Software for Future Colliders

Challenges and Opportunities

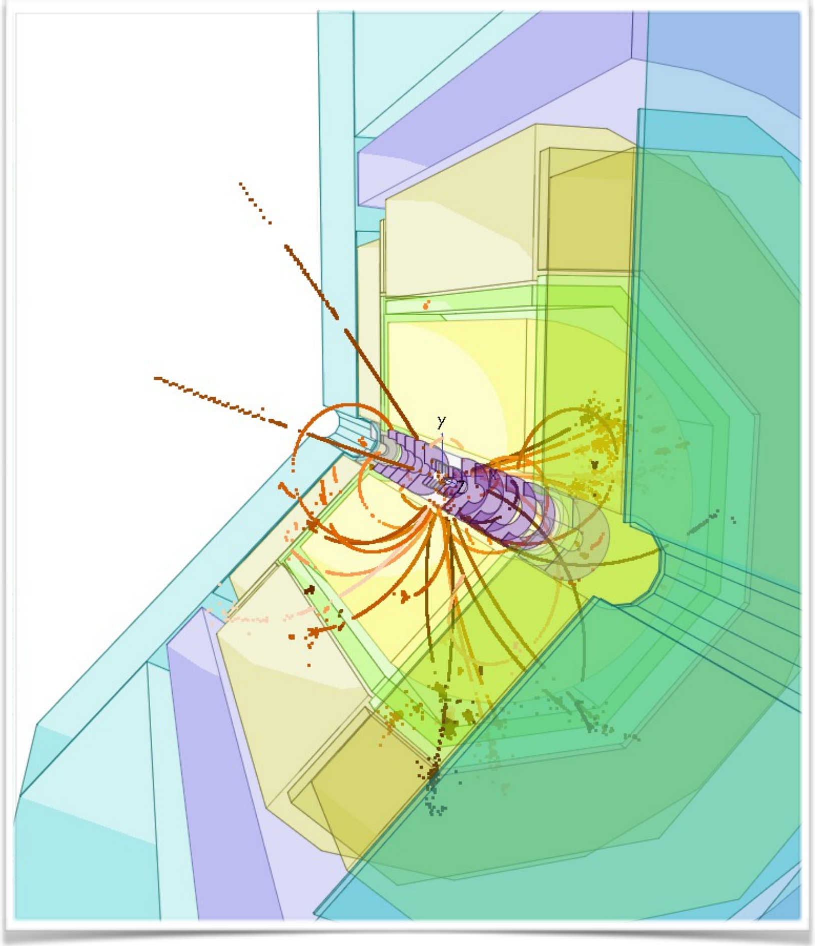
LCWS 2021

17.03.21

Frank Gaede, DESY

Outline

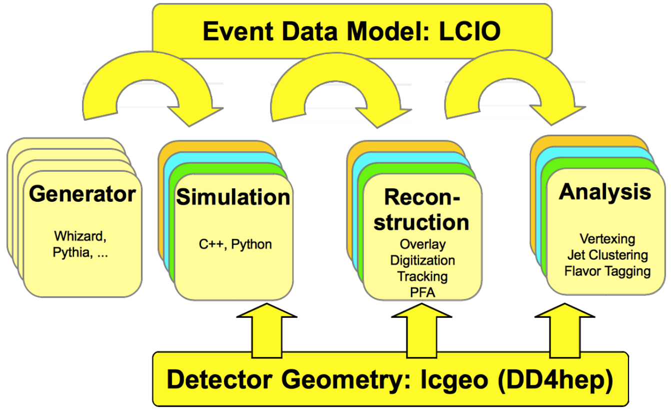
- core tools and framework
- simulation and reconstruction tools
- improving (high level) reconstruction
- machine learning and AI
- modernising the framework
  - MarlinMT, Key4HEP, EDM4hep
- Grid Computing
- IDT-WG3 Software WG
  - Mandate and first activities
- Conclusion and Outlook



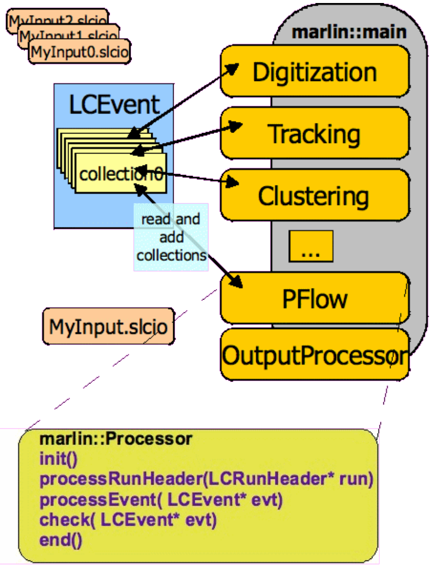
LCIO and Marlin

Setting the foundation for common software development

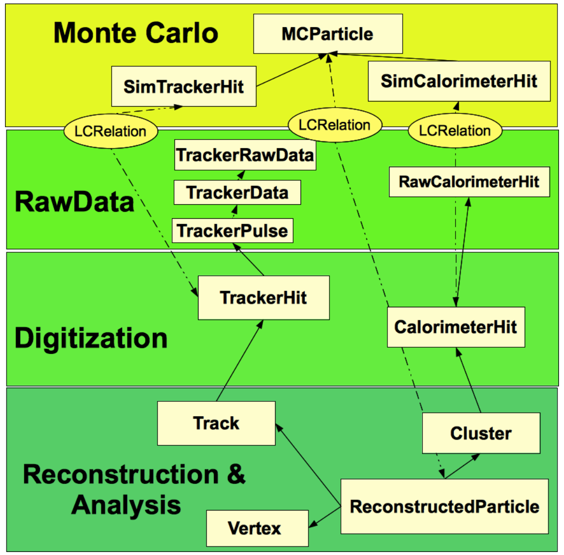
- the linear collider community has started to develop **common software** in 2003
- introducing a common event data model: **LCIO**
  - providing the ***language*** and a file format
- introducing a simple to use framework: **Marlin**
  - allow all LC groups to develop their algorithms in a common software ecosystem



a flexible, generic and easy to use software ecosystem provides the foundation for all future collider developments
- progress in physics analysis
- optimising the detector concepts



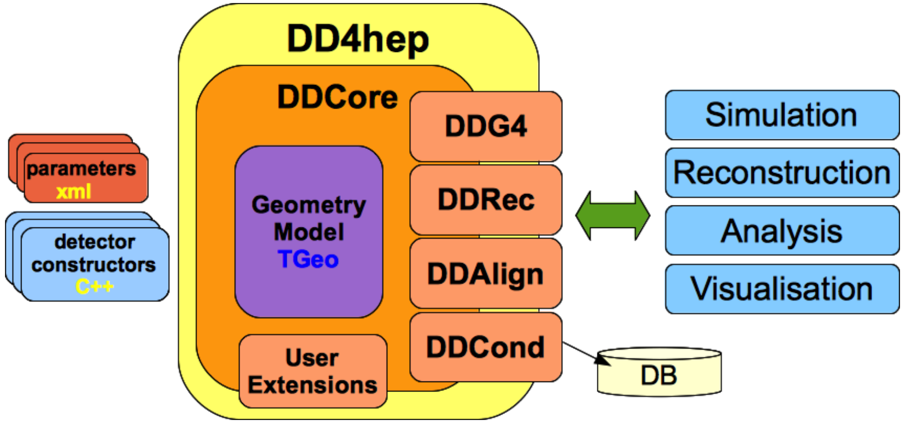
```
marlin::Processor
init()
processRunHeader(LCRunHeader* run)
processEvent( LCEvent* evt)
check( LCEvent* evt)
end()
```

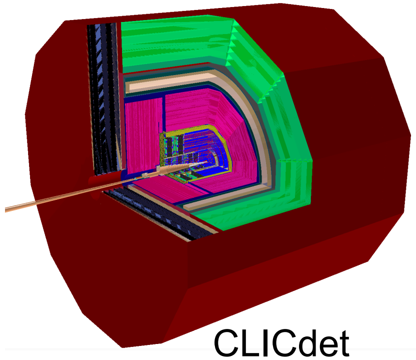


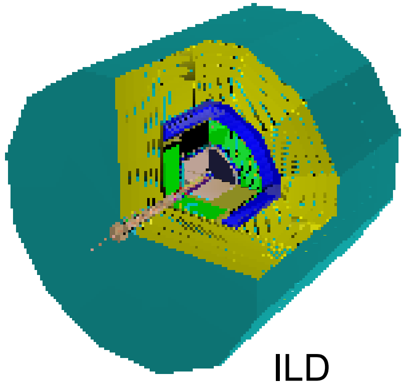
DD4hep geometry toolkit

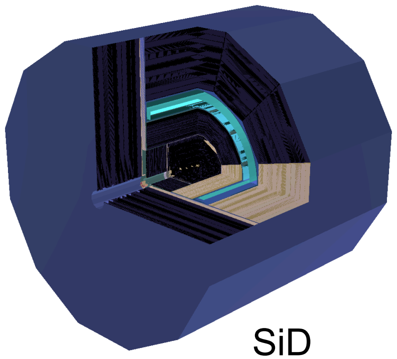
defining the detector geometry and different views on it

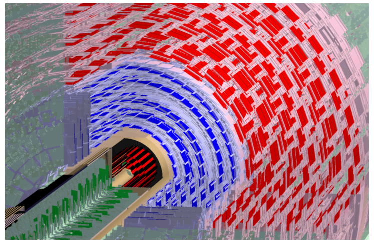
- LC community has developed a generic detector geometry system - based on best practises by ILC, CLIC, LHCb
- supporting the full life cycle of the experiment
- providing components and interfaces for
  - full simulation, reconstruction, conditions, alignment, visualisation and analysis
- adopted also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

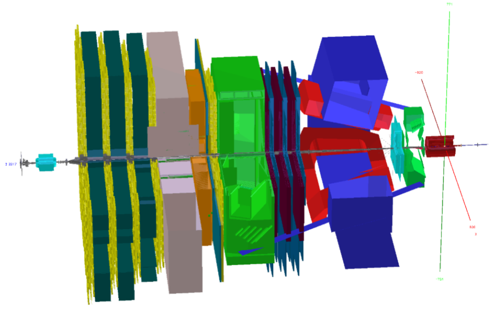


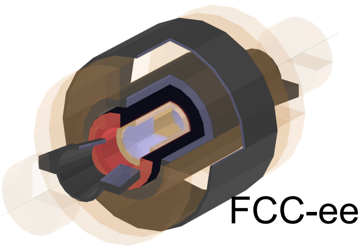

CLICdet

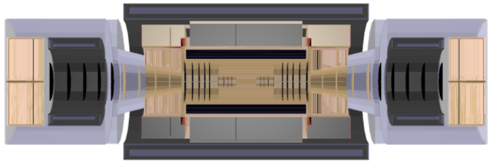

ILD


SiD


CMS

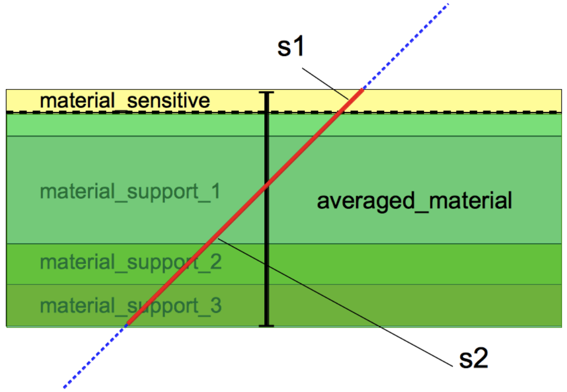

LHCb


FCC-ee

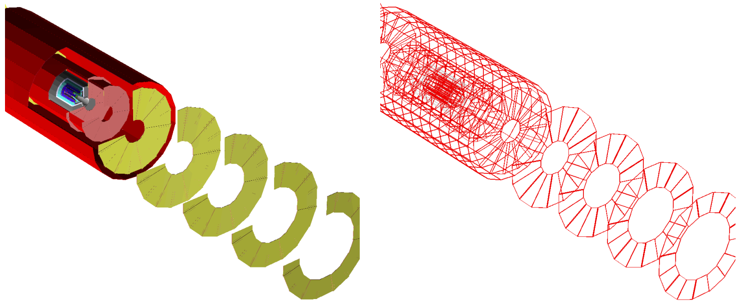

FCC-hh

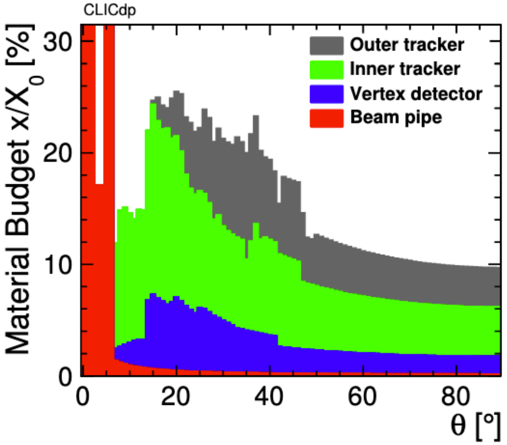
Detector models in DD4hep

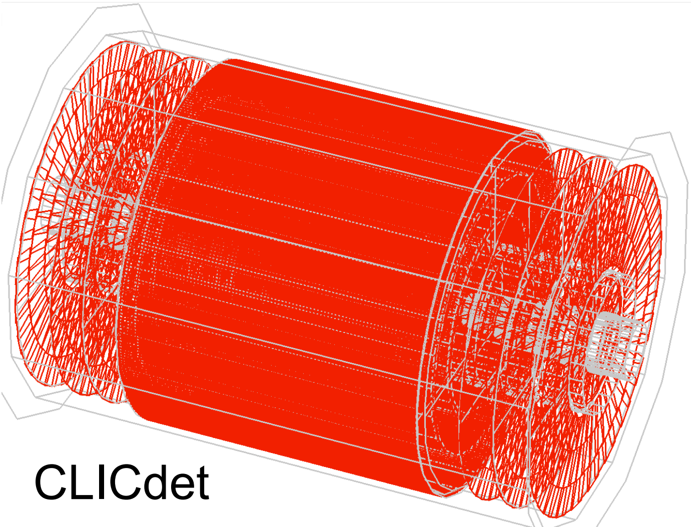
realistic description of active and passive materials



- **all LC detector concepts** have **realistic and detailed** detector models in DD4hep with
  - gaps and imperfections, dead material, …
- crucial for **realistic simulation and reconstruction**
- **surfaces w/ material properties** for energy loss and multiple scattering in track reconstruction
- full simulation w/ Geant4 from **any detector model** that is described in **DD4hep**

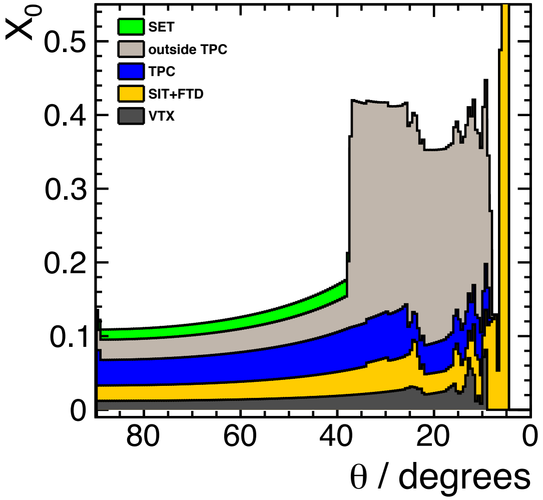


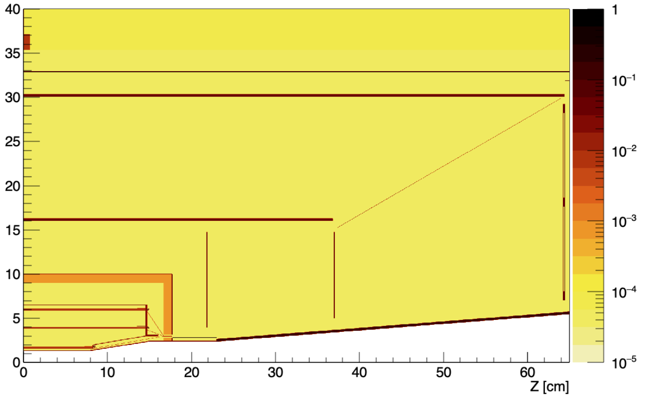




CLICdet

ILD

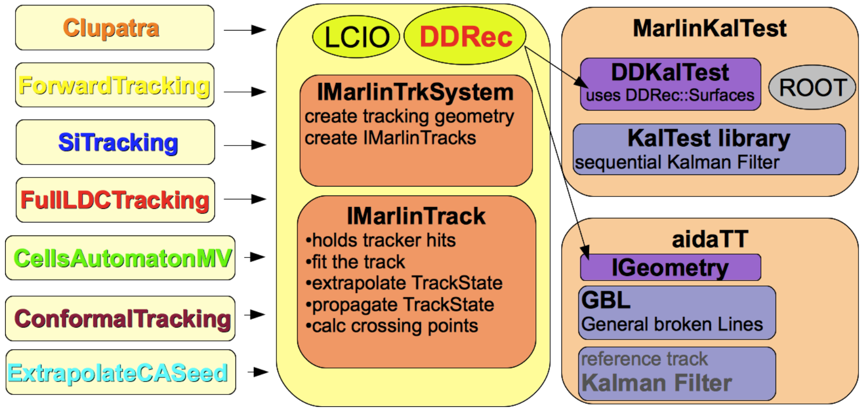




Tracking in iLCSoft

pattern recognition and Kalman-Filter

- generic tracking API MarlinTrk based on DDRec material surfaces
- many pattern recognition algorithms exist, e.g.
- **ConformalTracking**:
  - generic algorithm that works for all Si-Trackers
  - used by CLICdet and SiD (also works for ILD inner)



Clupatra

ForwardTracking

SiTracking

FullLDCTracking

CellsAutomatonMV

ConformalTracking

ExtrapolateCASeed

LCIO

DDRec

IMarlinTrkSystem
create tracking geometry
create IMarlinTracks

IMarlinTrack
- holds tracker hits
- fit the track
- extrapolate TrackState
- propagate TrackState
- calc crossing points

MarlinKalTest

DDKalTest
uses DDRec::Surfaces

ROOT

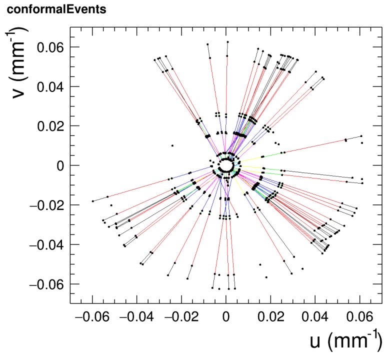
KalTest library
sequential Kalman Filter

aidaTT

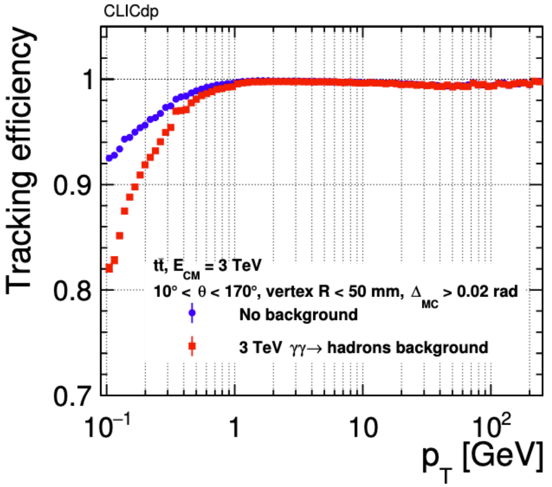
IGeometry

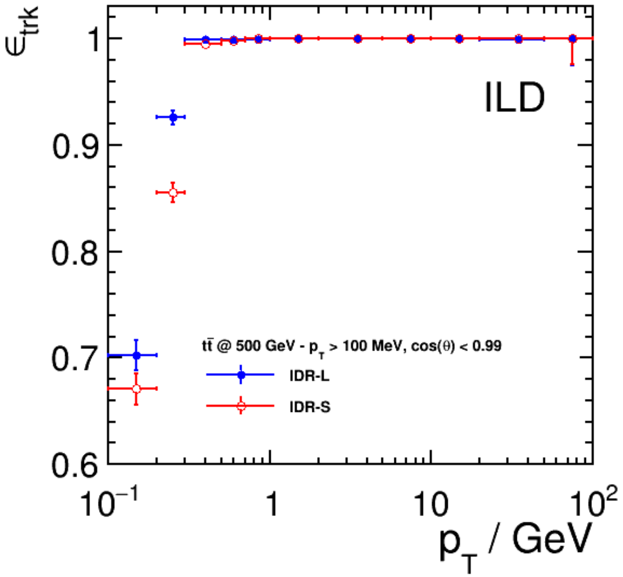
GBL
General broken Lines

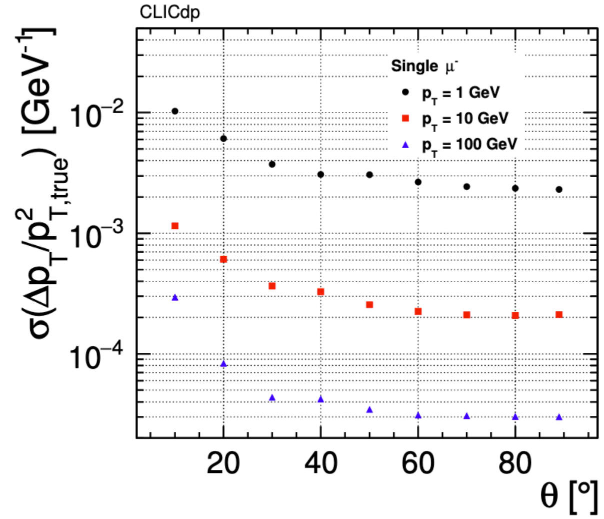
reference track
Kalman Filter

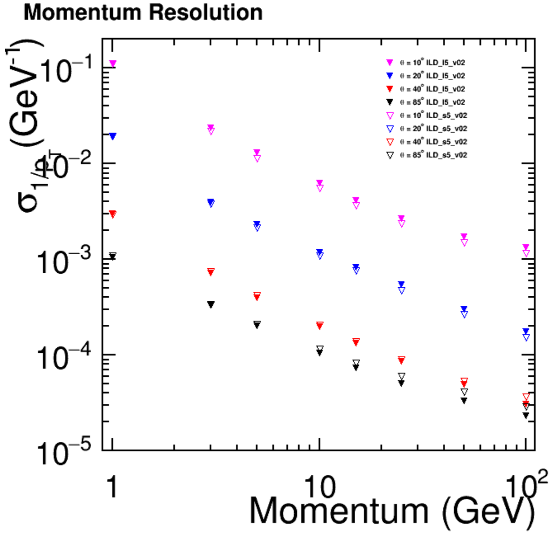


achieve excellent tracking efficiencies and resolution w/ realistic tracking codes





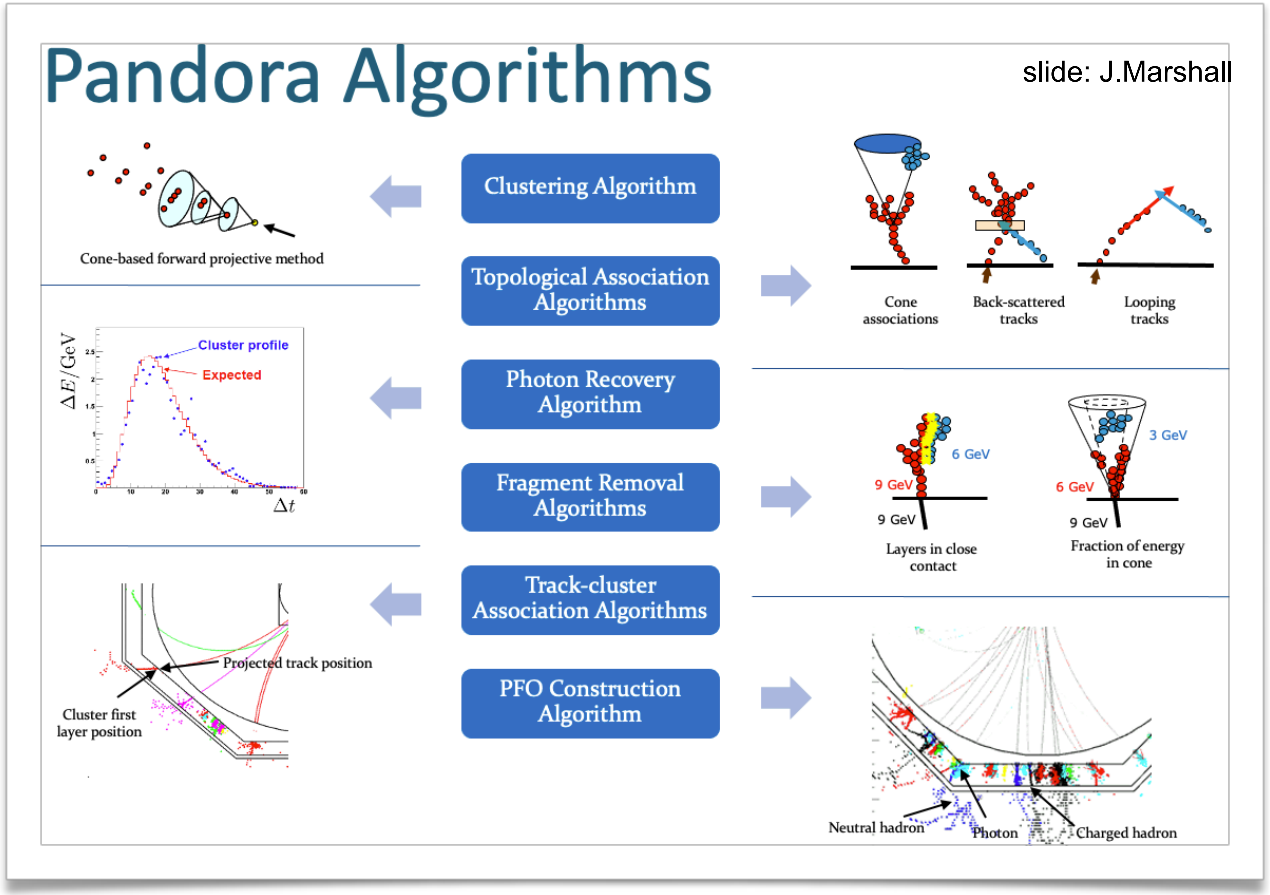




Particle Flow Algorithms

highly granular calorimeter reconstruction

- all current detector concepts for LC are based on highly granular calorimeters
  - optimised for the Particle Flow Algorithm
- **PandoraPFA** is the de **facto standard** used by ILD, SiD and CLICdp
- alternative PFA algorithms exist and provide possibility to cross check
  - Arbor (CEPC), April (SDHCal prototype)



Pandora Algorithms

slide: J.Marshall

Clustering Algorithm

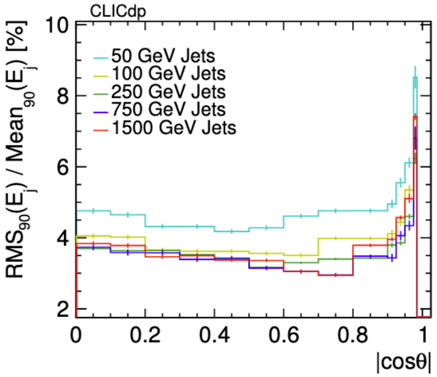
Topological Association Algorithms

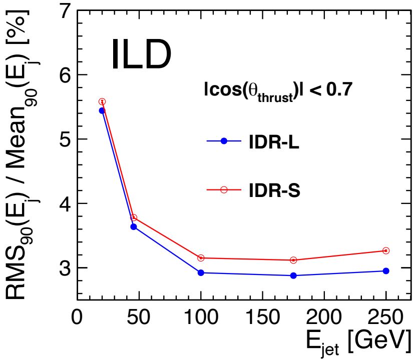
Photon Recovery Algorithm

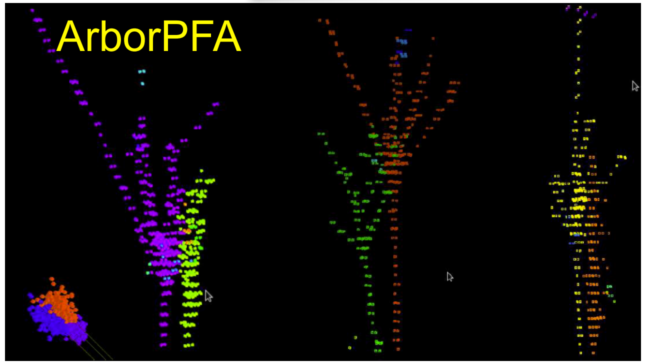
Fragment Removal Algorithms

Track-cluster Association Algorithms

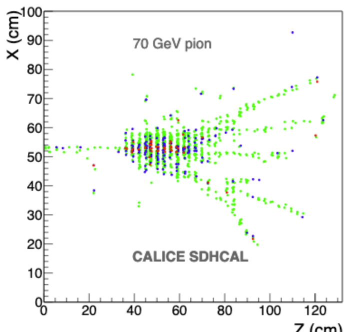
PFO Construction Algorithm

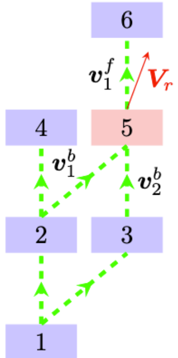






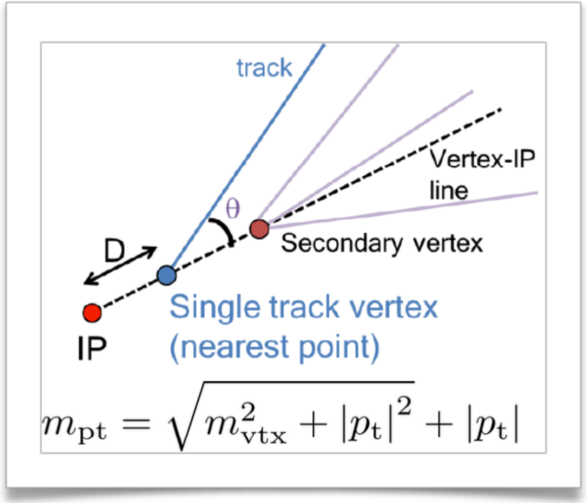
AprilPFA



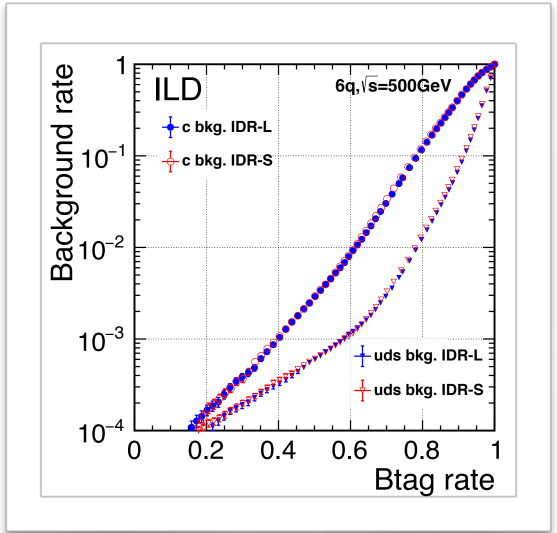


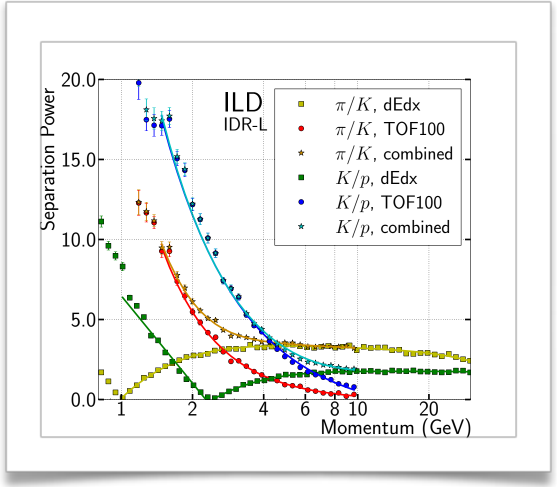
High Level Reconstruction

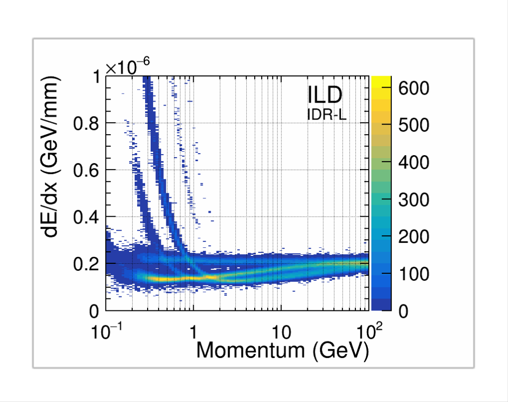
analysing the Particle Flow Objects



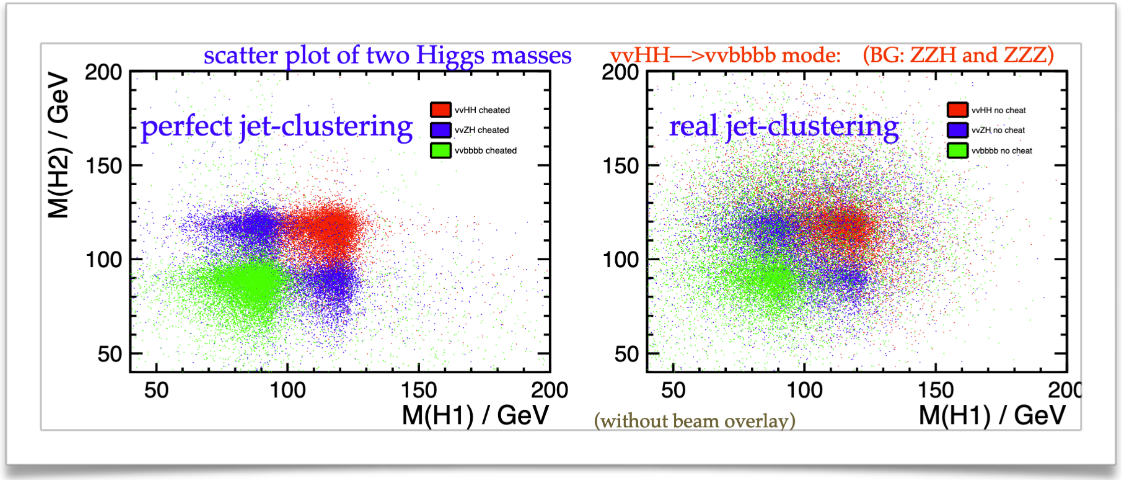
$$m_{\rm pt} = \sqrt{m_{\rm vtx}^2 + |p_{\rm t}|^2} + |p_{\rm t}|$$







- **High-Level reconstruction** algorithms are crucial to achieve the ultimate physics reach of detectors
- vertex finding and flavor tagging: **LCFIPlus**
- PID tools: dE/dx, TOF, shower shapes,…
- Jet clustering: Durham, Valencia, …



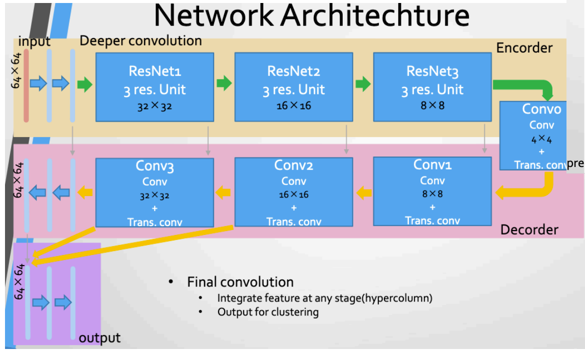
$\delta\lambda_{HHH}$ improves by 40% w/ perfect jet clustering

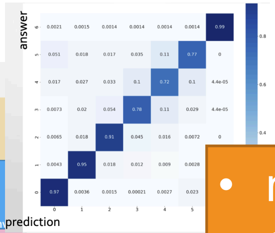
- very active field of development
  - already good set of tools available
- further improvement in HLR tools often directly impacts the final physics performance

Machine Learning for LC physics

applying AI to further improve the LC physics reach

some selected examples - many more talks at LCWS …
see **PD4 session on Wednesday**



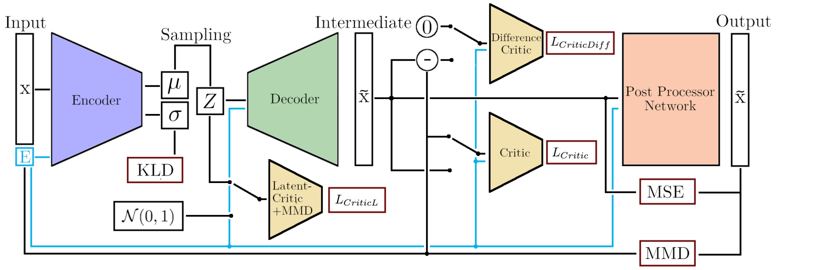


jet clustering with DNNs

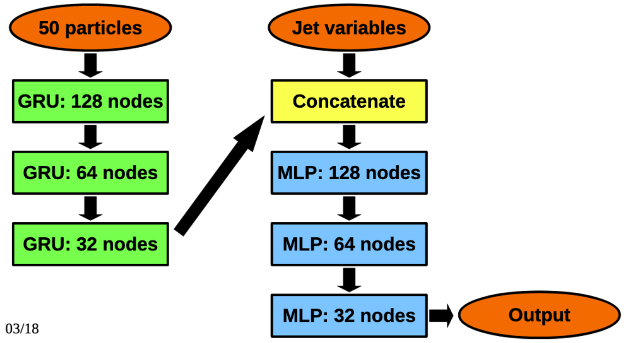
see talk M.Kurata Tuesday

- machine learning and deep learning offer exciting possibilities for improving the physics reach
- however not always immediately better than classical algorithms
- will see many more of this in the coming years …

fast shower simulation with generative networks



see talk E.Eren Tuesday


50 particles

GRU: 128 nodes

GRU: 64 nodes

GRU: 32 nodes

Jet variables

Concatenate

MLP: 128 nodes

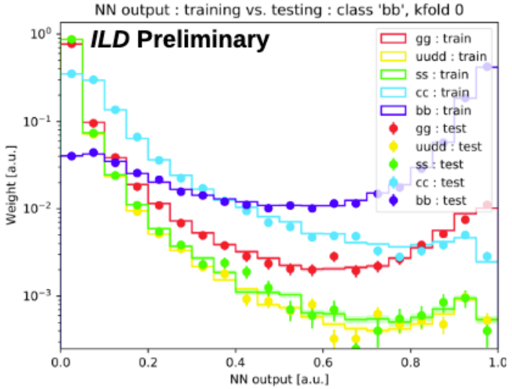
MLP: 64 nodes

MLP: 32 nodes

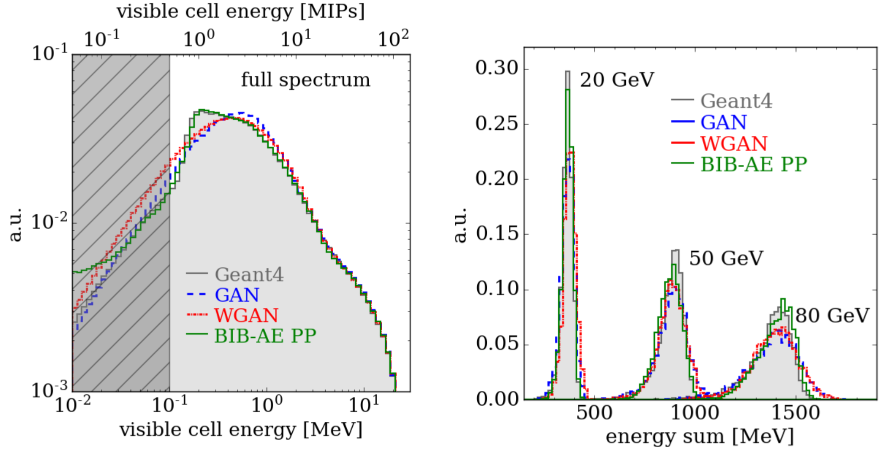
Output

03/18

see talk M.Basso Thursday



aim at extending the flavour tagging to ssbar

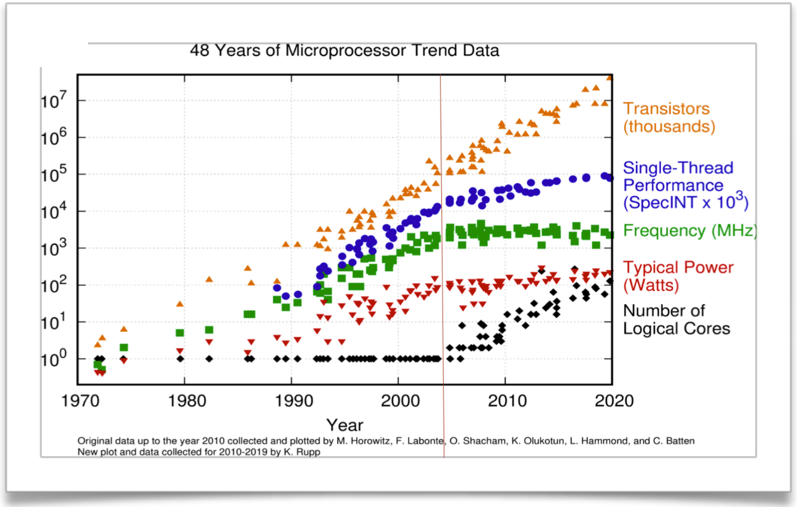


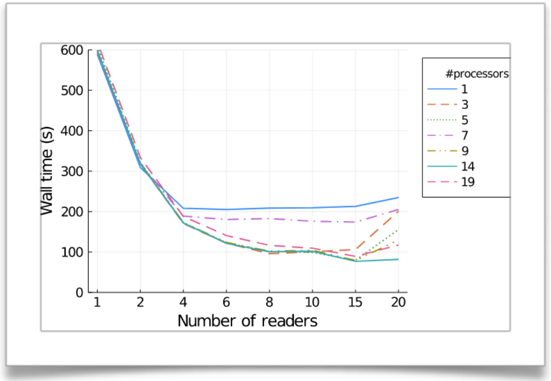
Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed, E. Buhmann ,S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka et al. (May 11, 2020), arXiv 2005.05334

Modernising iLCSoft

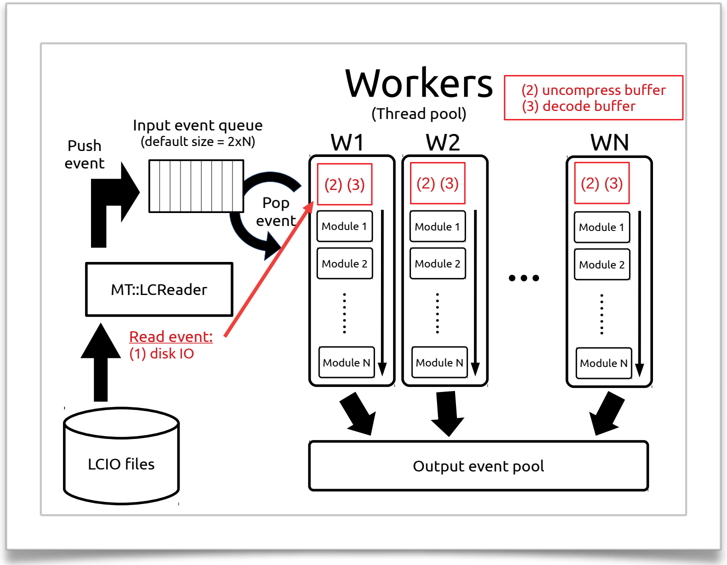
work needed for the next decade(s)

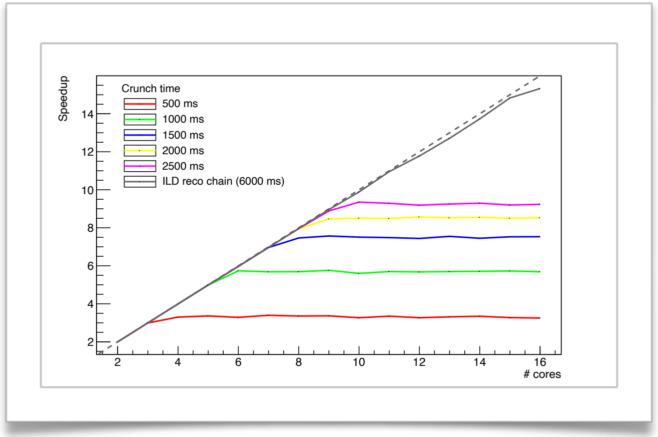
- complete software stack for full simulation and reconstruction exists
- heavily used by ILD, CLICdp, CEPC, SiD and partly FCC-ee
- need to maintain this functionality while making the software fit for the next decades:
  - memory per core decreases
  - **multi-core architectures** more dominant
  - need **parallel workflows**
- develop new tools in parallel, e.g. **MarlinMT**
  - proof-of-concept WIP
- need to address heterogeneous computing:
  - **GPU, FPGA, …**
  - should be able to benefit from developments for **HL-LHC**





Performance of Julia for High Energy Physics Analyses, S.Strube, M. Stanitzki arXive 2003.11952



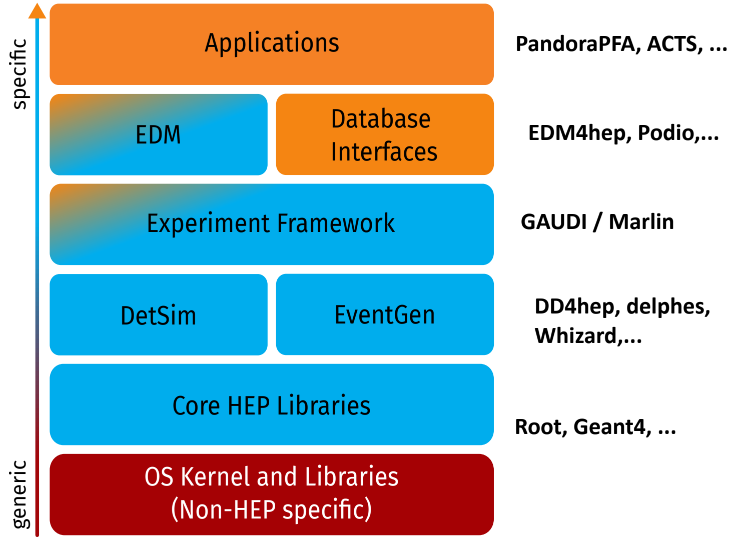


MarlinMT - Parallelising the Marlin Framework, R.Ete, F.Gaede, J.Benda, H.Grasland
https://doi.org/10.1051/epjconf/202024505022

Key4hep

turnkey software stack for all future colliders

- HEP community decided to develop a **common turnkey software stack** – for future collider studies
- Take best tools available and provide necessary interfaces
  - Based on positive experience with **iLCSoft**
- Involved communities: CEPC, CLIC, FCC, ILC,...
  - Supported by **HSF** and **CERN** and *AIDAinnova*
- For ILC a transition to key4hep should be a smooth evolution rather than a revolution
- Need to keep battleproven **iLCSoft** framework and simulation and reconstruction chains in place for physics analyses and detector optimization for the time being



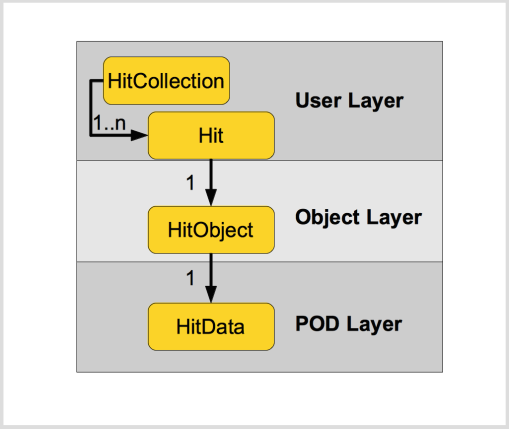
- moving from iLCSoft to key4hep offers great opportunity to **modernise the software stack**
- however need a well defined **migration strategy** helped by the fact that a lot of tools (DD4hep, PandoraPFA,…) are already used in iLCSoft
- **MarlinWrapper** will facilitate this

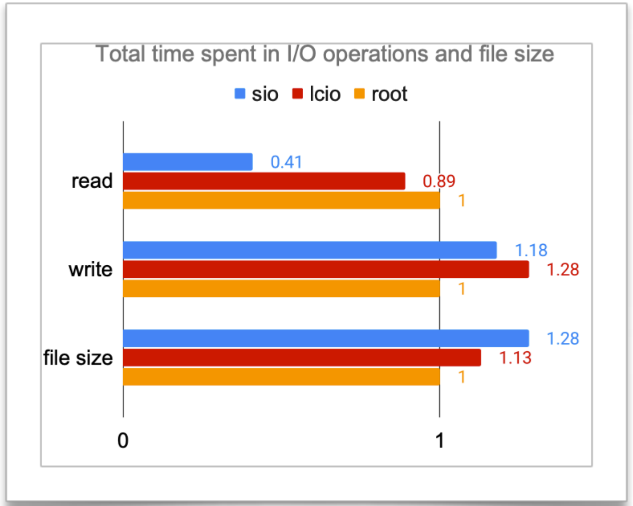
see talks by P.Fernandez and V.Volkl on Monday

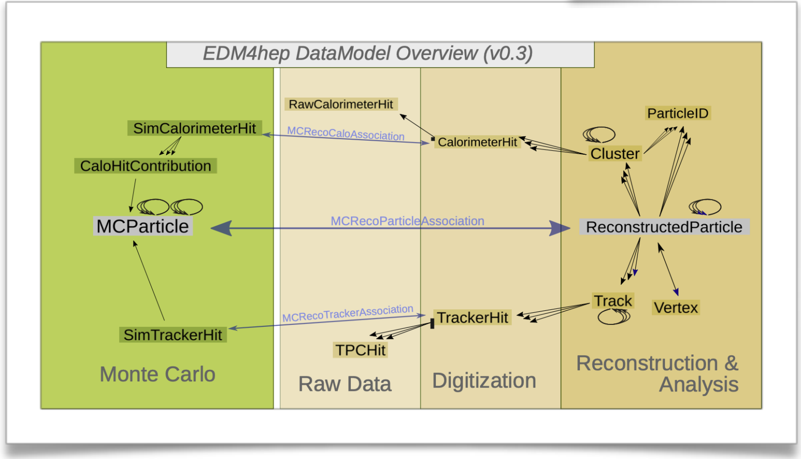
PODIO and EDM4hep

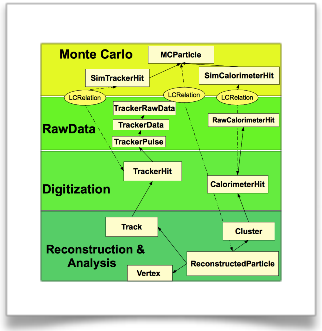
modernising the LCIO event data model

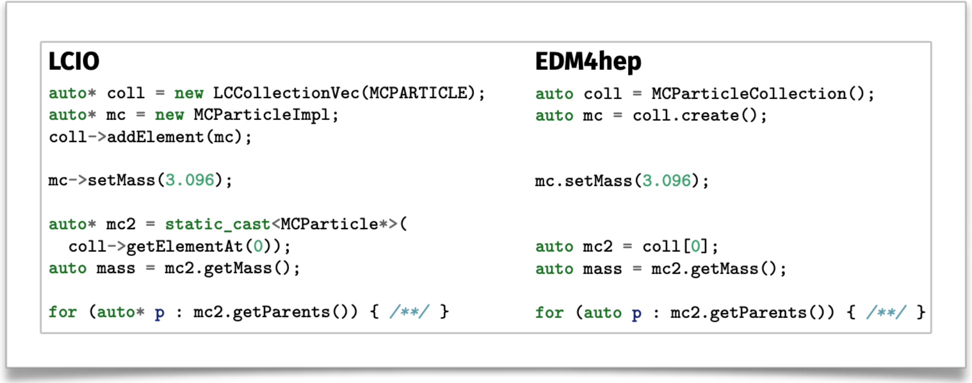
- PODIO: event data model toolkit
  - based on storing PODs (plain old data structures)
- EDM4hep uses PODIO to define a generic EDM for all future colliders
  - largely based on **LCIO- EDM**
  - ROOT I/O or SIO
  - offers more modern API and faster read times
- potentially the most difficult to replace in existing code base during a transition from iLCSoft to Key4Hep…




Total time spent in I/O operations and file size







LCIO

```
auto* coll = new LCCollectionVec(MCPARTICLE);
auto* mc = new MCParticleImpl;
coll->addElement(mc);

mc->setMass(3.096);

auto* mc2 = static_cast<MCParticle*>(
  coll->getElementAt(0));
auto mass = mc2.getMass();

for (auto* p : mc2.getParents()) { /**/ }
```

EDM4hep

```
auto coll = MCParticleCollection();
auto mc = coll.create();


mc.setMass(3.096);


auto mc2 = coll[0];
auto mass = mc2.getMass();

for (auto p : mc2.getParents()) { /**/ }
```

see talk T.Madlener Monday

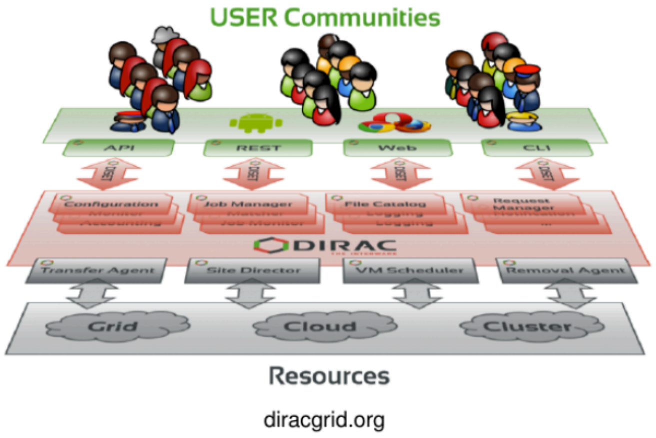
ILCDirac

managing large scale productions for ILC and CLIC

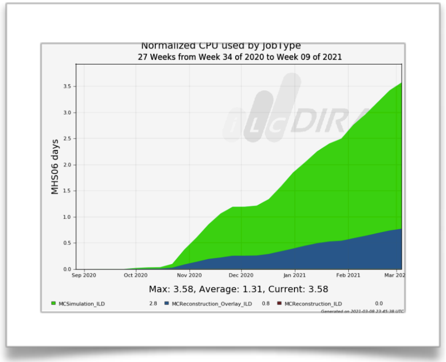
- central, large scale Monte Carlo production campaigns done with ILCDirac
  - “ILCDIRAC, a DIRAC extension for the linear collider community” DOI: 10.1088/1742-6596/513/3/032077
  - managed by CERN/CLICdp
- also individual user jobs done with ILCDirac

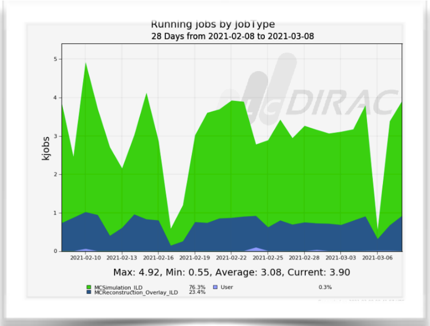
- resources for ILC-VO provided by all/most large EGI Grid sites
  - up to now no resource management for CPU done inside ILC-VO
  - rely on resources provided voluntarily by sites
    - typically a very small percentage of what they provide for LHC

- storage provided as needed by contributing labs: DESY, CERN, KEK,…

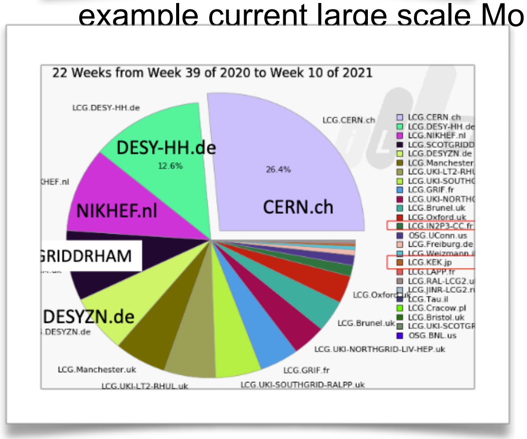


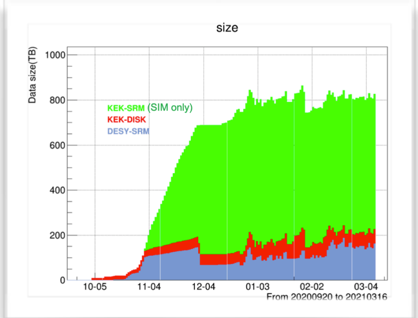
see talks by A.Sailer and H.Ono on Monday





example current large scale Monte Carlo production for ILD



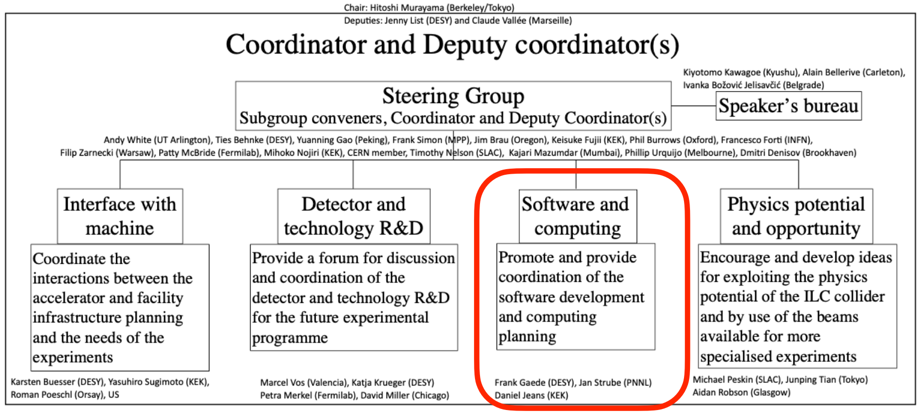


IDT-WG3 Software Working Group

preparing the ILC pre-lab

- Software and Computing working group created
- advance SW&Comp for the ILC in the IDT phase
- open to everyone interested in contributing to ILC software and computing activities
  - contact us if you want to get involved

WG3 Organisation and mandates



Chair: Hitoshi Murayama (Berkeley/Tokyo)
Deputies: Jenny List (DESY) and Claude Vallée (Marseille)

Coordinator and Deputy coordinator(s)

Steering Group
Subgroup conveners, Coordinator and Deputy Coordinator(s)

Kiyotomo Kawagoe (Kyushu), Alain Bellerive (Carleton), Ivanka Božović Jelisavčić (Belgrade)

Speaker's bureau

Andy White (UT Arlington), Ties Behnke (DESY), Yuanning Gao (Peking), Frank Simon (MPP), Jim Brau (Oregon), Keisuke Fujii (KEK), Phil Burrows (Oxford), Francesco Forti (INFN), Filip Zarnecki (Warsaw), Patty McBride (Fermilab), Mihoko Nojiri (KEK), CERN member, Timothy Nelson (SLAC), Kajari Mazumdar (Mumbai), Phillip Urquijo (Melbourne), Dmitri Denisov (Brookhaven)

Interface with machine

Coordinate the interactions between the accelerator and facility infrastructure planning and the needs of the experiments

Karsten Buesser (DESY), Yasuhiro Sugimoto (KEK), Roman Poeschl (Orsay), US

Detector and technology R&D

Provide a forum for discussion and coordination of the detector and technology R&D for the future experimental programme

Marcel Vos (Valencia), Katja Krueger (DESY) Petra Merkel (Fermilab), David Miller (Chicago)

Software and computing

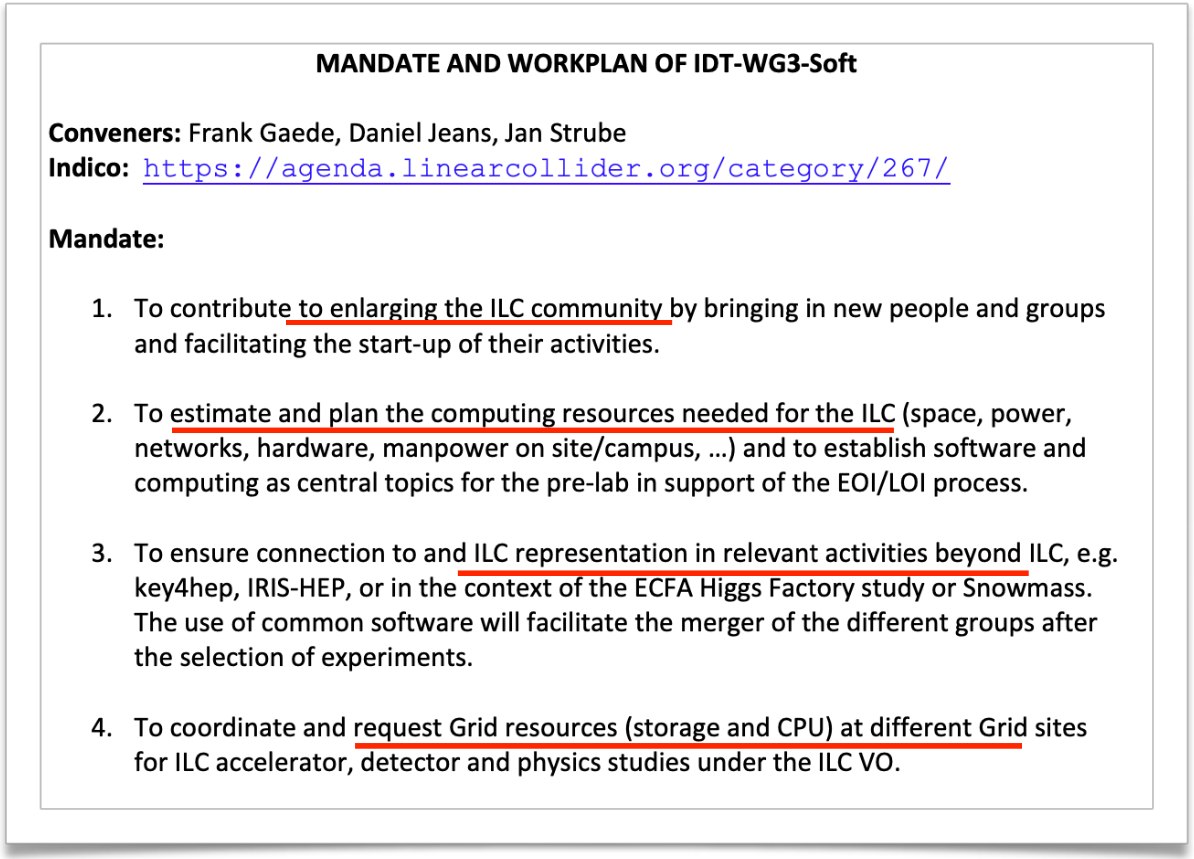
Promote and provide coordination of the software development and computing planning

Frank Gaede (DESY), Jan Strube (PNNL) Daniel Jeans (KEK)

Physics potential and opportunity

Encourage and develop ideas for exploiting the physics potential of the ILC collider and by use of the beams available for more specialised experiments

Michael Peskin (SLAC), Junping Tian (Tokyo) Aidan Robson (Glasgow)



MANDATE AND WORKPLAN OF IDT-WG3-Soft

Conveners: Frank Gaede, Daniel Jeans, Jan Strube
Indico: https://agenda.linearcollider.org/category/267/

Mandate:

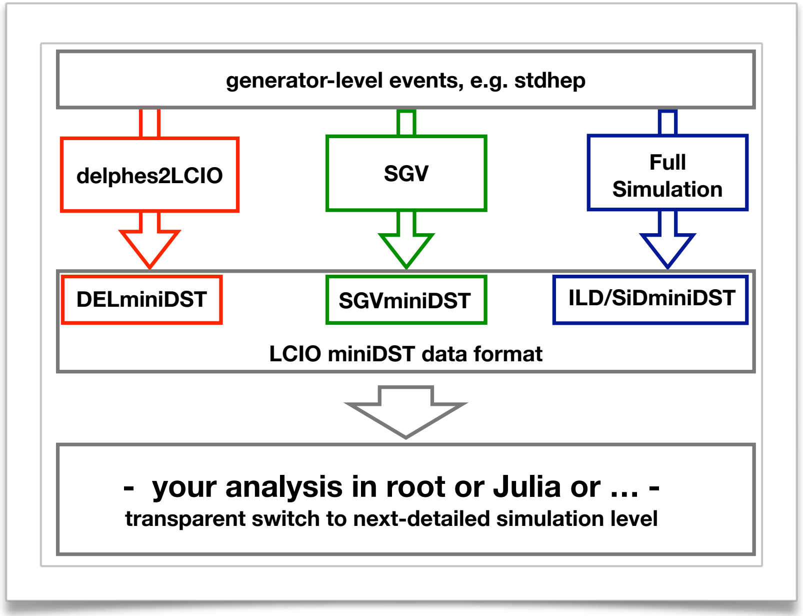
1. To contribute to enlarging the ILC community by bringing in new people and groups and facilitating the start-up of their activities.
2. To estimate and plan the computing resources needed for the ILC (space, power, networks, hardware, manpower on site/campus, …) and to establish software and computing 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.

Supporting the LC community

tutorials, documentation, examples

- quite some activity for promoting ILC software and analysis triggered by snowmass process in the US (to be continued by IDT-WG3-SW):
  - generic **ILC-detector Delphes card**
  - **delphes2LCIO** tool
  - definition of **mini-DST** format
  - large SM samples available
  - simple analysis examples w/ ROOT/LCIO only
    - also in Jupyter notebook form -> direct access to data
  - a number of (beginners) **tutorials**:
  - go there to get started w/ ILC

http://ilcsnowmass.org/



generator-level events, e.g. stdhep

delphes2LCIO → DELminiDST

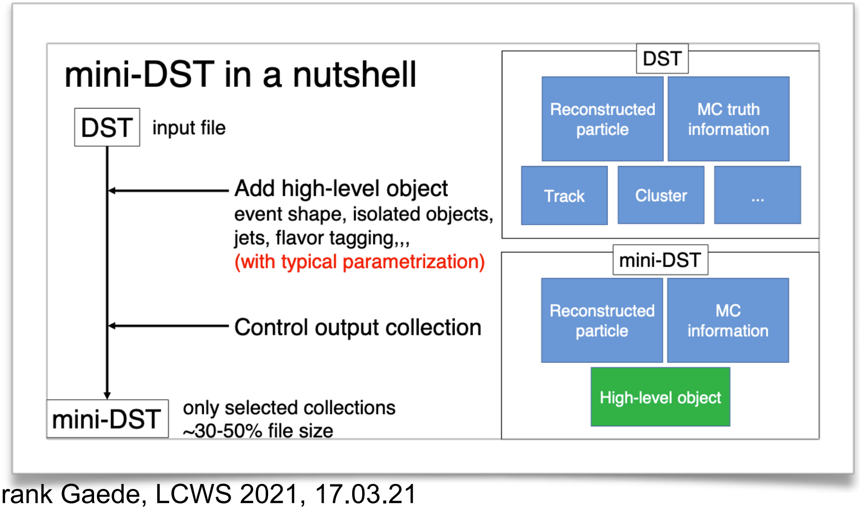
SGV → SGVminiDST

Full Simulation → ILD/SiDminiDST

LCIO miniDST data format

- your analysis in root or Julia or … -

transparent switch to next-detailed simulation level



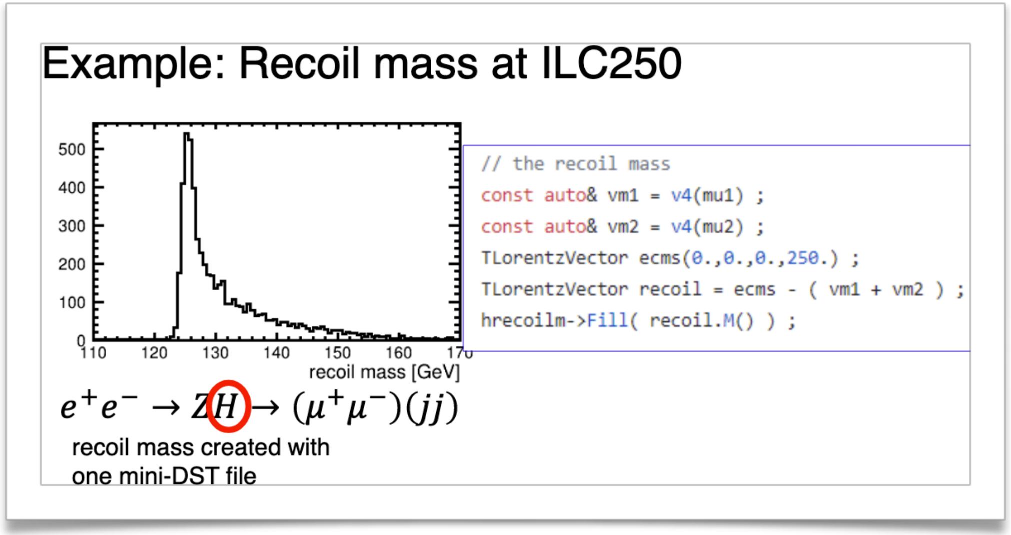
mini-DST in a nutshell

DST input file

Add high-level object
event shape, isolated objects, jets, flavor tagging,,,
(with typical parametrization)

Control output collection

mini-DST only selected collections
~30-50% file size



Example: Recoil mass at ILC250

```
// the recoil mass
const auto& vm1 = v4(mu1) ;
const auto& vm2 = v4(mu2) ;
TLorentzVector ecms(0.,0.,0.,250.) ;
TLorentzVector recoil = ecms - ( vm1 + vm2 ) ;
hrecoilm->Fill( recoil.M() ) ;
```

$e^+e^- \to ZH \to (\mu^+\mu^-)(jj)$

recoil mass created with one mini-DST file

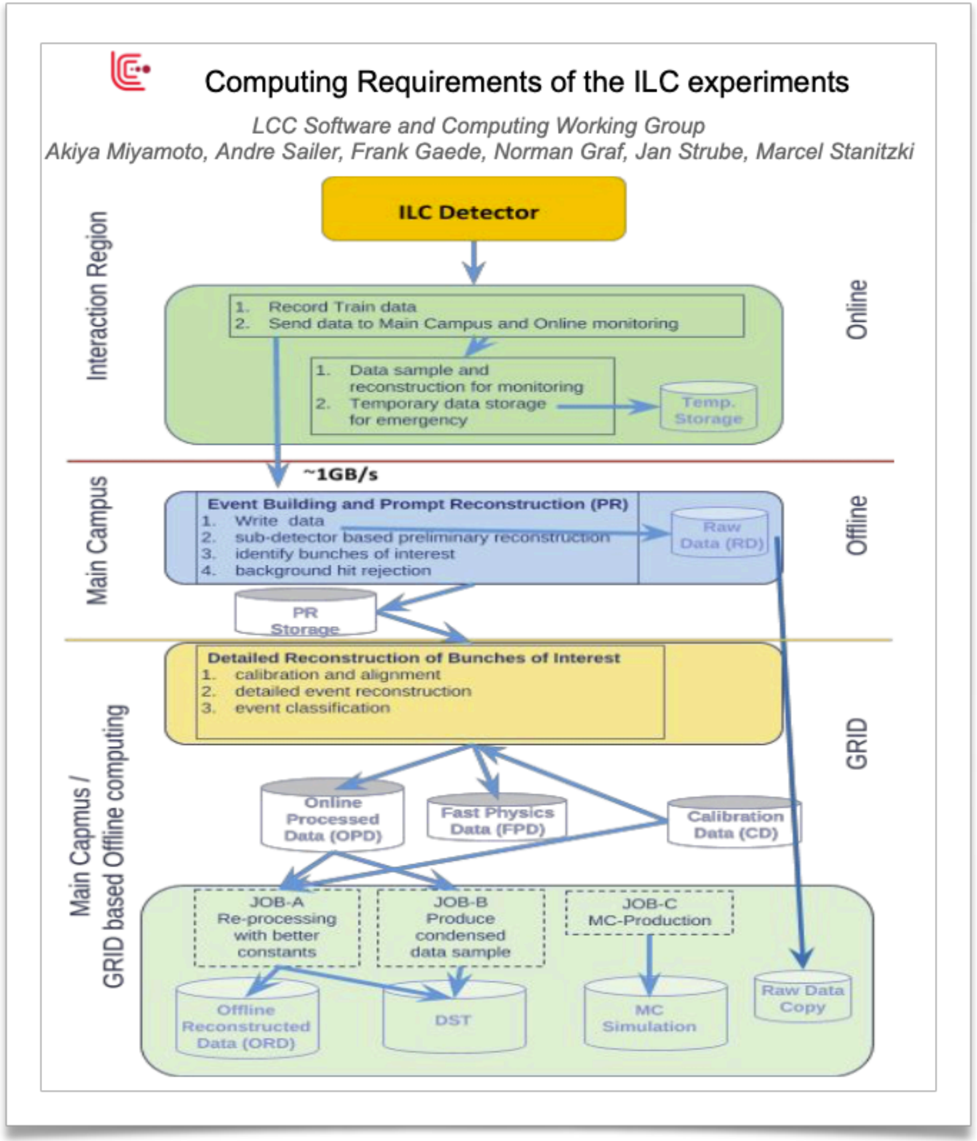
ILC Computing

Computing model and resources

- for the preparation of the ILC it is important to have a **realistic and reliable estimate** of the required **computing resources**
  - during **IDT, pre-lab phase: EOIs, LOIs, TDRs**
  - on **campus** and **at the collision point**
- last estimate done in 2014 and updated in 2017
  - need to iterate taking recent developments and trends into account

Access Building
Central Substation
Facility Building
154 kV Receive
Water Chiller & Pumps
Air Intake/Exhaust
66 kV Co-Generation
中央変電所
搬入棟
LNG for Co-Generator
He Compressor
Detector Assembling Building
Detector Preparing Building
Research Building
Computer Building
©Rey.Hori/KEK

Collision Point Surface-Campus



Computing Requirements of the ILC experiments

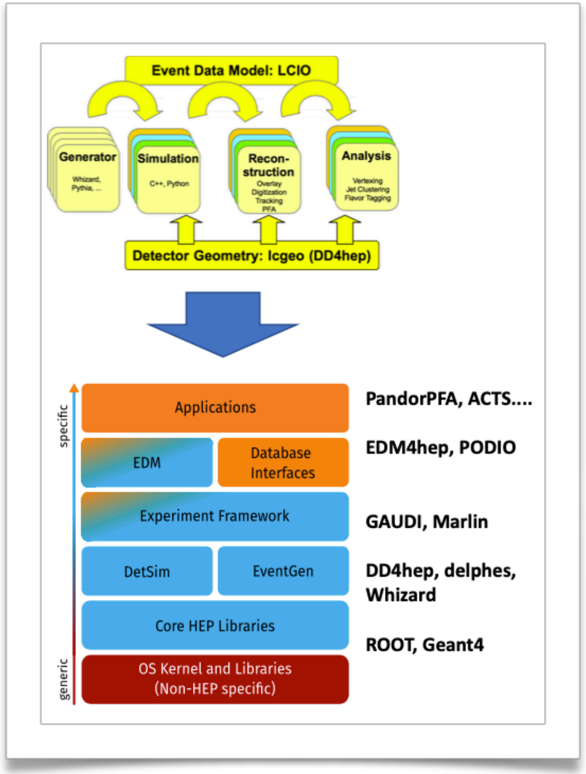
LCC Software and Computing Working Group

Akiya Miyamoto, Andre Sailer, Frank Gaede, Norman Graf, Jan Strube, Marcel Stanitzki

Interaction Region
Online
ILC Detector
1. Record Train data
2. Send data to Main Campus and Online monitoring
1. Data sample and reconstruction for monitoring
2. Temporary data storage for emergency
Temp. Storage
~1GB/s
Main Campus
Offline
Event Building and Prompt Reconstruction (PR)
1. Write data
2. sub-detector based preliminary reconstruction
3. identify bunches of interest
4. background hit rejection
Raw Data (RD)
PR Storage
Main Capmus / GRID based Offline computing
GRID
Detailed Reconstruction of Bunches of Interest
1. calibration and alignment
2. detailed event reconstruction
3. event classification
Online Processed Data (OPD)
Fast Physics Data (FPD)
Calibration Data (CD)
JOB-A Re-processing with better constants
JOB-B Produce condensed data sample
JOB-C MC-Production
Offline Reconstructed Data (ORD)
DST
MC Simulation
Raw Data Copy

Summary and Outlook

- iLCSoft provides the **framework**, **simulation** and **reconstruction** tools for detector and physics studies for ILC, CLIC and beyond
  - developed as a LC community effort for more than 15 years
- currently see a lot of activity in developing new **HLR algorithms**
  - **Classical** as well as **Machine Learning**
- in 2020 we started a new future collider community wide effort to put together a modern turnkey software stack: **Key4Hep**
  - **transition of existing iLCSoft tools to Key4hep need careful preparation**
  - **challenging to keep full functional code base at all times**
- **IDT-WG3-SW** working group created
  - aiming at increasing the ILC (SW) community and providing the support and coordination to accompany this development



Event Data Model: LCIO
Generator
Whizard, Pythia, ...
Simulation
C++, Python
Recon-struction
Overlay
Digitization
Tracking
PFA
Analysis
Vertexing
Jet Clustering
Flavor Tagging
Detector Geometry: lcgeo (DD4hep)
specific
generic
Applications
EDM
Database Interfaces
Experiment Framework
DetSim
EventGen
Core HEP Libraries
OS Kernel and Libraries (Non-HEP specific)
PandorPFA, ACTS....
EDM4hep, PODIO
GAUDI, Marlin
DD4hep, delphes, Whizard
ROOT, Geant4