Key4hep: Progress Report on Integrations

Erica Brondolin[1], *Juan Miguel* Carceller[1], *Wouter* Deconinck[2], *Wenxing* Fang[3], *Brieuc* Francois[1], *Frank-Dieter* Gaede[4], *Gerardo* Ganis[1], *Benedikt* Hegner[1], *Clement* Helsens[1,5,*], *Xingtao* Huang[6], *Sylvester* Joosten[7], *Sang Hyun* Ko[8], *Tao* Lin[3], *Teng* Li[6], *Weidong* Li[3], *Thomas* Madlener[4], *Leonhard* Reichenbach[1,9], *André* Sailer[1**], *Swathi* Sasikumar[1], *Juraj* Smiesko[1], *Graeme A* Stewart[1], *Alvaro* Tolosa-Delgado[1], *Valentin* Volkl[1], *Xiaomei* Zhang[3], and *Jiaheng* Zou[3]

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

Abstract. Detector studies for future experiments rely on advanced software tools to estimate performance and optimize their design and technology choices. The Key4hep project provides a flexible turnkey solution for the full experiment life-cycle based on established community tools such as ROOT, Geant4, DD4hep, Gaudi, podio and spack. Members of the CEPC, CLIC, EIC, FCC, and ILC communities have joined to develop this framework and have merged, or are in the progress of merging, their respective software environments into the Key4hep stack.

These proceedings will give an overview over the recent progress in the Key4hep project: covering the developments towards adaptation of state-of-the-art tools for simulation (DD4hep, Gaussino), track and calorimeter reconstruction (ACTS, CLUE), particle flow (PandoraPFA), analysis via RDataFrame, and visualization with Phoenix, as well as tools for testing and validation.

1 Introduction

Detector studies for future experiments require advanced software tools to optimize their design and technology choices and to estimate their performance. These advanced software tools must include the possibility for full or parameterized detector simulation, the reconstruction of tracks and calorimeter clusters, jet clustering, flavour tagging, and analysis. A combined solution for all these issues should also allow experiments to move seamlessly from different stages of their life cycle: for example, from parameterized detector studies to find a performance envelope of their experiment to full and detailed simulation and reconstruction

*Now at EPFL, Lausanne, Switzerland
**e-mail: andre.philippe.sailer@cern.ch

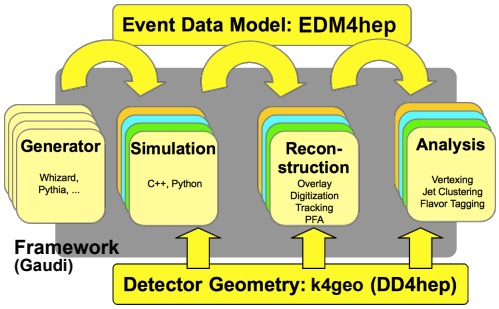


Figure 1: Main ingredients for the Key4hep project: geometry information, event data information, and a processing framework with a large number of algorithms.

studies to confirm the validity and feasibility of the assumptions used in the parameterized simulation. The consistency of the framework also allows experiments to extract performance parameterizations from detailed simulation studies to create large scale samples that are not feasible for small communities with limited computing resources available to them.

The Key4hep project provides a solution for these use-cases by means of a structured software stack, which integrates individual packages towards a complete data processing framework for HEP experiments. The sharing of common components will reduce the overhead faced by different communities otherwise. Moreover, Key4hep aims to provide an easy-to-use product for librarians, who provide software installations, the developers creating or adapting components, and the endusers.

Figure 1 schematically shows the three major ingredients for the project: a processing framework that connects all the pieces; a way to describe the geometry of the experiments and use the information for simulation, reconstruction or analysis; and an event data model to exchange data between the pieces, or for persistency. For Key4hep the processing framework is Gaudi [1], the geometry information is provided via DD4hep [2, 3], and the event data model is provided by EDM4hep [4–6] and podio [7, 8].

The current contributors and users of the Key4hep ingredients are part of the CEPC, CLIC, EIC [9], ILC, FCC, and Muon collider communities. The source code for all components under direct development of the Key4hep project is hosted on GitHub[1]. Weekly open meetings are held to discuss ongoing developments and issues in Key4hep. Newcomers are always welcome to join the meetings or contribute to the developments.

As the goal of the project is not to develop all components itself, but as much as possible reuse existing solutions, this paper, in Section 2, describes the current status of the integration of tools for simulation, reconstruction and analysis, then shows, in Section 3, how some of the testing for the project is done, before it ends with a summary and outlook in Section 4.

2 Integrations

One part of the integrations into Key4hep are the existing experiment software components from CEPCSW and FCCSW, which are already based on Gaudi. Their adaptation to Key4hep required mostly adaptations to the EDM4hep event data model [10–12]. For the integration of iLCSoft, used by the ILC and CLIC communities, the k4MarlinWrapper [13] was created to integrate *processors* from the Marlin framework [14] and its corresponding event data model

[1] https://github.com/key4hep

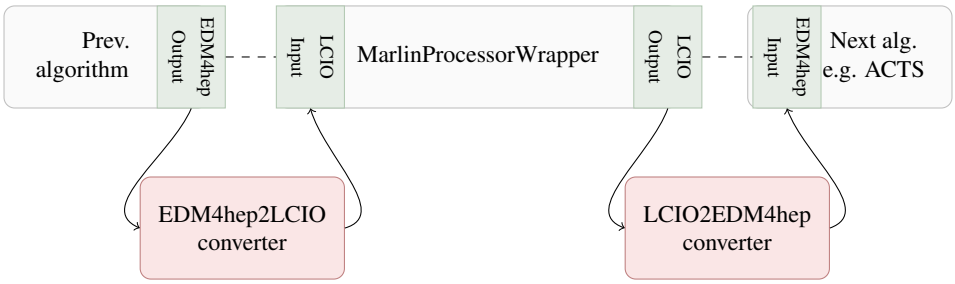


Figure 2: Schematic of how Marlin processors are integrated into Gaudi workflows in Key4hep.

LCIO [15, 16] into Gaudi. The idea of the k4MarlinWrapper is shown in Figure 2. To run any *processor* from Marlin in a Gaudi workflow, the event data is converted in memory from EDM4hep to LCIO before the execution of the processor and back to EDM4hep after.

In the following sections, the status of the integration for simulation, reconstruction and analysis tools is laid out.

2.1 Simulation

For full or parameterized simulation, different tools are already available in Key4hep. The parametric simulation program Delphes [17], together with some utilities to handle the generation of primary events, has been integrated into Gaudi as k4SimDelphes [18]. k4SimDelphes also contains standalone programs for different input files, such as HepMC, or controlling event generators, such as Pythia8. There are two possibilities for full detector simulation with Geant4 [19] and the DD4hep geometry. There is the ddsim [20] feature of DD4hep, which can produce EDM4hep output files and read a majority of generator output formats. In addition, there is the k4SimGeant4 [21] Gaudi integration, which came out of FCCSW. The framework integration of the full simulation via k4SimGeant4 – together with the other algorithms – allow one to run a complete chain from event generation to reconstructed objects in a single program execution.

In the meantime, also the Gaussino [22] functionalities from LHCb have become experiment agnostic [23] and will potentially provide a complete replacement of the k4SimGeant4 package.

2.2 Reconstruction

2.2.1 Tracking

The iLCSoft ecosystem contains tools for track pattern recognition and track fitting, all of which can be used in Key4hep via the k4MarlinWrapper For the track fitting `DD4hep::rec::Surfaces` are attached to any sensitive element [24] and some dead material. These surfaces provide an abstract view of the geometry needed for reconstruction, such as automatically averaged materials (as shown in Figure 3) and measurement directions. In many cases, these surfaces can be pragmatically attached to an existing DD4hep

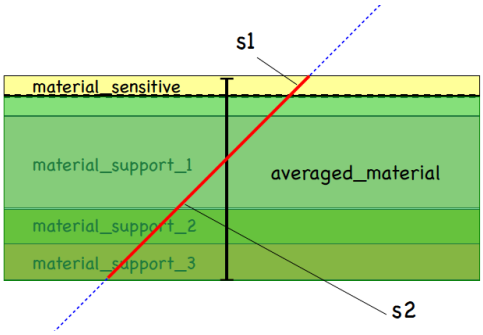


Figure 3: Sketch showing the averaging of materials when creating `DD4hep::rec::Surfaces`.

based geometry using run-time plugins. This easily enables track reconstruction for different detectors.

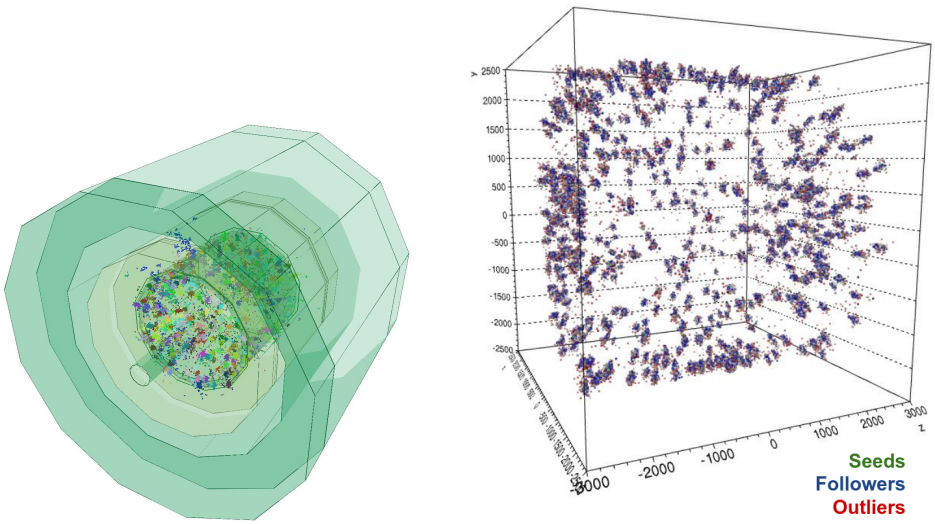
The ACTS [25, 26] integration into Key4hep progresses steadily. In the last months, a plugin to support the EDM4hep track format was added to ACTS [27]. Support for DD4hep geometry is under active development as well. It is already possible to load DD4hep geometries following a certain hierarchical structure into ACTS. The existing conversion of DD4hep to ACTS geometry is currently heavily used by the ACTS developers to test their algorithms against the generic Open Data Detector, a detector model used for benchmarking tracking and calorimeter reconstruction approaches [28]. However, the general use of `DD4hep::rec::Surfaces` is under development and will allow a broader range of detectors to be directly used. Using these surfaces also in the link to the ACTS reconstruction will enable a direct replacement of the iLCSoft track reconstruction with ACTS. Recently, the development of the necessary Gaudi algorithms to use ACTS from Key4hep has intensified. In particular, the focus is on enabling arbitrary track refits using different fitters via Gaudi.

2.2.2 Calorimeter Clustering

An important ingredient for the performance of future Higgs Factory experiments is the particle flow reconstruction for optimal jet energy resolutions. The Pandora particle flow algorithm package (PandoraPFA) [29, 30] was developed to study particle flow clustering at linear colliders.

The particle flow clustering with Pandora makes use of the extensions attached to detector geometries [24], such as `DD4hep::rec::LayeredCalorimeterData`, to provide the properties of the calorimeter, e.g., radiation length, interaction length, and dimensions to the reconstruction algorithms. To support a larger range of detectors, for example those that foresee a noble liquid calorimeter [31], the necessary information will be obtained in a more dynamic way. This step was materialized using the `DD4hep::MaterialManager` to extract the necessary information between arbitrary space points.

At least for high granularity calorimeters with large occupancies, the reconstruction time can become dominant. For the HGCAL project of CMS a GPU friendly algorithm, CLUE (CLUstering of Energy), was developed [32, 33]. An integration of this algorithm in Key4hep is ready to be used [34]. Figure 4 shows simulated hits from many photons in a single event, and how CLUE reconstructs them into clusters.



(a) Simulated photon clusters shown with CED (C-Event display, see Section 2.4).

(b) Reconstructed clusters from CLUE.

Figure 4: Simulated photons and their clusters reconstructed with k4clue.

```
theDataFrame
# define an alias for electron index collection
.Alias("Electron_idx", "Electron_objIdx.index")
# define the electron collection
.Define("electrons",  "ReconstructedParticle::get(Electron_idx, ReconstructedParticles)")
#select electrons on pT
.Define("selected_electrons", "ReconstructedParticle::sel_pt(10.)(electrons)")
# ...
.Define("zed_leptonic_recoil_m","ReconstructedParticle::get_mass(zed_leptonic_recoil)")
# create branch with leptonic charge
.Define("zed_leptonic_charge","ReconstructedParticle::get_charge(zed_leptonic)")
# Filter at least one candidate
.Filter("zed_leptonic_recoil_m.size()>0")
```

Listing 1: Example for the usage of EDM4hep objects for analysis with RDataFrame [36].

2.3 Analysis

The ROOT persistency of the EDM4hep event data model and its columnar storage model allows the use of the RDataFrame [35] feature for analysis. Listing 1 shows an example how the EDM4hep objects, here `ReconstructedParticles`, can be used with a dataframe to select events with different criteria. To take full advantage of the EDM4hep datamodel based on podio, for example to handle relationships between different objects, an `RDataSource` is being developed.

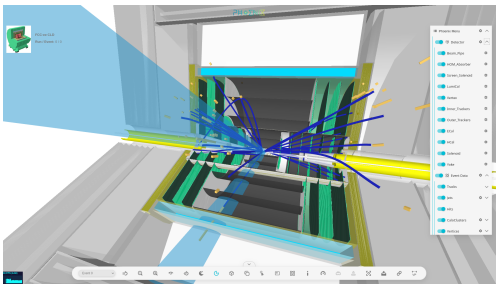


Figure 5: An event display of the CLD detector using Phoenix.

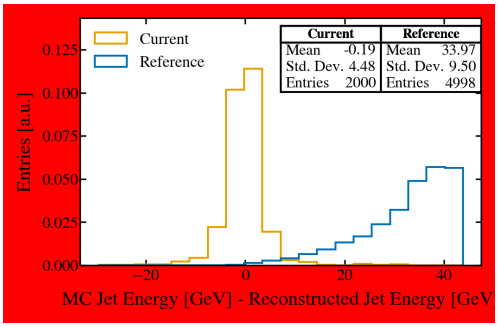


Figure 6: Bright red highlighting differing distributions. *Current* is after a bug was fixed.

2.4 Visualization

The DD4hep geometry, and the EDM4hep event data can both be converted to formats suitable for the Phoenix event display [37]. Figure 5 shows an event in the CLD detector for the FCCee [38], which is also shown in Figure 4a using the C-Event display (CED) from iLCSoft. The advantage of Phoenix is the possibility of using a web-browser, and its broad configurability. Phoenix allows one to centrally host the detectors on the web. The FCC detectors, for example, are hosted on a webserver[2] and continuously updated.

3 Testing

A continuous validation system has been set up for Key4hep. Every night, after the nightly build of the Key4hep stack is finished, detector simulation is done based on the preset detectors in the configuration. After that, a complete reconstruction is performed. The results of the reconstruction are then compared to a set of reference samples that were produced in known and reproducible conditions. If the new distributions are different from the reference ones, based on specified metrics, then the display of the plots in a webpage will point this out (see Figure 6). The plots in the webpage are classified in different categories depending on which class of results they belong to, for example plots about tracks in one category, those about jet reconstruction in another. The current system is evolving and several improvements are under development, such as easier configuration, more detectors being tested and a better reporting system when the new and reference distributions are different.

Besides the physics performance, the CPU performance has to be monitored and controlled as well. For this purpose the *Valprod* toolkit is under development, which enables the building of comprehensive validation jobs and offers CPU flame graph, I/O profiling and an integration with the prmon [39] program.

4 Summary & Outlook

The Key4hep project provides a common framework for future Higgs factories and other experiments and has been fully adopted by FCC and CLIC. It sees increasing adoption also from the ILC and CEPC communities. Beyond these initial communities, the project has caught interest of the EIC, Muon Collider communities, and LUXE experiment [40]. To

[2] https://fccsw.web.cern.ch/fccsw/phoenix/

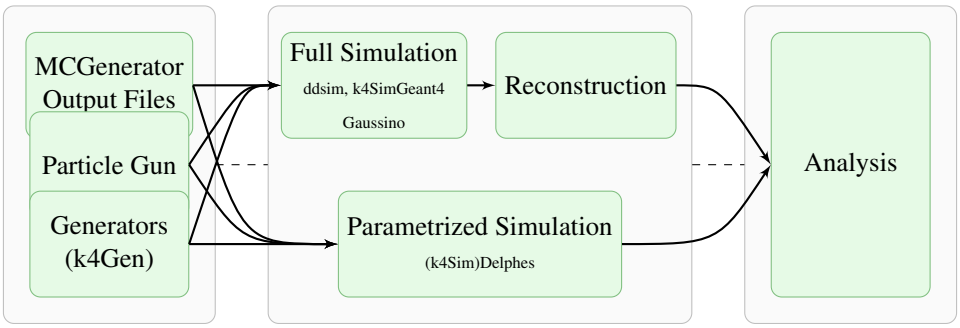


Figure 7: Potential data flows via full or fast simulation.

match the needs of the communities, the software stack is expanding to state-of-the-art tools such as ACTS, PandoraPFA, CLUE, or Phoenix. Their integrations, as outlined in the previous sections, that are or will be available soon as part of the Key4hep stack, will allow its users to perform all the tasks needed for detector studies, as shown in Figure 7.

Acknowledgements

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (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. This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 871072. This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research (grant no. 13E18CHA).

References

[1] G. Barrand et al., Comput. Phys. Commun. **140**, 45 (2001), 10.1016/S0010-4655(01)00254-5

[2] M. Frank, F. Gaede, M. Petric, A. Sailer, *AIDASoft/DD4hep*, 10.5281/zenodo.592244

[3] M. Frank, F. Gaede, C. Grefe, P. Mato, J. Phys. Conf. Ser. **513**, 022010 (2013), 10.1088/1742-6596/513/2/022010

[4] V. Volkl, T. Madlener, F. Gaede, A. Sailer, C. Helsens, P.F. Declara, G.A. Stewart, W. Deconinck, J. Smiesko, L. Forthomme et al., *key4hep/EDM4hep*, 10.5281/zenodo.4785062

[5] F. Gaede, G. Ganis, B. Hegner, C. Helsens, T. Madlener, A. Sailer, G.A. Stewart, V. Volkl, J. Wang, EPJ Web Conf. **251**, 03026 (2021), 10.1051/epjconf/202125103026

[6] F. Gaede, T. Madlener, P. Declara Fernandez, G. Ganis, B. Hegner, C. Helsens, A. Sailer, G. A. Stewart, V. Voelkl, PoS **ICHEP2022**, 1237 (2022), 10.22323/1.414.1237

[7] F. Gaede, B. Hegner, P. Mato, J. Phys. Conf. Ser. **898**, 072039 (2017), 10.1088/1742-6596/898/7/072039

[8] F. Gaede, B. Hegner, G.A. Stewart, EPJ Web Conf. **245**, 05024 (2020), 10.1051/epjconf/202024505024

[9] D. Lawrence, *EIC Software Overview*, in *CHEP 2023* (Norfolk, Virginia, USA, 2023), https://indico.jlab.org/event/459/contributions/11457/

[10] V. Volkl, G. Ganis, B. Hegner, C. Helsens, A. Sailer, E. Brondolin, J. Smiesko, F. Gaede, T. Madlener, W. Fang et al., PoS **ICHEP2022**, 234 (2022), 10.22323/1.414.0234

[11] P. Fernandez Declara et al., PoS **EPS-HEP2021**, 844 (2022), 10.22323/1.398.0844

[12] W. Fang, P. Fernandez Declara, F.D. Gaede, G. Ganis, B. Hegner, C. Helsens, X. Huang, S.H. Ko, T. Madlener, T. Li et al., J. Phys. Conf. Ser. **2438**, 012049 (2023), 10.1088/1742-6596/2438/1/012049

[13] P. Fernandez Declara, A. Sailer, V. Volkl, T. Madlener, *key4hep/k4MarlinWrapper*, 10.5281/zenodo.4719244

[14] F. Gaede, Nucl. Inst. & Meth. **A559**, 177 (2006), 10.1016/j.nima.2005.11.138

[15] F. Gaede, T. Behnke, N. Graf, T. Johnson, eConf **C0303241**, TUKT001 (2003), physics/0306114, 10.48550/arXiv.physics/0306114

[16] N. Graf, *LCIO Turns 20*, in *CHEP 2023* (Norfolk, Virginia, USA, 2023), https://indico.jlab.org/event/459/contributions/11536/

[17] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, M. Selvaggi (DELPHES 3), JHEP **02**, 057 (2014), 1307.6346, 10.1007/JHEP02(2014)057

[18] T. Madlener, V. Volkl, C. Helsens, M. Chrząszcz, F. Gaede, *key4hep/k4SimDelphes*, 10.5281/zenodo.4564682

[19] S. Agostinelli et al., Nucl. Inst. & Meth. **A506**, 250 (2003), 10.1016/S0168-9002(03)01368-8

[20] M. Petric, M. Frank, F. Gaede, S. Lu, N. Nikiforou, A. Sailer, J. Phys. Conf. Ser. **898**, 042015 (2016), 10.1088/1742-6596/898/4/042015

[21] A. Zaborowska, V. Volkl, J. Pöttgen, M. Selvaggi, Z. Drasal et al., *HEP-FCC/k4SimGeant4*, 10.5281/zenodo.4564573

[22] B.G. Siddi, D. Müller, *Gaussino - a Gaudi-Based Core Simulation Framework*, in *2019 IEEE NSS/MIC* (2019), pp. 1–4, 10.1109/NSS/MIC42101.2019.9060074

[23] M. Mazurek, G. Corti, M. Clemencic, A. Morris, *From prototypes to large scale detectors: how to exploit the Gaussino simulation framework for detectors studies, with a detour into machine learning*, in *CHEP 2023* (Norfolk, Virginia, USA, 2023), https://indico.jlab.org/event/459/contributions/11528/

[24] A. Sailer, M. Frank, F. Gaede, D. Hynds, S. Lu, N. Nikiforou, M. Petric, R. Simoniello, G. Voutsinas, CLICdp and ILD collaborations, J. Phys. Conf. Ser. **898**, 042017 (2017), 10.1088/1742-6596/898/4/042017

[25] X. Ai, C. Allaire, N. Calace et al., Comput. Softw. Big Sci. **6**, 8 (2022), 10.1007/s41781-021-00078-8

[26] P. Gessinger-Befurt, A. Salzburger, J. Niermann, J. Phys. Conf. Ser. **2438**, 012110 (2023), 10.1088/1742-6596/2438/1/012110

[27] P. Gessinger-Befurt, *Flexible, robust and minimal-overhead Event Data Model for track reconstruction in ACTS*, in *CHEP 2023* (Norfolk, Virginia, USA, 2023), https://indico.jlab.org/event/459/contributions/11443/

[28] A. Zaborowska, A. Salzburger, E. Brondolin, D. Salamani, P. Gessinger, A. Stefl, *The Open Data Detector*, in *CHEP 2023* (Norfolk, Virginia, USA, 2023), https://indico.jlab.org/event/459/contributions/11546/

[29] M. Thomson, Nucl. Inst. & Meth. **A611**, 25 (2009), 10.1016/j.nima.2009.09.009

[30] J. Marshall, M. Thomson, Eur. Phys. J. **C75**, 439 (2015), 10.1140/epjc/s10052-015-3659-3

[31] B. Francois, Nucl. Inst. & Meth. **A1040**, 167035 (2022), 10.1016/j.nima.2022.167035

[32] M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez, Frontiers in big Data **3**, 41 (2020), 10.3389/fdata.2020.591315

[33] E. Brondolin, on behalf of the CMS collaboration, J. Phys. Conf. Ser. **2438**, 012074 (2023), 10.1088/1742-6596/2438/1/012074

[34] E. Brondolin, M. Rovere, F. Pantaleo, *key4hep/k4Clue*, 10.5281/zenodo.7844854

[35] D. Piparo, P. Canal, E. Guiraud, X. Valls Pla, G. Ganis, G. Amadio, A. Naumann, E. Tejedor, EPJ Web Conf. **214**, 06029 (2019), 10.1051/epjconf/201921406029

[36] *FCCAnalyses github repository*, `https://github.com/HEP-FCC/FCCAnalyses`, Accessed: 2023-09-21

[37] F. Ali, E. Moyse, M.H. Khan, E.C. Labra, A. Pappas, J. Smiesko, G. Sharma, A. Hennequin, B. Couturier, E. Dreyer et al., *HSF/phoenix*, 10.5281/zenodo.2586998

[38] N. Bacchetta, J.J. Blaising, E. Brondolin, M. Dam, D. Dannheim, K. Elsener, D. Hynds, P. Janot, A.M. Kolano, E. Leogrande et al., *CLD – a detector concept for the FCC-ee* (2019), 10.48550/ARXIV.1911.12230

[39] G.A. Stewart, A.S. Mete, *prmon: process monitor*, 10.5281/zenodo.2554202

[40] H. Abramowicz et al., *Technical design report for the LUXE experiment* (2023), 10.48550/arXiv.2308.00515