



# Key4hep - software ecosystem for future HEP experiments

---



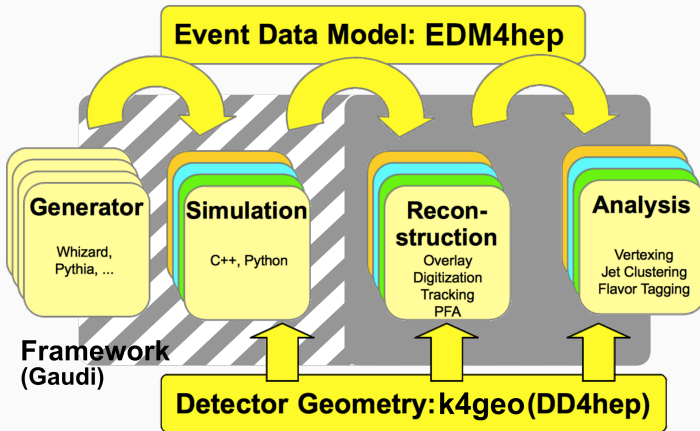
This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 101004761.

Thomas Madlener

Joint Meeting APC 13 and PRC 99

Apr 08 2025

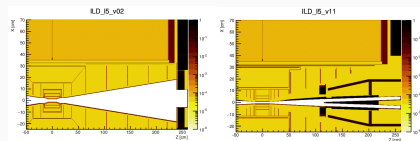
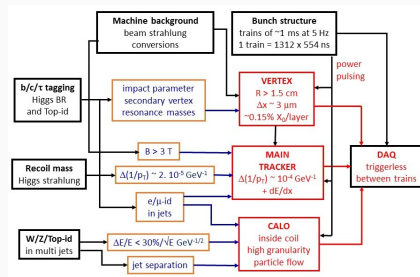
# From generation to analysis - the general workflow



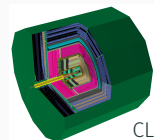
- Many steps involved from generating events to analyzing them
- Hundreds of SW packages
  - Building & deploying
  - Consistency
  - Reproducibility
- Key4hep aims at facilitating interoperability and focuses on common approaches

# Software Challenges for Future Colliders

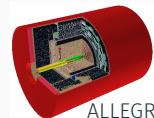
- Future Colliders demand unprecedented systematic uncertainties
  - Experimental sources
  - Theory predictions
- Detector concepts under active development
  - Multifaceted optimization problem
- Modular, flexible and well-maintained software is crucial



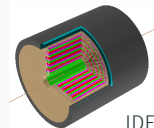
Adapting ILD for FCC-ee



CLD

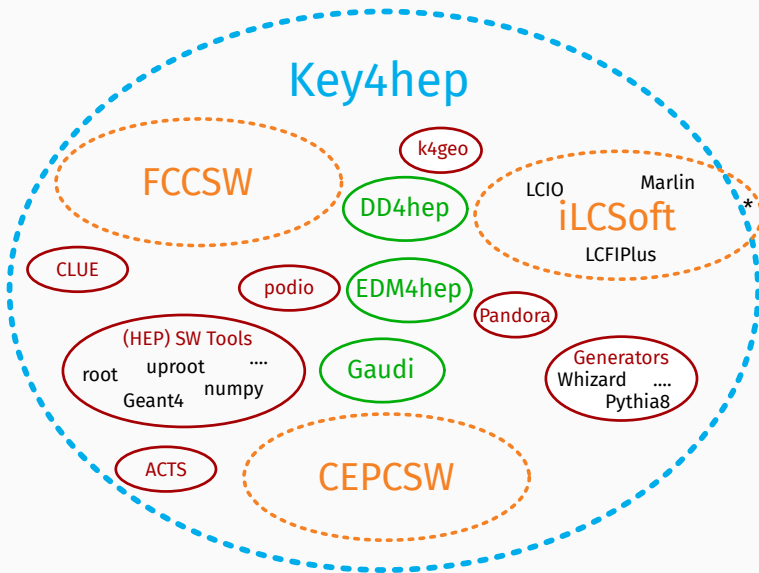


ALLEGRO



IDEA

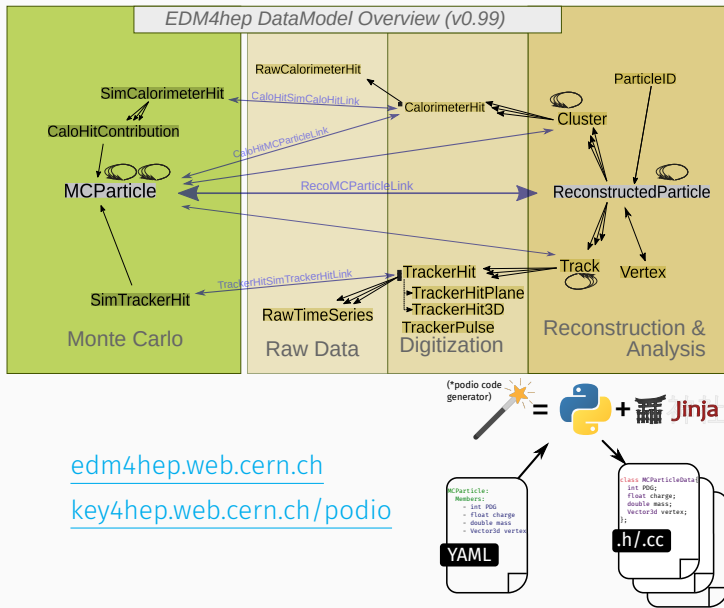
# Key4hep (simplified) overview



- Consolidated several existing SW stacks
- Supports full lifecycle for future collider studies
- Core components
  - **Detector description**
  - **Event Data Model**
  - **Event processing framework**


\*Some testbeam related SW not yet included

# EDM4hep - The common EDM for Key4hep

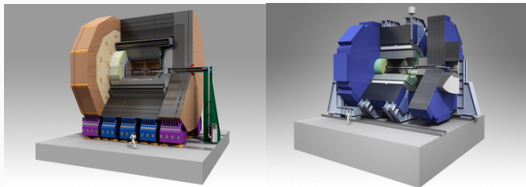
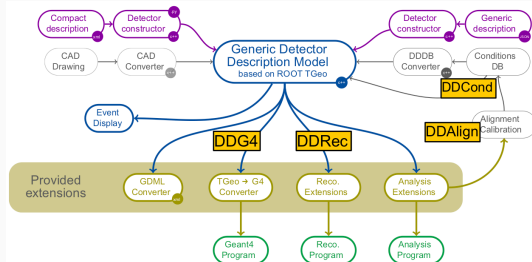


- Data format and vocabulary types for interfaces
- Based on LCIO and FCC-edm
  - Focus on usability in reconstruction & analysis
- Can easily be extended
  - Prototyping!
- Generated via podio
  - Different I/O backends

# DD4hep - Detector description

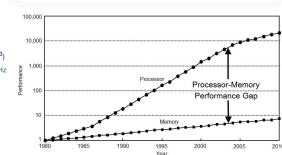
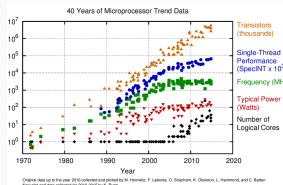
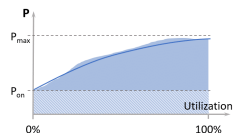
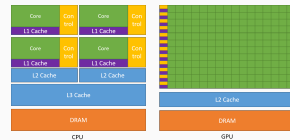
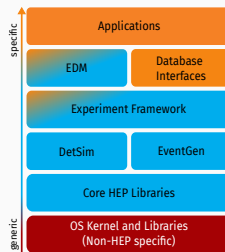
- Complete detector description
  - Geometry, materials, visualization, readout, alignment, calibration, ...
- From a **single source of information**
  - Simulation, reconstruction, analysis
- Comes with a powerful plug-in mechanism that allows customization
- More or less “industry standard” now
  - FCC, ILC, CLIC, EIC, LHCb, CMS, ODD, ...
-  [key4hep/k4geo](https://github.com/key4hep/k4geo) with many detector models
- `ddsim` - standalone simulation executable

[dd4hep.web.cern.ch](https://dd4hep.web.cern.ch)



# Common Challenges in SW for HEP (and science)

- Non-trivial ecosystem of interacting components
- Modern computing resources are evolving rapidly
- Programming paradigms for optimal exploitation change
  - Multithreading
  - Cache friendliness
  - GPU programming
- Limited personpower available
- Sustainable SW development is vital
  - Avoid loss of precious expertise

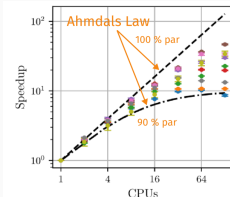
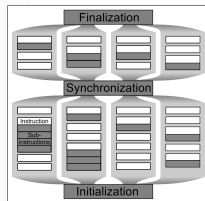
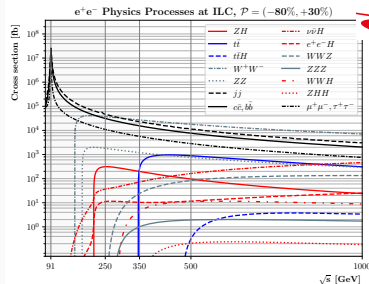


Year	2020	2021	2022	2023	2024	2025
Total FTE	3.5	4	3.5	3.5	4	3
Permanent FTE	0.1	0.1	0.1	0.6	0.5	0.3

FTEs working on core Key4hep components

# Accelerating the Whizard Event Generator

- Multi-purpose MC generator, LO & NLO
  - Main tool for ILC, CLIC (beam polarisation!)
  - Major tool for FCC-ee & MuonCollider
- Parallization via OpenMP and MPI
  - [EPJC 79 \(2019\) 4, 344](#)
  - 20-50 x speedup
- Ongoing work for GPU-offloading
  - Matrix-element evaluation & phase-space integration
- Exploring application of ML techniques
- Invertible Neural Networks (INNs),  
Normalizing Flows



See also: [J. Reuters talk @ LCWS 2024](#)



# Key4hep approaches to SW challenges

- Foster collaboration by establishing common standards
  - Data formats, build system, package manager, ...
- Follow modern SW engineering practices
  - Including (automated) tests
  - Version control everything
  - Open source, permissively licensed
- Write documentation
- Make it easily citable
- Automate, automate, automate
  - Continuous integration & deployment (CI/CD) pipelines



```
100% tests passed, 0 tests failed out of 109  
Total Test time (real) = 1.16 sec
```

## Class podio::Frame

class Frame

The Frame is a generalized (event) data container that aggregates all relevant data.

It is possible to store collections as well as parameters / meta data in a Frame and all I/O facilities of podio operate on Frames.

### Public Functions

inline Frame()

Empty Frame constructor.

```
template<typename FrameDataT>  
Frame(std::unique_ptr<FrameDataT>)
```

Frame constructor from (almost) arbitrary raw data.

## Cite this repository

If you use this software in your work, please cite it using the following metadata. [Learn more about CITATION files.](#)

APA BibTeX

Gaede, F., Madlener, T., Declara Fernandez,

[View citation file](#)

zenodo

## key4hep-build.yaml

on: schedule

### Matrix: build

- ✓ build (nightly, alma9, k... 8m 25s
- ✓ build (nightly, ubuntu2... 4m 58s
- ✓ build (nightly, ubuntu2... 4m 32s
- ✓ build (release, alma9, k... 5m 39s
- ✓ build (release, ubuntu2... 2m 19s

# 5 years of Key4hep experience

- Initially started by ILC, CLIC, FCC & CEPC
- Now also used (partially) by MuonCollider, EIC & LUXE
- ~ 500 packages built and deployed via `/cvmfs`
- Reaching consensus and community building require work
- **Sharing burden allows to reap benefits**
  - [ECFA report](#)
  - [FCC Feasibility Study](#)
  - New detector concepts
- Community appreciation does not necessarily imply better funding
- **Fruitful approach to SW development**

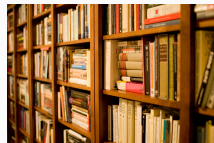
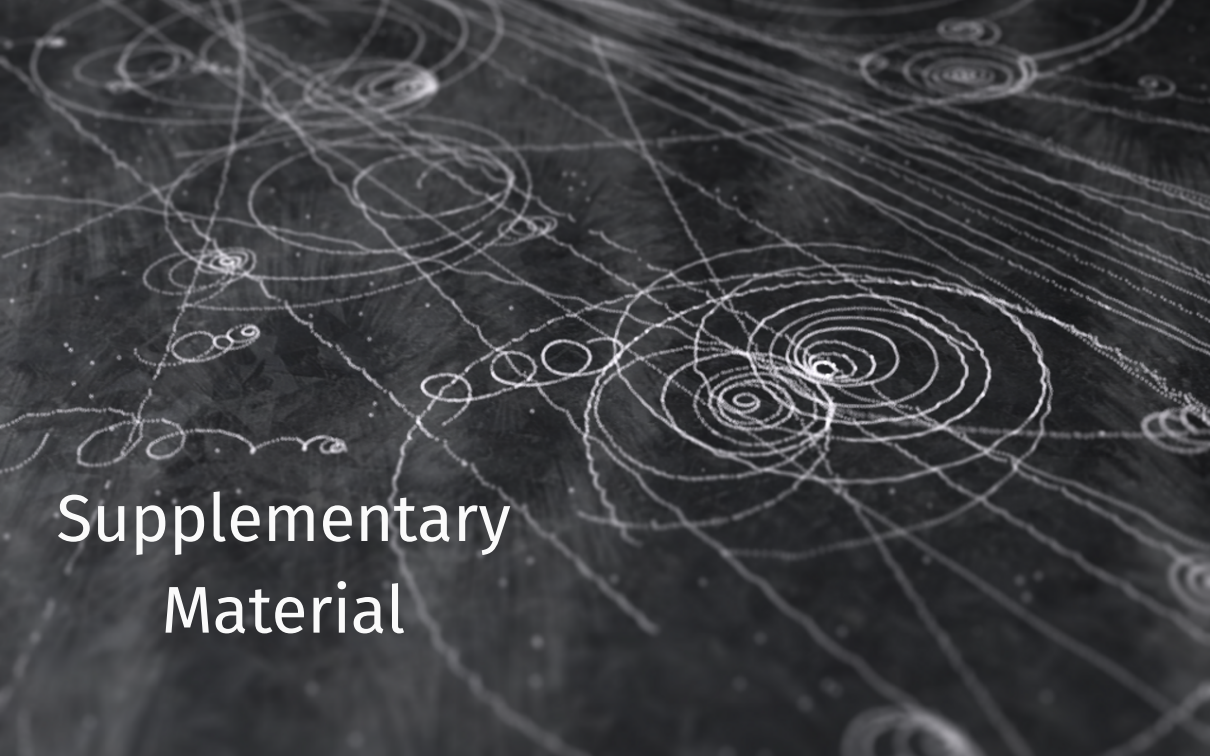


Photo by Stewart B. / [CC-BY](#)



# Summary

- Future collider studies require well maintained, yet flexible software
- Many challenges for sustainable software development in HEP (and sciences in general)
- Key4hep project aims at providing a software stack for all projects
- **Very successful in bringing communities together and focusing on common approaches**
- De-facto standard for all future collider project studies by now
- Possible blueprint for similar efforts in other areas of DESY or outside of HEP

The background is a dark gray, textured surface. It is covered with numerous white, hand-drawn or etched lines. These lines form a complex network of patterns, including several sets of concentric circles of varying sizes, some spirals, and many intersecting straight and curved lines that create a web-like structure. The lines vary in thickness and brightness, giving the impression of being drawn with a fine tool like a needle or a fine pen.

# Supplementary Material

# Keyhep releases and nightlies

- (Rolling) latest release of the complete Key4hep software stack
  - Full stacks for AlmaLinux9, Ubuntu22.04, Ubuntu24.04

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

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

- Documentation

- [key4hep.github.io/key4hep-doc](https://key4hep.github.io/key4hep-doc)
- Includes tutorials & How-tos

- **Release early and release often**

- Make fixes available early
- Discover problems and collect feedback as early as possible

- Biweekly, alternating meetings for Key4hep & EDM4hep

- [indico.cern.ch/category/11461/](https://indico.cern.ch/category/11461/)


```
Ubuntu 22.04 detected
Setting up the latest Key4hep software stack from CVMFS
Note that you are using the latest stack, which may point to a newer stack in
the future
Use the following command to reproduce the current environment:

    source /cvmfs/sw.hsf.org/key4hep/setup.sh -r 2024-04-12

If you have any issues, comments or requests, open an issue at https://github.
com/key4hep/key4hep-spac/issu
```

# Spack for Key4hep



- [Spack](#) is a package manager
  - Independent of operating system
  - Builds all packages from source
- Originally developed by the HPC community
  - Emphasis on dealing with **multiple configurations** of the same package
- Basic building block is a formalized build procedure → **spack recipe**
  - Build instructions, dependencies, versions and location of source code
  - ~ 8000 packages currently available from spack
  - Many Key4hep packages in  [key4hep/key4hep-spack](https://github.com/key4hep/key4hep-spack)
- The whole Key4hep software stack can be built from scratch using spack

```
spack install key4hep-stack
```

# Spack recipe

```
class Evtgen(CMakePackage):
    """EvtGen is a Monte Carlo event generator that simulates
    the decays of heavy flavour particles, primarily B and D mesons."""

    homepage = "https://evtgen.hepforge.org/"
    url = "https://evtgen.hepforge.org/downloads?f=EvtGen-02.00.00.tar.gz"

    tags = ["hep"]

    maintainers = ["vvolkl"]

    version("02.00.00", sha256="02372308e1261b8369d10538a3aa65fe60728ab343fcb64b224dac7313deb719")
    # switched to cmake in 02.00.00
    version(
        "01.07.00",
        sha256="2648f1e2be5f11568d589d2079f22f589c283a2960390bbdb8d9d7f71bc9c014",
        deprecated=True,
    )

    variant("pythia8", default=True, description="Build with pythia8")
    variant("tauola", default=False, description="Build with tauola")
    variant("photos", default=False, description="Build with photos")
    variant("hepmc3", default=False, description="Link with hepmc3 (instead of hepmc)")

    patch("g2c.patch", when="@01.07.00")
    patch("evtgen-2.0.0.patch", when="@02.00.00 ^pythia8@8.304:")

    depends_on("hepmc", when="~hepmc3")
    depends_on("hepmc3", when="+hepmc3")
    depends_on("pythia8", when="+pythia8")
```

Build system

Where to find source code

Available versions

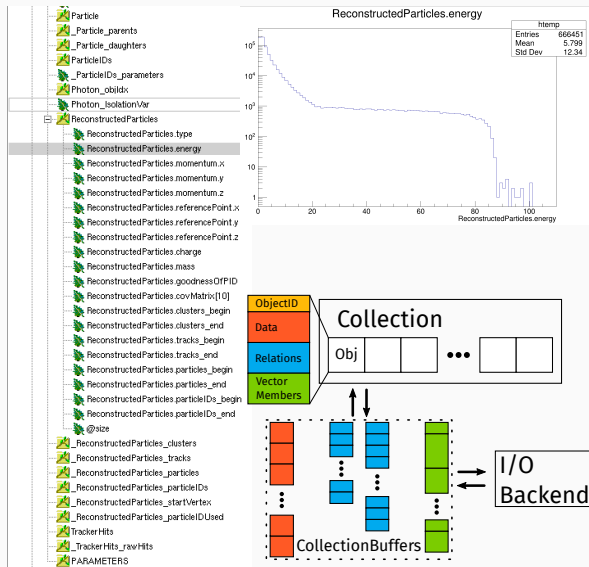
Variants / build options

On-the-fly patches

Dependencies

# podio supports different I/O backends

- Default **ROOT** backend
  - Effectively flat **TTree** / **RNTuple**
  - Files can be interpreted **without EDM library(!)**
  - Can be used in **RDataFrame** (**FCCAnalyses**) or with **uproot**
  - Also [with Julia](#)
- Adding more I/O backends is possible
  - Alternative SIO backend exists
  - **RDataSource** for better **RDataFrame** integration
- Generated interfaces provide many “convenience features”





# k4MarlinWrapper

- Wraps Marlin processor in a Gaudi algorithm and allows to run them unchanged
- Automatic, on-the-fly conversion between LCIO and EDM4hep
- Allows to “mix and match” existing reconstruction algorithms with new developments
  - Working horse for many FCC full simulation studies at the moment

