The Key4hep software stack: Beyond Future Higgs factories

Andre Sailer[1], Benedikt Hegner[1], Clement Helsens[2], Erica Brondolin[1], Frank-Dieter Gaede[3], Gerardo Ganis[1], Graeme A Stewart[1], Jiaheng Zou[4], Juraj Smiesko[1], Placido Fernandez Declara[1], Sang Hyun Ko[5], Sylvester Joosten[6], Tao Lin[4], Teng Li[7], Thomas Madlener[3], Valentin Volkl[1], Weidong Li[4], Wenxing Fang[4], Wouter Deconinck[8], Xingtao Huang[7], and Xiaomei Zhang[4]

[1]CERN, Switzerland
[2]KIT, Germany
[3]Deutsches Elektronen-Synchrotron DESY, Germany
[4]IHEP Beijing, China
[5]Seoul National University, Korea
[6]Argonne National Laboratory, Lemont, Illinois, USA
[7]Shandong University, China
[8]University of Manitoba, Winnipeg, Manitoba, Canada

E-mail: `valentin.volkl@cern.ch, andre.philippe.sailer@cern.ch`

Abstract.
The Key4hep project aims to provide a turnkey software solution for the full experiment lifecycle, based on established community tools. Several future collider communities (CEPC, CLIC, EIC, FCC, and ILC) have joined to develop and adapt their workflows to use the common data model EDM4hep and common framework. Besides sharing of existing experiment workflows, one focus of the Key4hep project is the development and integration of new experiment independent software libraries. Ongoing collaborations with projects such as ACTS, CLUE, PandoraPFA and the OpenDataDector show the potential of Key4hep as an experiment-independent testbed and development platform. In this talk, we present the challenges of an experiment-independent framework along with the lessons learned from discussions of interested communities (such as LUXE) and recent adopters of Key4hep in order to discuss how Key4hep could be of interest to the wider HEP community while staying true to its goal of supporting future collider designs studies.

1. Introduction

The Key4hep project offers a complete event processing framework for future collider experiments. It combines existing state-of-the art tools with new development in one software stack, seeking to strike a balance between flexibility and inter-usability: ideally all participating communties would opt to use the same components, allowing all participants to profit from new developments and maintain a shared codebase with minimimal duplication. However, this model of participation puts many constraints on each particular participant and may require unreasonable effort if there is a large existing codebase to be supported (as is the case with CLIC and ILC using the Marlin framework). In this particular case, a full adoption of Key4hep was achieved with a compatibility layer between frameworks (k4MarlinWrapper). In other cases, a looser integration of experiment software into the Key4hep ecosystem is possible and still mutually beneficial: detector components implemented in DD4hep, the common data model EDM4hep, visualization tooling, build systems and infrastructure can all be be shared apart from the event processing framework.

Existing experiment-independent software packages make up much of the core of the Key4hep software stack: Gaudi [1] as the event processing framework, DD4hep [2] for geometry information, Geant4 [3], Gaussino [4]) and Delphes [5] for simulation, as well as a number of event generators. There are some central new development within Key4hep. One core creation is the common event data model EDM4hep [6], which is using the podio [7] toolkit. To integrate the existing packages, bindings were added to the common framework: k4SimDelphes, k4SimGeant4, and k4Gen. Detector models are implemented in k4geo. And finally, event reconstruction and analysis tools were interfaced. The CEPC, CLIC, FCC, and ILC communities for future Higgs factories are already in various stages of adoption of this turnkey software stack [8, 9]. In these proceedings we look at other current usages of Key4hep, showing that it can be a useful tool for experiments beyond future Higgs factories.

2. Challenges for an Experiment Independent Framework

One of the main challenges faced by an experiment independent, or inter-experiment, framework, is to ensure the compatibility of developments with all participants. For example the event data model developments must not break existing workflows, or only for very good reasons. Here the *schema evolution* features of podio (see these proceedings), will at least ensure backward compatibility with existing files. However, it is mandatory for all communities to come to a consensus. The open developments of Key4hep on GitHub (`github.com/key4hep`) and the regular meetings (`https://indico.cern.ch/category/11461/`) allow all voices to be heard.

Similarly, technical compatibility of new developments has to be ensured. The spack package manager [10], used to build the Key4hep stack [11], allows us to build new developments, including upstream and downstream packages, so that issues can be easily spotted. The use of the spack package manager also lets everyone build the entire software stack by themselves, so that each community can deploy software releases according to their own time constraints and requirements.

To ensure not only technical, but also physics performance, a validation framework for Key4hep is under development, which will allow all communities to integrate their key performance indicators into a continuous validation system.

3. Recent Developments in Key4hep

Besides sharing of existing experiment workflows, one focus of the Key4hep project is the development and integration of new experiment independent software libraries. Ongoing collaborations with projects such as ACTS [12], CLUE [13, 14], PandoraPFA [15] and the OpenDataDector [16, 17] show the potential of Key4hep as an experiment-independent testbed and development platform, and allow different communities to reap the benefits of new solutions, without the need for different specific implementations. For the ACTS integration, recent developments to allow attaching ACTS specific information to arbitrary DD4hep based geometries will streamline the integration of this track reconstruction framework. The application of the CLUE calorimeter reconstruction algorithm for different high granularity calorimeters for CMS, or CLIC like detectors also proves the effectiveness of providing generic solutions.

Another example is the application of the Phoenix [18] event display. Fig. 1 shows the event display for a DD4hep based detector using EDM4hep event data, converted to json via an

EDM4hep utility, leveraging podio functionality. A workflow that is viable for all experiments integrated in Key4hep.

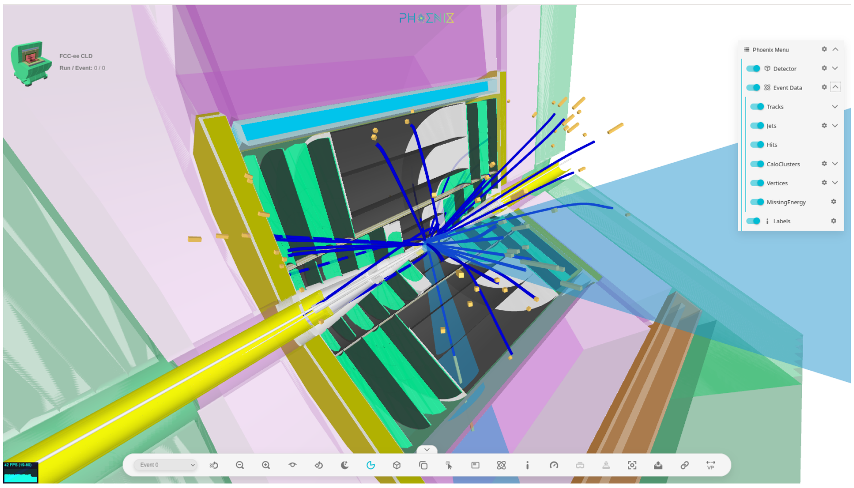

Figure 1. EDM4hep Event display using the Phoenix visualization tool.

4. Applications of Key4hep in other experiments

Key4hep has originally been motivated by the need to develop and support software for design studies for post-LHC Higgs factories, in particular FCC, CLIC and ILC. The common software stack has nevertheless applications outside of these particular communities. In this section, we report the current adoption of Key4hep in high energy physics.

4.1. Key4hep and the Electron-Ion Collider

The Electron-Ion Collider (EIC) has chosen DD4hep for its geometry description, and uses DD4hep also for Geant4 simulations. It has also adopted podio to create an EDM4hep-based data model, adapted to its streaming readout, but uses a different event processing framework: JANA2 [19] in place of Gaudi (Fig. 2). The common use of EDM4hep allows a broad collaboration nonetheless, and developments concerning a compatibility layer between Gaudi and JANA2 are ongoing.

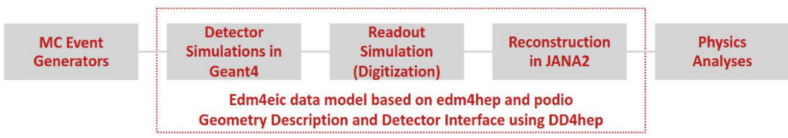

Figure 2. Diagram of steps and components of the Electron-Ion Collider software framework.

4.2. Key4hep and the Muon Collider

The Muon Collider Design Study already uses large parts of iLCSoft for full simulations (Fig. 3). Discussion on full adoption of Key4hep are ongoing. Similar to the transition by the CLIC and ILC communities, the k4MarlinWrapper compatibility layer can be used to seamlessly combine legacy iLCSoft components with new Key4hep components [20]. However, the large amount of background particles encountered in Muon Collider collisions will put a higher memory pressure on the reconstruction workflow, which makes the in-memory translation from EDM4hep to LCIO and vice-versa potentially too costly. Therefore, it will be necessary for some algorithms to be ported from Marlin to Gaudi for this reason alone.

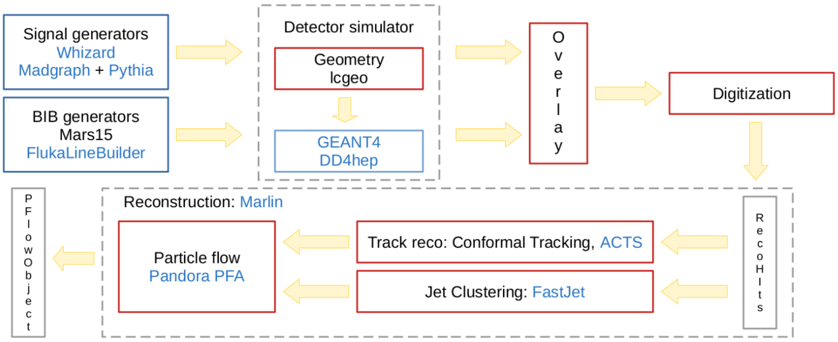

Figure 3. Diagram of steps and components of the Muon collider software framework.

4.3. Usage of Key4hep in other experiments: the LUXE example

Apart from the future collider projects there is also some interest to use Key4hep from smaller experiments. One of these is the LUXE experiment [21], planned to be built at DESY Hamburg aiming to study non-perturbative QED effects in collisions of high-energy electrons or photons with a laser. From a conceptual point of view, LUXE is somewhat comparable to a test beam target experiment with multiple instruments, including spectrometers to analyse the residue of the collisions. The experiment will cover a wide range of physical phase space, resulting in experimental conditions that will result in order of magnitude differences in detector occupancies.

Currently discussions are ongoing on which parts of Key4hep can and should be adapted by LUXE. While several things are rather obvious choices, e.g., DD4hep for detector description, or the general adaption of common workflows for building software via spack, other components require more consideration. Among these is the adaption of EDM4hep, which provides datatypes for describing, e.g. the detector measurements from the positron tracking device. However, currently it is completely lacking any representation of the measurements that would result from, e.g., a 2D scintillator screen, imaged through a high resolution camera, that is used on the high rate side of the spectrometer, or also measurements from a planned detector leveraging Cherenkov radiation. Here some prototyping will have to be done before a decision can be made on whether these datatypes could also be of general use in EDM4hep.

5. Conclusion

Beyond the existing collaboration members and interested future collider experiments, Key4hep can be adapted to the needs of other planned and existing experiments. Especially small experiments may profit from a ready-to-use software solutions, for example, LUXE. The shared development efforts across multiple smaller communities offers clear benefits, despite possible challenges, that can be overcome through social or technical solutions.

Acknowledgments

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (`https://cds.cern.ch/record/2649646/`, CERN-OPEN-2018-006). This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 101004761, and it has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 871072.

References

[1] Barrand G *et al.* 2001 *Comput. Phys. Commun.* **140** 45–55
[2] Frank M, Gaede F, Grefe C and Mato P 2013 *J. Phys. Conf. Ser.* **513** 022010
[3] Agostinelli S *et al.* 2003 *Nucl. Inst. & Meth.* **A506** 250–303
[4] Siddi B G and Müller D 2019 *2019 IEEE NSS/MIC* pp 1–4
[5] de Favereau J, Delaere C, Demin P, Giammanco A, Lemaître V, Mertens A and Selvaggi M (DELPHES 3) 2014 *JHEP* **02** 057 (*Preprint* 1307.6346)
[6] Gaede F, Ganis G, Hegner B, Helsens C, Madlener T, Sailer A, Stewart G A, Volkl V and Wang J 2021 *EPJ Web Conf.* **251** 03026
[7] Gaede F, Hegner B and Mato P 2017 *J. Phys. Conf. Ser.* **898** 072039 URL https://cds.cern.ch/record/2296801
[8] Fang W, Fernandez Declara P, Gaede F D, Ganis G, Hegner B, Helsens C, Huang X, Ko S H, Madlener T, Li T, Li W, Lin T, Sailer A, Stapf B S, Volkl V, Wang J and Zou J 2023 *J. Phys. Conf. Ser.* **2438** 012049
[9] Fernandez Declara P *et al.* 2022 *PoS* **EPS-HEP2021** 844
[10] Gamblin T, LeGendre M, Collette M R, Lee G L, Moody A, de Supinski B R and Futral S 2015 Supercomputing 2015 (SC'15) (Austin, Texas, USA) lLNL-CONF-669890 URL https://github.com/spack/spack
[11] Volkl V, Madlener T, Lin T, Wang J, Konstantinov D, Razumov I, Sailer A and Ganis G 2021 *EPJ Web Conf.* **251** 03056
[12] Ai X, Allaire C, Calace N *et al.* 2022 *Comput. Softw. Big Sci.* **6** 8
[13] Brondolin E and on behalf of the CMS collaboration 2023 *J. Phys. Conf. Ser.* **2438** 012074 URL https://dx.doi.org/10.1088/1742-6596/2438/1/012074
[14] Brondolin E 2022 k4clue https://github.com/key4hep/k4Clue
[15] Marshall J and Thomson M 2015 *Eur. Phys. J.* **C75** 439
[16] Allaire C, Gessinger P, Hdrinka J, Kiehn M, Kimpel F, Niermann J, Salzburger A and Sevova S 2022 Opendatadetector URL https://doi.org/10.5281/zenodo.6445359
[17] Gessinger-Befurt P, Salzburger A and Niermann J 2023 *J. Phys. Conf. Ser.* **2438** 012110 URL https://dx.doi.org/10.1088/1742-6596/2438/1/012110
[18] Ali F, Moyse E, Khan M H, Labra E C, Pappas A, Smiesko J, Sharma G, Hennequin A, Couturier B, Bianchi R M, Amarjeet, Bhosale A, Ponce S and Soleti S R 2023 Hsf/phoenix: v2.12.0 URL https://doi.org/10.5281/zenodo.7535649
[19] Lawrence D, Boehnlein A and Brei N 2020 *EPJ Web Conf.* **245** 01022 URL https://doi.org/10.1051/epjconf/202024501022
[20] Fernandez Declara P, Sailer A, Petric M and Volkl V 2021 key4hep/k4marlinwrapper: v00-03-01 URL https://doi.org/10.5281/zenodo.4719245
[21] Abramowicz H *et al.* 2021 *Eur. Phys. J. ST* **230** 2445–2560 (*Preprint* 2102.02032)