to advertise it as widely as possible
- Competing against numerous other more attractive software/computing positions
  - Both money and career prospects
- Decision to not have a third call, but rather try to re-direct the funds
  - No final decision yet on where the allocation might go

Report from IHEP group



Monte Carlo

EDM4hep:MCParticle

Cluster, ionization, pulse simulation

EDM4hep:SimPrimaryIonizationCluster
Sim ionization and pulse

Digitization

EDM4hep:TrackerPulse
Rec Pulse

Pulse reconstruction

EDM4hep:TimeSeries
Waveform

Waveform Simulation

Reconstruction

EDM4hep:RecIonizationCluster
Rec ionization cluster

Cluster reconstruction

EDM4hep:Track

dN/dx reconstruction

EDM4hep:RecDqdx
Rec dN/dx

✅ Extend EDM4hep to support ongoing activities in CEPC

- Reconstruction of number of primary ionizations in drift chamber
- Analysis of test beam data for vertex detector prototype

✅ CEPC Vertex Detector tracking with ACTS

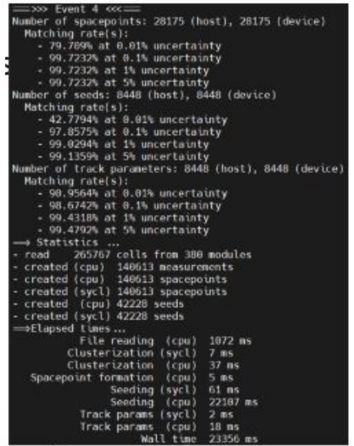
- Geometry in ACTS format (“manual” conversion from DD4hep)
- Use FATRAS and TRACCC for fast sim and tracking

✅ CEPC-on-Gaussino prototype

- Based on LHCb’s Gen/McEvent, Gaussino, CEPCS
- Geometry via DD4hepCnvSvc

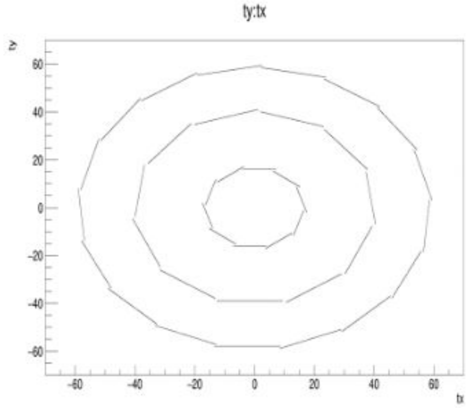
👷 Optimize data flow / communication between CPU and GPU. Investigate possibility of avoiding copies by building a connection between EDM4hep and TRACCCs vecmem

👷 Use EDM4hep for Gaussino



```
══>>> Event 4 <<<══
Number of spacepoints: 28175 (host), 28175 (device)
  Matching rate(s):
    - 79.789% at 0.01% uncertainty
    - 99.7232% at 0.1% uncertainty
    - 99.7232% at 1% uncertainty
    - 99.7232% at 5% uncertainty
Number of seeds: 8448 (host), 8448 (device)
  Matching rate(s):
    - 42.7794% at 0.01% uncertainty
    - 97.8575% at 0.1% uncertainty
    - 99.0294% at 1% uncertainty
    - 99.1359% at 5% uncertainty
Number of track parameters: 8448 (host), 8448 (device)
  Matching rate(s):
    - 90.9564% at 0.01% uncertainty
    - 98.6742% at 0.1% uncertainty
    - 99.4318% at 1% uncertainty
    - 99.4792% at 5% uncertainty
──> Statistics ...
- read    265767 cells from 380 modules
- created (cpu)  140613 measurements
- created (cpu)  140613 spacepoints
- created (sycl) 140613 spacepoints
- created  (cpu) 42228 seeds
- created (sycl) 42228 seeds
══>Elapsed times ...
              File reading  (cpu)  1072 ms
            Clusterization (sycl)  7 ms
            Clusterization  (cpu)  37 ms
     Spacepoint formation  (cpu)  5 ms
                   Seeding (sycl)  61 ms
                   Seeding  (cpu)  22187 ms
              Track params (sycl)  2 ms
              Track params  (cpu)  18 ms
                     Wall time  23356 ms
```

Computing performance of TRACCC
CPU: Intel Xeon Silver 4214 @ 2.20GHz
GPU: NVIDIA Quadro RTX 8000

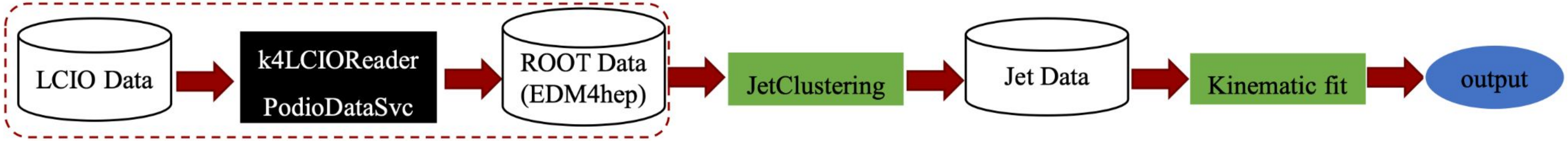


Visualization of vertex detector
(single sided)

Report from SDU group

✅ Development of an analysis toolkit based on RDataFrame

- Port of several existing tools (MarlinKinFit, JetClustering, …)



✅ Further development of Valprod toolkit to run validation

- Running CEPC CI jobs on grid

👷 Integrate more missing analysis tools to RDataFrame

👷 Continue development of validation system

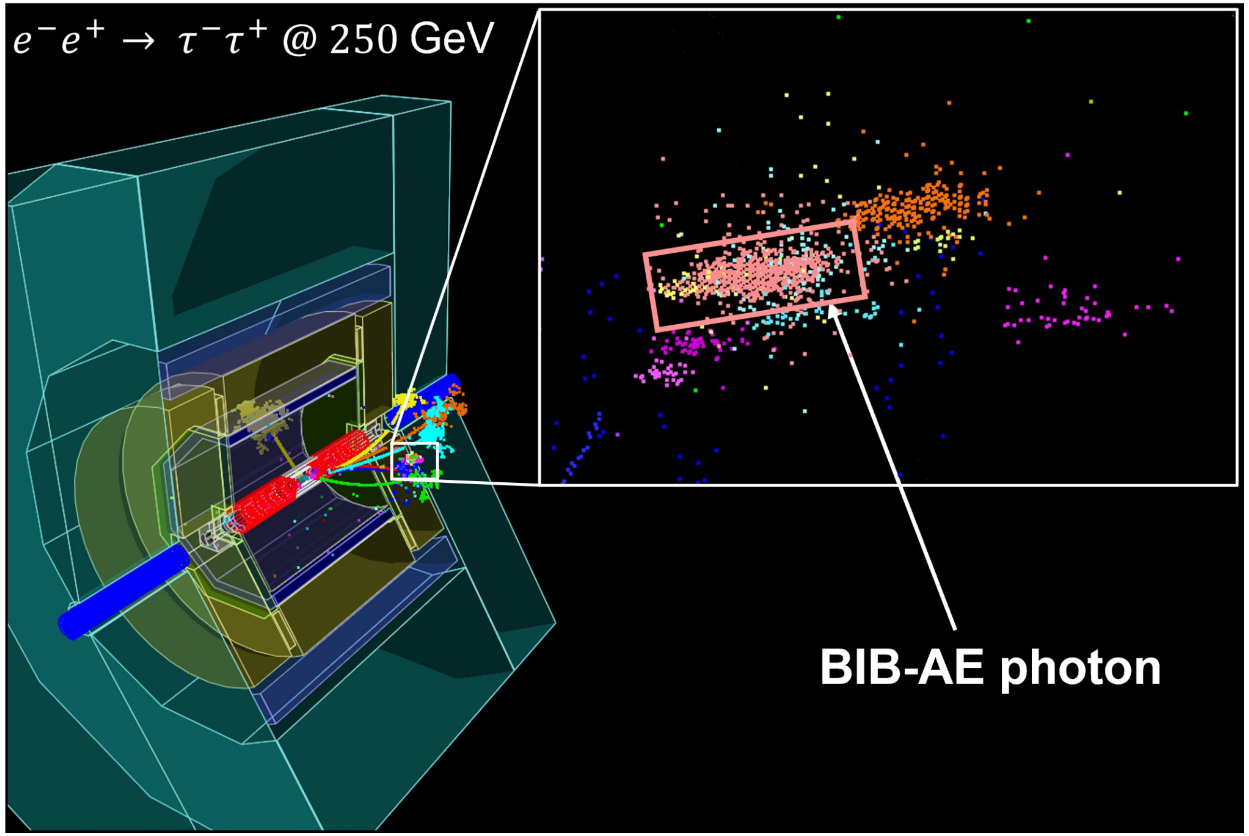
- Generalize grid resource usage possibilities

Interactions with other WPs

Integration of R&D into the software stack

- **Successfully ran simulation (ddsim) and standard reconstruction for ILD using the BIB-AE model for fast shower generation**
  - Part of the developments during last years hackathon
- ACTS integration ongoing
  - Stopped by need for TrackerHit interface in EDM4hep until recently
- PandoraPFA integration
  - Available through MarlinWrapper
  - See Swathis talk

courtesy P. McKeown



Summary / Outlook for the next year

- Finalized stable version of podio
  - Minor cleanup ongoing
- Finalizing EDM4hep v1.0
  - Some conceptual changes wrt LCIO (still ongoing)
  - Improved converter functionality
- First version of Gaudi Functional capable core components
  - Using podio functionality (no longer re-implementing readers / writers)
  - Necessary changes mostly transparent, few changes necessary for k4MarlinWrapper
  - Multi-threading support ongoing
- Integration of ML FastSim in DDML
  - Next steps: Add more models
- Tutorials and updated documentation
- Keep on working integrating more software tools and using them in simulation, reconstruction & analysis
  - Try to homogenize the approaches and avoid re-developing the same thing multiple times

Thank you