KEY4HEP & EDM4HEP - Common Software for Future Colliders

EP-R&D Monthly Meeting - 23.09.2020

Valentin Volkl (CERN), Placido Fernandez (CERN), Andre Sailer (CERN)

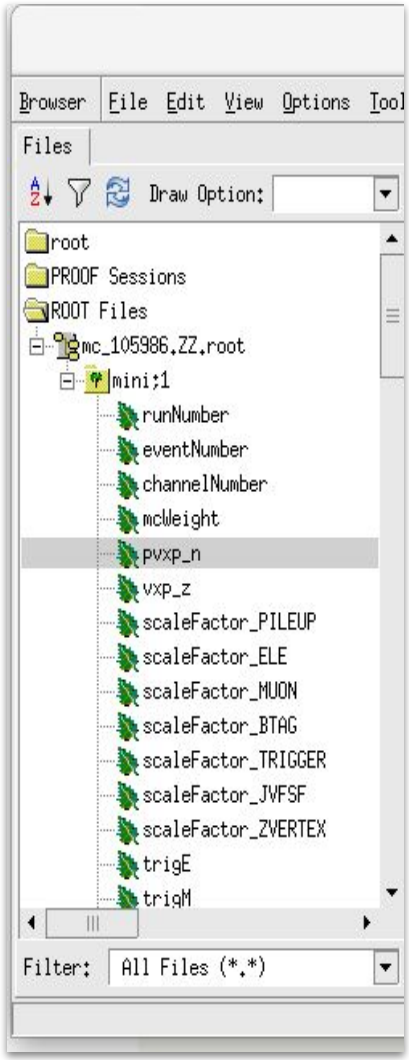
Table of Contents

Key4HEP Motivation

- Future detector studies critically rely on **well-maintained software stacks** to model detector concepts and to understand a detector's limitations and physics reach
- We have a scattered landscape of **specific software tools** on the one hand and **integrated frameworks** tailored for a specific experiment on the other hand
- Aim at a low-maintenance common stack for FCC, ILC/CLIC, CEPC with ready to use "plug-ins" to develop detector concepts
- Reached consensus among all communities for future colliders to develop a **common turnkey software stack** at recent Future Collider Software Workshop
- Identified as an important project in the CERN EP R&D initiative

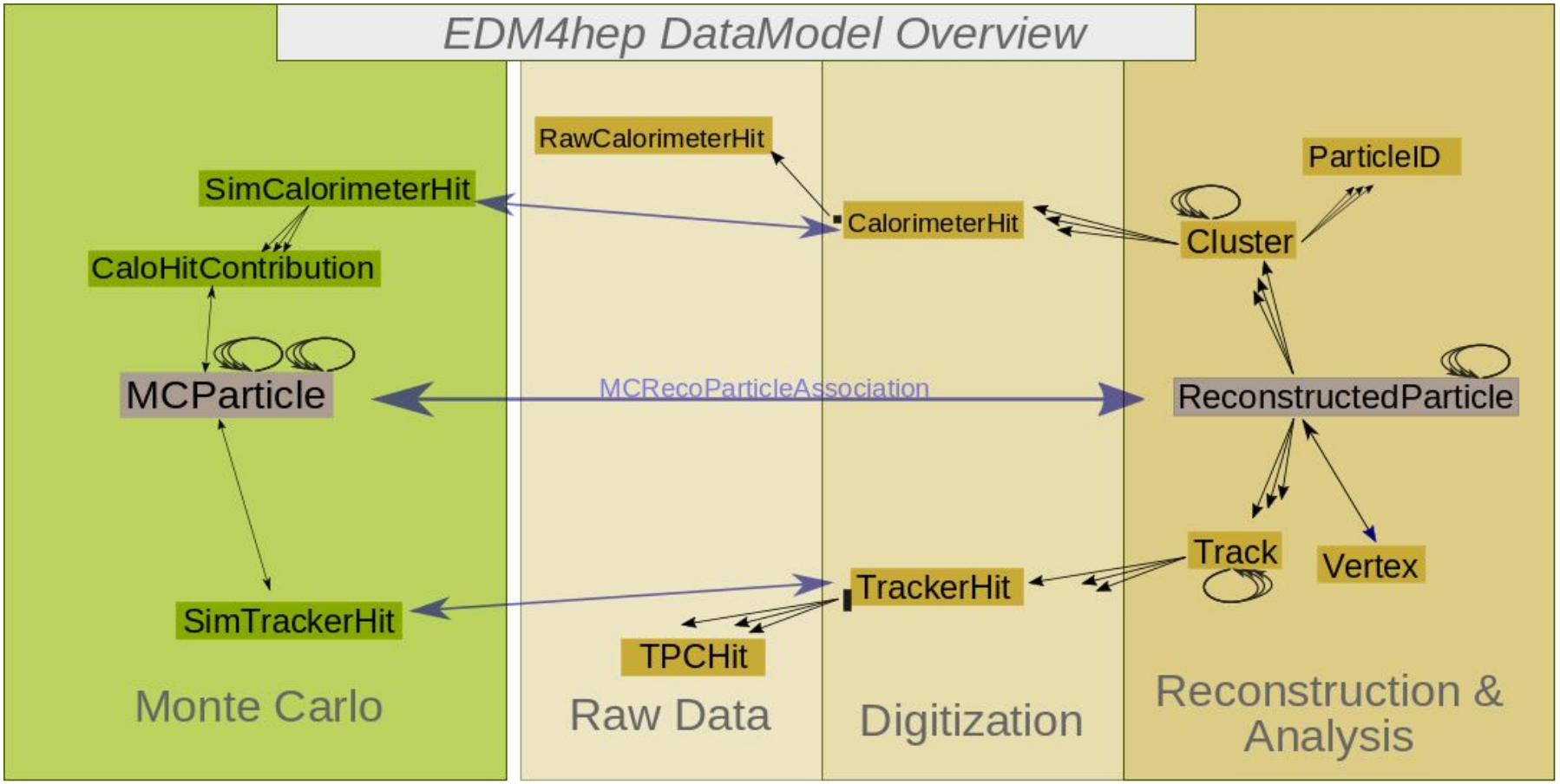
EDM4HEP - Introduction

- Event Data Model:
  - Describes structure of HEP Data:
    - definitions of **objects** and how they are **grouped**
    - **technical** implementation of persistency and processing
- Can be as simple as “Branch names in ROOT file”
  - But more sophisticated solutions can:
    - provide an **application programming interface** for HEP software
    - aid developers in writing more **efficient code**
    - enable **collaboration**



Relation Diagram

Code Reference under https://cern.ch/edm4hep

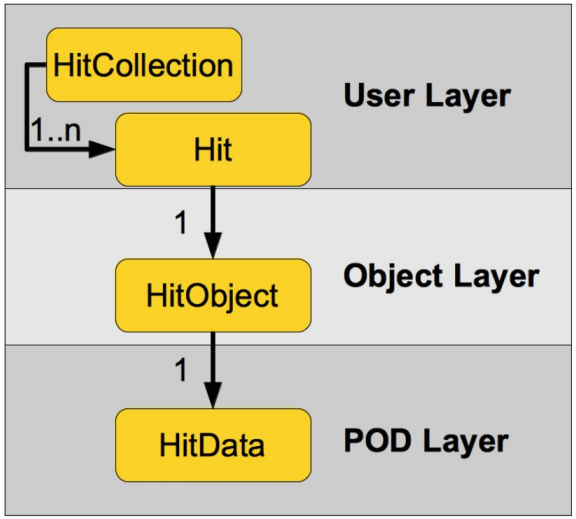


Technical: PODIO

- PODIO is an Event Data Model *toolkit* for HEP
  - developed in the Horizon2020 project AIDA2020
  - based on the use of **PODs** for the event data objects (Plain Old Data objects)
- PODIO originally developed in the context of the FCC study
  - addressing the problem of creating an EDM in a generic way
  - EDM described in yaml, C++ code auto-generated
  - Allowing potential re-use by other HEP groups
- PODIO is used since its first release by the FCC studies

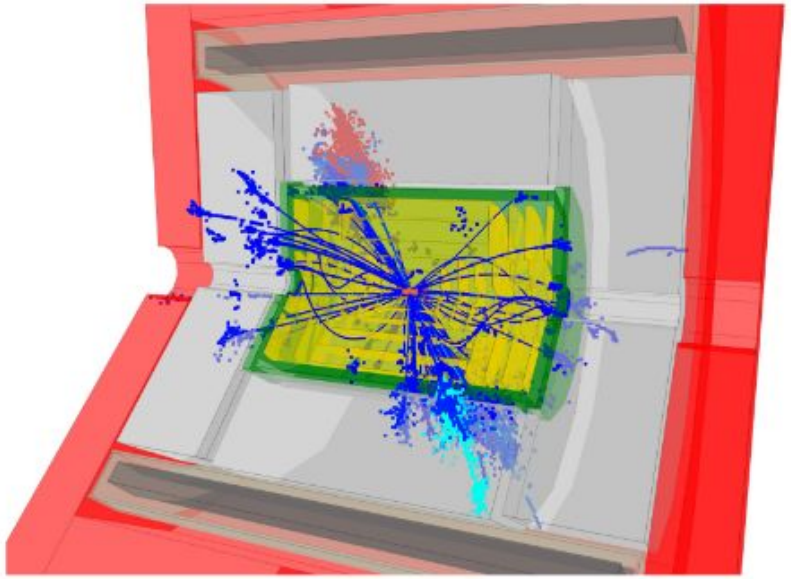
Recent/Ongoing Developments:

- **Addition of Metadata**
- **Template engine based on Jinja2**
- **Another backend (SIO)**
- **Improved RDataFrame Interface**



Applications: DDSim

DDSim (Standalone Geant4 simulation tool in dd4hep) can now produce edm4hep files



EDM4HEP plugin:

```
source /cvmfs/sw.hsf.org/key4hep/setup.sh

ddsim --compactFile
/cvmfs/sw-nightlies.hsf.org/key4hep/views/latest/x86_64-cent
os7-gcc8-opt/DDDetectors/compact/SiD.xml -N 10 -G
--gun.particle pi+ --outputFile my_edm4hep.root
--part.userParticleHandler=''
```

Application: DelphesEDM4HEP

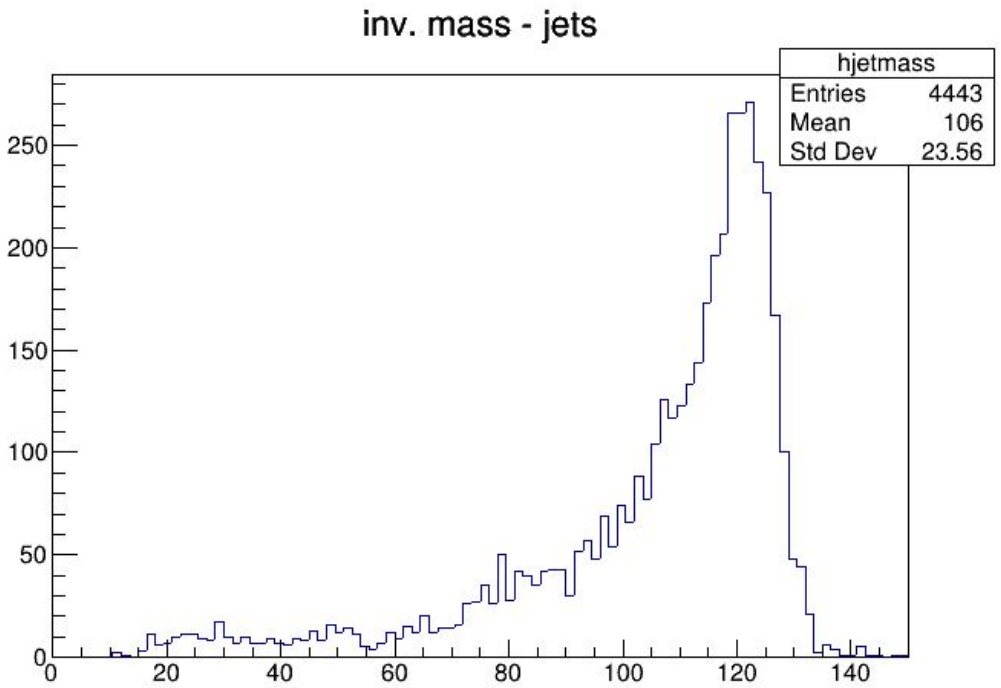
Second Plugin for Delphes output recently added:

```
DelphesPythia8_EDM4HEP    DelphesHepMC_EDM4HEP
DelphesSTDHEP_EDM4HEP  DelphesROOT_EDM4HEP
```

Adds executables like standard Delphes, outputting directly to EDM4HEP.

Higgs Recoil Analysis

example: Link



Key4HEP Framework

Meanwhile, developments on core functionality of the Gaudi-based framework:

- K4FWCore:
  - Data Service for Podio Collections
  - Overlay for backgrounds
  - https://github.com/key4hep/K4FWCore
- K4-project-template
  - Template repository showing how to build new components on top of the core Key4HEP framework
  - https://github.com/key4hep/k4-project-template

- Ongoing Work to collaborate more with Gaudi ecosystem (Gaussino)
- Add ACTS components

Key4HEP Framework

Meanwhile, developments on core functionality of the Gaudi-based framework:

- K4FWCore:
  - Data Service for Podio Collections
  - Overlay for backgrounds
  - https://github.com/key4hep/K4FWCore
- K4-project-template
  - Template repository showing how to build new components on top of the core Key4HEP framework
  - https://github.com/key4hep/k4-project-template

- Ongoing Work to collaborate more with Gaudi ecosystem (Gaussino)
- Add ACTS components

FCCSW - Key4HEP transition

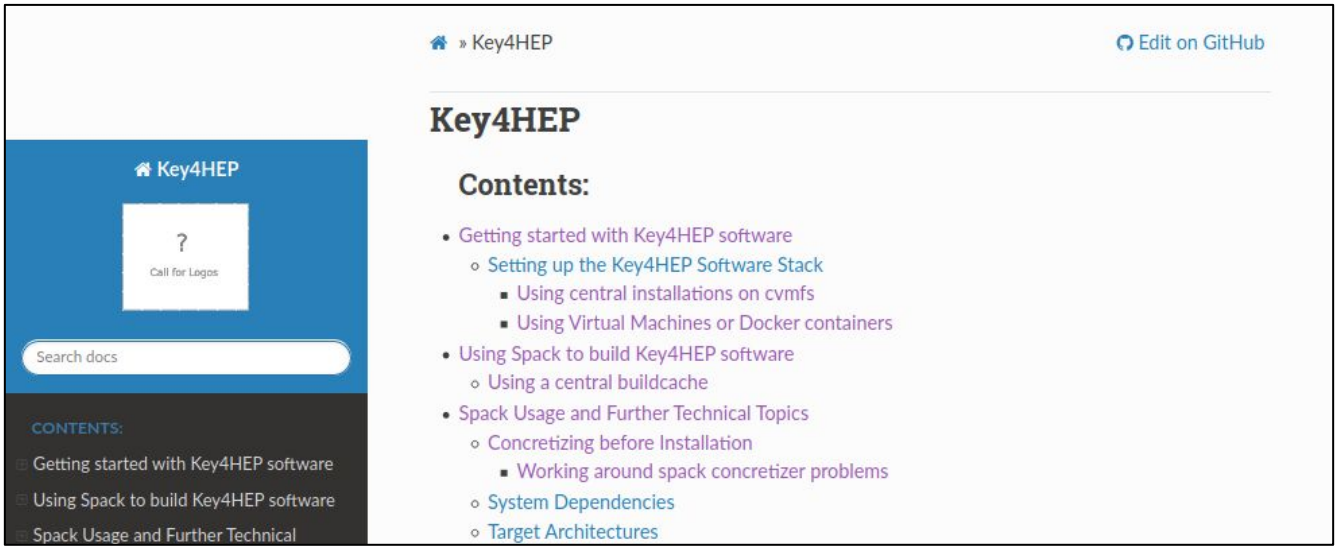
- Already Gaudi- and Podio based, so little technical challenges

```
gaudi_project(FCCSW  v0r13
              USE K4FWCore v0r1
              USE Gaudi v33r1 )
```

- Event model has a fairly straightforward correspondence
- Still: Many files need to be touched
  - Not yet clear if radical or soft (converter-based) update preferred
- Biggest hurdle: touching all components brings technical debts to light.

Software Infrastructure

- Regular meetings
  - https://indico.cern.ch/category/11461/
- Docpages
  - https://cern.ch/key4hep (main documentation site))
  - https://cern.ch/edm4hep (doxygen code reference)



- Modern CMake Configuration
- Automated Builds and Continuous Integration
  - Use of SPACK package manager
- Distribution via CVMFS

Spack for Key4HEP

- Spack is a package manager
  - Does not replace CMake, Autotools, …
  - Comparable to apt, yum, homebrew, ...
    - But not tied to operating system
    - And no central repository for binaries!
- Originally written for/by HPC community
  - Emphasis on dealing with **multiple configurations** of the same packages
    - Different versions, compilers, external library versions …
    - … may coexist on the same system
  - `Spec:` Syntax to describe package version configuration and dependencies
- Repository added with Key4HEP package recipes

```
git clone https://github.com/spack/spack.git
git clone https://github.com/key4hep/k4-spack.git
source spack/share/spack/setup-env.sh
spack repo add k4-spack
# install the meta-package for the key4hep-stack
spack install key4hep-stack
```

Some Experiences

- Collaboration with Spack developers fairly smooth
  - Some HEP colleagues have merging rights on the spack repo
  - Some HEP packages actively maintain their package recipes (ACTS!)
- Rapid pace of changes in upstream repository
  - Stable builds will need to pin the spack version used.
  - But miss out on the latest features.
- Spack developers very responsive, but roadmap sometimes a bit opaque:
  - The *concretizer* developments have been much delayed
- The recipes are very nice to persistify build system know-how

```
conflicts("%gcc@8.3.1",
     msg="There are known issues with compilers from redhat's devtoolsets" \
    "which are therefore not supported." \
    "See https://root-forum.cern.ch/t/devtoolset-gcc-toolset-compatibility/38286")
```

- Parallel builds not yet attempted

Spack: use for developers

- Use in developing software is pushing spack's intended purpose, but possible. Options:
  - Spack can build from branches.
  - Build can be done "as usual" after spack load / spack build-env
  - spack dev-build compiles local code according to the spack recipe
- Need to include build tools - would be nice to offload the build of these packages on LCG-releases

Towards the full LCG releases

- Ivan (SFT) has added a tremendous amount of packages - maybe 70% of packages included in the lcg releases already available in spack
- Key4HEP installation can be used as a test-bed

CVMFS directory tree

Already mounted in most places

```
/cvmfs/sw.hsf.org/key4hep/
|-- spackages /  $platform / $compiler / $pkgname-$spackhash / (bin  … )
|-- views    / $K4_version  / $platform / (bin include share … init.sh)
|-- setup.sh
|-- contrib

/cvmfs/sw-nightlies.hsf.org/key4hep/
|-- nightlies/ $timestamp / $platform / $pkgname-$spackhash / (bin  … )
|-- views    / $timestamp  / $platform / (bin include share … init.sh)
|-- setup.sh
|-- contrib
```

Used to test some new cvmfs features

CVMFS directory tree

```
/cvmfs/sw.hsf.org/key4hep/
|-- spackages /  $platform / $compiler / $pkgname-$spackhash / (bin  … )
|-- views    / $K4_version  / $platform / (bin include share … init.sh)
|-- setup.sh
|-- contrib

/cvmfs/sw-nightlies.hsf.org/key4hep/
|-- nightlies/ $timestamp / $platform / $pkgname-$spackhash / (bin  … )
|-- views    / $timestamp  / $platform / (bin include share … init.sh)
|-- setup.sh
|-- contrib
```

Contains some 300 packages

- 60 Experiment-specific
- 50 HEP-specific
- 200 System/General Purpose

14 GB install size, some 6h to build on single 4-core machine

GaudiMarlinWrapper -

See following presentation by Placido Fernandez

Conclusion

- Given the general agreement on moving to a common HEP software stackfrom future experiments
- Support joint developments between STC/SCT, and also FCC, ILC/CLIC,muon collider, CEPC
- Common detector geometry descriptions in DD4HEP
- Common event data model EDM4HEP
- Glue it all together with Gaudi in KEY4HEP

Complete Proto-LCG Stack with iLCSoft, FCCSW

hbase	gfal	astor	LCIO
oracle	dcap	atomicwrites	netcdf-c
pysqlite	pygsi	COOL	podio
sqlite	srm_ifce	CORAL	
		flatbuffers	
		Frontier	
		hdf5	

Status of LCG packages in Spack: Graphics

cairo	**graphviz**	**pydot**
cartopy	**harfbuzz**	**pydot_ng**
coin3d	**imagemagick**	**pygdal**
fontconfig	**pango**	**pygraphviz**
gdal	**pixman**	**pyproj**
gl2ps	**png**	**qt5**
gnuplot	**pycairo**	**tiff**

Status of LCG packages in Spack: Math

AIDA	BLAS	FASTJET	FFTW
GEOS	GSL	kiwisolver	libsvm
mpfi	mpfr	proj	sympy
Vc	vecgeom		

Status of LCG packages in Spack: XML

4suite	**defusedxml**	**Expat**	**libxml2**
libxlst	**lxml**	**xercesc**	**xquilla**
Geant4	**HepMC**	**HepMC3**	**HepPDT**

Status of LCG packages in Spack: Generators

agile	alpgen	ampt	baurmc
chaplin	collier	crmc	evtgen
feynhiggs	form	garfield++	gosam
gosam_contrib	heputils	herwig	herwig3
hijing	hoppet	hto4l	hydjet
hydjet++	jhu	jimmy	lhapdf

Status of LCG packages in Spack: Tools

acts	asn1crypto	astroid	autoconf
automake	autopep8	awkward	backports_abc
benchmark	bleach	Boost	Brotli
cachetools	caniusepython3	ccache	hcertifi
cffi	chardet	clhep	click
cloudpiclle	cmake	hsf-cmaketools	cmmnbuild
			lhapdf

Status of LCG packages in Spack: Tools

CouchDB	**coverage**	**cppgsl**	**cppunit**
cpymad	**cryptography**	**cycler**	**cython**
dask	**Davix**	**decorator**	**Delphes**
dill	**distlib**	**distributed**	**double_conversion**
doxygen	**eigen**	**elasticsearch**	**entrypoints**
fjcontrib	**flake8**	**fmt**	**fplll**
freetype	ftjam	future	gdb

Status of LCG packages in Spack: Tools

gflags	**glog**	**go_demangle**	**go_liner**
gomacro	**google_auth**	**gperf**	**gperftools**
gtest	**h5py**	**hadoop**	**HeapDict**
hepdata_converter	**hepdata_validator**	**hepmc3**	**hive**
html5lib	**idna**	**igprof**	**imageio**
ipaddress	**ipykernel**	**ipython**	**ipython_genutils**
ipywidgets	java	jedi	jemalloc

Status of LCG packages in Spack: Tools

jinja2	**glog**	**go_demangle**	**go_liner**
gomacro	**google_auth**	**gperf**	**gperftools**
gtest	**h5py**	**hadoop**	**HeapDict**
hepdata_converter	**hepdata_validator**	**hepmc3**	**hive**
html5lib	**idna**	**igprof**	**imageio**
ipaddress	**ipykernel**	**ipython**	**ipython_genutils**
ipywidgets	java	jedi	jemalloc