



Recent developments in Key4hep

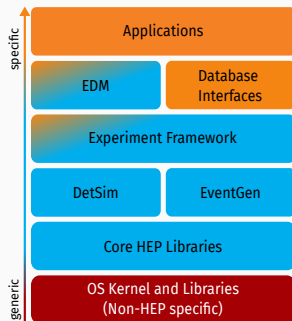


This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 101004761.

Thomas Madlener
20th Future Colliders@DESY
meeting
Jun 30, 2023

Key4hep - A (very) brief introduction

- Future detector studies rely on well maintained software for studying their potential
- Maintenance of a consistent HEP SW stack is non-trivial
 - Ecosystem of interacting components
- Sharing the burden allows everybody to reap the benefits
 - Make best use of scarce (human) resources
- **Regular contributions from ILC, CLIC, FCC, CEPC, EIC, (LUXE, MuonCollider, ...)**
- Support from major R&D initiatives
 - [CERN R&D for Future Experiments](#), [AIDAinnova WP12](#), ECFA



Key4hep goals

- Provide and maintain a consistent SW stack that allows to do physics studies for **all projects**
- Ensure interoperability of the necessary building blocks
- Reuse existing solutions where possible
 - A lot of experience from LHC experiments and LC communities
- Focus new developments on future collider specifics
- Share knowledge, processes, workflows and resources
 - Best practices, tutorials, documentation, ...

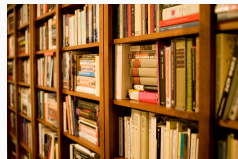
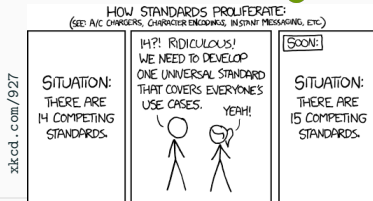


Photo by Stewart B. / CC-BY

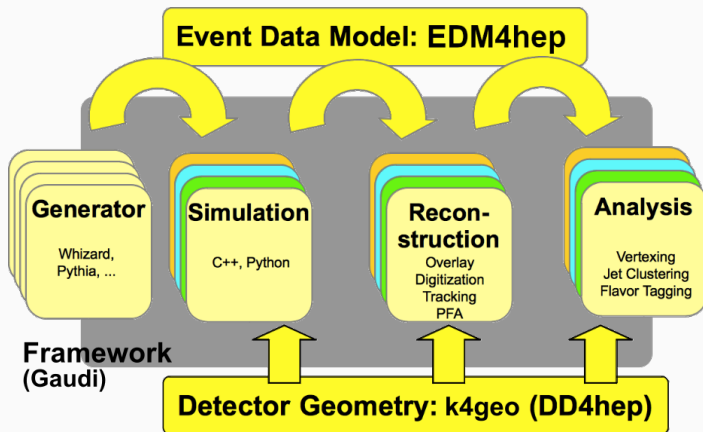


Non-goal

- Develop and maintain project specific software and workflows

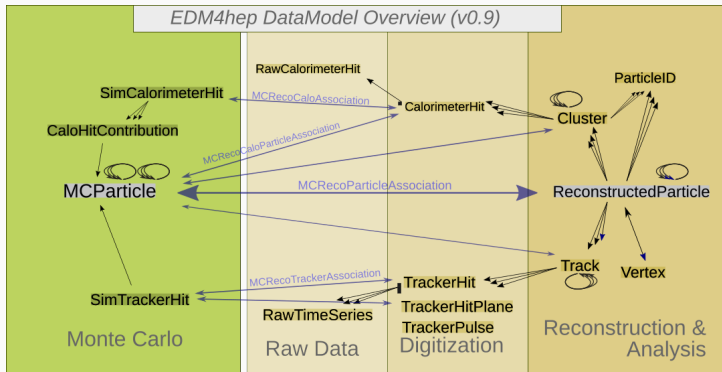


The general workflow and its core components



- **EDM4hep** - Common EDM enables interoperability of components
- **DD4hep** - Consistent geometry description in simulation & reconstruction
- **Gaudi** - Conducting all the parts

EDM4hep - The common EDM for Key4hep





- Based on LCI0 and FCC-edm
 - Focus on usability in analysis
- Quite stable over the last two years
- Addition of datatypes for CEPC drift chamber study
- Can easily be extended
 - Used by EDM4eic
- Generated via **podio**

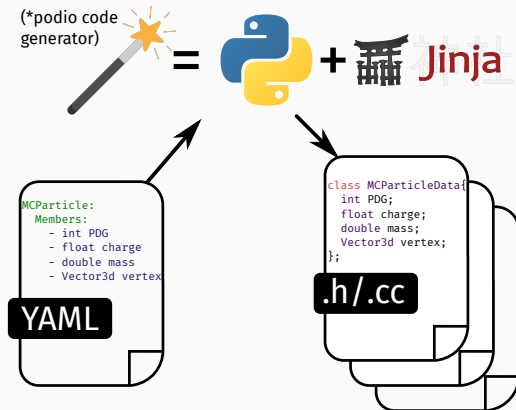
 [key4hep/EDM4hep](https://github.com/key4hep/EDM4hep)

edm4hep.web.cern.ch

 [AIDASoft/podio](https://github.com/AIDASoft/podio)

The podio EDM toolkit

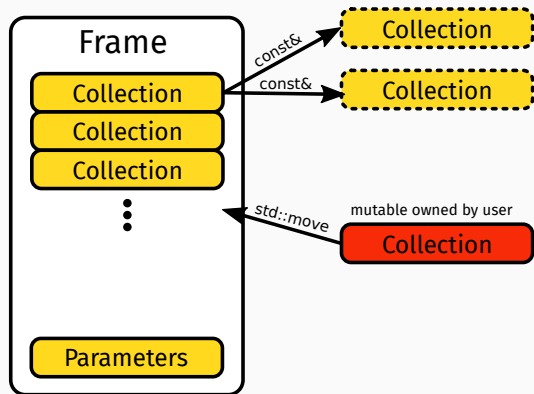
- Implementing a performant event data model (EDM) is non-trivial
- Use podio to generate code starting from a high level description
- Provide an easy to use interface to the users
- Main customers
 -  [key4hep/EDM4hep](https://github.com/key4hep/EDM4hep)
 -  [eic/EDM4eic](https://github.com/eic/EDM4eic)
- Finishing schema evolution for v1.0



 [AIDASoft/podio](https://github.com/AIDASoft/podio)

The `Frame` - A generalized (event) data container

- *Type erased* container aggregating all relevant data
- Defines an *interval of validity* / category for contained data
 - Event, Run, readout frame, ...
- Easy to use and thread safe interface for data access
 - Immutable read access only
 - Ownership model reflected in API
- Decouples I/O from operating on the data
- Replaces deprecated `EventStore`

















```
template<typename CollT>
const CollT& get(const std::string& name) const;

template<typename CollT, /*enable_if*/>
const CollT& put(CollT&& collection,
                const std::string& name);
```

Other recent developments

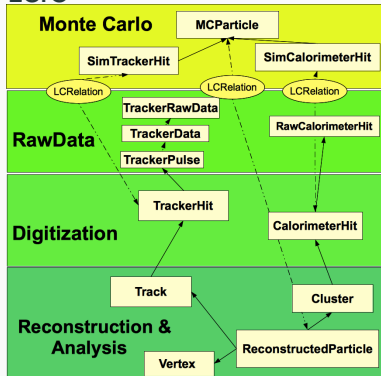
- More legible branch names for relations
- Stable collection IDs based on collection names
- Ongoing efforts to have [EDM4hep in coffea](#)
 - First version based on ILD DST files

```
OneToOneRelations:  
- edm4hep::Vertex          startVertex //s  
- edm4hep::ParticleID      particleIDUsed //p  
  
OneToManyRelations:  
- edm4hep::Cluster         clusters  
- edm4hep::Track           tracks  
- edm4hep::ReconstructedParticle particles  
- edm4hep::ParticleID      particleIDs
```

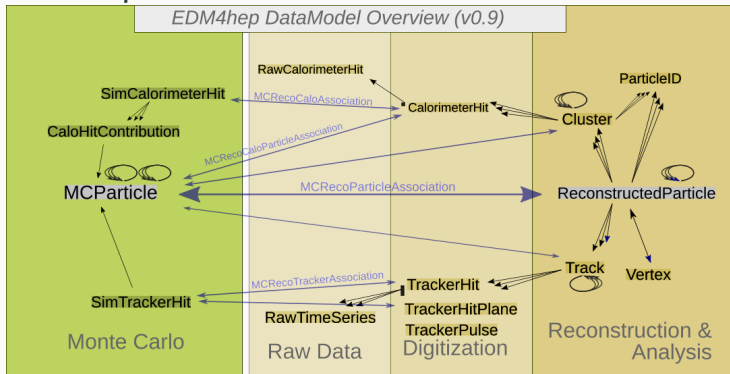
>  BCalRecoParticle old	>  BCalRecoParticle new
>  BCalRecoParticle#0	>  _BCalRecoParticle_clusters
>  BCalRecoParticle#1	>  _BCalRecoParticle_tracks
>  BCalRecoParticle#2	>  _BCalRecoParticle_particles
>  BCalRecoParticle#3	>  _BCalRecoParticle_particleIDs
>  BCalRecoParticle#4	>  _BCalRecoParticle_startVertex
>  BCalRecoParticle#5	>  _BCalRecoParticle_particleIDUsed

LCIO → EDM4hep converter reloaded

LCIO




EDM4hep



- Large existing data sets in LCIO format
- Very similar high level structure but some differences in details

LCIO → EDM4hep converter reloaded

- Complete overhaul of pre-existing functionality
 - Major effort from Finn Johannsen (DESY project student)
 - Shared library in  [key4hep/k4EDM4hep2LcioConv](https://github.com/key4hep/k4EDM4hep2LcioConv)
- Standalone executable (no Gaudi or Marlin!)

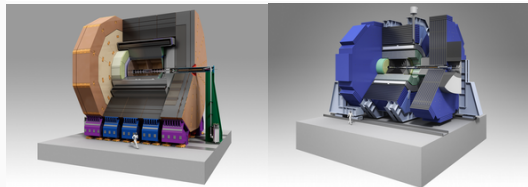
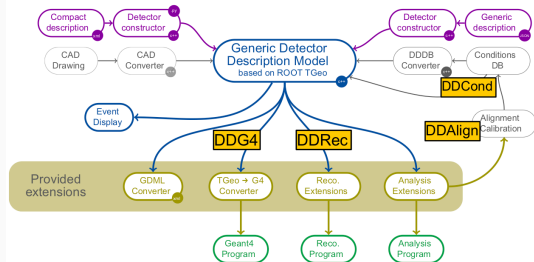
```
lcio2edm4hep input.slcio output.edm4hep.root
```

- For all details see [README](#)
 - Available in recent nightly builds
- Using the `podio::Frame`
- Support all features that are necessary for ILD


DD4hep - Detector description

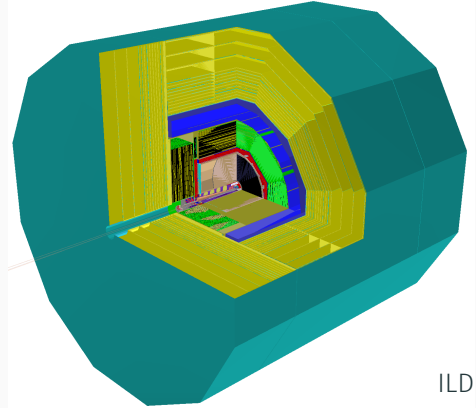
dd4hep.web.cern.ch

- Complete detector description
 - Geometry, materials, visualization, readout, alignment, calibration, ...
- From a **single source of information**
 - Simulation, reconstruction, analysis
- Comes with a powerful plug-in mechanism that allows customization
- More or less “industry standard” now
 - ILC, CLIC, FCC, CEPC, EIC, LHCb, CMS, ...
- `ddsim` - standalone simulation executable

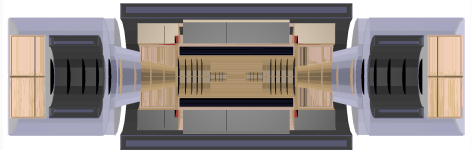


k4geo - The detector geometry repository

- iLCSoft/lcgeo → key4hep/k4geo
- Many existing detector models from LC studies
- Ongoing migration of detector concepts from  [HEP-FCC/FCCDetectors](https://github.com/HEP-FCC/FCCDetectors)
 - Noble liquid ECAL
- New ARC detector concept in CLD
- IDEA detector (work in progress)
- Goal: central repository for detector descriptions




ILD



FCC-hh

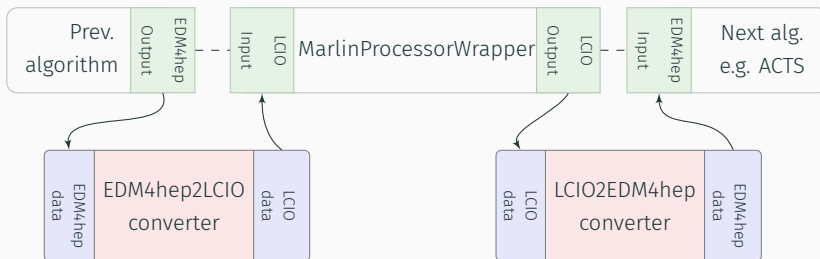
Experiment Framework

- **Gaudi**, originally developed by LHCb, now also used by ATLAS, FCCSW and smaller experiments
 - Supports concurrency
 - “Battle-proven” from data taking during LHC operations
- Key4hep has decided to adapt **Gaudi** as its experiment framework
 - Contribute to its development where necessary
- Integration and migration of iLCSoft algorithms into Key4hep with the help of a **Marlin**→**Gaudi** wrapper
 - Allows to use **Marlin** processors within the **Gaudi** framework
 -  [key4hep/k4MarlinWrapper](https://github.com/key4hep/k4MarlinWrapper)




k4MarlinWrapper

- Wraps **Marlin processor** in a Gaudi algorithm and allows to **run them unchanged**
- Automatic, on-the-fly conversion between LCIO and EDM4hep
 - Switch to new LCIO → EDM4hep converter ongoing
- Allows to “mix and match” existing reconstruction algorithms with new developments^(YMMV)



Frame based I/O in k4FWCore

-  [key4hep/k4FWCore](https://github.com/key4hep/k4FWCore) offers core Key4hep services for Gaudi
 - **Data service for podio generated EDMs**
 - Historically grown separate implementation
- Replaced custom Reader / Writer with podio provided ones
 - (Almost) completely transparent
- `podio::Frame` not visible to user
- Some usability improvements in the works

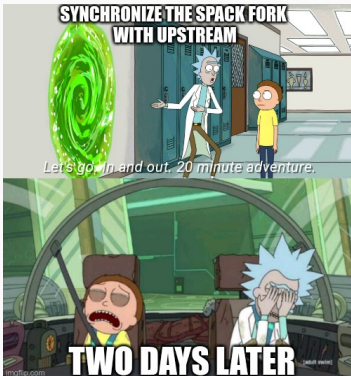
```
using namespace edm4hep;

// declare handle
DataHandle<MCParticleCollection> m_pHandle{
    "Particles",
    Gaudi::DataHandle::Reader,
    this};

// declare handle as property
declareProperty("ParticleColl",
               m_pHandle,
               "mc collection");

// use as
const auto particle = m_pHandle.get();
```

Spack & Nightly builds



- Key4hep uses the [spack](#) package manager
 - “Build everything from scratch”
 - Handle multiple configurations of packages
 - Many packages contributed by Key4hep
- Reworked our nightly builds to ditch our spack fork
 - Follow upstream spack closely
 - Run basic checks before deploying
 - Now available for Ubuntu22, CentOS7, AlmaLinux9

```
10:51:57 tmdlener@local:~$ source /cvmfs/sw-nightlies.hsf.org/key4hep/setup.sh
Ubuntu detected
Setting up the latest Key4HEP software stack from CVMFS ...
... Key4HEP release: 2023-06-11
... Use the following command to reproduce the current environment:
...
    source /cvmfs/sw-nightlies.hsf.org/key4hep/releases/2023-06-10/x86_64-ubuntu22.04-gcc11.3.0-opt/key4hep-stack/2023-06-11-kc603b/setup.sh
...
... If you have any issues, comments or requests open an issue at https://github.com/key4hep/key4hep-spack/issues
```


Release validation

- Prototype for release validation
key4hep-validation.web.cern.ch
- Full simulation & reconstruction chain
- Automatic comparison against known reference results
- Currently for CLIC jet quantities
- Extremely useful to catch problems from “trivial changes”

Home

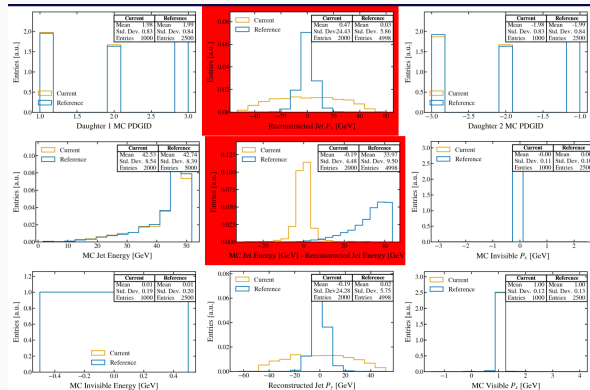
This is a webpage for validation of Key4hep software. The validation is done automatically against available options.

Simulation and reconstruction

Detector, geometry and process

[CLIC_v3_v15](#)

Last updated: 2023-06-13 10:09:58



Key4hep resources

- (Rolling) latest release of the complete Key4hep software stack

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

```
source /cvmfs/ilc.desy.de/key4hep/setup.sh
```

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

- Release early and release often

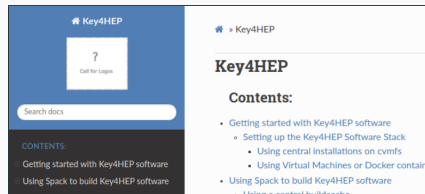
- Solicit feedback as early as possible

- Documentation available at key4hep.web.cern.ch

- Active weekly meetings (~ 10 – 15 attendees)

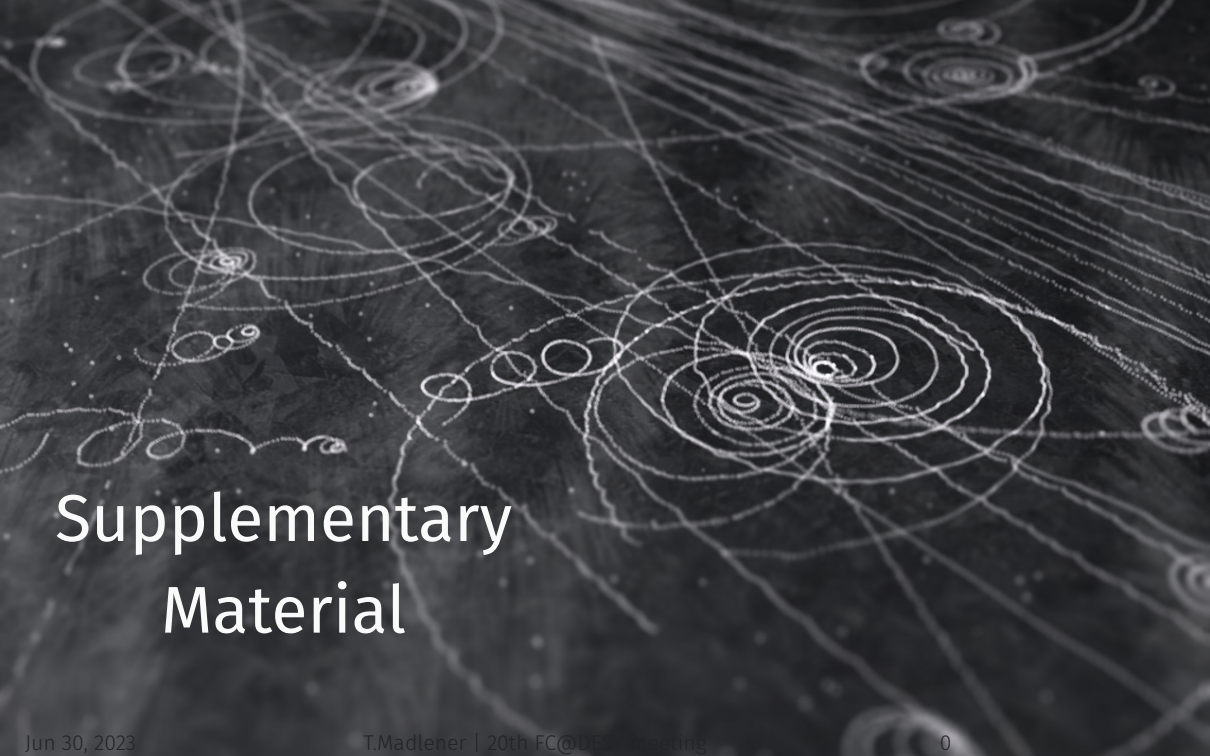
- <https://indico.cern.ch/category/11461/>

- Feedback and contributions are greatly appreciated



Summary & Conclusions

- Key4hep provides a common software stack for **all future collider projects**
- Very successful in **bringing together communities** and **focusing on common approaches**
- A lot of visible but also (almost) invisible developments recently
- **Key4hep can be (and is) used for physics studies now**
- Contributors are always welcome




Supplementary Material

Pointers to software (re)sources

- Key4hep

key4hep.github.io/key4hep-doc

 [key4hep](https://github.com/key4hep) - github organisation

- EDM4hep

 [key4hep/EDM4hep](https://github.com/key4hep/EDM4hep)

cern.ch/edm4hep

- DD4hep

 [AIDASoft/DD4hep](https://github.com/AIDASoft/DD4hep)

dd4hep.web.cern.ch

- iLCSoft










 [iLCSoft](https://github.com/iLCSoft) - github organisation

ilcsoft.desy.de



xkcd.com/138

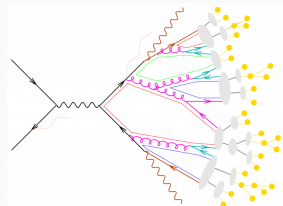
Key4hep packages

- **k4FWCore**  [key4hep/k4FWCore](https://github.com/key4hep/k4FWCore)
 - Core Key4hep framework providing core functionality, e.g.
 - Data Service for EDM4hep inputs
 - Overlay for backgrounds
- **k4SimDelphes** for Delphes fast simulation  [key4hep/k4SimDelphes](https://github.com/key4hep/k4SimDelphes)
- **k4MarlinWrapper** Marlin proc. wrapper  [key4hep/k4MarlinWrapper](https://github.com/key4hep/k4MarlinWrapper)
- Many packages migrated from FCCSW to Key4hep
 - **k4SimGeant4** for Geant4 simulation integration  [HEP-FCC/k4SimGeant4](https://github.com/HEP-FCC/k4SimGeant4)
 - **k4Gen** for generic generator interface  [HEP-FCC/k4Gen](https://github.com/HEP-FCC/k4Gen)
 - ...
- Ongoing work to integrate more components
 - ACTS tracking framework  [acts-project/acts](https://github.com/acts-project/acts) |  [key4hep/k4ActsTracking](https://github.com/key4hep/k4ActsTracking)
 - CLUE fast clustering algorithms  [.cern.ch/kalos/CLUE](https://cern.ch/kalos/CLUE) |  [key4hep/k4CLUE](https://github.com/key4hep/k4CLUE)

Generators in Key4hep

See [A. Siodmok's talk](#) for physics

- Generators are “just” software packages
- For inclusion in Key4hep a **spack recipe** is necessary
 - Building and installing becomes (almost) trivial
- Initial list from *LCG stacks* (mainly LHC focussed)
- Many e^+e^- additions since then
 - Including wrappers for better user experience



Generators currently available via spack and Key4hep

- Generators

babayaga ¹²	baurmc ²	bhlumi ¹²	crmc ²	evtgen	genie ²
gosam ²	guinea-pig ¹²	herwig3	herwigpp ²	kkmcee ¹	madgraph5amc
photos	pythia6 ²	pythia8	sherpa	starlight ²	superchic ²
tauola ²	vbfno	whizard	circe2 ²³		

- “Generator tools”

agile ²	alpgen ²	ampt ²	apfel ²	ccs-qcd ²	chaplin ²
collier ²	cuba ²	dire ²	feynhiggs ²	form ²	hepmc
hepmc3	heppdt	hoppet ²	hztool ²	lhapdf	lhapdfsets ²
looptools	openloops	professor ²	prophecy4f ²	qd ²	qgraf ²
recola ²	rivet	syscalc ²	thepeg	unigen ²	yoda

- Currently the **latest version** of each package is installed in Key4hep stack

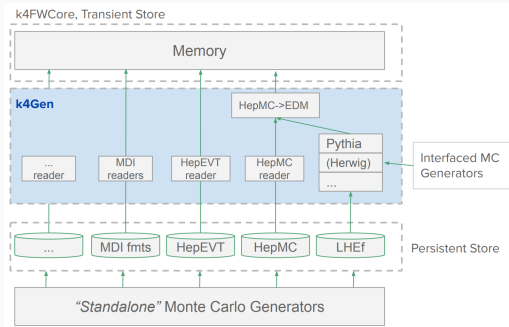
Installed with [current Key4hep stack](#)

¹ Available from [key4hep-spack](#) repository

² Single version only ³ Recently added

Generator interoperability

- Majority of generators come as standalone executables
- Some have callable interfaces
 - Pythia, EvtGen, Herwig, ...
- Interoperability requires **common, well defined, data formats** or interfaces
 - Fully hadronized outputs in **HEPMC3**, **EDM4hep** for simulation
 - APIs can also be accommodated
- *k4Gen* offers several readers and tools to work on MC events
 - Particle gun, particle filters, vertex smearing, ...



G.Ganis@ecfa.eu @ECFA generators workshop, Nov 2021