IDT-WG3 software and computing

overview

ILCX2021 ILC Workshop on Potential Experiments

Frank Gaede, Daniel Jeans, Jan Strube

Institute of Particle and Nuclear Studies

ilc-swc-core@ml.post.kek.jp

Software and Computing Subgroup (WG3)

Mandate

1. To contribute to enlarging the ILC community by bringing in new people and groups and facilitating the start-up of their activities.

2. To estimate and plan the computing resources needed for the ILC (space, power, networks, hardware, manpower on site/campus, …) and to establish software and computing as central topics for the pre-lab in support of the EOI/LOI process.

3. To ensure connection to and ILC representation in relevant activities beyond ILC, e.g. key4hep, IRIS-HEP, or in the context of the ECFA Higgs Factory study or Snowmass. The use of common software will facilitate the merger of the different groups after the selection of experiments.

4. To coordinate and request Grid resources (storage and CPU) at different Grid sites for ILC accelerator, detector and physics studies under the ILC VO.

grow community

understand required computing resources

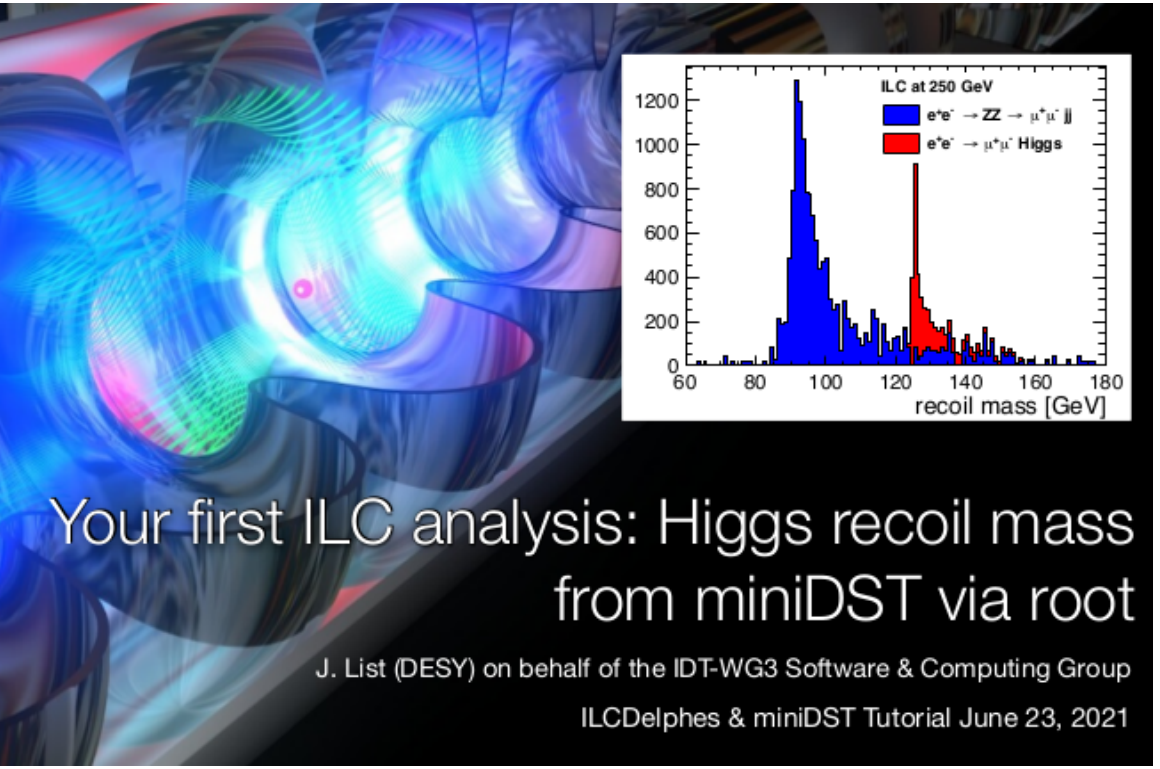
software evolution

software tutorials

newcomers → get started with physics analysis
more experienced users → gain extra powers

October 2021		
13 Oct	tutorial: SGV - fast simulation	Mikael Berggren
August 2021		
18 Aug	tutorial: Jet flavor identification with LCFIPlus	Taikan Suehara
July 2021		
21 Jul - 22 Jul	tutorial: Introduction to iLCSoft	Thomas Madlener
June 2021		
23 Jun - 24 Jun	tutorial: DELPHES + miniDST	Filip Zarnecki, Jenny List

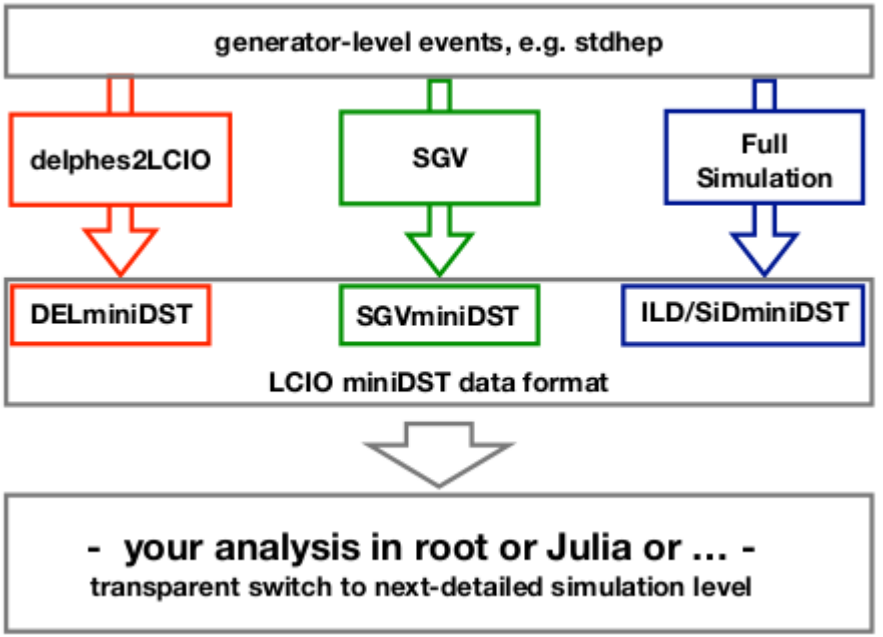
https://agenda.linearcollider.org/category/273/



Your first ILC analysis: Higgs recoil mass from miniDST via root

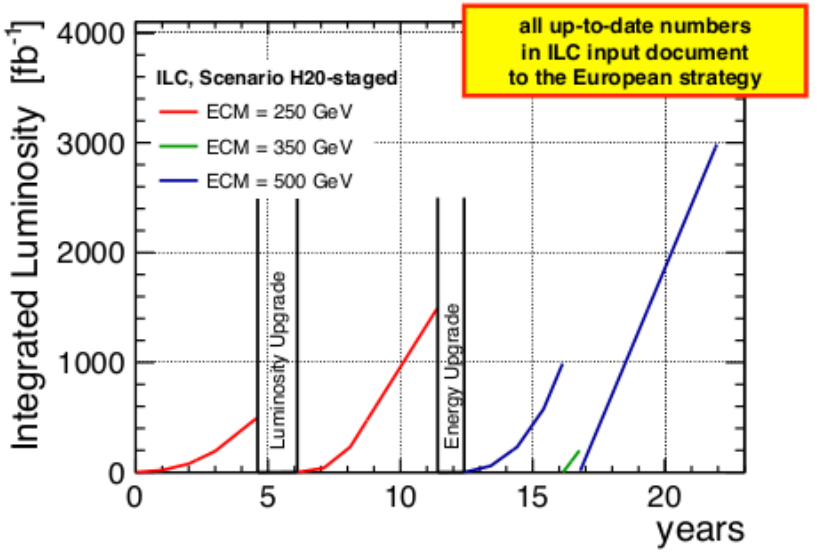
J. List (DESY) on behalf of the IDT-WG3 Software & Computing Group

ILCDelphes & miniDST Tutorial June 23, 2021



- beam polarisation absolute values:
  - Electron beam: $|P(e^-)| \geq 80\%$
  - Positron beam: $|P(e^+)| = 30\%$, at 500 GeV upgradable to 60% at 1 TeV assume 20%
- **Notation: (P(e⁻) , P(e⁺))**
- sharing of luminosity between polarisation signs:

√s	∫ℒ dt	-+	+-	++	- -
250 GeV	2 ab⁻¹	0.9 ab⁻¹	0.9 ab⁻¹	0.1 ab⁻¹	0.1 ab⁻¹
350 GeV	200 fb⁻¹	135 fb⁻¹	45 fb⁻¹	10 fb⁻¹	10 fb⁻¹
500 GeV	4 ab⁻¹	1.6 ab⁻¹	1.6 ab⁻¹	0.4 ab⁻¹	0.4 ab⁻¹
1 TeV	8 ab⁻¹	3.2 ab⁻¹	3.2 ab⁻¹	0.8 ab⁻¹	0.8 ab⁻¹
91 GeV	100 fb⁻¹	40 fb⁻¹	40 fb⁻¹	10 fb⁻¹	10 fb⁻¹
161 GeV	500 fb⁻¹	340 fb⁻¹	110 fb⁻¹	25 fb⁻¹	25 fb⁻¹



The ILC DELPHES Card Tutorial

Aleksander Filip Żarnecki

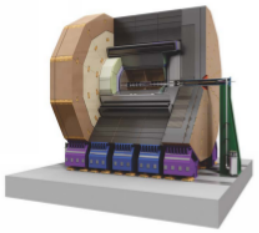
Experiments

Detector Requirements

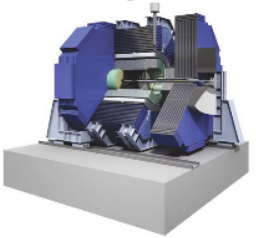
- Track momentum resolution: $\sigma_{1/p} < 5 \cdot 10^{-5}$ GeV^{-1}
- Impact parameter resolution: $\sigma_d < 5\mu m \oplus 10\mu m \frac{1\ \mathrm{GeV}}{p \sin^{3/2}\Theta}$
- Jet energy resolution: $\sigma_E/E = 3-4\%$ (for highest jet energies)
- Hermecity: $\Theta_{min} = 5$ mrad

Two detailed ILC detector concepts: arXiv:1306.6329

ILD

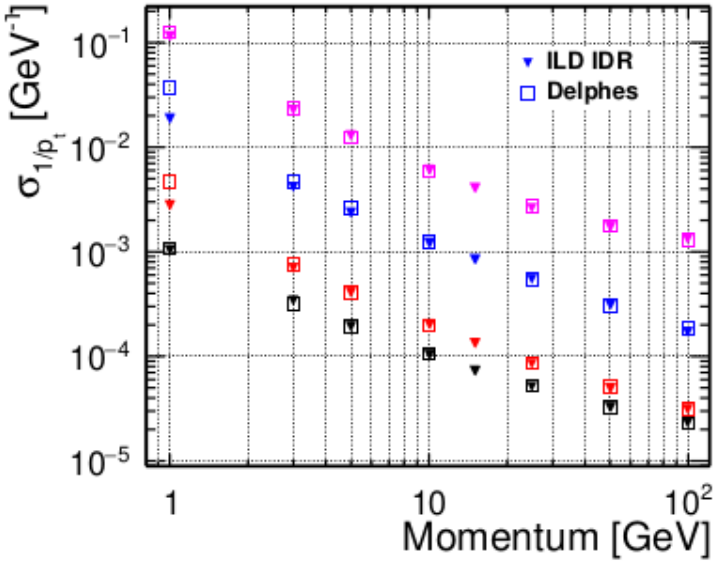


SiD



Different in design, but giving very similar performance

New DELPHES simulation results

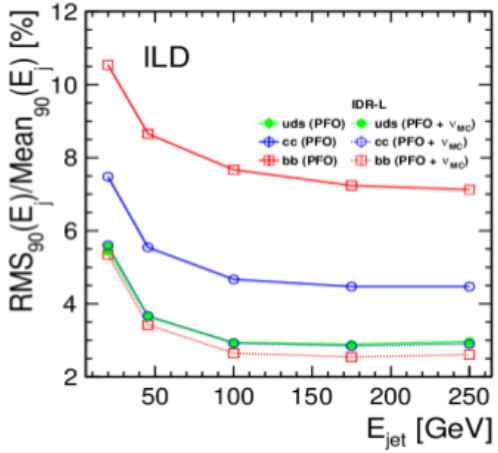


Jet energy resolution

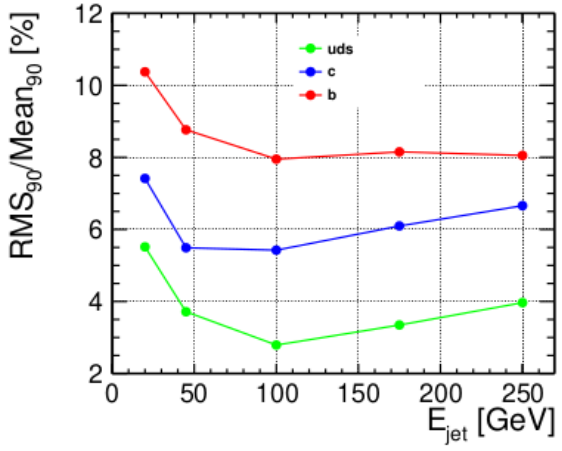
Surprisingly well reproduced with very simplified Particle Flow

Calorimeter granularity and energy response thresholds important!

ILD IDR Fig. 8.3 d ($Z \to q\bar{q}$)



DELPHES simulation

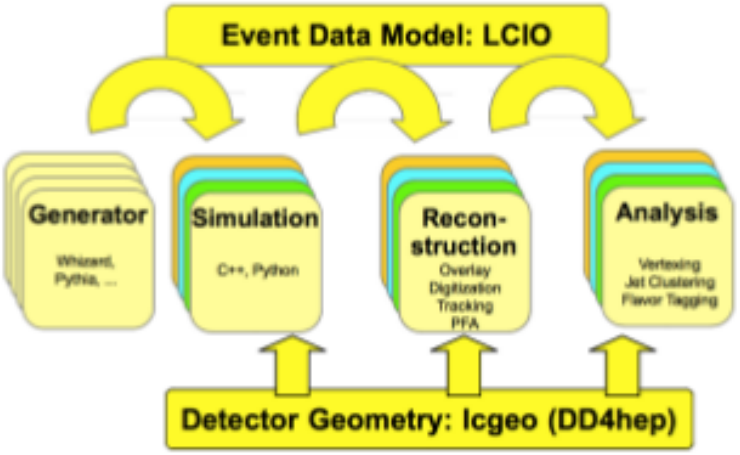


iLCSoft Tutorial

F.Gaede, T.Madlener, DESY

- Introduction to iLCSoft
  - the key components: LCIO, Marlin, DD4hep
  - where to find the code and installations
- First Steps: Running the complete Chain
  - Simulation
  - Reconstruction
  - *Analysis*
- How to write your own Marlin processor

- iLCSoft is the common software framework for Linear Collider detector studies
  - used by CLIC, ILD, SiD, Calice, LCTPC (and friends: FCC, CEPC, HPS, EIC, ...)
- key components in iLCSoft:
- **LCIO**
  - the common *event data model (EDM)*
- **DD4hep**
  - the common *detector geometry description*
- **Marlin**
  - the *application framework*



LCFIPlus Tutorial

T. Suehara (Kyushu U.)

Data/process flow

All in "lcfiplus" namespace

EventStore singleton for data pool

vector<Track *> vector<Vertex *>
vector<Neutral *> vector<Jet *>
vector<MCParticle *> any other types

- Automatic type identification (Allow one name with multiple types)
- Automatic creation/deletion (using ROOT class dictionary)

LCIO

LCIOStorer

- Automatic conversion from LCIO to lcfiplus classes (using hook in EventStore)
- Conversion to LCIO is manually invoked by LcfiplusProcessor

configuration

Algorithm

PrimaryVertex JetVertexRefiner
BuildUpVertex FlavorTag TrainMVA
JetClustering MakeNtuple ReadMVA etc.

- Parameters class used for type-safe configuration

LcfiplusProcessor

- Marlin processor
- Process Marlin parameters to be passed to Algorithm
- LCIO I/O configuration

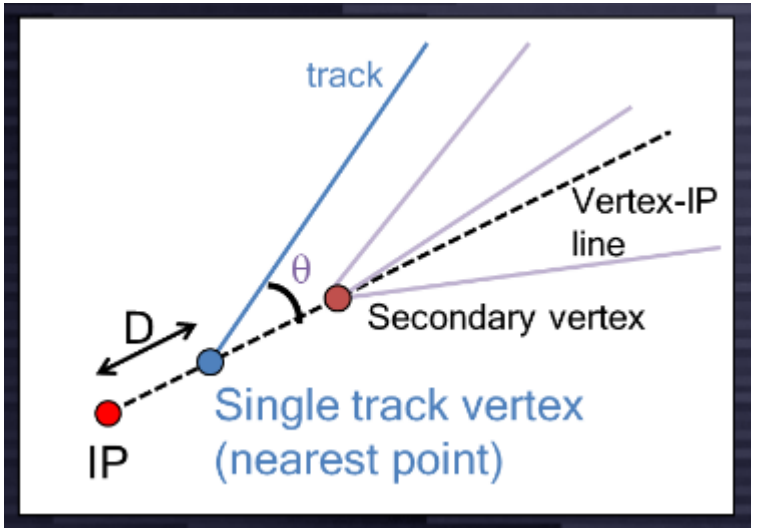
Internal algorithms

Independent

Marlin

Hands on

- Vertex finder
- Jet clustering
- Flavor tagging
  - Track ntuple (skip today)
  - Make ntuple
  - Training
  - Flavor tagging

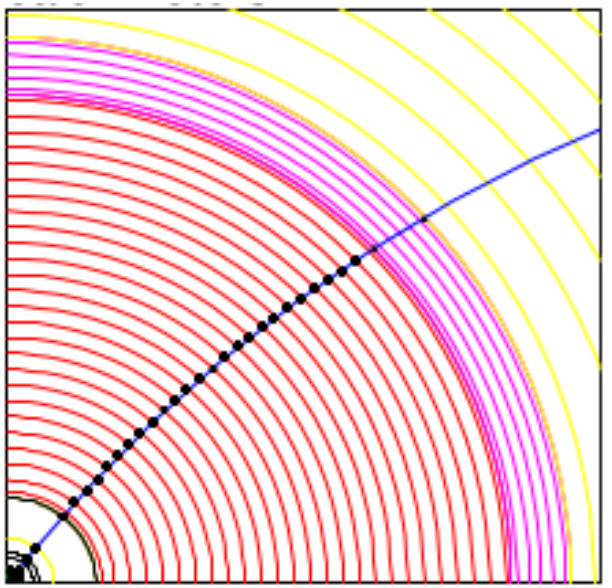


SGV fast detector simulation program : Tutorial

Mikael Berggren[1]

[1]DESY, Hamburg

IDT-WG3 SGV tutorial, Online, 13 October, 2021



Outline

Hands-on

If you have checked the system requirements explained on Indico, you can now get your hands on SGV. Now do this:

- Download `tutorial-step-by-step` from Indico.
- Start viewing the file in a separate window - that can be with `more, less, cat` - whatever. Or in the editor.
- ∼Everything in the file that is flush-left can be cut'n'pasted to the command line.
- I will share my desktop and go through the steps with you ...
  - My desktop is a brand-new Ububtu 20.04, running in VirtualBox.
  - I only did the explained preliminaries - including LCIO and root setup.
  - Apart from that, it's an out-of-the-box installation.

20 ~ 40 participants per tutorial
mix of newcomers and more experienced people

tutorials very well prepared,
with successful hands-on examples

thanks to all lecturers!

zoom recordings (almost all) linked from the indico page
→ take a look if you missed a tutorial

We have ideas for a few future tutorials
→ open to requests from the user community

ilc-swc-core@ml.post.kek.jp

(Jan Strube)

circuit of computing groups at US labs & institutes

BNL, IRIS-HEP, LBNL, FNAL, HEP-CCE-IOS, ...

→ Inform about current situation

→ prepare for possible future contribution,
when resources available

If you / your lab is interested to initiate a similar discussion,
let us know

grow community

understand required computing resources

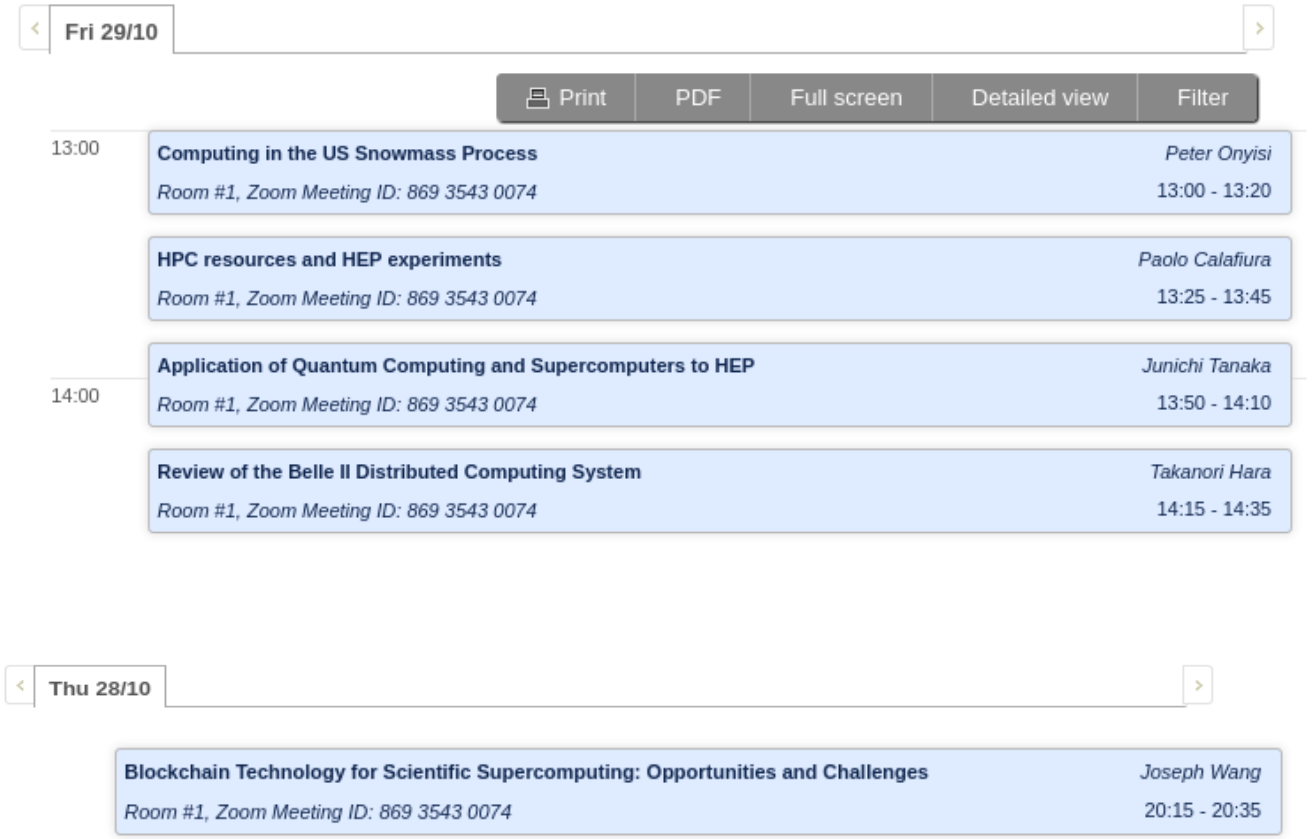
software evolution

revisit past estimate of required computing resources by SiD, ILD

→ new input from ILD's recent large full-simulation MC production

experience of distributed computing (e.g. at Belle2)

how will computing look 15 years from now?

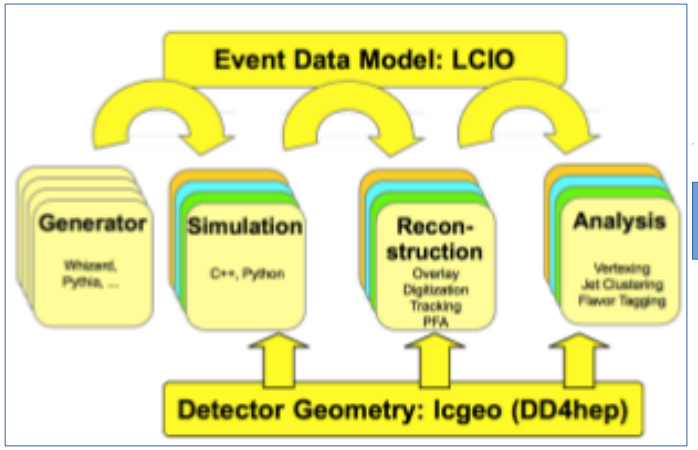


grow community

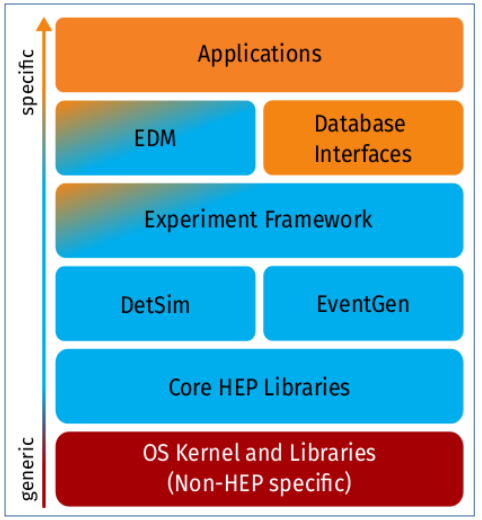
understand required computing resources

software evolution

iLCSoft



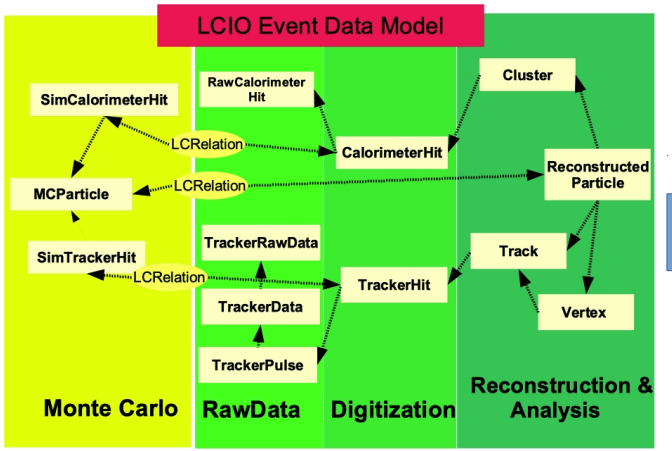
ILC, CLICdp, …

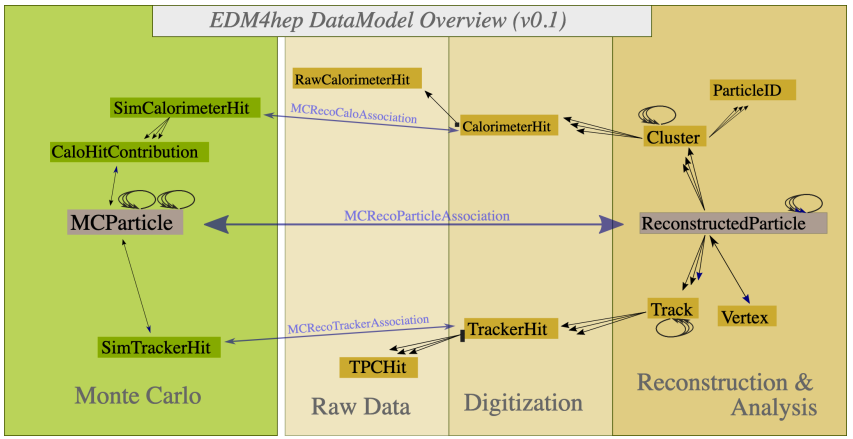


+ FCC, CEPC, ...

Key4HEP "Turnkey Software Stack"

for example, the event data model strongly inspired by LCIO



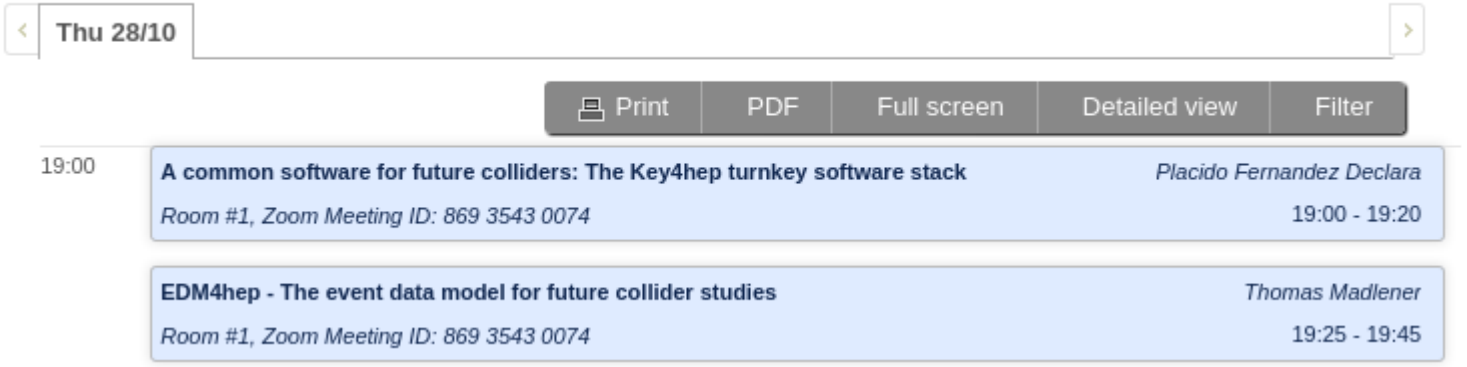


→ emphasise **common** tools, **common** development
contributions from both current and future collider experiments

→ several key iLCSoft people involved in Key4HEP

→ gradual move towards new framework,
while keeping existing tools running for ongoing analyses

talks on future software directions



summary

grow community

- tutorials for analysers, algorithm developers
- open discussion with potential contributors

understand required computing resources

- start to review previous estimates
- learn about computing trends

software evolution

- common development with nearby projects