(High Level) Reconstruction Tools for ILD

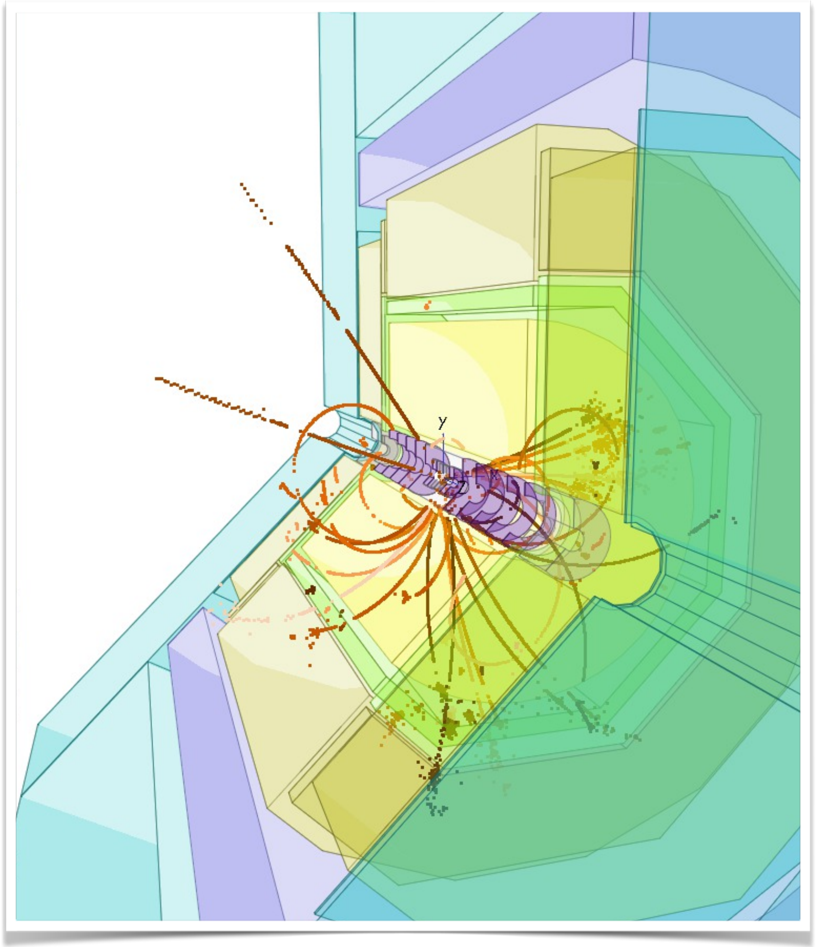
Towards “genuine” use of Key4hep ?

16.01.2024

Frank Gaede, DESY
ILD Meeting 2024, CERN

Outline

- Introduction ans Reminder
  - iLCSoft <-> Key4hep
  - DD4hep detector models and reconstruction
- Standard ILD reconstruction algorithms
- “Transition” to to Key4hep
- Recent (HL)R developments
- Conclusion and Outlook

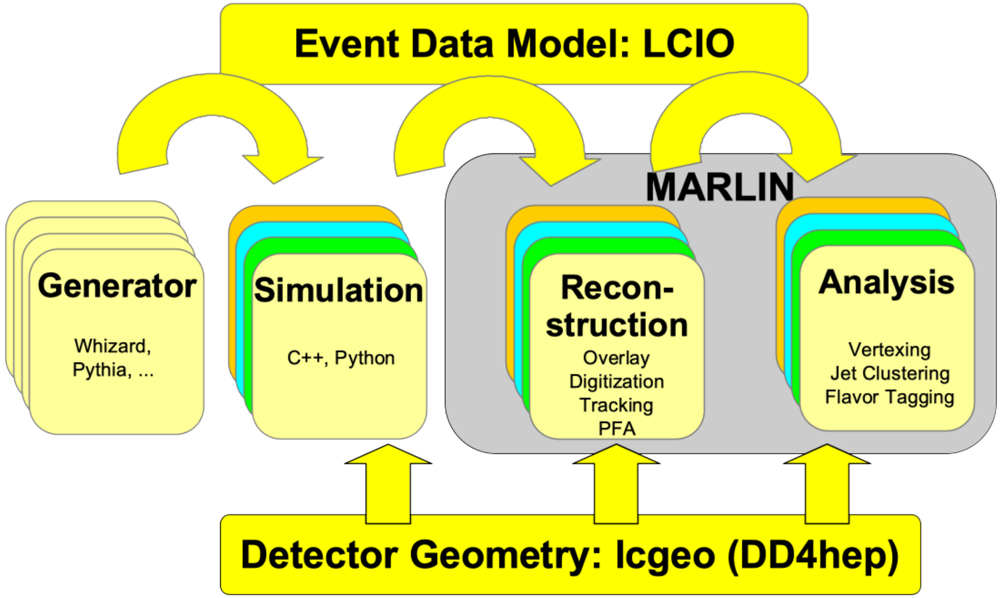


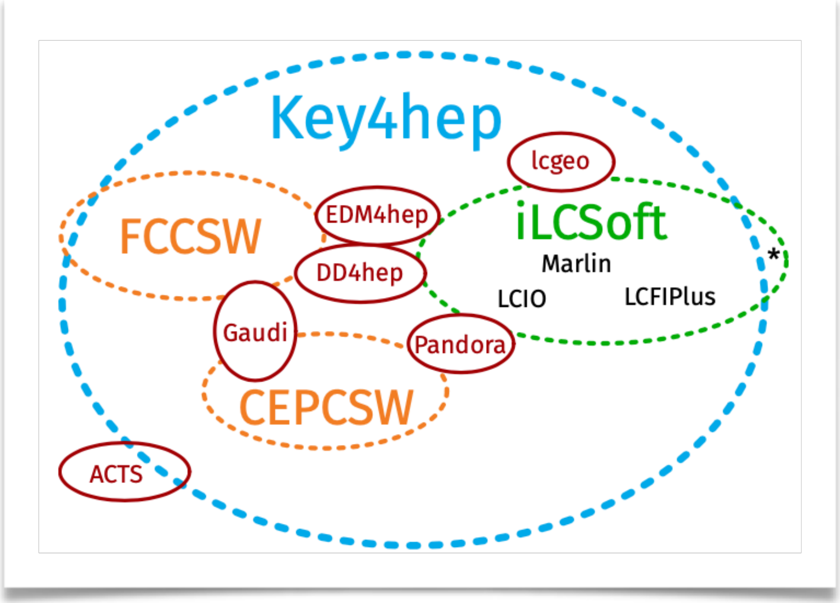
The common software vision

iLCSoft as integral part of Key4hep

- complete set of tools for
  - **generation, simulation, reconstruction, analysis**
  - build, package, test, deploy

- core ingredients of current **Key4hep**
  - **PODIO** for **EDM4hep** (based on LCIO and FCC-edm)
  - **Gaudi** framework, devel/used for (HL-)LHC
  - **DD4hep** for geometry
    - originally developed for LC now adopted by community
  - **spack** package manager



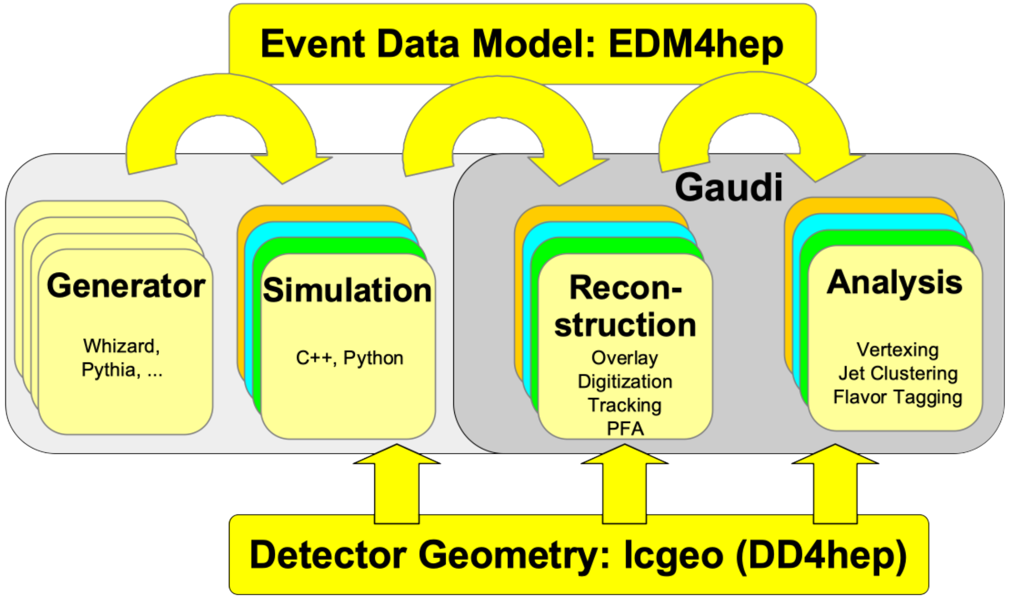


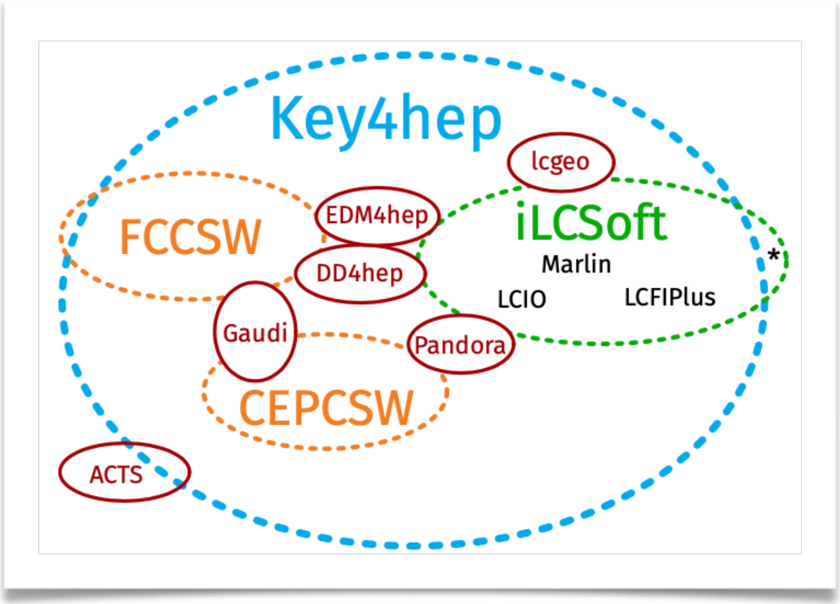
The common software vision

iLCSoft as integral part of Key4hep

- complete set of tools for
  - **generation, simulation, reconstruction, analysis**
  - build, package, test, deploy

- core ingredients of current **Key4hep**
  - **PODIO** for **EDM4hep** (based on LCIO and FCC-edm)
  - **Gaudi** framework, devel/used for (HL-)LHC
  - **DD4hep** for geometry
    - originally developed for LC now adopted by community
  - **spack** package manager

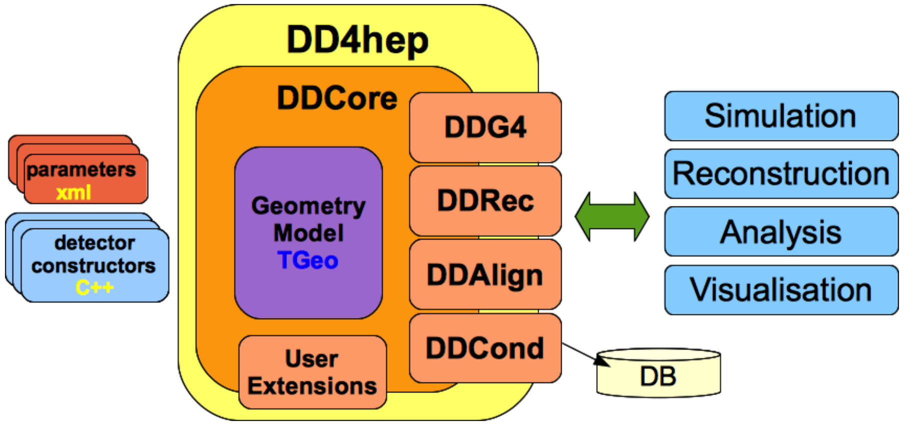




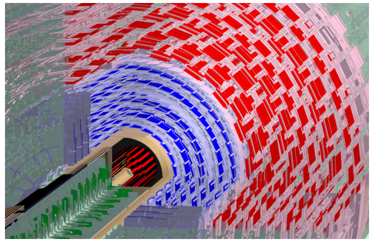
DD4hep geometry toolkit

defining the detector geometry and different views on it

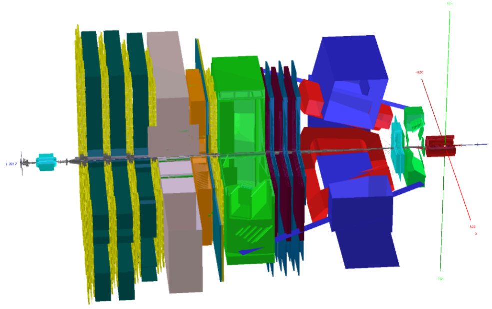
- LC community and CERN have developed a generic detector geometry system - based on best practises by ILC, CLIC, LHCb (*in AIDA, AIDA2020*)
- supporting the full life cycle of the experiment
- providing components and interfaces for
  - full simulation, reconstruction, conditions, alignment, visualisation and analysis
- adopted also by CMS and LHCb



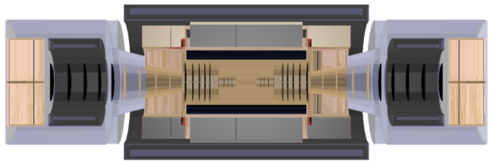
DD4hep: de facto industry standard



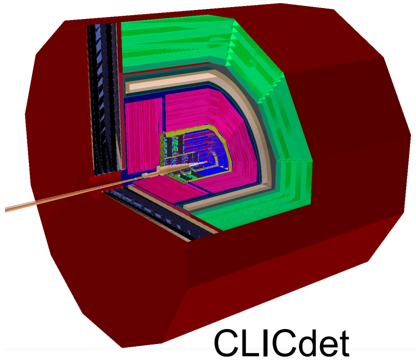
CMS



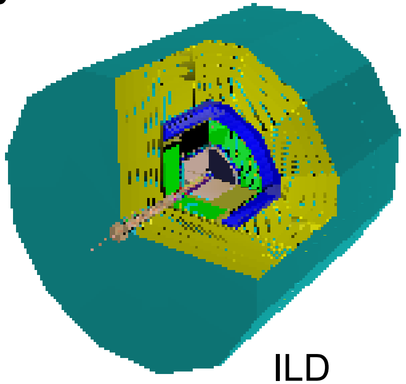
LHCb



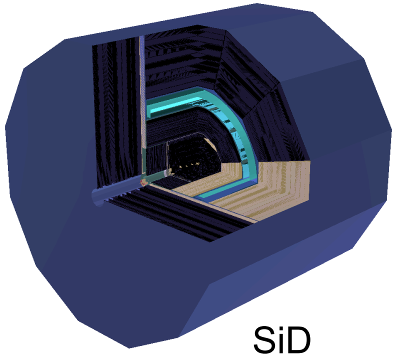
FCC-hh



CLICdet



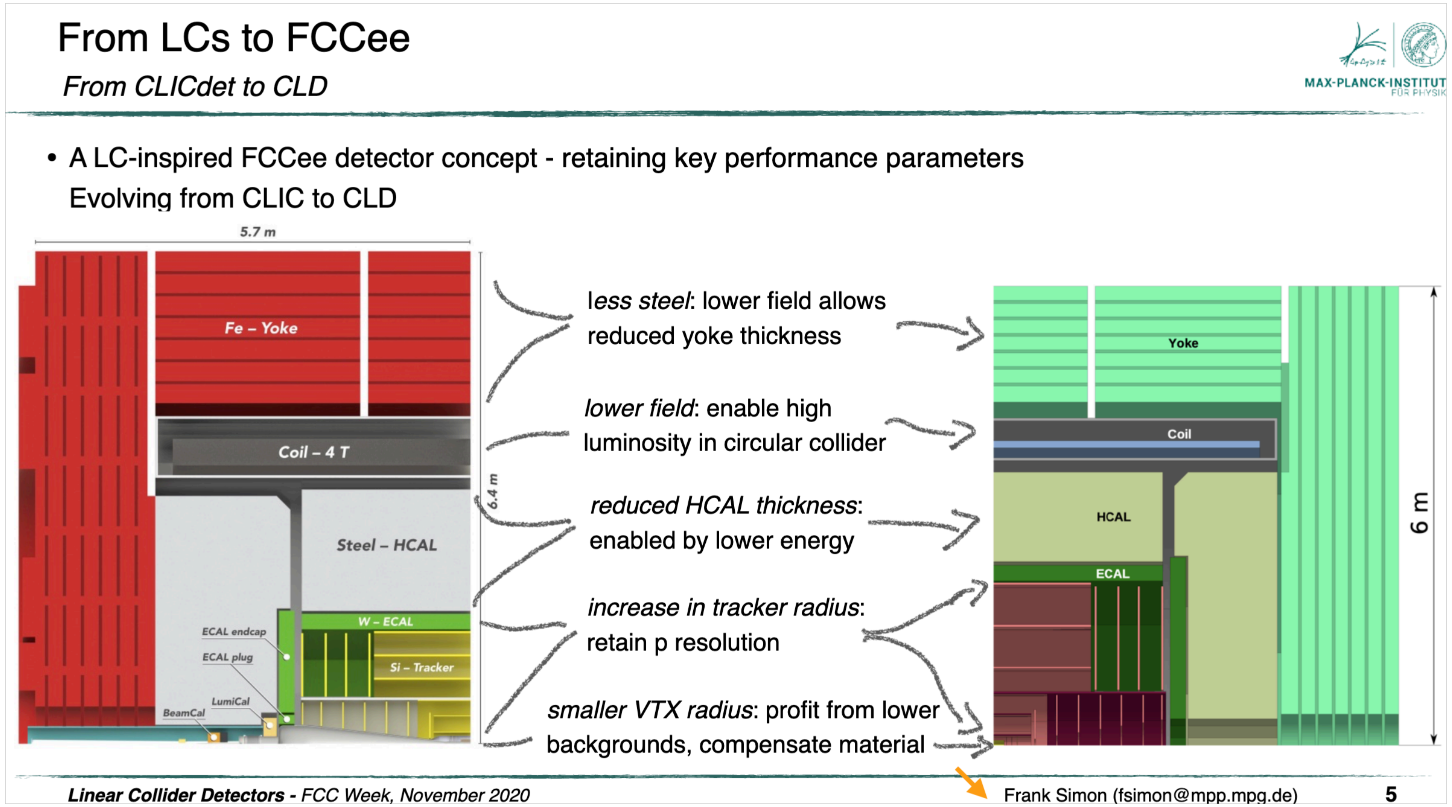
ILD



SiD

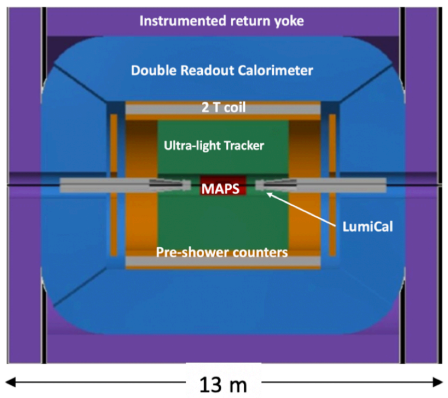
DD4hep detector models for FCCee

all Higgs factory detectors in new package *k4geo*

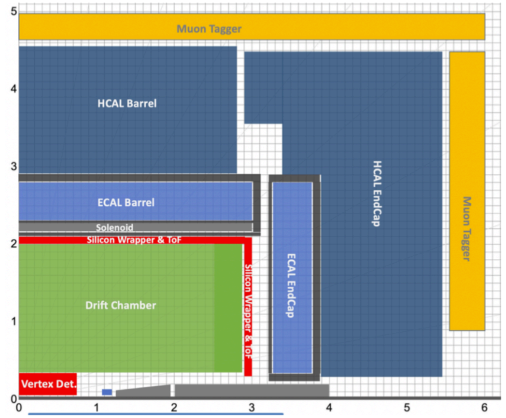


- CLD detector: based on CLICdet
  - adjusted for FCCee at lower energies and lower B-field:
  - larger tracker, thinner calorimeters,….

IDEA



Noble Liquid ECAL based

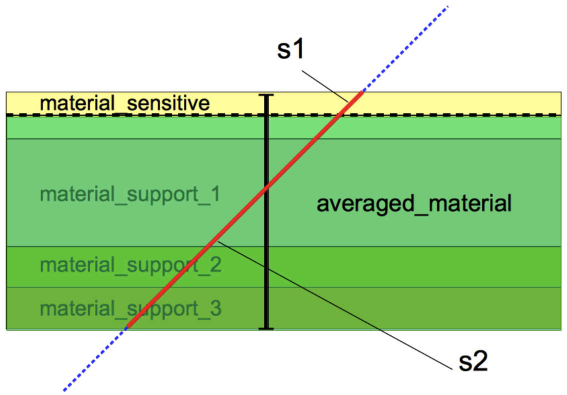


- ongoing work to implement the other two FCCee detector models in DD4hep
  - dual readout calorimeter
  - LAr/Noble Liquid calorimeter

interoperability in **Key4hep** can re-use for **ILD**:
- sub detector components
  - see: talk D. Jeans
- **(high level) reconstruction** algorithms
  - e.g. Si-Tracking or ACTS

DDRec

interfaces for reconstruction in DD4hep

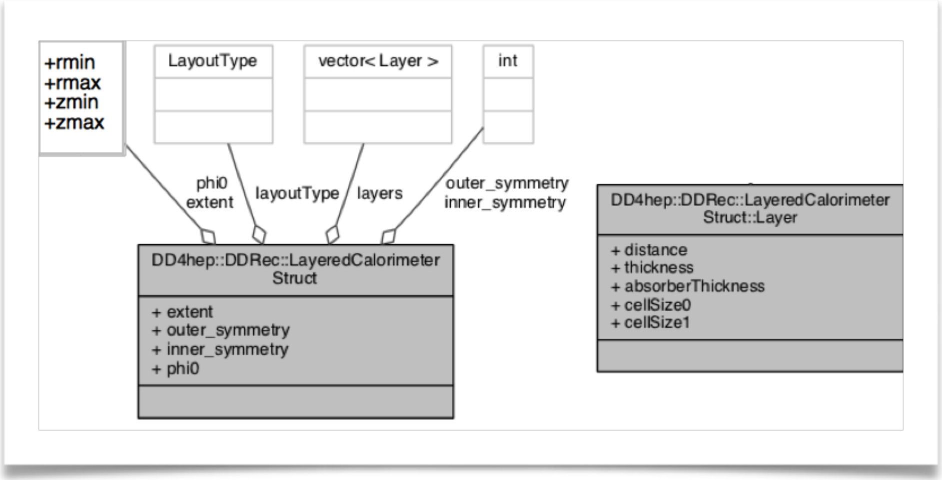


- DD4hep provides access to the detector geometry as needed in typical reconstruction algorithms in DDRec:
    - **tracking surfaces** attached to sensitive and dead material volumes in detailed model
    - material *automatically* averaged for multiple scattering and E-loss
    - measurement directions on surface

ILD inner tracker

- dedicated **high level reco API** for common sub detectors, e.g. *LayeredCalorimeterData*:
    - positions of absorber and sensitive layers
    - cell dimensions, symmetry (barrel, endcap)

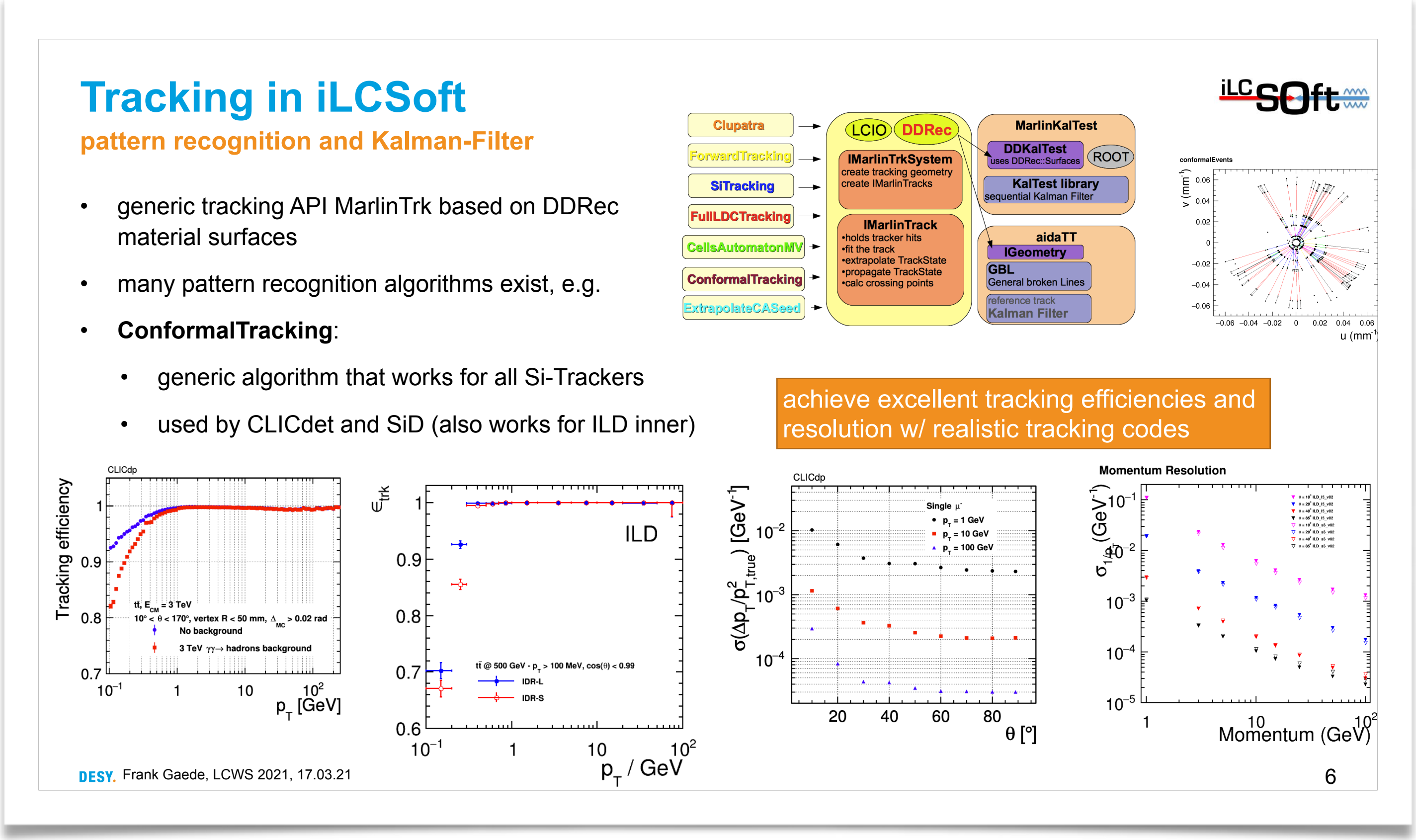
can exchange (high level) reconstruction algorithms with other detectors if they also use DDRec



large reconstruction code base in iLCSoft

Developed over >15 years for linear collider detectors - e.g. ILD

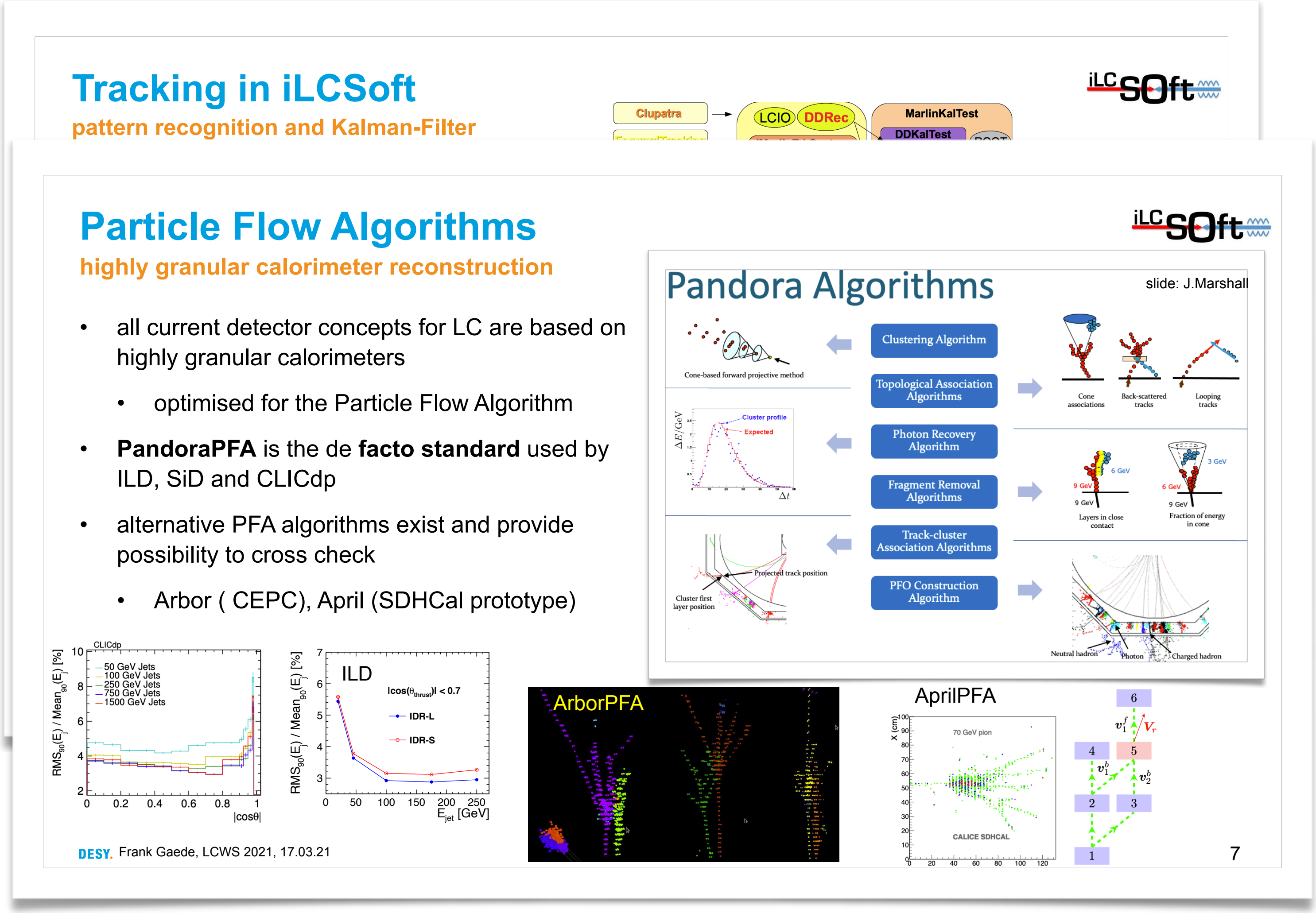
- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
  - generic API for fitting algorithms
  - large number of pattern recognition algorithms



large reconstruction code base in iLCSoft

Developed over >15 years for linear collider detectors - e.g. ILD

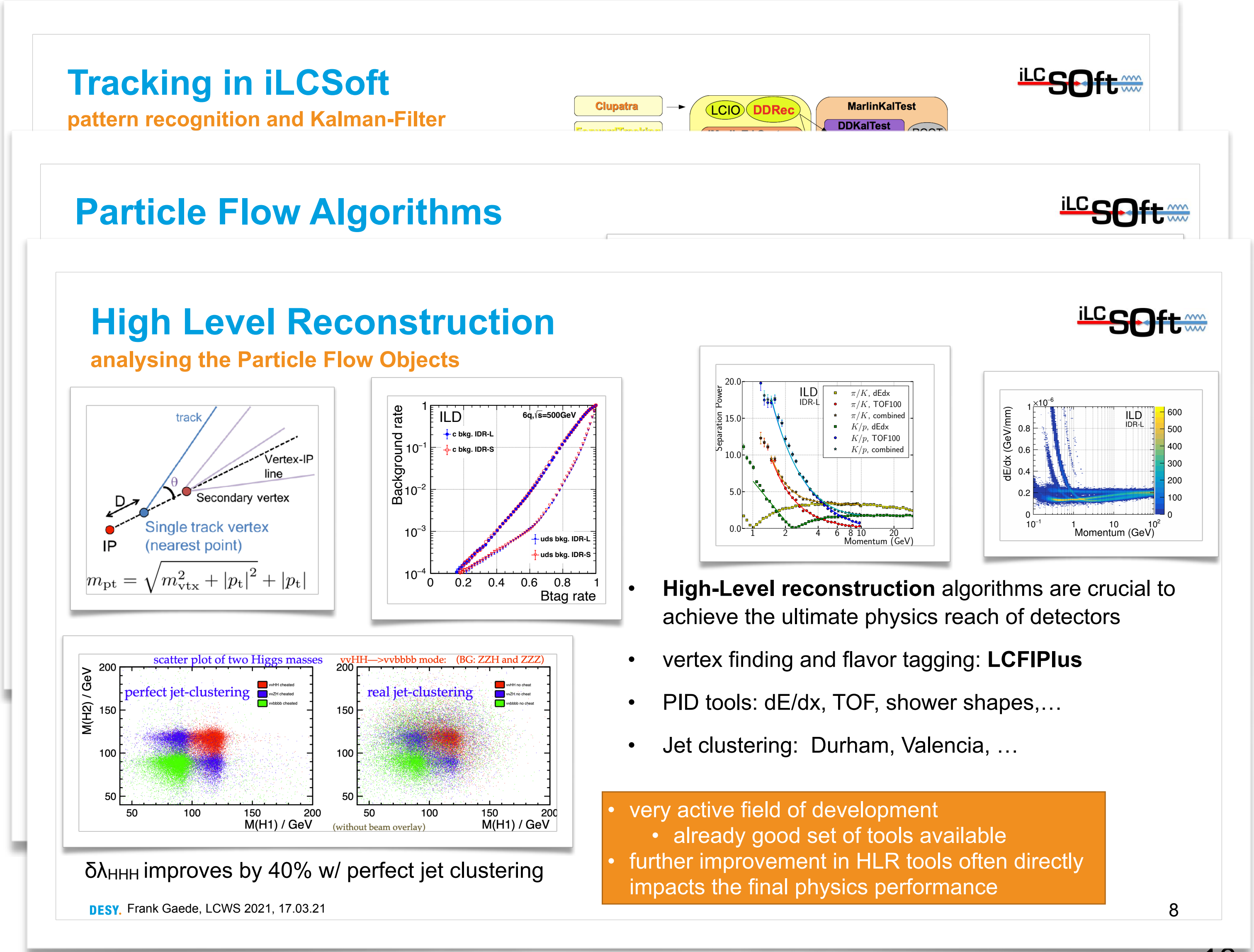
- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
    - generic API for fitting algorithms
    - large number of pattern recognition algorithms
- particle flow algorithms
    - PandoraPFA and Arbor, AprilPFA



large reconstruction code base in iLCSoft

Developed over >15 years for linear collider detectors - e.g. ILD

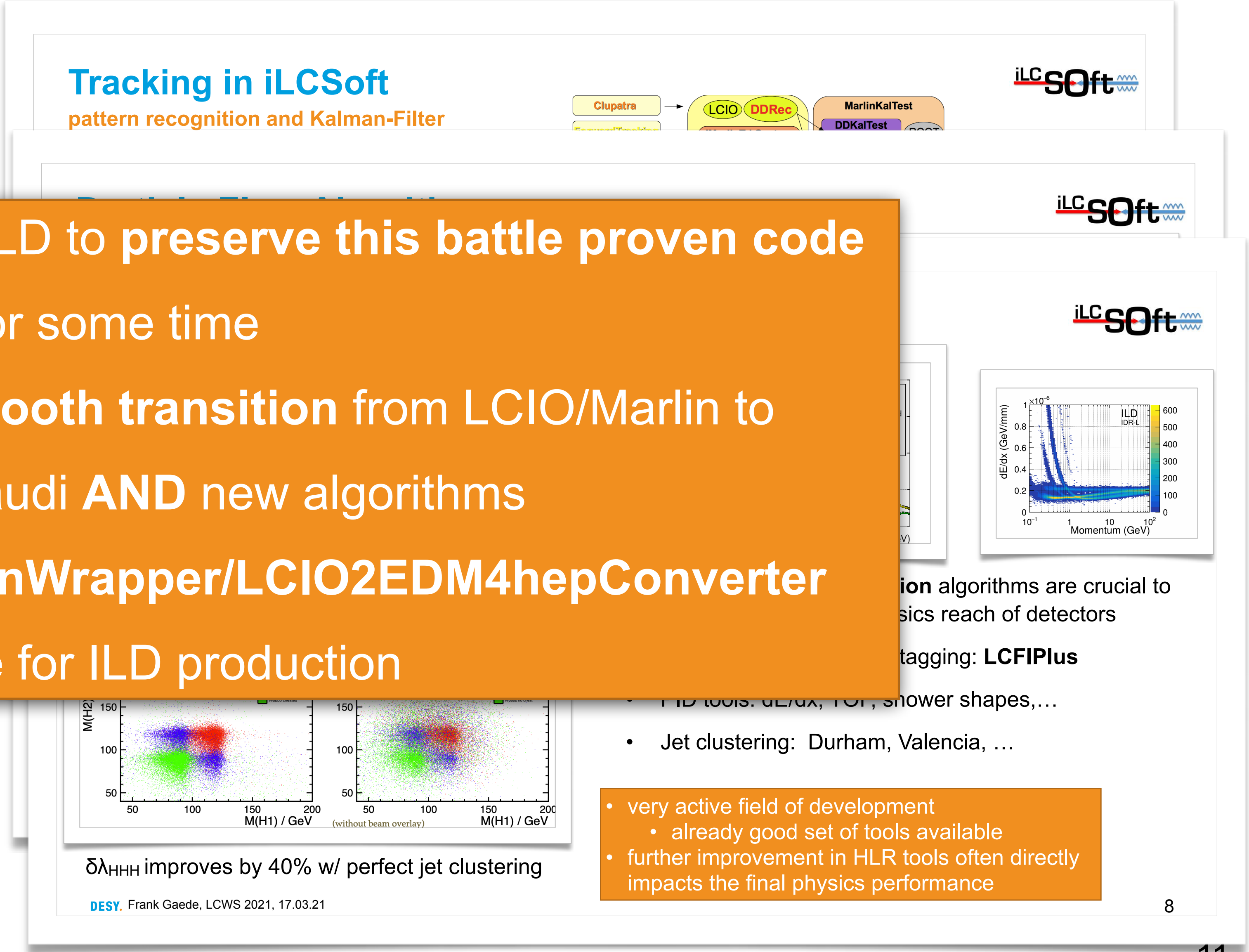
- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
  - generic API for fitting algorithms
  - large number of pattern recognition algorithms
- particle flow algorithms
  - PandoraPFA and Arbor, AprilPFA
- high level reconstruction
  - jet finding, flavor tagging, PID, TOF,…



large reconstruction code base in iLCSoft

Developed over >15 years for linear collider detectors - e.g. ILD

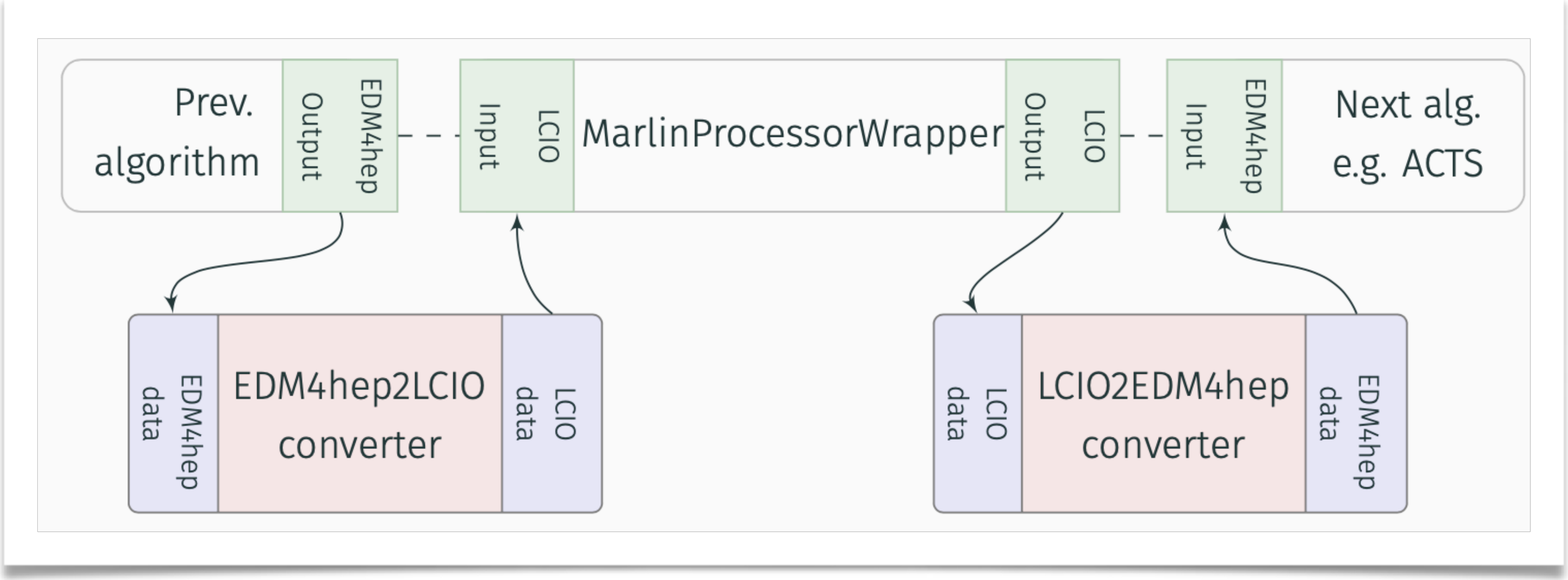
- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
  - generic API for f
  - large number of recognition algo
- particle flow algorith
  - PandoraPFA ar
- high level reconstru
  - jet finding, flavor tagging, PID, TOF,…



- it is **vital** for ILD to **preserve this battle proven code** in Key4hep for some time
- -> need a **smooth transition** from LCIO/Marlin to EDM4hep/Gaudi **AND** new algorithms
- will use **MarlinWrapper/LCIO2EDM4hepConverter** for some time for ILD production

K4MarlinWrappper

the vision: mix and match Marlin and Gaudi algorithms



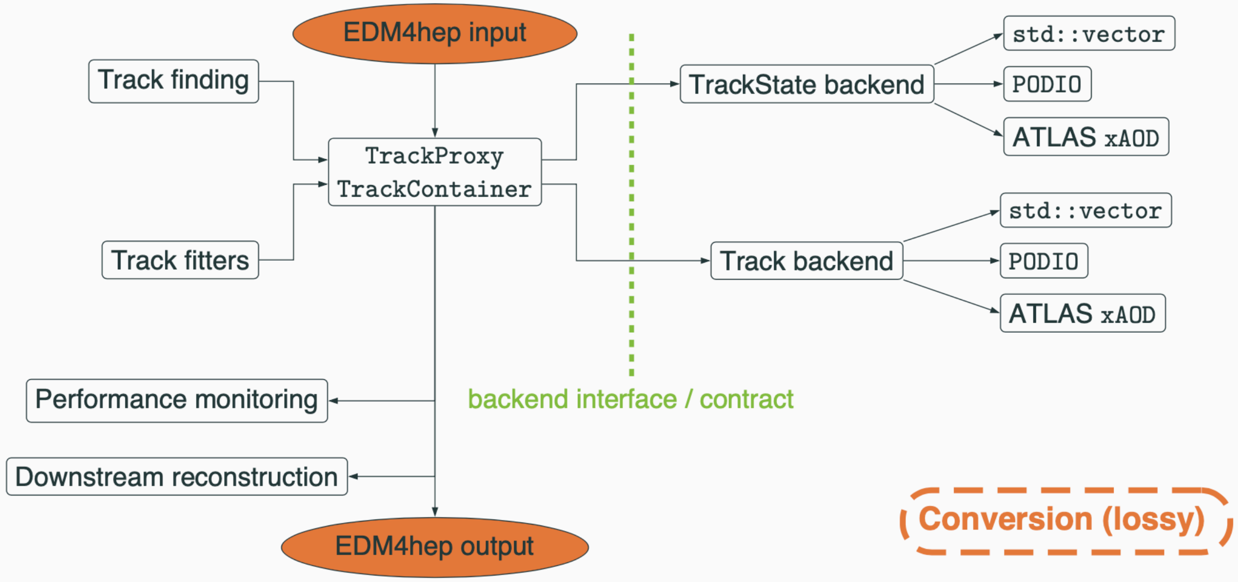
- in a transition phase algorithms developed in the new EDM4hep/Gaudi world can gradually replace older algorithms

- could start to think about actually developing new algorithms from the beginning in **Gaudi/EDM4hep** !
  - volunteers or candidates ?

ACTS

a common tracking toolkit

Architecture



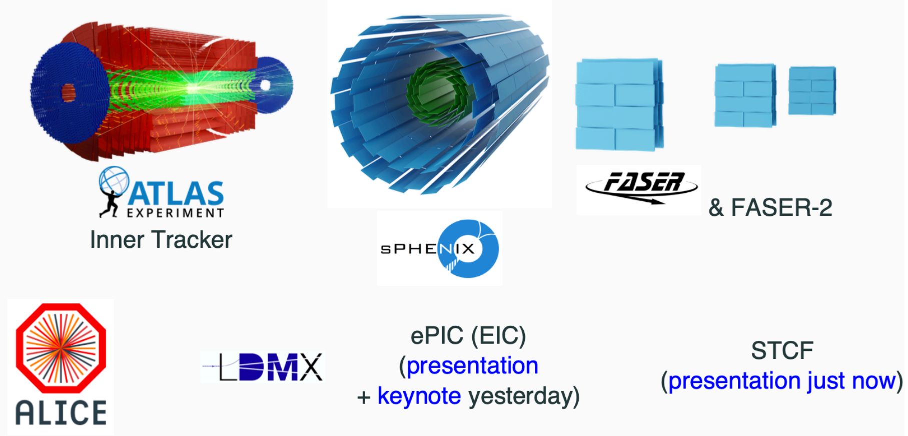
- ACTS tracking toolkit is the the current choice for track fitting (and finding !?) in Key4hep
- recently implemented interface to write out EDM4hep Tracks
  - some discussion needed w/ tracking and ACTS experts on details of the tracking data model
  - perigee vs. on-surface parameterisation …

What is ACTS?

- Experiment-independent toolkit for track reconstruction applications
- Modern architecture and code, unit tested, continuous integration
- Minimal external dependencies
- Ready for multi-threading by design

P.Gessinger, CHEP 2023

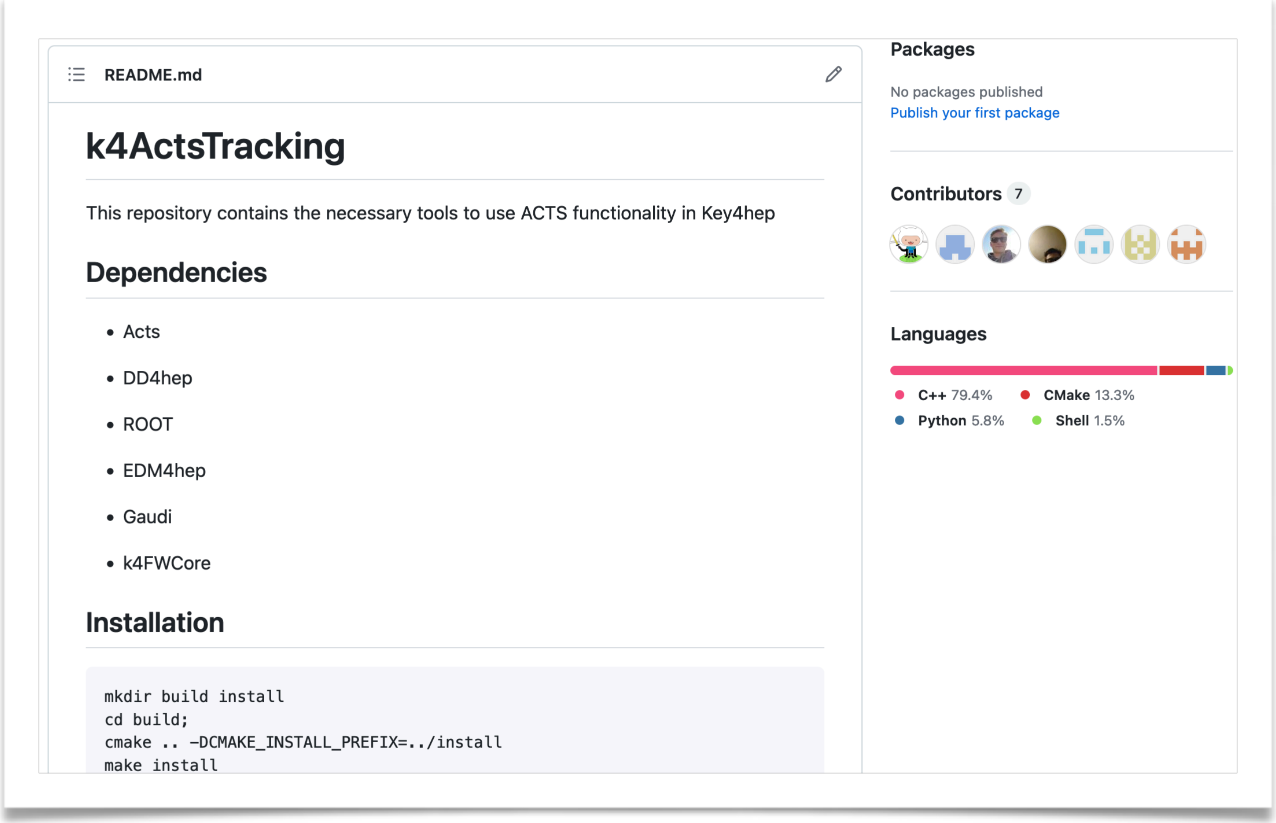
Evaluation and/or deployment by multiple experiments



k4ACTS

integration of ACTS in Key4hep

- first major reconstruction algorithm in Key4hep/ Gaudi
- ongoing work at CERN (L.Reichenbach) in context of electron reconstruction w/ ACTS for CLD
- one crucial issue for ILD is the interface to the tracking geometry
  - ACTS has interface to DD4hep to extract surface geometry
  - need to check compatibility w/ *ddrec::Surface* used in LC tracking


README.md

k4ActsTracking

This repository contains the necessary tools to use ACTS functionality in Key4hep

Dependencies

- Acts
- DD4hep
- ROOT
- EDM4hep
- Gaudi
- k4FWCore

Installation

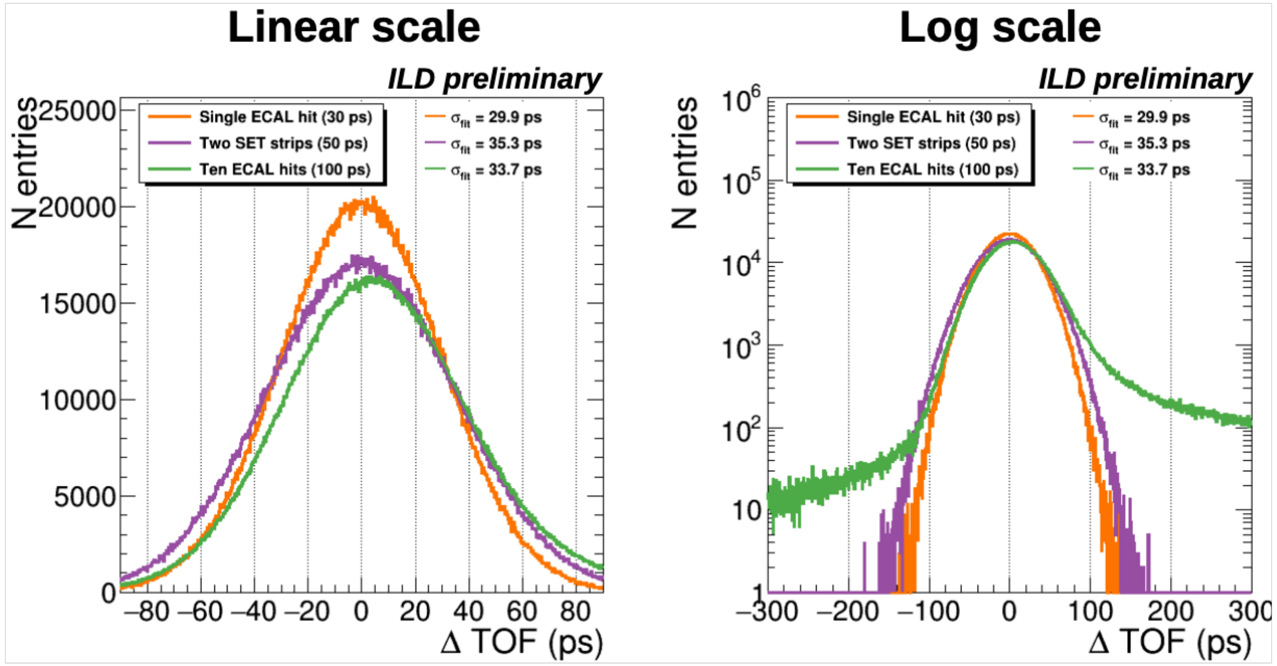
```
mkdir build install
cd build;
cmake .. -DCMAKE_INSTALL_PREFIX=../install
make install
```


- might eventually benefit from this implementation
- expect significant effort to adapt to ILD tracking (w. TPC)
- probably more a midterm project …

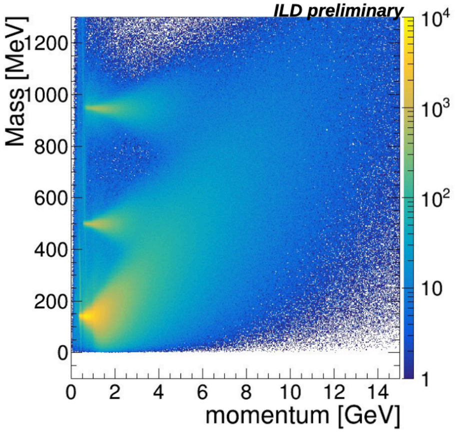
TOF Estimators

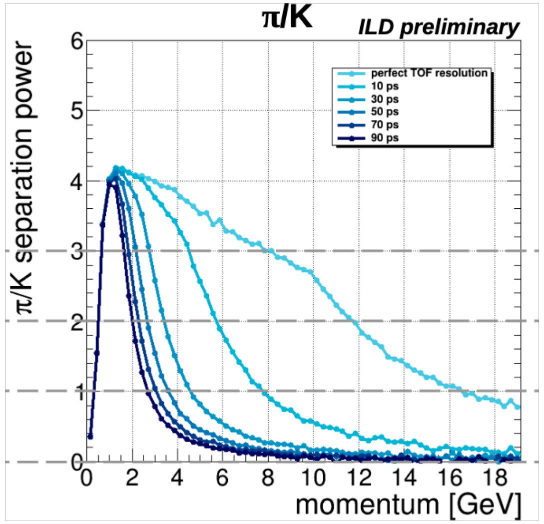
enhance particle identification

- studied various options and potential improvements for TOF in ILD
- track length calculation is important
- **need to decide on ILD goal - and strategy for TOF !**

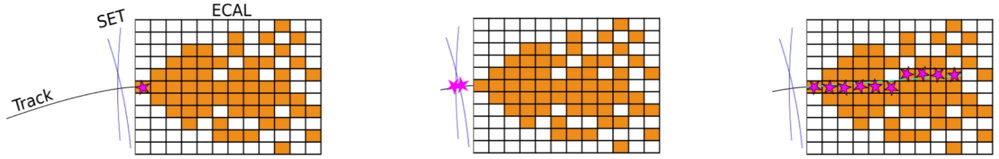


30 ps resolution per particle





Time-of-flight reconstruction in ILD

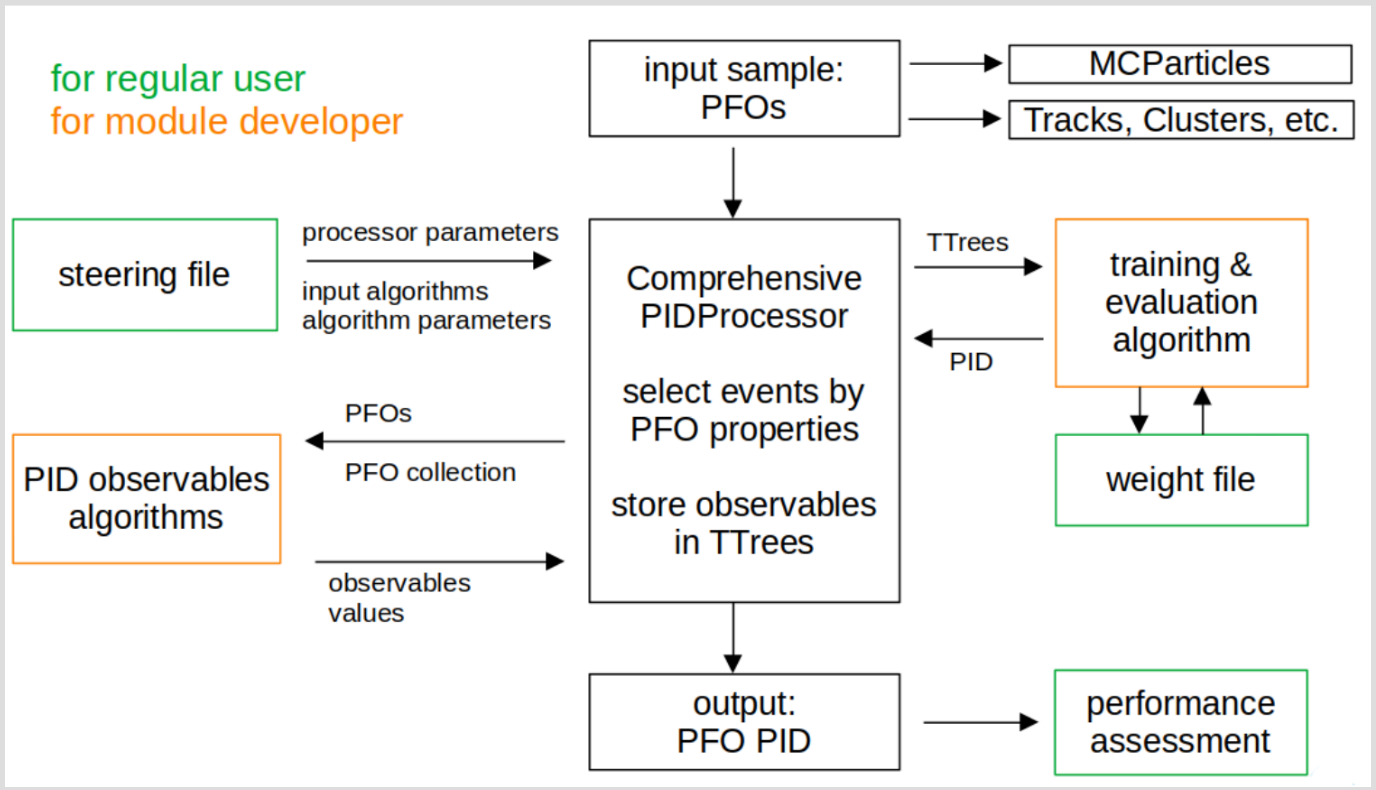


Placement:	First ECAL layer dedicated for timing	Two external tracker layers	10 ECAL layers (conservative timing)
Assumed hit time resolution:	~ 30 ps	~ 50 ps	~ 100 ps
Expected TOF resolution:	~ 30 ps	$\sim \frac{50}{\sqrt{2}}$ ps	$\sim \frac{100}{\sqrt{10}}$ ps

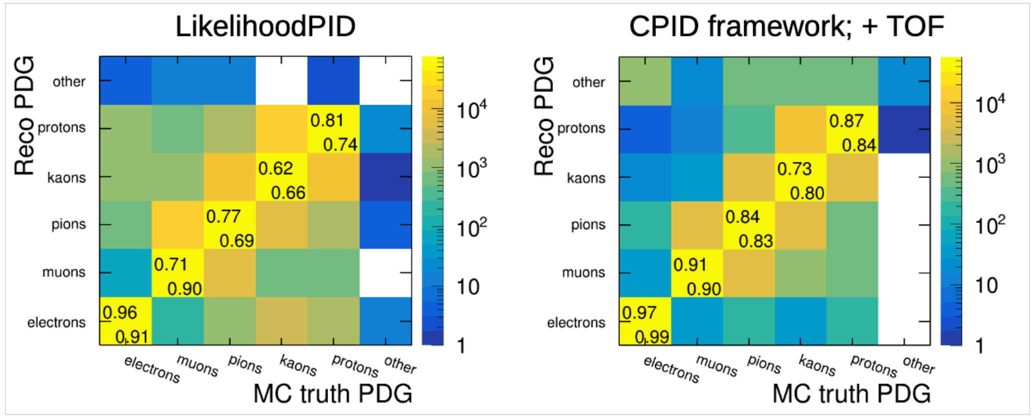
General PID

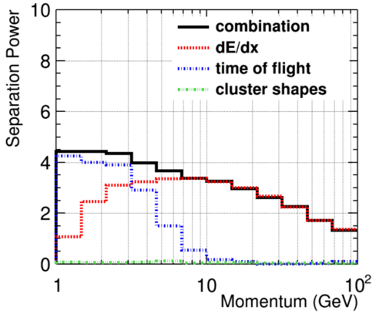
comprehensive particle identification CPID

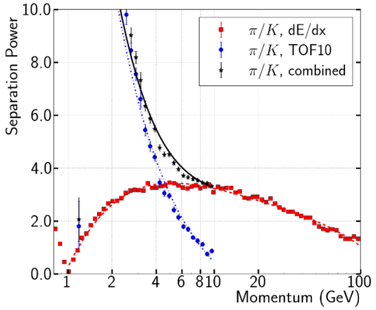
U.Einhaus



- developed a new PID toolkit - release in MarlinReco
- can use to train/evaluate/test novel deep learning algorithm for PID
- allows much easier combination of different measurements (dE/dx, TOF, cluster-shape) and shows better results





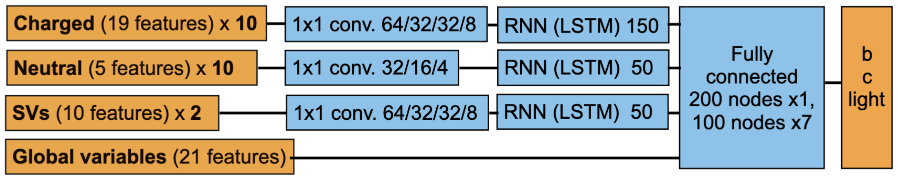


Flavour tagging

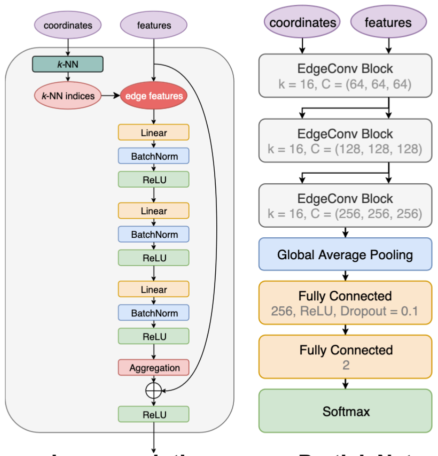
with deep learning methods

- implemented DeepJet and ParticleNet flavour tagging for ILD
    - achieve mostly better results than LCFIPlus
- implemented in Marlin processor

DeepJet: architecture



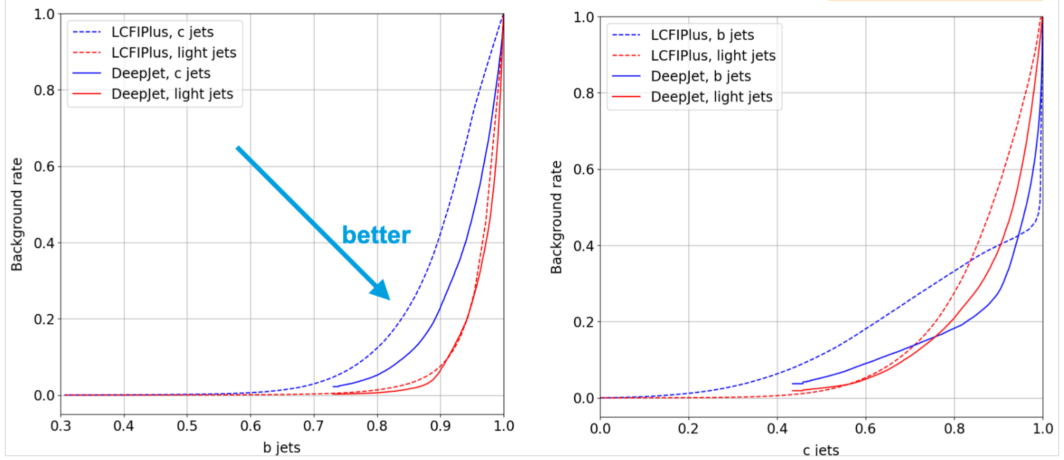
M.Meyer



edge convolution ParticleNet

DeepJet: ROC curves - comparison to LCFIPlus

validation data



Workflow training (PyTorch) & inference (iLCSoft/Marlin)

- run PV & SV finder, jet clustering and vertex refinement of LCFIPlus
- run Marlin processor that calculates and stores features needed for the flavor taggers

} iLCSoft/Marlin

- store variables in **root files** with four trees (charged, neutral, jets, sv)

} iLCSoft/Marlin

Training (python scripts & PyTorch):

- convert trees in root files to **pandas dataframes**, do some **checks** and **cleaning,** store dataframes in **hdf5-files**
- do **further pre-processing** and **training** in **PyTorch**
- use **torch library to convert trained model** into model that can be used in C++

Inference (iLCSoftMarlin)

- store variables via **PIDHandler** (**not optimal** in terms of memory, might be changed)
- run **Marlin processor** for **tagging with ParticleNet Model**
    - **read feature values** from PIDHandler
    - **store them** in the **vectors needed by the ParticleNet Model** (coordinates of const., features of the const., coordinates of SV, features of SV)
    - **convert vectors to torch tensors** and do the **pre-processing**
    - do the **inference** with the **converted model**
    - store output again using PIDHandler
- run Marlin processor to store outputs in trees and histogram that can be used to calculate ROCs etc.

Flavour tagging

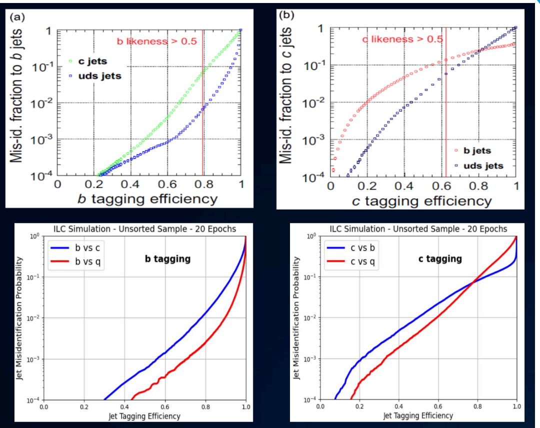
T.Suehara et al

with deep learning methods

- implemented ParticleTransformer flavour tagging for ILD
  - achieve dramatically better results than LCFIPlus
- observe strange improvement w/ more training data at FCC -> to be studied
- framework inference not yet implemented
  - could do this in **Gaudi/EDM4hep**

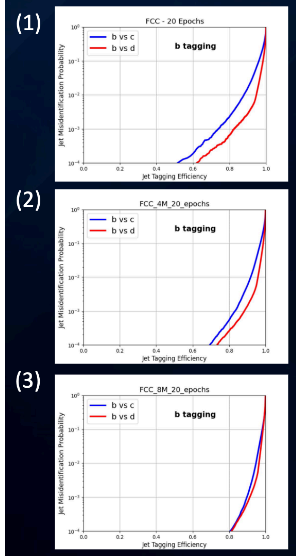
Application of ParT to ILD data
(ILD qq 91 GeV, 0.8M jets for training)

- Jet tagging performance is greatly improved by ParT immediately.
- The performance is improved by 4.05 – 9.80 times compared to LCFIPlus with the same set of data.
- 20 epochs are taken, 200 epochs do not help improving performance but give overtraining



Method	b-tag 80% eff. c-bkg acceptance	b-tag 80% eff. uds-bkg acceptance	c-tag 50% eff. c-bkg acceptance	c-tag 50% eff. uds-bkg acceptance
LCFIPlus	10%	1%	10%	2%
ParT	1.29%	0.25%	1.02%	0.43%

Sample size affects performance (FCCee sample)

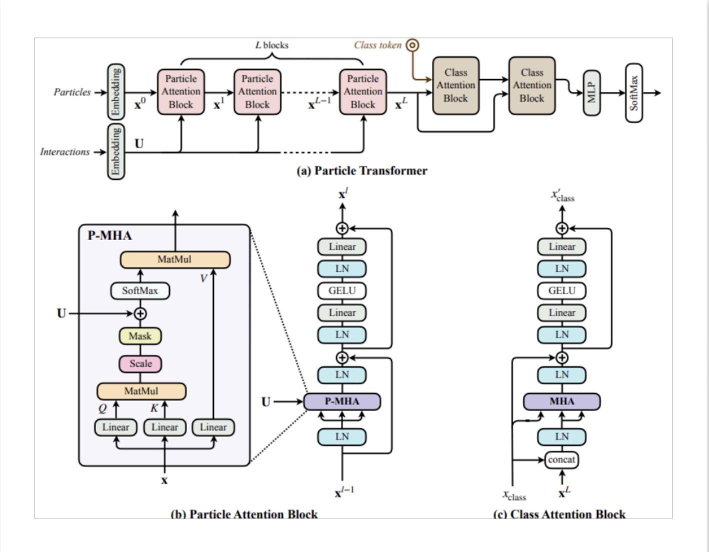


Plot Index	Particle ID	Impact Parameters	Jet Distance	Track Errors	Training Sample size	c-bkg acceptance @ b-tag 80% eff.	b-bkg acceptance @ c-tag 50% eff.
(1)	◍	◍	◍	◍	800k	0.23%	0.35%
(2)	◍	◍	◍	◍	4M	0.054%	0.20%
(3)	◍	◍	◍	◍	8M	Unreasonably good, TBC	

- Training performance significantly improved with bigger data sample size
- Training sample size change of FCC data:

  800k -> 4M : 4 times better performance (b-tagging)

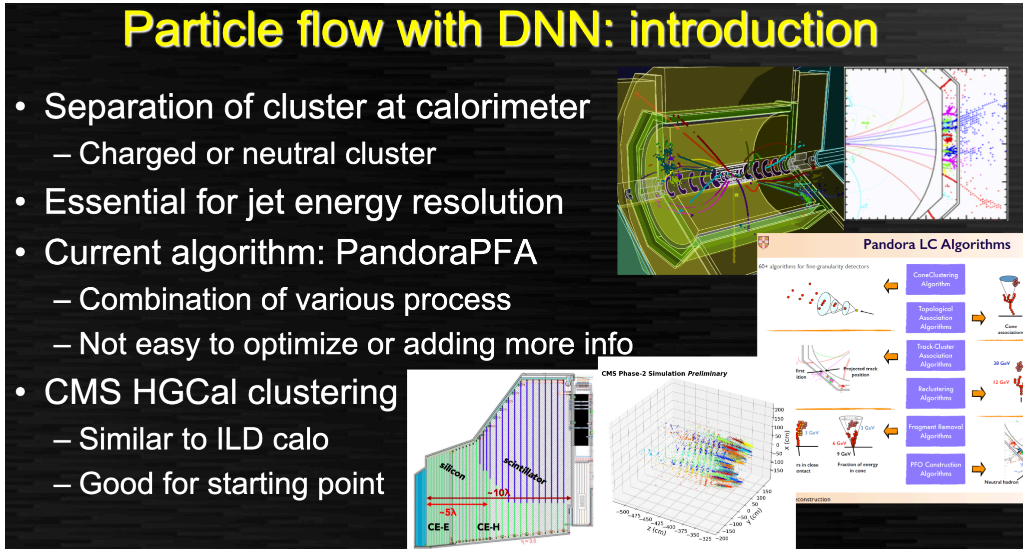
  4M -> 8M: 5 times better performance (b-tagging)
- This non-linearity of increase in performance should be further investigated.
- Bigger data size of ILD should be obtained for better performance, as well as comparison with FCC data for further investigation on its behaviour.

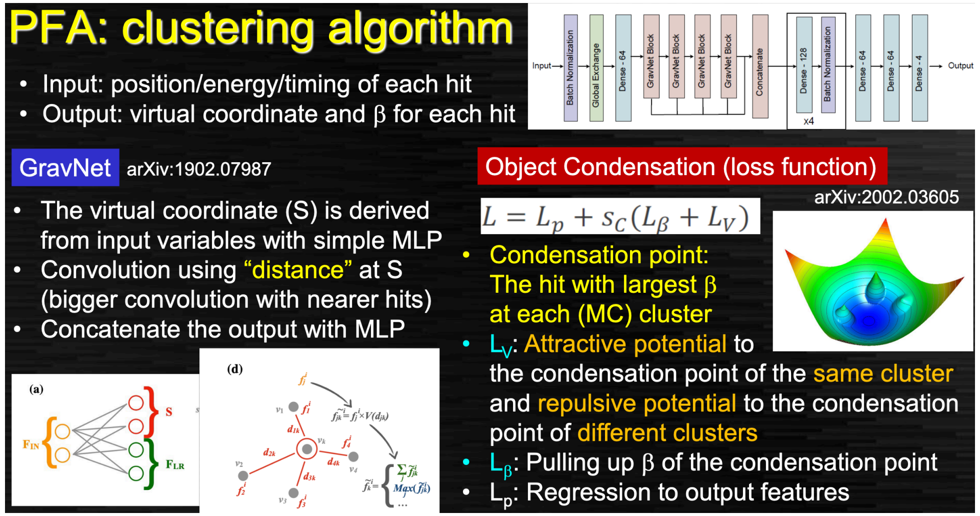


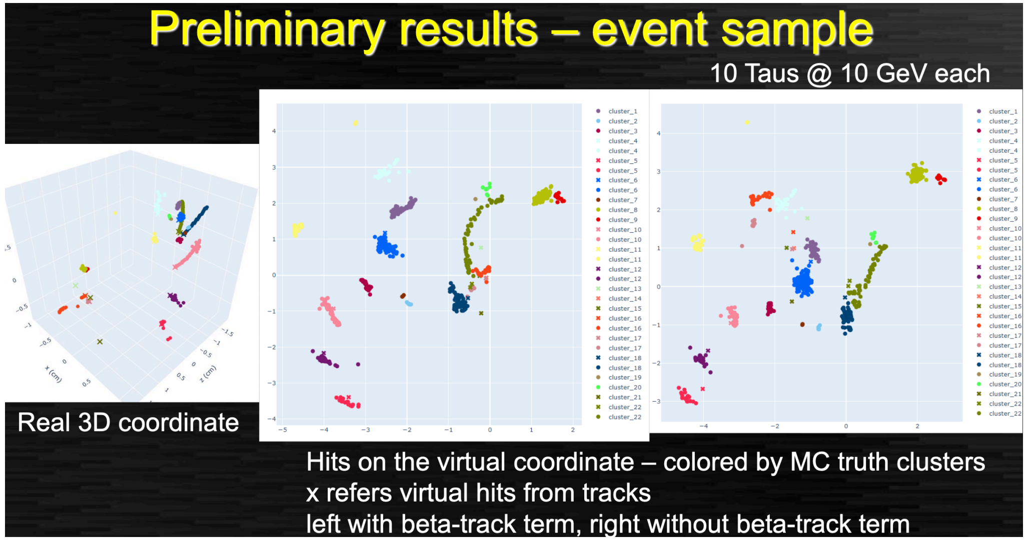
Deep Learning for PFA

- started to develop deep neural networks for particle flow - partly based on CMS HGCal
    - using GravNet and Object Condensation
- some early, promising results
- work in progress …

T.Suehara



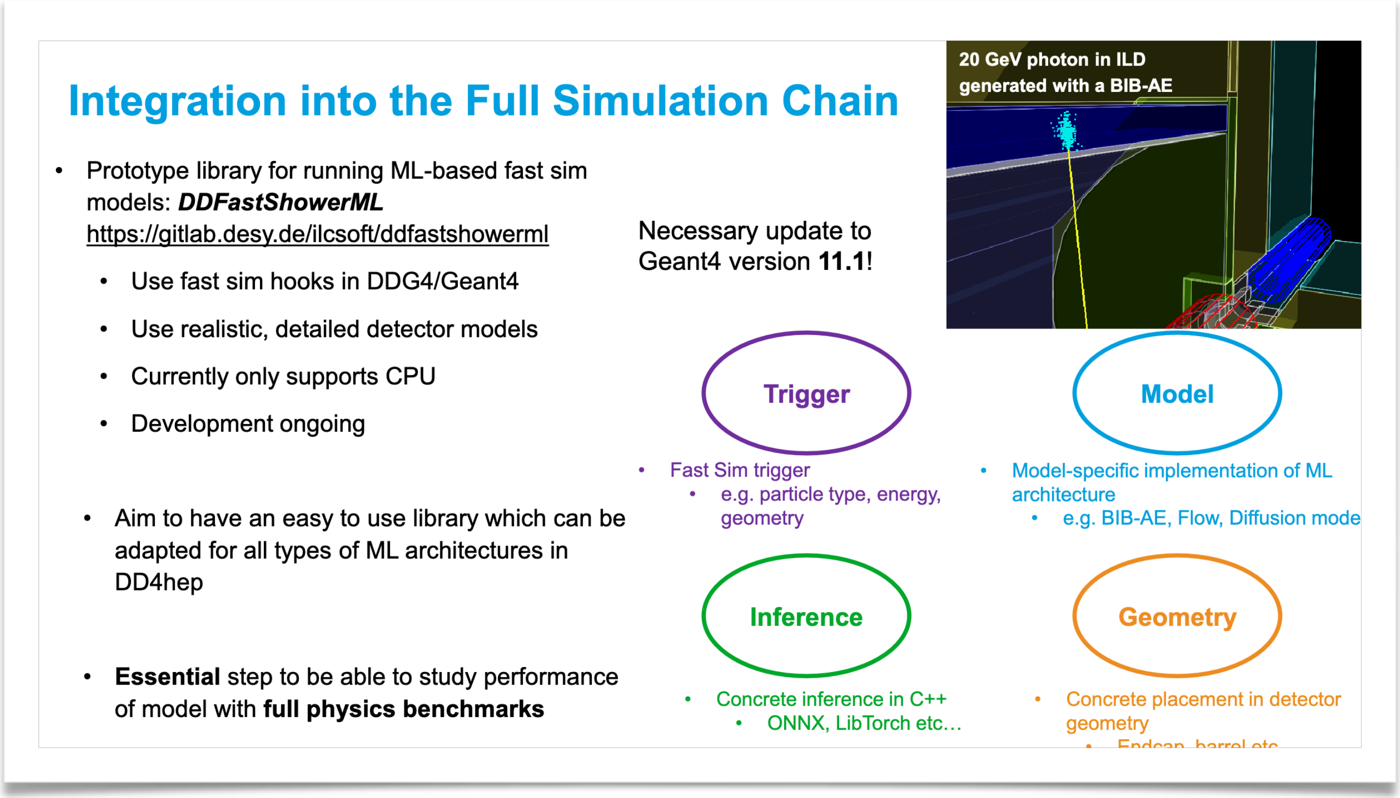




AI in Key4hep

integrate ML inference smoothly in code base

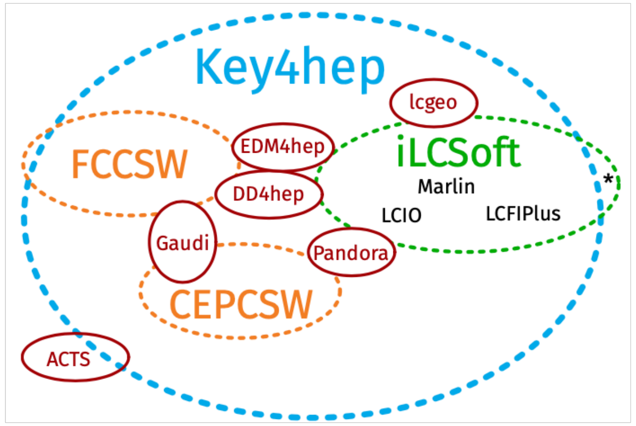
- we see more and more developments using AI/ML for (HL) reconstruction, e.g.
  - *LCFIPlus* (TMVA)
  - *CPID* (TMVA et al)
  - *MarlinML* flavour tagging
  - DNN for PFA
  - …
- should try and make an effort to unify and simplify the use of ML inference in Key4hep for reconstruction
  - some ideas developed in DDML for fast ML simulation



P.McKeown: integration of generative ML based fast simulation in dddmim for ILD

Summary and Outlook

- **Key4hep** started as a new future collider community wide effort in 2020 to put together a modern turnkey software stack
    - contributors: CEPC, CLIC, FCC, EIC, ILC, LUXE, Muon Collider …
- **iLCSoft and ILD software integral part of Key4hep from the start**
- **battle proven ILD standard reconstruction** can be run in Key4hep w/ **MarlinWrapper** as before - or w/ EDM4hep output
- many new developments in HLR tools - often ML/AI based
    - need to validate and integrate in ILD standard reconstruction



- Key4hep offers great opportunity to **modernise ILD software** stack AND **collaborate** w/ **other Higgs factories** - when studying ILD for FCC
- should make an attempt for next larger production/study in ILD to move to use more of the new tools in **Key4hep: Gaudi, EDM4hep**
    - have brief discussion on Wednesday
- limiting factor for all software developments: **manpower** …

pointers to documentation

entry points to Key4hep

- Key4hep GitHub Project
  - https://github.com/key4hep
- Key4hep main documentation page
  - https://key4hep.github.io/key4hep-doc/
- Doxygen available., e.g. for EDM4hep
  - https://edm4hep.web.cern.ch/
- iLCSoft Github Project
  - https://github.com/ilcsoft