

What Is key4hep?

Why And How To Get Started?

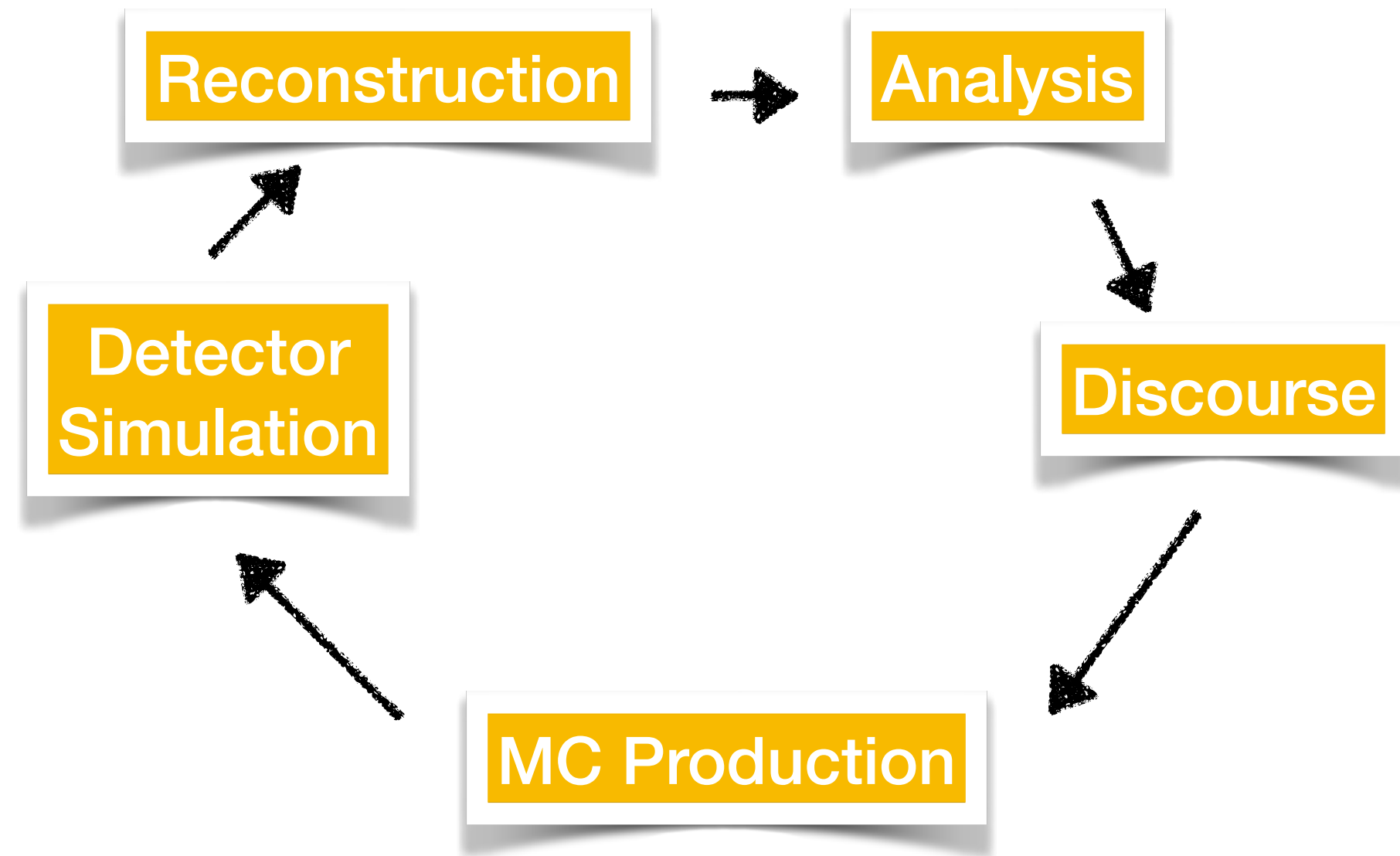
Carsten Hensel, CBPF



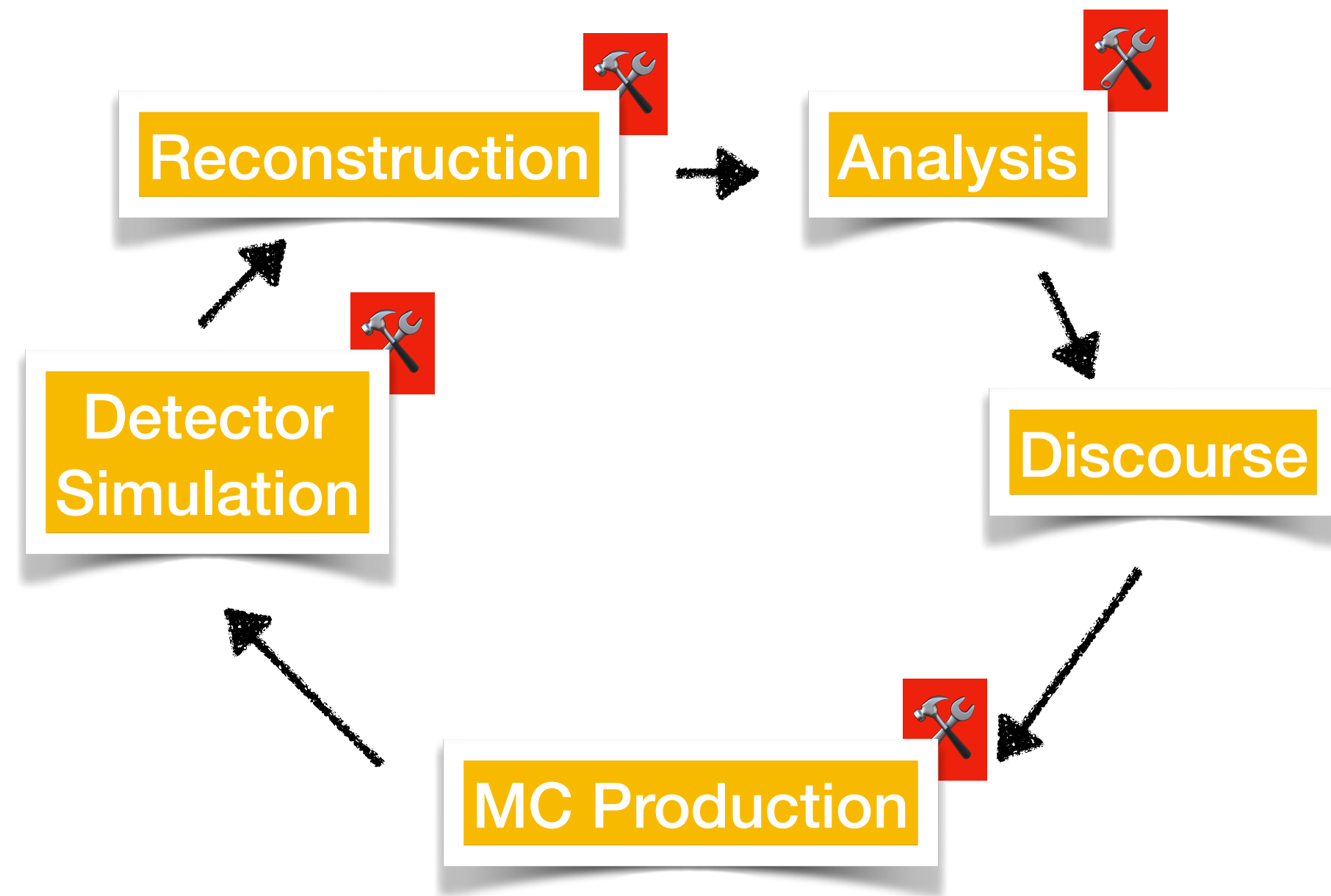
CBPF
Centro Brasileiro
de Pesquisas Físicas

UNIDADE DE PESQUISA DO MCTI

Physics Workflow

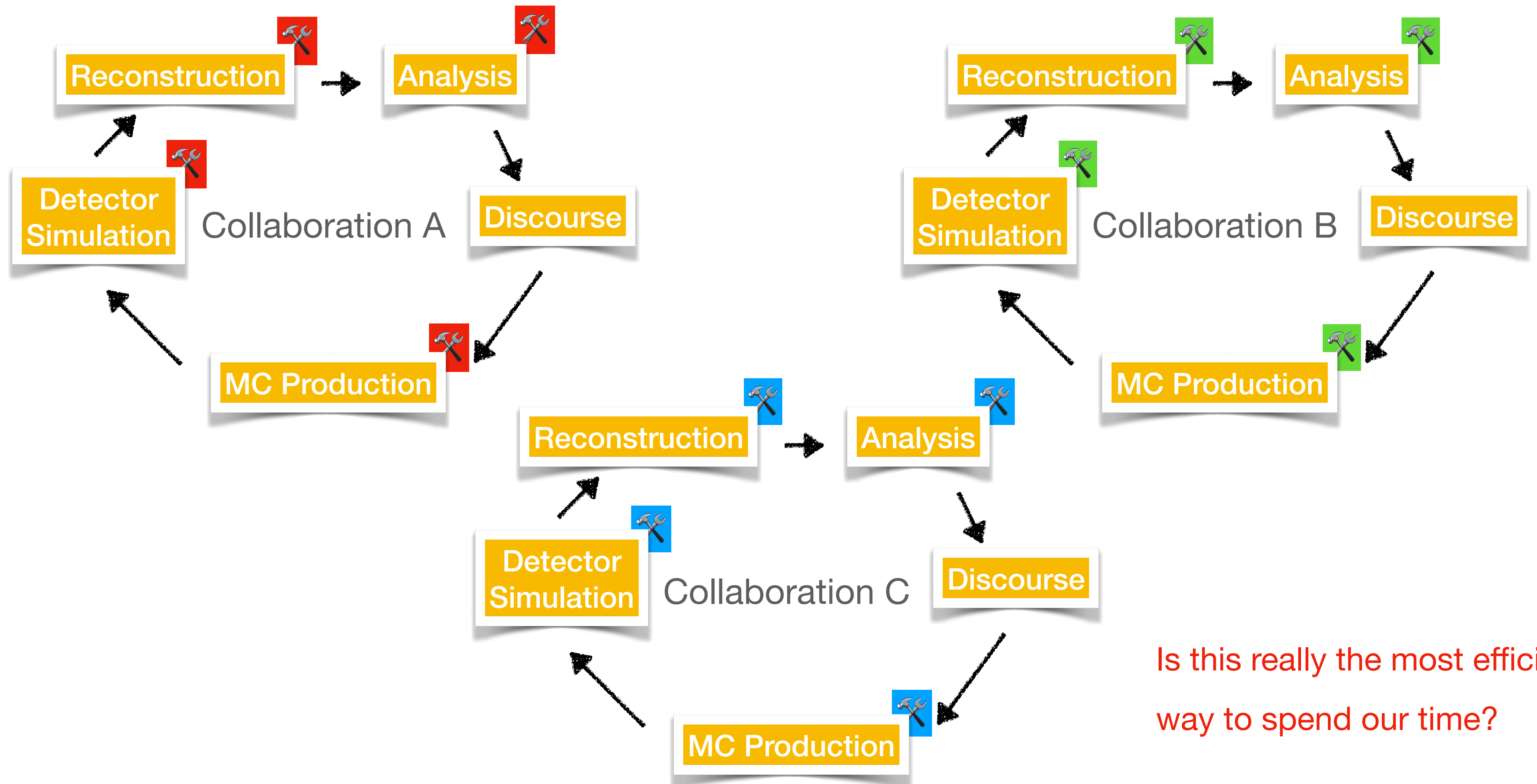


Physics Workflow



Most steps in a normal physics workflow require a suite of software tools.

Physics Workflow



Is this really the most efficient way to spend our time?

key4hep

Why?

What?

How?

key4hep

Why?

What?

How?

key4hep Motivation

- Future detector studies rely on **well maintained** software for studying their potential
- HEP software stack is **ecosystem** of interacting components
- **Maintenance** of a consistent HEP software stack **is non-trivial**
- (Human) resources are scarce
- **Sharing the burden** allows everybody to reap the benefits

key4hep Motivation

- Future detector studies rely on **well maintained** software for studying their potential
- HEP software stack is **ecosystem** of interacting components
- **Maintenance** of a consistent HEP software stack **is non-trivial**
- (Human) resources are scarce
- **Sharing the burden** allows everybody to reap the benefits

enter key4hep

key4hep Goals

- **Provide and maintain consistent software stack** that allows to do physics studies for all projects
- Ensure **interoperability** of the necessary building blocks
- **Reuse** existing solutions where possible (expertise from LHC and LC communities)
- **Focus** on new developments on EW/Higgs factory specifics
- **Share** knowledge, processes, workflows, and resources

Not a goal: develop and maintain project specific software and workflows.

key4hep

Why?

What?

How?

key4hep

Why?

What?

How?

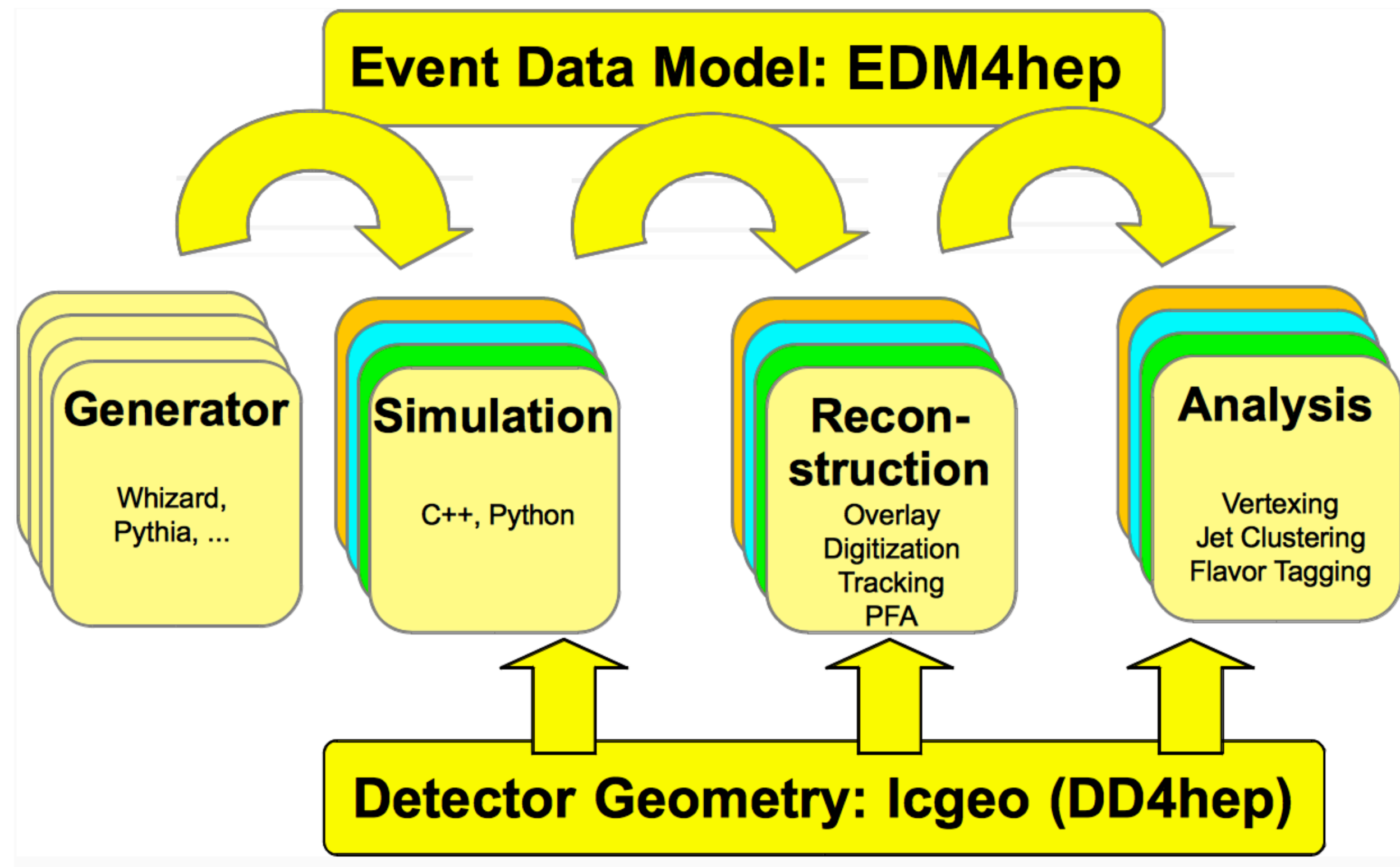
What Is key4hep?

- Software stack that connects and extends individual packages towards a complete data processing framework for detector studies.

- Fast/full simulation
- Reconstruction
- Analysis

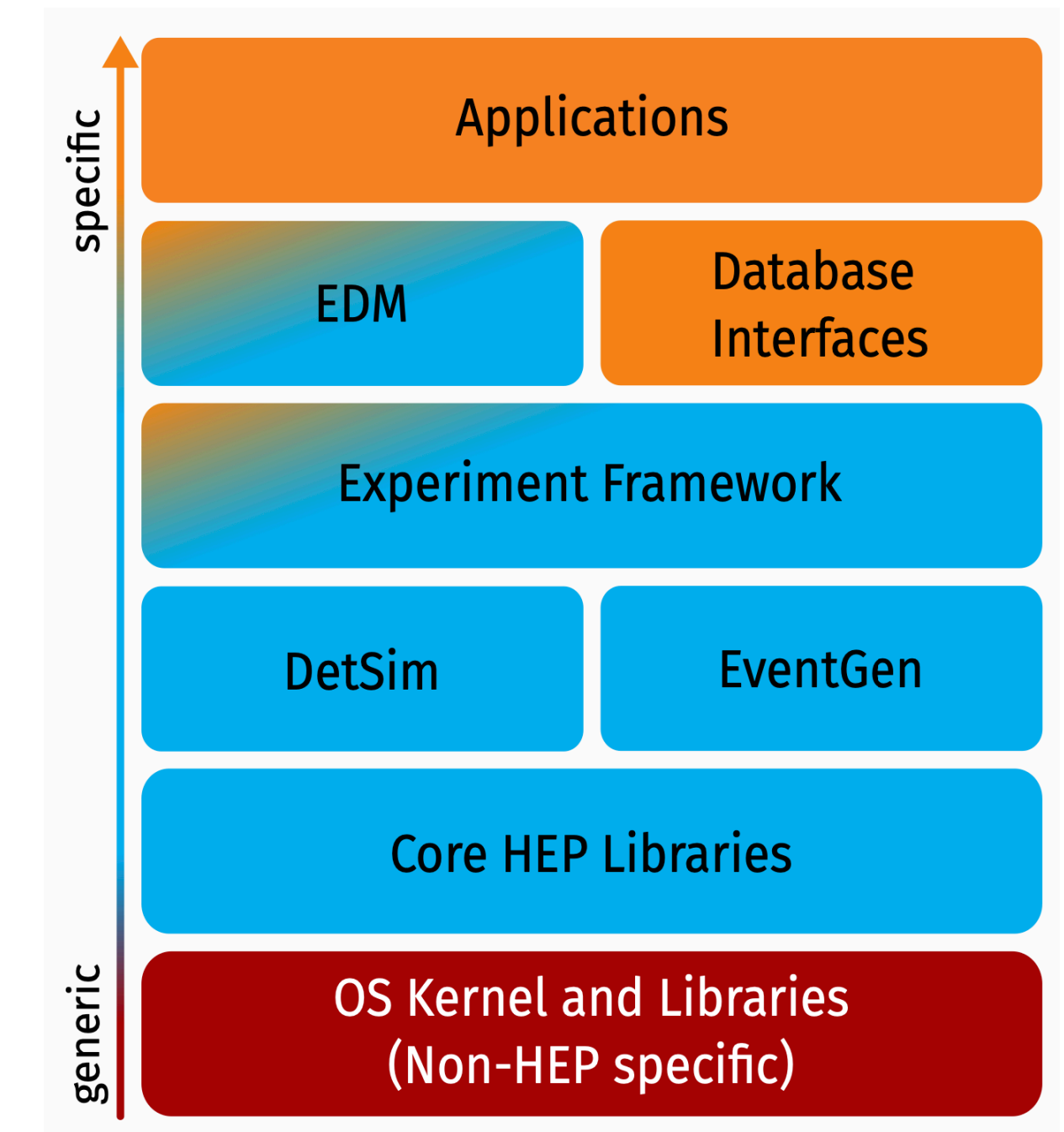
- Components:

- event data model: **EDM4hep**
- Geometry information: **DD4hep**
- Framework: **Gaudi**
- Packaging and deployment: **Spack**



key4hep Stack

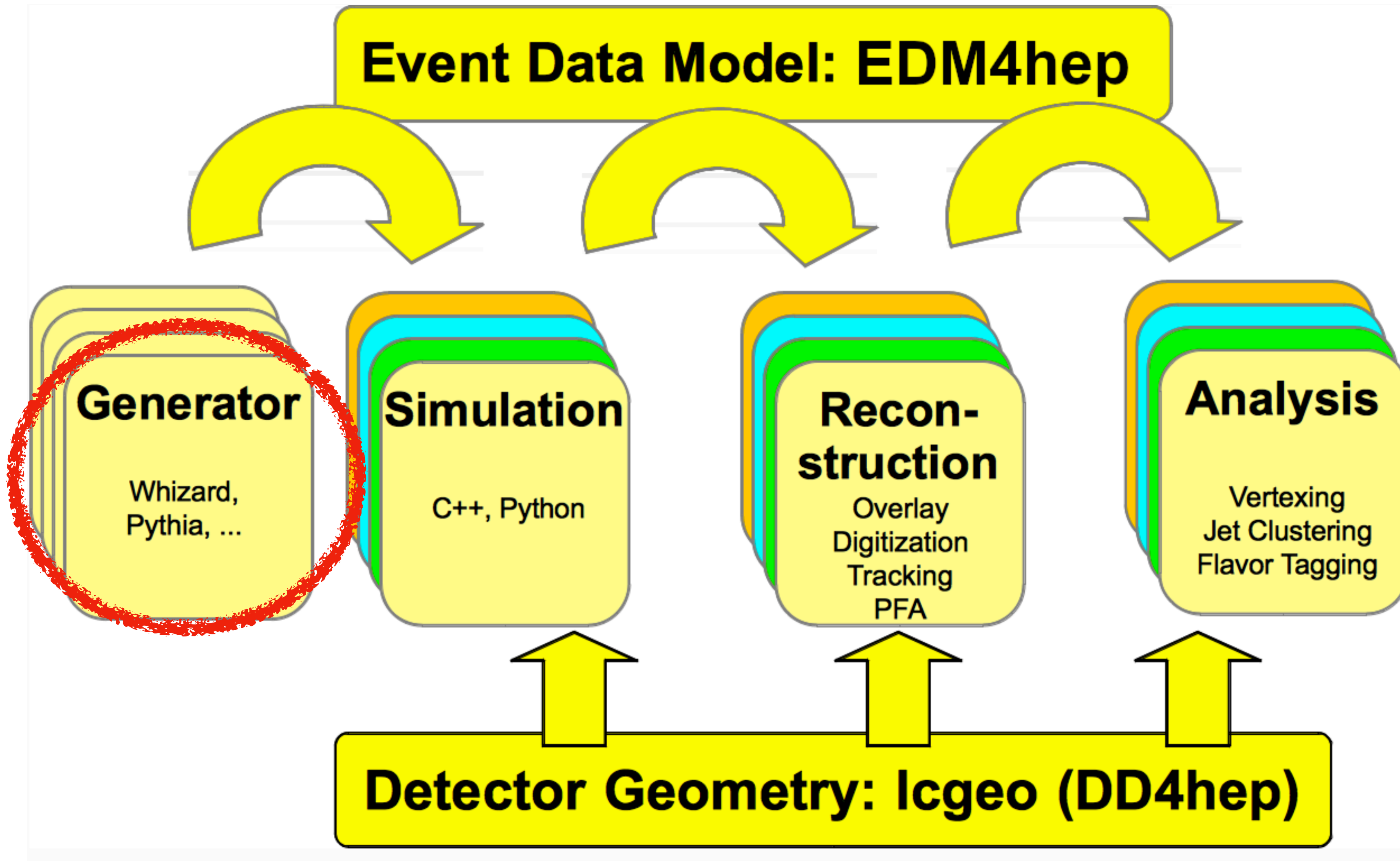
- Software provided in “stacks” deployed on [cvmfs](#)
- More than [500 packages](#) (most are dependencies)
- [Nightly builds](#) in [/cvmfs/sw-nightlies.hsf.org](#) with the latest of the key4hep packages and other packages.
- CentOS 7, AlmaLinux 9 and Ubuntu 22.04 supported
- [Releases](#) in [/cvmfs/sw.hsf.org](#) with version of the package



- [Easy setup](#) with cvmfs:

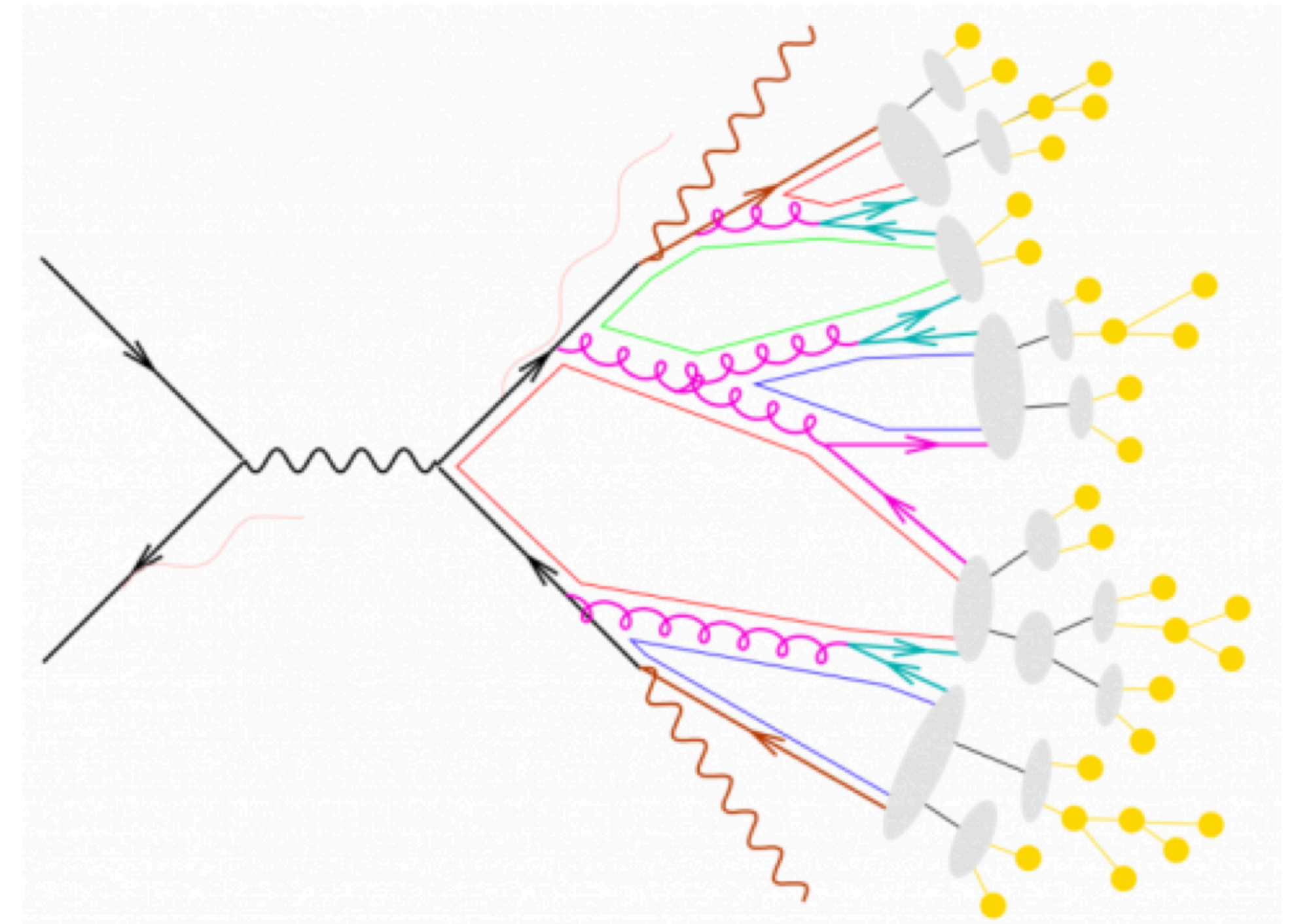
```
source /cvmfs/sw-nightlies.hsf.org/key4hep/releases/setup.sh # Latest nightly
source /cvmfs/sw.hsf.org/key4hep/releases/setup.sh # Latest release
```
- [Questions, problems, complaints](#) and anything else related to packages happens in <https://github.com/key4hep/key4hep-spack>

key4hep Stack

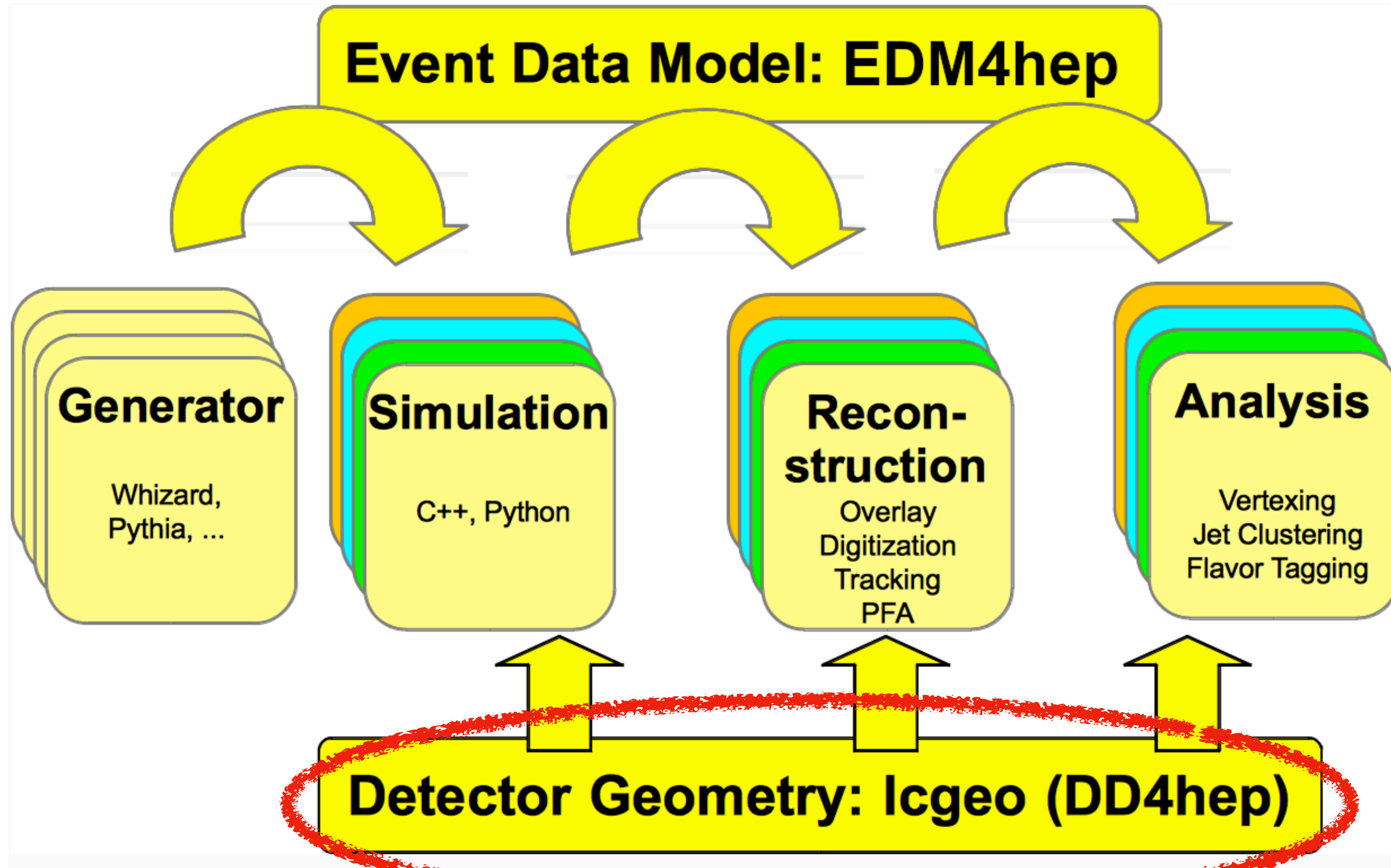


key4hep Components: Generators

- Generators are just [software packages](#)
- For inclusion in key4hep a [spack recipe](#) is necessary
- Building and installing “trivial”
- Initial list from LCG stacks (LHC focussed)
- Many [\$e^+e^-\$ additions](#) since then

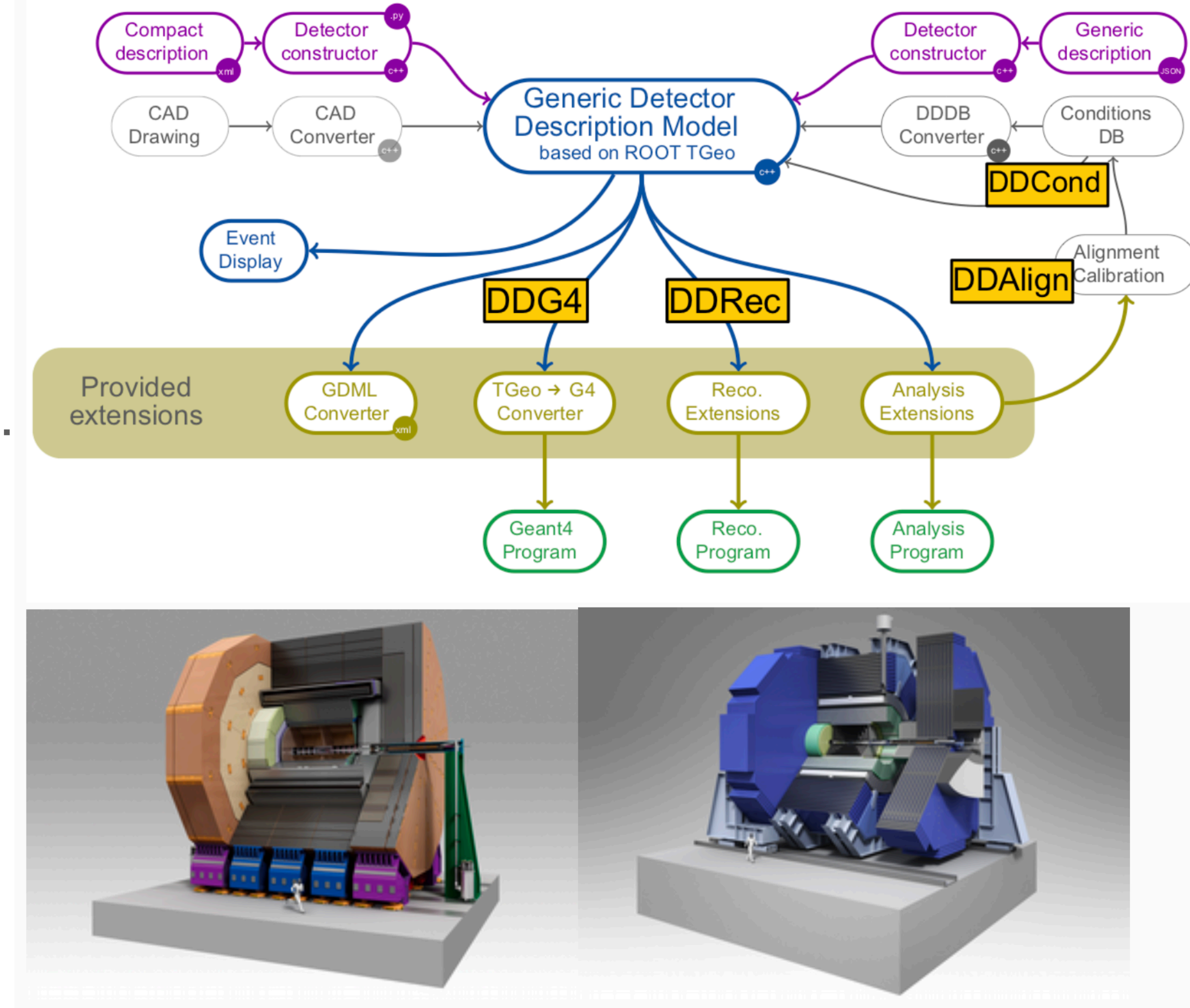


key4hep Stack

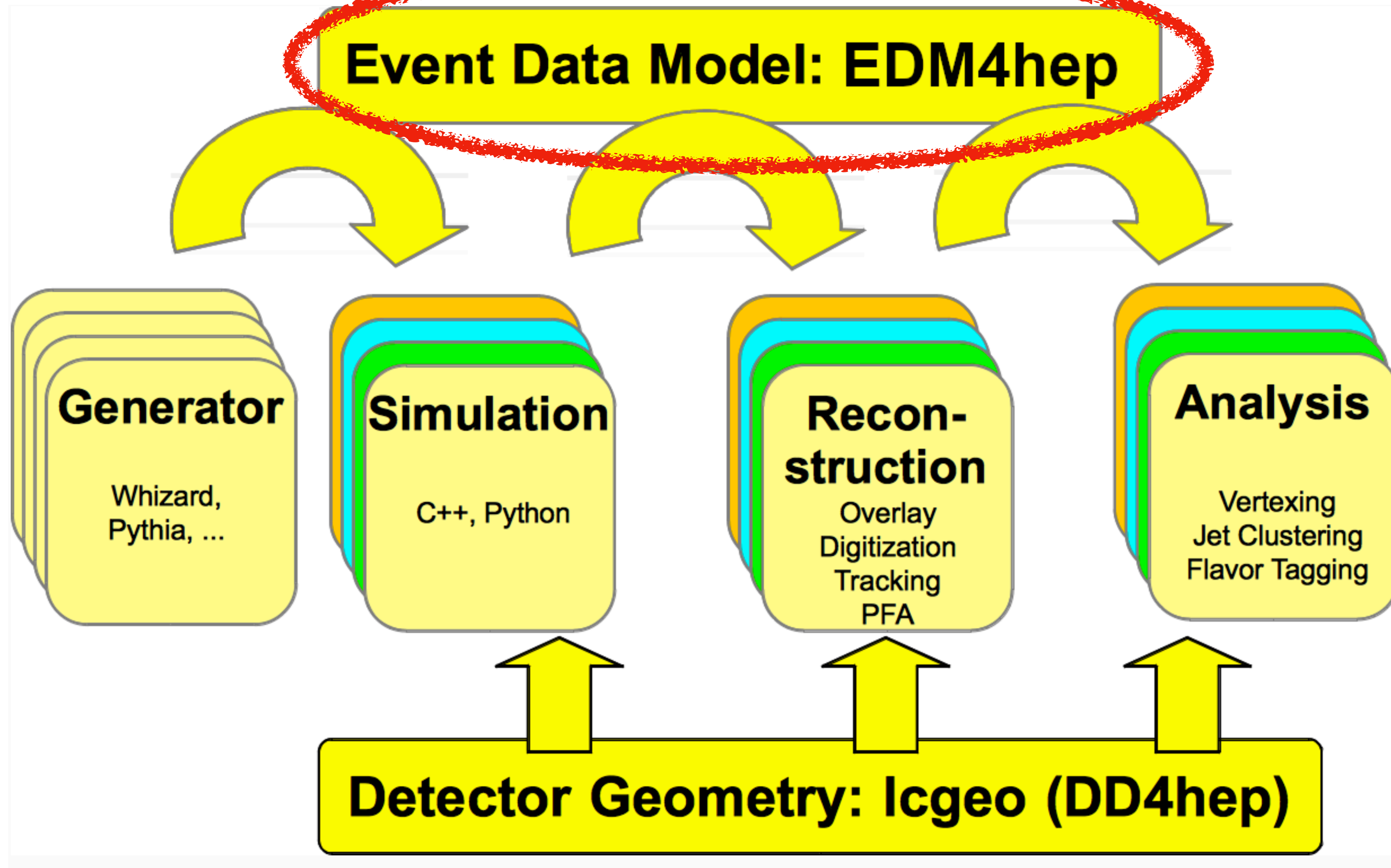


key4hep Components: DD4hep

- Originally for LC but targeting all of HEP from the start
- Complete **detector description**
- Simulation, reconstruction, analysis
- **“Industry” standard**: ILC, CLIC, FCC, CEPC, EIC, LHCb, CMS, ...
- Detectors that have been **added recently**:
 - IDEA
 - IDEA vertex detector
 - ALLEGRO
 - CLD with the ARC sub-detector

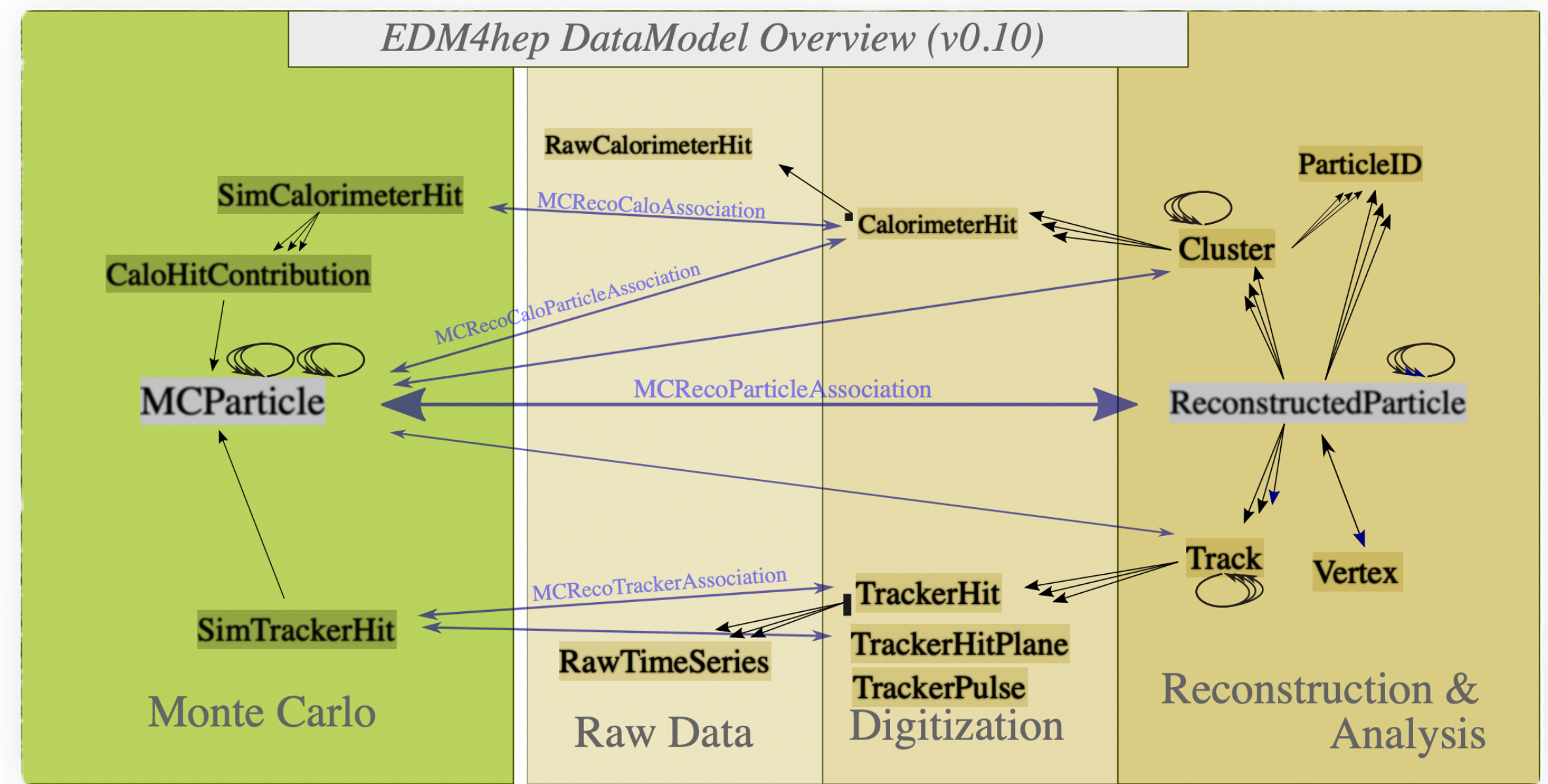


key4hep Stack

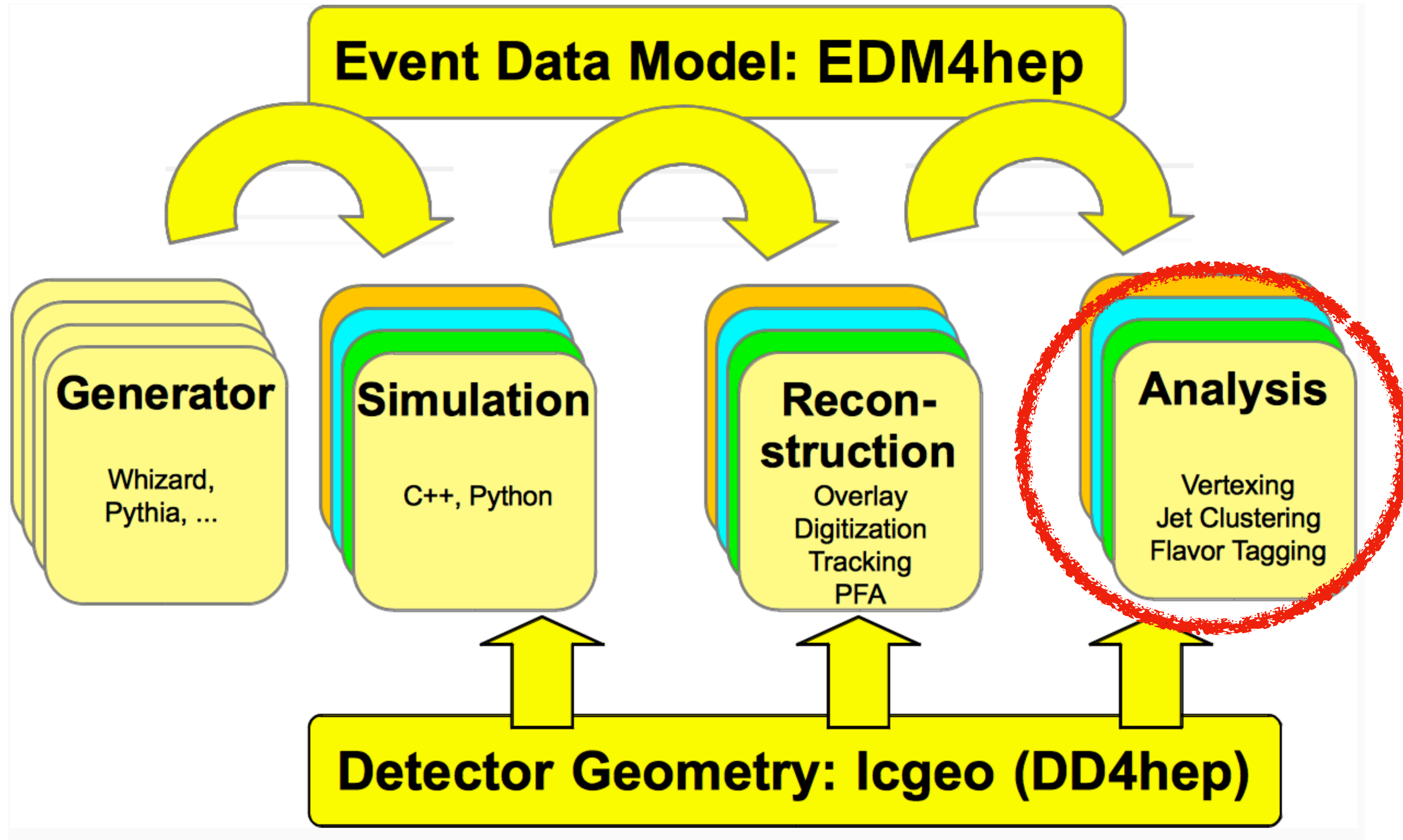


key4hep Components: EDM4hep

- **Interoperability** of different components requires a **common language**
- Based on LCIO and FCC-edm
 - Focus on **usability** in analysis
- Generated via **podio**
 - Supports prototyping of new datatypes
- Latest version: [EDM4hep 0.10.99](#)
- Currently finalizing v1 (backwards compatible from then)

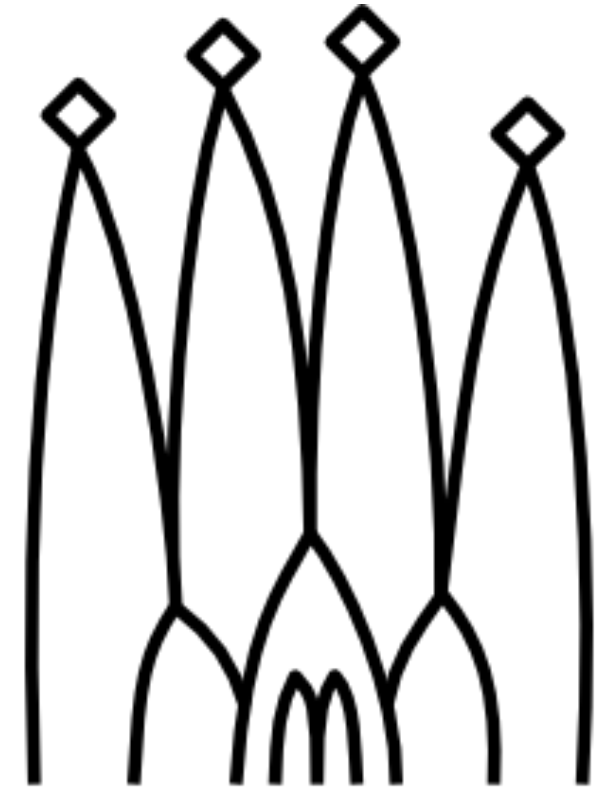
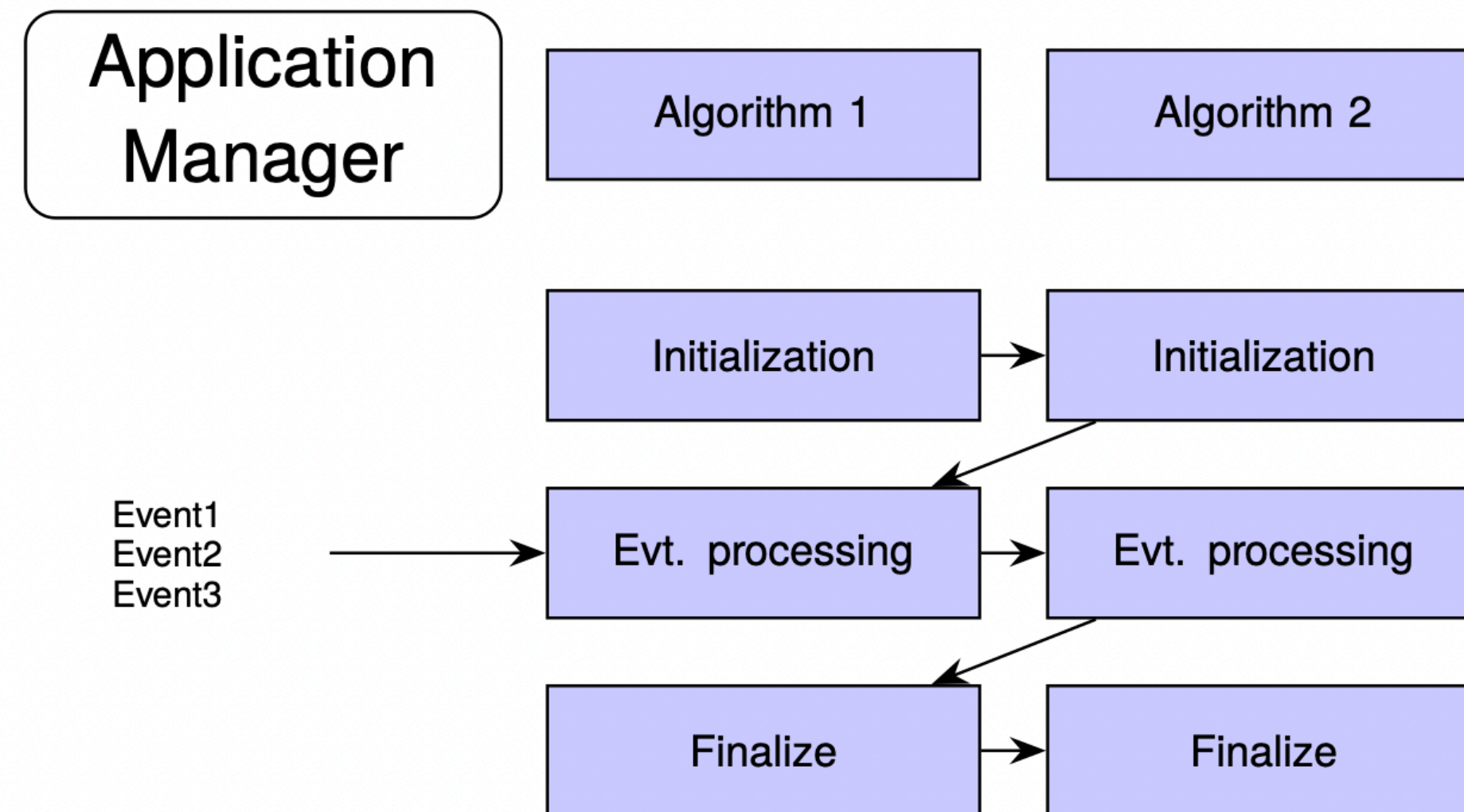


key4hep Stack



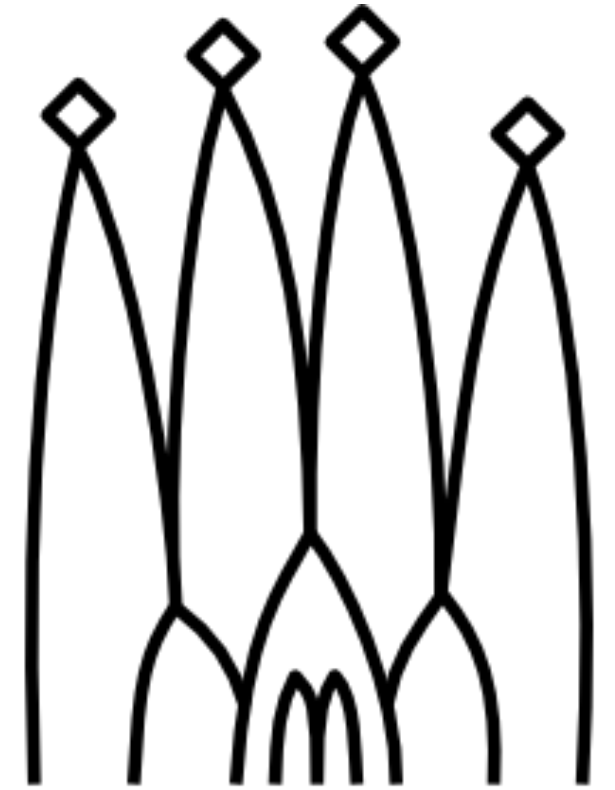
key4hep Components: Gaudi Framework

- [key4hep](#) has adopted [Gaudi](#) as its experiment framework
 - Originally developed by LHCb, used by ATLAS, FCCSW
 - “Battle-proven” by LHC data taking
 - Several (legacy) flavors
- k4FWCore core functionality
 - Data service for EDM4hep
- Dedicated packages for different tasks
- Main guideline: Use [EDM4hep](#) for event data and [DD4hep](#) for detector description



key4hep Components: Gaudi Framework

- Gaudi based core framework:
 - [k4Gen](#) for integration with generators
 - [k4SimGeant4](#) for integration with [Geant4](#)
 - [k4SimDelphes](#) for integration with [Delphes](#)
 - [k4FWCore](#) provides the interface between [EDM4hep](#) and [Gaudi](#)
 - [k4MarlinWrapper](#) to call Marlin processors



key4hep

Why?

What?

How?

key4hep

Why?

What?

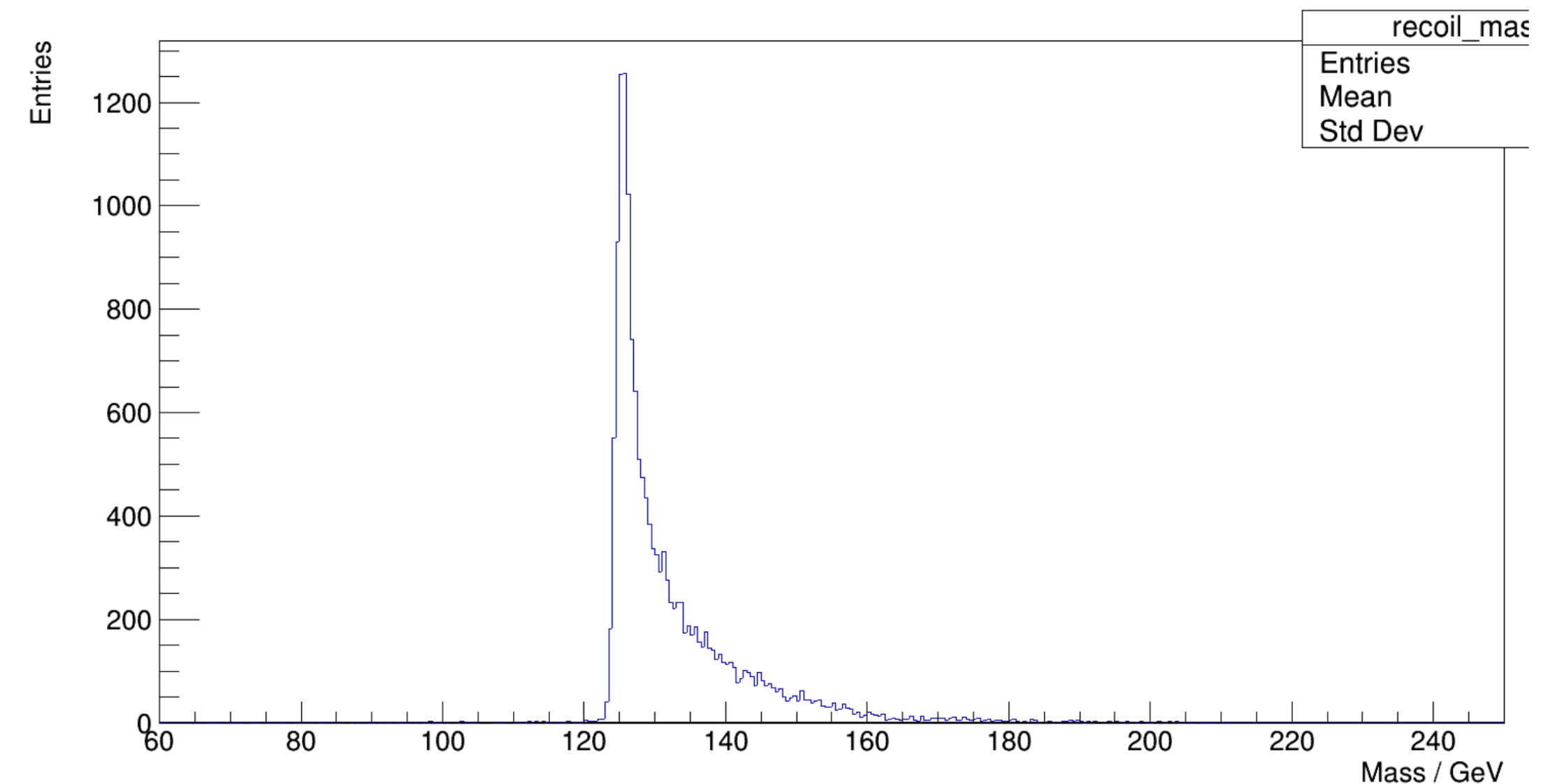
How?

key4hep Users

- FCCSW adopted EDM4hep (switched FCC-edm)
- CEPCSW using EDM4hep and switched from Marlin to Gaudi
- CLIC and ILD reconstruction can be run in Gaudi

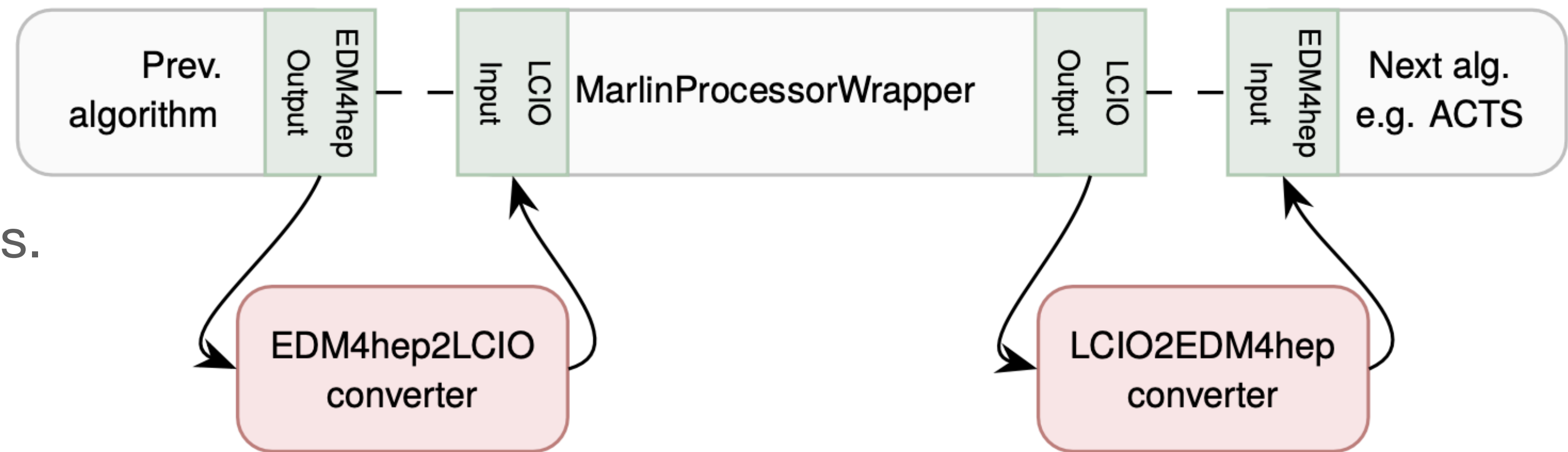
Where Do I Start?

- **Tutorials, tutorials, tutorials**
- There's bunch well documented tutorials available: [GitHub key4hep tutorials](#)
- Topics covered
 - EDM4hep
 - LCIO EDM4hep converters
 - Algorithms in key4hep using Gaudi
 - Plotting from files
- Feel free to [ask questions](#) / [report issues](#) about the tutorials via email or GitHub.



But What About My Marlin Processor?

- Do I have to re-write my Marlin processor?
- No need to re-do existing work: [k4MarlinWrapper](#)
- Wraps Marlin processors as Gaudi algorithms.
- Automatic, [on-the-fly conversion](#) between LCIO and EDM4hep
- [Converter](#) for xml \rightarrow py config files exists



Making the Switch



Gaudi
EDM4hep

Marlin
LCIO

(Valentin Volk)

Conclusion

- key4hep provides a **common software stack** for all future collider projects
- Very successful in **bringing together communities** and focusing on common approaches
 - **Common EDM4hep format** with increasing maturity and adoption
 - **DD4hep** for detector description
 - **Shared tools** for building, developing and deploying software stack
- **key4hep is ready to be used for future colliders studies now**
- Still a lot of room for your contributions
 - Now is the ideal time to get onboard



Final Remarks/Resources

- [key4hep documentation:](https://key4hep.github.io/key4hep-doc/)
<https://key4hep.github.io/key4hep-doc/>
- [Regular meetings:](https://indico.cern.ch/category/11461/)
<https://indico.cern.ch/category/11461/>
- Thanks to Thomas Madlener and Juan Carceller for providing inputs.
- [key4hep](#)
- [EDM4hep](#)
- [DD4hep](#)
- [FCCSW](#)
- [k4FWCore](#)
- [k4SimDelphes](#)
- [k4MarlinWrapper](#)

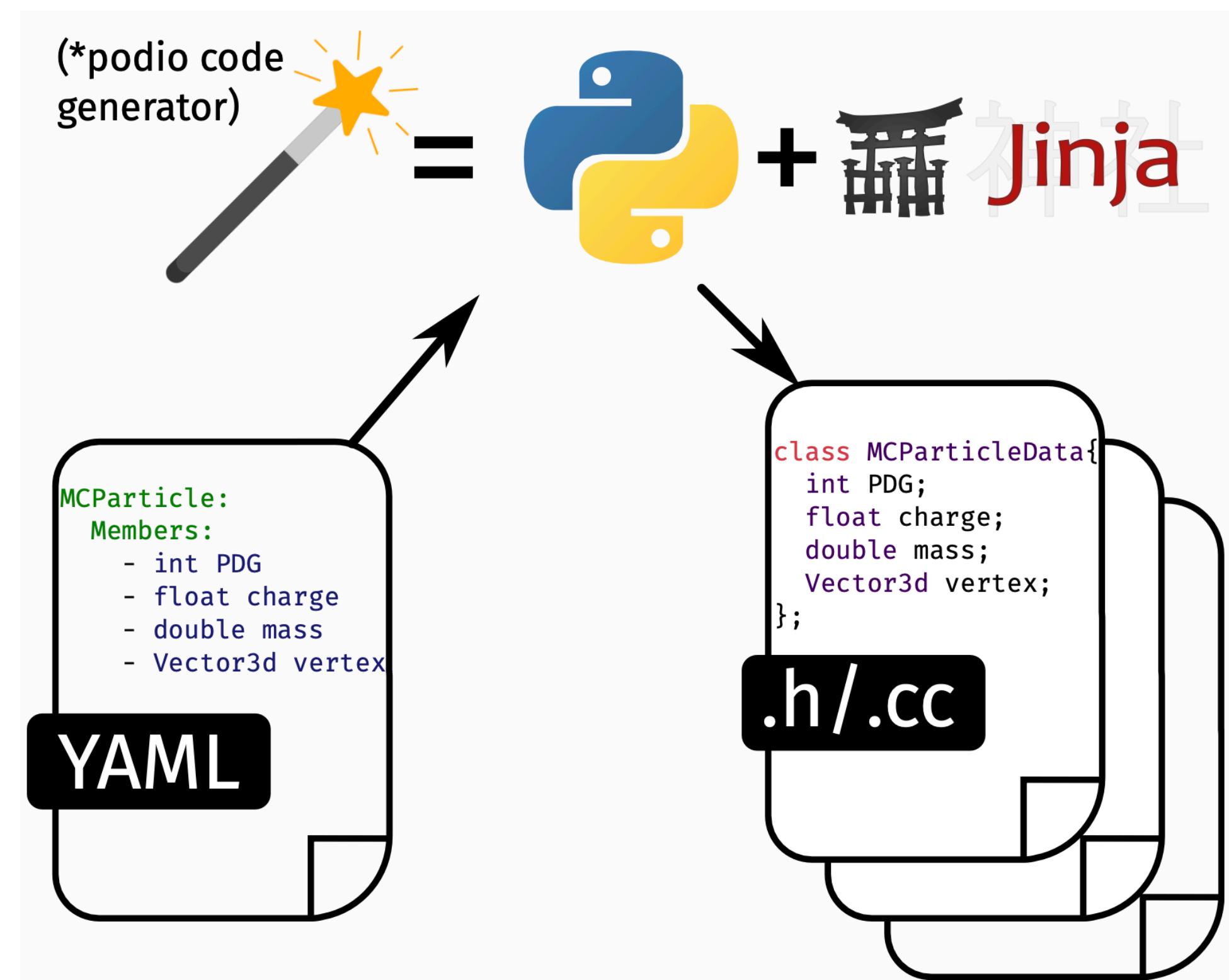
Additional Information

Spack for key4hep

- Spack is a [package manager](#)
 - Independent of OS
 - 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
 - ~[6700 packages](#) currently available from spack
 - key4hep maintains repository with additional packages
- The whole [key4hep](#) software stack can be [built from scratch](#) using: `spack install key4hep-stack`

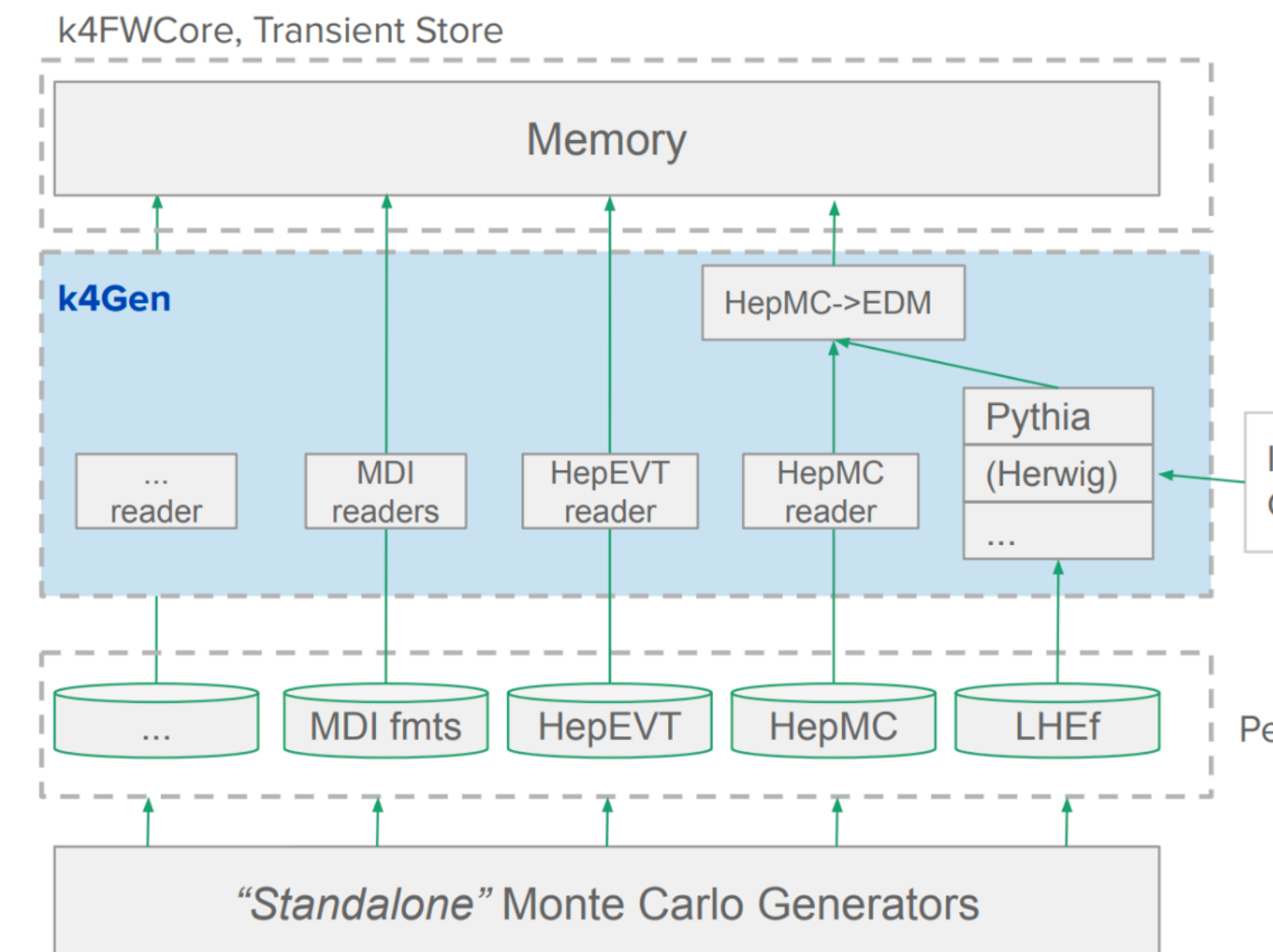
podio as generator for EDM4hep

- Traditionally HEP C++ EDMs are heavily Object Oriented
- Use podio to generate thread safe code starting from a high level description
- Provide an easy to use interface to the users
- AIDAsoft/podio



Generator Interoperability

- Majority of generators comes 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
 - API can also be accommodated
- **k4Gen** offers several readers and tools to work on MC events
 - Particle gun, particle filters, vertex smearing, ...



key4hep Components: Generators available via Spack

- Generators

<code>babayaga^{*†}</code>	<code>baurmc[†]</code>	<code>bhlumi^{*†}</code>	<code>crmc[†]</code>	<code>evtgen</code>	<code>genie[†]</code>
<code>gosam[†]</code>	<code>guinea-pig^{*†}</code>	<code>herwig3</code>	<code>herwigpp[†]</code>	<code>kkmcee[*]</code>	<code>madgraph5amc</code>
<code>photos</code>	<code>pythia6[†]</code>	<code>pythia8</code>	<code>sherpa</code>	<code>starlight[†]</code>	<code>superchic[†]</code>
<code>tauola[†]</code>	<code>vbfnlo</code>	<code>whizard</code>			

- “Generator tools”

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

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

Installed with current Key4hep stack

* Available from `key4hep-spack` repository

† Single version only