Key4hep and ILD software

Thomas Madlener

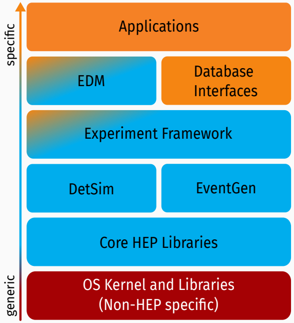
ILD workshop

Jan 16, 2024

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

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 initatives
    - 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 experiment specifics
- Share knowledge, processes, workflows and resources
  - Best practices, tutorials, documentation, ...

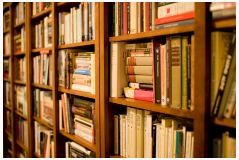
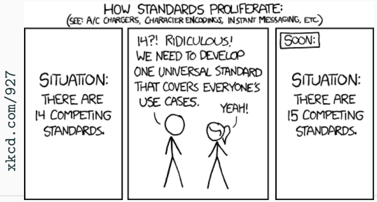


Photo by Stewart B. / CC-BY

Non-goal

- Develop and maintain project specific software and workflows

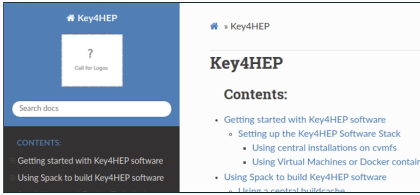

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.
xkcd.com/927

Key4hep resources

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

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

- **Release early and release often**
  - Solicit feedback as early as possible
- Documentation available at key4hep.web.cern.ch
- Active weekly meetings ($\sim 10 - 15$ attendees)
  - https://indico.cern.ch/category/11461/



Key4HEP

CONTENTS:

Getting started with Key4HEP software

Using Spack to build Key4HEP software

» Key4HEP

Key4HEP

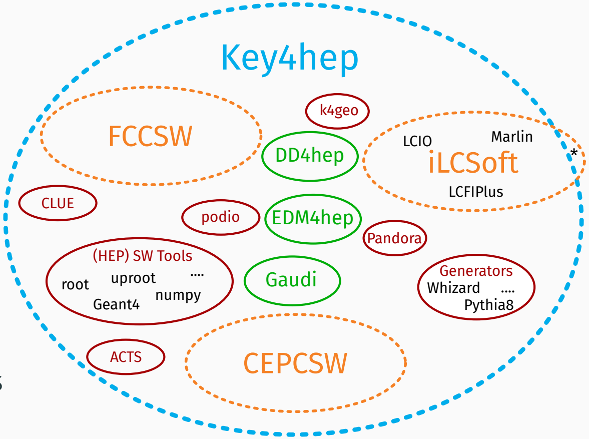
Contents:

- Getting started with Key4HEP software
  - Setting up the Key4HEP Software Stack
    - Using central installations on cvmfs
    - Using Virtual Machines or Docker contair
- Using Spack to build Key4HEP software

- **Feedback and contributions are greatly appreciated** (We really need a logo!)

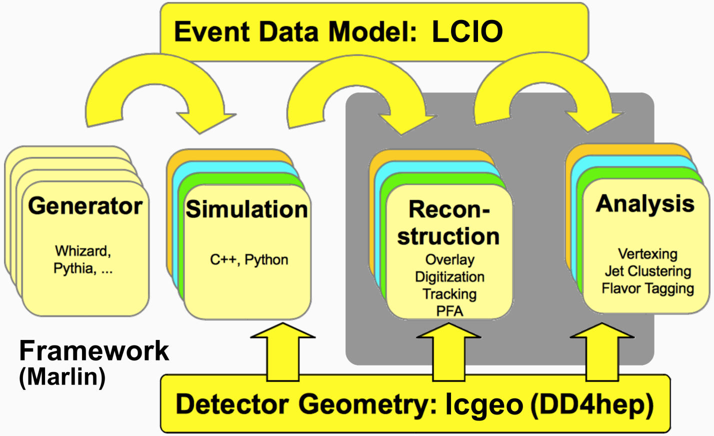
iLCSoft in Key4hep

- **iLCSoft is contained in Key4hep**
  - Things "just keep working"
  - Standard workflows, developing iLCSoft packages, ...
- `iLCInstall` vs `spack`
  - **Only build artifacts in Key4hep**
- No equivalent iLCSoft and Key4hep release yet
  - Key4hep with latest iLCSoft package versions
  - Different external package versions (ROOT, Geant4, ...)
  - Different tool chains (compilers, python, ...)



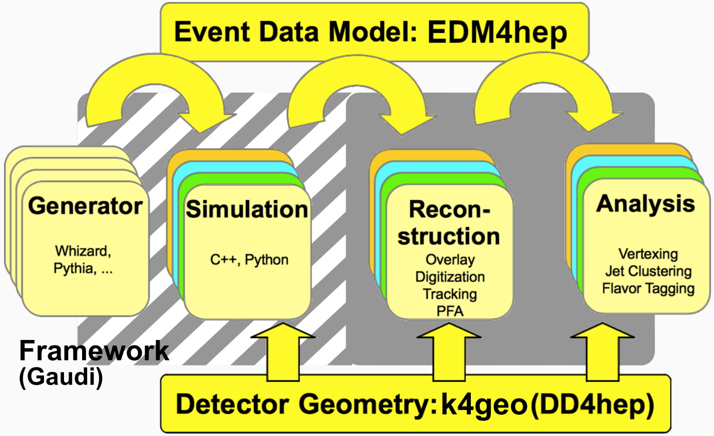
*Few testbeam packages not available

Standard workflow in iLCSoft



- Generators and simulation (`ddsim`) run outside of framework
- Reconstruction & (parts of) Analysis via `Marlin`
- **This works unchanged in a Key4hep environment!**

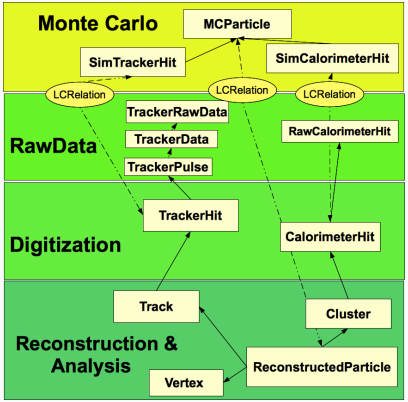
Standard workflow in Key4hep



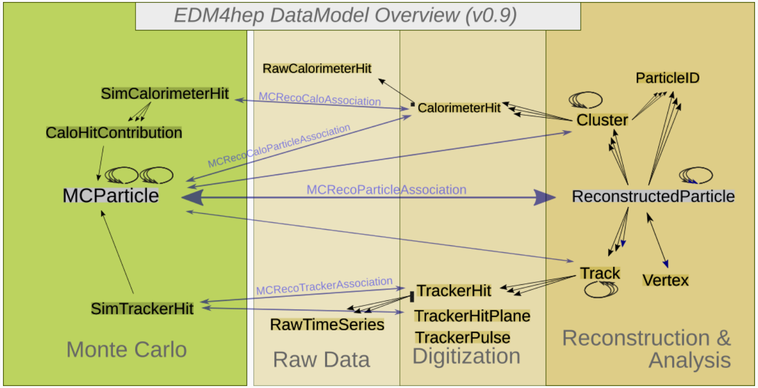
- **Conceptually the same as before** with some parts replaced
- `k4geo` (renamed from `lcgeo`) now also with first FCC geometries
- Gaudi in principle supports running Gen & Sim in FW (postponed in Key4hep)

LCIO vs EDM4hep (at the highest level)

LCIO

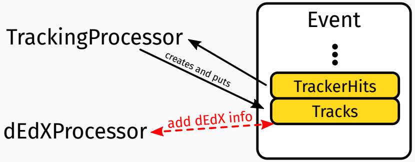


EDM4hep



- Since EDM4hep is based on LCIO the high-level structure is very similar
- Some differences in philosophy and implementation
- **Conversion in both directions available in Key4hep** as a library

LCIO vs EDM4hep (the philosophical differences)



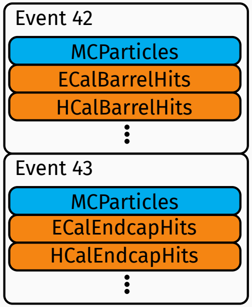
Mutability of collections and objects

- **LCIO:** Always, **EDM4hep:** Only during creation

→ Conceptual changes necessary for some workflows

→ Need to adapt EDM4hep schema

- Trivial thread-safety, improved provenance

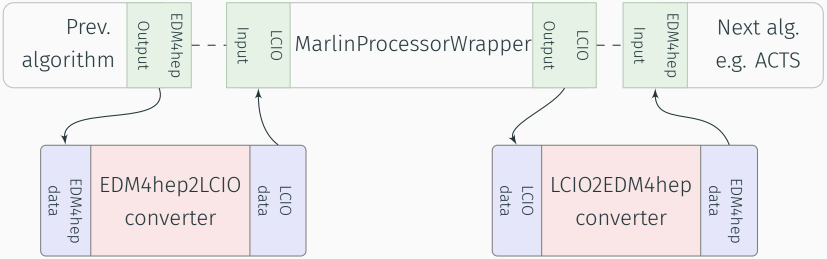


Possible event contents

- **LCIO:** Completely independent,
  EDM4hep: Consistency enforced by ROOT
- Minor impact; standard workflows switched to consistent contents

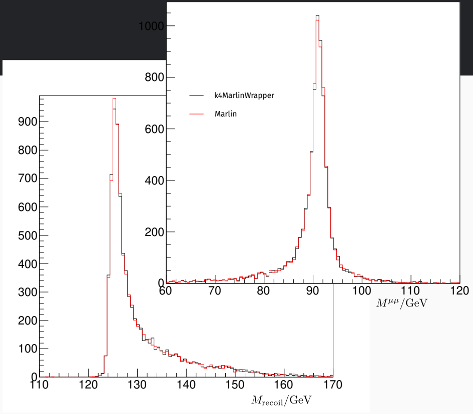
k4MarlinWrapper

- 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**

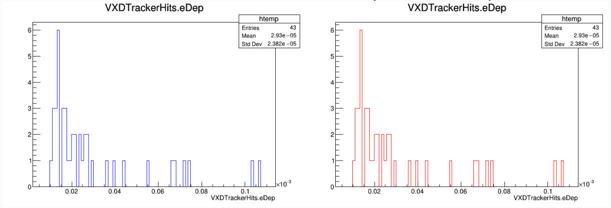


k4MarlinWrapper validation

- Full ILD, CLIC & CLD reconstruction chains have been run
    - EDM4hep inputs to EDM4hep outputs working
- Physics validation via Higgs recoil for ILD
- Converters with standalone tests
- **Ready to be used for "real" work**
    - Feedback is vital

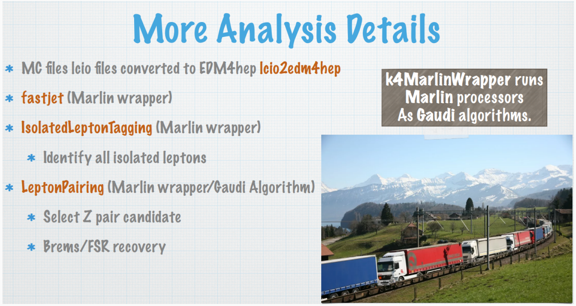


courtesy of B. Stacey



k4MarlinWrapper validation

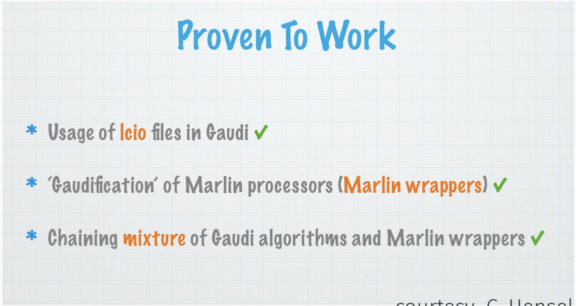
- Full ILD, CLIC & CLD reconstruction chains have been run
    - EDM4hep inputs to EDM4hep outputs working
- Physics validation via Higgs recoil for ILD
- Converters with standalone tests
- **Ready to be used for "real" work**
    - Feedback is vital



More Analysis Details

- MC files lcio files converted to EDM4hep lcio2edm4hep
- fastjet (Marlin wrapper)
- IsolatedLeptonTagging (Marlin wrapper)
    - Identify all isolated leptons
- LeptonPairing (Marlin wrapper/Gaudi Algorithm)
    - Select Z pair candidate
    - Brems/FSR recovery

k4MarlinWrapper runs Marlin processors As Gaudi algorithms.



Proven To Work

- Usage of lcio files in Gaudi ✓
- 'Gaudification' of Marlin processors (Marlin wrappers) ✓
- Chaining mixture of Gaudi algorithms and Marlin wrappers ✓

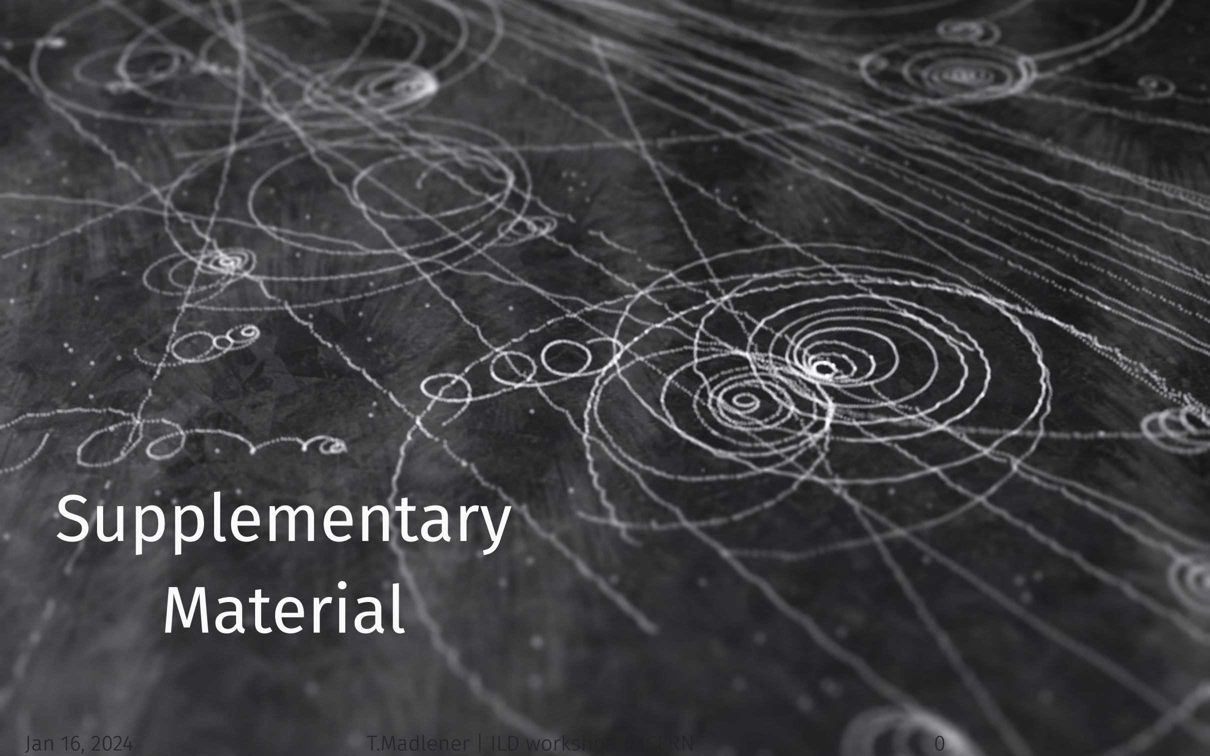
courtesy. C. Hensel

Next steps in practice

- Switch standard reco to Gaudi & MarlinWrapper (+ EDM4hep output)
  - Update (& automatize) validation to use EDM4hep
  - Use Key4hep components for new *new developments*
- Start migration of some High Level Reco (e.g. FlavorTagging) tools towards "native" Key4hep
  - Test for EDM4hep schema
  - Facilitate sharing / comparison with FCC & others
  - **(Re)-emphasize commitment of ILD to Key4hep**
- Integrate new common tools
  - e.g. ACTS for tracking (TPC possible?)
- Changes to production system & infrastructure?
- **Manpower?**

Summary

- As a community building tool Key4hep has been very successful
- **iLCSoft (components) are an integral part of Key4hep**
    - Things that work now, will also work in Key4hep
- k4MarlinWrapper and EDM converters can be / are being used for “real work”
    - **We need feedback!** (good or bad)
- Start of migration towards Key4hep components is possible



Supplementary Material

Adaption status of other projects

- FCC
  - FCCAnalyses as (high-level) analysis framework (not really using EDM4hep!)
  - Detector migration / implementation in k4geo ongoing
  - Using Key4hep as basis for developments and some productions
- CEPC
  - Started from iLCSoft but now fully switched to Gaudi & EDM4hep
  - Recent addition of drift chamber study datatypes
  - Many developments but slightly opaque
- EIC
  - Use `Jana2` framework, but try to develop framework agnostic algorithms
  - `EDM4eic` extends EDM4hep $\rightarrow$ Plan to upstream generally useful types
- MuColl
  - Partially forked iLCSoft $\rightarrow$ Plan to upstream their changes
  - Running reconstruction with Gaudi, using EDM4hep for analysis

LCIO → EDM4hep standalone converter

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

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

  - For all details see README
  - Available in recent nightly builds
- Support all features that are necessary for ILD

Marlin vs Gaudi

- Conceptually the two frameworks are very similar
  - Schedule different working units
  - Marshall data
- Most obvious differences in naming conventions
  - As always some differences emerge when looking at the details

	Marlin	Gaudi
language	C++	C++
working unit	Processor	Algorithm
config language	XML	Python
transient data format	LCIO	anything
set up function	`init`	`initialize`
work function	`processEvent`	`execute`
wrap up function	`end`	`finalize`

Key4hep packages

- `k4FWCore` 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
- `k4MarlinWrapper` Marlin proc. wrapper key4hep/k4MarlinWrapper
- Many packages migrated from FCCSW to Key4hep
  - `k4SimGeant4` for Geant4 simulation integration HEP-FCC/k4SimGeant4
  - `k4Gen` for generic generator interface HEP-FCC/k4Gen
  - ...
- Ongoing work to integrate more components
  - ACTS tracking framework acts-project/acts | key4hep/k4ActsTracking
  - CLUE fast clustering algorithms .cern.ch/kalos/CLUE | key4hep/k4CLUE