

# **Key4hep: Status and Plans**

André Sailer for the Key4hep Developers

CERN-EP-SFT

CEPC Workshop October 28, 2022

#### **Table of Contents**



Key4hep Software Stack Ingredients and Developments Summary

# Key4hep Software Stack

### Key4hep: Turnkey Software Stack

Create a software stack that connects and extends individual packages towards a complete data processing framework for detector studies with fast or full simulation, reconstruction, and for analysis

- Major ingredients: Event Data Model (EDM), Geometry Information, Processing Framework
- Sharing common components reduces overhead for all users
- Should be easy to use for librarians, developers, users
  - easy to deploy, extend, set up
- Full of functionality: plenty of examples for simulation and reconstruction of detectors
- Preserve and adapt existing functionality into the stack, e.g., from iLCSoft, FCCSW, CEPCSW





## Ingredients and Developments

#### A. Sailer

### The Key4hep EDM: EDM4hep



- Using podio to manage the EDM (described by yaml) and easily change the persistency layer (ROOT, SIO, ...)
- http://github.com/key4hep/edm4hep
- Recent developments for podio or EDM4hep
  - EDM4hep: additional types, associations; various tweaks to types many triggered by CEPC requirements
  - podio: podio-dump tool; EDM extensions; a thread-safe data container(*Frame*); Schema Evolution's first PR





### Building Key4hep Stacks with Spack



#### Mostly ongoing maintenance work

- Continuous deployment to CVMFS /cvmfs/sw-nightlies.hsf.org/key4hep
- Regular releases /cvmfs/sw.hsf.org/key4hep
- GitLab pipeline to build multiple PRs together available (manually triggered)
- Scaling up to more compilers etc.
- Test builds with newer compilers, but keep older version as default
- "Spack for developers" not so straightforward investigate spack-batteries-included as a pre-configured option



#### EDM4hep Output for Guinea-Pig





- Guinea-Pig Beam–Beam simulation used by all high energy lepton colliders
- Classical ASCII output storing minimal information
  - Conversions needed for, e.g., simulations
- Recently implemented EDM4hep output
  - Output contains background particles as MCParticles
  - Conversions done at the source
  - Particles can be directly fed to simulation



- Support for Fast or Full Simulation with standalone programs or framework integration
- Fast Simulation with Delphes paremeterised simulation
- Geant4 with ddsim or k4SimGeant
  - On going investigation to adapt LHCb's Gaussino for Gaudi-integrated Geant4 simulation of DD4hep based geometries
    - Including interfaces to MC generators (similar to k4Gen)





- Support for Fast or Full Simulation with standalone programs or framework integration
- Fast Simulation with Delphes paremeterised simulation
- Geant4 with ddsim or k4SimGeant
  - On going investigation to adapt LHCb's Gaussino for Gaudi-integrated Geant4 simulation of DD4hep based geometries
    - Including interfaces to MC generators (similar to k4Gen)





- Support for Fast or Full Simulation with standalone programs or framework integration
- Fast Simulation with Delphes paremeterised simulation
- Geant4 with ddsim or k4SimGeant
  - On going investigation to adapt LHCb's Gaussino for Gaudi-integrated Geant4 simulation of DD4hep based geometries
    - Including interfaces to MC generators (similar to k4Gen)





- Support for Fast or Full Simulation with standalone programs or *framework* integration
- Fast Simulation with Delphes paremeterised simulation
- Geant4 with ddsim or k4SimGeant
  - On going investigation to adapt LHCb's Gaussino for Gaudi-integrated Geant4 simulation of DD4hep based geometries
    - Including interfaces to MC generators (similar to k4Gen)





- Support for Fast or Full Simulation with standalone programs or framework integration
- Fast Simulation with Delphes paremeterised simulation
- Geant4 with ddsim or k4SimGeant
  - On going investigation to adapt LHCb's Gaussino for Gaudi-integrated Geant4 simulation of DD4hep based geometries
    - Including interfaces to MC generators (similar to k4Gen)



### k4Clue



- Investigating use of the GPU friendly algorithm <u>CLUE</u> (CLUstering of Energy) as part of particle flow reconstruction
- CLUE Gaudi algorithm created: <u>k4Clue</u> and run as part of the CLIC reconstruction chain
- Validation and use of the clusters pending
- k4CLUE successfully used to reconstruct ECal Clusters in Barrel and Endcap



## k4ACTS



#### ACTS Tracking Software integration to Key4hep

- Now in DD4hep: VariantParameters extension:
  - General purpose parameter map extension that can be attached to any detector element.
  - Break out of dependency hell between ACTS and DD4hep
  - Attach information needed by ACTS at will, via DD4hep's extension mechanism (DD4hep\_ParametersPlugin)
- Expand the use of the OpenDataDetector for generic key4hep examples!
- Reconstruction to be implemented next

#### <lccdd>

```
<detectors>
  <detector id="ODD Solenoid ID" name="Solenoid"</pre>
            type="ODDCylinder"
            beampipe="false" vis="Aluminum">
    <type flags type="DetType TRACKER" />
    <boundary_material surface="inner"
                       binning="binPhi.binZ"
                       bins0="mat sol bPhi"
                       bins1="mat_sol_bZ"/>
    <tubs name="Solenoid" rmin="sol rmin"
          rmax="sol rmax" dz="sol hlength"
          material="Aluminum">
      <laver material surface="representing"
                      binning="binPhi.binZ"
                      bins0="mat sol bPhi"
                      bins1="mat sol bZ"/>
```

```
</tubs></detector></detectors>
```

```
<plugins>
  <plugin name="DD4hep_ParametersPlugin">
      <argument value="Solenoid"/>
      <argument value="passive_layer: bool = true"/>
      </plugin>
</plugins>
</lead>
```

#### **Phoenix Event Display**



- Phoenix: a browser based event display developed by ATLAS
- Adaptation to Key4hep environment by J. Smiesko
- EDM4hep to json converter for event data
- A number of manual steps to get from DD4hep geometry to the event display, but Nicely Documented





### Summary



- A lot of developments in the plumbing of Key4hep:
  - Preparations for better multi-threaded processing
  - Spack build system maintenance, improvements and up-scaling
  - Event Data Model refinements and Quality-of-Life improvements
- Integration of new features: reconstruction, event display
- Important investments for future maintainability and synergies!

Thanks to Valentin Volkl, Placido Fernandez, Erica Brondolin, Juraj Smiesko, Paul Gessinger for material.

Thank you for your attention!