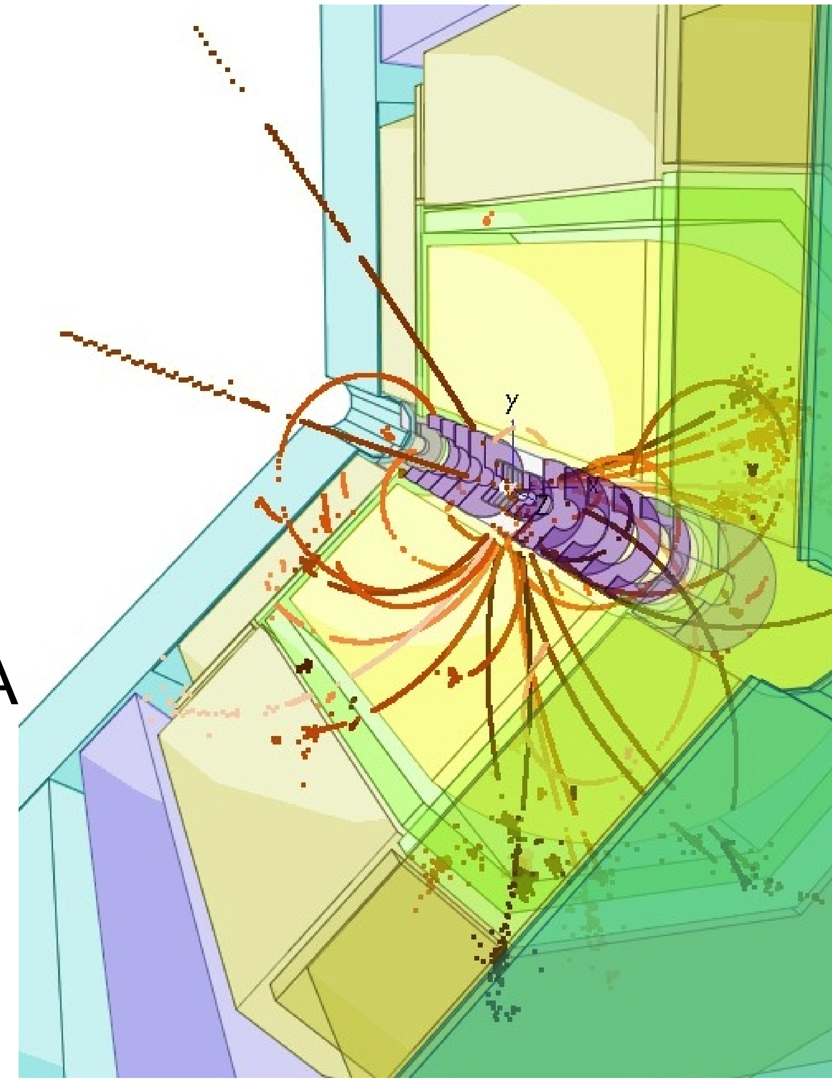
ILC Software
Overview and recent developments

Frank Gaede
134th ILC@DESY General Project Meeting
DESY, May 27, 2016

Outline

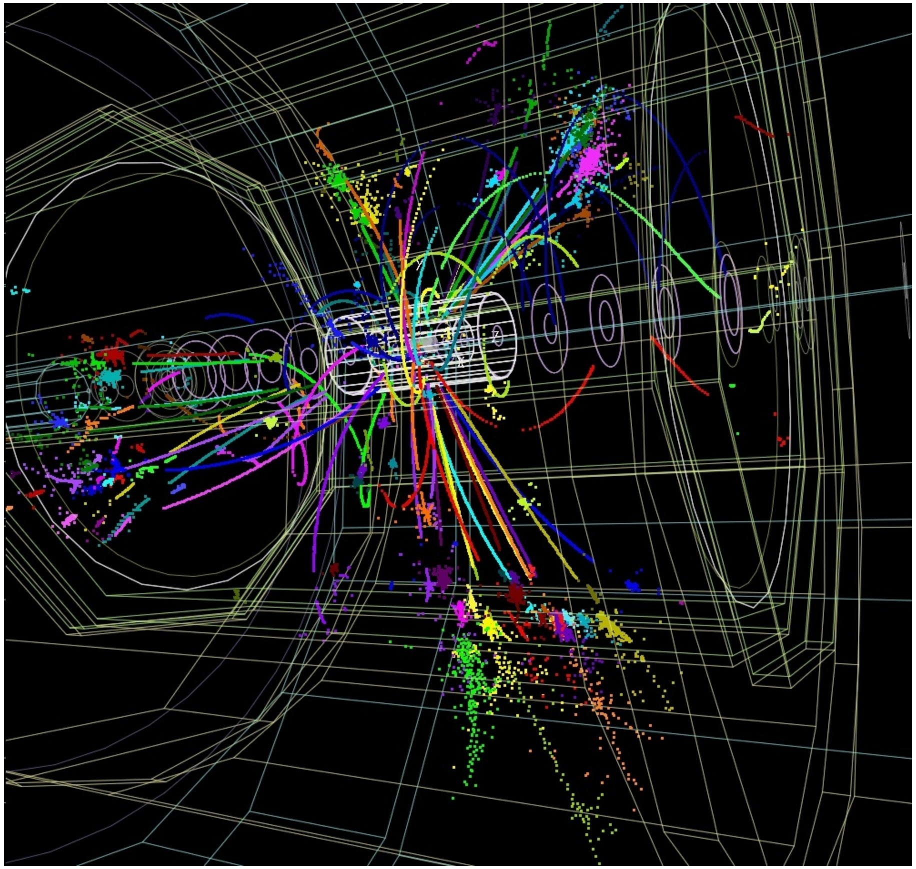
- Introduction to iLCSoft
- core tools
- ILD simulation and reconstruction software chain
- new developments
- DD4hep, DDRec, DDMarlinTrk/PFA
- some results
- Summary and Outlook



ILC detectors and software

ILC detectors

- very high tracking efficiency
- extremely good impact parameter and momentum resolution
- unprecedented jet-energy resolution
- optimized for particle flow **PFA**



$$\sigma_{E_{jet}}^2 = \epsilon_{trk}^2 \sum_i E_{trk,i}^4 + \epsilon_{ECal}^2 E_{ECal} + \epsilon_{HCal}^2 E_{HCal} + \sigma_{confusion}^2$$

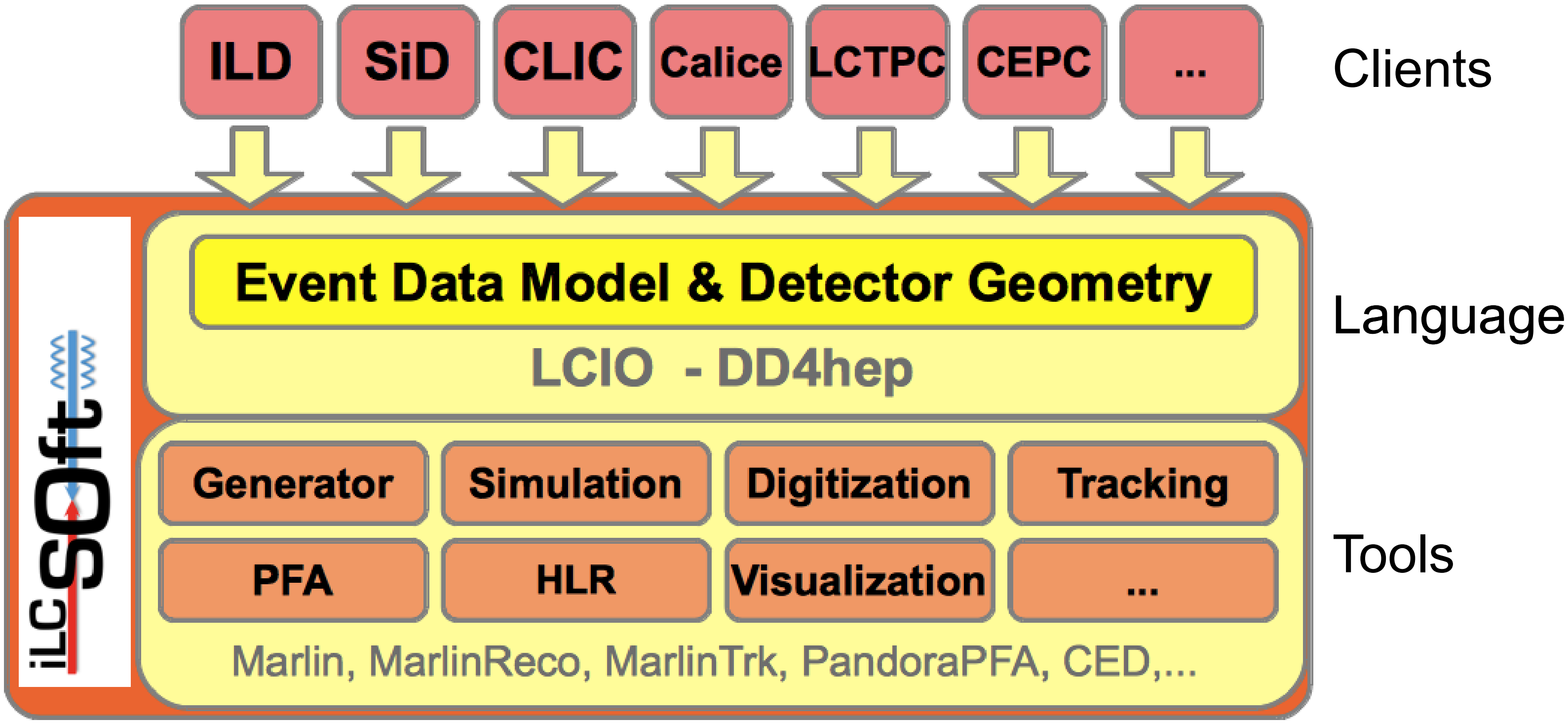
$$\epsilon_{trk} = \delta(1/p) \approx 5\cdot 10^{-5}, \quad \epsilon_{ECal} = \frac{\delta E}{\sqrt{E}} \approx 0.2, \quad \epsilon_{HCal} \approx 0.5$$

ILC software

- need flexible and easy to use tools
- allowing for fast and detailed simulation of detector variants
- efficient algorithms for reconstruction and analysis

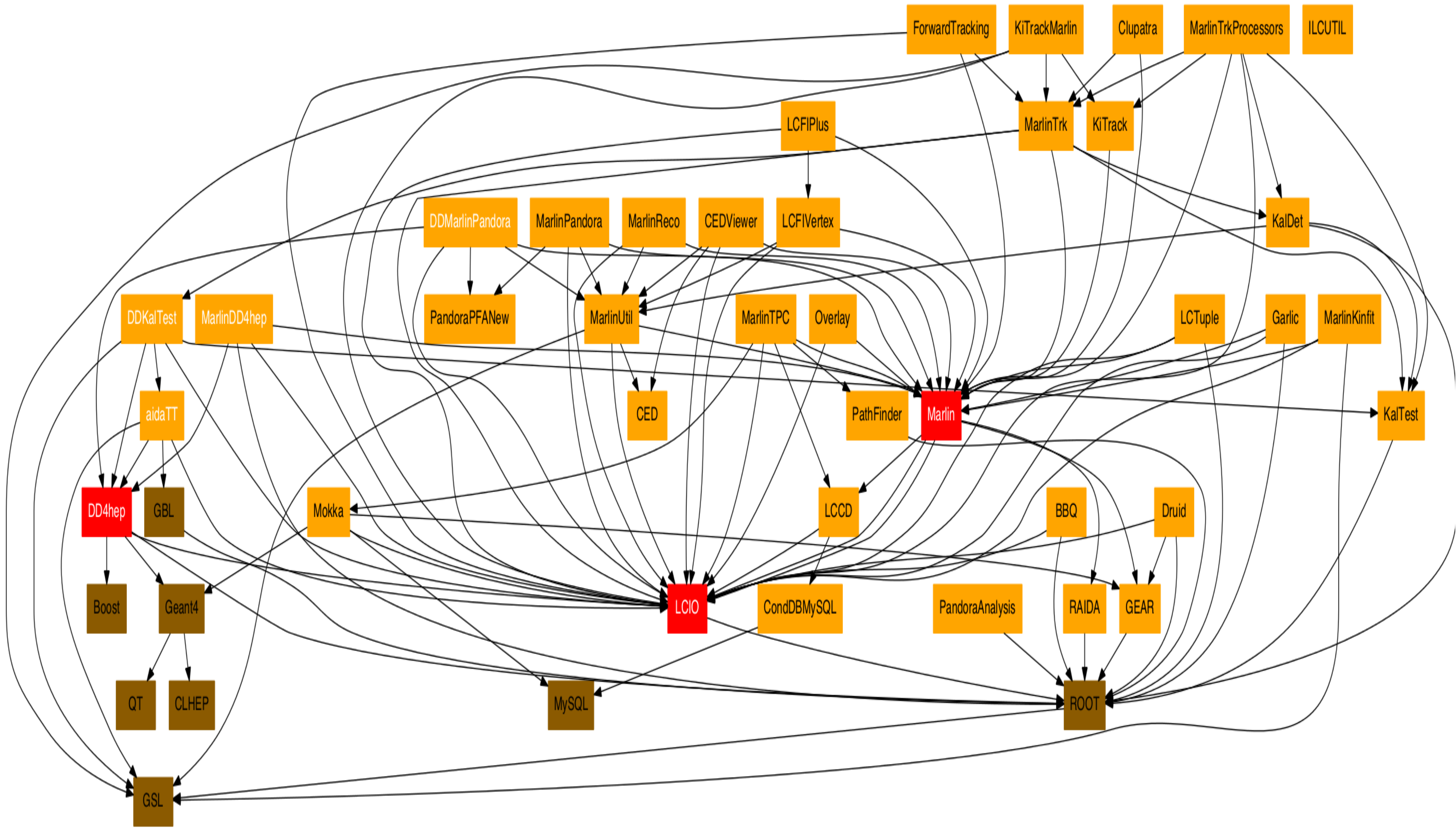
achieving optimal detector performance requires optimal detector and software

the iLCSoft framework



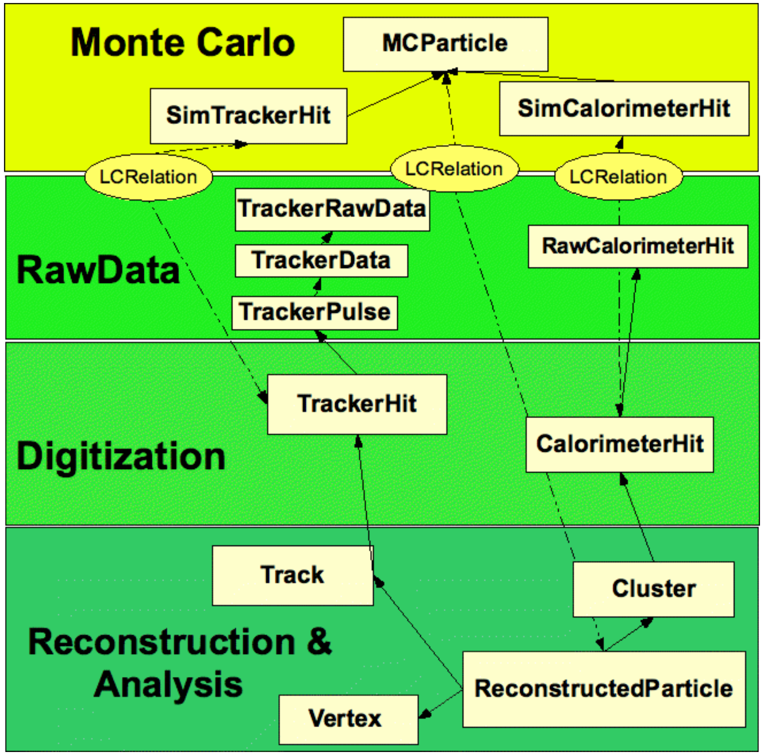
- well defined **interfaces** to Event Data and Detector Geometry
- define "Language" through which actual tools are used
- allows for flexible re-use of tools by many clients (all LC community)
  - DESY involved in almost all aspects of the framework

iLCSoft packages



LCIO - common Event Data Model

- common event data model (EDM) and persistency for linear collider community
  - joined DESY and SLAC (and LLR) project - first presented @ CHEP **2003**
  - used by ILD, SiD, CLICdp and test beams for more than 10 years
  - **common EDM proven to be crucial for collaborative SW development across detector concepts**



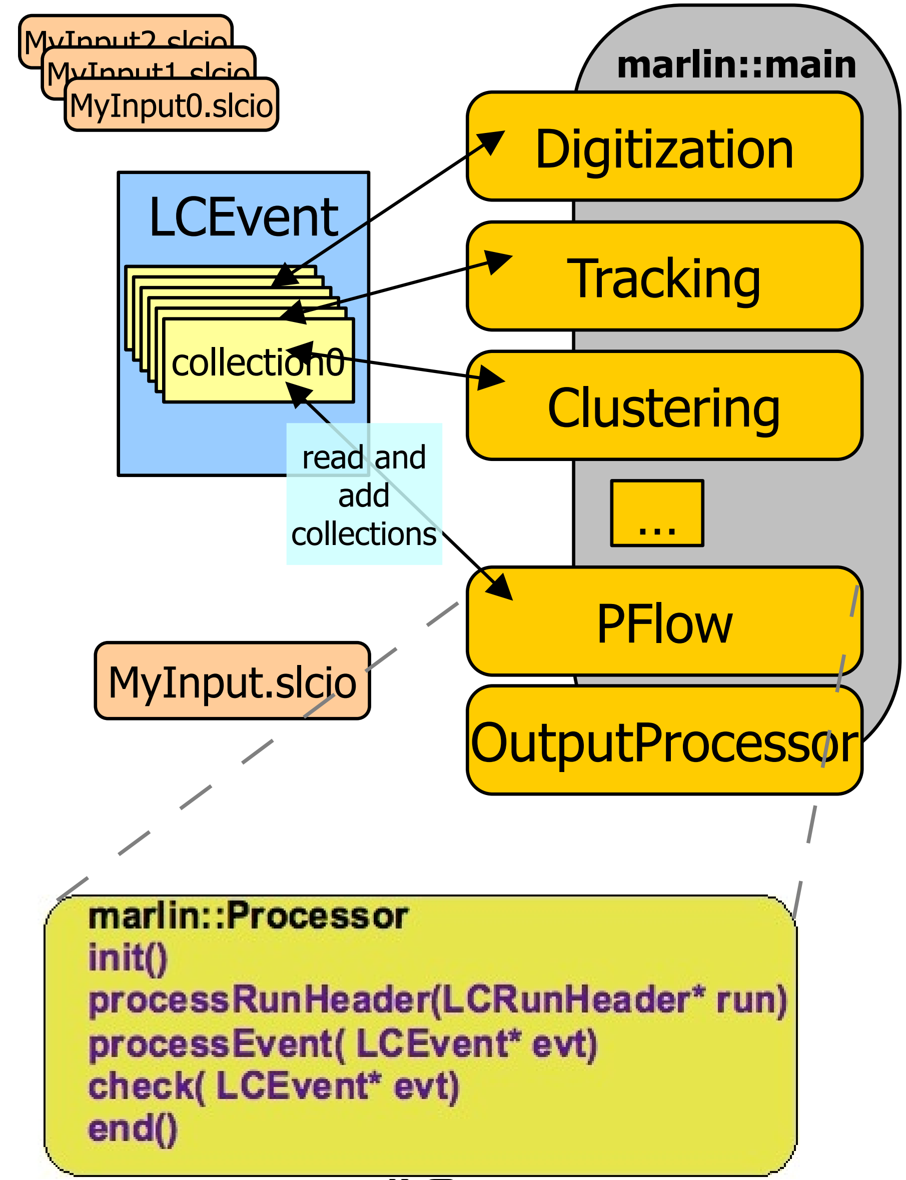
- develop new I/O layer in AIDA2020: Podio (with FCC)
  - allow for efficient parallel I/O and
  - simplify parallel processing (GPUs)

Marlin – application framework

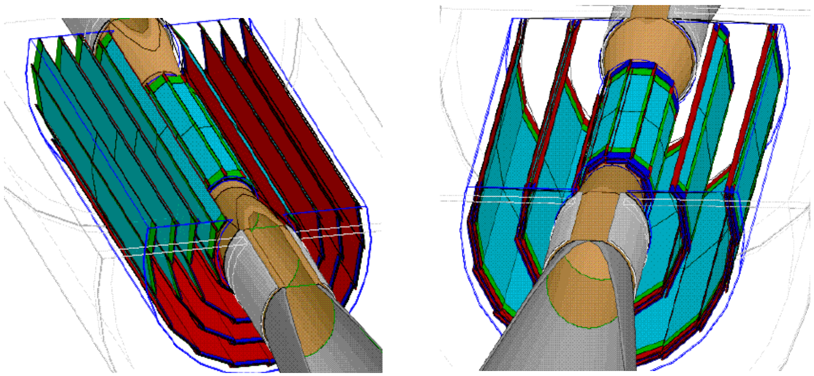
Modular **A**nalysis & **R**econstruction for the **L I N**ear Collider

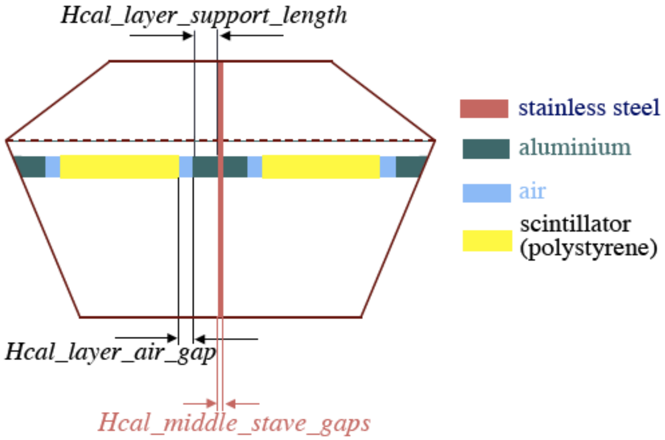
- modular C++ application framework for the analysis and reconstruction of ILC data
- **LCIO** as transient data model
  - event data bus/white board model
- xml steering files:
  - fully configure application
  - order of modules/processors
  - parameters global + processor
- self documenting
  - parameters registered in user code
- consistency check of input/output collection types

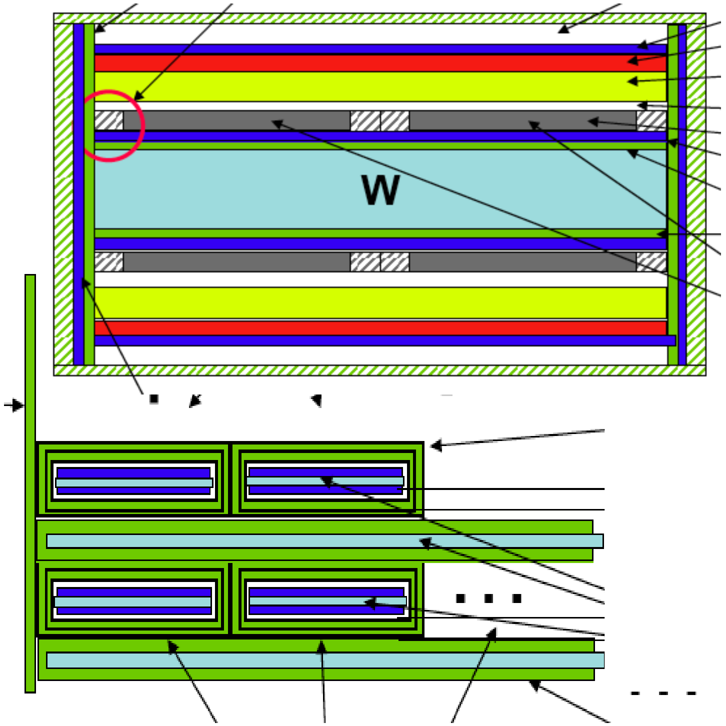
- **Plug & Play** of modules



ILD detector in Mokka simulation







- for DBD created detailed simulation model of ILD in Mokka
- engineering level of detail:
  - cables and services
  - electronics
  - gaps and imperfections
- goal:
  - realistic material budget for tracking
  - realistic PFA performance and efficiencies

Digitization in a nutshell

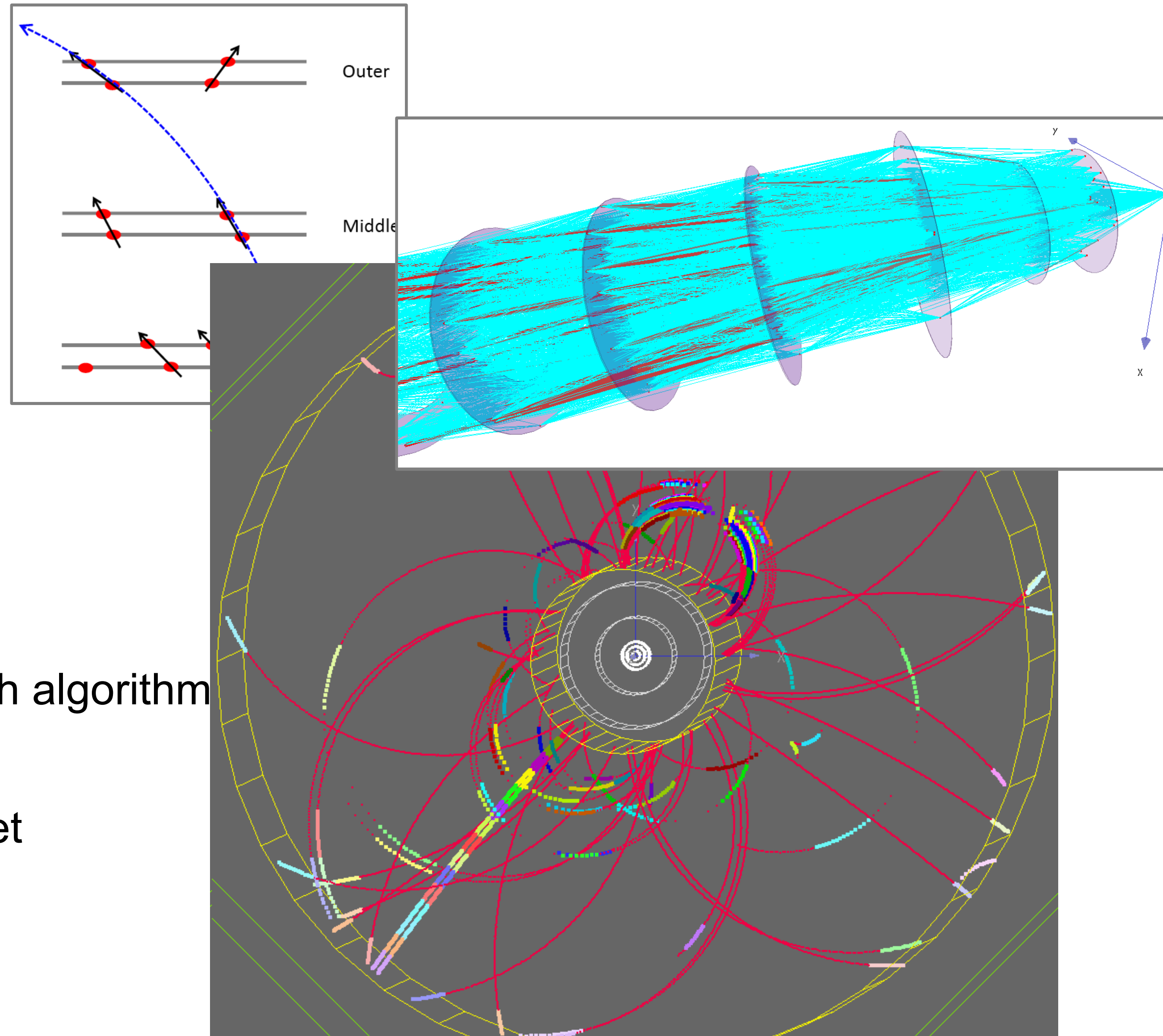
- Si-trackers and TPC
  - parameterized smearing according to established point resolutions
  - space point builder for double layers of 1d strip detectors

Detector	Point Resolution
VTX	$\sigma_{r\phi,z} = 2.8\mu\mathrm{m}$ (layer 1)
	$\sigma_{r\phi,z} = 6.0\mu\mathrm{m}$ (layer 2)
	$\sigma_{r\phi,z} = 4.0\mu\mathrm{m}$ (layers 3-6)
SIT	$\sigma_{\alpha_z} = 7.0\mu\mathrm{m}$
	$\alpha_z = \pm 7.0^\circ$ (angle with z-axis)
SET	$\sigma_{\alpha_z} = 7.0\mu\mathrm{m}$
	$\alpha_z = \pm 7.0^\circ$ (angle with z-axis)
FTD *Pixel*	$\sigma_r = 3.0\mu\mathrm{m}$
	$\sigma_{r_\perp} = 3.0\mu\mathrm{m}$
FTD *Strip*	$\sigma_{\alpha_r} = 7.0\mu\mathrm{m}$
	$\alpha_r = \pm 5.0^\circ$ (angle with radial direction)
TPC	$\sigma^2_{r\phi} = (50^2 + 900^2 \sin^2\phi + \left((25^2/22) \times (4T/B)^2 \sin\theta\right)(z/\mathrm{cm}))\ \mu\mathrm{m}^2$
	$\sigma^2_z = (400^2 + 80^2 \times (z/\mathrm{cm}))\ \mu\mathrm{m}^2$
	where ϕ and θ are the azimuthal and polar angle of the track direction

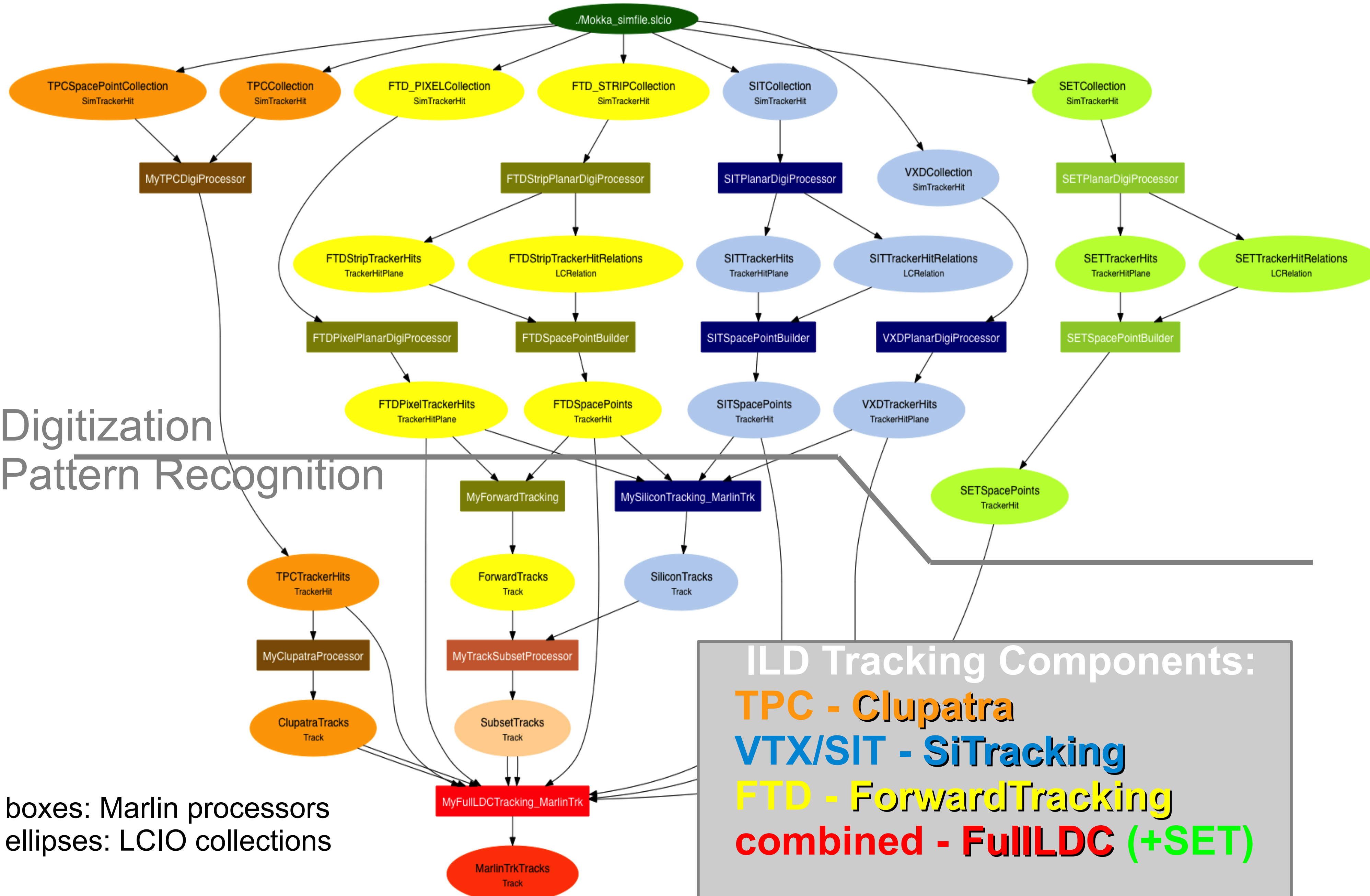
- Calorimeters:
  - simple 'calibration' - scaling of visible energy to deposited energy $\rightarrow$ sampling fraction
  - amplitude thresholds
  - post-DBD:
    - timing cuts
    - electronic noise
    - saturation effects
    - ….

ILD Pattern recognition (i.a.n.)

- VXD/SIT
  - triplet search
  - mini-vectors
  - cellular automatons
- FTD
  - cellular automatons
  - Hopfield Network
- TPC
  - toplogical clustering
  - followed by road search algorithm
- Combined
  - select consistent subset
  - assign leftover hits



Marlin modules for ILD Tracking



Digitization

Pattern Recognition

ILD Tracking Components:

TPC - Clupatra

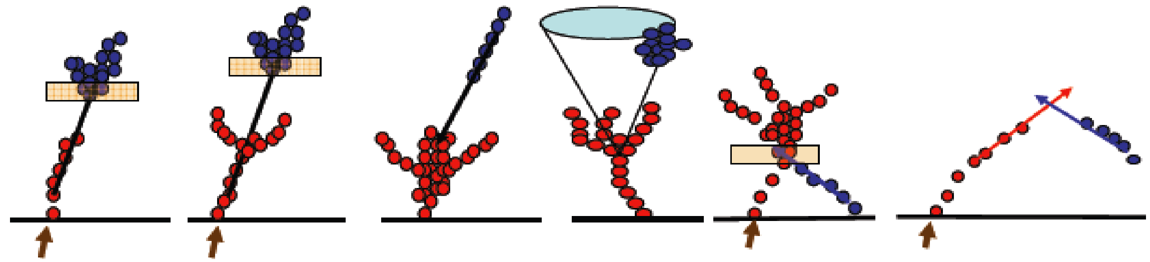
VTX/SIT - SiTracking

FTD - ForwardTracking

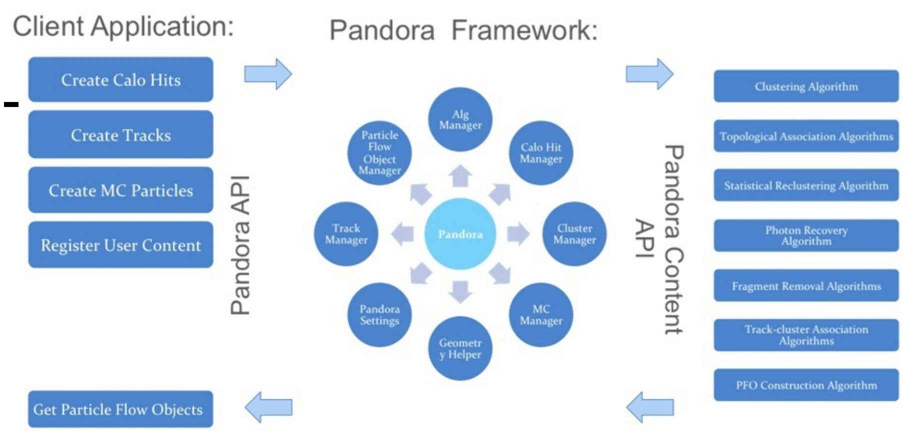
combined - FullLDC (+SET)

boxes: Marlin processors
ellipses: LCIO collections

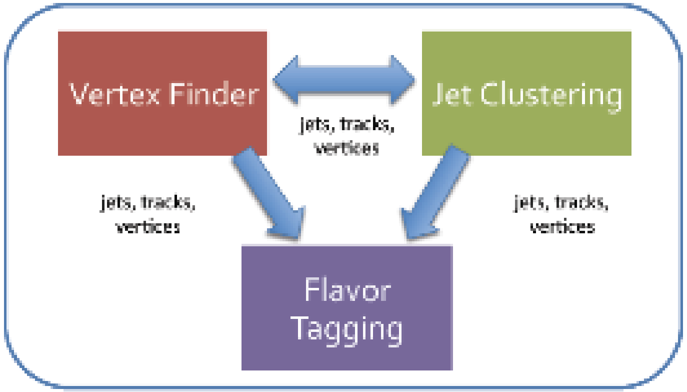
other reconstruction tools



- **PandoraPFA**
  - Particle Flow Algorithm
  - originally developed for ILD, now applied to
  - CLICdp, Calice, SiD, neutrino physics (Liquid-Ar TPC), LHC...
  - redesigned to be framework independent

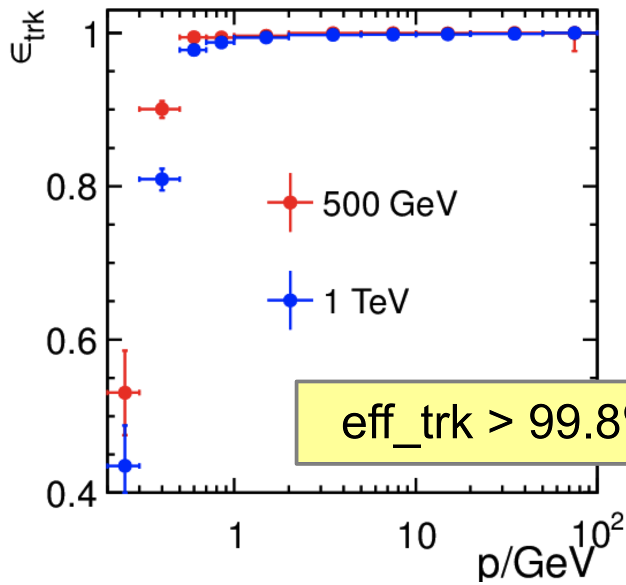


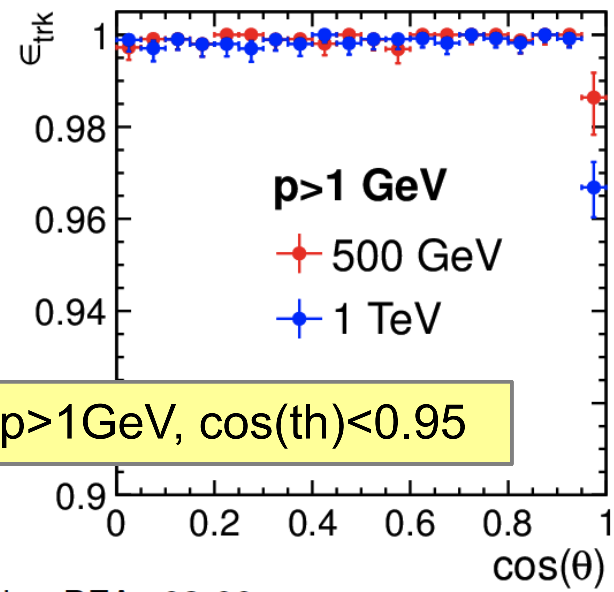
- **LCFIPlus**
  - flavor tagging - based on vertex reconstruction (ZVtop,...) and Neural Networks
  - used by all LC concepts (no geometry)



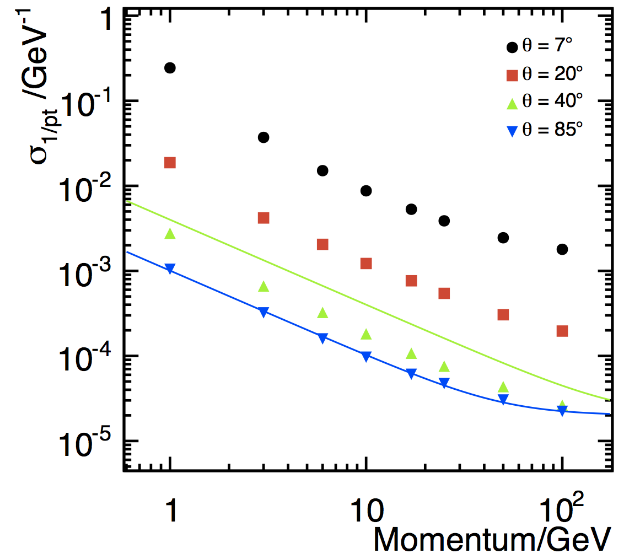
- **HLR**
  - jet finder, particleID, ...

achieved (detector) performance

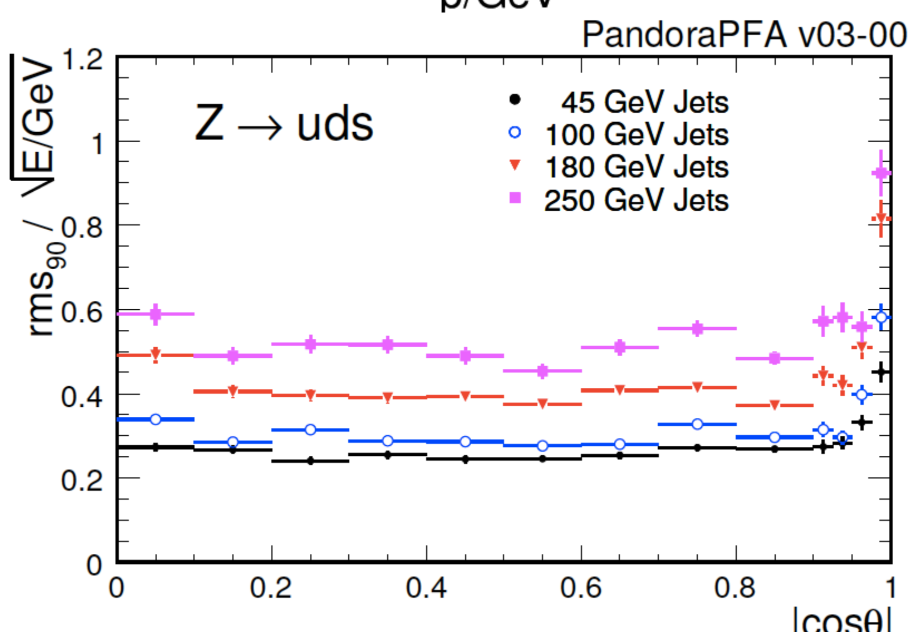




eff_trk > 99.8%, p>1GeV, cos(th)<0.95



$$\sigma_{1/p_T} = \frac{2 \times 10^{-5}}{GeV} \oplus \frac{1 \times 10^{-3}}{p_T \sin\theta}$$

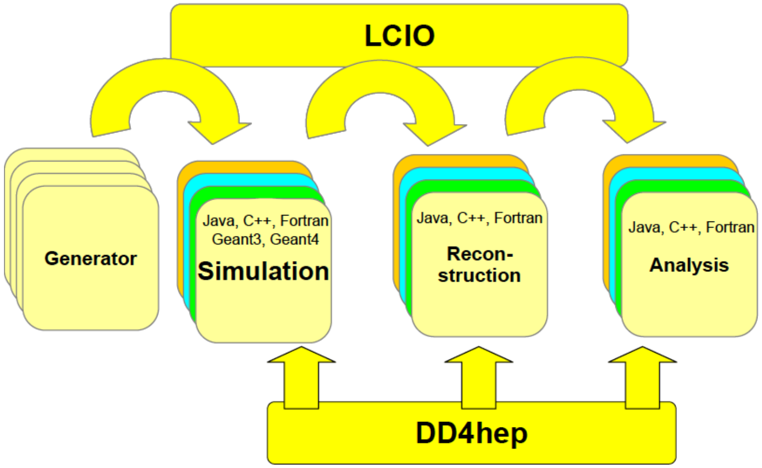


- achieved design goals for ILD detector performance
- **PFA** works !

plots from TDR Vol.4 (2013)

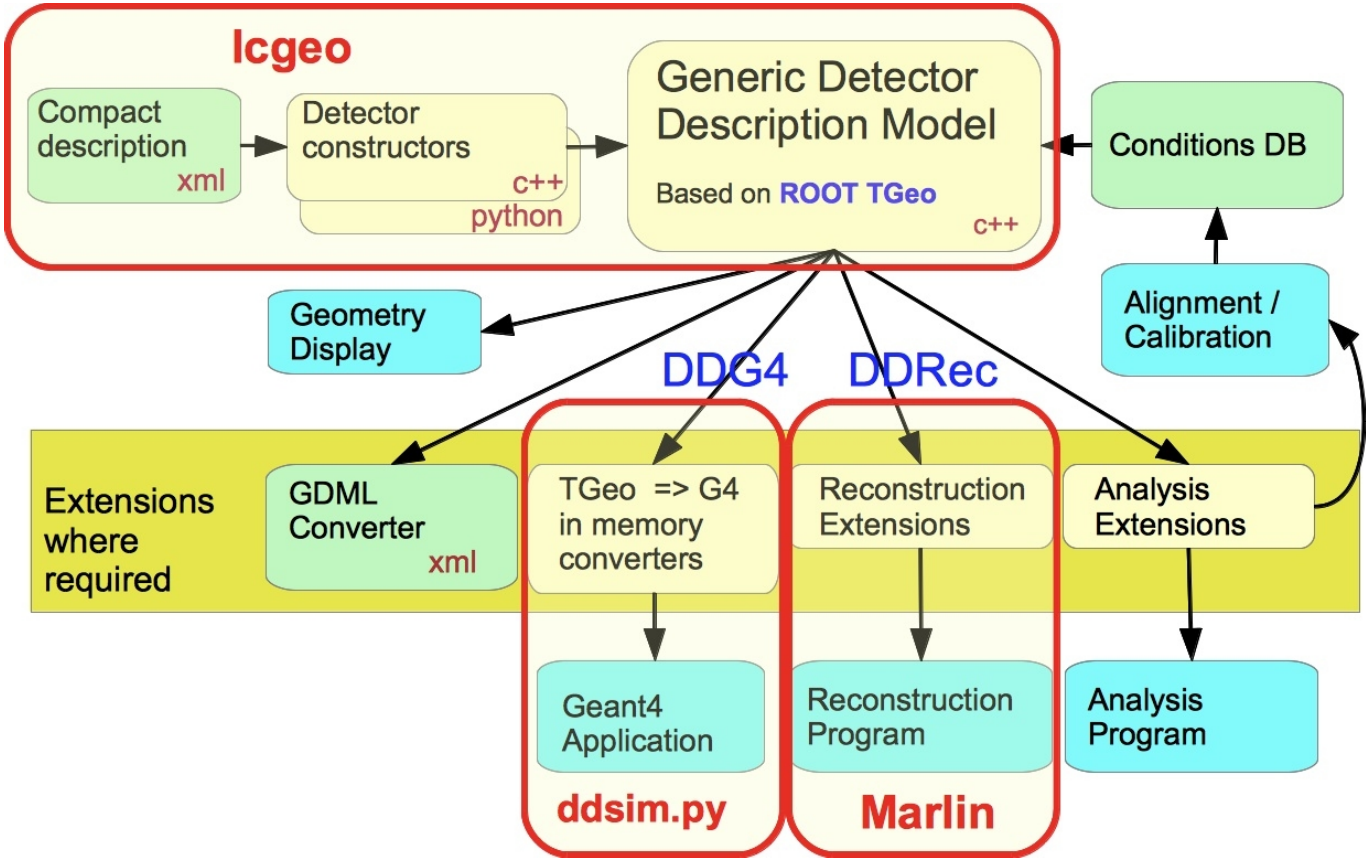
Why change a running system ?

- in Linear Collider Software Meetings 2012/2013 decided to use the **DD4hep** detector geometry description as a basis for common LC simulation and reconstruction framework



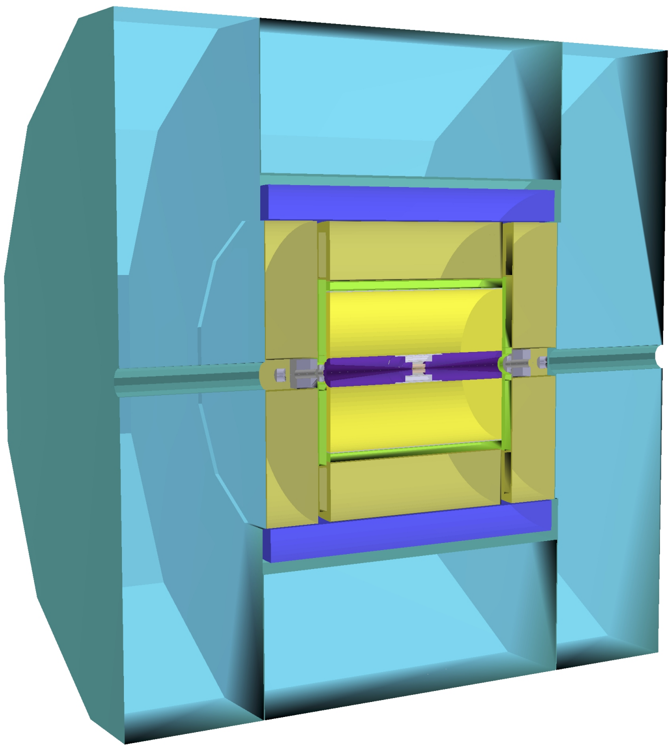
- defining a common geometry API is the second step - after the common EDM: LCIO - that is needed to have an open and modular software framework
- allows to share existing common software tools between ILC and CLIC and also develop new common simulation and reconstruction tools
- will provide more flexibility in changing detector layout or technologies for optimization studies

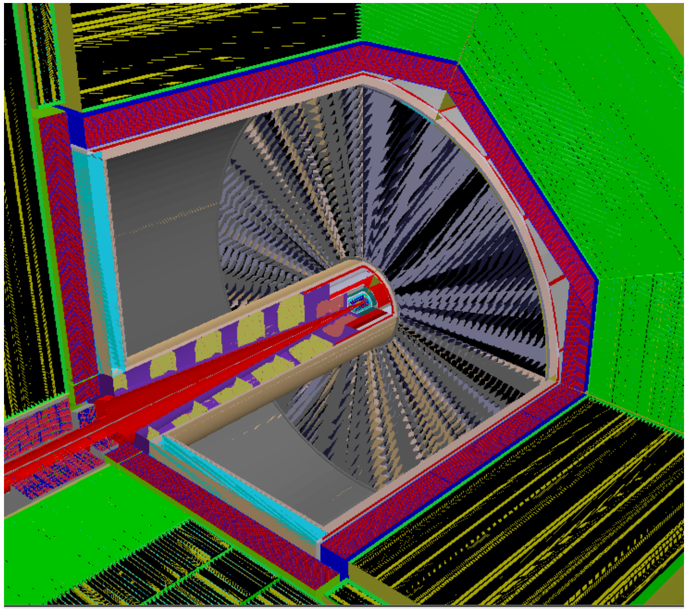
DD4hep - detector description



- one single source of geometry description for all HEP processing
  - DDG4: gateway Geant4 simulation
  - DDRec: interface to reconstruction
  - lcgeo: LC detector models (ILD, SiD, CLICdp)

ILD simulation model in lcgeo (DD4hep)

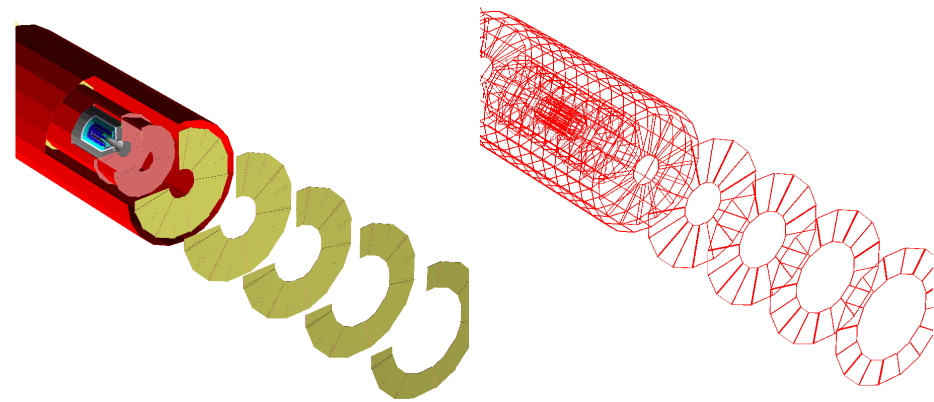




- ILD_o1_v05 Mokka model ported one-to-one to DD4hep
- introduced mandatory envelope volumes
  - validation and scaling behaviour
- model is fully functional and ready for detailed validation
- ddsim python simulation tool in place

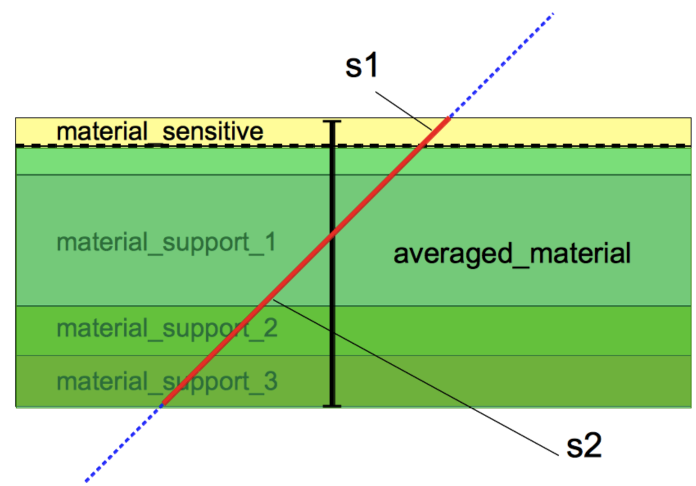
DDRec interface for reconstruction

- high level information on sub detectors for reconstruction - from detailed geometry model
  - detector layout
  - shapes, #layers, technology
  - material properties
  - … user defined ...



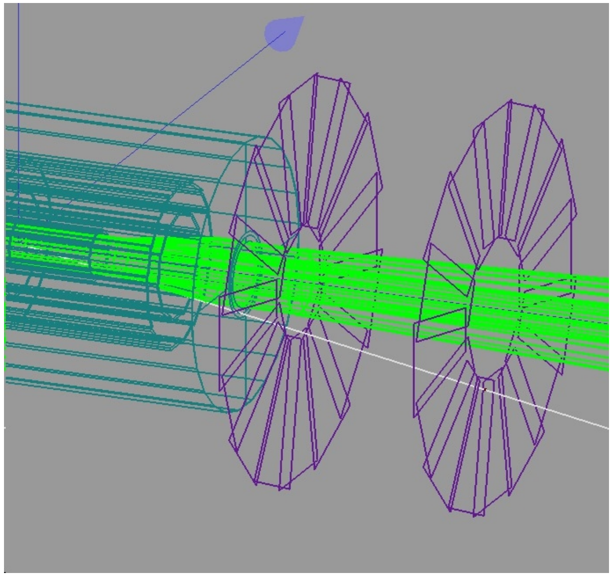
- tracking surfaces:
  - measurement directions
  - material effects for
    - multiple scattering
    - energy loss
  - automatically averaged from detailed model

  => consistent tracking geometry from one source



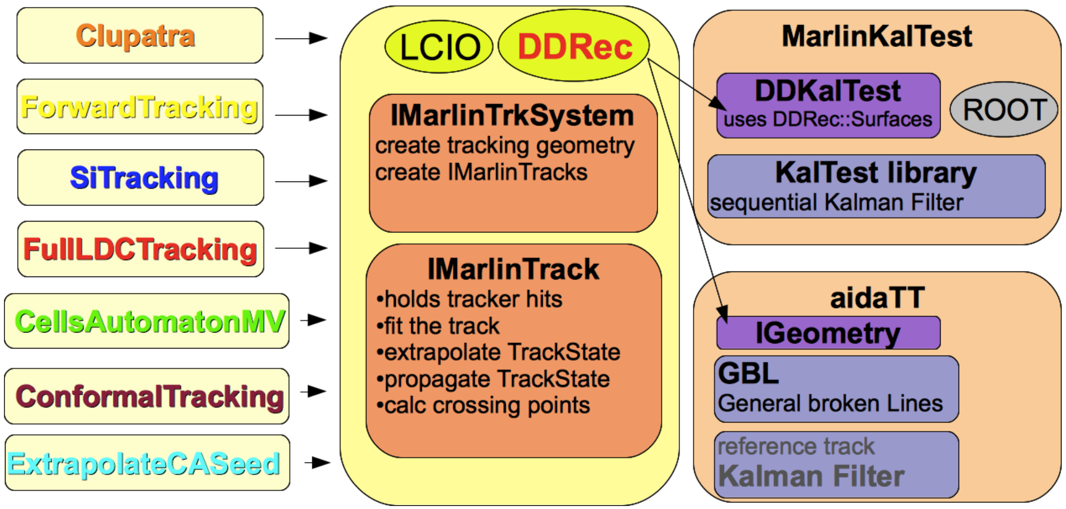
DDKalTest

- new package that provides measurement surfaces needed by KalTest using DDRec::Surfaces:
- DDPlanarMeasLayer
  - 1D,2D Si-tracker - barrel/endcap
  - dead materials (endcaps)
- DDCylinderMeasLayer
  - 2D hits in TPC
  - supports (cryostat, field cage,...)
- DDConeMeasLayer
  - conical sections of beam pipe

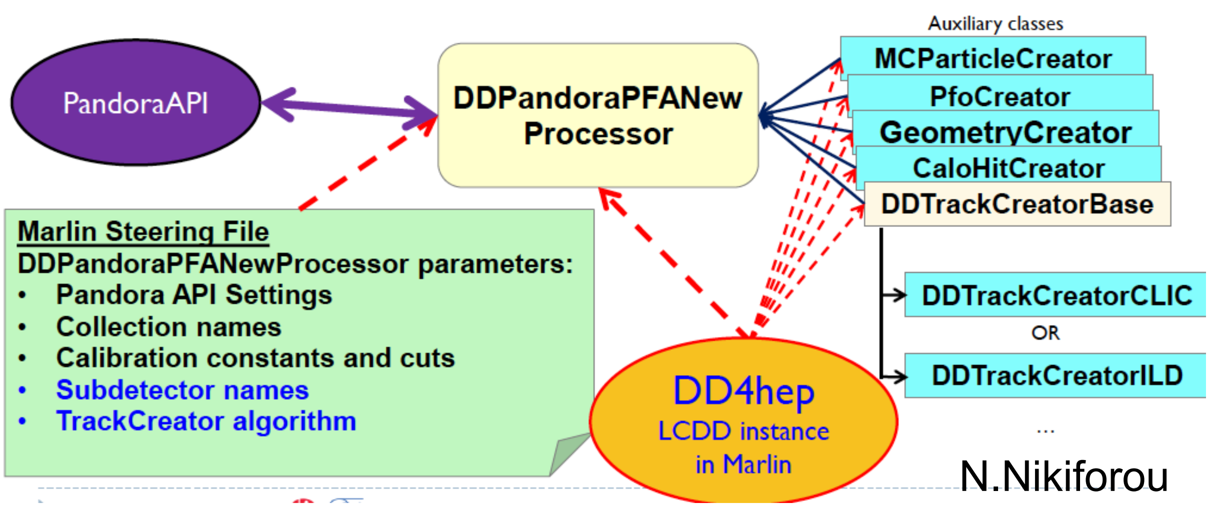


with DDKalTest we can run the track fitting for **every detector** that has a **DD4hep** geometry description (and the surfaces added) !
=> generic tracking package

Reconstruction Tools for DD4hep



- MarlinTrk tracking tools are now fully compatible w/ DD4hep
- can run existing pattern recognition
  - aidaTT-GBL allows for alignment studies

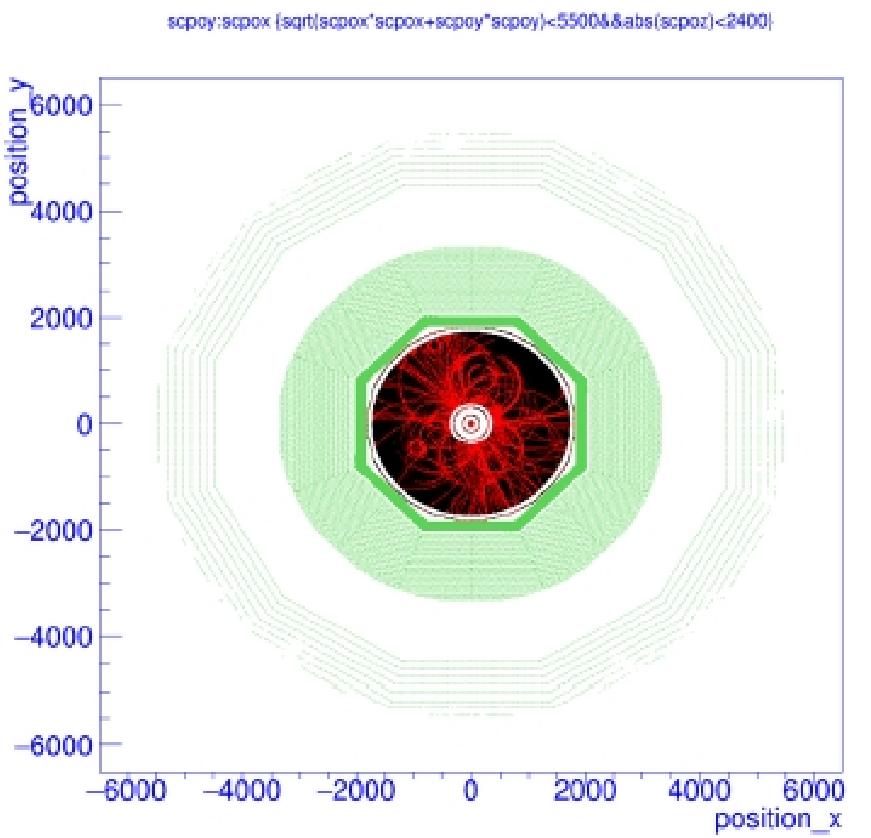


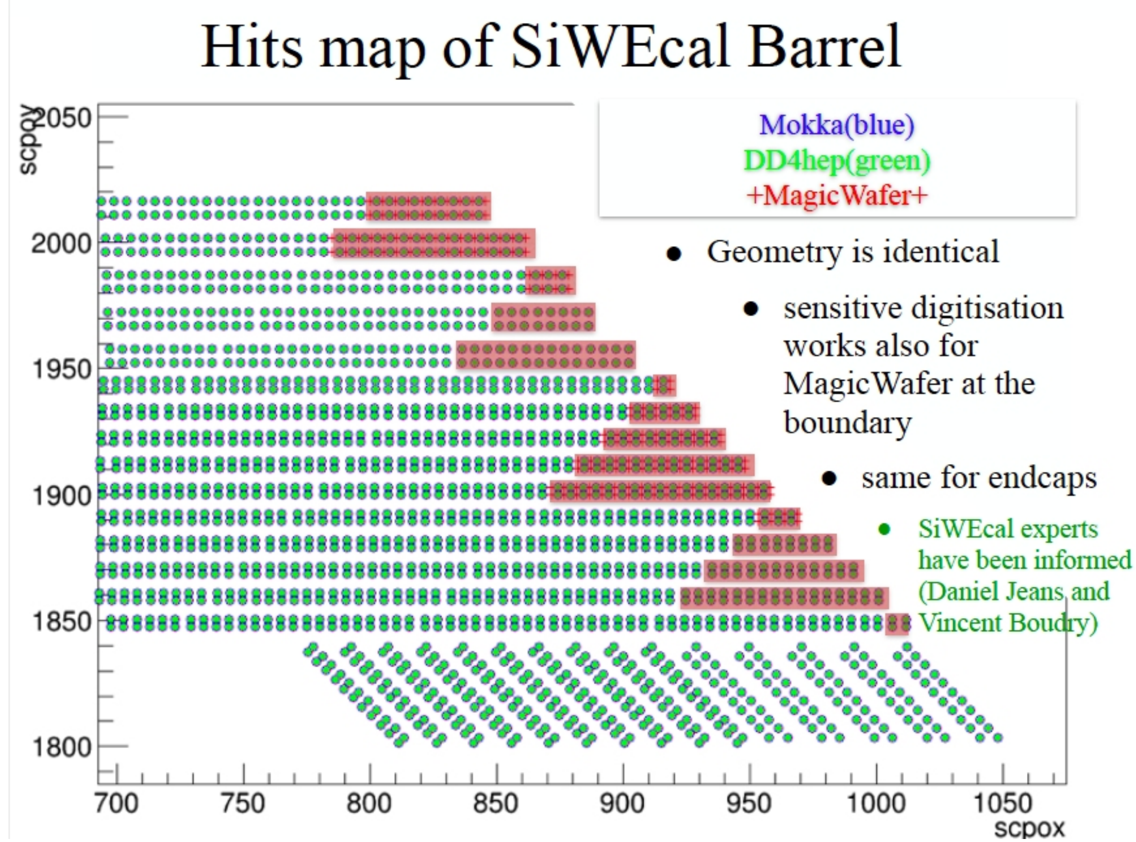
N.Nikiforou

- DDMarlinPandora rewrite of MarlinPandora using DD4hep
- can run Pandora as before

validating the new DD4hep model

S.Lu

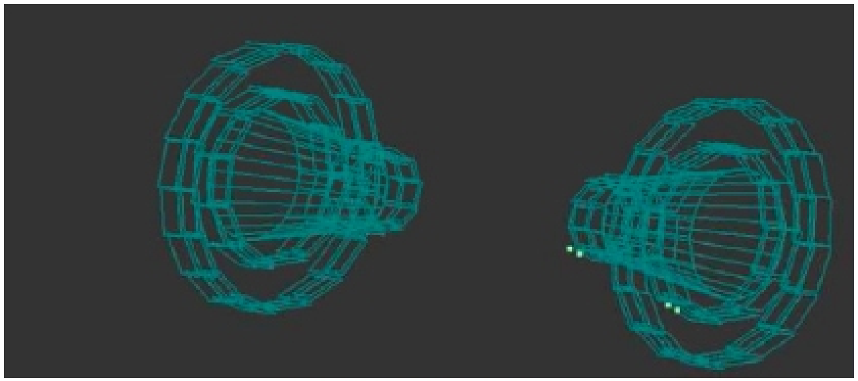


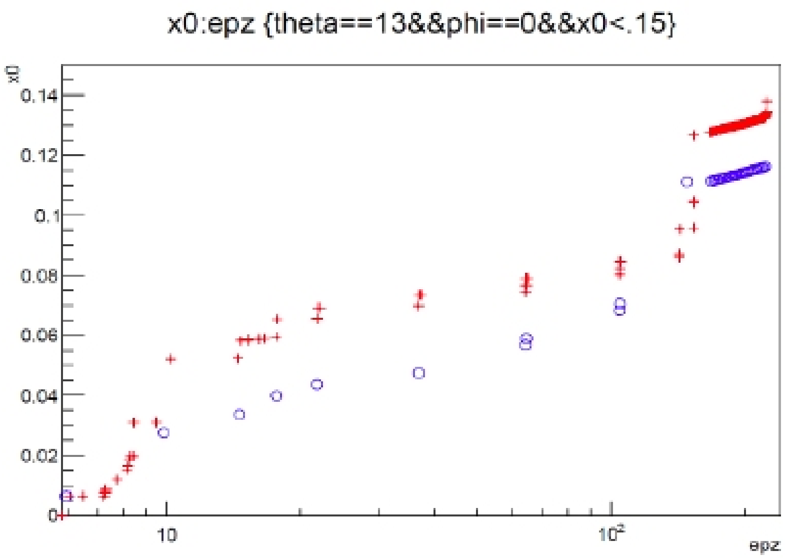


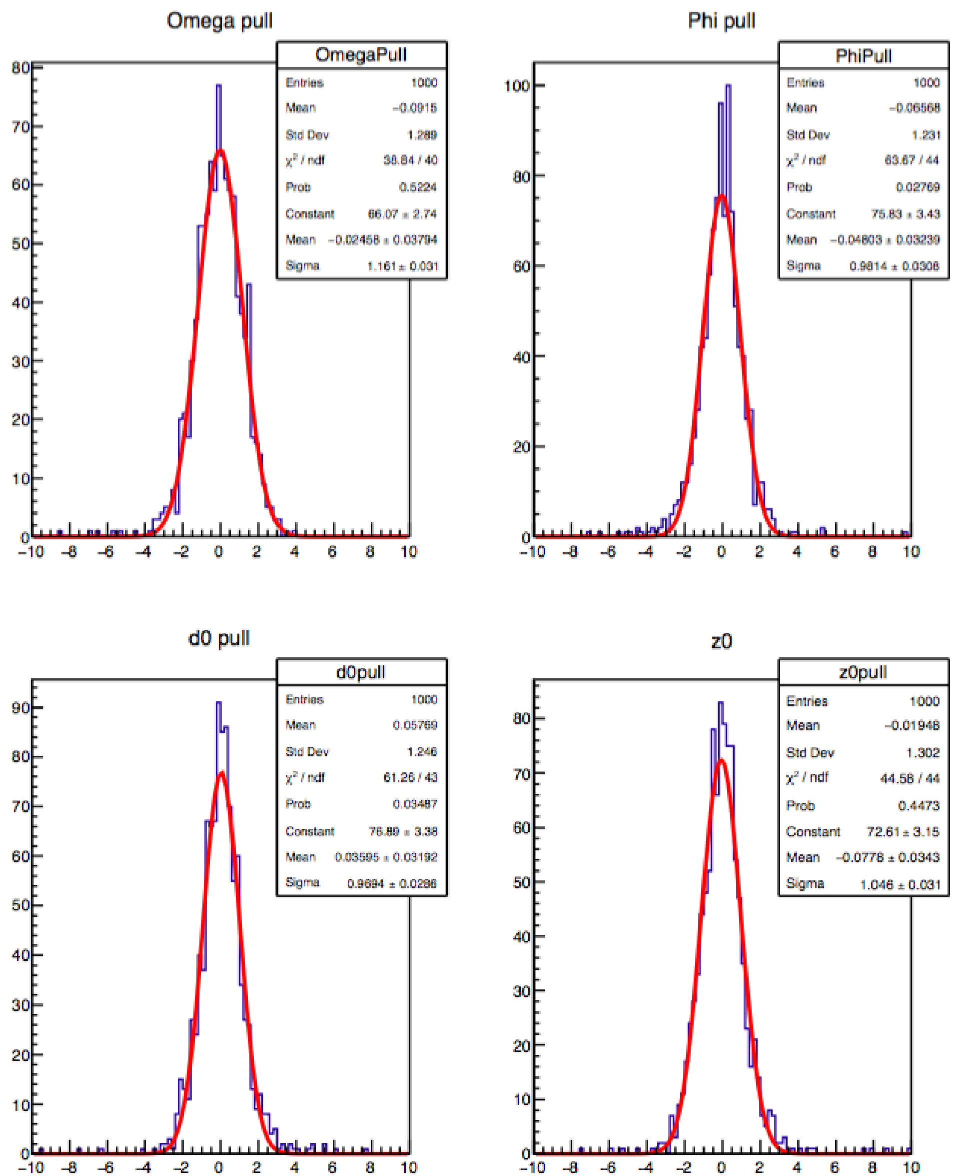
- quite some validation done by software experts, e.g. using hit maps, hit energy distributions, hit positions,.....
- => a detailed validation will have to be done by sub-detector experts

validating the new DD4hep model

Y.Voutsinas

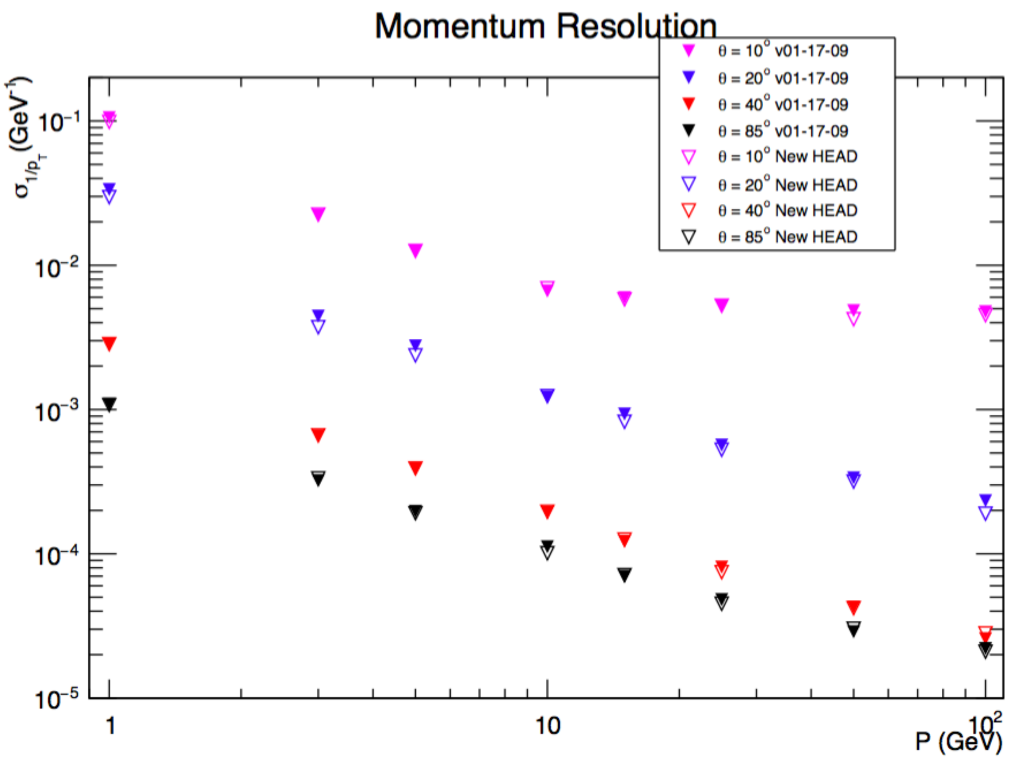


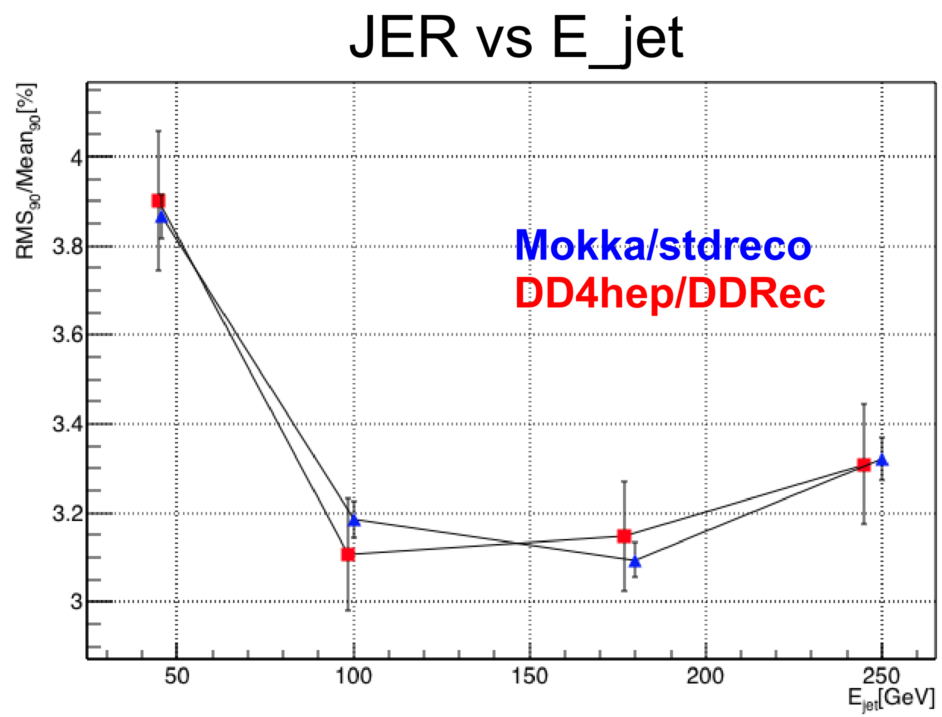




- for tracking check:
  - material budget
  - track parameter pulls
  - ...

Tracking/PFA performance w/ DD4hep





- now have reached for the first time the same performance with new DD4hep based software chain
- further validation and cross checks needed...

Summary & Outlook

- iLCSoft provides the software tools for the Linear Collider community (and beyond) for the development of
  - simulation models
  - reconstruction algorithms
  - analysis tools
- combined detector and software performance achieved design goals for ILD at time of TDR
- recently developed new - more generic - software tools: DD4hep/DDRec, generic tracking, HLR,...

  in the process of having as much as possible common software in the LC community
- new simulation/reconstruction chain now starts to achieve the same performance as the 'old' one
- will continue the validation process and eventually use for next round of ILD detector optimization

additional material

automatic material averaging for surfaces

- material properties are averaged along normal of the surface
- along given thicknesses

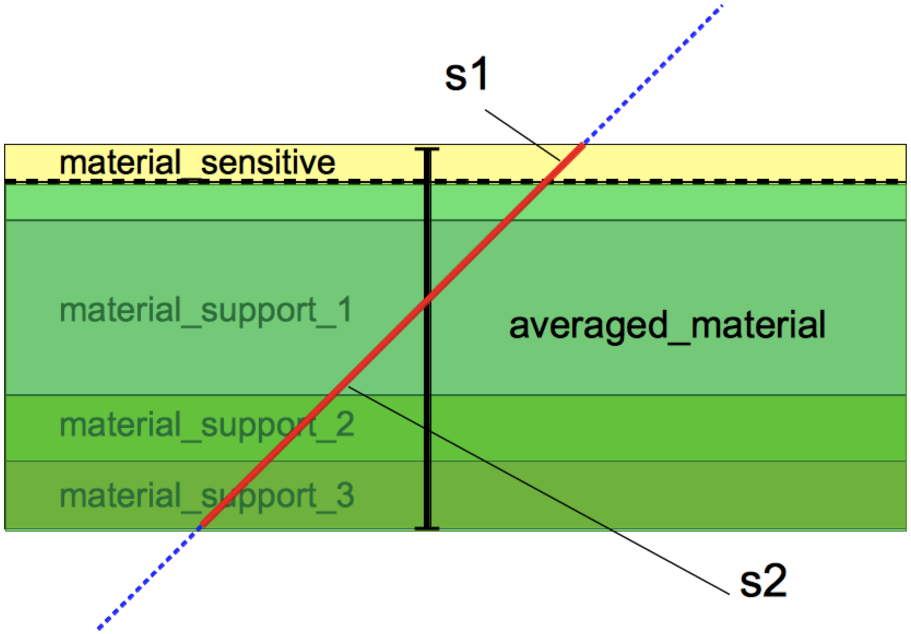
$$< A > = \left(\sum_i^N \rho_i t_i\right) / \left(\sum_i^N \rho_i \frac{t_i}{A_i}\right)$$

$$< Z > = \left(\sum_i^N \rho_i \frac{t_i Z_i}{A_i}\right) / \left(\sum_i^N \rho_i \frac{t_i}{A_i}\right)$$

$$< \rho > = \left(\sum_i^N \rho_i t_i\right) / \left(\sum_i^N t_i\right)$$

$$< X_0 > = \left(\sum_i^N t_i\right) / \left(\sum_i^N \frac{t_i}{X_{0i}}\right)$$

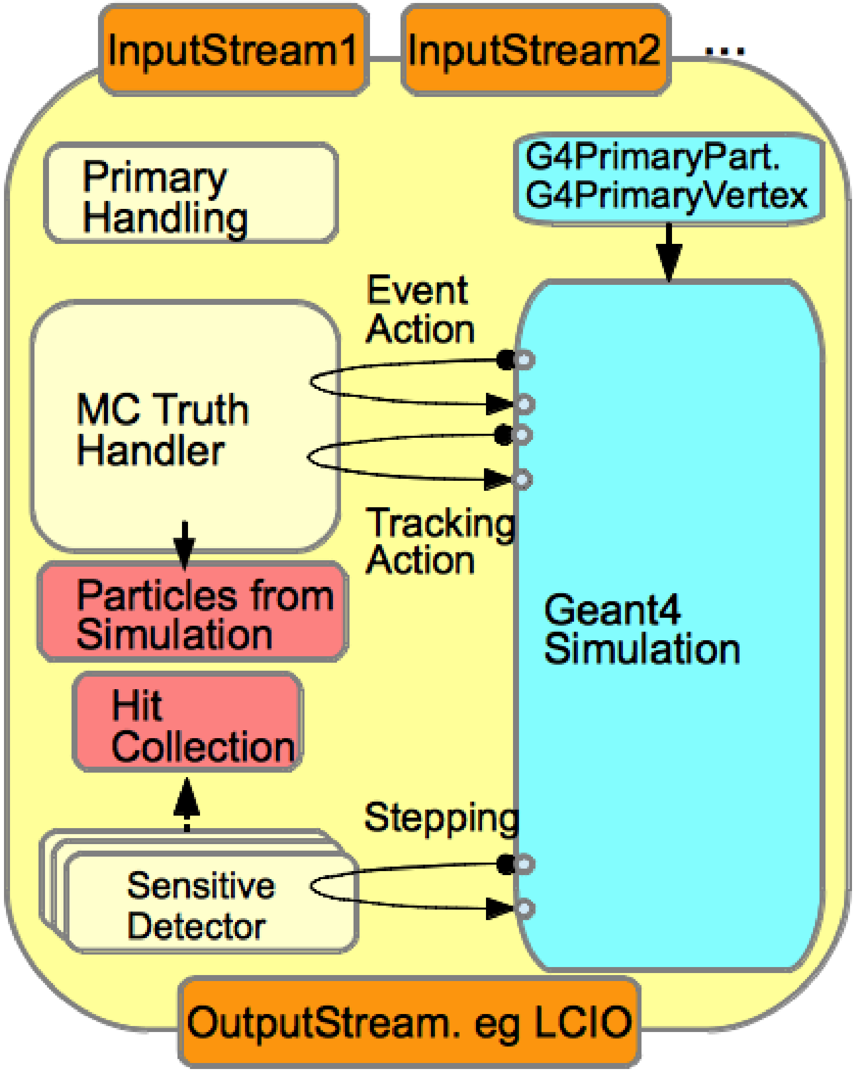
$$< \lambda > = \left(\sum_i^N t_i\right) / \left(\sum_i^N \frac{t_i}{\lambda}\right)$$



- roughly equivalent to individual materials for Bethe-Bloch
- identical for multiple scattering

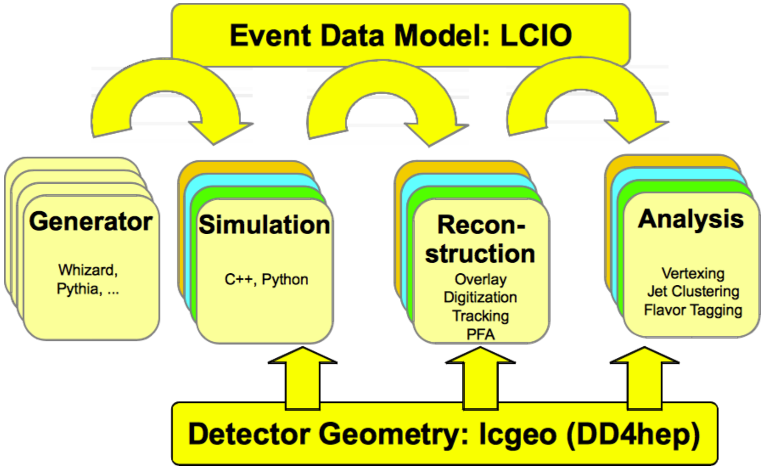
DDG4 - gateway to Geant4

- fully configurable binding to Geant4 full simulation
- standard input/output formats: LCIO, stdhep, HepEvt, HepMC,...
  - can easily add new ones
- Modules:
  - event overlay
  - IP-smearing
  - MC-Truth link
    - every hit knows its truth history
  - user defined
- can run as Python or C++ program (or ROOT macro)
- supports multithreading



ILC Software

- LC community has a long tradition of collaborating on common software tools
  - for ILC up to 4 different detector concepts in 3 regions and test beams and also CLICdp
- allow users to share core software tools an focus an algorithm and analysis development => do not re-invent the wheel



- basic strategy:
  - use well defined and agreed upon interfaces
  - keep it simple (as simple as possible but no simpler)
  - be as light weight, modular and flexible as possible
- followed this strategy over the years in several projects
  - EUDET, AIDA, AIDA2020, (HSF)
  - recently scope also partly extended to beyond LC:
    - LHC, FCC, CEPC and neutrino