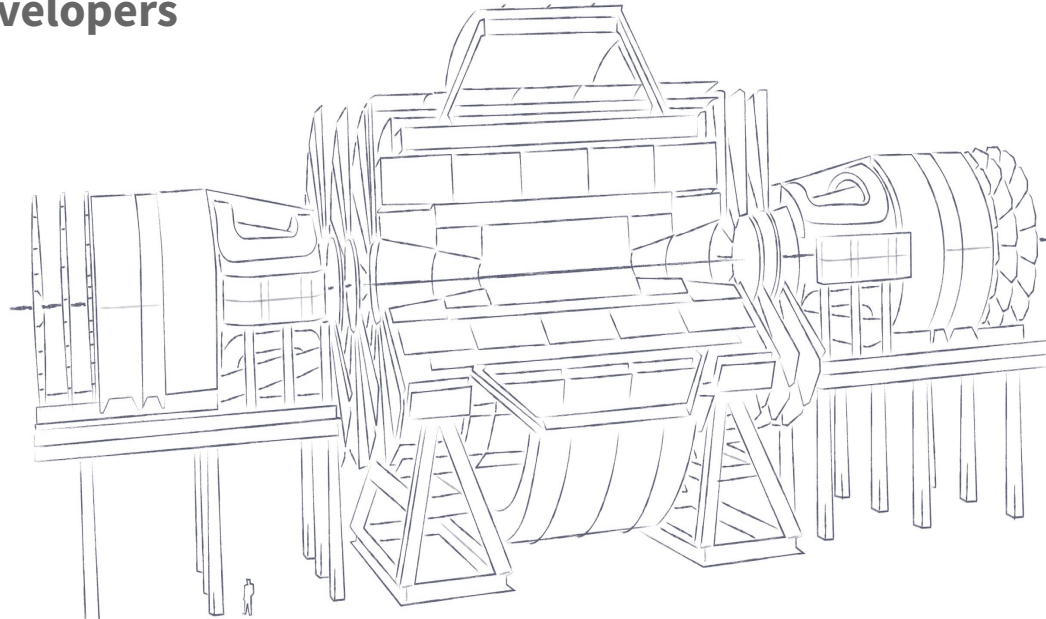


Status of Key4hep

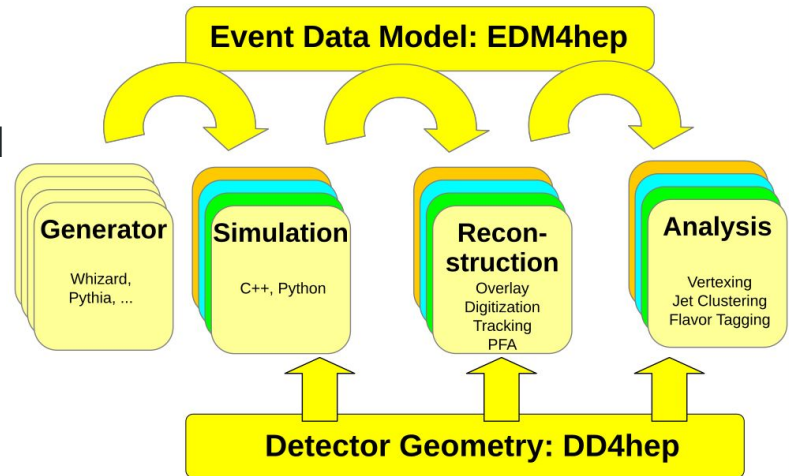
Benedikt Hegner
for the Key4hep developers

ACAT 2024
Stony Brook
13 March 2024



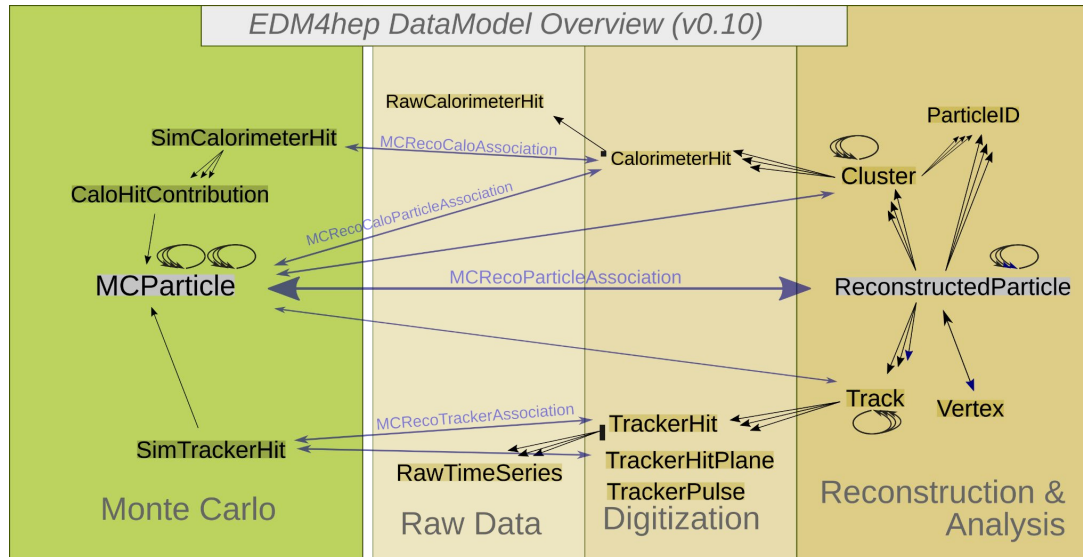
Key4hep

- Develop common software solutions for future collider projects
 - ILC, CLIC, FCC-ee & FCC-hh, CEPC, Muon Collider, EIC, ...
- Reduce maintenance and development cost and allow everyone to benefit from its improvements
- Investing into what really is essential for combining software building blocks into an application
 - Common data model
 - Easy to use geometries



The EDM at the core of HEP software

- Different components of HEP experiment software have to talk to each other
- The event data model defines the language for this communication
- Users express their ideas in the same language



Update: Learning from the past

- In the first phase of Key4hep the aim was to be as inclusive as possible to all communities already doing detector studies
 - Get a running system quickly to gain experience...
 - ... at the price of picking up some legacy initially
- Some of this legacy can be tackled piece by piece
 - E.g. thread-safety of existing code and transforming them into *Gaudi Functional*
- Others are disruptive
 - E.g. change of data models

```
struct ExampleFunctionalConsumer final
    : Gaudi::Functional::Consumer<void(const edm4hep::MCParticleCollection& input), BaseClass_t> {
    ExampleFunctionalConsumer(const std::string& name, ISvcLocator* svcLoc)
        : Consumer(name, svcLoc, KeyValue("InputCollection", "MCParticles")) {}

    void operator()(const edm4hep::MCParticleCollection& input) const override {
        // Do something with the input
    }
};
```

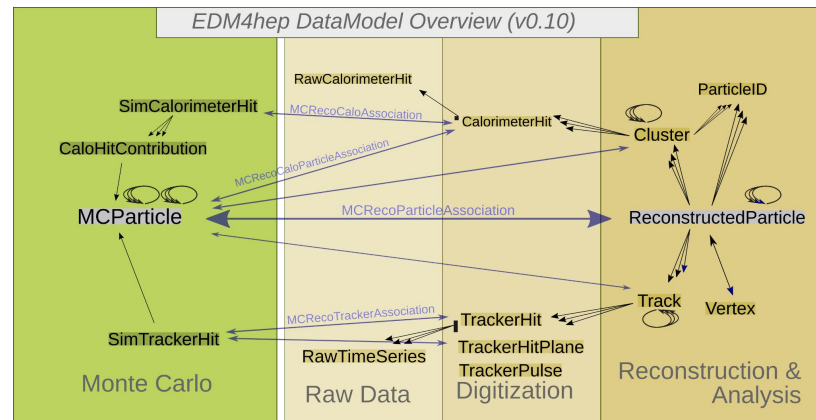
Decided to break now EDM4hep compatibility before moving to next phase of the Key4hep project

⇒ Schema evolution and helper tools will help in the transition

An example of backwards incompatibility

- One of EDM4hep's biggest inspiration LCIO assumed mutability of objects
 - E.g. Particle ID could be attached after the fact
 - *An absolute no-go in multithreaded environments!*
- The EDM4hep data model discouraged, but did not really drop this old behaviour
 - New code was written in the “proper” way
 - Now old case has to be fixed

Now providing more intuitive and maintainable code for Key4hep user base



RNtuple integration for EDM4hep / PODIO

TTree based

```
ROOTFrameWriter writer(filename);  
writer.writeFrame(frame);  
writer.finish();
```

```
ROOTFrameReader reader{};  
reader.openFile(filename)  
auto event = podio::Frame(reader.readEntry("events", 0));
```

RNTuple based

```
ROOTNTupleWriter writer(filename);  
writer.writeFrame(frame);  
writer.finish();
```

```
ROOTNTupleReader reader{};  
reader.openFile(filename)  
auto event = podio::Frame(reader.readEntry("events", 0));
```

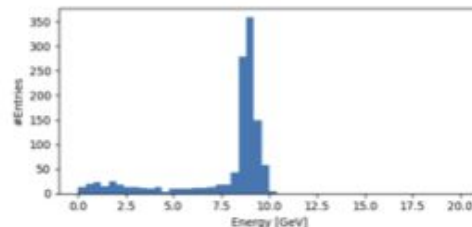
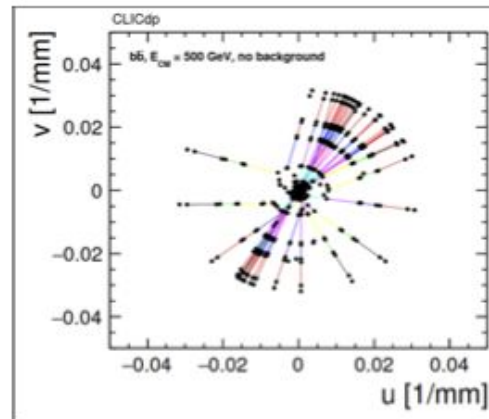
On the todo list:

- Performance comparison between backends (reading and writing speed, size)
- Improved C++ interface to make runtime switching easier

More integration of native algorithms

Significant progress with the development of native tools and the integration of other tools

- ACTS in Key4hep
- Pandora Particle Flow
- More code from ILCSoft and FCC ported upstream



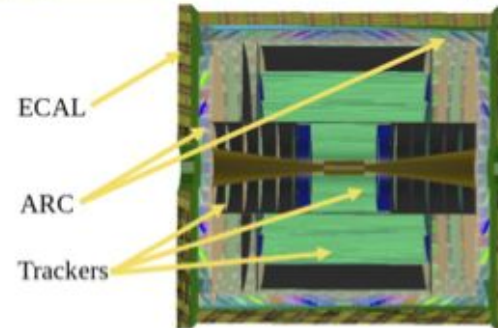
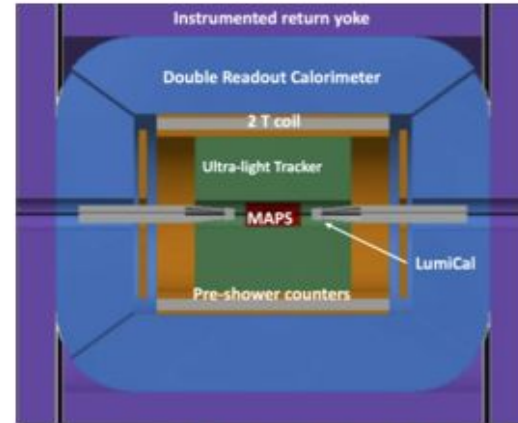
Based on the Pandora PFA

Detector and Reconstruction studies

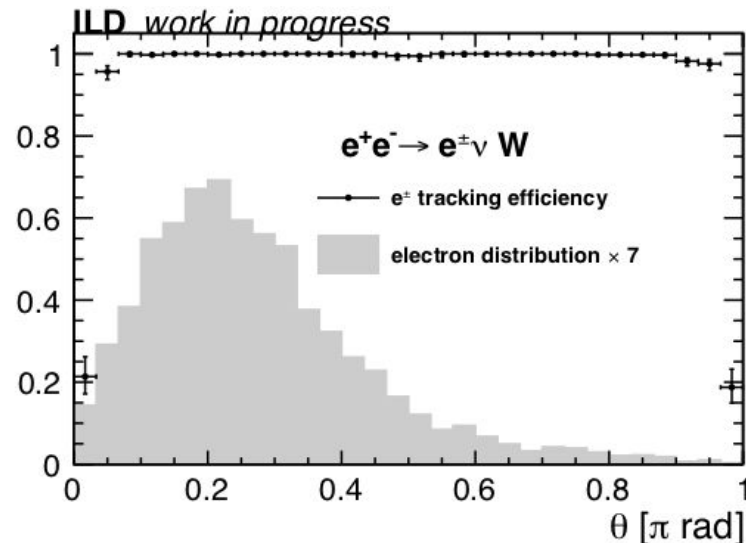
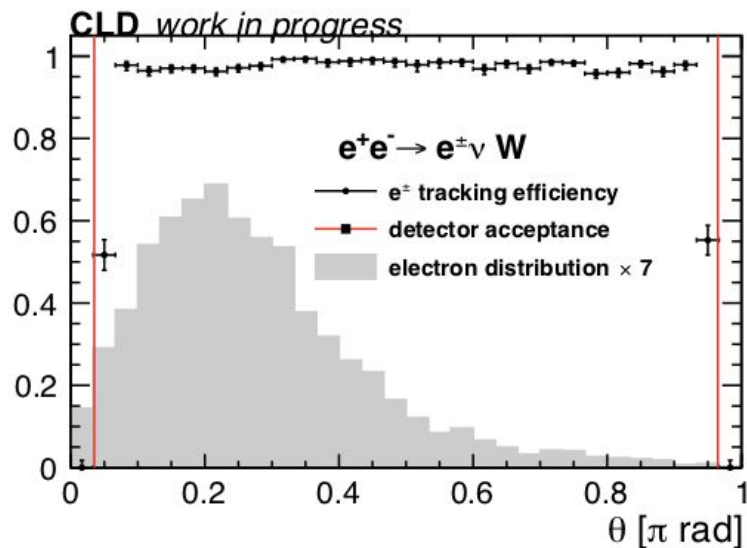
New detectors have been added recently to k4geo, some migrated from FCC:

- IDEA
- ALLEGRO
- CLD with ARC subdetector
- IDEA Vertex Detector

The setup of Key4hep allows one to swap the detector with only a small effort



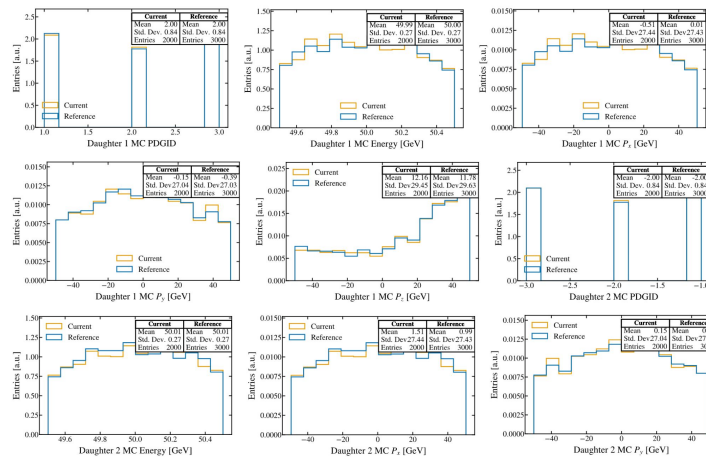
Physics studies - different detectors, shared code



Key4hep validation

Check simulation and reconstruction chain

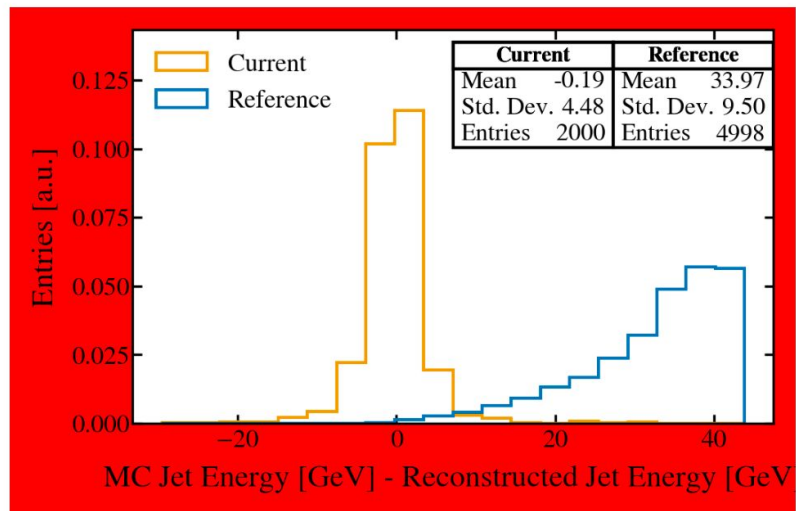
- Run daily on nightlies
 - entire chain from sim to final plots
 - compare with references
- Results published daily
 - <https://key4hep-validation.web.cern.ch/>
- As always - work in progress
 - Coverage continuously extended



Key4hep validation

Check simulation and reconstruction chain

- Run daily on nightlies
 - entire chain from sim to final plots
 - compare with references
- Results published daily
 - <https://key4hep-validation.web.cern.ch/>
- As always - work in progress
 - Coverage continuously extended

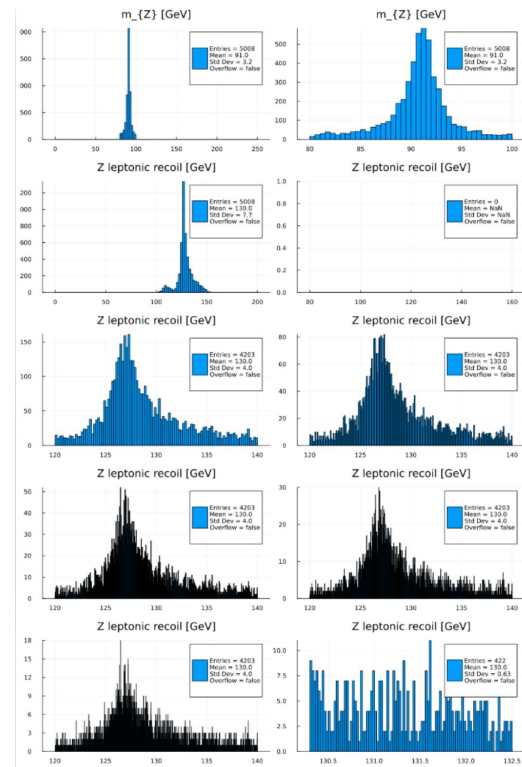


Example of a bug in the *reference*

Exploratory developments - Julia

- Julia is a powerful language with a potential to solve the usability vs. speed dilemma we are facing with Python vs. C++
- Ongoing work to make EDM4hep available in Julia
 - Generate Julia code based on data description of EDM4hep in YAML files - similar to what is done for the C++ interface
 - UnROOT as I/O backend
- Able to re-play some FCC analyses purely in Julia
- More details in

<https://github.com/peremato/EDM4hep.jl/blob/main/docs/src/index.md>



Summary

- Lots of progress in Key4hep in various areas
- Key4hep supports / is used by multiple communities
 - Leading to a significant extension of features provided
- Key4hep is a proof that a **software stack can be used across communities**
 - Even going beyond individual libraries
- Software is maturing and a huge emphasis on quality...
 - ... while following future trends as well

Pointers to software resources

- EDM4hep
 - <https://github.com/key4hep/EDM4hep>
- podio
 - <https://github.com/AIDAsoft/podio>
- Key4hep
 - <https://github.com/key4hep>
 - <https://key4hep.web.cern.ch>



xkcd.com/138